Supply Chain Network Design with Concave Costs: Theory and Applications

by

Ahmed Saif

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Management Sciences

Waterloo, Ontario, Canada, 2015

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Many practical decision models can be formulated as concave minimization problems. Supply chain network design problems (SCNDP) that explicitly account for economies-of-scale and/or risk pooling often lead to mathematical problems with a concave objective and linear constraints. In this thesis, we propose new solution approaches for this class of problems and use them to tackle new applications.

In the first part of the thesis, we propose two new solution methods for an important class of mixed-integer concave minimization problems over a polytope that appear frequently in SCNDP. The first is a Lagrangian decomposition approach that enables a tight bound and a high quality solution to be obtained in a single iteration by providing a closed-form expression for the best Lagrangian multipliers. The Lagrangian approach is then embedded within a branch-and-bound framework. Extensive numerical testing, including implementation on three SCNDP from the literature, demonstrates the validity and efficiency of the proposed approach. The second method is a Benders approach that is particularly effective when the number of concave terms is small. The concave terms are isolated in a low-dimensional master problem that can be efficiently solved through enumeration. The subproblem is a linear program that is solved to provide a Benders cut. Branch-and-bound is then used to restore integrality if necessary. The Benders approach is tested and benchmarked against commercial solvers and is found to outperform them in many cases.

In the second part, we formulate and solve the problem of designing a supply chain for chilled and frozen products. The cold supply chain design problem is formulated as a mixed-integer concave minimization problem with dual objectives of minimizing the total cost, including capacity, transportation, and inventory costs, and minimizing the global warming impact that includes, in addition to the carbon emissions from energy usage, the leakage of high global-warming-potential refrigerant gases. Demand is modeled as a general distribution, whereas inventory is assumed managed using a known policy but without explicit formulas for the inventory cost and maximum level functions. The Lagrangian approach proposed in the first part is combined with a simulation-optimization approach to tackle the problem. An important advantage of this approach is that it can be used with different demand distributions and inventory policies under mild conditions. The solution approach is verified through extensive numerical testing on two realistic case studies from different industries, and some managerial insights are drawn.

In the third part, we propose a new mathematical model and a solution approach for the SCNDP faced by a medical sterilization service provider serving a network of hospitals. The sterilization network design problem is formulated as a mixed-integer concave minimization program that incorporates economies of scale and service level requirements under stochastic demand conditions, with the objective of minimizing long-run capacity, transportation, and inventory holding costs. To solve the problem, the resulting formulation is transformed into a mixed-integer second-order cone programming problem with a piecewise-linearized cost function. Based on a realistic case study, the proposed approach was found to reach high quality solutions efficiently. The results reveal that significant cost savings can be achieved by consolidating sterilization services as opposed to decentralization due to better utilization of resources, economies of scale, and risk pooling.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor Professor Samir Elhedhli for his invaluable guidance and support. The period I spent working under his supervision was the most rewarding and enjoyable in my life. I learned a lot from his technical expertise and professional ethics, but above all from his exemplary personal character. His dedicated guidance and generous support, in terms of time, effort and finance, were the decisive factors in my ability to reach this point. I could not ask for a more knowledgable, committed or companionate advisor in my Ph.D. journey.

I would like also to thank my supervisory committee, Professors Jim Bookbinder, Fatih Safa Erenay, and Ali Elkamel and the former member Professor David Fuller for the time and effort they dedicated to help me throughout my degree. Their sincere advice and insightful recommendations contributed immensely to my successful Ph.D. pursuit. For Professor Fuller, I wish him a long and joyous retirement. I also thank Professor Timothy Chan, my external examiner, for his time and thoughtful questions.

I was honoured to be among the first cohort of students in the WatOpt lab and is extremely grateful to be surrounded by this group of bright minds and joyful spirits. I am particularly grateful to Professor Fatma Gzara, the co-founder and director of the lab, for her exceptional and relentless support. Professor Gzara's comments and suggestions during the group meetings and presentations constituted the bases for several ideas in this thesis. Moreover, she provided me with some key materials that I used in the topic of sterilization network design. I got an exceptional opportunity to work under Professor Gzara's supervision in three industry internships and the guidance I received from her was invaluable. I am also thankful for the help and encouragement I received from my colleagues in these internships, especially Da Lu, Daniel Bayer, Burak Yildiz, and Ugur Yildiz. I learned many new skills and concepts through working with them.

My experience at the University of Waterloo was made successful and enjoyable through the hard work of many people, including the instructors of the courses I enrolled in, to whom I owe what I know today. I am grateful to Professors Frank Safayeni, Hossein Abouee-Mehrizi, Stephen Vavasis, and again, Samir Elhedhli, Fatma Gzara and David Fuller for making difficult and dry topics so easy and interesting to learn. I would like also to thank the administrative team in the Department of Management Sciences for their cooperative attitude and hard work.

Lastly, I would like to thank my family for all their love and encouragement. For my parents who raised me with a love of knowledge and supported me in all my pursuits. And most of all for my loving, supportive, encouraging, and patient wife Mayada and my lovely daughters Arwa, Reem and Sara. I was able to got that far only because you were beside me. Thank you.

Table of Contents

Author's Declaration ii

Abstract iii

Acknowledgements v

List of Tables x

List of Figures xii

1 Solution approaches for concave minimization over a polytope 1

1.1 Introduction 1

1.2 A Lagrangian approach for concave minimization 4

1.2.1 Lagrangian decomposition 4

1.2.2 Branch-and-bound 10

1.2.3 Computational results 13

1.3 Applications to supply chain design problems 15

1.3.1 The production-transportation problem 15

1.3.2 The plant location and technology acquisition problem 18

1.3.3 The location-inventory problem 23

1.4 A Benders approach for concave minimization 37

1.4.1 Benders decomposition 37

1.4.2 Numerical testing 40
1.5 Conclusions 43

2 Cold supply chain design: a simulation-optimization approach 44
2.1 Introduction 44
2.2 Literature review 46
2.2.1 Supply chain design with environmental considerations 46
2.2.2 Simulation-optimization methods for supply chain design problems 48
2.3 Effect of refrigerant gases in cold supply chains 49
2.4 Cost and global warming effect of cold storage warehouses 50
2.5 Model formulation 51
2.6 Solution method 54
2.6.1 Lagrangian decomposition 54
2.6.2 Branch-and-bound 64
2.6.3 Concavity of the inventory functions 65
2.6.4 Special cases with known inventory functions 66
2.6.5 Simulation-optimization algorithm 67
2.7 Numerical Testing 69
2.7.1 Case I: Maple Leaf Foods 69
2.7.2 Case II: A cold supply chain network for vaccines in Ontario 74
2.7.3 Case III: A hypothetical case 77
2.8 Conclusions 84

3 Sterilization network design 90
3.1 Introduction 90
3.2 The sterilization cycle 93
3.3 Problem description 96
3.4 Model formulation 97

3.5 Solution method 100
3.6 Numerical testing 102
3.7 Conclusions 109

Conclusions and future directions **110**

Bibliography **112**

List of Tables

1.1 Test functions for the Lagrangian approach from [65] 13

1.2 Results for the Lagrangian approach test problems from [65] 14

1.3 Results for the Lagrangian approach test problems from [53] 15

1.4 Results for the production-transportation problem 17

1.5 Cost parameters of the plant location and technology acquisition problem . 21

1.6 Results for the plant location and technology acquisition problem 22

1.7 Results for the location-inventory problem (88 node dataset) 29

1.8 Results for the location-inventory problem (150 node dataset) 30

1.9 Results for the location-inventory problem (500 node dataset) 32

1.10 Results for the location-inventory problem (1000 node dataset) 33

1.11 Cost breakdown for the location-inventory problem (500 and 1000 node datasets) 34

1.12 Results for the location-inventory problem (500 and 1000 node dataset when the demand mean and variance are uncorrelated) 36

1.13 Test functions for the Benders approach 40

1.14 Results for the Benders approach 41

2.1 Average daily demand in the MLF case 71

2.2 Product characteristics in the MLF case 71

2.3 Cost and GHG emissions for shipping products between DCs and retailers in the MLF case 73

2.4 Optimal solutions for the MLF case 74
2.5 Distance between cities in the vaccination network case 75
2.6 Annual demand for vaccines in the vaccination network case 76
2.7 Optimal solutions for the vaccination network case 77
2.8 Demand parameters in the hypothetical case 78
2.9 Truck characteristics in the hypothetical case 79
2.10 Results for the EOQ policy in the hypothetical case 80
2.11 Results for the (r,nQ) policy in the hypothetical case 82

3.1 RMD net characteristics in the SND problem 102
3.2 Sterilization resources in the SND problem 102
3.3 Estimated number of surgeries per year in the SND problem 103
3.4 Distance between demand points in the SND problem 104
3.5 Results for the full consolidation scheme in the SND problem 105

List of Figures

1.1 Illustration of Corollary 2 9
1.2 Illustrative example of the branch-and-bound algorithm 12

2.1 Components of the total cost and GHG emissions of the cold supply chain 54
2.2 The solution method for the cold supply chain design problem 86
2.3 The trade-off between cost and emissions in the MLF case 87
2.4 Breakdown of the cost components in the MLF case 87
2.5 The trade-off between cost and emissions in the vaccination network case . 88
2.6 Breakdown of the cost components in the vaccination network case 88
2.7 Difference between simplified and actual EOQ models in the hypothetical case 89

3.1 The Sterilization cycle 94
3.2 Cost breakdown for the base case scenario in the SND problem 107
3.3 Cost breakdown for the zero capacity cost scenario in the SND problem . . 108

Chapter 1

Solution approaches for concave minimization over a polytope

1.1 Introduction

Concave minimization is an important class of global optimization problems with many applications in production planning [80], facility location [39], and network flow [59]. Concave and quasi-concave terms in the objective function typically arise in models with economies of scale or fixed charges, but can also result from transforming other types of problems such as linear complementary [75], zero-one integer [83], and bilinear programming [97].

The difficulty of solving concave minimization problems stems primarily from the fact that local optima may not be globally optimal, and thus, applying efficient convex optimization methods can not guarantee global optimality. Concave minimization problems have been, in general, proved to be **NP**-hard, even the special case of minimizing a negative quadratic function over a hypercube [82]. Despite this worst-case performance, many methods for concave minimization perform well in practice. The reader is referred to the textbooks of Horst and Tuy [66] and Floudas [52] for a detailed account of global optimization methods, and the review papers of Floudas and Gounaris [54] and Pardalos et al. [81] for recent advances.

One of the most successful methods to solve global optimization problems is branch-and-bound (B&B), in which the feasible set is partitioned into successively refined regions through branching, while partitioned regions are eliminated if they are proven to be infeasible or suboptimal. As expected, B&B methods differ primarily in two aspects: the partitioning technique and the bounding scheme. Partitioning techniques include: conical partitioning [101], simplicial partitioning [64] and rectangular partitioning [48]. Rectangular partitioning is particularly suitable for separable problems since the convex envelope is easy to compute over a rectangular set. Bounding schemes include linear underestimation [48],[92], [87], [12], Lagrangian relaxation [47], [19], [40], [11] and outer approximation [62], [22], among others.

Most modern global optimization approaches combine B&B with cutting planes, domain reduction and outer approximation techniques to enhance their performance [95]. One of the most successful approaches proposed for both continuous and discrete global optimization problems is the polyhedral branch-and-cut approach that exploits convexity in order to generate polyhedral cutting planes and relaxations for multivariate nonconvex problems. This approach is the basis of the commercial solver *Baron* [94]. Another widely used B&B-based mixed-integer global optimization solver is *Couenne* (Convex Over- and Under-ENvelopes for Non-linear Estimation), which implements linearization, bound reduction, and branching methods within a spacial B&B framework [18].

In this chapter, we focus on problems having concave cost functions defined over linear combinations of the decision variables and with a polyhedral feasible region. In section 1.2.1 we propose a Lagrangian approach to solve problems of this class. We use Lagrangian decomposition, a variant of Lagrangian relaxation that duplicates certain variables to separate the feasible set; and then relaxes the linking constraints [58]. Lagrangian decomposition was applied to decompose large-scale concave minimization problems in [69] and [70]. However, we use it in a different manner to isolate the nonconvex terms in the objective function so that the problem is decomposed into a linear objective subproblem over the original constraints and a number of single-variable easily-solvable concave minimization subproblems. Surprisingly, for this decomposition we could provide a closed-form expression for the optimal Lagrangian multipliers, allowing the direct calculation of the Lagrangian bound without resorting to classical iterative methods such as subgradient or cutting plane methods. In addition, the solution of the first subproblem is feasible to the original problem, providing a feasible solution and an upper bound. This direct calculation of the multipliers is not common in general Lagrangian approaches and is a special feature of the decomposition we apply. Therefore, after solving the subproblems only once, a feasible solution and a Lagrangian bound are obtained. In most cases, the feasible solution is of high quality. In cases where it is not, a branch-and-bound algorithm is utilized to close the optimality gap.

Concave minimization problems in supply chain network design are of particular interest as they provide a wealth of models that fit the framework we study. While classical supply chain design models have predominantly linear costs, some have concave objective functions. Concave costs often result from economies of scale with respect to capacity of the facilities [35], [39], [92] or transportation between facilities [43], and from combining location and inventory decisions [37], [25]. Inventory cost is generally known to be concave in the demand served, whereas safety stock cost is concave due to the risk-pooling effect when demand is stochastic [45]. In section 1.3, we implement the proposed Lagrangian approach to solve three supply chain design problems from the literature: the production-transportation problem [71], the plant location and technology acquisition problem [35] and the location-inventory problem [37]. For each problem, extensive numerical testing is conducted to verify its validity and efficiency.

A new Benders approach is proposed in section 1.4, for the same class of problems, that is particularly effective when the number of concave terms in the objective function is small. The Benders subproblem is linear integer while the master problem is a low-dimensional concave minimization problem with continuous variables that can be solved by enumeration. The Benders approach is embedded within a classical B&B algorithm to obtain an integer solution. The approach is tested using different functions and benchmarked against the state-of-the-art commercial solvers *Baron* and *Couenne*.

The main contributions of this chapter are:

1. A novel Lagrangian approach for an important class for concave minimization problems with linear constraints the appears frequently in supply chain design models. The distinctive advantage of the approach is that the optimal Lagrangian multipliers are reachable using closed-form expressions, enabling the Lagrangian bound and a high quality feasible solution to be obtained in a single step. The proposed Lagrangian approach is then embedded in a branch-and-bound algorithm to close the optimality gap. Numerical testing, including three practical supply chain design problems from the literature, reveals the validity and efficiency of the proposed approach.

2. A Benders approach for concave minimization problems that shifts the objective function concavity to a low-dimensional master problem that is solved using iterative enumeration. The approach is particularly suitable for problems with a small number of concave terms in the objective function. Numerical testing shows that the proposed approach outperforms some state-of-the-art commercial solvers.

1.2 A Lagrangian approach for concave minimization

1.2.1 Lagrangian decomposition

In this section we propose a Lagrangian decomposition approach to solve the concave minimization problem

$$[\mathcal{P}]: \qquad \min_{x \in X} \quad c^T x + \sum_{i=1}^{m} f_i(d_i^T x),$$

where X is a polyhedral set that is assumed bounded, $c, d \in \mathbb{R}^n$, $f(x)$ are concave over X, and $x_k \in \mathbb{Z}$, $\forall\, k \in K \subseteq \{1, 2, \ldots, n\}$.

We begin by introducing the auxiliary variables $y_i = d_i^T x$, so $\mathcal{P}$ can be written as

$$\begin{aligned} \min\ & c^T x + \sum_{i=1}^{m} f_i(y_i), \\ \text{s.t.}\ & y_i = d_i^T x, \qquad i = 1, \ldots, m, \\ & \underline{y_i} \le y_i \le \overline{y_i}, \\ & x \in X, \end{aligned} \tag{1.1}$$

where $\underline{y_i}$ and $\overline{y_i}$ are obtained by solving $\min\limits_{x \in X} d_i^T x$ and $\max\limits_{x \in X} d_i^T x$ which correspond to extreme points of X. Since X is assumed bounded, both $\underline{y}_i$ and $\overline{y}_i$ are finite. Applying Lagrangian relaxation on (1.1) with multipliers $\lambda_i, i = 1, \ldots, m$ leads to $m + 1$ subproblems:

$$[\mathcal{SP}1]: \begin{aligned} \min\ & (c + \sum_{i=1}^{m} \lambda_i d_i)^T x, \\ \text{s.t.}\ & x \in X. \end{aligned} \qquad \text{and} \qquad [\mathcal{SP}2_i]: \begin{aligned} \min\ & f_i(y_i) - \lambda_i y_i, \\ \text{s.t.}\ & \underline{y_i} \le y_i \le \overline{y_i}. \end{aligned}$$

$\mathcal{SP}1$ has a linear objective and is solved either as an LP or MIP, whereas $\mathcal{SP}2_i$ are single-variable concave minimization problems whose solutions are at one of the two extreme points $\underline{y_i}$ or $\overline{y_i}$, *i.e.*, the solution of $\mathcal{SP}2_i$, $i = 1, \ldots, m$ is $\min \Big\{ f_i(\underline{y_i}) - \lambda_i \underline{y_i}, f_i(\overline{y_i}) - \lambda_i \overline{y_i} \Big\}$. The solution of $\mathcal{SP}1$, on the other hand, is

$$\min_{h=1,\ldots,H} (c^T + \sum_{i=1}^{m} \lambda_i d_i^T) x^h,$$

where $\{1,\ldots,H\}$ is the index set of the vertices of X if $\mathcal{SP}1$ is LP and of the feasible solutions to $\mathcal{SP}1$ if it is MIP. Therefore, the Lagrangian bound is given by

$$LR(\lambda) = \min_{h=1,\ldots,H} (c^T + \sum_{i=1}^{m} \lambda_i d_i^T) x^h + \sum_{i=l}^{m} \min\big(f_i(\underline{y_i}) - \lambda_i \underline{y_i}, f_i(\overline{y_i}) - \lambda_i \overline{y_i}\big).$$

The Lagrangian dual problem is $\max\limits_{\lambda} LR(\lambda)$. Its linear programming reformulation, known as the dual master problem, is:

$$\begin{aligned}
[\mathcal{DMP}]:\quad & \max\ \theta_0 + \sum_{i=1}^{m} \theta_i, && \\
\text{s.t.}\quad & \theta_0 - \sum_{i=1}^{m} d_i^T x^h \lambda_i \le c^T x^h, && h = 1,\ldots,H, \\
& \theta_i + \underline{y_i}\lambda_i \le f_i(\underline{y_i}), && i = 1,\ldots,m, \\
& \theta_i + \overline{y_i}\lambda_i \le f_i(\overline{y_i}), && i = 1,\ldots,m.
\end{aligned}$$

This problem is usually solved using a cutting plane method. Its dual, $\mathcal{MP}$, is the Dantzig-Wolfe master problem, which is solved by column generation:

$$\begin{aligned}
[\mathcal{MP}]:\ \min\ & \sum_{h=1}^{H} c^T x^h \alpha_h + \sum_{i=1}^{m} f_i(\underline{y_i})\beta_i + \sum_{i=1}^{m} f_i(\overline{y_i})\gamma_i && \\
\text{s.t.}\quad & \sum_{h=1}^{H} \alpha_h = 1, && \\
& \beta_i + \gamma_i = 1, && i = 1,\ldots,m, \\
& \underline{y_i}\beta_i + \overline{y_i}\gamma_i = \sum_{h=1}^{H} (d_i^T x^h)\alpha_h, && i = 1,\ldots,m, \\
& \alpha_h, \beta_i, \gamma_i \ge 0. &&
\end{aligned}$$

Surprisingly and unprecedentedly, we are able to give a closed-form expression for the optimal Lagrangian multipliers. The following proposition provides the details.

Proposition 1 *The optimal multipliers λ^* are given by*

$$\lambda_i^* = \frac{f_i(\overline{y_i}) - f_i(\underline{y_i})}{\overline{y_i} - \underline{y_i}}, \quad i = 1,\ldots,m.$$

Proof: *First, let us solve the subproblems for* λ_i^*, $i = 1, \ldots, m$. *Let the solution of* $\mathcal{SP}1$, $\min_{x \in X}(c^T + \sum_{i=1}^m \lambda_i^* d_i^T)x$, *be* x^{h^*}.

The solution of $\mathcal{SP}2_i$ *is*

$$\begin{aligned} & \min\left(f_i(\overline{y}) - \lambda_i^* \overline{y_i}, f_i(\underline{y}) - \lambda_i^* \underline{y_i}\right) \\ = & \min\left(\frac{\overline{y_i} f_i(\underline{y_i}) - \underline{y_i} f_i(\overline{y_i})}{\overline{y_i} - \underline{y_i}}, \frac{\overline{y_i} f_i(\underline{y_i}) - \underline{y_i} f_i(\overline{y_i})}{\overline{y_i} - \underline{y_i}}\right) \\ = & \frac{\overline{y_i} f_i(\underline{y_i}) - \underline{y_i} f_i(\overline{y_i})}{\overline{y_i} - \underline{y_i}}. \end{aligned}$$

Thus, the Lagrangian lower bound corresponding to λ_i^* *is*

$$LB(\lambda^*) = (c^T + \sum_{i=1}^m \lambda_i^* d_i^T)x^{h^*} + \sum_{i=1}^m \frac{\overline{y_i} f_i(\underline{y_i}) - \underline{y_i} f_i(\overline{y_i})}{\overline{y_i} - \underline{y_i}}. \tag{1.2}$$

Second, let us consider the relaxed dual master problem $\mathcal{RDMP}$ *corresponding to* $H = \{h^*\}$*:*

$$\begin{aligned} [\mathcal{RDMP}]: \quad & \max\ \theta_0 + \sum_{i=1}^m \theta_i \\ & \text{s.t.} \quad \theta_0 - \sum_{i=1}^m d_i^T x^{h^*} \lambda_i \le c^T x^{h^*}, && (1.3) \\ & \qquad \theta_i + \underline{y_i}\lambda_i \le f_i(\underline{y_i}), \qquad i = 1, \ldots, m, && (1.4) \\ & \qquad \theta_i + \overline{y_i}\lambda_i \le f_i(\overline{y_i}), \qquad i = 1, \ldots, m, && (1.5) \end{aligned}$$

and let us check whether the solution $(\lambda_i = \lambda_i^*, \theta_0 = (c^T + \sum_{i=1}^m \lambda_i^* d_i^T)x^{h^*}, \theta_i = \frac{\overline{y_i} f_i(\underline{y_i}) - \underline{y_i} f_i(\overline{y_i})}{\overline{y_i} - \underline{y_i}})$ *is feasible to* $\mathcal{RDMP}$. *By substituting in (1.3), (1.4) and (1.5), all these constraints are found to be satisfied as equalities. Let* $y_i^* = d_i^T x^{h^*}$. *The objective function value of* $\mathcal{RDMP}$ *corresponding to the proposed solution is*

$$\begin{aligned} \mathbf{v}[\mathcal{RDMP}] &= (c^T + \sum_{i=1}^m \lambda_i^* d_i^T)x^{h^*} + \sum_{i=1}^m \frac{\overline{y_i} f_i(\underline{y_i}) - \underline{y_i} f_i(\overline{y_i})}{\overline{y_i} - \underline{y_i}} \\ &= c^T x^{h^*} + \sum_{i=1}^m \left(\frac{f_i(\overline{y_i}) - f_i(\underline{y_i})}{\overline{y_i} - \underline{y_i}}\right) y_i^* + \sum_{i=1}^m \frac{\overline{y_i} f_i(\underline{y_i}) - \underline{y_i} f_i(\overline{y_i})}{\overline{y_i} - \underline{y_i}} \\ &= c^T x^{h^*} + \sum_{i=1}^m f_i(\underline{y_i}) \frac{\overline{y_i} - y_i^*}{\overline{y_i} - \underline{y_i}} + \sum_{i=1}^m f_i(\overline{y_i}) \frac{y_i^* - \underline{y_i}}{\overline{y_i} - \underline{y_i}}. \end{aligned}$$

To prove the optimality of the proposed solution, we resort to the dual of $\mathcal{RDMP}$, the restricted master problem

$$
\begin{aligned}
[\mathcal{RMP}]:\ \min\ & c^T x^{h^*}\alpha_h^* + \sum_{i=1}^m f_i(\underline{y_i})\beta_i + \sum_{i=1}^m f_i(\overline{y_i})\gamma_i \\
\text{s.t.}\ \ & \alpha_h^* = 1, && && (1.6)\\
& \beta_i + \gamma_i = 1, && i = 1,\ldots,m \\
& \underline{y_i}\beta_i + \overline{y_i}\gamma_i = d_i^T x^{h^*}\alpha_h^*, && i = 1,\ldots,m && (1.7)\\
& \alpha_h, \beta_i, \gamma_i \geq 0.
\end{aligned}
$$

By substituting the value of α_h^* from (1.6) in (1.7), $\mathcal{RMP}$ can be decomposed into m subproblems, each having a single-point feasible set that can be found by simultaneously solving the two equations $\beta_i + \gamma_i = 1$ and $\underline{y_i}\beta_i + \overline{y_i}\gamma_i = d_i^T x^{h^*}$. Thus, the solution of $\mathcal{RMP}$ is

$$
\begin{aligned}
\beta_i &= \frac{\overline{y_i} - y_i^*}{\overline{y_i} - \underline{y_i}} \\
\gamma_i &= \frac{y_i^* - \underline{y_i}}{\overline{y_i} - \underline{y_i}},
\end{aligned}
$$

which are nonnegative. Thus,

$$
\mathbf{v}[\mathcal{RMP}] = c^T x^{h^*} + \sum_{i=1}^m f_i(\underline{y_i})\frac{\overline{y_i} - y_i^*}{\overline{y_i} - \underline{y_i}} + \sum_{i=1}^m f_i(\overline{y_i})\frac{y_i^* - \underline{y_i}}{\overline{y_i} - \underline{y_i}}.
$$

By strong duality, since $\mathbf{v}[\mathcal{RDMP}] = \mathbf{v}[\mathcal{RMP}]$, the proposed solution is optimal to $\mathcal{RDMP}$. Furthermore, $LB(\lambda^*) \leq \mathbf{v}[\mathcal{DMP}]$. According to (1.2), $LB(\lambda^*) = \mathbf{v}[RDMP]$, therefore, $\mathbf{v}[\mathcal{RDMP}] = \mathbf{v}[\mathcal{DMP}] = LB(\lambda^*)$, the Lagrangian bound. Thus, λ^* are optimal. ■

Next, we note that $x^{h^*} \in X$ and so it is feasible to $\mathcal{P}$. Its corresponding objective $c^T x^{h^*} + \sum_{i=1}^m f_i(y_i^*)$ is an upper bound. The difference between this upper bound and the Lagrangian bound is given by

$$
\begin{aligned}
\delta &= c^T x^{h^*} + \sum_{i=1}^m f_i(y_i^*) - (c^T + \sum_{i=1}^m \frac{f_i(\overline{y_i}) - f_i(\underline{y_i})}{\overline{y_i} - \underline{y_i}} d_i^T) x^{h^*} - \sum_{i=1}^m \frac{\overline{y_i} f_i(\underline{y_i}) - \underline{y_i} f_i(\overline{y_i})}{\overline{y_i} - \underline{y_i}} \\
&= \sum_{i=1}^m \Big(f_i(y_i^*) - f_i(\overline{y_i})\frac{y_i^* - \underline{y_i}}{\overline{y_i} - \underline{y_i}} - f_i(\underline{y_i})\frac{\overline{y_i} - y_i^*}{\overline{y_i} - \underline{y_i}} \Big), \qquad (1.8)
\end{aligned}
$$

which quantifies how far x^{h^*} is from the optimal.

Corollary 1 *If $y_i^* = \underline{y_i}$ or $\overline{y_i}$, $\forall i$, x^{h^*} is optimal to $\mathcal{P}$.*

Proof: *By substituting in (1.8), we get the desired result.* ■

Corollary 2 *The optimality gap (δ) is approximately proportional to $f_i''(\underline{y_i})$.*

Proof: *Rearranging (1.8), we get*

$$\delta = \sum_{i=1}^{m} (y_i^* - \underline{y_i})(\rho_i - \pi_i), \tag{1.9}$$

where $\rho_i = \frac{f_i(y_i^*) - f_i(\underline{y_i})}{y_i^* - \underline{y_i}}$ is the slope of the line segment connecting the points $(\underline{y_i}, f_i(\underline{y_i}))$ and $(y_i^*, f_i(y_i^*))$, and $\pi_i = \frac{f_i(\overline{y_i}) - f_i(\underline{y_i})}{\overline{y_i} - \underline{y_i}}$ is the slope of the line segment connecting the points $(\underline{y_i}, f_i(\underline{y_i}))$ and $(\overline{y_i}, f_i(\overline{y_i}))$ (see Fig.1.1). Let f_i' and f_i'' be the first and second derivatives of f_i. By using the first three terms of the Taylor series expansion, $f_i(y_i^*)$ and $f_i(\overline{y_i})$ can be approximated as:

$$\begin{aligned} f_i(y_i^*) &\approx f_i(\underline{y_i}) + (y_i^* - \underline{y_i}) f_i'(\underline{y_i}) + \frac{1}{2}(y_i^* - \underline{y_i})^2 f_i''(\underline{y_i}), \\ f_i(\overline{y_i}) &\approx f_i(\underline{y_i}) + (\overline{y_i} - \underline{y_i}) f_i'(\underline{y_i}) + \frac{1}{2}(\overline{y_i} - \underline{y})^2 f_i''(\underline{y_i}). \end{aligned}$$

Thus,

$$\begin{aligned} \rho_i &= \frac{f_i(y_i^*) - f_i(\underline{y_i})}{y_i^* - \underline{y_i}} \approx f_i'(\underline{y_i}) + \frac{1}{2}(y_i^* - \underline{y_i}) f_i''(\underline{y_i}), \\ \pi_i &= \frac{f_i(\overline{y_i}) - f_i(\underline{y_i})}{\overline{y_i} - \underline{y_i}} \approx f_i'(\underline{y_i}) + \frac{1}{2}(\overline{y_i} - \underline{y_i}) f_i''(\underline{y_i}). \end{aligned}$$

which implies that

$$\begin{aligned} \delta &\approx \sum_{i=1}^{m} (y_i^* - \underline{y_i}) \left(f_i'(\underline{y_i}) + \frac{1}{2}(y_i^* - \underline{y_i}) f_i''(\underline{y_i}) - f_i'(\underline{y_i}) + \frac{1}{2}(\overline{y_i} - \underline{y_i}) f_i''(\underline{y_i}) \right) \\ &= \sum_{i=1}^{m} \frac{(y_i^* - \underline{y_i})(y_i^* - \overline{y_i})}{2} f_i''(\underline{y_i}). \end{aligned} \tag{1.10}$$

Therefore, the gap is (approximately) linearly proportional to the second derivative of the concave functions. ■

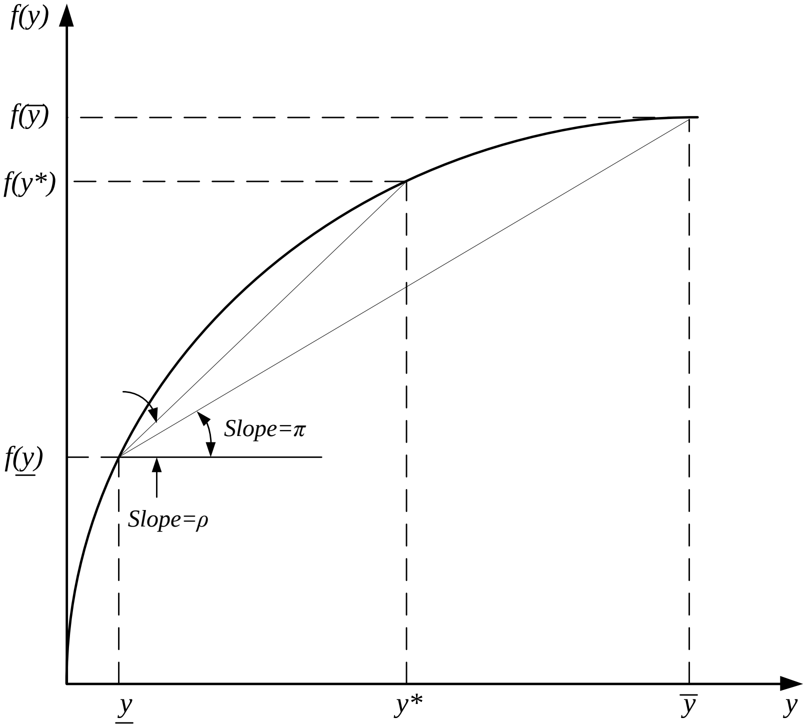

Figure 1.1: Illustration of Corollary 2

When $f_i''(y_i)$, $\forall i = 1, \dots, m$ is constant (*i.e.*, f_i is a quadratic function), the gap formula (1.10) is exact. In this case, it is easy to show that the gap is maximum at the midpoint between $\underline{y_i}$ and $\overline{y_i}$, with a maximum value of $\sum_{i=1}^{m} |\frac{f_i''(y_i)(\overline{y_i}-\underline{y_i})^2}{8}|$.

A final observation from (1.8) is that the optimality gap is a non-increasing function of $(\overline{y} - \underline{y})$. Thus, one way to reduce, and eventually close, this gap is by successively tightening the feasible range of the auxiliary variables y_i. A branch-and-bound algorithm that partitions the feasible region in this fashion is proposed in the next section for that purpose.

1.2.2 Branch-and-bound

To close the optimality gap, we embed the Lagrangian decomposition in a Falk and Soland's [48] type branch-and-bound algorithm. Although the original problem $\mathcal{P}$ in x_j is non-separable, the substitution $y = d^T x$ enables us to tackle it using the aforementioned algorithm. Branching is performed around the optimal solution of $\mathcal{SP}1$ using the auxiliary variables y_i, $i = 1, \dots, m$. At any node, upon solving $\mathcal{SP}1$ we get the vector x^* and calculate $y^* = d^T x^*$. The variables y_i^* , $i = 1, \dots, m$ are classified to two sets: *extreme* when $y_i^* = \underline{y_i}$ or $\overline{y_i}$, and *non-extreme* otherwise. If all the variables are extreme, the gap is guaranteed to close according to corollary 1, otherwise, a non extreme variable $y_{\hat{i}}$ is selected for branching. In one child node, we add the oblique cut $d_{\hat{i}}^T x \leq y_{\hat{i}}^*$ to $\mathcal{SP}1$ and $y_{\hat{i}} \leq y_{\hat{i}}^*$ to $\mathcal{SP}2_{\hat{i}}$, and in the other child node, we add $d_{\hat{i}}^T x \geq y_{\hat{i}}^*$ to $\mathcal{SP}1$ and $y_{\hat{i}} \geq y_{\hat{i}}^*$ to $\mathcal{SP}2_{\hat{i}}$ and solve using the Lagrangian approach. In every node, an upper bound is obtained from the feasible solution of $\mathcal{SP}1$ and the incumbent is updated if a better upper bound is found.

In any node, one of the following three outcomes is possible:

1. The gap is closed, in which case this branch is fathomed.

2. The gap is not closed, but the lower bound is higher than the incumbent, implying that further branching will not lead to better feasible solutions, so the branch is fathomed.

3. The gap is not closed, but the lower bound is lower than the incumbent. In this case, we continue branching using a non-extreme variable as described earlier.

Branching continues until all branches are fathomed. Falk and Soland have shown that, when the feasible region is a polyhedron, this branch-and-bound algorithm is complete in the sense that it finds the optimal solution of $\mathcal{P}$ after solving a finite number of nodes [48].

Let $M_s,\ s \in S$ be the partitioning subsets of the feasible region. For each subset (*i.e.*, node) we can get a lower and an upper bound using the Lagrangian approach, denoted $\mathcal{L}(M_s)$ and $\mathcal{U}(M_s)$, respectively. Then, $\mathcal{L} = \min\limits_{s\in S} \mathcal{L}(M_s),\ \mathcal{U} = \min\limits_{s\in S} \mathcal{U}(M_s)$ are the overall bounds, respectively. If at any iteration $\mathcal{L} = \mathcal{U}$, the branch-and-bound algorithm terminates.

To illustrate the branch-and-bound algorithm, let's consider the following example:

$$\begin{aligned} &\min\ -x_1^2 - x_2^2 - 2x_1x_2 - 9x_1 - x_2 \\ &\text{s.t.}\quad x \in X = \{x : 2x_1 + x_2 \le 6;\ 0 \le x_1 \le 2;\ 0 \le x_2 \le 4\}. \end{aligned}$$

The objective function is equivalent to $\min\ -9x_1 - x_2 - (x_1 + x_2)^2$, implying that $c = [-9, -1],\ d = [1, 1],\ f(y) = -y^2$. The example is displayed in figure 1.2. The extreme points of the feasible set are $\{(0, 0), (0, 4), (1, 4), (2, 2), (2, 0)\}$. The bounds $\underline{y}$ and $\overline{y}$ are obtained by solving $\min\limits_{x\in X} d^T x$ and $\max\limits_{x\in X} d^T x$, respectively, so $\underline{y} = 0$ and $\overline{y} = 5$.

At the root node, $\underline{y} = 0,\ \overline{y} = 5,\ \lambda^* = \frac{-25+0}{5-0} = -5$, and the subproblems solved are: $\mathcal{SP}1$: $\min\ -14x_1 - 6x_2,\ x \in X$, and $\mathcal{SP}2$: $\min\ 5y - y^2,$ s.t. $0 \le y \le 5$. The solutions are $x^*_{\mathcal{SP}1} = (2, 2)$, $\mathbf{v}[\mathcal{SP}1] = -40$, $\mathbf{v}[\mathcal{SP}2] = 0$, $y^* = 4$. This gives a lower bound (LB) of -40 and a feasible solution $(2, 2)$ of value -36 (the incumbent). As LB$\neq$ incumbent, we branch around $(2, 2)$ by forcing $d^T x = x_1 + x_2 \le y^* = 4$ in node 1, and $d^T x = x_1 + x_2 \ge y^* = 4$ in node 2.

In Node 1, $\underline{y} = 0,\ \overline{y} = 4,\ \lambda^* = \frac{-16+0}{4-0} = -4$, and the subproblems solved are: $\mathcal{SP}1$: $\min\ -13x_1 - 5x_2,\ x \in X_1 = \{x : x \in X,\ x_1 + x_2 \le 4\}$, and $\mathcal{SP}2$: $\min\ 4y - y^2,$ s.t. $0 \le y \le 4$. The solutions are $x^*_{\mathcal{SP}1} = (2, 2)$, $\mathbf{v}[\mathcal{SP}1] = -36$, $\mathbf{v}[\mathcal{SP}2] = 0$, $y^* = 4$. This gives a lower bound (LB) of -36 and a feasible solution $(2, 2)$ of value -36, so the incumbent remains unchanged. As LB=incumbent, the node is fathomed.

In Node 2, $\underline{y} = 4,\ \overline{y} = 5,\ \lambda^* = \frac{-25+16}{5-4} = -9$, and the subproblems solved are: $\mathcal{SP}1$: $\min\ -18x_1 - 10x_2,\ x \in X_2 = \{x : x \in X,\ x_1 + x_2 \ge 4\}$, and $\mathcal{SP}2$: $\min\ 9y - y^2,$ s.t. $4 \le y \le 5$. The solutions are: $x^*_{\mathcal{SP}1} = (1, 4)$, $\mathbf{v}[\mathcal{SP}1] = -58$, $\mathbf{v}[\mathcal{SP}2] = 20$, $y^* = 5$. This gives a lower bound (LB) of -38 and a feasible solution $(1, 4)$ with value -38, so the new incumbent equals -38. As LB=incumbent, the node is fathomed. So, $x^*_{\mathcal{P}} = (1, 4)$, $\mathbf{v}[\mathcal{P}] = -38$.

In Fig. 1.2, the areas X_1 and X_2 correspond to the feasible regions of the subproblems in nodes 1 and 2, respectively. The curve is the contour line of -38, the optimal value of $\mathcal{P}$.

This example, albeit trivial, sheds light on the functionality of the branch-and-bound algorithm. Moving from one partitioning level to the next splits the feasible region between the two child nodes, while keeping the optimal solution of the parent node in both.

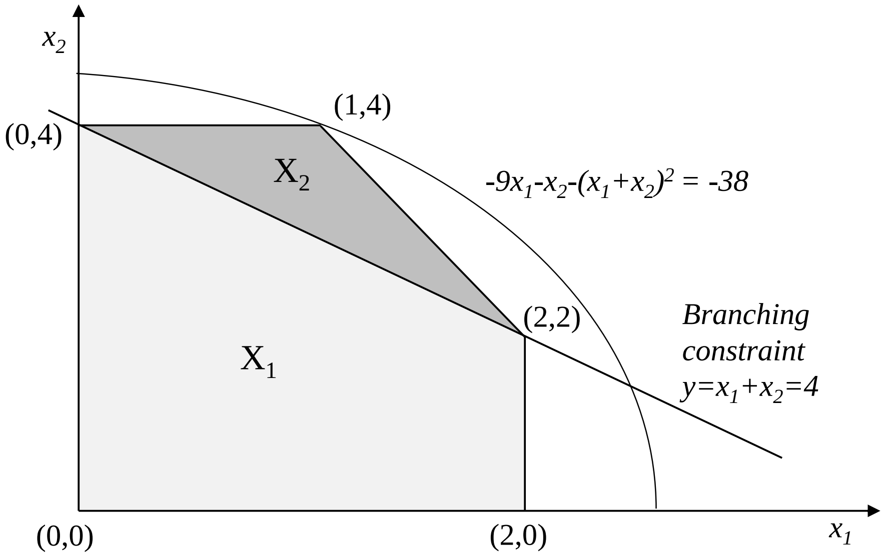
x_2
(0,4)
(1,4)
X_2
$-9x_1-x_2-(x_1+x_2)^2 = -38$
(2,2)
Branching constraint $y=x_1+x_2=4$
X_1
(0,0)
(2,0)
x_1

Figure 1.2: Illustrative example of the branch-and-bound algorithm

Table 1.1: Test functions for the Lagrangian approach from [65]

function type	function form
1	$-\lvert x_1 + \sum_{j=2}^{n}(j - \frac{1}{j})x_j \rvert^{3/2}$
2	$-\lvert \sum_{j=1}^{n} \frac{1}{j} x_j \rvert \ln \left(1 + \lvert \sum_{j=1}^{n} \frac{1}{j} x_j \rvert \right)$
3	$-\exp(\lvert \sum_{j=1}^{n} x_j / n^2 \rvert)$

Furthermore, changing the bounds on y_i increases the Lagrangian multiplier in one child node and decreases it in the other, so the objective changes in each node.

1.2.3 Computational results

To evaluate the performance of the proposed approach, we test on several benchmark instances from the literature. The codes are written in $Matlab\ R2013b$ and the LP/MIP subproblems are solved using $CPLEX$ 12.51. All experiments are performed using an $Intel\ Core\ i7-3635QM$ 2.4 GHz CPU machine. When the B&B is required, we implement a depth-first search strategy. If there is more than one non-extreme variable, the one having value closer to the midpoint between $\underline{y_i}$ and $\overline{y_i}$ is selected for branching.

We first test on the concave functions from [65] depicted in Table 1.1. To ensure feasibility of the instances, we constructed the polytopes $X = \{x : Ax \leq b\}$ containing the feasible points as follows: the elements of A are uniformly distributed random numbers in the interval $[0, n]$. The right-hand side column vector is generated as $b = A\hat{x} + s$, where $\hat{x}$ is a vector of size n with elements selected randomly from a uniform distribution between 0 and 10, whereas s is vector of size m of uniformly distributed random numbers from the interval $[0, 1]$. We generated instances of four sizes ranging between (10×10) and (100×100). Two types of problems were considered: problems with continuous variables and problems with discrete variables. In both cases, we set $0 \leq x_j \leq 10$, $j = 1, \ldots, n$. For each combination of objective function type, instance size, and type of variables, 100 random instances were generated and solved. Table 1.2 summarizes the results. CPU times are in seconds.

The optimality gap is closed without need for branching in all instances except one.

Table 1.2: Results for the Lagrangian approach test problems from [65]

function	size	Continuous				Discrete			
		Gap (%)		CPU (sec.)		Gap (%)		CPU (sec.)	
type	$m \times n$	mean	max	mean	max	mean	max	mean	max
1	10×10	-	-	0.00	0.01	-	-	0.12	0.33
2	10×10	-	-	0.00	0.02	-	-	0.10	0.16
3	10×10	-	-	0.00	0.03	-	-	0.29	1.19
1	20×20	-	-	0.00	0.02	-	-	0.28	0.52
2	20×20	-	-	0.00	0.03	-	-	0.20	0.34
3	20×20	-	-	0.01	0.03	-	-	1.06	2.66
1	50×50	-	-	0.00	0.02	-	-	15.33	44.44
2	50×50	-	-	0.00	0.02	-	-	2.65	10.56
3	50×50	-	-	0.01	0.03	-	-	7.65	15.20
1	100×100	-	-	0.02	0.08	-	-	532.58	3115.94
2	100×100	-	-	0.01	0.03	-	0.15%	81.62	185.72
3	100×100	-	-	0.02	0.05	-	-	471.83	2183.13

In this 100-variables MIP instance with a concave function of type 2, a single branching step is required to improve both the Lagrangian lower bound and the upper bound and to close the gap. The computational time required to solve this instance to optimality is 160.05 seconds compared with an average of 80.62 seconds before branching for instances of the same size and objective function type. The computational time for LP instances is extremely small even for the largest instances tested. On the other hand, the computational time for MIP instances grows exponentially with the problem size, which is expected. For small instances (10×10 and 20×20) function type 3 seems to take longer to solve, followed by type 1, then type 2. For larger instances, function type 1 takes the longest to solve, followed by type 3, then type 2.

We then test on difficult instances from [53] that have several concave terms as well as a linear term in the objective function. Table 1.3 displays the results. The column 'Best solution' presents the best solution found in [53], whereas 'LB', 'UB', 'Gap', 'CPU', and 'Nodes' correspond to the Lagrangian bound, the objective function value corresponding to the feasible solution obtained, the optimality gap corresponding to LB and UB, the computational time in seconds, before and after branching, and the number of nodes tested, respectively.

In all the instances, branching is required to close the optimality gap. However, the

Table 1.3: Results for the Lagrangian approach test problems from [53]

Prob. #	Best solution	Before branching				After branching		
		LB	UB	Gap	CPU	UB	CPU	Nodes
2.5	-268.01	-269.08	-268.01	0.40%	0.03	-268.01	0.02	7
2.7-I	-394.75	-475.52	-375.79	20.97%	0.02	-394.75	0.51	1077
2.7-II	-884.75	-965.52	-865.79	10.33%	0.02	-884.75	0.51	1077
2.7-III	-8695.01	-10310.47	-8315.77	19.35%	0.02	-8695.01	0.45	1079
2.7-IV	-754.75	-835.52	-735.79	11.94%	0.02	-754.75	0.42	1077
2.8	15639	14439	16431	13.80%	0.02	15639	0.23	177
4.5	-11.96	-13.55	-13.40	1.07%	0.01	-13.40	0.03	3

optimal solution was reached in less than 1 second in all cases. For problem 4.5, an optimal solution, superior to the one reported in [53], was reached.

1.3 Applications to supply chain design problems

1.3.1 The production-transportation problem

The production-transportation problem was first addressed in [71]. In this problem, the aim is to find the optimal production and transportation quantities from M factories to N warehouses. Due to economies-of-scale, the production cost at factory i is concave and non-decreasing in the number of units produced, captured by function $f_i(.)$, $i = 1, \ldots, m$. The cost of shipping a unit from factory i to warehouse j is c_{ij}. Each warehouse has a demand of b_j and each factory has a production capacity of u_i such that $\sum_{j\in N} b_j \leq \sum_{i\in M} u_i$. We consider two variants of the problem: multiple sourcing and single sourcing. The multiple sourcing problem is formulated as follows:

$$
\begin{aligned}
[\mathcal{MSPTP}]\text{: } \min \quad & \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{i\in M} f_i(\sum_{j\in N} x_{ij}) & \\
\text{s.t.} \quad & \sum_{i\in M} x_{ij} = b_j, & j \in N, \\
& l_i \leq \sum_{j\in N} x_{ij} \leq u_i, & i \in M, \\
& x_{ij} \geq 0, & i \in M,\ j \in N,
\end{aligned}
$$

where x_{ij} is the quantity shipped from factory i to warehouse j.

As per the proposed Lagrangian approach, the variables $y_i,\ i \in M$ are introduced to denote the total production of each factory and the constraints $y_i = \sum_{j \in N} x_{ij}$ are relaxed using the optimal Lagrangian multipliers $\lambda_i = \frac{f_i(u_i) - f_i(l_i)}{u_i - l_i}$. Therefore, the production transportation problem is decomposed into a linear subproblem in x and M single variable concave subproblem in $y_i,\ i \in M$.

In the production-transportation problem with single sourcing, the binary variables x_{ij} take value 1 if warehouse j is served by factory i and 0 otherwise. So the problem is formulated as:

$$
\begin{aligned}
[\mathcal{SSPTP}]:\ \min\ & \sum_{(i,j) \in A} c_{ij} b_j x_{ij} + \sum_{i \in M} f_i(\sum_{j \in N} b_j x_{ij}) & \\
\text{s.t.}\ & \sum_{i \in M} x_{ij} = 1, & j \in N, \\
& l_i \leq \sum_{j \in N} b_j x_{ij} \leq u_i, & i \in M, \\
& x_{ij} \in \{0,1\}, & i \in M,\ j \in N.
\end{aligned}
$$

For the single sourcing variant of the problem the new variables denoting the total production of each factory are defined as $y_i = \sum_{j \in N} b_j x_{ij},\ i \in M$. The Lagrangian decomposition is performed exactly as was done for the multiple sourcing problem. The resulting $\mathcal{SP}1$ is an integer programming problem with $(M \times N)$ binary variables.

Following [71], the concave production functions are defined as $f_i(y_i) = \beta\sqrt{y_i}$, for β uniformly random in [10,20]. The test problems were generated in the same manner described in the previous reference as follows: c_{ij} are random numbers between 0 and 1, u_j are all set to 200, and b_j are set to the round-off value of $\alpha(\sum_{i \in M} u_i)/M$ for $\alpha = 0.6,\ 0.75$ and 0.9. The sizes of (M, N) range from (5,25) to (10,50).

Table 1.4 depicts the results for the production-transportation instances tested with both multiple and single sourcing. The columns show the optimality gap at the root node (Gap), the computational time in seconds for the root node (CPU0), the total computational time in seconds (CPU) and the number of nodes tested (Nodes) until the gap is closed. For each instance size, the upper row shows the average values for 10 randomly generated instances, whereas the lower row shows the maximum values.

For the multiple sourcing problem, the results obtained show that the proposed Lagrangian approach is capable of reaching the optimal solutions in an average of 2.86 seconds for the tested instances, with a maximum of less than 36 seconds for the most difficult

Table 1.4: Results for the production-transportation problem

$M \times N$	α	Multiple sourcing				Single sourcing			
		Gap	CPU0	CPU	Nodes	Gap	CPU0	CPU	Nodes
5×25	0.60	9.62	0.01	0.35	497	9.54	0.02	1.79	271
		(11.25)	(0.02)	(0.66)	(833)	(11.49)	(0.06)	(3.73)	(585)
	0.75	4.73	0.01	0.09	117	4.97	0.04	1.44	230
		(5.57)	(0.03)	(0.17)	(233)	(5.64)	(0.25)	(2.59)	(427)
	0.90	1.66	0.00	0.03	34	1.68	0.01	0.35	63
		(2.06)	(0.03)	(0.08)	(105)	(1.90)	(0.03)	(0.45)	(63)
5×50	0.60	9.21	0.01	0.86	870	9.17	0.01	6.04	688
		(11.01)	(0.03)	(1.89)	(1,941)	(11.03)	(0.03)	(13.38)	(1,521)
	0.75	5.06	0.00	0.29	278	5.11	0.02	4.17	453
		(5.99)	(0.00)	(0.55)	(578)	(5.63)	(0.03)	(6.65)	(755)
	0.90	1.84	0.00	0.08	72	1.82	0.01	1.04	120
		(2.09)	(0.03)	(0.22)	(191)	(2.10)	(0.05)	(2.39)	(291)
10×25	0.60	9.43	0.00	2.43	2,935	8.86	0.01	22.05	2,321
		(12.96)	(0.02)	(5.41)	(6,513)	(12.66)	(0.03)	(40.78)	(4,423)
	0.75	4.78	0.01	0.61	707	5.28	0.02	27.92	3,042
		(6.88)	(0.03)	(3.06)	(3,625)	(7.04)	(0.05)	(53.88)	(5,815)
	0.90	1.67	0.00	0.07	73	Infeasible			
		(2.56)	(0.01)	(0.23)	(215)				
10×50	0.60	10.53	0.00	20.43	15,645	11.64	0.02	573.93	37,788
		(12.65)	(0.03)	(34.48)	(27,089)	(12.88)	(0.03)	(1,710.85)	(112,077)
	0.75	6.14	0.00	7.84	5,604	6.37	0.02	455.51	30,389
		(6.58)	(0.03)	(35.47)	(25,029)	(7.06)	(0.05)	(1,495.77)	(98,739)
	0.90	2.29	0.00	1.26	908	2.17	0.01	23.37	2,047
		(2.70)	(0.02)	(7.27)	(5,227)	(2.41)	(0.03)	(24.59)	(2,047)

instance. The computational time is largely influenced by the parameter α, the ratio between total demand and total capacity. When α is closer to one, the branch-and-bound algorithm tests fewer nodes and terminates faster. For example, in the 5×25 instances, an average of 34 nodes are needed for $\alpha = 0.9$, compared to 117 and 497 nodes when α is 0.75 and 0.6, respectively. For the single sourcing instances where all the variables are binary, the average computational time per node is large compared to the corresponding multiple sourcing instances, leading to larger computational times. However, the average computational time for the most difficult instance of size (10×50) with $\alpha = 0.6$ is less than 10 minutes, a very reasonable time for this difficult design problem. On the other hand, the computational time when the ratio of demand to capacity is high (i.e., $\alpha = 0.9$) is very attractive, not exceeding 25 seconds. The infeasibility of the (10×25) single-sourcing instances with $\alpha = 0.9$ is due to the fact that, with these parameters, the demand of each warehouse is 72 units, thus it is not possible to assign more than 2 warehouses to a factory with a capacity of 200 units. Therefore, there is no feasible partitioning of the 25 warehouses that respects the capacity constraints of the 10 factories.

The computational results underline the excellent performance of the proposed approach for solving the production-transportation problem, especially when the subproblem is linear (as in the case with multiple sourcing) and when the total demand is close to the total supply of the system (as in the case when $\alpha = 0.9$. Even with the more difficult single sourcing version of the problem, the approach performs quite well even for the largest tested instance, except for those with a low demand-to-supply ratio.

1.3.2 The plant location and technology acquisition problem

The plant location and technology acquisition problem introduced in [35] is a single-echelon, multi-product, production-distribution problem. We adhere here to the notations used in the original paper. Let $I = \{i : i = 1, \ldots, m\}$, $J = \{j : j = 1, \ldots, n\}$, $P = \{p : p = 1, \ldots, r\}$ be index sets of potential plant locations, customer zones, and products, respectively. Each product p can be produced using a number of disjoint technologies indexed by $h_p \in H_p$, where the site-specific technology cost functions $f_{h_p i}(.)$ are affine or concave. Since each facility selects a technology for each product that minimizes its production cost based on capacity to be built, the production cost can be represented by the single piecewise concave function $f_{ip}(.) = \min_{h_p \in H_p} \{f_{h_p i}(.)\}$, $\forall (i,p)$. Let F_i be the fixed cost of opening plant i, C_{ijp} be the total cost of serving the entire demand of customer j for product p from plant i, and D_{jp} be the demand of customer j for product p. The following variables are used:

Y_i : binary variable for opening plaint i;

x_{ijp} : fraction of demand from customer j for product p supplied from plant i.

With this, the problem is formulated as

$$\begin{aligned}
[\mathcal{P}]: \quad \min \quad & \sum_{i\in I} F_i Y_i + \sum_{i\in I}\sum_{p\in P} f_{ip}\left(\sum_{j\in J} D_{jp}x_{ijp}\right) + \sum_{i\in I}\sum_{j\in J}\sum_{p\in P} C_{ijp}x_{ijp} \\
\text{s.t.} \quad & \sum_{i\in I} x_{ijp} = 1, \quad \forall (j,p), && (1.11) \\
& 0 \le x_{ijp} \le Y_i, \quad \forall (i,j,p), && (1.12) \\
& Y_i \in \{0,1\}, \quad \forall i. && (1.13)
\end{aligned}$$

The objective function represents the sum of the fixed costs, variable capacity acquisition and operating costs, and shipping costs. The first constraint ensures that the entire demand is served, whereas the second constraint stipulates that facilities can serve demand only if they are opened. Dasci and Verter [35] propose a progressive piecewise linear underestimation technique to obtain an iteratively improving lower bound. They show that subproblems can be formulated as two-echelon uncapacitated facility location problems. Branch-and-bound is used to solve the subproblems where a dual-ascent procedure is applied on the LP-relaxation to obtain a lower bound, and a heuristic is used to obtain a feasible solution and an upper bound.

Obviously, the difficulty of the problem stems primarily from the piecewise concave terms in the objective function. If these terms are eliminated, the model is reduced to a multiple-product uncapacitated facility location problem (UFLP). Note that single assignment is guaranteed because of the uncapacitated nature of the problem, *i.e.*, $x_{ijp} \in \{0,1\}$. Despite being **NP**-complete, UFLPs are well-studied and several algorithms have been developed to solve them efficiently [105]. To apply the proposed Lagrangian approach, we first introduce the variables

$$Q_{ip} = \sum_{j\in J} D_{jp}x_{ijp}, \ \forall (i,p).$$

and relax this equality with multipliers μ_{ip} to get the multiple-product UFLP subproblem

$$\begin{aligned}
[\mathcal{SP}1]: \quad \min \quad & \sum_{i\in I} F_i Y_i + \sum_{i\in I}\sum_{j\in J}\sum_{p\in P} (\mu_{ip}D_{jp} + C_{ijp})x_{ijp} \\
\text{s.t.} \quad & (1.11) - (1.13),
\end{aligned}$$

and mr single-variable concave minimization subproblems

$$[\mathcal{SP}2_{ip}]: \quad \min \quad f_{ip}\left(Q_{ip}\right) - \mu_{ip} Q_{ip}$$
$$\underline{Q}_{ip} \leq Q_{ip} \leq \overline{Q}_{ip}.$$

According to Proposition 1, the optimal multipliers are given by:

$$\mu_{ip}^{*} = \frac{f_{ip}(\overline{Q}_{ip}) - f_{ip}(\underline{Q}_{ip})}{\overline{Q}_{ip} - \underline{Q}_{ip}}. \tag{1.14}$$

Thus, the best Lagrangian bound is

$$\sum_{i \in I} F_i Y_i^{*} + \sum_{i \in I} \sum_{j \in J} \sum_{p \in P} (\mu_{ip}^{*} D_{jp} + C_{ijp}) x_{ijp}^{*} + \sum_{i \in I} \sum_{p \in P} \frac{\overline{Q}_{ip} f_{ip}(\underline{Q}_{ip}) - \underline{Q}_{ip} f_{ip}(\overline{Q}_{ip})}{\overline{Q}_{ip} - \underline{Q}_{ip}},$$

where Y_i^{*} and x_{ijp}^{*} are the solution of $\mathcal{SP}1$ when $\mu_{ip} = \mu_{ip}^{*}$. The Lagrangian approach is embedded in a branch-and-bound algorithm.

We test on a classical data set from the literature to measure its performance in terms of computational time and solution quality. The solution method was coded in *Matlab* and solved using *CPLEX 12.3* on a machine with a *3.4 GHz Intel Core i-7 4770* processor. Following [35], we base our instances on a capacitated warehouse location problem from the OR-Library [17] called *cap*71. The base case parameters are set identical to those of problem set 2b in [35], whereas these parameters are changed in subsequent instances to test the sensitivity of the approach to variability and nonlinearity.

All tested instances are of size $(m, n) = (16, 50)$. The number of products is changed between 1 and 5. Five technologies are assumed for each product. The technology costs, as a function of the acquired capacity, take the form $f_{h_p i}(.) = a + b(.)^c$. The values of these parameters in the base case are depicted in Table 1.5. Each plant has a fixed cost of \$75,000. For the base case, demand and shipping cost parameters are uniformly distributed with endpoints 20% below and above the *cap*71 parameters. Furthermore, two more scenarios are tested: 1) Higher variability: endpoints for the uniformly distributed demand and shipping costs are 40% below and above the original values; and 2) Higher nonlinearity: the exponents c are multiplied by 0.9 for all technologies to represent stronger

Table 1.5: Cost parameters of the plant location and technology acquisition problem

Function	$f_{1_pi}(.)$	$f_{2_pi}(.)$	$f_{3_pi}(.)$	$f_{4_pi}(.)$	$f_{5_pi}(.)$
a	0	0	0	0	[4000,5000]
b	[45,50]	[2.5,3.5]	[22,28]	[12,18]	[1.5,2.5]
c	[0.65,0.70]	1	[0.72,0.77]	[0.79,0.84]	1

economies of scale, whereas fixed and shipment costs are set to 50% of their original values to maintain their relative percentage of the total cost. For each set of parameters, 10 random instances are solved. A depth-first search strategy is implemented for the branch-and-bound algorithm.

To benchmark the proposed approach, we compare its results with those obtained from a piecewise linearization, a classical generic approach to solve concave minimization problems. We linearize the functions $f_{ip}\left(Q_{ip}\right)$ using piecewise segments linking the points $\left\{(0,0), \left(Q_{ip}^1, f_{ip}(Q_{ip}^1)\right), \left(Q_{ip}^2, f_j(Q_{ip}^2)\right), \ldots, \left(Q_{ip}^R, f_{ip}(Q_{ip}^R)\right)\right\}$ and special ordered sets of type 2 (SOS2) [16]. We can write Q_{ip} and the approximated functions $F_{ip}(Q_{ip})$ as:

$$\begin{aligned} Q_{ip} &= \sum_{r=0}^{R} w_{ip}^r Q_{ip}^r, && \forall (i,p) \\ F_{ip}(Q_{ip}) &= \sum_{r=0}^{R} w_{ip}^r f_{ip}(Q_{ip}^r), && \forall (i,p) \\ \sum_{r=0}^{R} w_{ip}^r &= 1, && \forall (i,p) \end{aligned}$$

and $\left\{w_{ip}^0, w_{ip}^1, \ldots, w_{ip}^R\right\}$ are special ordered sets of type 2. The number of equally-spaced breakpoints, $R+1$, is increased incrementally until the sought relative optimality gap is reached in all instances.

Table 1.6 shows the computational results. The columns entitled 'gap-node0' and 'cpu-node0' show the relative optimality gaps and the computational times in seconds at the root node, *i.e.*, before branching. The computational times required to reach an ϵ-optimal solution ($\epsilon = 0.02$) are shown in the columns titled 'cpu-B&B'. The last column shows the computational times of the piecewise linearization approach. For each metric, the mean, minimum and maximum values are reported.

As can be seen, without branching, the proposed approach is able to reach solutions

Table 1.6: Results for the plant location and technology acquisition problem

r	gap-node0 (%)			cpu-node0 (sec.)			cpu-B&B (sec.)			cpu-PWL (sec.)		
	mean	min	max	mean	min	max	mean	min	max	mean	min	max
Base case												
1	1.58	0.87	2.03	0.01	0.01	0.01	0.01	0.01	0.03	0.34	0.03	0.92
2	2.29	1.90	3.20	0.05	0.01	0.26	0.26	0.01	1.57	1.50	0.33	2.67
3	3.33	2.71	3.87	0.07	0.02	0.24	2.72	0.61	5.70	15.11	4.79	23.81
4	3.85	3.45	4.51	0.08	0.03	0.29	32.87	10.63	67.05	100.78	32.62	300.43
5	4.34	3.94	4.92	0.10	0.04	0.21	500.85	10.92	1,061.90	2,357.06	401.98	7,587.65
Higher variability												
1	2.09	1.58	2.32	0.01	0.01	0.01	0.05	0.01	0.26	0.46	0.12	1.67
2	3.18	2.68	4.45	0.04	0.01	0.17	0.41	0.09	0.80	1.91	0.81	2.65
3	3.89	3.35	4.48	0.02	0.02	0.03	6.30	1.29	16.84	21.71	6.82	32.04
4	4.31	4.00	4.50	0.13	0.03	0.31	379.61	31.20	882.91	587.14	72.42	1,468.87
5	4.69	4.34	5.13	0.17	0.14	0.18	2,366.40	761.69	3,288.30	13,850.97	1,373.90	39,133.30
Higher nonlinearity												
1	1.93	0.64	2.62	0.01	0.01	0.01	0.02	0.01	0.03	0.52	0.03	1.92
2	2.65	1.96	3.51	0.02	0.02	0.02	0.13	0.02	0.39	1.09	0.27	1.76
3	3.71	2.96	4.44	0.05	0.02	0.22	3.18	0.29	9.24	10.36	5.07	26.19
4	5.74	4.12	7.74	0.50	0.03	1.12	109.70	40.48	296.39	315.37	65.08	526.36
5	4.72	4.30	5.00	0.13	0.05	0.23	1,227.48	285.13	2,980.92	6,010.28	798.48	17,137.49

within 6% of the optimal in less than 0.5 second on average. As anticipated, initial optimality gap and computational time increase as the problem size increases. Compared to the base case, the initial optimality gap slightly increases for higher variability or nonlinearity. However, there is a significant increase in the computational time, especially when the variability of the parameters increases, implying that the approach is quite sensitive to the problem parameters. The substantial growth in the computational time for the highly nonlinear instances compared to the base case ones is explained by the need for more branching nodes to reach the threshold gap.

Finally, the Lagrangian heuristic combined with B&B performs significantly better than the classical piecewise linear approximation in terms of computational time, especially for difficult instances. For example, the average computational time ratio between the two approaches for the 4-product, highly nonlinear instances is 1.55, and increases to 5.85 for similar 5-product instances. Nonetheless, the proposed heuristic is slower than other algorithms developed specifically for the problem under consideration, including the algorithm proposed by Dasci and Verter [35]. But unlike the latter, our proposed algorithm is applicable to a broader class of problems.

1.3.3 The location-inventory problem

For the location-inventory problem, we use the same notation as in [37], where a set $I = \{i \mid i = 1, 2, \ldots, m\}$ of retailers are facing independent normally-distributed daily demand for a single commodity with mean μ_i and variance σ_i^2. There is also a set $J = \{j \mid j = 1, 2, \ldots, n\}$ of potential distribution center (DC) locations, each having fixed set-up cost f_j. Each retailer has to be assigned to a DC. The unit shipment cost between DC j and retailer i is d_{ij}. The working inventory and safety stock are managed at the DC level, so that each retailer retains a minimal level of inventory and DC's hold enough safety stock for a type-1 service level α. There is a fixed order cost F_j, a fixed and a per-unit shipment cost from the single supplier to each DC, g_j and a_j, respectively. Once an order is placed, there is a lead time L before it arrives to the DC's, and each unit has an annual holding cost h. The goal is to decide on the number and location of DC's to open, retailer allocation to them, the level of safety stock to maintain and the frequency of reordering at the DC's, in order to minimize the total location, shipment, working inventory, and safety stock costs. Note that potential DC locations could be selected at the retailers, *i.e.*, a subset of the retailers are upgraded to act as storage/distribution centers.

Daskin et al. [37] formulated the problems as:

$$
\begin{aligned}
\min\ & \sum_{j\in J}\left\{ f_j X_j + \sum_{i\in I}\hat{d}_{ij}Y_{ij} + K_j\sqrt{\sum_{i\in I}\mu_i Y_{ij}} + \Theta\sqrt{\sum_{i\in I}\hat{\sigma}_i^2 Y_{ij}} \right\} && && (1.15)\\
\text{s.t.}\ & \sum_{j\in J} Y_{ij} = 1, && \forall i \in I, && (1.16)\\
& Y_{ij} \le X_j, && \forall i \in I, \forall j \in J, && (1.17)\\
& X_j \in \{0,1\}, && \forall j \in J, && (1.18)\\
& Y_{ij} \in \{0,1\}, && \forall i \in I, \forall j \in J. && (1.19)
\end{aligned}
$$

where $X_j = 1$ if a DC is opened at candidate location j, and $Y_{ij} = 1$ if retailer i is served by the DC located at j. The problem parameter are calculated using the following formulas:

$$
\begin{aligned}
\hat{d}_{ij} &= \beta\chi\mu_i(d_{ij} + a_j),\\
K_j &= \sqrt{2\theta h\chi(F_j + \beta g_j)},\\
\Theta &= \theta h z_\alpha,\\
\hat{\sigma}_i^2 &= L\sigma_i^2.
\end{aligned}
$$

where χ denotes the number of days in a year, whereas β and θ are weights on the transportation and inventory cost components.

The four terms in the objective function correspond to the location, shipment, working inventory, and safety stock costs, respectively. For a detailed derivation refer to [37]. The model has two sets of constraints: (1.16) assigns each retailer to exactly one DC, while (1.17) stipulates that a DC must be opened before any retailers can be assigned to it.

The nonlinearity of the last two terms in (1.15) increases the complexity of the problem. When the holding cost (h) is negligible, or when the demand is deterministic and the fixed order and shipment costs (F_j and g_j) are very small compared to the transportation costs, the problem can be approximated by the traditional UFL problem. However, when the inventory costs are large, this approximation may lead to a solution far from the optimal.

The leading solution approaches to solve the problem rely on Lagrangian relaxation, where constraints (1.16) are relaxed and constraints (1.17) appear in the nonlinear subproblem. Daskin et al. [37] solved a special case of the problem when the variance-to-mean ratio is the same for all retailers using a sorting algorithm. Shen et al. reformulated the problem as a set covering model and devised a column generation approach to solve the same special case. Later, the general case of the pricing subproblem (*i.e.*, where two concave terms appear in the subproblem) was solved in [90] and [89] through a set partitioning algorithm that utilizes VC-dimensions to limit the search space. Recently, Atamturk et al. [14] reformulated this problem as a conic quadratic mixed-integer problem. They Strengthened the formulation using valid inequalities to improve the computational results.

We use the Lagrangian approach proposed earlier to solve the problem. First, we define the new variables P_j and Q_j , such that $P_j = \sum_{i\in I} \mu_i Y_{ij}$, and $Q_j = \sum_{i\in I} \hat{\sigma}_i^2 Y_{ij}$:

$$
\begin{aligned}
[\mathcal{P}]: \qquad & \min \sum_{j\in J} \left\{ f_j X_j + \sum_{i\in I} \hat{d}_{ij} Y_{ij} + K_j \sqrt{P_j} + \Theta \sqrt{Q_j} \right\} && && (1.20)\\
& \text{s.t. } (1.16) - (1.19) \\
& \qquad P_j = \sum_{i\in I} \mu_i Y_{ij}, && \forall j \in J, && (1.21)\\
& \qquad Q_j = \sum_{i\in I} \hat{\sigma}_i^2 Y_{ij}, && \forall j \in J, && (1.22)\\
& \qquad 0 \le P_j \le \sum_{i\in I} \mu_i, && \forall j \in J, \\
& \qquad 0 \le Q_j \le \sum_{i\in I} \hat{\sigma}_i^2, && \forall j \in J, .
\end{aligned}
$$

And then relax (1.21) and (1.22) with Lagrangian multipliers δ_j and λ_j, respectively. The resulting subproblems are:

$$
\begin{aligned}
[\mathcal{SP}1]: \qquad & \min \sum_{j\in J} \left\{ f_j X_j + \sum_{i\in I} (\hat{d}_{ij} - \delta_j \mu_i - \lambda_j \hat{\sigma}_i^2) Y_{ij} \right\} \\
& \text{s.t. } \sum_{j\in J} Y_{ij} = 1 && \forall i \in I, \\
& \qquad Y_{ij} \le X_j, && \forall i \in I, \forall j \in J, \\
& \qquad X_j \in \{0,1\}, && \forall j \in J, \\
& \qquad Y_{ij} \in \{0,1\}, && \forall i \in I, \forall j \in J,
\end{aligned}
$$

which is an UFL problem. The first concave term appears in the second subproblem:

$$[\mathcal{SP}2]: \qquad \min \sum_{j\in J} \left\{ K_j\sqrt{P_j} + \delta_j P_j \right\}$$
$$\text{s.t.} \quad 0 \le P_j \le \sum_{i\in I} \mu_i, \qquad \forall j \in J,$$

which is decomposed to n subproblems, each with a single variable P_j:

$$[\mathcal{SP}2_j]: \qquad \min \ K_j\sqrt{P_j} + \delta_j P_j$$
$$\text{s.t.} \quad 0 \le P_j \le \sum_{i\in I} \mu_i.$$

As the objective function is concave, one of the extreme points is optimal, *i.e.*, $P_j = 0$ (none of the retailers are assigned to the DC) or $P_j = \sum\limits_{i\in I} \mu_i$ (all retailers are assigned to the DC), depending on the values of K_j and δ_j.

Likewise, the second concave term appears in the third subproblem:

$$[\mathcal{SP}3]: \qquad \min \sum_{j\in J} \left\{ \Theta\sqrt{Q_j} + \lambda_j Q_j \right\}$$
$$\text{s.t.} \quad 0 \le Q_j \le \sum_{i\in I} \hat{\sigma}_i^2, \qquad \forall j \in J,$$

which is also decomposed to n subproblems, each with a single variable Q_j:

$$[\mathcal{SP}3_j]: \qquad \min \ \Theta\sqrt{Q_j} + \lambda_j Q_j$$
$$\text{s.t.} \quad 0 \le Q_j \le \sum_{i\in I} \hat{\sigma}_i^2.$$

Again, one of the extreme points $Q_j = 0$ or $Q_j = \sum\limits_{i\in I} \hat{\sigma}_i^2$ is optimal. Thus, the best Lagrangian bound resulting from this relaxation is:

$$LB^* = \max_{\delta,\lambda} \left\{ \mathbf{v}[\mathcal{SP}1] + \sum_{j\in J} \Big(\mathbf{v}[\mathcal{SP}2_j] + \mathbf{v}[\mathcal{SP}3_j] \Big) \right\}$$

where $v[\mathcal{SP}1]$ is the optimal objective value of $\mathcal{SP}1$, $v[\mathcal{SP}2_j] = \min\big(0, K_j\sqrt{\sum_{i\in I}\mu_i} + \delta_j\sum_{i\in I}\mu_i\big)$ and $v[\mathcal{SP}3_j] = \min\big(0, \Theta\sqrt{\sum_{i\in I}\hat{\sigma}_i^2} + \lambda_j\sum_{i\in I}\hat{\sigma}_i^2\big)$. According to Proposition 1, the best Lagrangian multipliers can be calculated using the closed-form expressions:

$$\delta_j^* = -\frac{K_j}{\sqrt{\sum_{i\in I}\mu_i}} \tag{1.23}$$

$$\lambda_j^* = -\frac{\Theta}{\sqrt{\sum_{i\in I}\hat{\sigma}_i^2}} \tag{1.24}$$

Since the solution of $\mathcal{SP}1^*$ ($\overline{X}_j$ and $\overline{Y}_{ij}$) is feasible to the original problem, we can assess its quality by comparing its corresponding objective:

$$\overline{Z} = \sum_{j\in J}\left\{ f_j\overline{X}_j + \sum_{i\in I}\hat{d}_{ij}\overline{Y}_{ij} + K_j\sqrt{\sum_{i\in I}\mu_i\overline{Y}_{ij}} + \Theta\sqrt{\sum_{i\in I}\hat{\sigma}_i^2\overline{Y}_{ij}} \right\}$$

to the Lagrangian lower bound:

$$LR = \sum_{j\in J}\left\{ f_j\overline{X}_j + \sum_{i\in I}\Big(\hat{d}_{ij} + \frac{K_j\mu_i}{\sqrt{\sum_{i\in I}\mu_i}} + \frac{\Theta\hat{\sigma}_i^2}{\sqrt{\sum_{i\in I}\hat{\sigma}_i^2}}\Big)\overline{Y}_{ij} \right\}.$$

The difference ($\overline{Z} - LR$) provides an upper bound on the optimality gap:

$$0 \le \overline{Z} - LR \le \sum_{j\in J}\left\{ K_j\Big(\sqrt{\sum_{i\in I}\mu_i\overline{Y}_{ij}} - \frac{\sum_{i\in I}\mu_i\overline{Y}_{ij}}{\sqrt{\sum_{i\in I}\mu_i}}\Big) + \Theta\Big(\sqrt{\sum_{i\in I}\hat{\sigma}_i^2\overline{Y}_{ij}} - \frac{\sum_{i\in I}\hat{\sigma}_i^2\overline{Y}_{ij}}{\sqrt{\sum_{i\in I}\hat{\sigma}_i^2}}\Big) \right\}.$$

The solution $\overline{X}_j$, $\overline{Y}_{ij}$ is optimal if the right-hand-side is zero, which is the case when all retailers are assigned to a single DC (there exists $j \in J$ such that $Y_{ij} = 1\ \forall i \in I$), a situation that happens when the transportation costs are small compared to the fixed setup costs and the saving from risk pooling. If more than one DC is opened, the gap may theoretically persist. However, experiments have shown that in many cases with multiple DCs opened, the gap is entirely closed without branching, especially when the inventory costs are relatively small (*i.e.*, K_j and Θ are small).

In order to narrow the gap, we use a simple, single-layer branch-and-bound algorithm. Branching is done based on fixing the number of open DCs to $p = 1, \ldots, n$. We generate n branches, where the branching constraint:

$$\sum_{j \in J} X_j = p$$

is added to $\mathcal{SP}1$. We also know that in the optimal solution, at least one retailer must be assigned to each opened DC. So if p DCs are to be opened, any single DC can not serve more than $(m - p + 1)$ retailers. This is enforced by adding the following constraints:

$$0 \leq P_j \leq \sum_{k=1}^{m-p+1} \mu_{(k)}, \quad \forall j \in J,$$

$$0 \leq Q_j \leq \sum_{k=1}^{m-p+1} \hat{\sigma}^2_{(k)}, \quad \forall j \in J,$$

to $[\mathcal{SP}2_j]$ and $[\mathcal{SP}3_j]$, where $\mu_{(k)}$ and $\sigma^2_{(k)}$ are the k^{th} largest demand mean and k^{th} largest demand variance, respectively.

The proposed approach was tested on several datasets to measure its performance in terms of computational and solution quality. The solution method was coded in *Matlab* and solved using *CPLEX 12.3* on a *2.26 GHz Intel Xeon E5607* processor machine. First, the approach was tested on two classical datasets from the literature: an 88 node dataset representing the major U.S. metropolitan centers from [36], and a 150 node dataset representing the largest 150 cities in the Continental U.S from [37]. We used the same parameters and weights (θ and β) used in [37] for comparability. Tables 1.7 and 1.8 show the results for the two datasets. The columns display the weights β and θ, the optimal objective function value, the computational time reported in [37] (CPU$_D$) and [14] (CPU$_A$), the Lagrangian lower bound (LB), the upper bound (UB) based on the best feasible solution found, the gap between the feasible solution and the lower bound (Gap1), the gap relative to the optimal solution (Gap2), and the computational time in seconds (CPU), both before and after branching.

According to the table, the quality of the lower bound depends largely on the weights β and θ. For the 88 node dataset, the relative optimality gap 'Gap1' ranges between 3.76%

Table 1.7: Results for the location-inventory problem (88 node dataset)

Weights		Benchmark Results			Before Branching					After Branching				
β	θ	optimal	CPU$_D$	CPU$_A$	LB	UB	Gap1	Gap2	CPU	LB	UB	Gap1	Gap2	CPU
0.001	0.1	13,229.6	2.08	1	12,591.0	13,229.6	5.07%	0.00%	0.26	12,591.7	13,229.6	5.06%	0.00%	35.30
0.002	0.1	19,975.3	3.40	1	19,251.6	19,976.2	3.76%	0.00%	0.24	19,252.6	19,976.2	3.76%	0.00%	31.71
0.003	0.1	25,306.7	2.69	2	24,355.7	25,306.7	3.90%	0.00%	0.32	24,357.5	25,306.7	3.86%	0.00%	41.09
0.004	0.1	28,752.6	5.87	2	27,614.6	28,786.4	4.24%	0.12%	0.24	27,618.7	28,752.6	4.11%	0.00%	32.32
0.005	0.1	31,391.7	3.57	2	30,162.9	31,393.3	4.08%	0.01%	0.22	30,168.0	31,391.7	4.06%	0.00%	35.61
0.001	0.1	13,229.6	2.09	1	12,591.0	13,229.6	5.07%	0.00%	0.26	12,591.7	13,229.6	5.06%	0.00%	35.30
0.002	0.2	20,491.2	2.69	1	19,417.3	20,493.5	5.54%	0.02%	0.25	19,418.8	20,492.7	5.53%	0.01%	38.94
0.005	0.5	33,794.9	2.74	3	30,700.0	33,864.2	10.31%	0.21%	0.24	30,713.1	33,794.9	10.03%	0.00%	35.84
0.005	0.1	31,391.7	3.62	2	30,162.9	31,393.3	4.08%	0.01%	0.22	30,168.0	31,391.7	4.06%	0.00%	35.61
0.005	0.5	33,794.9	2.75	3	30,700.0	33,864.2	10.31%	0.21%	0.24	30,713.1	33,794.9	10.03%	0.00%	35.84
0.005	1.0	35,876.7	1.27	2	31,186.6	36,098.2	15.75%	0.62%	0.24	31,206.0	35,876.7	14.97%	0.00%	41.17
0.005	5.0	47,348.4	1.93	5	34,019.2	49,134.3	44.43%	3.77%	0.23	34,081.9	47,349.6	38.93%	0.00%	37.15
0.005	10.0	57,960.5	1.98	10	36,973.7	62,726.6	69.65%	8.22%	0.24	37,080.5	58,066.6	56.60%	0.18%	41.95
0.005	20.0	74,761.0	3.96	30	42,367.7	87,541.4	106.62%	17.09%	0.24	42,554.9	74,782.2	75.73%	0.03%	40.90
Min							3.76%	0.00%	0.22			3.76%	0.00%	31.71
Avg							24.28%	2.73%	0.25			20.24%	0.02%	37.45
Max							106.62%	17.09%	0.32			75.73%	0.18%	41.95

Table 1.8: Results for the location-inventory problem (150 node dataset)

Weights		Benchmark Results			Before Branching					After Branching				
β	θ	optimal	CPU$_A$	CPU$_D$	LB	UB	Gap1	Gap2	CPU	LB	UB	Gap1	Gap2	CPU
0.0004	0.01	3977.3	13.62	12	3664.0	3980.9	8.65%	0.09%	0.77	3665.9	3977.3	8.47%	0.00%	199.54
0.0006	0.01	4867.8	13.90	8	4499.0	4867.8	8.19%	0.00%	0.81	4501.6	4867.8	8.12%	0.00%	197.65
0.0008	0.01	5580.7	15.32	11	5150.7	5587.3	8.48%	0.12%	0.89	5154.1	5580.7	8.26%	0.00%	200.08
0.001	0.01	6163.0	15.87	10	5704.5	6165.4	8.08%	0.04%	0.77	5708.2	6163.0	7.98%	0.01%	201.09
0.0005	0.01	4459.3	13.68	10	4111.4	4470.7	8.74%	0.26%	0.78	4113.8	4459.3	8.39%	0.00%	201.08
0.001	0.02	6410.1	15.55	11	5753.9	6417.1	11.53%	0.11%	0.79	5759.3	6410.1	11.31%	0.01%	199.05
0.002	0.04	8988.6	4.78	23	7835.7	9000.3	14.86%	0.13%	0.80	7847.8	8988.6	14.54%	0.00%	200.02
0.001	0.01	6163.0	15.81	10	5704.5	6165.4	8.08%	0.04%	0.77	5708.2	6163.0	7.98%	0.01%	201.09
0.001	0.1	7508.5	10.94	24	5980.4	7570.3	26.58%	0.82%	0.82	5993.2	7509.3	25.30%	0.01%	199.99
0.001	0.5	10175.7	13.74	56	6591.5	10681.1	62.05%	4.97%	0.76	6624.4	10179.0	53.66%	0.03%	199.36
0.001	1.0	12380.6	13.46	185	7144.0	13494.3	88.89%	9.00%	0.76	7195.1	12380.6	72.09%	0.00%	200.39
Min							8.08%	0.00%	0.76			7.98%	0.00%	197.65
Avg							24.61%	1.55%	0.80			21.81%	0.01%	199.83
Max							88.89%	9.00%	0.89			72.09%	0.03%	201.09

and 106.62% with an average of 24.28% before branching, and between 3.76% and 75.73% with an average of 20.24% after branching. For the 150 node dataset, 'Gap1' ranges between 8.08% and 88.89% with an average of 24.61% before branching and between 7.98% and 72.09% with an average of 21.81% after branching. Notably, the optimality gap is quite narrow except for the cases where the cost of inventory is large relative to the cost of transportation (*i.e.*, very high θ/β ratios). This is expected since the terms relaxed in the objective function are those associated with the inventory costs. Second, although the optimality gap is not closed, the upper bound 'UB' after branching is always very close, if not identical, to the optimal solution. The optimal solution was found in 11 out of 14 cases for the 88 node dataset and in 9 out of 11 cases for the 150 node dataset. The rest of the solutions are less than 0.2% from the optimal. Third, although the partial B&B algorithm led to a marginal improvement in the lower bound, the improvement in the upper bound was enough to close the optimality gap in all cases. Therefore, the global optima in all cases is reached, though without proof of optimality. For the 88 node dataset, 'Gap2' ranges between 0 and 17.09% with an average of 2.73% before branching, and between 0 and 0.18% with an average of 0.02% after branching. For the 150 node dataset, 'Gap2' ranges between 0 and 9% with an average of 1.55% before branching, and between 0 and 0.03% with an average of 0.01% after branching. Last, the computational time before branching is very small, averaging 0.25 seconds for the 88 node dataset and 0.8 seconds for the 150 nodes dataset. Even with the partial B&B step, the average execution time is 37.45 and 199.83 seconds for the 88 and 150 node datasets, respectively.

Next, the algorithm was tested on larger instances. New datasets of 500 and 1000 nodes representing the largest cities and towns in the U.S. according to the 2010 census were generated [28]. As in [37], the demand mean and variance were obtained by dividing the population by 1000 and rounding the result to the nearest integer. We set the holding cost to 1, $z_\alpha = 1.96$, $a_i = 5$, $g_i = 10$, and $F_i = 10$ for all $i \in I$. The fixed facility location costs were set uniformly to 100, whereas the scalars L and χ were both set to 1. The CPU time limit was set to 10,000 seconds. The cost factors were selected so that θ ranges between 0.01 and 1 whereas the ratio θ/β is varied between 1 and 100. Since the optimal solutions for these new instances are not known, we compare the feasible solution from the Lagrangian approach with a lower bound obtained through a piecewise linear approximation of the objective function. The results are shown in Tables 1.9 and 1.10.

For the piecewise linearization approach, the tables display the optimal value of the approximated problem (LB1), which constitutes a lower bound on the original problem, the upper bound (UB1) obtained by plugging the optimal solution of the approximated problem in the original problem, the relative optimality gap (Gap3 = (UB1-LB1)/LB1), and the computational time (CPU1) in seconds. For the Lagrangian approach, the tables

Table 1.9: Results for the location-inventory problem (500 node dataset)

Weights		Piecewise Approximation				Lagrangian Approach					
β	θ	LB1	UB1	Gap3	CPU1	LB2	UB2	Gap1	Gap4	CPU2	# of DCs
0.01	0.01	33,470.35	33,743.04	0.81%	10,000	32,031.63	33,645.65	5.04%	0.52	15.27	177
0.005	0.01	24,389.38	24,809.86	1.72%	10,000	23,280.75	24,616.10	5.74%	0.93%	35.93	123
0.001	0.01	11,033.85	11,517.74	4.39%	10,000	10,492.15	11,359.33	8.27%	2.95%	34.28	53
0.0005	0.01	7,961.20	8,440.71	6.14%	10,000	7,641.81	8,371.12	9.54%	5.15%	36.30	39
0.0001	0.01	3,513.22	3,797.58	8.09%	2,672	3,378.56	3,780.51	11.90%	7.61%	37.85	15
0.05	0.05	73,218.58	73,860.84	0.88%	10,000	68,145.10	73,639.76	8.06%	0.58%	46.44	333
0.005	0.05	25,832.59	27,172.24	5.19%	10,000	23,484.36	26,624.89	13.37%	3.07%	45.63	123
0.0005	0.05	8,626.98	9,665.79	12.04%	10,000	7,845.02	9,560.41	21.86%	13.90%	44.42	39
0.1	0.1	108,293.52	108,884.32	0.55%	10,000	99,449.33	108,543.98	9.15%	0.23%	38.93	405
0.01	0.1	36,667.02	38,709.51	5.57%	10,000	32,401.21	37,966.36	17.18%	3.54%	51.82	177
0.001	0.1	12,390.67	14,240.90	14.93%	10,000	10,860.03	13,851.45	27.54%	14.93%	99.76	53
1.0	1.0	620,407.76	620,473.54	0.01%	10,000	570,681.58	620,473.54	8.72%	0.01%	75.64	500
0.1	1.0	135,646.78	138,491.77	2.10%	10,000	101,049.41	137,067.65	35.64%	1.05%	74.08	405
0.01	1.0	49,344.68	59,021.84	19.61%	10,000	33,958.17	56,168.46	65.40%	17.25%	69.13	177
Min				0.01%	2672			5.04%	0.01%	15.27	
Avg				5.86%	9477			17.67%	5.12%	50.39	
Max				19.61%	10,000			65.40%	17.25%	99.76	

Table 1.10: Results for the location-inventory problem (1000 node dataset)

Weights		Piecewise Approximation				Lagrangian Approach					
β	θ	LB1	UB1	Gap3	CPU1	LB2	UB2	Gap1	Gap4	CPU2	# of DCs
0.01	0.01	45,031.46	45,539.33	1.13%	10,000	43,384.43	45,367.47	4.57%	0.75%	134.43	208
0.005	0.01	31,418.51	31,940.39	1.66%	10,000	30,131.44	31,755.52	5.39%	1.07%	250.85	141
0.001	0.01	13,155.63	13,900.23	5.66%	10,000	12,651.31	13,667.08	8.03%	3.89%	154.45	57
0.0005	0.01	9,262.32	9,914.34	7.04%	10,000	8,960.24	9,858.78	10.03%	6.44%	291.35	46
0.0001	0.01	4,078.67	4,446.57	9.02%	10,000	3,923.74	4,414.42	12.51%	8.23%	166.15	17
0.05	0.05	106,309.52	107,537.70	1.16%	10,000	99,737.69	107,053.62	7.34%	0.70%	75.84	469
0.005	0.05	32,735.81	34,730.15	6.09%	10,000	30,359.12	34,178.68	12.58%	4.41%	140.35	141
0.0005	0.05	9,827.82	11,460.29	16.61%	10,000	9,187.47	11,300.91	23.00%	14.50%	235.27	46
0.1	0.1	158,524.75	159,481.12	0.60%	10,000	146,467.95	158,981.24	8.54%	0.29%	282.09	605
0.01	0.1	50,058.23	51,129.74	2.14%	10,000	43,797.70	50,635.26	15.61%	1.15%	504.34	208
0.001	0.1	14,065.50	17,132.67	21.81%	10,000	13,063.03	16,566.63	28.82%	17.78%	434.83	57
1.0	1.0	829,583.85	829,776.57	0.02%	10,000	750,647.93	829,756.27	10.54%	0.02%	477.57	997
0.1	1.0	195,510.91	199,646.43	2.12%	10,000	148,257.22	197,814.49	33.43%	1.18%	603.89	605
0.01	1.0	60,280.49	76,871.19	27.52%	10,000	45,538.74	72,827.14	59.92%	20.81%	79.61	208
Min				0.02%	10,000			4.57%	0.02%	75.84	
Avg				7.33%	10,000			17.17%	5.80%	273.64	
Max				27.52%	10,000			59.92%	20.81%	603.89	

Table 1.11: Cost breakdown for the location-inventory problem (500 and 1000 node datasets)

Weights		500 node dataset				1000 node dataset			
β	θ	Loc.%	Trans.%	WI%	SS%	Loc.%	Trans.%	WI%	SS%
0.01	0.01	52.61	42.15	5.03	0.22	45.85	49.41	4.54	0.20
0.005	0.01	49.97	44.00	5.78	0.25	44.40	49.95	5.41	0.24
0.001	0.01	46.66	44.38	8.58	0.38	41.71	49.63	8.30	0.36
0.0005	0.01	46.59	42.90	10.07	0.44	46.66	42.52	10.36	0.45
0.0001	0.01	39.68	45.72	13.99	0.61	38.51	46.57	14.30	0.63
0.05	0.05	45.22	46.83	7.26	0.69	43.81	48.98	6.58	0.63
0.005	0.05	46.20	40.68	11.96	1.17	41.25	46.41	11.24	1.10
0.0005	0.05	40.79	37.57	19.71	1.93	40.70	37.10	20.22	1.98
0.1	0.1	37.31	53.81	7.84	1.04	38.05	53.69	7.29	0.96
0.01	0.1	46.62	37.35	14.09	1.94	41.08	44.27	12.88	1.78
0.001	0.1	38.26	36.40	22.25	3.08	34.41	40.95	21.65	3.00
1.0	1.0	8.06	83.49	6.45	2.00	12.02	78.09	7.55	2.34
0.1	1.0	29.55	42.61	19.63	8.20	30.58	43.15	18.52	7.74
0.01	1.0	31.51	25.25	30.11	13.13	28.56	30.78	28.31	12.35

display the Lagrangian lower bound (LB2), the upper bound (UB2) based on the feasible solution, the relative optimality gap (Gap1 = (UB2-LB2)/LB2), the relative gap between the feasible solution obtained from the Lagrangian approach and the solution of the piecewise linearization (Gap4 = (UB2-LB1)/LB1), the computational time (CPU2) in seconds , and the number of opened facilities (# of DCs). It is clear that, when compared to the piecewise linearization, the proposed Lagrangian approach is superior in terms of computational time due to the much smaller sizes of problems to solve. While most instances exceeded the CPU time limit of 10,000 seconds in the piecewise approximation approach, the average computational time for the Lagrangian approach is 50.39 and 273.64 seconds for the 500 and 1000 node datasets, respectively. Moreover, while the linear approximation approach often leads to a better lower bound, the Lagrangian approach was able to reach better feasible solutions in all cases. As the value of 'Gap4' is always greater than the real optimality gap for the Lagrangian approach, all feasible solutions obtained from it are within 21% of the optimal for the highest ratios of θ to β. But as shown in Tables 1.7 and 1.8, these solutions may in fact be very close to the optimal.

Table 1.11 outlines the contribution of each cost component in the total cost of the best solutions obtained. For each of the newly tested 500- and 1000- node instances, the columns titled 'Loc.', 'Trans.', 'WI' and 'SS' depict the ratios of the fixed location cost, the transportation cost, the working inventory cost, and the safety stock cost respectively to the total cost under different weights β and θ. The relative importance of each cost component varies widely depending on the selection of the cost weights. Again, it is obvious that as the nonlinear inventory components become more dominant, the computational time increases and the bound quality deteriorates.

The number of opened DCs is greatly affected by the contribution of different cost components as we change the cost weights θ and β. As anticipated, more DCs are opened when the cost of transportation is high compared to the inventory-related costs to reduce the shipment distances, whereas more retailers are served by each DC to take advantage of the concave inventory cost functions when the inventory costs are relatively large. For a constant θ/β ratio, more DCs are opened when the transportation and inventory costs are large compared to the fixed setup costs. These results agree with those noticed in [37].

Finally, to demonstrate the capability of the proposed approach to handle the general version of the problem we test on 500 and 1000 node instances with demand mean and variance that are not correlated across the retailers. All the problem parameters are set identical to the instances tested earlier (*i.e.*, Tables 1.9 and 1.10) except the demand variance which is set as $\sigma_i^2 = 2.\text{rand}_i.\mu_i$, where rand_i are uniformly distributed pseudo-random numbers over $(0, 1)$. Comparison of the results presented in Table 1.12 with those in Tables 1.9 and 1.10 shows no significant differences in solution quality or computational

Table 1.12: Results for the location-inventory problem (500 and 1000 node dataset when the demand mean and variance are uncorrelated)

Weights		500 node dataset				1000 node dataset			
β	θ	LB	UB	Gap1	CPU	LB	UB	Gap1	CPU
0.01	0.01	32,031.92	33,645.46	5.04%	37.35	43,384.32	45,364.45	4.56%	150.09
0.001	0.01	10,491.99	11,358.06	8.25%	35.54	12,651.42	13,667.27	8.03%	159.31
0.0001	0.01	3,378.47	3,780.35	11.90%	37.05	3,923.94	4,414.94	12.51%	177.54
0.1	0.1	99,447.64	108,476.09	9.08%	38.13	146,466.71	158,929.42	8.51%	470.69
0.01	0.1	99,447.64	108,476.09	9.08%	37.24	43,799.55	50,636.78	15.61%	156.06
0.001	0.1	10,857.78	13,836.25	27.43%	36.15	13,063.59	16,565.10	26.80%	156.33
1.0	1.0	570,693.92	619,499.50	8.55%	38.81	750,618.23	828,194.14	10.33%	151.71
0.1	1.0	101,036.30	136,365.90	34.97%	35.60	148,265.93	197,391.36	33.13%	148.09
0.01	1.0	33,965.65	55,940.49	64.70%	33.51	45,540.78	72,690.50	59.62%	152.76

time between the general case instances and their corresponding special case instances where μ and σ^2 are set equal. These results signify the generic nature of the approach and its ability to solve problems with several concave terms in the objective functions.

Through the aforementioned computational tests, we found that the proposed approach was able to reach high quality solutions for instances of the location-inventory problem from the literature efficiently and, unlike the classical solution approaches, could handle the general case (*i.e.*, uncorrelated demand mean and variance) of the problem. We were also able to solve new extremely large instances of up-to 1000 nodes to near-optimality in less than 5 minutes in average.

1.4 A Benders approach for concave minimization

1.4.1 Benders decomposition

Benders decomposition is a classical approach that is well-suited for a specific class of problems, namely problems with 'complicating variables' [20]. Had these variables been temporarily fixed, the resulting subproblem(s) would have been much easier to solve than the original problem. The idea is to use the dual solution of the subproblems obtained in every iteration to generate a cut that is added to a master problem. When solved, the master problem provides new 'trial' values for the complicating variables and a lower bound, whereas an upper bound and a new cut are obtained from the subproblem(s). The algorithm iterates between the master problem and the subproblem(s) until the upper and the lower bounds coincide, in which case the solution is declared optimal.

While the classical Benders decomposition was originally devised for mixed-integer programming problems (where fixing the integer variables renders the subproblem linear), the idea has been extended to other classes of problems. Geoffrion [56] has studied the case when the subproblems are concave for any value of the complicating variables and developed a method known as 'Generalized Benders decomposition' to generate valid cuts. On the other hand, when the subproblem is an integer program (meaning that the strong duality theory cannot be applied to obtain dual variables), the 'Logic-based Benders Decomposition' approach proposed by Hooker [63] can be used to generate 'logical' cuts from the integer subproblem directly.

In this section, we propose a Benders approach for solving $\mathcal{P}$ that is particularly effective when $m \ll n$, *i.e.*, the number of concave terms in the objective function is much smaller than the number of variables. This situation is not uncommon in practice. Consider, for instance, a complete digraph that has n nodes and $n(n-1)$ arcs. If the flow in arcs carries a linear cost per unit whereas nodes enjoy economies of scale, the number of concave terms in the cost function is much smaller than the number of linear terms for practical size problems. Another example is the subproblem resulting from relaxing the single assignment constraint in the inventory-location problem [37] which has only two concave terms (or one if the variance to mean demand ratio is the same for all retailers).

To present the approach, let us consider the (0-1) concave minimization problem

$$\begin{array}{lll} [\mathcal{P}]: & \min & c^T x + \sum_{i=1}^{m} f_i(d_i^T x) \\ & \text{s.t.} & Ax \geq b \\ & & x \in \{0,1\}^n \end{array}$$

where $c, d_i \in \mathbb{R}^n$, f_i are real concave functions over the feasible set, and $m \ll n$. Let $y_i = d_i^T x$. Benders decomposition is applied by fixing y to $\hat{y} = [\hat{y}_i]$ to get a (0-1) subproblem in x. Then we relax the integrality constraint to get the linear Benders subproblem

$$\begin{array}{llll} [\mathcal{BSP}]: & \min & c^T x & \\ & \text{s.t.} & Ax \geq b & (u) \\ & & d^T x = \hat{y} & (v) \\ & & Ex \geq e & (w) \\ & & x \geq 0. & \end{array} \tag{1.25}$$

Note that (1.25) are box constraints on x, *i.e.*, $x <= \mathbf{1}$. We use them later for branching purposes by fixing the elements of E and e to force x_j to 0 or 1. We proceed by taking the dual of $\mathcal{BSP}$ to get the dual subproblem

$$\begin{array}{lll} [\mathcal{DSP}]: & \max & b^T u + \hat{y}^T v + e^T w \\ & \text{s.t.} & A^T u + dv + E^T w \leq c \\ & & u, w \geq 0 \end{array}$$

Let H^p and H^r be the index sets of extreme points and extreme rays of the set $\{(u,v,w) : A^T u + dv + E^T w \leq c,\ u, w \geq 0\}$. By introducing the variables $\theta = \mathbf{v}[\mathcal{BSP}] = \mathbf{v}[\mathcal{DSP}] = \max_{h \in H^p}(b^T u^h + y^T v^h + e^T w^h)$, the Benders master problem is

$$
\begin{aligned}
[\mathcal{BMP}]: \quad & \min \quad && \theta + \sum_{i=1}^{m} f_i(y_i) && \\
& \text{s.t.} && \theta - v^h y \geq b^T u^h + e^T w^h && \forall h \in H^p && (1.26)\\
& && - v^h y \geq b^T u^h + e^T w^h && \forall h \in H^r && (1.27)\\
& && \underline{y}_i \leq y_i \leq \overline{y}_i && \forall i \in m && (1.28)
\end{aligned}
$$

It is not practical to generate the constraints corresponding to all extreme points and rays and include them in $\mathcal{BMP}$ at once. So we start with a subset of constraints which includes only (1.28) and solve a relaxed version of the master problem $\mathcal{RBMP}$ in every iteration to obtain a new set of y values that are sent back to $\mathcal{BSP}$. A Benders optimality cuts (1.26) is generated and appended to $\mathcal{BMP}$ every time a feasible subproblem solution is obtained. If a subproblems turns out to be infeasible (or, equivalently, its dual subproblem is unbounded), a Benders feasibility cut (1.27) is generated and appended instead, where the extreme ray (v^h, u^h, w^h) is obtained by solving the auxiliary subproblem

$$
\begin{aligned}
[\mathcal{ASP}]: \quad & \max \quad && 0 \\
& \text{s.t.} && A^T u + dv + E^T w \leq 0 \\
& && b^T u + \hat{y}^T v + e^T w = 1 \\
& && u, w \geq 0
\end{aligned}
$$

This approach shifts the nonlinearity of the objective function to the relaxed master problem, which has fewer variables and constraints than the original problem. Since $\mathcal{BMP}$ is a concave minimization problem, it has a global optimum at an extreme point of its feasible polytope. Therefore, it can be solved by evaluating the objective value at all vertices. The problem is of dimension $m + 1$, so vertices result from the intersection of $m + 1$ linearly independent hyperplanes. The optimal vertex at any iteration is found by enumerating possible combinations of $m + 1$ binding constraints at a time. Since only a single cut is added in every iteration, it is sufficient to use the new cut with combinations of m old cuts to generate all the new vertices. The newly found vertices are tested for feasibility with respect to the old constrains. Likewise, the old vertices carried forward from the previous iteration are filtered using the new cut. The set of feasible vertices are then evaluated to find the relaxed Master problem's optimal solution (θ, y) and a lower bound. Although this approach is equivalent to complete enumeration, only a few extreme

Table 1.13: Test functions for the Benders approach

	function	form
$f_1(x) =$	$-\sum_{j=1}^{n} x_j + \left(\sum_{j=1}^{n} (\sqrt{j} + \frac{1}{\sqrt{j}}) x_j \right)^{2/3}$	increasing
$f_2(x) =$	$\left(\sum_{j=1}^{n} j x_j - \left(\sum_{j=1}^{n} \sqrt{j} x_j \right)^2 \right) /1000$	decreasing
$f_3(x) =$	$-(n/10) \sum_{i=1}^{j} x_j + n \sin \left(\frac{2\pi \sum_{j=1}^{n} j x_j}{n(n+1)} \right)$	increasing then decreasing

points are generated and tested at every iteration. It has proven to be pretty efficient in practice.

Since $\mathcal{RBMP}$ is a relaxation of $\mathcal{BMP}$, which is in turn is a reformulation of $\mathcal{P}$, its solution provides a lower bound on the optimal solution of $\mathcal{P}$. On the other hand, the solution of $\mathcal{BSP}$ is feasible to $\mathcal{P}$, so $\mathbf{v}[\mathcal{BSP}] + \sum_{i=1}^{m} f_i(\hat{y}_i)$ provides an upper bound on the optimal solution of $\mathcal{P}$. The master and the subproblem are solved iteratively until the gap between the upper and lower bounds is sufficiently small. Upon solving the linearly relaxed subproblem, if some of x variables are fractional, branch-and-bound is used.

1.4.2 Numerical testing

To evaluate the proposed Benders approach, we test on instances with a single concave term in the objective function. Three concave functions of different forms over linear combinations of (0-1) variables are used for testing. The functions are depicted in Table 1.13. The polytopes $X = \{x : Ax \leq b\}$ containing the feasible points are generated as follows: the elements of A are uniformly distributed random numbers in the interval $[0, n]$. The right-hand side column vector is generated as $b = A\hat{x} + s$, where $\hat{x}$ and s are vectors of size n and m, respectively, of uniformly distributed random numbers from the interval $]0, 1[$. We generated instances of two sizes: (25×25) and (50×50), where the first figure represents the number of constraints and the second the number of variables. Five instances are solved for each size/function combination. To benchmark the performance of the proposed approach, we compare to the state-of-the-art global optimization solvers *Couenne* and *Baron* on *NEOS server* [2].

Table 1.14: Results for the Benders approach

ins	fun	size	Benders approach			Couenne		Baron	
			obj	cpu	nodes	obj	cpu		
1	1	25 × 25	-1.198	1.28	155	-1.198	1.42	-1.198	0.17
2	1	25 × 25	-2.160	0.41	41	-2.160	1.18	-2.160	1.10
3	1	25 × 25	-1.249	2.20	273	-1.249	1.64	-1.249	0.09
4	1	25 × 25	-1.276	1.83	227	-1.276	2.17	-1.276	0.37
5	1	25 × 25	-1.302	1.38	151	-1.302	2.21	-1.302	0.25
1	2	25 × 25	-2.006	5.12	1721	-2.006	1.07	2.006	0.45
2	2	25 × 25	-3.580	3.62	1.281	-3.580	1.15	-3.580	0.46
3	2	25 × 25	-2.413	1.40	445	-2.413	0.89	-2.413	0.42
4	2	25 × 25	-2.965	1.91	685	-2.965	0.64	-2.965	0.24
5	2	25 × 25	-2.643	5.95	2109	-2.643	0.81	-2.643	0.34
1	3	25 × 25	-12.417	0.18	21	-12.417	12.22	-	-
2	3	25 × 25	-23.199	13.38	2423	-23.199	10.58	-	-
3	3	25 × 25	-12.611	0.05	3	-12.611	29.81	-	-
4	3	25 × 25	-15.737	7.13	1079	-15.737	36.53	-	-
5	3	25 × 25	-12.711	9.81	1075	12.711	47.57	-	-
1	1	50 × 50	-3.812	11.77	685	-3.812	2,298.28	-3.812	0.89
2	1	50 × 50	-3.183	10.68	675	-3.183	5,151.51	-3.183	0.70
3	1	50 × 50	-3.609	4.60	259	-3.609	3,166.23	-3.609	0.19
4	1	50 × 50	-3.576	10.94	625	-3.567	3,171.26	-3.567	0.62
5	1	50 × 50	-3.204	6.84	383	-3.204	3,463.33	-3.204	0.65
1	2	50 × 50	-19.778	124.55	26217	-19.778	14.36	-19.778	2.63
2	2	50 × 50	-19.523	57.29	10819	-19.523	11.49	-19.523	3.06
3	2	50 × 50	-19.431	223.77	48305	-19.431	8.34	-19.431	0.84
4	2	50 × 50	-18.157	147.42	27245	-18.157	37.61	-18.157	8.92
5	2	50 × 50	-16,475[†]	10,000	114,697	-17.965	34.06	-17.965	9.08
1	3	50 × 50	-84.797	65.67	3,121	-84.104[†]	10,000	-	-
2	3	50 × 50	-80.635[†]	10,000	109,336	-79.759[†]	10,000	-	-
3	3	50 × 50	-82.329	105.33	4935	-81.474[†]	10,000	-	-
4	3	50 × 50	-81.958	115.25	4703	-80.863[†]	10,000	-	-
5	3	50 × 50	-77.704	313.18	15541	-77.143[†]	10,000	-	-

[†] suboptimal solution

The results are depicted in Table 1.14. The columns *ins*, fun, and *size* present the the instance number, the objective function type (according to Table 1.13), and the instance size ($m \times n$), respectively. For each of the three solution methods we provide the computational time in seconds (*cpu*) and the best objective value reached (*obj*). For the Benders approach we also provide the number of nodes in the B&B tree (*nodes*).

The proposed Benders approach was able to solve 28 out of the 30 tested instances to proven optimality in the cut-off time of 10,000 seconds. The computational time varies widely between the instances depending on their size and objective function type. In absolute terms, the Benders approach performed well with the smaller instances and with function 1 (the increasing function). The average computational times for the smaller (25×25) instances are 1.42, 3.6 and 6.11 seconds for the three function types, respectively. The average computational time for the larger (50×50) instances with function 1 is 8.97 seconds. The larger instances with functions 2 and 3 took significantly longer times to solve, where one instance of each function type could not be solved to optimality.

When compared with the two commercial solvers *Couenne* and *Baron* one has to keep in mind that the instances were solved on servers with very different characteristics. Therefore, the computational times reported should be used with caution and considered indicative only. When the computational time is reported as 10,000 seconds, it means that the solver was not able to find the optimal solution within the cut-off time. From the tests conducted, it seems that *Baron* has a wide performance edge over both *Couenne* and our Benders approach, with a serious drawback of its inability to handle the *sine* function even in a period where it is purely concave. For the smaller instances, *Couenne* has, on average, a performance comparable to our approach for the increasing function and a better performance for the decreasing function. However, for the *sine* function, our approach substantially outperformed *Couenne* with a computational time ratio of approximately 2:9. For the larger (50×50) instances, our approach exhibited impressive performance compared to *Couenne* with the first and third functions, but lagged behind with the second. In fact, our approach is the only one that could solve the larger instances with a *sine* function to proven optimality in 4 out of the 5 tested instances. In comparison, *Couenne* reached suboptimal solutions in all tested instances with optimality gaps ranging between 18.3% and 21.2%.

In general, the performance of the Benders approach is quite encouraging, given that the approach could be enhanced further, *e.g.*, by adding cuts. The approach has to be tested on larger instances and with multiple concave terms to assess its effectiveness.

1.5 Conclusions

In this chapter, we presented two new approaches for solving an important class of concave minimization problems over a polytope that appears frequently in supply chain design models. Furthermore, we demonstrated the application of the first on three supply chain network design models from the literature.

The first approach is a Lagrangian decomposition approach embedded in a branch-and-bound algorithm to solve problems with concave terms over linear combinations of the decision variables in their objective functions. The problem is decomposed into a linear-objective subproblem over the same feasible set and a set of easily-solvable single-variable concave minimization problems. The first subproblem is solved using efficient LP/MIP commercial solvers. A closed-form expression for the optimal Lagrangian multipliers is provided, enabling the calculation of the Lagrangian bound in a single iteration. Since the solution of the linear-objective subproblem is feasible to the original problem, an upper bound is obtained immediately. The approach is then embedded in a B&B algorithm where the Lagrangian lower bound and the feasible solution are used for bounding and branching. The approach is tested on problems from the literature and is shown to perform quite well on problems with different concave functions and feasible regions, including continuous and discrete. The Lagrangian approach could be extended to handle different forms of nonlinearity such as quasi-concave functions, as long as the second subproblem yields solutions at the boundaries.

The proposed Lagrangian approach is then used to tackle the production-transportation problem, the plant location and technology acquisition problem, and the location-inventory problem. The results are compared with those obtained using other algorithms, highlighting the ability of the Lagrangian approach to handle a wide array of practical problems effectively.

Finally, we propose a new Benders approach to tackle concave minimization problems over a polytope, when the number of concave terms in the objective function is much smaller than the number of constraints. Our approach is different from the classical Benders decomposition in that the subproblem is integer and the master problem is concave with continuous variables. To overcome the integrality issue, we solve the linear relaxation of the subproblem and then use B&B to restore integrality. The master problem is solved through enumerating the feasible region vertices iteratively. The approach is tested on random instances using different objective functions and was found to outperform some of the well-known commercial solvers, especially when nonlinear convex relaxations of the functions are not readily available in the solver (as the case of trigonometric functions).

Chapter 2

Cold supply chain design: a simulation-optimization approach

2.1 Introduction

[1] Global warming has become a pressing issue in the last few decades, particularly with the plethora of recent scientific research providing strong evidence for its existence and showing its severe negative effects [93]. Currently, specialists concur that the radioactive forcing attributed to the anthropogenic greenhouse gas (GHG) emissions is the main cause of the global warming phenomenon [6]. Thus, many endeavors have been made by governments, organizations and firms around the world to reduce the emissions of GHG.

In many industries, supply chain operations are a significant source of GHG emissions. It was estimated that more than three quarters of the GHG emissions associated with many industrial sectors are attributed to supply chain activities [67]. Companies have devoted considerable attention to reduce the environmental footprint of their supply chains, aiming to achieve their sustainability commitments, mitigate risk on their brand value, and satisfy their environmentally-conscious customers [8]. Nowadays, several global companies including IBM, Johnson&Johnson and PepsiCo require their suppliers to report or control their GHG emissions, whereas other companies are taking steps to control their supply chain emissions. For example, Wal-Mart has recently announced that it is on track to reduce GHG emissions from its supply chain by 18 million metric tons by 2015 [4].

[1] Some material in this chapter has been accepted for publication in: A. Saif, S. Elhedhli, Cold supply chain design with environmental considerations: A simulation-optimization approach, European Journal of Operational Research (2015), http://dx.doi.org/10.1016/j.ejor.2015.10.056.

In cold supply chains, products must be stored and transported at low temperatures near or below the freezing mark. This necessitates the use of refrigerated warehouses and trucks that consume large quantities of energy for refrigeration. Higher energy consumption is associated with higher carbon dioxide (CO2) emissions in power generation facilities. Furthermore, refrigeration systems utilize large quantities of HydroFluoroCarbon (HFC) gases that have high global warming potential (GWP) and very long lifetime in the atmosphere. Regular and catastrophic leakage of HFC gases from cold supply chain constitutes a significant components of the global warming impact. Therefore, these gases must be taken into account when determining the best design and operations of cold supply chains.

In this chapter, we study the cold supply chain design problem and provide a mathematical model to represent its economic and environmental effects. The problem is formulated as a concave mixed-integer programming problem, where the objective is to minimize the expected total cost of the supply chain, including capacity, transportation and inventory costs, in addition to costs associated with the global warming impact due to GHG emissions. We consider the environmental effects of both CO2 emissions due to energy consumption and leakage of refrigerant gas in warehouses and vehicles.

To solve the model, we propose a novel Lagrangian approach embedded in a branch-and-bound framework. Unlike classical Lagrangian relaxation approaches that use iterative methods such as subgradient optimization or cutting plane methods, we are able to provide a closed-form expression for the best Lagrangian multipliers, so we get the Lagrangian bound in a single iteration. Since the solution of the main subproblem is feasible to the original problem, we also get an upper bound immediately. A branch-and-bound algorithm is used to close the optimality gap. The proposed approach requires the evaluation of the inventory cost and the maximum inventory level at the branching points. Since we address the case of general demand pattern and inventory policy where explicit formulas for the inventory functions are rarely available, we resort to a simulation-optimization algorithm to estimate these functions. Discrete-event simulation is embedded into a bisection search algorithm to find the best control parameters of the inventory system.

The main contributions of this chapter are:

1. A new mathematical formulation for the cold supply chain design problem with dual objectives of minimizing the total cost and the global warming impacts, including those associated with the usage of refrigerant gases. The proposed formulation takes into consideration economies-of-scale and bases the warehouse capacity on the physical inventory holding requirements, leading to a mixed integer concave minimization problem.

2. A novel solution approach for the cold supply chain design problem that combines the Lagrangian decomposition proposed developed in section 1.2 with a simulation-optimization algorithm in a branch-and-bound framework. The proposed solution approach can handle the case of general demand distribution and inventory policy when the inventory functions cannot be expressed explicitly. The computational performance of the proposed approach is tested on a hypothetical case under a wide range of scenarios.

3. Important managerial insights are drawn from testing the proposed approach on two realistic case studies from different industries. The tradeoffs between the economic and environmental considerations and between the different cost components in each case are studied. It has been shown that significant reductions in the global warming impact of cold supply chains can been achieved at a small marginal increase in the cost.

2.2 Literature review

2.2.1 Supply chain design with environmental considerations

Until the last decade, little attention was given to the environmental impact of supply chains. As noted by Current et al. [34], only a few supply chain network design papers have included environmental metrics in their objective functions. Furthermore, these were *ad hoc* models designed for specific applications and not generic ones. However, with the escalating pressure from both governments and consumers to reduce the environmental footprint, the interest in designing green supply chains has risen sharply in the last decade. Several aspects of green supply chains were considered in the literature, including reverse supply chains, green manufacturing and remanufacturing, and environmentally-conscious lot-sizing.

Incorporating environmental aspects in supply chain design necessarily entails a trade-off between economic and environmental objectives. However, as shown by Benjaafar et al. [21], by making minor operational changes, it is possible to achieve vast reductions in the environmental footprint of supply chains without significantly increasing the cost. A variety of supply chain design models that incorporated GHG emission minimization as an objective have appeared recently in the literature; each was based on certain assumptions and has a specific focus. Ramudhin et al. [84] proposed a green supply chain design model that integrates carbon trading considerations but assumed that facility locations and sizes are known in advance. Conversely, Diabat and Simchi-Levi [38] considered a similar carbon-capped supply chain design problem that treats the manufacturing and storage capacity of the manufacturers as variables, but does not account for the possibility of carbon trading. Harris et al. [61] considered a multi-objective variant of the traditional uncapacitated facility location problem with economic and environmental objectives and implemented an evolutionary algorithm to find a set of non-dominated solutions. Bin and Jun [26] presented a nonlinear MIP model for a green supply chain design, showing the positive economic and environmental effects of implementing e-commerce on the supply chain operations. A more detailed and sophisticated multi-objective model that embeds life-cycle assessment (LCA) concepts within the supply chain design process is presented by Bojarski et al. [27]. The strategic decisions addressed in the model are facility location, processing technology selection and production/distribution planning. Trade-offs between the total cost and the environment influence are thoroughly studied by Wang et al. [107], who applied a normalized normal constraint method to find a set of evenly distributed Pareto optimal solutions and studied their sensitivity to the problem parameters. The closed-loop supply chain design framework proposed by Chaabane et al. [30] combined multiple aspects from the previous references, including LCA and emission trading. Their MIP model takes into account the economic and environmental costs of manufacturing, distribution, warehousing and recycling activities. For a recent survey on location models within a supply chain environment, the reader is referred to [76].

Within the current literature, three observations are worth mentioning. First, most of the models address the strategic location-allocation decisions in isolation from tactical ones such as inventory and routing decisions; Therefore they fail to exploit the significant potential savings in economic and environmental costs that can be achieved by considering the two levels of decisions simultaneously. Second, with very few exceptions (*e.g.*, [43]), the literature has assumed that the environmental impact of the supply chain is linearly proportional to its scale. A more accurate and realistic approach is to consider economies of scale inherent in supply chain operations, leading to non-linear models. A final critique of the supply chain design models that consider emissions is that they have a narrow focus on CO2 as the only GHG that deserves attention, ignoring the strong global warning impact of other GHGs. This impact is especially important in cold supply chains that use large quantities of refrigerant gases, which are known to have very high GWP. Although including refrigerant gases beside CO2 in the emissions minimization objective function does not fundamentally change its general form, it significantly increases its magnitude and alters its specific mathematical form, leading to different results and managerial implications.

In the remainder of this chapter, we incorporate these three features into a mathematical model that combines strategic location-allocation and tactical inventory decisions, takes into consideration the nonlinearities resulting from both economies of scale and the risk pooling effect, and incorporates the global warming impact of other GHGs beside CO2. We also propose an approach that combines Lagrangian decomposition, simulation-optimization approach to solve the resulting model.

2.2.2 Simulation-optimization methods for supply chain design problems

Combining simulation and optimization methods to solve stochastic problems is an increasingly popular approach. While mathematical programming/optimization methods can handle a large number of variables efficiently and rigourously, simulation methods are favorable in dealing with realistic complex, stochastic, and dynamic systems when it is extremely difficult (or even impossible) to depict complex relationships between the components of a system explicitly.

In the context of supply chain network design problems, simulation and optimization methods were combined through two main approaches:

- Optimization and simulation models are used iteratively to tune the parameters of each other. [78] used this approach to solve a deterministic supply chain planning and scheduling problem that entails minimizing the production, transportation, and inventory holding costs subject to capacity and inventory balance constraints. Independent optimization and simulation models were developed for the problem, then the two models were linked through a set of variables/parameters, by which some outputs of the optimization models are used as inputs for the simulation model, and vice-versa. Iterating between the two models continue until the total cost in both converge. Similar approaches were pursued by [13], [85], [72], and [10], among others, to solve a supply chain network design problems.

- The problem is decomposed to two subproblems, where one is tackled through optimization and the other through simulation. This approach is often used when it is difficult to tackle the problem directly through optimization due to specific model attributes (*e.g.*, stochastic parameters, unknown function, etc.). Jung et al. [68] used this approach to solve stochastic supply chain planning and scheduling problem.

The approach we present in this work belongs to the second class. Lagrangian decomposition is used to isolate the stochastic and non-linear terms in the objective function so they can be estimated independently through simulation, whereas the remaining part is solved as a deterministic MIP.

2.3 Effect of refrigerant gases in cold supply chains

The Kyoto protocol requires countries to commit to specific reductions in the emissions of six types of GHG. Although CO2, among them, contributes the most to global warming due to the vast quantities of its emissions resulting from fossil fuel use, other gases play an important role and may become a dominant factor in global warming in the near future. In particular, concentrations of many of the fluorine-containing GHGs (HFCs, PFCs, SF6) increased by a factor of 1.3 to 4.3 between 1998 and 2005, and their total radiative forcing is rapidly increasing by roughly 10% per year [93]. The global warming impact of a greenhouse gas is determined by two factors: its concentration in the atmosphere and its GWP, defined as the ratio of the time-integrated radiative forcing from the instantaneous release of 1 kg of this specific gas to that from the release of 1 kg of carbon dioxide. Despite the fact that the concentrations of these gases in the atmosphere are low compared to those of CO2, many of them have very high GWP. For example, HFC-134a, a haloalkane refrigerant commonly used in domestic refrigerators and automobile air conditioners, has a GWP of 3830 over 20 years. Although the concentration of HFC-134a in 2005 was only about 10^{-7} of that of carbon dioxide, this concentration has increased by 349% between 1998 and 2005 compared to an increase of only 13% of the carbon dioxide concentration during the same period. Velders [104] estimated that if the objectives of reducing CO2 emissions are accomplished but nothing is done about HFCs, they will be responsible for between 28% and 45% of carbon-equivalent emissions by 2050. Even if no action is taken on CO2, HFCs will still be responsible for between 10% and 20% of carbon-equivalent emissions by 2050.

Among different applications, air conditioning (domestic and automotive) and refrigeration (domestic, industrial, commercial and transport) use, by far, the largest quantities of HFC gases. Food processing, cold storage and transportation applications contain about two thirds of the total HFC quantities used in refrigeration applications and generate about the same percentage of HFC emissions [7]. Therefore, incorporating the contribution of refrigerant gases while calculating the total global warming impact is particulary important for firms using these gases extensively. For instance, it has been estimated that, in an environment with an average energy mix, the refrigerant emissions represent about 60% of the total emissions of GHG resulting from refrigeration system operation in the commercial sector, whereas the rest is indirect emissions caused by power production [7]. Since most of the refrigerant emissions take place in the transportation and warehousing stages, it is crucial to incorporate the greenhouse impact of refrigerant gases in the logistics/supply chain design models developed, to minimize or control the environmental damage caused by these firms.

2.4 Cost and global warming effect of cold storage warehouses

The set-up and operational costs of conventional (non-chilled) warehouses are known to exhibit strong economies of scale, and thus were modeled as concave functions of their volumetric capacity [49]. However, classical capacitated facility location models assumed that the capacities of potential warehouses are determined exogenously and known *a priori*, such that these capacities only impose limits on demand quantities (*i.e.*, throughput) handled by warehouses [105]. Even when the warehouse capacity was considered a variable to be determined by the program, it was linked directly to the throughput via a linear [55] or a concave function [77], and was not based on the actual physical stock holding requirements. This treatment of the warehouse capacity is justifiable for conventional warehouses as the capacity cost is usually modest. However, upon adding the cost of the refrigeration system to the capacity cost in cold supply chains, the latter substantially increases, and consequently, the throughput-based sizing approach may not be suitable for determining the optimal capacity accurately. Therefore, in our model we use a novel approach to determine the warehouse capacity based on the actual storage requirements dictated by the demand pattern, inventory policy, and customer service requirements.

It is easy to show that the cost and global warming impacts of the refrigeration system are also concave functions of the warehouse volumetric capacity. First, there are economies of scale with respect to the equipment sizes required to serve a certain cooling load. Many refrigeration system components, such as motors, compressors, condensers, values, and control units, cost less per unit size as they grow larger. Moreover, the mechanical, electrical, and thermal efficiencies of the refrigeration system components are usually higher for larger sizes, thereby reducing their energy consumption per unit of refrigeration load served. Second, The relationship between the volumetric capacity of a cold store and its refrigeration load is concave. This relationship results from two major heat load components that increase sub-proportionally with the cold store volume, namely the external heat and the infiltration loads. The external heat load is approximately proportional to the surface area of the cold store, whereas the number of air changes in a cold store has an inverse relationship with its volume [5].

The amount of refrigerant gas leakage is usually estimated as a percentage of the amount of the refrigerant charged (*i.e.*, refrigerant bank) [7]. The amount of refrigerant required per ton of refrigeration is a constant referred to as the 'net refrigeration effect' for each refrigerant type [60], implying that the leakage quantity is proportional to the cooling capacity of the system. Again, this leads to the conclusion that the leakage quantity is a concave function of the volumetric capacity of the warehouse.

The two environmental effects of the refrigeration system can be combined in a single metric known as the total equivalent warming impact (TEWI), defined as the sum of the direct equivalent (chemical emissions) and the indirect (energy use) global warming effects [51].

2.5 Model formulation

Let us define indices $i = 1, \ldots, m$, $j = 1, \ldots, n$, $k = 1, \ldots, p$ and $l = 1, \ldots, q$, corresponding to retailers/customers, potential warehouse locations, production plants and products, respectively. Each plant has sufficient capacity to satisfy the total demand, so plant capacity is not an issue. This infinite capacity assumption is not restrictive and can be easily relaxed by adding a plant capacity constraint that has no effect on the solution approach. The number, location and size of warehouses to open is not known in advance. There is a fixed annual cost a_j for opening a warehouse at location j. In addition, there is a volume-dependent capacity cost represented by a concave function $f_j(.)$ to capture economies of scale. Products are shipped from plants to open warehouses in bulk quantities using single-type long-haul trucks, so c_{jkl}, the cost of shipping one unit of product l from plant k to warehouse j, is constant. Each unit of product l has volume u_l. To ensure better customer service, each retailer is assigned to a single warehouse for serving its requirements of all products. No restrictions are placed on the demand pattern of retailers, *i.e.*, time and composition of orders can follow any probability distribution. However, we assume that the planner can characterize this distribution with reasonable accuracy. Let d_{il} be the expected annual demand of product l from retailer i. Ordered products are shipped to customers using trucks of different types, depending on order size and distance, so the transportation cost is not linearly proportional to the number of units shipped. However, since the demand distribution of each customer is known, r_{ij}, the annual cost of serving customer i from warehouse j, can be estimated. Finally, warehouses implement an inventory policy that aims to minimize the total inventory cost, including: ordering, holding, and backordering or lost sales costs. Upon knowing the inventory policy in each warehouse, we can construct its total inventory cost function $t_{jl}(.)$ and the maximum inventory level function $s_{jl}(.)$ for every product; both are functions of the total demand served by the warehouse. The maximum inventory level depends on the inventory policy implemented by warehouses. For example, if a warehouse uses a base-stock policy or an (s, S) policy, it is equal to the order-up-to level (R or S, respectively); whereas for reorder point (Q, R) or (R, nQ) policies, that maximum is equal to $Q + R$, and so on.

To model the problem, we introduce the following variables:
x_{ij} : binary variable takes value 1 if retailer i is assigned to warehouse j.
y_{jkl}: units of product l shipped from plant k to warehouse j.
z_j : binary variable takes value 1 if a warehouse is opened in location j.

The expected total annual cost of the system includes: warehouse capacity cost (fixed and size-dependent), transshipment cost between plants and warehouses, and between warehouses and retailers, and inventory costs (*i.e.*, ordering, holding, and backordering or lost sales cost), which can be formulated as:

$$\begin{aligned} TC &= \sum_{j=1}^{n} a_j z_j + \sum_{j=1}^{n}\sum_{k=1}^{p}\sum_{l=1}^{q} c_{jkl} y_{jkl} + \sum_{j=1}^{n}\sum_{i=1}^{m} r_{ij} x_{ij} \\ &+ \sum_{j=1}^{n}\sum_{l=1}^{q} t_{jl} \left(\sum_{i=1}^{m} d_{il} x_{ij} \right) + \sum_{j=1}^{n} f_j \left(\sum_{l=1}^{p} u_l s_{jl} \left(\sum_{i=1}^{m} d_{il} x_{ij} \right) \right) \end{aligned}$$

Likewise, the global warming effect of the system is the result of GHG emissions through the shipment of units between plants and warehouses, and between warehouses and retailers, and GHG emissions corresponding to the warehousing of products. The schematic diagram in Fig.2.1 illustrates the different cost and GHG emission components.

Let $eCO2_{jkl}$ be the average CO2 emissions corresponding to the shipment of one unit of product l between plant k and warehouse j and $eHFC_{jkl}$ be the average HFC gas leakage per unit of product l shipped between plant k and warehouse j. The combined global warming effect of both emitted gases is quantified using the principle of equivalence between the emission of a unit of HFC and GWP_{HFC} units of CO2, where GWP_{HFC} is the GWP of HFC described earlier. Thus, e_{jkl}, the CO2-equivalent amount of GHGs emitted for shipping a unit between a plant and a warehouse is calculated as:

$$e_{jkl} = GWP_{HFC}.eHFC_{jkl} + eCO2_{jkl}$$

Likewise, o_{ij}, the annual CO2-equivalent emissions for serving retailer i from warehouse j, b_j, the annual fixed CO2-equivalent emissions from warehouse j, and the concave function $g_j(.)$, the annual CO2-equivalent emissions from a warehouse as a function of its volume, can be found in the same way. Thus the total CO2-equivalent emissions of the system can be expressed mathematically as:

$$TE = \sum_{j=1}^{n} b_j z_j + \sum_{j=1}^{n}\sum_{k=1}^{p}\sum_{l=1}^{q} e_{jkl} y_{jkl} + \sum_{j=1}^{n}\sum_{i=1}^{m} o_{ij} x_{ij} + \sum_{j=1}^{n} g_j \left(\sum_{l=1}^{q} u_l s_{jl} \left(\sum_{i=1}^{m} d_{il} x_{ij} \right) \right)$$

Let w be the weight assigned to the emissions minimization objective, which can be the price of carbon emissions in the context of a carbon trading scheme. Thus, to simultaneously minimize the total cost and the total CO2-equivalent emissions the objective function is: $TC + w.TE$. With that, the problem is formulated as:

$$
\begin{aligned}
[\mathcal{P}]: \quad \min \quad & \sum_{j=1}^{n} \hat{a}_j z_j + \sum_{j=1}^{n}\sum_{k=1}^{p}\sum_{l=1}^{q} \hat{c}_{jkl} y_{jkl} + \sum_{j=1}^{n}\sum_{i=1}^{m} \hat{r}_{ij} x_{ij} + \sum_{j=1}^{n}\sum_{l=1}^{q} t_{jl} \left(\sum_{i=1}^{m} d_{il} x_{ij} \right) && (2.1)\\
& \quad + \sum_{j=1}^{n} \hat{f}_j \left(\sum_{l=1}^{q} u_l s_{jl} \left(\sum_{i=1}^{m} d_{il} x_{ij} \right) \right) \\
\text{s.t.} \quad & \sum_{j=1}^{n} x_{ij} = 1, && \forall i, \quad (2.2)\\
& \sum_{k=1}^{p} y_{jkl} = \sum_{i=1}^{m} d_{il} x_{ij}, && \forall j, l, \quad (2.3)\\
& x_{ij} \leq z_j, && \forall i, j, \quad (2.4)\\
& x_{ij}, z_j \in \{0,1\}, \quad \forall i, j, \quad y_{jkl} \geq 0, && \forall j, k, l, \quad (2.5)
\end{aligned}
$$

where $\hat{f}_j(.) = f_j(.) + wg_j(.)$, $\hat{c}_{jkl} = c_{jkl} + we_{jkl}$, $\hat{r}_{ij} = r_{ij} + wo_{ij}$, and $\hat{a}_j = a_j + wb_j$. The first constraint ensures that every retailer is assigned to exactly one warehouse. The second constraint represents the quantity balance for every warehouse and product. Whereas the last constraint stipulates that a retailer can not be assigned to a warehouse unless it is open. This model resembles the single-echelon, multi-commodity supply chain design problem in [88], which also considered economies of scale within the supply chain using a concave term in the cost function. A similar problem structure is also analyzed in the more recent work of Berman, Krass, and Menezes [24] on inventory location problems.

In order to find r_{ij} and o_{ij}, the annual cost and carbon-equivalent emissions for serving retailer i by warehouse j, we assume that each retailer is served individually, *i.e.*, a truck is sent to a single retailer to fulfill its order. This is a reasonable assumption when the order frequency is low (*i.e.*, it is quite unlikely to have several adjacent retailers ordering at the same day), when the shipment sizes are large, or when the retailers are widely dispersed. Let $V = \{1, \ldots, v\}$ be the set of different vehicle types that can be used to ship products between warehouses and retailers. Upon knowing the distribution of demand from each retailer, we can determine the expected annual number of shipments using a specific type of trucks:

$$E(N_v) = E(N).\Pr(U_{v-1} < \sum_{l=1}^{q} N_l u_l \leq U_v).1_{v \in V_{ij}}$$

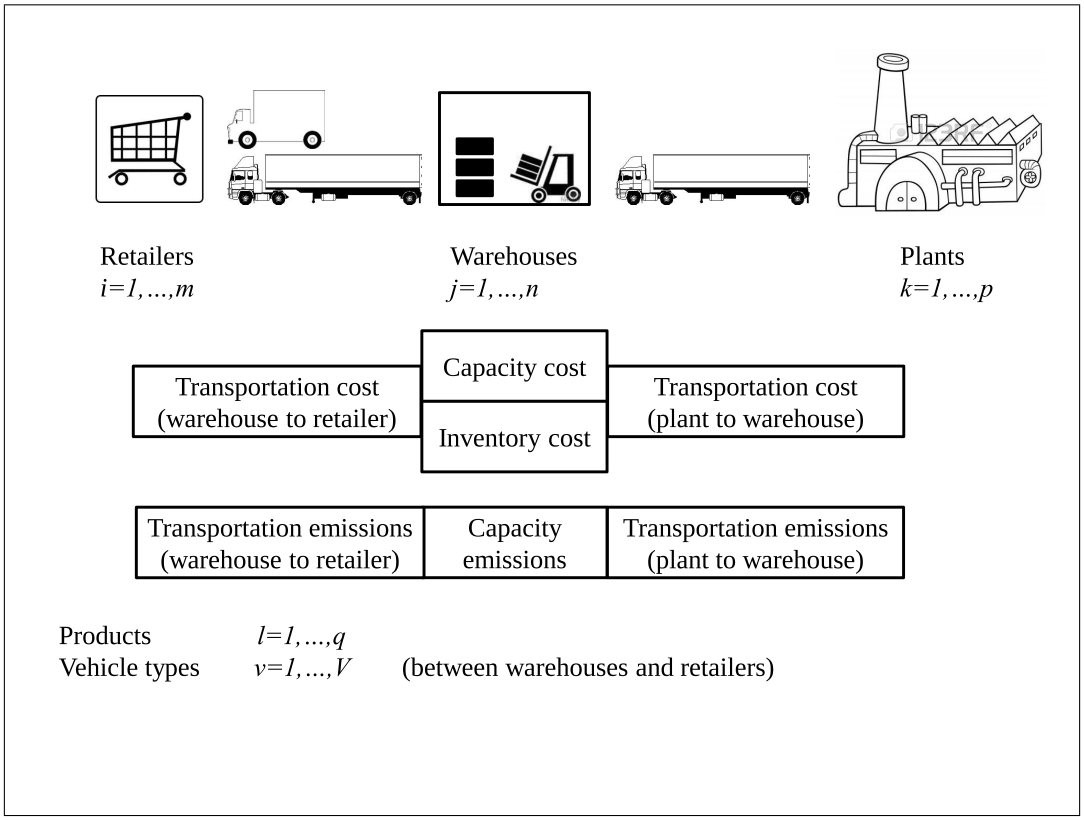
Retailers
i=1,...,m
Warehouses
j=1,...,n
Plants
k=1,...,p
Capacity cost
Transportation cost (warehouse to retailer)
Inventory cost
Transportation cost (plant to warehouse)
Transportation emissions (warehouse to retailer)
Capacity emissions
Transportation emissions (plant to warehouse)
Products l=1,...,q
Vehicle types v=1,...,V (between warehouses and retailers)

Figure 2.1: Components of the total cost and GHG emissions of the cold supply chain

where N_v is the number of shipments using truck type v, U_v and U_{v-1} are the volumetric capacity of truck types v and its next smaller type $v-1$, N_l is the number of units shipped of product l, and $1_{v \in V_{ij}}$ is an indicator function taking value 1 if this truck types belong to the set V_{ij} of trucks that can be used between i and j, since some truck types (*e.g.*, electric trucks) can be used only within a certain travel range.

2.6 Solution method

2.6.1 Lagrangian decomposition

The solution methodology is build on the assumption that the last two terms in the objective function of $\mathcal{P}$ are concave functions. We discuss the validity of this assumption in section 2.6.3. Let us define the new auxiliary variables $Q_j = \sum_{l=1}^{q} u_l s_{jl} \left(\sum_{i=1}^{m} d_{il} x_{ij}\right)$. Thus, $\mathcal{P}$ can be written as:

$$\begin{aligned}
\min \quad & \sum_{j=1}^{n} \hat{a}_j z_j + \sum_{k=1}^{p}\sum_{j=1}^{n}\sum_{l=1}^{q} \hat{c}_{jkl} y_{jkl} + \sum_{j=1}^{n}\sum_{i=1}^{m} \hat{r}_{ij} x_{ij} + \sum_{j=1}^{n}\sum_{l=1}^{q} t_{jl} \left(\sum_{i=1}^{m} d_{il} x_{ij}\right) + \sum_{j=1}^{n} \hat{f}_j \left(Q_j\right) \\
\text{s.t.} \quad & (2.2) - (2.5), \\
& Q_j = \sum_{l=1}^{q} u_l s_{jl} \left(\sum_{i=1}^{m} d_{il} x_{ij}\right), \qquad \forall j, \qquad (2.6) \\
& 0 \le Q_j \le \sum_{l=1}^{q} u_l s_{jl} \left(\sum_{i=1}^{m} d_{il}\right), \qquad \forall j. \qquad (2.7)
\end{aligned}$$

Since $s_{jl}(.)$ is a non-decreasing function in the total demand $\left(\sum_{i=1}^{m} d_{il} x_{ij}\right)$, Q_j reaches its upper bound when all the demand is served by a single warehouse, *i.e.*, $Q_j = \sum_{l=1}^{q} u_l s_{jl} \left(\sum_{i=1}^{m} d_{il}\right)$. Upon relaxing (2.6) with multipliers λ_j, $j = 1, \ldots, n$, we get the subproblem

$$\begin{aligned}
[\mathcal{SP}0]: \quad \min \quad & \sum_{j=1}^{n} \hat{a}_j z_j + \sum_{k=1}^{p}\sum_{j=1}^{n}\sum_{l=1}^{q} \hat{c}_{jkl} y_{jkl} + \sum_{j=1}^{n}\sum_{i=1}^{m} \hat{r}_{ij} x_{ij} + \sum_{j=1}^{n}\sum_{l=1}^{q} t_{jl} \left(\sum_{i=1}^{m} d_{il} x_{ij}\right) \\
& + \sum_{j=1}^{n} \hat{f}_j \left(Q_j\right) + \sum_{j=1}^{n} \lambda_j \sum_{l=1}^{q} u_l s_{jl} \left(\sum_{i=1}^{m} d_{il} x_{ij}\right) - \sum_{j=1}^{n} \lambda_j Q_j \\
\text{s.t.} \quad & (2.2 - 2.5), (2.7).
\end{aligned}$$

Next, we introduce the auxiliary variables $P_{jl} = \sum_{i=1}^{m} d_{il} x_{ij}$, so $\mathcal{SP}0$ can be written as

$$\begin{aligned}
\min \quad & \sum_{j=1}^{n} \hat{a}_j z_j + \sum_{k=1}^{p}\sum_{j=1}^{n}\sum_{l=1}^{q} \hat{c}_{jkl} y_{jkl} + \sum_{j=1}^{n}\sum_{i=1}^{m} \hat{r}_{ij} x_{ij} + \sum_{j=1}^{n}\sum_{l=1}^{q} t_{jl}(P_{jl}) \\
& + \sum_{j=1}^{n} \hat{f}_j(Q_j) + \sum_{j=1}^{n} \lambda_j \sum_{l=1}^{q} u_l s_{jl}(P_{jl}) - \sum_{j=1}^{n} \lambda_j Q_j \\
\text{s.t.} \quad & (2.2 - 2.5), (2.7), \\
& P_{jl} = \sum_{i=1}^{m} d_{il} x_{ij}, \qquad \forall j, l, \qquad (2.8) \\
& 0 \le P_{jl} \le \sum_{i=1}^{m} d_{il}, \qquad \forall j, l, \qquad (2.9)
\end{aligned}$$

By relaxing (2.8) with multipliers μ_{jl}, we get the subproblem

$$\begin{aligned}
[\mathcal{SP}00]: \quad \min \quad & \sum_{j=1}^{n} \hat{a}_j z_j + \sum_{k=1}^{p}\sum_{j=1}^{n}\sum_{l=1}^{q} \hat{c}_{jkl} y_{jkl} + \sum_{j=1}^{n}\sum_{i=1}^{m} \hat{r}_{ij} x_{ij} + \sum_{j=1}^{n}\sum_{l=1}^{q} t_{jl}(P_{jl}) \\
& + \sum_{j=1}^{n} \hat{f}_j(Q_j) + \sum_{j=1}^{n} \lambda_j \sum_{l=1}^{q} u_l s_{jl}(P_{jl}) - \sum_{j=1}^{n} \lambda_j Q_j \\
& + \sum_{j=1}^{n}\sum_{l=1}^{p} \mu_{jl} \sum_{i=1}^{m} d_{il} x_{ij} - \sum_{j=1}^{n}\sum_{l=1}^{q} \mu_{jl} P_{jl} \\
\text{s.t.} \quad & (2.2 - 2.5), (2.7), (2.9).
\end{aligned}$$

With that, $\mathcal{SP}00$ can be decomposed into $n(q+1)+1$ subproblems. The first is an integer programming problem in the original variables

$$\begin{aligned}
[\mathcal{SP}1]: \quad \min \quad & \sum_{j=1}^{n} \hat{a}_j z_j + \sum_{k=1}^{p}\sum_{j=1}^{n}\sum_{l=1}^{q} \hat{c}_{jkl} y_{jkl} + \sum_{j=1}^{n}\sum_{i=1}^{m} \left(\sum_{l=1}^{q} \mu_{jl} d_{il} + \hat{r}_{ij} \right) x_{ij} \\
\text{s.t.} \quad & (2.2) - (2.5).
\end{aligned}$$

The second set of subproblems are n single variable concave minimization problems in Q_j

$$\begin{aligned}
[\mathcal{SP}2_j]: \quad \min \quad & \hat{f}_j(Q_j) - \lambda_j Q_j \\
\text{s.t.} \quad & 0 \le Q_j \le \overline{Q}_j.
\end{aligned} \tag{2.10}$$

where $\overline{Q}_j = \sum_{l=1}^{q} u_l s_{jl} \left(\sum_{i=1}^{m} d_{il} \right)$. As the objective function is concave, one of the extreme points is optimal, *i.e.*, $Q_j^* = 0$ or $\overline{Q}_j$.

The third set of subproblems are nq single variable concave minimization problems in P_{jl}

$$\begin{aligned}
[\mathcal{SP}3_{jl}]: \quad \min \quad & t_{jl}(P_{jl}) + \lambda_j u_l s_{jl}(P_{jl}) - \mu_{jl} P_{jl} \\
& 0 \le P_{jl} \le \overline{P}_{jl} \qquad \forall j, l,
\end{aligned}$$

where $\overline{P}_{jl} = \sum_{i=1}^{m} d_{il}$. Again, the optimal solution exists at one of the extreme points, *i.e.*, $P_{jl}^* = 0$ or $\overline{P}_{jl}$. Thus, the best Lagrangian bound resulting from this relaxation is obtained by solving the Lagrangian dual problem:

$$LB^* = \max_{\lambda,\mu} \left\{ \mathbf{v}[\mathcal{SP}1] + \sum_{j=1}^{n} \mathbf{v}[\mathcal{SP}2_j] + \sum_{j=1}^{n}\sum_{l=1}^{q} \mathbf{v}[\mathcal{SP}3_{jl}] \right\}$$

where $\mathbf{v}[SP1]$ is the optimal solution of $[\mathcal{SP}1]$, $\mathbf{v}[\mathcal{SP}2_j] = \min(0, \hat{f}_j(\overline{Q}_j) - \lambda_j \overline{Q}_j)$, $\mathbf{v}[\mathcal{SP}3_{jl}] = \min(0, t_{jl}\left(\overline{P}_{jl}\right) + \lambda_j u_l s_{jl}\left(\overline{P}_{jl}\right) - \mu_{jl}\overline{P}_{jl})$. Furthermore, let H be the index set of the feasible solutions of $\mathcal{SP}1$:

$$H = \left\{ h : \sum_{j=1}^{n} x_{ij}^h = 1; \sum_{k=1}^{p} y_{jkl}^h = \sum_{i=1}^{m} d_{il} x_{ij}^h; x_{ij}^h \leq z_j^h; x_{ij}^h, z_j^h \in \{0,1\}, y_{jkl}^h \geq 0; \forall i,j,k \right\}$$

The Lagrangian dual problem can be reformulated as a linear programming problem by introducing the variables:

$$\begin{aligned}
\theta_1 &= \min_{h \in H} \Big(\sum_{j=1}^{n} \hat{a}_j z_j^h + \sum_{k=1}^{p}\sum_{j=1}^{n}\sum_{l=1}^{q} \hat{c}_{jkl} y_{jkl}^h + \sum_{j=1}^{n}\sum_{i=1}^{m} (\sum_{l=1}^{q} \mu_{jl} d_{il} + \hat{r}_{ij}) x_{ij}^h \Big), \\
\theta_{2j} &= \min \Big(0, \hat{f}_j\left(\overline{Q}_j\right) - \lambda_j \overline{Q}_j \Big), && \forall j, \\
\theta_{3jl} &= \min\left(0, t_{jl}\left(\overline{P}_{jl}\right) + \lambda_j u_l s_{jl}\left(\overline{P}_{jl}\right) - \mu_{jl}\overline{P}_{jl}\right), && \forall j, l.
\end{aligned}$$

Therefore, the dual master problem is

$$[\mathcal{DMP}]\text{: } \max_{\lambda,\mu} \left\{ \theta_1 + \sum_{j=1}^{n} \theta_{2j} + \sum_{j=1}^{n}\sum_{l=1}^{q} \theta_{3jl} \right\}$$

$$\text{s.t. } \theta_1 - \sum_{l=1}^{q}\sum_{j=1}^{n} \mu_{jl} \sum_{i=1}^{m} d_{il} x_{ij}^h \leq \sum_{j=1}^{n} \hat{a}_j z_j^h + \sum_{k=1}^{p}\sum_{j=1}^{n}\sum_{l=1}^{q} \hat{c}_{jkl} y_{jkl}^h + \sum_{j=1}^{n}\sum_{i=1}^{m} \hat{r}_{ij} x_{ij}^h, \quad \forall h \in H, \tag{2.11}$$

$$\begin{aligned}
&\theta_{2j} + \lambda_j \overline{Q}_j \leq \hat{f}_j(\overline{Q}_j), && \forall j, && (2.12) \\
&\theta_{3jl} - \lambda_j u_l s_{jl}(\overline{P}_{jl}) + \mu_{jl}\overline{P}_{jl} \leq t_{jl}(\overline{P}_{jl}), && \forall j, l, && (2.13) \\
&\theta_{2j} \leq 0, && \forall j, && (2.14) \\
&\theta_{3jl} \leq 0, && \forall j, l. && (2.15)
\end{aligned}$$

Its dual, the Dantzig-Wolfe master problem is:

$$
\begin{aligned}
[\mathcal{MP}]:\quad \min \quad & \sum_{h\in H}\left\{\sum_{j=1}^{n}\hat{a}_j z_j^h + \sum_{k=1}^{p}\sum_{j=1}^{n}\sum_{l=1}^{q}\hat{c}_{jkl}y_{jkl}^h + \sum_{j=1}^{n}\sum_{i=1}^{m}\hat{r}_{ij}x_{ij}^h\right\}\alpha_h \\
& + \sum_{j=1}^{n}\hat{f}_j(\overline{Q}_j)\beta_j + \sum_{j=1}^{n}\sum_{l=1}^{q}t_{jl}(\overline{P}_{jl})\gamma_{jl} \\
\text{s.t.}\ & \sum_{h\in H}\alpha_h = 1 \\
& 0 \le \beta_j \le 1 && \forall j, \\
& 0 \le \gamma_{jl} \le 1 && \forall j, l, \\
& \overline{Q}_j\beta_j - \sum_{l=1}^{q}u_l s_{jl}(\sum_{i=1}^{m}d_{il})\gamma_{jl} = 0, && \forall j, \\
& \sum_{i=1}^{m}d_{il}\gamma_{jl} - \sum_{h\in H}\sum_{i=1}^{m}d_{il}x_{ij}^h\alpha_h = 0, && \forall j, l, \\
& \alpha_h \ge 0, && \forall h \in H.
\end{aligned}
$$

Obviously, its impractical to consider all feasible solutions of $\mathcal{SP}1$ in the index set H to generate constraints in the dual master problem $\mathcal{DMP}$, since the number of these constraints is very large. Instead, a classical cutting plane algorithm starts with a subset of feasible solutions to generate a relaxed dual master problem $\mathcal{RDMP}$ that is solved to get a set of Lagrangian multipliers (λ_j and μ_{jl}) and an upper bound. These multipliers are then used to solve the subproblems and get a lower bound. The index set H is updated and a new constraint is added to $\mathcal{RDMP}$, which is then solved again to get new multipliers. This process continues until the upper and lower bounds coincide, in which case the Lagrangian bound is reached. In the problem under consideration, however, we are able to find the best Lagrangian bound directly.

In the next proposition, we provide a closed-form expression for the optimal multipliers in the general case, *i.e.*, ($\underline{Q}_j \le Q_j \le \overline{Q}_j$) and ($\underline{P}_{jl} \le P_{jl} \le \overline{P}_{jl}$), so we can use it to devise a branch-and-bound algorithm in the upcoming section. To get a lower bound on $\mathcal{P}$, we just need to set $\underline{Q}_j$ and $\underline{P}_{jl}$ to 0.

Proposition 2 *The optimal multipliers are given by*

$$
\begin{aligned}
\lambda_j^* &= \frac{\hat{f}_j(\overline{Q}_j) - \hat{f}_j(\underline{Q}_j)}{\overline{Q}_j - \underline{Q}_j}, \quad j = 1, \ldots, n. \\
\mu_{jl}^* &= \frac{t_{jl}(\overline{P}_{jl}) - t_{jl}(\underline{P}_{jl}) + \lambda_j^* u_l(s_{jl}(\overline{P}_{jl}) - s_{jl}(\underline{P}_{jl}))}{\overline{P}_{jl} - \underline{P}_{jl}}, \quad j = 1, \ldots, n,\ l = 1, \ldots, q.
\end{aligned}
$$

Proof: *First, let us solve the subproblems for* λ_j^*, $j = 1, \ldots, n$ *and* μ_{jl}^*, $j = 1, \ldots, n$, $l = 1, \ldots, q$. *Let* $h^* \in H$ *be the index of the optimal solution of* $\mathcal{SP}1$.

The solution of $\mathcal{SP}2_j$ *is:*

$$\begin{aligned}
\mathbf{v}[\mathcal{SP}2_j] &= \min\left(\hat{f}_j(\underline{Q}_j) - \lambda_j^* \underline{Q}_j, \hat{f}_j(\overline{Q}_j) - \lambda_j^* \overline{Q}_j\right) \\
&= \min\left(\frac{\hat{f}_j(\underline{Q}_j)\overline{Q}_j - \hat{f}_j(\overline{Q}_j)\underline{Q}_j}{\overline{Q}_j - \underline{Q}_j}, \frac{\hat{f}_j(\underline{Q}_j)\overline{Q}_j - \hat{f}_j(\overline{Q}_j)\underline{Q}_j}{\overline{Q}_j - \underline{Q}_j}\right) \\
&= \frac{\hat{f}_j(\underline{Q}_j)\overline{Q}_j - \hat{f}_j(\overline{Q}_j)\underline{Q}_j}{\overline{Q}_j - \underline{Q}_j}.
\end{aligned}$$

And the solution of $\mathcal{SP}3_{jl}$ *is:*

$$\begin{aligned}
\mathbf{v}[\mathcal{SP}3_{jl}] &= \min\left(t_{jl}(\underline{P}_{jl}) + \lambda_j^* u_l s_{jl}(\underline{P}_{jl}) - \mu_{jl}^* \overline{P}_{jl}, t_{jl}(\overline{P}_{jl}) + \lambda_j^* u_l s_{jl}(\overline{P}_{jl}) - \mu_{jl}^* \overline{P}_{jl}\right) \\
&= \min\Bigg(\frac{t_{jl}(\underline{P}_{jl})\overline{P}_{jl} - t_{jl}(\overline{P}_{jl})\underline{P}_{jl} + \lambda_j^* u_l (s_{jl}(\underline{P}_{jl})\overline{P}_{jl} - s_{jl}(\overline{P}_{jl})\underline{P}_{jl})}{\overline{P}_{jl} - \underline{P}_{jl}}, \\
&\qquad \frac{t_{jl}(\underline{P}_{jl})\overline{P}_{jl} - t_{jl}(\overline{P}_{jl})\underline{P}_{jl} + \lambda_j^* u_l (s_{jl}(\underline{P}_{jl})\overline{P}_{jl} - s_{jl}(\overline{P}_{jl})\underline{P}_{jl})}{\overline{P}_{jl} - \underline{P}_{jl}}\Bigg) \\
&= \frac{t_{jl}(\underline{P}_{jl})\overline{P}_{jl} - t_{jl}(\overline{P}_{jl})\underline{P}_{jl} + \lambda_j^* u_l (s_{jl}(\underline{P}_{jl})\overline{P}_{jl} - s_{jl}(\overline{P}_{jl})\underline{P}_{jl})}{\overline{P}_{jl} - \underline{P}_{jl}}.
\end{aligned}$$

Thus, the Lagrangian lower bound corresponding to the proposed multipliers is:

$$\begin{aligned}
LB(\lambda^*, \mu^*) &= \sum_{j=1}^n \hat{a}_j z_j^{h^*} + \sum_{k=1}^p \sum_{j=1}^n \sum_{l=1}^q \hat{c}_{jkl} y_{jkl}^{h^*} + \sum_{j=1}^n \sum_{i=1}^m (\sum_{l=1}^q \mu_{jl}^* d_{il} + \hat{r}_{ij}) x_{ij}^{h^*} \\
&+ \sum_{j=1}^n \frac{\hat{f}_j(\underline{Q}_j)\overline{Q}_j - \hat{f}_j(\overline{Q}_j)\underline{Q}_j}{\overline{Q}_j - \underline{Q}_j} \\
&+ \sum_{j=1}^n \sum_{l=1}^q \frac{t_{jl}(\underline{P}_{jl})\overline{P}_{jl} - t_{jl}(\overline{P}_{jl})\underline{P}_{jl} + \lambda_j^* u_l (s_{jl}(\underline{P}_{jl})\overline{P}_{jl} - s_{jl}(\overline{P}_{jl})\underline{P}_{jl})}{\overline{P}_{jl} - \underline{P}_{jl}}
\end{aligned} \tag{2.16}$$

Second, let us consider the relaxed dual master problem $\mathcal{RDMP}$ *corresponding to* $H = \{h^*\}$:

$$\text{[}\mathcal{RDMP}\text{]:} \quad \max_{\lambda,\mu} \left\{ \theta_1 + \sum_{j=1}^{n} \theta_{2j} + \sum_{j=1}^{n}\sum_{l=1}^{q} \theta_{3jl} \right\}$$

$$s.t.\ \theta_1 - \sum_{l=1}^{q}\sum_{j=1}^{n} \mu_{jl} \sum_{i=1}^{m} d_{il} x_{ij}^{h^*} \leq \sum_{j=1}^{n} \hat{a}_j z_j^{h^*} + \sum_{k=1}^{p}\sum_{j=1}^{n}\sum_{l=1}^{q} \hat{c}_{jkl} y_{jkl}^{h^*} + \sum_{j=1}^{n}\sum_{i=1}^{m} \hat{r}_{ij} x_{ij}^{h^*}, \tag{2.17}$$

$$\theta_{2j} + \lambda_j \overline{Q}_j \leq \hat{f}_j(\overline{Q}_j), \qquad \forall j, \tag{2.18}$$

$$\theta_{2j} + \lambda_j \underline{Q}_j \leq \hat{f}_j(\underline{Q}_j), \qquad \forall j, \tag{2.19}$$

$$\theta_{3jl} - \lambda_j u_l s_{jl}(\overline{P}_{jl}) + \mu_{jl} \overline{P}_{jl} \leq t_{jl}(\overline{P}_{jl}), \qquad \forall j, l, \tag{2.20}$$

$$\theta_{3jl} - \lambda_j u_l s_{jl}(\underline{P}_{jl}) + \mu_{jl} \underline{P}_{jl} \leq t_{jl}(\underline{P}_{jl}), \qquad \forall j, l, \tag{2.21}$$

and let us check whether the solution

$$\begin{aligned}
\lambda_j &= \lambda_j^*, \\
\mu_{jl} &= \mu_{jl}^*, \\
\theta_1 &= \sum_{j=1}^{n} \hat{a}_j z_j^{h^*} + \sum_{k=1}^{p}\sum_{j=1}^{n}\sum_{l=1}^{q} \hat{c}_{jkl} y_{jkl}^{h^*} + \sum_{j=1}^{n}\sum_{i=1}^{m} (\sum_{l=1}^{q} \mu_{jl} d_{il} + \hat{r}_{ij}) x_{ij}^{h^*}, \\
\theta_{2j} &= \frac{\hat{f}_j(\underline{Q}_j)\overline{Q}_j - \hat{f}_j(\overline{Q}_j)\underline{Q}_j}{\overline{Q}_j - \underline{Q}_j}, \\
\theta_{3jl} &= \frac{t_{jl}(\underline{P}_{jl})\overline{P}_{jl} - t_{jl}(\overline{P}_{jl})\underline{P}_{jl} + \lambda_j^* u_l (s_{jl}(\underline{P}_{jl})\overline{P}_{jl} - s_{jl}(\overline{P}_{jl})\underline{P}_{jl})}{\overline{P}_{jl} - \underline{P}_{jl}}
\end{aligned}$$

is feasible to $\mathcal{RDMP}$. By substituting in (2.17)-(2.21), all these constraints are found to be satisfied as equalities. The objective function value of $\mathcal{RDMP}$ corresponding to the proposed solution is

$$\begin{aligned}
\mathbf{v}\text{[}\mathcal{RDMP}\text{]} &= \sum_{j=1}^{n} \hat{a}_j z_j^{h^*} + \sum_{k=1}^{p}\sum_{j=1}^{n}\sum_{l=1}^{q} \hat{c}_{jkl} y_{jkl}^{h^*} + \sum_{j=1}^{n}\sum_{i=1}^{m} \left(\sum_{l=1}^{q} \mu_{jl}^* d_{il} + \hat{r}_{ij} \right) x_{ij}^{h^*} \\
&+ \sum_{j=1}^{n} \frac{\hat{f}_j(\underline{Q}_j)\overline{Q}_j - \hat{f}_j(\overline{Q}_j)\underline{Q}_j}{\overline{Q}_j - \underline{Q}_j} \\
&+ \sum_{j=1}^{n}\sum_{l=1}^{q} \frac{t_{jl}(\underline{P}_{jl})\overline{P}_{jl} - t_{jl}(\overline{P}_{jl})\underline{P}_{jl} + \lambda_j^* u_l (s_{jl}(\underline{P}_{jl})\overline{P}_{jl} - s_{jl}(\overline{P}_{jl})\underline{P}_{jl})}{\overline{P}_{jl} - \underline{P}_{jl}}.
\end{aligned}$$

To prove the optimality of the proposed solution, we resort to the dual of $\mathcal{RDMP}$, the restricted master problem

$$
\begin{aligned}
\text{[}\mathcal{RMP}\text{]:}\ \min \quad & \left\{\sum_{j=1}^{n} \hat{a}_j z_j^{h^*} + \sum_{k=1}^{p}\sum_{j=1}^{n}\sum_{l=1}^{q} \hat{c}_{jkl} y_{jkl}^{h^*} + \sum_{j=1}^{n}\sum_{i=1}^{m} \hat{r}_{ij} x_{ij}^{h^*}\right\} \alpha_{h^*} \\
& + \sum_{j=1}^{n} \hat{f}_j(\overline{Q}_j)\beta_{1j} + \sum_{j=1}^{n} \hat{f}_j(\underline{Q}_j)\beta_{2j} + \sum_{j=1}^{n}\sum_{l=1}^{q} t_{jl}(\overline{P}_{jl})\gamma_{1jl} + \quad \sum_{j=1}^{n}\sum_{l=1}^{q} t_{jl}(\underline{P}_{jl})\gamma_{2jl} \\
s.t.\ & \alpha_{h^*} = 1, && (2.22) \\
& \beta_{1j} + \beta_{2j} = 1 \quad \forall j, && (2.23) \\
& \gamma_{1jl} + \gamma_{2jl} = 1 \quad \forall j, l, && (2.24) \\
& \overline{Q}_j\beta_{1j} + \underline{Q}_j\beta_{2j} - \sum_{l=1}^{q} u_l s_{jl}(\overline{P}_{jl})\gamma_{1jl} - \sum_{l=1}^{q} u_l s_{jl}(\underline{P}_{jl})\gamma_{2jl} = 0, \forall j, && (2.25) \\
& \overline{P}_{jl}\gamma_{1jl} + \underline{P}_{jl}\gamma_{1jl} - \sum_{i=1}^{m} d_{il} x_{ij}^{h^*} \alpha_{h^*} = 0, \quad \forall j, l. && (2.26)
\end{aligned}
$$

By substituting the value of α_h^* from (2.22) in (2.26) and solving (2.24) and (2.26) simultaneously, we get:

$$
\begin{aligned}
\gamma_{1jl} &= \frac{\sum_{i=1}^{m} d_{il} x_{ij}^{h^*} - \underline{P}_{jl}}{\overline{P}_{jl} - \underline{P}_{jl}}, \\
\gamma_{2jl} &= \frac{\overline{P}_{jl} - \sum_{i=1}^{m} d_{il} x_{ij}^{h^*}}{\overline{P}_{jl} - \underline{P}_{jl}}.
\end{aligned}
$$

And by solving (2.23) and (2.25) simultaneously, we get:

$$
\begin{aligned}
\beta_{1j} &= \frac{\sum_{l=1}^{q} u_l s_{jl}(\overline{P}_{jl})\gamma_{1jl} + \sum_{l=1}^{q} u_l s_{jl}(\underline{P}_{jl})\gamma_{2jl} - \underline{Q}_j}{\overline{Q}_j - \underline{Q}_j}, \\
\beta_{2j} &= \frac{\overline{Q}_j - \sum_{l=1}^{q} u_l s_{jl}(\overline{P}_{jl})\gamma_{1jl} - \sum_{l=1}^{q} u_l s_{jl}(\underline{P}_{jl})\gamma_{2jl}}{\overline{Q}_j - \underline{Q}_j}.
\end{aligned}
$$

Thus,

$$
\begin{aligned}
\mathbf{v}[\mathcal{RMP}] &= \sum_{j=1}^{n} \hat{a}_j z_j^{h^*} + \sum_{k=1}^{p}\sum_{j=1}^{n}\sum_{l=1}^{q} \hat{c}_{jkl} y_{jkl}^{h^*} + \sum_{j=1}^{n}\sum_{i=1}^{m} \hat{r}_{ij} x_{ij}^{h^*} \\
&+ \sum_{j=1}^{n} \hat{f}_j(\overline{Q}_j) \frac{\sum_{l=1}^{q} u_l s_{jl}(\overline{P}_{jl})\gamma_{1jl} + \sum_{l=1}^{q} u_l s_{jl}(\underline{P}_{jl})\gamma_{2jl} - \underline{Q}_j}{\overline{Q}_j - \underline{Q}_j} \\
&+ \sum_{j=1}^{n} \hat{f}_j(\underline{Q}_j) \frac{\overline{Q}_j - \sum_{l=1}^{q} u_l s_{jl}(\overline{P}_{jl})\gamma_{1jl} - \sum_{l=1}^{q} u_l s_{jl}(\underline{P}_{jl})\gamma_{2jl}}{\overline{Q}_j - \underline{Q}_j} \\
&+ \sum_{j=1}^{n}\sum_{l=1}^{q} t_{jl}(\overline{P}_{jl}) \frac{\sum_{i=1}^{m} d_{il} x_{ij}^{h^*} - \underline{P}_{jl}}{\overline{P}_{jl} - \underline{P}_{jl}} + \sum_{j=1}^{n}\sum_{l=1}^{q} t_{jl}(\underline{P}_{jl}) \frac{\overline{P}_{jl} - \sum_{i=1}^{m} d_{il} x_{ij}^{h^*}}{\overline{P}_{jl} - \underline{P}_{jl}} \\
&= \sum_{j=1}^{n} \hat{a}_j z_j^{h^*} + \sum_{k=1}^{p}\sum_{j=1}^{n}\sum_{l=1}^{q} \hat{c}_{jkl} y_{jkl}^{h^*} + \sum_{j=1}^{n}\sum_{i=1}^{m} \hat{r}_{ij} x_{ij}^{h^*} \\
&+ \sum_{j=1}^{n} \frac{\hat{f}_j(\underline{Q}_j)\overline{Q}_j - \hat{f}_j(\overline{Q}_j)\underline{Q}_j}{\overline{Q}_j - \underline{Q}_j} \\
&+ \sum_{j=1}^{n} \frac{\hat{f}_j(\overline{Q}_j) - \hat{f}_j(\underline{Q}_j)}{\overline{Q}_j - \underline{Q}_j} \left(\sum_{l=1}^{q} u_l s_{jl}(\overline{P}_{jl})\gamma_{1jl} + \sum_{l=1}^{q} u_l s_{jl}(\underline{P}_{jl})\gamma_{2jl} \right) \\
&+ \sum_{j=1}^{n}\sum_{l=1}^{q} \frac{t_{jl}(\underline{P}_{jl})\overline{P}_{jl} - t_{jl}(\overline{P}_{jl})\underline{P}_{jl}}{\overline{P}_{jl} - \underline{P}_{jl}} + \sum_{j=1}^{n}\sum_{l=1}^{q} \frac{(t_{jl}(\overline{P}_{jl}) - t_{jl}(\underline{P}_{jl})) \sum_{i=1}^{m} d_{il} x_{ij}^{h^*}}{\overline{P}_{jl} - \underline{P}_{jl}} \\
&= \sum_{j=1}^{n} \hat{a}_j z_j^{h^*} + \sum_{k=1}^{p}\sum_{j=1}^{n}\sum_{l=1}^{q} \hat{c}_{jkl} y_{jkl}^{h^*} + \sum_{j=1}^{n}\sum_{i=1}^{m} \hat{r}_{ij} x_{ij}^{h^*} \\
&+ \sum_{j=1}^{n} \frac{\hat{f}_j(\underline{Q}_j)\overline{Q}_j - \hat{f}_j(\overline{Q}_j)\underline{Q}_j}{\overline{Q}_j - \underline{Q}_j} \\
&+ \sum_{j=1}^{n}\sum_{l=1}^{q} \frac{\lambda_j^* u_l (s_{jl}(\underline{P}_{jl})\overline{P}_{jl} - s_{jl}(\overline{P}_{jl})\underline{P}_{jl})}{\overline{P}_{jl} - \underline{P}_{jl}} + \sum_{j=1}^{n}\sum_{l=1}^{q} \frac{\lambda_j^* u_l \sum_{i=1}^{m} d_{il} x_{ij}^{h^*} (s_{jl}(\overline{P}_{jl}) - s_{jl}(\underline{P}_{jl}))}{\overline{P}_{jl} - \underline{P}_{jl}} \\
&+ \sum_{j=1}^{n}\sum_{l=1}^{q} \frac{t_{jl}(\underline{P}_{jl})\overline{P}_{jl} - t_{jl}(\overline{P}_{jl})\underline{P}_{jl}}{\overline{P}_{jl} - \underline{P}_{jl}} + \sum_{j=1}^{n}\sum_{l=1}^{q} \frac{(t_{jl}(\overline{P}_{jl}) - t_{jl}(\underline{P}_{jl})) \sum_{i=1}^{m} d_{il} x_{ij}^{h^*}}{\overline{P}_{jl} - \underline{P}_{jl}} \\
&= \sum_{j=1}^{n} \hat{a}_j z_j^{h^*} + \sum_{k=1}^{p}\sum_{j=1}^{n}\sum_{l=1}^{q} \hat{c}_{jkl} y_{jkl}^{h^*} + \sum_{j=1}^{n}\sum_{i=1}^{m} (\sum_{l=1}^{q} \mu_{jl}^* d_{il} + \hat{r}_{ij}) x_{ij}^{h^*} \\
&+ \sum_{j=1}^{n} \frac{\hat{f}_j(\underline{Q}_j)\overline{Q}_j - \hat{f}_j(\overline{Q}_j)\underline{Q}_j}{\overline{Q}_j - \underline{Q}_j} \\
&+ \sum_{j=1}^{n}\sum_{l=1}^{q} \frac{t_{jl}(\underline{P}_{jl})\overline{P}_{jl} - t_{jl}(\overline{P}_{jl})\underline{P}_{jl} + \lambda_j^* u_l (s_{jl}(\underline{P}_{jl})\overline{P}_{jl} - s_{jl}(\overline{P}_{jl})\underline{P}_{jl})}{\overline{P}_{jl} - \underline{P}_{jl}}.
\end{aligned}
$$

By strong duality, since $\mathbf{v}[\mathcal{RDMP}] = \mathbf{v}[\mathcal{RMP}]$*, the proposed solution is optimal to* $\mathcal{RDMP}$*. Furthermore,* $LB(\lambda^*, \mu^*) \leq \mathbf{v}[\mathcal{DMP}]$*. According to (2.16),* $LB(\lambda^*, \mu^*) = \mathbf{v}[\mathcal{RDMP}]$*, therefore,* $\mathbf{v}[\mathcal{RDMP}] = \mathbf{v}[\mathcal{DMP}] = LB(\lambda^*, \mu^*)$*, the Lagrangian bound. Thus,* λ^* *and* μ^* *are optimal.* ■

Next, we note that $(x^{h^*}, y^{h^*}, z^{h^*})$, the optimal solution of $\mathcal{SP}1_{(\lambda^*,\mu^*)}$, is also feasible to $\mathcal{P}$ as both problems have the same feasible set. Its corresponding objective

$$\sum_{j=1}^{n} \hat{a}_j z_j^{h^*} + \sum_{k=1}^{p}\sum_{j=1}^{n}\sum_{l=1}^{q} \hat{c}_{jkl} y_{jkl}^{h^*} + \sum_{j=1}^{n}\sum_{i=1}^{m} \hat{r}_{ij} x_{ij}^{h^*} + \sum_{j=1}^{n}\sum_{l=1}^{q} t_{jl}(P_{jl}^*) + \sum_{j=1}^{n} \hat{f}_j(Q_j^*),$$

where $P_{jl}^* = \sum_{i=1}^m d_{il} x_{ij}^{h^*}$ and $Q_j^* = \sum_{l=1}^q u_l s_{jl}\left(\sum_{i=1}^m d_{il} x_{ij}^{h^*}\right)$, is an upper bound. The difference between this upper bound and the Lagrangian bound is given by

$$\begin{aligned}
\delta \;&=\; \sum_{j=1}^{n}\sum_{l=1}^{q} t_{jl}(P_{jl}^*) + \sum_{j=1}^{n} \hat{f}_j(Q_j^*) - \sum_{j=1}^{n} \frac{\hat{f}_j(\underline{Q}_j)\overline{Q}_j - \hat{f}_j(\overline{Q}_j)\underline{Q}_j}{\overline{Q}_j - \underline{Q}_j} \\
&- \sum_{j=1}^{n}\sum_{l=1}^{q} \frac{\lambda_j^* u_l \left[s_{jl}(\underline{P}_{jl})(\overline{P}_{jl} - P_{jl}^*) + s_{jl}(\overline{P}_{jl})(P_{jl}^* - \underline{P}_{jl})\right])}{\overline{P}_{jl} - \underline{P}_{jl}} \\
&- \sum_{j=1}^{n}\sum_{l=1}^{q} \frac{t_{jl}(\underline{P}_{jl})(\overline{P}_{jl} - P_{jl}^*) + t_{jl}(\overline{P}_{jl})(P_{jl}^* - \underline{P}_{jl})}{\overline{P}_{jl} - \underline{P}_{jl}}
\end{aligned}$$

which provides an upper limit on the optimality gap.

Corollary 3 *If* $Q_j^* = \overline{Q}_j$ *or* $\underline{Q}_j$*,* $(x^{h^*},\, y^{h^*},\, z^{h^*})$ *is optimal to* $\mathcal{P}$*.*

Proof: *Let us first note that the gap can be decomposed by* j *such that* $\delta = \sum_{j=1}^n \delta_j$*, and*

$$\begin{aligned}
\delta_j \;&=\; \sum_{l=1}^{q} t_{jl}(P_{jl}^*) + \hat{f}_j(Q_j^*) - \frac{\hat{f}_j(\underline{Q}_j)\overline{Q}_j - \hat{f}_j(\overline{Q}_j)\underline{Q}_j}{\overline{Q}_j - \underline{Q}_j} \\
&- \sum_{l=1}^{q} \frac{\lambda_j^* u_l \left[s_{jl}(\underline{P}_{jl})(\overline{P}_{jl} - P_{jl}^*) + s_{jl}(\overline{P}_{jl})(P_{jl}^* - \underline{P}_{jl})\right])}{\overline{P}_{jl} - \underline{P}_{jl}} \\
&- \sum_{l=1}^{q} \frac{t_{jl}(\underline{P}_{jl})(\overline{P}_{jl} - P_{jl}^*) + t_{jl}(\overline{P}_{jl})(P_{jl}^* - \underline{P}_{jl})}{\overline{P}_{jl} - \underline{P}_{jl}}. \qquad (2.27)
\end{aligned}$$

Also note that since $\overline{Q}_j = \sum_{l=1}^{q} u_l s_{jl}(\overline{P}_{jl})$, $Q_j^* = \overline{Q}_j$ *if and only if* $P_{jl}^* = \overline{P}_{jl}$, $\forall l$*. By substituting in (2.27), we get*

$$\begin{aligned}
\delta_j &= \sum_{l=1}^{q} t_{jl}(\overline{P}_{jl}) + \hat{f}_j(\overline{Q}_j) - \frac{\hat{f}_j(\underline{Q}_j)\overline{Q}_j - \hat{f}_j(\overline{Q}_j)\underline{Q}_j}{\overline{Q}_j - \underline{Q}_j} \\
&- \lambda_j^* \sum_{l=1}^{q} s_{jl}(\overline{P}_{jl}) - \sum_{l=1}^{q} t_{jl}(\overline{P}_{jl}) \\
&= \frac{\overline{Q}_j \left[\hat{f}_j(\overline{Q}_j) - \hat{f}_j(\underline{Q}_j) \right]}{\overline{Q}_j - \underline{Q}_j} - \lambda_j^* \overline{Q}_j = 0.
\end{aligned}$$

Likewise, since $\underline{Q}_j = \sum_{l=1}^{q} u_l s_{jl}(\underline{P}_{jl})$, $Q_j^* = \underline{Q}_j$ *if and only if* $P_{jl}^* = \underline{P}_{jl}$, $\forall l$*. By substituting in (2.27), we get*

$$\begin{aligned}
\delta_j &= \sum_{l=1}^{q} t_{jl}(\underline{P}_{jl}) + \hat{f}_j(\underline{Q}_j) - \frac{\hat{f}_j(\underline{Q}_j)\overline{Q}_j - \hat{f}_j(\overline{Q}_j)\underline{Q}_j}{\overline{Q}_j - \underline{Q}_j} \\
&- \lambda_j^* \sum_{l=1}^{q} s_{jl}(\underline{P}_{jl}) - \sum_{l=1}^{q} t_{jl}(\underline{P}_{jl}) \\
&= \frac{\underline{Q}_j \left[\hat{f}_j(\overline{Q}_j) - \hat{f}_j(\underline{Q}_j) \right]}{\overline{Q}_j - \underline{Q}_j} - \lambda_j^* \underline{Q}_j = 0.
\end{aligned}$$

Thus, $\delta = 0$ *and optimality is proven.* ■

Based on this corollary , if the optimal solution of $\mathcal{SP}1$ is to open a single warehouse, the optimality gap is guaranteed to close in the root node (*i.e.*, without branching). More importantly, this corollary helps us to devise an efficient branch-and-bound algorithm as described in the next section.

2.6.2 Branch-and-bound

To close the optimality gap, we embed the Lagrangian decomposition in a branch-and-bound algorithm similar to the one explained in section 1.2.2. For completeness, we present it here again. Branching is performed around the optimal solution of $\mathcal{SP}1$ using the auxiliary variables Q_j, $j = 1, \dots, n$. At any node, upon solving $\mathcal{SP}1$ we get the vector x^* and calculate Q^*_j. The variables Q^*_i $,i = 1, \dots, n$ are classified to two sets: *extreme* when $Q^*_j = \underline{Q}_j$ or $\overline{Q}_j$, and *non-extreme* otherwise. If all the variables are extreme, the gap is guaranteed to close according to corollary 3, otherwise, a non extreme variable $Q_{\hat{j}}$ is selected for branching. In one child node, we set $\underline{Q_{\hat{j}}} = Q^*_{\hat{j}}$ and $\underline{P_{\hat{j}l}} = P^*_{\hat{j}l}$, $\forall l$, and add the constraints $\sum_{i=1}^m d_{il}x_{i\hat{j}} \geq P^*_{\hat{j}l}$, $\forall l$ to $\mathcal{SP}1$; And in the other child node, we set $\overline{Q_{\hat{j}}} = Q^*_{\hat{j}}$ and $\overline{P_{\hat{j}l}} = P^*_{\hat{j}l}$, $\forall l$, and add the constraints $\sum_{i=1}^m d_{il}x_{i\hat{j}} \leq P^*_{\hat{j}l}$, $\forall l$ to $\mathcal{SP}1$. In each node we obtain a lower bound using the Lagrangian approach. Furthermore, an upper bound is obtained from the feasible solution of $\mathcal{SP}1$ and the incumbent is updated if a better upper bound is found. Let M_s, $s \in S$ be the partitioning subsets of the feasible region. For each subset (*i.e.*, node) we can get a lower and an upper bound using the Lagrangian approach, denoted $\mathcal{L}(M_s)$ and $\mathcal{U}(M_s)$, respectively. Then, $\mathcal{L} = \min_{s \in S} \mathcal{L}(M_s)$, $\mathcal{U} = \min_{s \in S} \mathcal{U}(M_s)$ are the overall bounds, respectively. If at any iteration $\mathcal{L} = \mathcal{U}$, the branch-and-bound algorithm terminates.

2.6.3 Concavity of the inventory functions

The proposed approach is valid only when the inventory functions are concave. Concavity of the total inventory cost function (t_{jl}) is assured for inventory policies with explicit formulas such as the EOQ. Other than these special cases, little has been reported in the literature about the nature of the cost function. This issue, however, is closely related to the topic of warehouse consolidation which was studied extensively. The effect of increasing the demand served by a certain warehouse is equivalent to consolidating it with an identical warehouse that serves the additional demand. The cost effect of warehouse consolidation was first studied by Eppen [45], who concluded that cost saving is achieved when demand from different sources is managed in a centralized manner. In particular, he showed that when demand is independent and normally distributed, inventory cost can be represented as a square root function in a multi-echelon newsvendor problem. Teo et al. [96] studied the (r, Q) systems and noted that consolidation leads to lower total inventory cost if the demands are *i.i.d.*, or when they follow independent but possibly nonidentical Poisson processes. Although they showed also that, for a general demand distribution, consolidation can lead to higher inventory cost, this happens only under special conditions including very different ordering cost and lead times across warehouses. Clearly, when considering the cost function of a single warehouse, this case is irrelevant. Lim et al. [74] reached the same conclusion for multi-echelon systems and noted that “consolidated systems are rarely suboptimal”. Apart from analytical models, the functional form of the total inventory cost for the complex (s, S) inventory policy was approximated using regression methods for a wide range of system parameters and demand distributions and was shown to be concave in the demand served [42]. Numerical results presented in other studies have shown that the inventory cost is concave, although that was not explicitly reported.

When it comes to the maximum inventory level function (s_{jl}), it is even more difficult to ascertain its concavity with respect to the demand in the general case. Since most inventory policies are simplifications of the (s, S) policy, we focus our attention on this policy. Ehrhardt [41] and Schneider et al. [86] have used regression analysis to derive power approximations of the (s, S) policy under a wide range of conditions. The regression models developed suggest that both s and $(S - s)$ are concave functions of the demand rate, implying that the maximum inventory level S is also a concave function.

While the results discussed above do not lead to a definite conclusion that inventory functions are concave in the general case, they provide a strong indication. Note that, since the maximum inventory cost function $s_{jl}(P_{jl})$ is embedded within the concave function $\hat{f}_j(Q_j)$, the nested function may still be concave in P_{jl} even if s_{jl} are slightly convex. To further validate the concavity of the inventory functions, we estimated their values using simulation and observed the results.

2.6.4 Special cases with known inventory functions

In the proposed Lagrangian/B&B approach, we need to determine the value of inventory functions s_{jl} and t_{jl} at the extreme points encountered. In special cases, these functions are explicitly given by closed-form expressions, thus their values can be easily determined.

Let us, for example, consider the case when the demand of retailers is deterministic and uniform, meaning that, for each retailer, both order size and time between orders are constant. In this case, the classical EOQ model is suitable for managing the inventory at the warehouses. While the model assumes continuous demand, it provides closed form expressions that tightly approximate the maximum inventory level and the total inventory cost functions when the demand is discrete. Let h, b and K be the holding cost ($\$/unit/day$), the shortage penalty ($\$/unit/day$) and the fixed ordering cost ($\$/order$), respectively; Thus:

$$Q^*_{jl} = \sqrt{\frac{2KP_{jl}}{h}}.\sqrt{\frac{b+h}{b}} \tag{2.28}$$

$$B^*_{jl} = \frac{h}{b+h}.Q^*_{jl} \tag{2.29}$$

$$s_{jl}(P_{jl}) = Q^*_{jl} - B^*_{jl} \tag{2.30}$$

$$t_{jl}(P_{jl}) = \frac{KP_{jl}}{Q^*_{jl}} + \frac{hQ^*_{jl}}{2} - h.B^*_{jl} + \frac{{B^*_{jl}}^2}{2Q^*_{jl}}(b+h) \tag{2.31}$$

Since $\frac{h}{b+h} < 1$, s_{jl} is concave in the demand served. Furthermore, it is easy to show that t_{jl} is also concave when $h, b > 0$, which is always the case. Therefore this problem can be solved directly using the proposed Lagrangian approach.

Likewise, for some special cases of stochastic inventory models, closed form expressions are available for s_{jl} and t_{jl}. For example, when demand follows a Poisson distribution and inventory at warehouses is managed using a base-stock policy with continuous review, zero ordering cost and fixed lead time (L), these functions can be approximated as:

$$s_{jl}(P_{jl}) = R^*_{jl} = P_{jl}L + z_{b/(b+h)}\sqrt{P_{jl}L}$$
$$t_{jl}(P_{jl}) = (b+h)\sqrt{P_{jl}L}\phi(z_{b/(b+h)})$$

where $z_{b/(b+h)}$ is the parameter corresponding to area of $b/(b+h)$ under the standard normal distribution curve. Obviously, these assumptions are quite restrictive so we need other approaches to evaluate the maximum inventory level and the inventory cost functions in the general case. In the next subsection, we implement a simulation-optimization approach for this purpose.

2.6.5 Simulation-optimization algorithm

In most realistic situations there are no closed-form expressions for the inventory functions s_{jl} and t_{jl}, but they can be estimated at specific points using discrete-event simulation. A reasonable assumption is that each warehouse selects the control parameters of the inventory policy (*i.e.*, reorder point, ordering frequency, order quantity, etc.) to achieve the minimum inventory cost. Thus, the simulation runs are embedded in a heuristic optimization routine to find the optimal control parameters. We refer to this process as the 'simulation-optimization' algorithm. In what follows, we address the case when the inventory at warehouses is managed using the (r, nQ) inventory policy. It is important to notice, though, that this is just a representative example for the application of simulation-optimization algorithms within the proposed framework, and that other inventory policies can be handled similarly.

The (r, nQ) policy is suitable when products are ordered and shipped in batches (*e.g.*, pallets or full-truck-loads). This is a periodic review policy in which a reorder level r is set such that an order of size nQ is placed whenever the inventory position falls to or below r. Q is a pre-specified quantity that represents the batch size, whereas n is the smallest integer that restores the inventory position above r, *i.e.*, between $r + 1$ and $r + Q$. Our interest in the (r, nQ) policy is stimulated by both theoretical and practical reasons. On one hand, it has been shown that the (r, nQ) policy gives results that are nearly as good as those obtained from the (s, S) policy, which is proven to be optimal under modest assumptions [106]. On the other hand, this policy seems suitable for the problem under consideration in which shipping between plants and warehouses is carried out in bulk quantities using long-haul trucks, meaning that the batch size can, intuitively, be set equal to the full capacity of a truck. In fact, when the transportation cost is taken into account, the (r, nQ) policy might lead to better results than the (s, S) policy due to the higher utilization of trucks.

It is known that the total inventory cost function is convex in the reorder level [108]. Therefore, we propose a bisection search algorithm to find the optimal reorder level. We start with two values of the reorder level r_1 and r_2 such that $r_1 + 1 < r^*$ and $r_2 > r^*$. We set $r_3 = \lfloor \frac{r_1+r_2}{2} \rfloor$ and estimate the total inventory cost at r_3 and $r_3 + 1$ using simulation. If $t(r_3) > t(r_3 + 1)$, r_3 becomes r_1 for the next iteration, otherwise, it becomes r_2. The updated values of r_1 and r_2 are used to obtain a new midpoint r_3, and the process continues until $r_2 = r_1 + 1$, at which point the algorithm terminates and reports $r^* = r_2$.

```
Bisection search algorithm
  Select integers $r_1 < r_2$ such that $t(r_1 + 1) < t(r_1)$ and $t(r_2 + 1) > t(r_2)$
  while $r_2 - r_1 > 1$ do
    $r_3 = \left\lfloor \frac{r_1+r_2}{2} \right\rfloor$
    if $t(r_3 + 1) < t(r_3)$ then
      $r_3 \to r_1$
    else
      $r_3 \to r_2$
    end if
  end while
  $r_2 \to r^*$
```

Obviously, the algorithm is guaranteed to reach the global optimal solution when the cost function is unimodal convex, which is true for the actual cost function. However, since the inventory cost is estimated based on a finite sample size (*i.e.*, a number of simulated days), the resulting cost function is just an estimator of the population (*i.e.*, infinite horizon) cost function and sampling error is inevitable. If this sampling error is large, the estimated function may become multimodal and the algorithm may get stuck in a local optimum. One way to overcome this issue is to increase the sample size to reduce sampling error. Also, common random numbers (CRN) can be used when searching for the optimal reorder level. This technique is shown to reduce the variance of the estimate [57], thereby minimizing the likelihood of premature termination of the algorithm. Both ways are used to improve the bisection search algorithm

Figure 2.2 presents a flowchart that illustrates the solution method including the Lagrangian decomposition, the branch-and-bound and the simulation-optimization stages.

2.7 Numerical Testing

We test the proposed cold supply chain design approach on three data sets. The first two are realistic cases from different industries, with the aim of studying the trade-offs between the economic and the environmental effects in cold supply chains. The first is drawn from the bulk-volume, low-margin processed meat industry, and the second deals with a supply chain network for medical vaccines, a niche product characterized by its low-volume and high-value. We also test the computational performance of the algorithm on a third set of hypothetical instances of different sizes and parameters. Testing is performed on a workstation with *Intel Core-i7* processor of the 4th generation and 8GB of RAM. The approach, including the simulation-optimization algorithm, is coded in *Matlab2013b* and the mixed integer subproblem is solved using *CPLEX12.5*. Simulation runs are conducted in parallel using the parallel computing toolbox in *Matlab* with a maximum of 4 simultaneous threads.

2.7.1 Case I: Maple Leaf Foods

Maple Leaf Foods (MLF) is the largest producer of prepared meats in Canada with revenues of over $3 billion in 2014. The company has recently embarked on an ambitious plan to restructure its supply chain that involves the consolidation of plants and distribution centers (DCs), aiming to achieve significant savings in operational cost and environmental impact. MLF has a strong commitment towards sustainability as demonstrated by its environmental sustainability program that has GHG emissions reduction as one of its pillars [1].

The proposed approach is used to design MLF's cold supply chain in the Canadian province of Ontario. In particular, we consider the demand originating from the province's 20 largest cities. MLF has two plants in Ontario: one in Brampton that produces sausage (P_1) and meat snacks (P_2), and another in Hamilton that specializes in wieners (P_3) and deli products (P_4). In the absence of real data, the demand is estimated as the product of the population of each city, the average consumption per capita of each product, and the company market share, and was verified using the revenue data of the company obtained from its annual report. This case data is collected from a variety of public sources, including industry reports and governmental statistics. The daily demand of each product is assumed to have a Poisson distribution with the rates shown in Table 2.1. Every demand node is also a candidate location for a DC.

Products are shipped from plants to DCs in $40'$ (feet) refrigerated (reefer) containers loaded on conventional (diesel operated) long-haul trucks. For shipping products from DCs to retailers, both $40'$ and $20'$ reefer containers are used, depending on the quantity shipped. The $20'$ containers can be used with one of two types of smaller trucks: conventional or electric. The latter are used for shipment distances of 100 km or less. The cost and GHG emissions per km for a long-haul truck with a fully loaded $40'$ reefer container are \$2 and 6.3 kg, respectively. The number of units of each product for a $40'$ container, alongside the holding and backordering costs per day, are shown in Table 2.2. The annual cost and global warming impact of shipping products between DCs and retailers is calculated based on the distance, the frequency of shipments, and the type of trucks used. Table 2.3 shows the annual cost (the upper row, in thousands of dollars) and the annual CO2-equivalent emissions (the lower row, in metric tonnes) for each DC-retailer pair.

Annual capacity cost of each DC has two components: a fixed cost of \$55,000 and a volume-dependent cost that follows the function $f_j(Q_j) = 1.281Q_j^{0.68}$ in thousand dollars. Each DC has base annual GHG emissions of 112 tonnes of CO2-equivalent emissions and volume-dependent GHG emissions that follow the function $g_j(Q_j) = 11.1Q_j^{0.76}$ tonnes of CO2-equivalent emissions. To reach these functions, the cost and emissions of real cold stores of different sizes are calculated using a bottom-up approach, starting with the cooling load requirements to estimate the equipment size, cost, and energy consumption. Then, assuming that the size-dependent cost and emissions follow functions of the form $\beta_1 Q_j^{\beta_2}$, regression techniques are implemented to estimate the coefficients β_1 and β_2. This regression model is found to provide a good approximation of the relationship between both the DC variable cost and emissions and its size. Each warehouse manages its inventory independently using the (r, nQ) inventory policy, where Q is the capacity of a $40'$ reefer container.

Table 2.1: Average daily demand in the MLF case

i	City	P_1	P_2	P_3	P_4
1	Toronto	686	262	706	961
2	Ottawa	232	88	238	325
3	Mississauga	187	71	193	262
4	Brampton	138	52	141	193
5	Hamilton	137	52	140	191
6	London	96	37	99	135
7	Markham	79	30	82	111
8	Vaughan	76	29	78	106
9	Kitchener	58	22	59	81
10	Windsor	55	21	57	78
11	Burlington	46	18	48	65
12	Sudbery	42	16	43	59
13	Oshawa	39	15	41	55
14	Barrie	36	14	37	50
15	St. Catharines	34	13	35	48
16	Cambridge	33	13	34	47
17	Kingston	32	12	33	45
18	Guelph	32	12	33	45
19	Thunder Bay	28	11	29	40
20	Waterloo	26	10	27	37

Table 2.2: Product characteristics in the MLF case

Product	P_1	P_2	P_3	P_4
Number of products per truckload	1240	1240	1116	1340
Unit holding cost ($/day)	0.046	0.034	0.034	0.040
Unit backordering cost ($/day)	1	1	1	1

Whenever an order is placed, a fixed cost of \$100 is incurred and the order arrives after 2 days. In our simulation model, the sequence at which events occur in DCs is as follows: first, ordered products arrive from the plants and added to the inventory-on-hand, then demand from retailers is realized. If the cumulative demand is less than or equal to the inventory-on-hand, it is satisfied immediately, otherwise, the shortage is carried forward to the next day. At the end of the day, the inventory position is evaluated and, if needed, an order is placed to replenish the inventory.

We test for various values of w, the weight assigned to the GHG emissions minimization objective, in order to investigate the cost-emissions trade-off. w is changed between 0 (minimizing the cost only) and 6. The results are depicted in Table 2.4. The 'Solution' column shows the optimal solution obtained: the opened DCs outside the parenthesis; and for each opened DC, inside the parenthesis, its cold storage area in cubic meters, followed (after the semicolon) by the retailers assigned to it. Each DC is found to serve the retailer in its location. The columns 'TC' and 'TE' show the total cost in thousand dollars, and the total GHG emissions in metric tonnes of CO2-equivalent.

It is clear from these results that as w is increased, fewer DCs are opened. This observation is explained by two factors: First, it seems that emissions have stronger economies of scale than cost due to the high fixed emissions for DCs, which implies higher demand consolidation. Second, once emissions are penalized, there is a stronger incentive to assign close-by retailers to opened DCs in order to utilize electric trucks for shipping. The second factor explains the alternative optimal solution obtained with the same cost but with lower emissions when w is increased from 0 to 0.1. We also notice that the gross storage area of the DCs decreases as fewer DCs are opened due to the well-known risk pooling effect. The primary trade-off in this case is between capacity and transportation cost, whereas the inventory cost is comparatively insignificant because of the low-cost nature of the products.

The trade-off between cost and emissions is illustrated in Fig. 2.3, which shows the Pareto frontier curve. We notice that when one objective is optimized, the other does not deteriorate sharply. For example, when minimizing only the cost by setting w to 0, the emissions level is merely 14.2% higher than when w is set to 6, at which case the emissions minimization objective is dominant. Likewise, the cost increases only by 10.7% when the focus is shifted from minimizing cost to minimizing emissions. It is even possible to decrease the quantity of emissions with no increase in cost by assigning a small weight, *i.e.*, $w = 0.1$, to emissions in the objective function. This weak trade-off between cost and emissions agrees with the result reported in [21].

Figure 2.4 shows a breakdown of the different components in the combined objective function. Note that as w increases, the inventory component decreases dramatically be-

Table 2.3: Cost and GHG emissions for shipping products between DCs and retailers in the MLF case

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	-	659	43	53	92	184	35	46	96	314	53	243	41	57	45	47	89	38	331	20
	-	2070	133	99	169	453	7	9	231	759	10	587	7	10	108	8	216	7	800	49
2	1669	-	620	534	579	610	350	395	435	679	338	305	200	189	211	187	58	168	357	94
	5021	-	1938	1535	1656	1502	847	956	1052	1642	818	738	484	456	511	453	141	407	863	226
3	116	630	-	26	66	165	58	46	86	297	35	243	55	57	45	33	96	30	326	20
	289	1978	-	50	121	405	11	9	15	718	6	587	10	10	8	6	233	5	789	4
4	155	644	29	-	79	165	58	35	86	289	44	243	62	51	45	33	103	21	331	18
	386	2024	89	-	145	405	11	7	15	698	8	587	11	9	108	6	249	4	800	3
5	271	703	72	79	-	126	116	93	75	263	18	274	72	69	30	24	113	21	343	16
	675	2208	222	149	-	310	23	17	13	636	3	663	175	167	5	4	274	4	829	3
6	705	923	245	206	157	-	198	179	96	161	101	330	134	120	73	47	151	41	326	20
	2120	2898	766	593	449	-	481	433	231	390	244	798	323	289	177	8	366	100	789	49
7	116	542	72	66	131	203	-	23	104	322	80	237	34	51	57	42	86	42	328	22
	289	1702	222	124	241	501	-	4	252	780	14	572	6	9	137	102	208	8	794	53
8	155	615	58	40	105	184	23	-	107	314	62	224	41	40	53	47	93	38	324	20
	386	1932	177	75	193	453	5	-	19	759	11	542	7	7	128	8	225	7	783	49
9	408	732	116	106	92	107	113	116	-	246	62	287	83	69	49	9	120	13	348	2
	1227	2300	355	199	169	262	275	22	-	595	11	693	202	167	118	2	291	2	840	-
10	1372	1172	505	413	374	184	359	348	252	-	230	448	234	198	146	107	213	103	302	53
	4128	3680	1577	1186	1070	453	870	842	610	-	557	1,085	565	478	353	259	516	249	731	129
11	232	688	58	66	26	136	105	81	75	272	-	268	67	64	30	24	110	25	340	18
	579	2162	177	124	48	334	20	15	13	657	-	648	161	156	5	4	266	5	823	3
12	1446	718	562	474	531	513	359	339	400	611	310	-	211	133	195	172	199	151	240	83
	4352	2254	1758	1360	1518	1263	870	819	968	1478	749	-	511	323	471	416	482	366	581	200
13	232	527	116	119	157	232	58	69	130	356	86	237	-	64	69	54	72	48	340	29
	579	1656	355	224	449	572	11	13	316	862	209	572	-	156	167	129	175	116	823	71
14	387	600	145	119	181	252	105	81	130	365	101	181	78	-	77	61	117	45	307	29
	964	1886	443	224	518	620	20	15	316	882	244	437	188	-	187	148	283	108	743	71
15	408	762	130	134	79	174	132	122	104	305	53	299	95	87	-	47	127	38	355	24
	1227	2392	399	384	145	429	320	296	252	739	10	723	229	211	-	8	308	91	858	58
16	387	718	101	92	66	118	104	116	21	238	44	280	78	74	50	-	117	8	345	7
	964	2254	310	174	121	38	252	22	4	575	8	678	188	178	9	-	283	2	835	1
17	964	249	404	364	398	426	236	254	304	526	230	361	117	156	150	130	-	117	388	66
	2901	782	1262	1046	1139	1049	572	615	736	1272	557	874	283	378	363	314	-	283	938	160
18	348	718	101	66	66	116	116	104	32	255	53	274	78	60	45	9	117	-	345	7
	868	2254	310	124	121	286	23	20	6	616	10	663	188	145	108	2	283	-	835	1
19	5154	2197	1975	1688	1737	1327	1304	1279	1270	1078	1030	629	796	593	605	554	560	499	-	268
	15510	6900	6175	4848	4968	3266	3158	3096	3072	2606	2490	1521	1924	1435	1463	1340	1355	1206	-	648
20	408	747	130	106	92	107	113	103	11	246	71	280	89	74	53	14	124	13	348	-
	1227	2346	399	199	169	262	275	250	2	595	13	678	215	178	128	3	299	2	840	-

Table 2.4: Optimal solutions for the MLF case

w	Solution	TC	TE
0.0	1(1056;7,13,17),2(346),4(553;3,8,14),5(595;6,9,11,15,16,18,20),10(280),12(285;19)	2927	12825
0.1	1(1108;7,13,14,17),2(346),4(436;3,8),5(595;6,9,11,15,16,18,20),10(280),12(285;19)	2927	12693
0.2	1(1108;7,13,14,17),2(346),4(436;3,8),5(595;6,9,11,15,16,18,20),10(280),12(285;19)	2927	12693
0.3	1(1765;3,4,7,8,13,14,17),2(346),5(627;6,9,10,11,15,16,18,20),12(285;19)	2972	12227
0.4	1(1056;7,13,17),2(346);3(890;3,5,8,11,12,14,15,19),16(330;6,9,10,18,20)	3021	11812
0.5	1(1056;7,13,17),2(346);3(890;3,5,8,11,12,14,15,19),16(330;6,9,10,18,20)	3021	11812
0.6	1(1056;7,13,17),2(346);3(890;3,5,8,11,12,14,15,19),16(330;6,9,10,18,20)	3021	11812
0.7	2(346),3(2057;1,4,5,7,8,11,12,13,14,15,17,19),16(330;6,9,10,16,18,20)	3096	11297
0.8	2(346),3(2057;1,4,5,7,8,11,12,13,14,15,17,19),16(330;6,9,10,16,18,20)	3096	11297
0.9	2(346),3(2057;1,4,5,7,8,11,12,13,14,15,17,19),16(330;6,9,10,16,18,20)	3096	11297
1.0	2(346),3(2057;1,4,5,7,8,11,12,13,14,15,17,19),16(330;6,9,10,16,18,20)	3096	11297
2.0	2(346;),3(1985;1,4,7,8,9,12,13,14,16,17,18,19,20),5(380;6,10,11,15)	3119	11249
3.0	3(2834;1,2,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20)	3241	11231
6.0	3(2834;1,2,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20)	3241	11231

cause, unlike the other components, inventory does not directly contribute to the supply chain emissions. Surprisingly, the contribution of the transportation component to the total cost decreases slightly as the environmental objective weight is increased, which can be attributed to the increased use of the less polluting, yet more expensive, electric trucks.

The computational performance of the algorithm is reasonable for a simulation-optimization approach. The computational time ranges between 2082 and 9290 seconds, with an average of 4271 seconds for the tested instances.

2.7.2 Case II: A cold supply chain network for vaccines in Ontario

In this case we use our approach to design a cold supply chain for publicly-funded vaccines in Ontario. We use the same network of Case I (*i.e.*, the 20 largest cities in the province) and focus on the 13 vaccines having annual demand of more than 100,000 doses, which represent about 96% of the total number of non-flu vaccine doses delivered. Flu vaccine is excluded because it is highly seasonal and has different logistical considerations.

Vaccines are small volume, high margin products that are extremely sensitive to the ambient temperature and have to be maintained under tight temperature control to preserve their efficacy. Vaccine packages are shipped in bulk quantities from two local manufacturing facilities (*Sanofi Pasteur* facility in Toronto and *GlaxoSmithKline* facility in Mississauga)

Table 2.5: Distance between cities in the vaccination network case

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	450	30	40	70	190	30	40	110	370	60	390	60	100	110	100	260	90	1390	110
2		430	440	480	630	370	420	500	800	470	490	360	410	520	490	170	490	1500	510
3			20	50	170	50	40	80	350	40	390	80	100	90	70	280	70	1370	90
4				60	170	50	30	80	340	50	390	90	90	110	70	300	50	1390	80
5					130	100	80	70	310	20	440	130	150	60	50	330	50	1440	70
6						210	190	110	190	140	530	240	260	180	100	440	120	1370	110
7							20	120	380	90	380	50	90	140	110	250	100	1380	120
8								100	370	70	360	60	70	130	100	270	90	1360	110
9									290	70	460	150	150	120	20	350	30	1460	10
10										320	720	420	430	360	280	620	300	1270	290
11											430	120	140	60	50	320	60	1430	80
12												380	290	480	450	580	440	1010	450
13													140	170	140	210	140	1430	160
14														190	160	340	130	1290	160
15															100	370	110	1490	130
16																340	20	1450	30
17																	340	1630	360
18																		1450	30
19																			1460

to DCs, and then to local dispensaries in each city, before they are dispatched to vaccination centers upon request. Demand for vaccines is quite predictable and fairly uniform, thus it makes sense to use a simple inventory management policy like the EOQ with backordering in DCs. Vaccines are shipped to demand points at constant (*e.g.*, weekly) intervals using small refrigerated vans. Thus, the annual cost and emissions for serving a dispensary from a DC is proportional to the distance between them, with coefficients $\$78/km$ and 142 $kgCO2/km$, respectively. As with the first case, the demand points are the candidate locations for DCs. The distance matrix is shown in Table 2.5.

The closed form expressions for the maximum inventory level s_{jl} (2.30) and the total inventory cost t_{jl} (2.31) functions in the classical EOQ with backordering policy are used directly without need for simulation. While the cost per vaccine dose varies widely, we assume equal annual holding and backordering costs for all vaccine types of $\$250/package/year$, where a package consists of 100 doses. The ordering cost is set to $\$200/order$, and the shipping cost and GHG emissions between suppliers and DCs equal $\$0.04/km/package$ and 0.08 $kgCO2/km/package$, respectively. The annual demand for vaccines is estimated

Table 2.6: Annual demand for vaccines in the vaccination network case

Type/City	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
DTaP-IPV-Hib	1073	362	293	215	213	150	124	118	90	87	72	66	62	56	54	52	50	50	44	41
HB	453	153	124	91	90	63	52	50	38	37	30	28	26	23	23	22	21	21	19	17
HPV4	243	82	67	49	49	34	28	27	20	20	17	15	14	13	12	12	11	11	10	9
Men-C-ACWY	225	76	61	45	45	32	26	25	19	19	15	14	13	12	11	11	11	11	9	9
Men-C-C	296	100	81	59	59	41	34	33	25	24	20	18	17	15	15	14	14	14	12	11
MMR	552	186	150	111	110	77	64	61	47	45	37	34	37	29	28	27	26	26	23	21
Pneu-C-13	854	288	233	171	170	120	99	94	72	69	57	52	49	44	43	41	40	30	35	33
Pneu-P-23	389	132	106	78	77	55	45	42	33	31	27	24	22	20	20	19	18	18	16	15
Rot-1	470	159	128	94	93	66	54	52	40	38	32	29	27	24	24	23	22	22	19	18
Td	602	203	164	121	120	84	70	66	51	49	41	37	35	31	30	29	28	28	25	23
Tdap	1271	429	347	255	253	178	148	140	107	103	86	78	73	66	64	62	60	59	52	49
Tdap-IPV	275	93	75	55	55	38	32	30	23	22	19	17	16	14	14	13	13	13	11	11
Var	710	240	194	142	141	99	82	78	60	57	48	43	41	37	36	34	33	33	29	27

from [9] based on the population of each city and is depicted in Table 2.6. Since the size of products is very small, the storage volume, and consequently the variable capacity cost of DCs, is negligible. The annual cost and GHG emissions of a DC are considered fixed at \$9000 and 3000 $kgCO2$, respectively, regardless of its capacity.

As in Case I, we test using different values of w, ranging between 0 (pure cost minimization) and 50 (dominant emissions minimization objective). The results are shown in Table 2.11. In the column titled ‘Solution’ we report the opened DCs and the retailers assigned to them inside the parenthesis, whereas the columns ‘TC’ and ‘TE’ show the total cost (in \$) and the total GHG emissions (in $kgCO2$), respectively. The computational time ranges between 19 and 1331 seconds with a mean of 528 seconds.

It is interesting to notice that, contrary to Case I, the primary trade-off in this case is between transportation and inventory costs, whereas the role of the capacity cost is comparatively marginal. When w is small, the risk-pooling effect related to the inventory cost is dominant and hence demand consolidation is beneficial and fewer DCs are opened. When w is increased, the effect of the inventory cost diminishes (since inventory does not have an environmental impact) and the transportation cost becomes dominant, calling for more DCs to be opened.

The trade-off between the two objectives, cost and emissions minimization, is illustrated in Fig. 2.5 that shows the Pareto frontier curve. It can be noticed, as in Case I, that a substantial reduction in GHG emissions can be achieved without significantly increasing the cost. For example, 25% reduction in the GHG emissions can be achieved with only 3%

Table 2.7: Optimal solutions for the vaccination network case

w	Solution	TC	TE
0.0	3(1,2,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,20),19	353870	384380
0.1	3(1,2,4,5,6,7,8,9,10,11,13,14,15,16,17,18,20),12,19	354952	346190
0.2	3(1,2,4,5,6,7,8,9,10,11,13,14,15,16,18,20),12,17,19	359282	316652
0.5	3(1,2,4,5,6,7,8,9,11,13,14,15,16,18,20),10,12,17,19	366824	286752
1.0	3(1,2,4,5,7,8,11,13,14,15),10,12,16(6,9,18,20),17,19	391722	260572
2.0	3(1,2,4,5,7,8,11,13,14,15),10,12,16(6,9,18,20),17,19	391722	260572
5.0	3(1,2,4,5,7,8,11),10,12,13,14,15,16(6,9,18,20),17,19	444461	239054
10.0	3(1,4,5,11),6,7(2,8),9(16,18,20),10,12,13,14,15,17,19	506731	230582
20.0	3(1,4,5,11),6,7(2,8),9(20),10,12,13,14,15,16(18),17,19	525954	228757
50.0	1,3(4,5),6,7(2,8),9(20),10,11,12,13,14,15,16(18),17,19	590188	226413

increase in cost by setting w to 0.5 instead of 0. However, the cost of emissions reduction increases exponentially beyond this point.

Figure 2.6 shows the breakdown of the different components in the combined objective function. The inventory cost contribution is high initially due to high value of the products. When w is increased, the inventory cost increases as an absolute value as more DCs are opened and the risk-pooling advantage is lost. However, it decreases as a ratio of the total objective value when the other cost components (transportation and capacity) increase at a faster pace thanks to the environmental penalty imposed.

2.7.3 Case III: A hypothetical case

To verify the simulation model and evaluate the computational performance of the approach, we construct a hypothetical case study on the same network of Case I but with different model parameters. Two facilities in London and Kingston produce two types of products to satisfy the demand in the 10 or 20 largest cities in Table 2.8. Products are shipped in bulk quantities from the production facilities to warehouses, where they are stored and then distributed to retailers in the demand nodes. Every demand node is also a candidate location for locating a warehouse.

Two subcases are considered: stochastic and deterministic demand. In the stochastic case, orders from retailers follow a binomial distribution, *i.e.*, each retailer at a given day makes an order with a probability p_i which is constant and independent across days. When

Table 2.8: Demand parameters in the hypothetical case

i	City	Pop_i	p_i	ν_{i1}	ν_{i2}
1	Toronto	2,615	1/3	10	16
2	Ottawa	883	1/5	6	9
3	Mississauga	713	1/6	6	9
4	Brampton	524	1/8	6	8
5	Hamilton	520	1/8	6	8
6	London	366	1/10	5	7
7	Markham	302	1/11	4	7
8	Vaughan	288	1/11	4	6
9	Kitchener	220	1/14	4	6
10	Windsor	211	1/14	4	6
11	Burlington	176	1/16	4	6
12	Sudbery	160	1/17	4	5
13	Oshawa	150	1/18	4	5
14	Barrie	136	1/19	3	5
15	St. Catharines	131	1/19	3	5
16	Cambridge	127	1/20	3	5
17	Kingston	123	1/20	3	5
18	Guelph	122	1/20	3	5
19	Thunder Bay	108	1/22	3	5
20	Waterloo	100	1/23	3	5

an order is made, the number of units ordered of product l is a random variable that follows a Poisson distribution with rate ν_{il}. In the deterministic case, an order is placed by retailer i every $1/p_i$ days and consists of ν_{il} units of each product. In both cases, the annual demand of product l by retailer i is $(p_i \nu_{il} \chi)$, where χ is the number of business days in a year, assumed here to be 300 days. We generate the demand data as follows: using Pop_i, the population of city i in thousands according to the 2012 census, we set the order probability: $p_i = \frac{1}{\left\lfloor 500/Pop_i^{2/3} \right\rceil}$. Average order size is calculated as: $\nu_{il} = \left\lfloor \frac{Pop_i}{\xi_l p_i} \right\rceil$, where $\xi_1 = 750$ and $\xi_2 = 500$. This selection of the parameters ensures that the average order size and the average inter-arrival time of orders are both integers while the correlations between the population and the annual demand are very high, exceeding 0.999 for both products. Table 2.8 depicts the demand parameters.

Products are shipped from plants to warehouses in long-haul trucks of capacity $90m^3$

Table 2.9: Truck characteristics in the hypothetical case

Truck type	1	2	3
Fuel type	Diesel	Diesel	Electricity
Capacity (m^3)	90	30	30
Shipment range (km)	unlimited	unlimited	100
Cost ($\$/km$)	1.5	0.7	1
CO2-equivalent emissions (g/km)	1100	800	0

that cost $\$1.5/km$. A unit of product 1 and product 2 occupies $3m^3$ and $1m^3$, respectively. Products are shipped from warehouses to retailers in 2 types of conventional trucks with capacities of 30 and $90m^3$, and one type of electric truck of $30m^3$ capacity that can be used for shipment distances of 100 km or less. Based on the demand parameters presented in Table 2.8, the probability of a shipment size of 0 or $> 90m^3$ is very small, so these possibilities are ruled out in our calculations. Table 2.9 presents the characteristics of each type of truck.

We use two weight factors w_c and w_i to change the contribution of the capacity and inventory costs, respectively, relative to the transportation cost in the objective function. Both the capacity and inventory cost weights are changed between 1 and 4. Annual capacity cost of the warehouses has two components: a fixed cost of $\$2000w_c$, and a volume-dependent cost that follows the function $f_j(Q_j) = 50w_cQ_j^{exp_1}$. Each warehouse has a base annual CO2-equivalent emissions of $50w_c$ metric tonne and a volume-dependent emissions that follow the function $g_j(Q_j) = 2w_cQ_j^{exp_2}$ tonne of CO2-equivalent emissions. For the base case scenario, the exponents exp_1 and exp_2 are set to 0.8 and 2/3, respectively. Each warehouse manages its inventory independently. Whenever an order is placed, a fixed cost of $\$20w_i$ is incurred, whereas the holding cost is $\$0.2w_i/day$ and $\$0.1w_i/day$ and the backordering (shortage) cost is $\$0.5w_i/day$ and $\$0.2w_i/day$ for a unit of inventory of product 1 and 2, respectively. There is a lead time of 2 days for shipping orders from plants to warehouses.

First, we consider the case when demand is deterministic and inventory is managed using an EOQ policy with backordering. Formulas (2.28-2.31) are used to get an approximate solution under the assumption of continuous and constant demand (*i.e.*, "sawtooth" inventory pattern), then simulation-optimization is implemented to evaluate the the maximum inventory level s_{jl} and the total inventory cost t_{jl} functions and solve the problem under the more realistic assumption of discrete demand (*i.e.*, stepwise inventory pattern). That was done primarily to validate the simulation model by comparing the results of the

Table 2.10: Results for the EOQ policy in the hypothetical case

w_c	w_i	Simplified				Actual			
		Optimal	CPU	Nodes	Facilities	Optimal	CPU	Nodes	Facilities
1	1	96,765	134	2019	2,3,6,19	106,319	1,472	1,459	1,2,6,19
1	2	104,222	192	2923	2,3,6,19	114,741	1,436	1,467	1,2,6,19
1	4	118,457	184	2929	2,3,6,19	130,076	1,335	1,427	2,11
2	1	118,073	19	509	1,2,6,19	126,427	231	241	2,11
2	2	124,950	17	567	2,11,19	132,636	230	239	2,11
2	4	136,975	17	457	2,11,19	144,271	232	233	3
4	1	144,074	2	59	3	145,445	29	25	3
4	2	148,367	2	51	3	150,215	33	29	3
4	4	156,955	1	49	3	159,754	31	27	3

two methods, but also to quantify the effect of relaxing the smooth demand assumption. When using simulation to evaluate the inventory function, we calculate the Economic Order Quantity using (2.28), whereas the reorder level is found by conducting simulation runs with numerous levels and selecting the one that results in the lowest inventory cost. The maximum inventory level and the total inventory cost are found at the optimal reorder point. In this round of experiments, we focus on the 20-node case only. All tested instances were solved to optimality.

The columns ‘Optimal’, ‘CPU’, ‘Iter’ and ‘Facilities’ depict the optimal solution, the computational time in seconds, the number of B&B nodes, and the opened facilities, respectively.

It is clear that as the weight of the capacity and inventory cost components increase, the number of opened warehouses decreases to take advantage of economies of scale. The number of warehouses drops from 4 warehouses when a weight of 1 is used for both cost components to a single warehouse when these weights are increased to 4. Capacity cost can increase as a result of higher land, construction or equipment cost, but also as a result of higher electricity cost, especially in hot and humid climates and with frozen products (as opposed to chilled ones) when the power consumption of the refrigeration system is relatively high. On the other hand, inventory holding cost increases proportionally with the value of the products stocked. For high value products, the cost of inventory holding and backordering may constitute a sizable proportion of the total cost of the supply chain. We also notice significant differences in the total cost between the simplified and the actual cases. Most of the cost difference is attributed to the additional capacity required to accommodate the increased maximum inventory level in warehouses. There is also a cost difference due to unequal inventory holding and backordering costs.

By comparing the computational time between the simplified and the actual cases, one can conclude that most of it is spent on evaluating the inventory functions using the simulation-optimization algorithm. It is also noticed that the number of B&B nodes and the computational time decreased significantly with the increase in capacity cost, whereas the effect of the inventory cost on the computational burden did not show a clear trend. This observation can be explained by the fact that when the capacity cost is dominant, fewer warehouses are opened and thus less simulation runs are needed. In all tested cased, the computational time did not exceed 25 minutes, which is acceptable for this design problem. This includes the time required to calculate the problem parameters from the raw data.

Next, we test for the case when inventory is managed using an (r, nQ) inventory policy where Q is the full-truck load (*i.e.*, $90m^3$) of the product. The problem is solved for the stochastic demand case using the simulation-optimization algorithm. In addition to the base case scenario, three other scenarios are tested:

- Zero emissions cost where the cost of emissions is entirely disregarded. The purpose is to evaluate the effect of including (or excluding) the environmental considerations when selecting the supply chain configuration and operational plan.

- Strong economies of scale where the exponents exp_1 and exp_2 are halved to represent steeper diminishing marginal cost and emissions functions. This is important since the extent of which economies of scale takes into effect varies widely from a technology and/or situation to another.

- Extended lead time where the lead time (L) is doubled to investigate the effect of delay in shipping products to warehouses. This delay can be a result of the production policy implemented by the suppliers (*e.g.*, order-to-produce or batch production) or to logistical inefficiencies (*e.g.*, congestion).

For each scenario and cost weights set, 10 randomly generated instances are tested to obtain a 95% confidence interval for the true mean of the system cost. The results are presented in Table 2.11. We also include the capacity of each warehouse opened between the brackets after its number in the ‘Facilities’ column. Since the warehouse capacity obtained varies from a simulation run to another, we report the highest capacity obtained or the upper limit of the 99% confidence interval rounded to the next integer, whichever is higher.

Table 2.11: Results for the (r,nQ) policy in the hypothetical case

w_c	w_i	10 nodes				20 nodes			
		Optimal	CPU	Nodes	Facilities	Optimal	CPU	Nodes	Facilities
Base case									
1	1	75,349±138	39	59	2(180),3(192),6(180)	109,913±113	930	997	2(180),3(210),6(180)
1	2	83,561±220	46	71	2(180),3(198),6(180)	118,614±183	963	1,076	2(180),3(201),6(180)
1	4	99,111±381	69	111	2(180),5(214)	135,358±303	897	1,069	2(180),3(232)
2	1	93,095±338	17	24	2(180),3(226)	130,334±251	90	133	2(180),11(228)
2	2	99,398±261	16	21	2(180),3(214)	137,222±316	84	124	2(180),3(228)
2	4	112,347±482	17	24	2(180),3(214)	151,113±473	88	118	2(180),3(228)
4	1	118,113±263	8	8	3(222)	156,812±453	16	23	3(250)
4	2	122,922±346	9	9	3(225)	152,172±524	17	23	3(247)
4	4	132,256±275	9	10	3(222)	172,478±254	17	26	3(244)
Zero emissions cost									
1	1	67,903±70	78	111	2(180),3(192),6(180)	99,942±169	2,059	2,071	2(180),3(210),6(180),19(180)
1	2	76,107±95	76	112	2(180),3(192),6(180)	110,471±178	2,043	2,395	2(180),3(198),6(180)
1	4	92,521±172	111	177	2(180),3(192),6(180)	127,756±320	2,593	2,985	2(180),3(201),6(180)
2	1	83,622±143	24	35	2(180),3(198),6(180)	117,572±123	218	294	2(180),3(201),6(180)
2	2	91,714±200	35	54	2(180),3(210)	126,147±116	229	304	2(180),3(198),6(180)
2	4	102,633±671	40	64	2(180),3(210)	140,173±384	256	351	2(180),3(232)
4	1	104,513±185	13	17	2(180),3(210)	140,007±259	66	85	2(180),3(228)
4	2	110,928±222	14	19	2(180),3(210)	147,905±334	64	85	2(180),3(228)
4	4	120,943±265	10	13	3(222)	160,874±380	63	83	3(247)
Strong economies of scale									
1	1	63,670±86	36	51	2(180),3(195),6(180)	94,408±189	662	635	2(180),3(210),6(180),19(180)
1	2	71,786±170	31	45	2(180),3(195),6(180)	104,845±119	771	819	2(180),3(201),6(180),19(180)
1	4	87,954±356	45	65	2(180),3(195),6(180)	124,187±303	900	1,068	2(180),3(201),6(180)
2	1	74,614±123	16	20	2(180),3(198),6(180)	108,970±41	47	51	2(180),3(198),6(180)
2	2	82,783±242	16	21	2(180),3(198),6(180)	117,819±157	63	78	2(180),3(210),6(180)
2	4	98,740±505	24	37	2(180),3(198),6(180)	134,957±279	107	149	2(180),5(226)
4	1	74,614±123	15	20	2(180),3(198),6(180)	127,742±107	17	18	2(180),3(228)
4	2	97,894±125	8	7	2(180),3(211),6(180)	134,706±108	20	24	2(180),3(223)
4	4	110,661±287	9	9	2(180),3(211)	148,669±425	23	30	2(180),3(232)
Extended lead time									
1	1	76,585±156	47	72	2(180),3(249),6(180)	111,350±105	442	508	2(180),3(273),6(183)
1	2	85,152±187	54	85	2(180),3(249),6(180)	120,479±208	477	549	2(180),3(261),6(192)
1	4	99,911±220	60	94	2(180),3(286)	138,233±379	375	512	2(180),3(316)
2	1	96,046±370	19	28	2(180),3(300)	133,239±319	93	125	2(180),3(318)
2	2	102,785±393	18	27	2(180),3(300)	140,670±261	97	134	2(180),3(316)
2	4	116,363±407	19	29	2(180),3(295)	155,184±308	84	117	2(180),3(316)
4	1	124,358±471	8	8	3(318)	164,106±228	20	25	3(399)
4	2	129,473±486	9	9	3(318)	169,348±404	17	23	3(338)
4	4	139,793±569	9	10	3(318)	180,670±515	18	25	3(339)

First we look at the base-case results. As with the EOQ case, the number of opened warehouses decreases when the inventory and capacity costs increase, and both the computational time and the number of tested nodes have inverse relations with the weight assigned to the capacity cost. Furthermore, we note that the total cost in the (r, nQ) policy case is slightly higher than that of the corresponding EOQ case. There are two reasons for this difference: first, when the shipment quantity is deterministic, smaller, less expensive trucks are used for all shipments between warehouses and retailers except the retailer in Toronto which requires a large truck for handling its demand. On the other hand, the shipment size in the stochastic demand case varies widely, which necessitates the use of the Large trucks more frequently. Second, the order quantity in the (r, nQ) policy is pre-determined exogenously, whereas it is optimized in the EOQ policy. We should remember, however, that the shipment cost per unit between production plants and warehouses is assumed constant based on full capacity utilization of trucks, which is true for the (r, nQ) case but not the EOQ case. If the less-than-perfect truck utilization is taken into consideration, the EOQ policy might result in significantly higher total cost.

The computational time and the number of nodes ratios between the 20 nodes and the corresponding 10 nodes cases vary between around 2 and 20, depending on the cost weights. When the weights are small, more warehouses are opened and simulated, significantly increasing the computational time. However, for the base-case instances, the mean time did not exceed 1000 second for the smallest weights. It is interesting to note the variability of warehouse selection when the weights are changed.

For the free emissions scenario, we note that more warehouses are opened compared to the bases case. This is expected since the exponent of the cost function is selected smaller than that of the emission function, implying weaker economies of scale, and consequently less demand pooling. With the emission cost is neglected, one would expect more usage of less expensive conventional trucks that can travel longer distance, which leads to a more demand assignment concentration and less opened warehouses. However, the effect of this factor seems minimal. With the removal of emissions cost, the contribution of the 'stochastic' variable capacity cost decreases compared to the 'deterministic' transportation and fixed costs, leading to less variability in the results as shown in the narrower confidence intervals. Also, with the more warehouses opened comes higher computational time and larger number of nodes as explained earlier.

When the capacity cost and emissions functions exhibit stronger economies of scale, one would expect higher concentration of demand and less warehouses to open. Surprisingly, the opposite happened in our experiments. This can be explained by fact that when the exponents of the capacity cost and emission function are reduced beyond a certain level, the variable capacity cost becomes very small (approaching 1 as the exponent goes to 0) and the contribution of the capacity cost diminishes compared to the transportation costs. As the transportation costs become dominant, it is better to open more warehouses to minimize the total cost. It is important to note that the warehouse capacity is always equal to or greater than 180, the size of two full-truck-loads so it can accommodate shipments of the two products if they arrive at the same day.

Finally, when the lead time is doubled, we notice a remarkable increase in the capacity of warehouses, especially those serving high demand like the warehouse in node 3 (Mississauga). The reorder level at these warehouses is set quite high so they can serve most of the stochastic demand during the lead time. As a result, the inventory cost increased significantly compared to the base case and, consequently, the variability of the total cost increased.

2.8 Conclusions

In this chapter, we developed a new mathematical model for designing cold supply chains with environmental considerations and proposed a novel approach to solve it efficiently. The model links the different components of the design problem by minimizing the capacity, inventory, and transportation costs simultaneously while accounting for realistic considerations such as stochastic demand, the inventory policy used in warehouses, and economies of scale inherent in different aspects of the system. Our model differs from the traditional green supply chain models found in the literature in three main aspects:

1. It considers the entire global warming impact of the cold supply chain, including the effect of refrigerant gas leakage in addition to the CO2 emissions related to energy consumption.

2. It does not stipulate any restrictive assumptions regarding the demand pattern or inventory policy implemented at warehouses except concavity of the inventory functions.

3. It bases the determination of the warehouse capacity on the actual storage requirements as determined by the inventory policy and not on the throughput.

The solution approach combines the efficiency of optimization methods with the accuracy of simulation methods. We were able to provide a closed-form expression for the best Lagrangian multipliers so that the Lagrangian bound is obtained in a single iteration alongside a feasible solution. Lagrangian decomposition is embedded in a B&B framework to close the optimality gap. When the inventory functions cannot be expressed explicitly in the mathematical model, a simulation-optimization algorithm is used within the Lagrangian approach to estimate these functions at the branching points.

The proposed approach was tested on two realistic cases representing industries with different logistical characteristics and a set of hypothetical instances for testing the computational performance. The results show that it is possible to substantially reduce the global warming effect of cold supply chains with a small increase, and sometimes with virtually no increase, in cost. Furthermore, we shed light on how the contribution of the different cost components affect the design of the cold supply chain. The managerial insights drawn from these results enable the decision-makers to identify and target the primary cost and emissions drivers in their supply chain networks. Also, through these tests, the proposed approach is shown to be versatile and can be tailored to suit many real life situations including different demand patterns, inventory policies, transportation modes, and operational constraints.

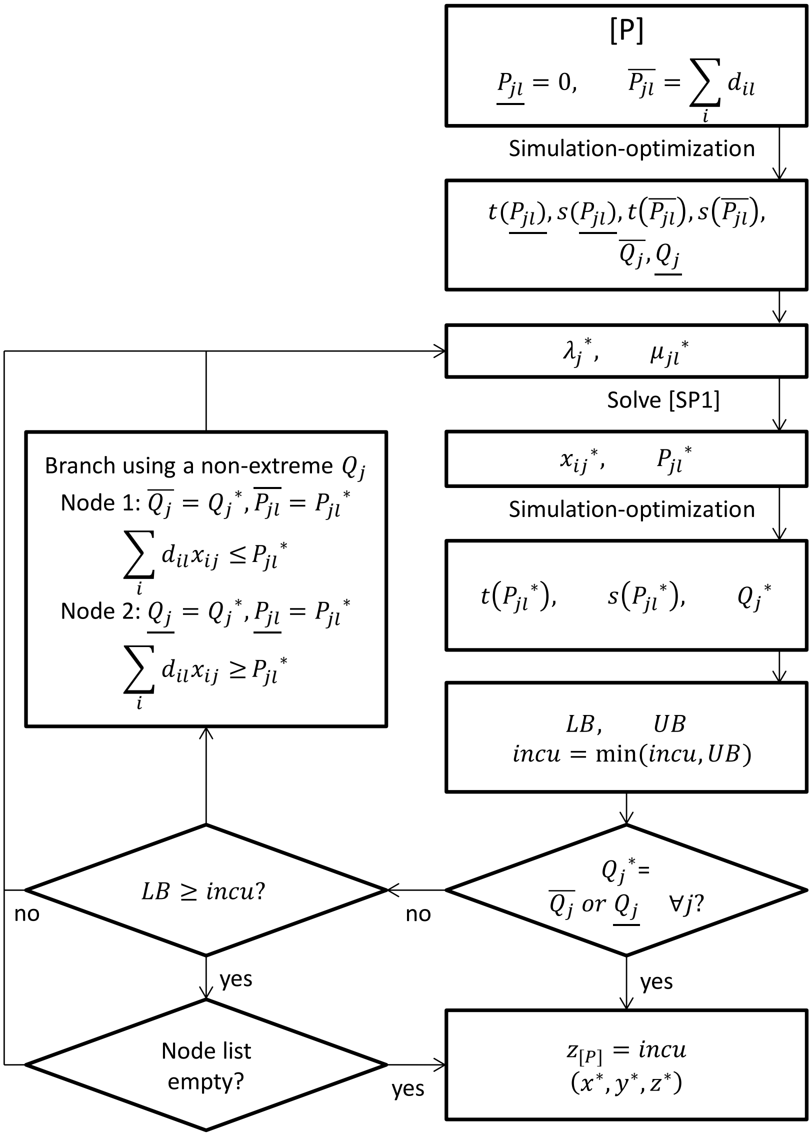
[P]
$\underline{P_{jl}} = 0, \quad \overline{P_{jl}} = \sum_i d_{il}$
Simulation-optimization
$t(\underline{P_{jl}}), s(\underline{P_{jl}}), t(\overline{P_{jl}}), s(\overline{P_{jl}}),$
$\overline{Q_j}, \underline{Q_j}$
$\lambda_j^*, \quad \mu_{jl}^*$
Solve [SP1]
$x_{ij}^*, \quad P_{jl}^*$
Simulation-optimization
$t(P_{jl}^*), \quad s(P_{jl}^*), \quad Q_j^*$
$LB, \quad UB$
$incu = \min(incu, UB)$
$Q_j^* =$ $\overline{Q_j}$ or $\underline{Q_j}$ $\forall j?$
no
$LB \geq incu?$
no
yes
Node list empty?
yes
yes
$z_{[P]} = incu$
(x^*, y^*, z^*)
Branch using a non-extreme Q_j
Node 1: $\overline{Q_j} = Q_j^*, \overline{P_{jl}} = P_{jl}^*$
$\sum_i d_{il} x_{ij} \leq P_{jl}^*$
Node 2: $\underline{Q_j} = Q_j^*, \underline{P_{jl}} = P_{jl}^*$
$\sum_i d_{il} x_{ij} \geq P_{jl}^*$

Figure 2.2: The solution method for the cold supply chain design problem

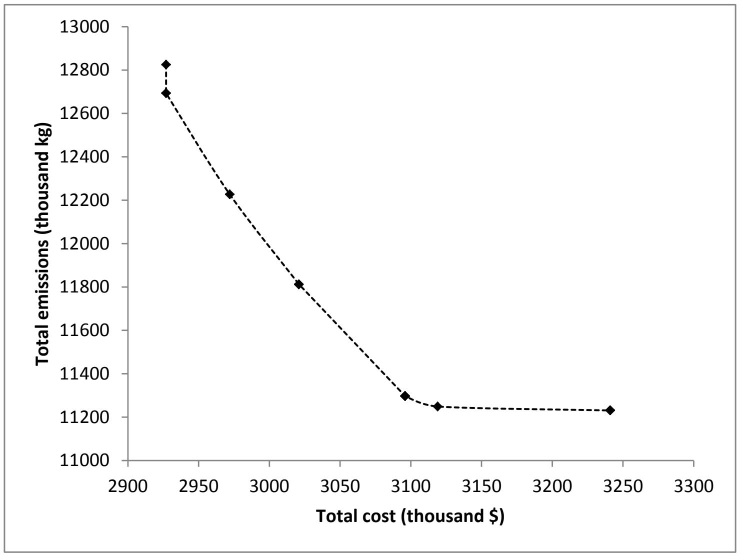

Figure 2.3: The trade-off between cost and emissions in the MLF case

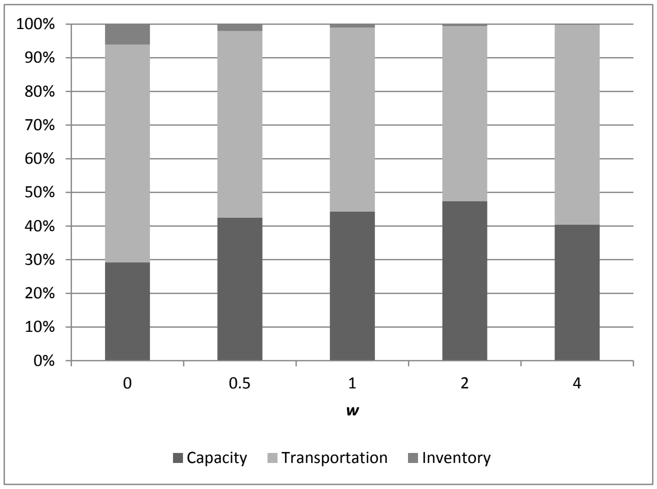

Figure 2.4: Breakdown of the cost components in the MLF case

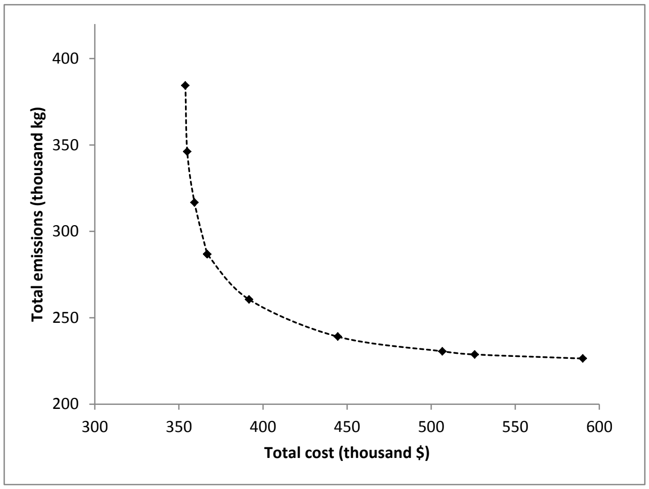

Figure 2.5: The trade-off between cost and emissions in the vaccination network case

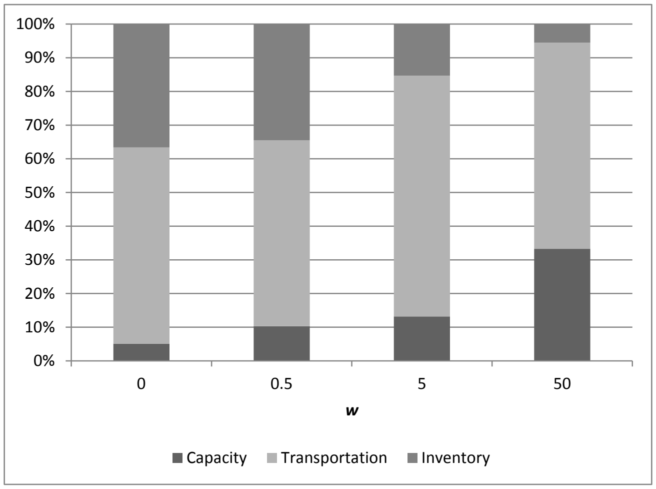

Figure 2.6: Breakdown of the cost components in the vaccination network case

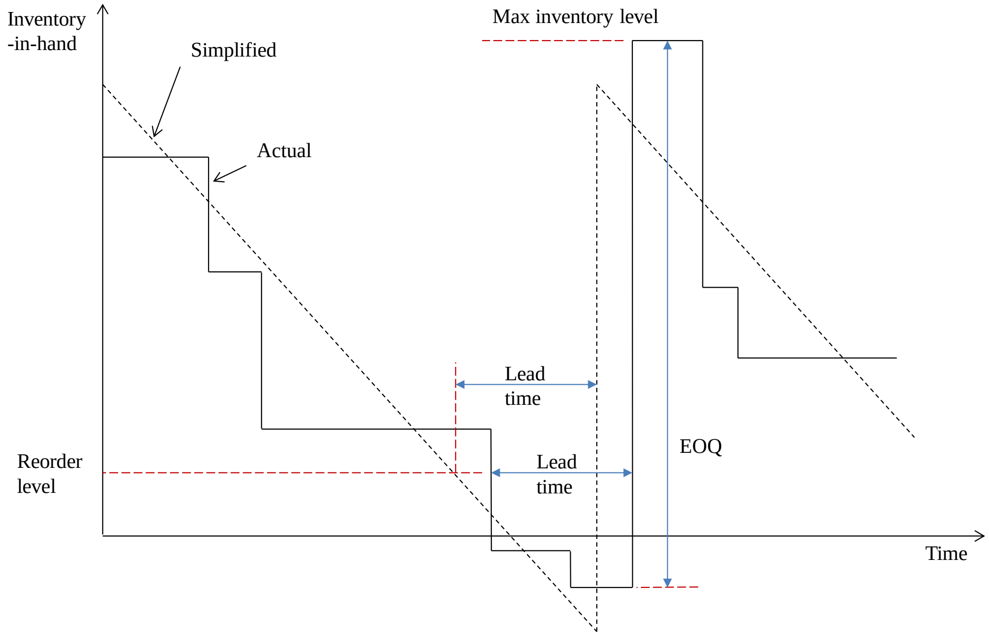

Figure 2.7: Difference between simplified and actual EOQ models in the hypothetical case

Chapter 3

Sterilization network design

3.1 Introduction

The cost of health care in developed nations is increasing rapidly. In Canada, for instance, public health care expenditures in dollar terms grew by 7.0% per year during the 10-year period from 2000 to 2010, while as a percentage of GDP it increased from 9.2% to 11.9% [33]. Population aging, high costs of new technologies, and the high salaries of health care professionals, among other factors, are behind this sharp growth in expenditures. Hospital costs represented 32.5% of the total health care spending in 2010, making it the largest cost component [33].

This unsustainable trend has urged health care administrators to look for new ways to cut costs through more efficient operations and better organization of services. Pooling of hospital resources to take advantage of economies of scale and scope was identified as a promising alternative to cut health care costs while maintaining high quality services [103]. Resource pooling is fostered by a growing tendency of many hospitals to organize themselves in networks to exploit synergies and cost saving opportunities [73]. Several types of hospital operations are candidates for pooling, including: procurement of medical supplies [46], sourcing and transfusion of donated blood [79], purchasing of pharmaceutical products [29], and sterilization of medical devices [99].

Compared to other core areas of health care management, sterilization logistics have received little attention from the OR community. However, the last decade has witnessed a growing interest in this topic as the magnitude of sterilization costs became more apparent. Van de Klundert et al. [102] estimated the investments in medical sterilization equipment to exceed 500 million Euros in the Netherlands alone. In a recent market research report it was estimated that the global sterilization equipment and disinfectants market was valued at USD 5.13 billion at 2012, and is expected to grow by an average of 8.5% annually during the forecast period 2013-2019 [100].

Besides the possibility for self-organizing the sterilization services within a network of hospitals, these services can be outsourced to a third party. Nowadays, many hospitals and hospital networks opt to contract with external sterilization service providers for various reasons, including: freeing scarce hospital space, regaining focus on their core business, and achieving higher operational standards. This trend is underway in Europe since 2001 when *Sterience*, currently a subsidiary of *Dalkia*, opened its first sterilization center in the outskirts of the French city Lyon. In 2011, *SteriPro*, a Canadian subsidiary of *Dalkia* opened a sterilization center in Mississauga, Ontario to serve hospitals in the Greater Toronto Area (GTA). The number of hospitals served by *SteriPro* is growing steadily and, like its sister company in France, will have to open more sterilization centers to serve the growing demand [91].

In this chapter, we address the location, capacity and allocation problem faced by an entity providing sterilization services to hospitals in centralized facilities. This entity can be directed by a hospital network or a third-party company like *SteriPro*. The sterilization service provider aims to minimize its long-term total cost while ensuring high service level for its clients by determining the number of sterilization centers (SCs) to open, their location and size, the assignment of hospitals to them, the quantity of resources (equipment and personnel) to deploy, and the quantity of reusable medical devices (RMDs) to stock. Total cost includes set-up cost of SCs, both fixed and size-dependent, transportation cost between SCs and hospitals, holding costs of RMD stocks, and the annual cost of equipment and personnel.

Early work related to sterilization services organization includes the paper of El-Shafei [44], who proposed an exact procedure to solve location problems and used it to determine the optimal location of a central sterilization department in a hospital. Fineman and Kapadia [50] used inventory theory to determine the stock requirements of RMDs in a hospital under the assumption of constant demand. They broke the total RMD stock requirements into two components: processing stock to replenish used items through the processing cycle, and replacement stock to replace worn, lost, or damaged items. The replacement stock was further divided into working and safety stocks. Through their analysis, they ranked the level of supply requirement of RMDs according to their use frequency and prepackaging requirements.

The problem of sterilization logistics within a hospital was tackled by Van de Klundert et al. [102]. They first formulated the location-transportation problem as a deterministic mixed integer program, which was shown to represent a special case of the fixed charge network flow problem, and used a dynamic programming algorithm to solve it. Then, they extended their model to deal with dynamic, nondeterministic demand by comparing four operational policies in a simulation environment using different scenarios. Finally, they considered the problem of optimizing the composition of the RMD 'nets' and showed that it is strongly **NP**-complete. Tlahig et al. [98] developed a two-step iterative approach to decide whether to perform the sterilization services centrally or within each wing of the hospital under constant demand and cost assumptions. In the first step, the optimum configuration that minimizes the sterilization activity cost was determined, whereas in the second step the size of SC(s) was optimized. The proposed approach can not guarantee global optimality since the location-allocation and capacity problems were tackled separately.

In the previous references, the focus was on optimizing the sterilization logistics within a single hospital. As with many logistical applications, it is often beneficial to group clients and assign them to centralized hubs. Collective organization of the sterilization functions in multiple hospitals was first studied by Tlahig et al. [99]. They considered a group of hospitals that has to decide between distributed and centralized sterilization functions. An assumption was made that either all the hospitals are assigned to a single SC or each hospital performs its sterilization functions internally. The problem was formulated as a deterministic multi-period mixed-integer program that aims to minimize the sum of sterilization fixed and variable costs, transportation, transfer and storage costs. The problem was solved using a commercial solver after adding valid cuts. Due to the large number of variables and constraints in the proposed model and the use of a general-purpose solver, the approach is practical only for networks of few hospitals. Furthermore, the model assumptions are quite restrictive and some of them are unrealistic, especially the deterministic demand assumption.

We propose an alternative approach to that presented in [99] for designing a sterilization network to serve a set of hospitals. Unlike [99], we do not stipulate that all hospitals must have the same choice between performing sterilization functions internally or using a centralized SC. Furthermore, we tackle the more realistic case of stochastic demand with a threshold service level requirement and incorporate economies of scale for facilities through a concave capacity cost function. Our model is simpler, has a much smaller number of variables and constraints, and avoids some of the restrictive assumptions and simplifications of [99]. The new formulation leads to a mixed-integer concave minimization problem with linear and concave constraints. To solve this problem, we first reformulate it as a mixed-integer second-order cone programming problem and approximate the concave cost function using piecewise-linear functions. This reformulation enables us to solve large instances efficiently using powerful commercial solvers like CPLEX. The proposed approach is tested on a realistic case study under different scenarios and organization schemes. The results reveal that it is possible to achieve significant cost savings by consolidating the sterilization functions in specialized centers as opposed to the current decentralized scheme in which each hospital performs its sterilization functions independently. We also study the contribution of and the trade-offs between the main cost components of the sterilization network and the sources of cost savings in each scenario and organization scheme. It has been shown that ample cost savings can be achieved even when the capacity cost is entirely neglected due to the better utilization of resources and the risk-pooling effect. Both the methodology and the managerial insights drawn from the test case are of great interest to sterilization service providers.

The main contributions of this chapter are:

1. A new mathematical formulation for the sterilization network design that minimizes the capacity, transportation, resources and inventory-holding costs under a stochastic demand assumption, and that considers economies-of-scale and the risk-pooling effect.

2. A solution strategy that transforms the resulting mixed-integer concave minimization problem with concave constrains into a mixed-integer second-order cone program that can be solved easily using commercial solvers.

3. A comparison between different organization schemes of the sterilization services under different scenarios that shed light on the advantages and limitations of service consolidation and identify the main cost drivers in each scheme.

3.2 The sterilization cycle

A detailed description of a typical sterilization cycle in a hospital is provided in [102]; so here we present it briefly. Sterilized RMDs are placed in stocks (also called nets) such that each net contains the instruments needed for a certain operation. Just before the operation, nets are moved from the clean storage area in carts to the operating rooms. After the operation, all the instruments in the net are considered contaminated regardless of whether they are used or not, and are collected in the contaminated storage area. Next, they are moved in batches to the SC, whether internal or external.

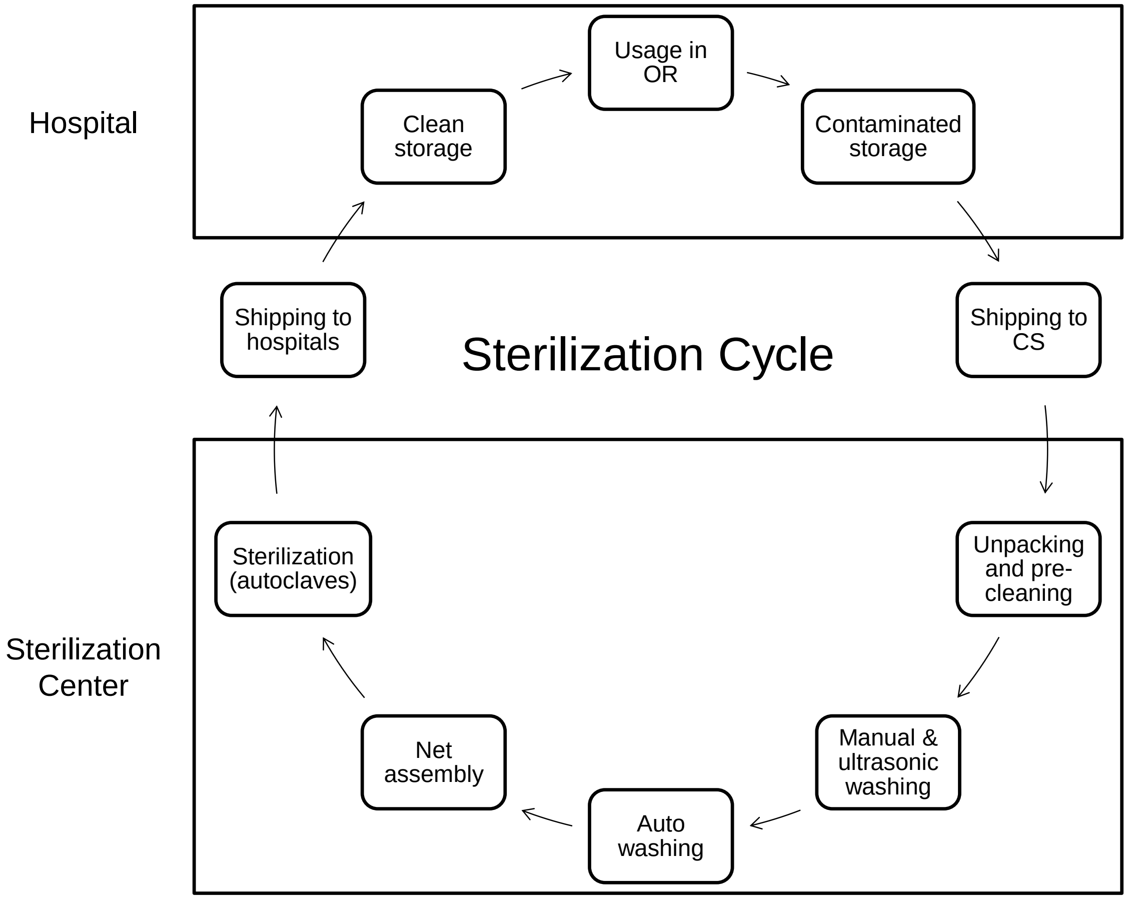

Figure 3.1: The Sterilization cycle

The following description of the sterilization cycle within a SC is based on the process flow at *SteriPro* Mississauga facility as outlined in [91], which is considered a standard procedure.

1. Carts of used RMD arrive from hospitals in trucks, where they are unloaded, scanned and staged in the appropriate receiving queues.

2. Carts are sent to the decontamination and sorting area where cases from the transport carts are unloaded, sorted and precleaned.

3. RMDs are sent to the manual washing stations, where they are manually washed or given ultrasonic washing, depending on the sterilization instructions, before they are placed back on the wash carts and passed to the automatic washer queue.

4. Wash carts are unloaded onto the Air Glide System (the conveyor) to enter the washers for a specified wash cycle, automatically determined by wash cart contents.

5. Trays are removed from the clean wash carts and distributed to the assembly tables based on hospital and surgery type.

6. Nets are assembled according to on-screen instructions and complete nets are sent down the roller line towards the autoclaves.

7. After being sterilized in the autoclaves, nets are let stand for the required cool down period.

8. Sterilized nets are placed into the transport carts and moved to the outbound shipping area, where they are staged and loaded into the outbound shipping trucks.

Figure (3.1) illustrates the complete sterilization cycle.

Realistically, as estimated in [91], RMDs processing in the SC takes about ten hours. If the transportation and usage times are added, a full cycle typically takes more than half a day. Thus, it is reasonable to assume that the RMD can be used only once every day. This means that a sufficient RMD stock must be held to meet the demand of one day with high probability.

Sterilization is a costly process that requires both specialized equipment and highly trained manpower. The most expensive equipment in the SC are the autoclaves and the washing machines. Automation of the washing and sterilization processes and the usage of conveyor systems to move RMD trays have reduced the number of workers required in the SC. However, there is still need for manpower to perform the unloading, manual washing, inspection and loading tasks.

3.3 Problem description

Consider a sterilization facility providing sterilization services to a group of hospitals. Different types of medical operations are performed in the hospitals, and each operation requires certain instruments, usually grouped in nets. It is often the case that nets are composed so that they are useable for a range of medical operations, not just a specific type. On the other hand, an operation may require more that one net. Thus, hereinafter we use nets as the unit of demand. Operations are of two types: planned and unplanned. For planned operations, the quantity required of each type of nets can be determined with certainty before the cycle (*e.g.*, day) begins. For unplanned operations, each hospital has to keep adequate quantities of nets such that it can fulfill the demand with very high reliability.

When choosing between centralized and distributed sterilization functions there is an underlying trade-off between transportation and capacity/process costs. On the one hand, the fewer the number of SCs, the higher the cost of transporting RMDs to and from hospitals. On the other hand, centralization results in the following cost savings:

1. Better utilization of resources: it is often the case that the sterilization equipment and personnel are underutilized due to the limited demand in a single hospital. Thus, it is possible to use the same resources to serve other hospitals as well.
2. Less resources required due to risk pooling: in a variable demand environment, the quantity of equipment and personnel is determined such that they can serve the maximum demand expected with a certain probability. When combining the demand of two or more hospitals, unless their demands are perfectly correlated, the maximum demand of them combined is lower than the sum of their maximum demands. Thus, fewer resources are needed.
3. Lower capacity cost due to economies of scale: larger premises usually cost less per unit capacity due to a variety of reasons, including: the more organization flexibility, the higher energy efficiency, and the sharing of overheads. Furthermore, quantity discounts for the acquisition and maintenance of sterilization equipment, the procurement of supplies, and the training of manpower can lead to significant savings.
4. Lower inventory cost due to risk pooling: If RMD stock is kept at the SCs and dispatched to hospitals as needed, less devices are needed than when hospitals keep their own inventory for the same reason explained in point 2.

While cost saving due to better utilization of resources is considered in [99], potential savings related to risk pooling and economies of scale are disregarded. Note that for savings from the last source to be realized, SCs should act as storage and distribution centers for RMDs in addition to their primary function of medical sterilization. Pooling has, repeatedly, been shown to reduce the demand variability, and consequently, the need for larger safety stocks [45], [31], [32]. This is particularly true for situations with moderate demand variability [23], as often the case for the demand of RMDs in hospitals. However, hospital managements might be reluctant to give up control over their stocks of RMDs without significant gains. The *status quo* is that each hospital keeps and manages its own stock of RMDs independently. However, they might opt to pool their stocks if the cost saving achievable by this shift outweighs the operational concerns like ownership and availability. It is still unclear whether the cost saving due to inventory pooling is sufficient to justify this shift in management practices.

We address the problem faced by the sterilization services provider when serving a group of spatially dispersed hospitals such that neither the transportation cost nor the service-related costs are dominant. In other words, it is not clear whether a centralized or a distributed sterilization system is better. In addition to classical trade-off between transportation and setup costs encountered in fixed-charge facility location models, we incorporate the effects of risk pooling and economies of scale, both favoring centralized schemes, in our study. In particular, we compare three schemes for managing the sterilization services:

1. Distributed service and RMD stock: each hospital undertakes the sterilization functions and holds its own RMD stock internally without need for any kind of coordination.

2. Centralized service and distributed RMD stock: sterilization functions can be performed by external centers but RMD stock is held and managed in-house.

3. Centralized service and RMD stock: SCs undertake the RMD stock holding and managing functions in addition to the sterilization functions.

3.4 Model formulation

The mathematical model is based on the centralized service and RMD stock scheme. Necessary modifications to suit it for the first two schemes are discussed later. Let $i \in I$,

$j \in J$, $p \in P$, $r \in R$ be the indices of hospitals, potential SC locations, RMD net types, sterilization resources (equipment and human), respectively. To formulate the problem, the following variables are introduced:

x_j	binary variable that takes value 1 if a SC is opened in location j;
y_{ij}	binary variable that takes value 1 if hospital i is assigned to SC j;
z_{jp}	stock of RMD nets of type p held in SC j;
s_{jr}	units of resource r deployed in SC j.

The objective is to minimize the long-run total cost of the sterilization network, including: set-up cost for SCs, the cost of transporting instruments between hospitals and SCs, RMDs holding cost in SCs, and the cost of equipment and human resources required to perform the sterilization services. Set-up cost has two components: a fixed cost c_j and a variable, size-dependent cost. In calculating the size of a sterilization center we consider the area A_j required for resources and RMD stocks: $A_j = \sum_{p \in P} u_p z_{jp} + \sum_{r \in R} v_r s_{jr}$, where u_p and v_r are the areas required for a unit of RMD net of type p and a unit of resource of type r, respectively, according to operating codes. We assume a concave variable cost function $f_j\left(A_j\right)$ that accounts for economies of scale.

Each hospital is served by a single SC. If the SC is in the hospital itself, no transportation cost is incurred. If the hospital is served by an external SC, a vehicle ships RMD nets from the SC to the hospital in the morning and returns the contaminated instruments at the end of the day. We assume that all shipments are direct without routing between hospitals. The annual cost of serving hospital i by SC j is d_{ij}, which depends on the distance and the type of vehicle used.

The daily demand for net type p originating from hospital i is an independent random variable that is assumed to follow a normal distribution with mean μ_{ip} and variance σ_{ip}^2. Therefore, the daily demand of net type p observed by the SC j is also normally distributed with mean $\sum_{i \in I} \mu_{ip} y_{ij}$ and variance $\sum_{i \in I} \sigma_{ip}^2 y_{ij}$. The stock of RMD nets held at each SC must satisfy, in full, the daily demand of the hospitals assigned to it with a certain threshold probability τ (*i.e.*, the 'fill rate'). Hence, the stock of RMD net type p held at SC j equals at least $\sum_{i \in I} \mu_{ip} y_{ij} + a\sqrt{\sum_{i \in I} \sigma_{ip}^2 y_{ij}}$, where $a = \Phi^{-1}(\tau)$, the inverse standard Normal CDF function corresponding to the fill rate τ. Even when the demand of individual hospitals is not normally distributed, according to the Central Limit Theorem, the consolidated demand can be reasonably approximated using a normal distribution.

Notice that the stock level decision z_{jp} depends on the location-assignment decisions only in the last scheme, namely when RMD stocks are managed at the SC level rather than the hospital level. Whereas when stock at each hospital is managed autonomously, the quantity of RMD nets held at each hospital can be determined exogenously. Let h_p denotes the holding cost of a unit of RMD net type p for one year, thus the inventory cost at SC j is $\sum_{p\in P} h_p z_{jp}$.

We assume that the transitional costs incurred to shift from the *status quo* to the new configuration are negligible compared to the steady-state costs, so they are disregarded. We also do not include the costs related to the storage of RMDs near the operating rooms and their movements within the hospitals as these costs will be incurred regardless of the network configuration. Furthermore, since the number of instruments to be sterilized is given and independent of the network configuration, we do not include the direct cost of sterilization which includes power, water and chemical agents in our model. However, the indirect costs of equipment and manpower must be included since these costs depend on the network configuration. Let C_r be the daily unit capacity of resource r, q_{pr} be the quantity of resource r required to process a unit of RMD net p and e_r be the annualized cost of a unit of resource r. Each SC must have adequate resources to process its entire RMD stock of the hospitals assigned to it in one day.

With that, the sterilization network design problem formulation is given by

$$[\mathcal{SNDP}]\text{:}\quad \min \quad \sum_{j\in J} c_j x_j + \sum_{j\in J} f_j\left(A_j\right) + \sum_{j\in J}\sum_{i\in I} d_{ij} y_{ij} + \sum_{j\in J}\sum_{p\in P} h_p z_{jp} + \sum_{j\in J}\sum_{r\in R} e_r s_{jr}$$

$$\text{s.t.} \quad \sum_{j\in J} y_{ij} = 1 \qquad \forall\, i \in I \tag{3.1}$$

$$x_j \geq y_{ij} \qquad \forall\, i \in I, j \in J \tag{3.2}$$

$$A_j = \sum_{p\in P} u_p z_{jp} + \sum_{r\in R} v_r s_{jr} \qquad \forall\, j \in J \tag{3.3}$$

$$z_{jp} \geq \sum_{i\in I} \mu_{ip} y_{ij} + a\sqrt{\sum_{i\in I} \sigma_{ip}^2 y_{ij}} \qquad \forall\, j \in J, p \in P \tag{3.4}$$

$$\sum_{p\in P} q_{pr} z_{jp} \leq C_r s_{jr} \qquad \forall\, j \in J, r \in R \tag{3.5}$$

$$x_j, y_{ij} \in \{0,1\}, \quad s_{jq} \geq 0, \text{ integer}, \quad z_{jp}, A_j \geq 0 \tag{3.6}$$

The first constraint assigns each hospital to exactly one SC. Constraint (3.2) ensures that this assignment is possible only if the SC is opened. Total SC area and RMD stock level are given by (3.3) and (3.4). Inequality (3.5) sets the minimum resources requirements at each SC. The terms of the objective function represent the fixed and variable (size-dependent) capacity, transportation, inventory holding, annualized resources costs, respectively. This formulation results in a concave mixed-integer program with linear and concave constraints. In the next section an approach is devised to solve the problem.

3.5 Solution method

Although the Lagrangian approach proposed in section 1.2 can be used to decompose the problem in order to deal with the concave terms in the objective function, the resulting subproblem is a mixed-integer problem with concave constraints that is difficult to solve. Since the subproblem has to be solved many times (*i.e.*, in every branching node), this inevitably leads to an excessive computational burden. Instead, we propose an alternative solution strategy for this problem that requires a larger mixed-integer second-order cone programming problem to be solved once to get a near optimal solution tractably. We start by approximating the concave variable capacity cost using a classical piecewise linearization. We linearize the functions $f_j\left(A_j\right)$ using piecewise segments linking the points

$$\left\{(0,0),\left(A_j^1, f_j(A_j^1)\right),\left(A_j^2, f_j(A_j^2)\right),\ldots,\left(A_j^{T_j}, f_j(A_j^{T_j})\right)\right\}$$

and special ordered sets of type 2 (SOS2) [16]. We can write A_j and the approximated functions $F_j(A_j)$ as:

$$A_j = \sum_{t=0}^{T_j} w_j^t A_j^t, \qquad \forall j \in J, \tag{3.7}$$

$$F_j(A_j) = \sum_{t=0}^{T_j} w_j^t f_j(A_j^t), \qquad \forall j \in J, \tag{3.8}$$

$$\sum_{t=0}^{T_j} w_j^t = 1, \qquad \forall j \in J. \tag{3.9}$$

and $\left\{w_j^0, w_j^1, \ldots, w_j^{T_j}\right\}$ are SOS2 variables. Note that we assumed, without loss of generality, that $A_j^0 = f_j(A_j^0) = 0$.

Next, we introduce the variable $\bar{z}_{jp} = z_{ip} - \sum_{i\in I} \mu_{ip} y_{ij}$. By substituting in (3.4), and noting that, since y_{ij} is binary, $y_{ij} = y_{ij}^2$, the stock level constraint can be written as

$$a^2 \sum_{i\in I} \sigma_{ip}^2 y_{ij}^2 \leq \bar{z}_{jp}^2 \qquad \forall\, j \in J, p \in P, \tag{3.10}$$

which is a second-order cone constraint. With that, we get the reformulated problem

$$\begin{aligned}
[\mathcal{RSNDP}]: \quad \min \quad & \sum_{j\in J} c_j x_j + \sum_{j\in J} \sum_{t=0}^{T_j} w_j^t f_j(A_j^t) + \sum_{j\in J} \sum_{i\in I} d_{ij} y_{ij} + \sum_{j\in J} \sum_{p\in P} h_p z_{jp} + \sum_{j\in J} \sum_{r\in R} e_r s_{jr} \\
\text{s.t.} \quad & \sum_{j\in J} y_{ij} = 1 && \forall\, i \in I \\
& x_j \geq y_{ij} && \forall\, i \in I, j \in J \\
& \sum_{t=0}^{T_j} w_j^t A_j^t = \sum_{p\in P} u_p z_{jp} + \sum_{r\in R} v_r s_{jr} && \forall\, j \in J \\
& a^2 \sum_{i\in I} \sigma_{ip}^2 y_{ij}^2 \leq \bar{z}_{jp}^2 && \forall\, j \in J, p \in P \\
& \sum_{p\in P} q_{pr} z_{jp} \leq C_r s_{jr} && \forall\, j \in J, r \in R \\
& \bar{z}_{jp} = z_{ip} - \sum_{i\in I} \mu_{ip} y_{ij} && \forall\, j \in J, p \in P \\
& \sum_{t=0}^{T_j} w_j^t = 1 && \forall j \in J \\
& x_j, y_{ij} \in \{0,1\}, \;\; s_{jq} \geq 0, \text{ integer}, \;\; z_{jp}, w_j^t \geq 0, \;\; \left\{ w_j^0, w_j^1, \ldots, w_j^{T_j} \right\} \; SOS2.
\end{aligned}$$

which is a mixed-integer second-order cone programming (MISOCP) problem that can be solved directly using commercial solvers. The solution algorithms for MISOCP problems have improved significantly in the last decade, and they are now capable of handling large problem easily. For example, Atamturk [15] has reported solving instances of the location-inventory problem (sec. 1.3.3) with 300 conic constraints and 22,650 integer variables after reformulating them as MISOCP in CPU times ranging between 7 and 260 seconds, depending on the problem parameters.

Table 3.1: RMD net characteristics in the SND problem

p	average demand per surgery	$u_p,\ q_{pr}$	h_p
1	1.50	0.25	100
2	1.25	0.30	120
3	1.00	0.30	150
4	0.75	0.35	180
5	0.50	0.35	200

Table 3.2: Sterilization resources in the SND problem

r	C_r	e_r	v_r
1	20	3000	8
2	30	6000	6
3	100	30,000	15

3.6 Numerical testing

To study the inherent trade-offs in the sterilization network design problem and validate our approach, we implement it to design a sterilization network for the public hospitals in southwestern Ontario, Canada, *i.e.*, Local Health Integration Networks (LHINs) # 1,2,3 and 4 [3]. Since most hospitals do not keep records of their usage of RMDs, we could not entirely base our model on real data. However, real and hypothetical estimated data are mingled to come up with a realistic set of model parameters that emulates a typical sterilization system.

We include only the 20 hospitals of Group A (General/teaching hospitals) and Group B (General hospitals with more than 100 beds) in the region under consideration. Other groups of hospitals (*e.g.*, rehabilitation, psychiatric, and chronic diseases hospitals) have minimal demand of RMDs. Out of the 508,238 surgeries performed in Ontario in 2012-2013, the estimated number of surgeries in every hospital is assumed proportional to its size (measured in the number of beds). There assumed to be 5 types of RMD nets, each having its expected demand per surgery, size (which is also equal to resources usage) and holding cost depicted in Table 3.1. Three resources are considered: automatic washing machines, sterilizers (autoclaves), and manpower, with the capacity, unit cost, and area required of each resource unit depicted in Table 3.2.

A single type of vehicles is used to ship RMD nets daily between hospitals and SCs.

Table 3.3: Estimated number of surgeries per year in the SND problem

i	hospital name	surgeries
1	Hamilton Health Sciences Corp.	32,070
2	London Health Sciences Centre	26,003
3	Niagara Health System	14,283
4	St. Mary's General Hospital	13,321
5	Windsor Regional Hospital	11,730
6	Guelph General Hospital	6,446
7	Bluewater Health	5,786
8	Brant Community Healthcare	4,295
9	Joseph Brant Hospital	4,177
10	Chatham-Kent Health Alliance	4,069
11	Grey Bruce Health Services	3,598
12	Cambridge Memorial Hospital	3,500
13	St. Thomas-Elgin General Hospital	2,801
14	Woodstock General Hospital	2,427
15	Stratford General Hospital	2,173

Thus, the shipping cost is assumed proportional to the distance, with a coefficient of \$500/km/year. We combined the demand of hospitals if the distance between them is less than 10 km, so we ended-up with 15 demand points only. The candidate locations for SCs are the demand points themselves. Table 3.4 shows the driving distance between demand points in km. There is an annual fixed cost of \$10,000 for each opened SC, whereas the annual size-dependent cost is calculated using the function \$$500A^{0.8}$, where A is the total SC area in m^2. The number of surgeries per year for each demand point is shown in Table 3.3.

The fill rate, τ (which is also type I service level), is set to 0.99, meaning that with probability 99% each SC is able to fill the RMD demand originating from the hospitals assigned to it in full. The demand variance in the base case is set equal to half of the mean demand. The number of breakpoints in the piecewise linearization is increased incrementally until the relative optimality gap becomes less than 0.001. The test problem has 285 integer variables, 375 continuous variables, 370 linear constraints, 75 conical constraints, and 15 SOS2 sets. Testing is performed on a workstation with *Intel Core-i7* processor of the 4th generation and 8GB of RAM. The problem is coded on *Matlab2013b* and the MISOCP problem is solved using *CPLEX12.6*.

Table 3.4: Distance between demand points in the SND problem

i	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	120	55	75	300	57	220	44	26	228	199	47	116	79	107
2		180	103	188	118	113	85	138	127	197	99	29	48	66
3			123	359	107	279	103	53	287	255	101	195	135	180
4				283	26	203	34	72	216	118	22	119	56	45
5					298	150	264	317	105	403	278	184	231	252
6						219	49	56	232	134	25	135	72	72
7							185	238	45	255	199	115	153	139
8								62	193	181	22	101	35	69
9									250	204	50	153	93	112
10										308	207	114	161	182
11											166	234	178	152
12												116	53	59
13													69	90
14														38

The results are shown in Table 3.5. The column entitled 'solution' presents the best solution obtained. Opened SCs are shown outside the parenthesis, followed by the size of the SC, the required resources, and the hospitals assigned to it, respectively, inside the parenthesis and separated by semicolons. The total cost, in dollars, corresponding to this solution is shown in the column 'TC'. The last column show the computational time in seconds.

In addition to the base case (scenario 1), we test on different scenarios to study the effect of changing the problem parameters. Scenarios 2 and 3 pertain to strong and no economies of scale, attained by changing the exponent of the size-dependent capacity cost function to 0.6 and 1, respectively. Scenario 4 is based on the premise that the capacity cost is negligible (*i.e.*, set to zero), a case realized when the hospitals have surplus areas to install sterilization facilities. Scenario 5 is when the RMD demand is assumed deterministic, *i.e.*, $\sigma_{ip}^2 = 0$, an extreme case for low-variability demand. Finally, Scenario 6 represents demand with a Poisson distribution, which can be tightly approximated using a normal distribution with $\sigma_{ip}^2 = \mu_{ip},\ \forall\, i \in I,\ p \in P$.

By comparing the results of scenarios 1, 2 and 3, one can see that changing the exponent of the size-dependent capacity cost, an indicator of the extent of economies of scale, has not led to a change in the solution. As the exponent decreases, the effect of economies

Table 3.5: Results for the full consolidation scheme in the SND problem

scenario	solution	TC	cpu
1	2(290;8,6,2;2,13,14,15),3(139;4,3,1;3),5(127;4,3,1;5),7(104;3,2,1;7,10), 11(60;2,1,1;11),12(518;15,10,3;1,4,6,8,9,12)	1,275,648	1174
2	2(290;8,6,2;2,13,14,15),3(139;4,3,1;3),5(127;4,3,1;5),10(104;3,2,1;7,10), 11(60;2,1,1;11),12(518;15,10,3;1,4,6,8,9,12)	1,078,248	918
3	2(290;8,6,2;2,13,14,15),3(139;4,3,1;3),5(127;4,3,1;5),7(104;3,2,1;7,10), 11(60;2,1,1;11),12(518;15,10,3;1,4,6,8,9,12)	1,635,197	31
4	2(274;8,5,2;2,13,14),3(139;4,3,1;3),5(127;4,3,1;5),9(311;9,6,2;1,9), 10(104;3,2,1;7,10),11(60;2,1,1;11),12(139;4,3,1;6,8,12),15(145;4,3,1;4,15)	1,006,996	3
5	4(163;5,3,1;4,6,15),8(311;9,6,2;1,8,12,14),9(141;4,3,1;3,9),10(161;5,3,1;5,7,10), 11(44;1,1,1;11),13(220;6,4,2;2,13)	1,149,900	3
6	2(290;8,6,2;2,13,14),3(179;5,4,1;3,9),5(133;4,3,1;5),10(110;3,2,1;7,10), 11(63;2,1,1;11),12(522;15,10,3;1,4,6,8,12,15)	1,312,455	6,696

of scale becomes stronger, supposedly leading to more centralization. However, with the smaller exponent the magnitude of the capacity cost becomes smaller as well, favoring a less consolidated structure, which balances-off the economies of scale effect. When the capacity cost is neglected altogether in scenario 4, the number of opened SCs doubled. Still, not all hospitals opened their own SCs, proving that there is value for demand consolidation even with free capacity due to the risk pooling effect and the better utilization of resources explained earlier.

The effect of demand variability is observed from scenarios 5 and 6. The opened SCs and the assignment of hospitals to them remains unchanged when the demand variance is doubled. However, the safety stock of RMDs is increased to meet the higher demand variability, leading to higher inventory, capacity and resources costs as well. On the other hand, the solution obtained when the demand is assumed deterministic has the same number of opened SCs as the base case but with different opened SCs and assignments. Moreover, as expected, the stock level decreased, and consequently the total cost is significantly lower than the base case. The cost difference between the two scenarios can be thought of as 'the cost of uncertainty'.

We note that the computational time varies tremendously between the different scenarios, ranging between 3 seconds and about 2 hours. It is clear that as the non-linearity of the problem increases it becomes harder to solve. For larger problems one can add valid cuts suitable for MISOCP problems such as polymatroids [15] to enhance the computational performance.

Next, we take a closer look at the cost breakdown in scenarios 1 and 4 to spot the sources of cost saving explained in section 3.3 across the three organization schemes. Figure 3.2 illustrates the cost breakdown in scenario 1. It can be noticed that a saving of 19.1% can be achieved by shifting from a decentralized sterilization functions scheme to a partially centralized one (*i.e.*, scheme 2). This saving is primarily due to the lower capacity and resources costs outweighing the additional transportation cost. An additional 1.8% cost saving is achievable if the SCs maintain centralized RMD stocks due to the risk-pooling effect. Health care administrators might consider this saving trivial against the inconveniences related to losing their control over the RMD stocks, or they might opt for the fully centralized scheme 3 to take advantage of it.

When the capacity cost is assumed negligible in scenario 4, one would expect that the value of consolidation mostly diminishes. Surprisingly, as seen in Fig. 3.3, there is still ample room for saving amounting to 13.8% of the original (scheme 1) total cost. This saving is attributed merely to the better utilization of the resources, especially the human resources, in the partially-centralized scheme 2. As with the first scenario, an additional

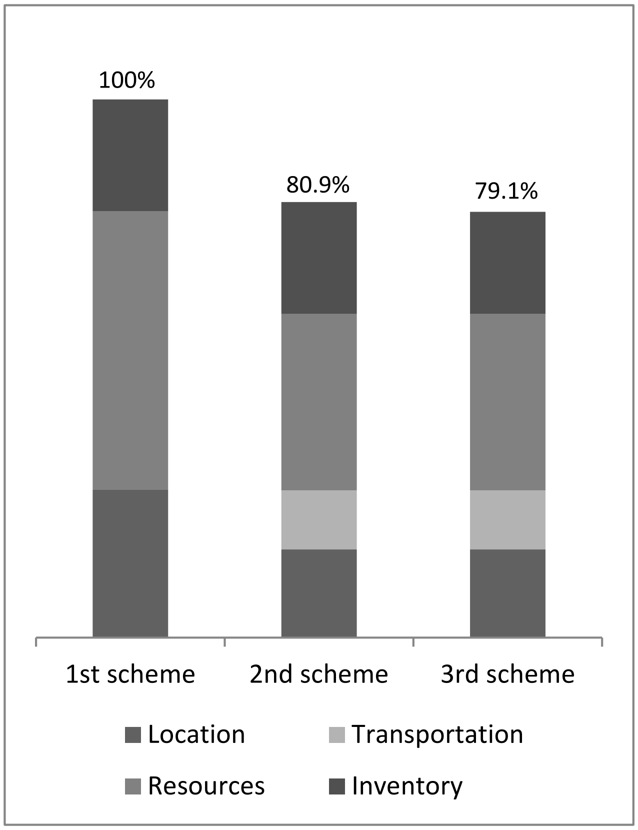

Figure 3.2: Cost breakdown for the base case scenario in the SND problem

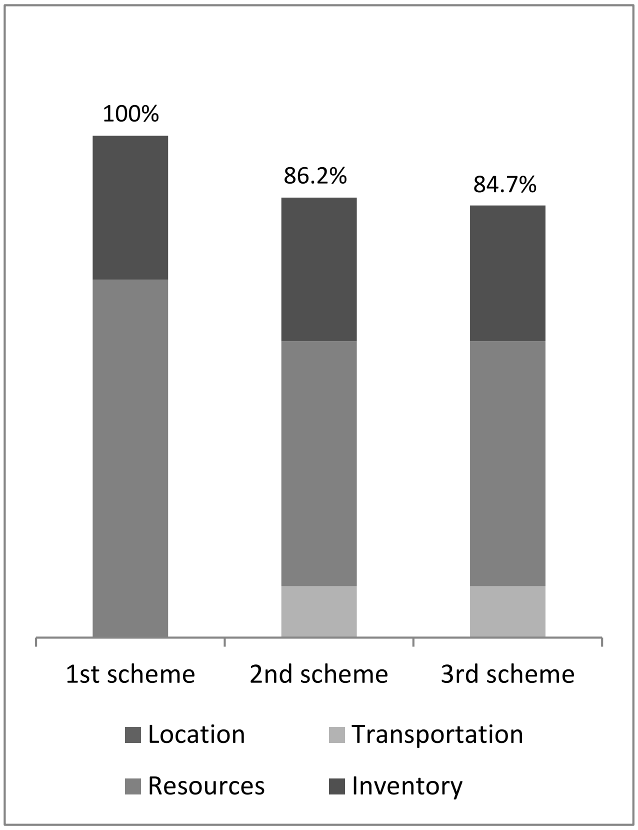

Figure 3.3: Cost breakdown for the zero capacity cost scenario in the SND problem

1.5% cost saving is realized from risk-pooling if scheme 3 is selected.

3.7 Conclusions

In this chapter, we present a new approach that can be used by a medical sterilization service provider to determine the optimal configuration of its network. This includes the number, location, and size of the SCs to open, the assignment of hospitals to them, the human and machinery resources and the stock of RMDs to be held in every SC. Our model aims to minimize the long-run total cost of the network while taking into account economies of scale, the risk pooling effect, and the stochastic nature of demand. The problem formulation results is a mixed-integer concave minimization problem with concave constraints. We transform the concave constraints into second-order cone constraints and use piecewise linearization to linearize the concave cost function, turning the problem into a mixed-integer second-order cone program that is solved using a commercial solver.

We used our approach to compare three schemes, ranging from a fully decentralized structure to a fully consolidated one where SCs not only preform the centralization services centrally, but they also administer the RMD stock for the hospitals. The proposed sterilization network design approach is implemented on a realistic case study from southwestern Ontario. The results obtained highlight the trade-offs embedded in the network design process and demonstrate the cost saving the can be achieved through centralization. It has been shown that significant saving is accomplished by pooling the sterilization function, whereas centralizing the inventory holding function results in a small cost saving that has to be weighed against other operational and legal considerations.

Conclusions and future directions

Economies of scale and risk pooling are inherent features in many application problems, often leading to concave minimization problems over a polyhedron. In this thesis, we focus on this class of problems both from an algorithmic and an application perspective. The contributions are two-fold.

On the one hand, we proposed new solution approaches for an important class of problems that frequently appears in supply chain design models, namely, minimizing a set of concave functions defined over affine combinations of continuous or integer decision variables with linear constraints. The first is a Lagrangian decomposition approach that allows the best Lagrangian multiplies to be calculated directly from the problem parameters, enabling the best Lagrangian bound and a high quality solution to be obtained in a single iteration. The second is a Benders approach that is well-suited for problems with a few concave variables. The concave terms are isolated in a low-dimensional master problem that can be solved efficiently through iterative enumeration.

On the other hand, we developed novel supply chain network design models for two important applications: cold supply chains and medical sterilization networks. These models incorporate more real-life requirements and less restrictive assumptions than those used in classical models. For the cold supply chain design problem, we proposed a hybrid solution method that combines the Lagrangian approach we presented earlier with a simulation-optimization approach to address the case of generic demand distribution and inventory policy. For the sterilization network design problem we reformulated the resulting mixed-integer concave minimization problem with concave constraints into a piecewise-linearized mixed integer problem with cone constraints that can be solved efficiently. For both problems, we tested on realistic case studies and drawn important managerial insights.

Moving forward, we would like to augment the solution approaches proposed in this thesis and to tackle more applications with them. The Benders approach proposed in Chapter 1 is a promising alternative that we want to investigate further. We are going to test it on problems with multiple concave terms to assess its performance. Moreover, we will try to implement a Logic-based Benders approach to generate Benders cuts from the integer subproblem directly instead of using its linear relaxation within a B&B framework. Logic-based Benders was shown to outperform classical Benders decomposition for mixed-integer linear problems [63], and we want to see if the same holds for mixed-integer concave minimization problem.

Bibliography

[1] MLF environment. http://www.mapleleafsustainability.ca/environment.html. Accessed: 2015-07-28.

[2] Neos server. http://www.neos-server.org/neos/.

[3] Ontario's Local Health Integration Networks. http://www.lhins.on.ca/. Accessed: 2015-09-22.

[4] Walmart. http://corporate.walmart.com/global-responsibility/environment-sustainability/greenhouse-gas-emissions. Accessed: 2015-07-28.

[5] *Air conditioning refrigerating data book*. The American Society of Refrigerating Engineers, 6th edition, 1956.

[6] *Radiative Forcing of Climate Change: Expanding the Concept and Addressing Uncertainties*. The National Academies Press, 2005.

[7] *Safeguarding the Ozone Layer and the Global Climate System*. Cambridge University Press, Cambridge, United Kingdom, 2005.

[8] Managing supply chain greenhouse gas emissions: Lessons learned for the road ahead. Technical report, Environmental Protection Agency, USA, 2010.

[9] Annual report on vaccine safety in ontario, 2012. Technical report, Ontario Agency for Health Protection and Promotion (Public Health Ontario), Toronto, ON: Queen's Printer for Ontario, 2014.

[10] Y. Acar, S. N. Kadipasaoglu, and J. M. Day. Incorporating uncertainty in optimal decision making: Integrating mixed integer programming and simulation to solve combinatorial problems. *Comp. & Ind. Engg.*, 56(1):106 – 112, 2009.

[11] N. Adhya, M. Tawarmalani, and N. V. Sahinidis. A Lagrangian approach to the pooling problem. *Industrial & Engineering Chemistry Research*, 38(5):1956–1972, 1999.

[12] F. A. Al-Khayyal and H. D. Sherali. On finitely terminating branch-and-bound algorithms for some global optimization problems. *SIAM J. on Optimization*, 10(4):1049–1057, August 1999.

[13] C. Almeder, M. Preusser, and R. F. Hartl. Simulation and optimization of supply chains: alternative or complementary approaches? *OR Spectrum*, 31(1):95–119, 2009.

[14] A. Atamtürk, G. Berenguer, and Z. (Max) Shen. A conic integer programming approach to stochastic joint location-inventory problems. *Operations Research*, 60(2):366–381, 2012.

[15] A. Atamtürk and V. Narayanan. Polymatroids and mean-risk minimization in discrete optimization. *Operations Research Letters*, 36(5):618 – 622, 2008.

[16] E. M. Beale and J. J. Forrest. Global optimization using special ordered sets. *Mathematical Programming*, 10(1):52–69, 1976.

[17] J. E. Beasley. Or-library: distributing test problems by electronic mail. *Journal of the Operational Research Society*, 41(11):1069–1072, 1990.

[18] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wachter. Branching and bounds tighteningtechniques for non-convex minlp. 24(4-5):597–634, August 2009.

[19] A. Ben-Tal, G. Eiger, and V. Gershovitz. Global minimization by reducing the duality gap. *Mathematical Programming*, 63(1-3):193–212, 1994.

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

[21] S. Benjaafar, Yanzhi L., and M. Daskin. Carbon footprint and the management of supply chains: Insights from simple models. *IEEE Trans. Autom. Sci. Engg.*, 10(1):99–116, Jan 2013.

[22] H.P. Benson. Separable concave minimization via partial outer approximation and branch and bound. *Operations Research Letters*, 9(6):389 – 394, 1990.

[23] O. Berman, D. Krass, and M. Mahdi Tajbakhsh. On the benefits of risk pooling in inventory management. *Production and Operations Management*, 20(1):57–71, 2011.

[24] O. Berman, D. Krass, and M. B. C. Menezes. Directed assignment vs. customer choice in location inventory models. Working Paper (2015) – Kedge School of Business.

[25] O. Berman, D. Krass, and M. M. Tajbakhsh. A coordinated location-inventory model. *Eur. J. Oper. Res.*, 217(3):500–508, 2012.

[26] Y. Bin and H Jun. An analysis on green supply chain management in E-commerce under the economic globalization. In *Bus. Intelligence and Financial Engg. Int. Conf.*, pages 595–599, 2009.

[27] A. D. Bojarski, J. M. Lainez, A. Espuña, and L. Puigjaner. Incorporating environmental impacts and regulations in a holistic supply chains modeling: An LCA approach. *Comp. & Chem. Engg*, 33(10):1747 – 1759, 2009.

[28] U.S. Census Bureau. Incorporated places and minor civil divisions datasets: Subcounty resident population, 2012.

[29] L. R. Burns and J. A. Lee. Hospital purchasing alliances: Utilization, services, and performance. *Health Care Management Review*, 33:203–215, 2008.

[30] A. Chaabane, A. Ramudhin, and M. Paquet. Design of sustainable supply chains under the emission trading scheme. *Int. Jour. of Prod. Econ.*, 135(1):37–49, 2012.

[31] M. Chen and C. Lin. Effects of centralization on expected costs in a multi-location newsboy problem. *The Journal of the Operational Research Society*, 40(6):597–602, June 1989.

[32] M. Cherikh. On the effect of centralization on expected profits in a multi-location newsboy problem. *The Journal of the Operational Research Society*, 51(6):755761, June 2000.

[33] CIHI. National health expenditure trends, 1975 to 2012. Technical report, Canada Institute of Health Information, 2012.

[34] J. Current, H. Min, and D. Schilling. Multiobjective analysis of facility location decisions. *Eur. J. Oper. Res.*, 49(3):295 – 307, 1990.

[35] A. Dasçi and V. Verter. The plant location and technology acquisition problem. *IIE Transactions*, 33(11):963–974, 2001.

[36] M. S. Daskin. *Network and discrete location: models, algorithms, and applications.* Wiley-Interscience, 1 edition, 1995.

[37] M. S. Daskin, C. R. Coullard, and Z. M. (Max) Shen. An inventory-location model: Formulation, solution algorithm and computational results. *Annals of Operations Research*, 110(1):83–106, 2002.

[38] A. Diabat and D. Simchi-Levi. A carbon-capped supply chain network problem. In *IEEE Int. Conf. Indust. Engg. and Engg. Mgmt.*, pages 523–527, 2009.

[39] L. Dupont. Branch and bound algorithm for a facility location problem with concave site dependent costs. *International Journal of Production Economics*, 112(1):245 – 254, 2008.

[40] M. Dür and R. Horst. Lagrange duality and partitioning techniques in nonconvex global optimization. *Journal of Optimization Theory and Applications*, 95(2):347–369, 1997.

[41] R. Ehrhardt. The power approximation for computing (s, S) inventory policies. *Management Sci.*, 25(8):777–786, 1979.

[42] R. Ehrhardt. Analytic approximations for (s, S) inventory policy operating characteristics. *Nav. Res. Logist. Q*, 28(2):255–266, 1981.

[43] S. Elhedhli and R. Merrick. Green supply chain network design to reduce carbon emissions. *Transportation Res. C*, 17(5):370 – 379, 2012.

[44] A.N. Elshafei. An approach to locational analysis. *Journal of the Operational Research Society*, 26:167–181, 1975.

[45] G. Eppen. Effects of centralization on expected costs in a multi-location newsboy problem. *Management Sci.*, 25(5):498–501, 1979.

[46] I. E. Essoussi and P. Ladet. Towards resource pooling in cooperative health care networks: Case of medical supply centralization. In *Computers Industrial Engineering, 2009. CIE 2009. International Conference on*, pages 600–605, July 2009.

[47] J. Falk. Lagrange multipliers and nonconvex programs. *SIAM Journal on Control*, 7(4):534–545, 1969.

[48] J. E. Falk and R. M. Soland. An algotithm for separable nonconvex programming problems. *Management Science*, 15(9):550–569, 1969.

[49] E. Feldman, F. A. Lehrer, and T. L. Ray. Warehouse location under continuous economies of scale. *Management Sci.*, 12(9):670–684, 1966.

[50] S. J. Fineman and A. S. Kapadia. An analysis of the logistics of supplying and processing sterilized items in hospitals. *Computers & Operations Research*, 5(1):47 – 54, 1978.

[51] S.K. Fischer, P.J. Hughes, P.D. Fairchild, C.L. Kusik, J.T. Dieckmann, E.M. McMahon, and N. Hobday. *Energy and global warming impacts of CFC alternative technologies.* Dec 1991.

[52] C. A. Floudas. *Detreministic Global Optimization: Theory, Methods and Applications.* Kluwer Academic Publisher, 2000.

[53] C. A. Floudas and P. M. Pardolas. *A collection of test problems for constrained global optimization algorithms.* Springer-Verlag, 1990.

[54] C.A. Floudas and C.E. Gounaris. A review of recent advances in global optimization. *Journal of Global Optimization*, 45(1):3–38, 2009.

[55] A. M. Geoffrion and G. W. Graves. Multicommodity distribution system design by benders decomposition. *Management Sci.*, 20(5):822–844, 1974.

[56] A.M. Geoffrion. Generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, 1972.

[57] P. Glasserman and D. D. Yao. Some guidelines and guarantees for common random numbers. *Management Sci.*, 38(6):884–908, Jun 1992.

[58] M. Guignard and S. Kim. Lagrangean decomposition: A model yielding stronger Lagrangean bounds. *Mathematical Programming*, 39:215–228, 1987.

[59] G. M. Guisewite and P. M. Pardalos. Minimum concave-cost network flow problems: Applications, complexity, and algorithms. *Annals of Operations Research*, 25(1):75–99, 1990.

[60] R. C. Gunther. *Refrigeration, air conditioning, and cold storage.* Chilton Book Company, Philadelphia, 2nd edition, 1969.

[61] I. Harris, C. Mumford, and M. Naim. The multi-objective uncapacitated facility location problem for green logistics. In *IEEE congress Evolut. Comput.*, pages 2732–2739, 2009.

[62] K. Hoffman. A method for globally minimizing concave functions over convex sets. *Mathematical Programming*, 20(1):22–32, 1981.

[63] J.N. Hooker. Logic-based benders decomposition. *Mathematical Programming*, 96:2003, 1995.

[64] R. Horst. An algorithm for nonconvex programming problems. *Mathematical Programming*, 10(1):312–321, 1976.

[65] R. Horst, N. V. Thoai, and H. P. Benson. Concave minimization via conical partitions and polyhedral outer approximation. *Mathematical Programming*, 50(1-3):259–274, 1991.

[66] R. Horst and H. Tuy. *Global Optimization: Deterministic approaches.* Springer-Verlag, second edition, 1992.

[67] Y. A. Huang, C. L. Weber, and H. S. Matthews. Categorization of scope 3 emissions for streamlined enterprise carbon footprinting. *Environ. Sci. Technol.*, 43(22):8509–8515, 2009.

[68] J. Y. Jung, G. Blau, J. F. Pekny, G. V. Reklaitis, and D. Eversdyk. A simulation based optimization approach to supply chain management under demand uncertainty. *Comp. & Chem. Engg.*, 28(10):2087 – 2106, 2004.

[69] R. Karuppiah and I. E. Grossmann. A Lagrangean based branch-and-cut algorithm for global optimization of nonconvex mixed-integer nonlinear programs with decomposable structures. *Journal of Global Optimization*, 41(2):163–186, 2008.

[70] A. Khajavirad and J. J. Michalek. A deterministic Lagrangian-based global optimization approach for quasiseperable non-convex mixed-integer nonlinear programs. *Journal of Mechanical Design*, 131:1–8, 2009.

[71] T. Kuno and T. Utsunomiya. A Lagrangian based branch-and-bound algorithm for production-transportation problems. *Journal of Global Optimization*, 18(1):59–73, 2000.

[72] Y. H. Lee, S. H. Kim, and C. Moon. Production-distribution planning in supply chain using a hybrid approach. *Prod. Plan. & Cont.*, 13(1):35–46, 2002.

[73] F. Lega. Strategies for multi-hospital networks: a framework. *Health Services Management Research*, 18(2):86–99, 2005.

[74] W. Lim, J. Ou, and C. Teo. Inventory cost effect of consolidating several one-warehouse multiretailer systems. *Oper. Res.*, 51(4):668–672, 2003.

[75] O. L. Mangasarian. Characterization of linear complementarity problems as linear programs. volume 7 of *Mathematical Programming Studies*, pages 74–87. 1978.

[76] M.T. Melo, S. Nickel, and F. Saldanha da Gama. Facility location and supply chain management a review. *Eur. J. Oper. Res.*, 196(2):401 – 412, 2009.

[77] S. Moon. Application of generalized benders decomposition to a nonlinear distribution system design problem. *Nav. Res. Logistics*, 36(3):283–295, 1989.

[78] A. Nikolopoulou and M. G. Ierapetritou. Hybrid simulation based optimization approach for supply chain management. *Comp. & Chem. Engg.*, 47:183 – 193, 2012.

[79] I. Or and W. P. Pierskalla. A transportation location-allocation model for regional blood banking. *A I I E Transactions*, 11(2):86–95, 1979.

[80] J. Ou. Economic lot sizing with constant capacities and concave inventory costs. *Naval Research Logistics (NRL)*, 59(7):497–501, 2012.

[81] M. P. Pardalos, H. E. Romeijn, and H. Tuy. Recent developments and trends in global optimization. *Journal of Computational and Applied Mathematics*, 124(12):209 – 228, 2000.

[82] P. M. Pardalos and G. Schnitger. Checking local optimality in constrained quadratic programming is np-hard. *Operations Research Letters*, 7(1):33 – 35, 1988.

[83] M. Raghavachari. On connections between zero-one integer programming and concave programming under linear constraints. *Operations Research*, 17(4):680–684, 1969.

[84] A. Ramudhin, A. Chaabane, M. Kharoune, and M. Paquet. Carbon market sensitive green supply chain network design. In *IEEE Int. Conf. Indust. Engg. and Engg. Mgmt.*, pages 1093–1097, 2008.

[85] N. Sahay and M. Ierapetritou. Hybrid simulation based optimization framework for centralized and decentralized supply chains. *Ind. & Engg. Chem. Res.*, 53(10):3996–4007, 2014.

[86] H. Schneider and J. L. Ringuest. Power approximation for computing (s, S) policies using service level. *Management Sci.*, 36(7):822–834, 1990.

[87] J. P. Shectman and N. V. Sahinidis. A finite algorithm for global minimization of separable concave programs. *Journal of Global Optimization*, 12(1):1–36, 1998.

[88] Z. J. (Max) Shen. A multi-commodity supply chain design problem. *IIE Trans.*, 37(8):753 – 762, Aug 2005.

[89] Z. J. (Max) Shen. Integrated stochastic supply-chain design models. *Computing in Science and Engg.*, 9(2):50–59, March 2007.

[90] J. Shu, C. P. Teo, and Z. J. (Max) Shen. Stochastic transportation-inventory network design problem. *Oper. Res.*, 53(1):48–60, 2005.

[91] C. Smellie, M. Quinlan, A. Dueck, and D. Effa. Steripro operations analysis. Case study, University of Waterloo.

[92] R. M. Soland. Optimal facility location with concave costs. *Operations Research*, 22(3):373–382, 1974.

[93] S. Solomon, Q. Qin, M. Manning, Z. Chen, M. Marquis, K. B. Averyt, M. Tignor, and H. L. Miller, editors. *Climate Change 2007: The Physical Science Basis.* Cambridge University Press, Cambridge, UK, 2007.

[94] M. Tawarmalani and N. V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103:225–249, 2005.

[95] M. Tawarmalani and N.V. Sahinidis. Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical Programming*, 99(3):563–591, 2004.

[96] C. Teo, J. Ou, and M. Goh. Impact on inventory costs with consolidation of distribution centers. *IIE Trans.*, 33(2):99–110, 2001.

[97] T. V. Thieu. A note on the solution of bilinear programming problems by reduction to concave minimization. *Mathematical Programming*, 41(1-3):249–260, 1988.

[98] H. Tlahig, A. Jebali, and H. Bouchriha. A two-phased approach for the centralisation versus decentralisation of the hospital sterilisation service department. *European J. of Industrial Engineering*, 3(2):227–246, 2009.

[99] H. Tlahig, A. Jebali, H. Bouchriha, and P. Ladet. Centralized versus distributed sterilization service: A locationallocation decision model. *Operations Research for Health Care*, 2(4):75 – 85, 2013.

[100] TMR. Sterilization equipment and disinfectants market - global industry analysis, size, share, growth, trends and forecast, 2013 - 2019. Technical report, Transparency Market Research, 2012.

[101] H. Tuy. Concave programming under linear constraints. *Soviet Math*, 5:1437–1440, 1964.

[102] J. van de Klundert, P. Muls, and M. Schadd. Optimizing sterilization logistics in hospitals. *Health Care Management Science*, 11(1):23–33, 2008.

[103] P. T. Vanberkel, R. J. Boucherie, E. W. Hans, J. L. Hurink, and Nelly Litvak. Efficiency evaluation for pooling resources in health care. *OR Spectrum*, 34(2):371–390, 2012.

[104] G. M. Velders, D. W. Fahey, J. S. Daniel, M. McFarland, and S. O. Andersen. The large contribution of projected HFC emissions to future climate forcing. *Proc. Natl. Acad. Sci.*, 2009.

[105] V. Verter. Uncapacitated and capacitated facility location problems. In H. A. Eiselt and Vladimir Marianov, editors, *Foundations of Location Analysis*. Springer, US, 2011.

[106] H. M. Wagner, M. O'Hagan, and B. Lundh. An empirical study of exactly and approximately optimal inventory policies. *Management Sci.*, 11(7):690–723, 1965.

[107] F. Wang, X. Lai, and N. Shi. A multi-objective optimization for green supply chain network design. *Decision Support Systems*, 51(2):262 – 269, 2011.

[108] P. Zipkin. *Foundations of Inventory Management*. McGraw-Hill, 1 edition, 2000.