# ROBUST BINARY LINEAR PROGRAMMING UNDER IMPLEMENTATION UNCERTAINTY

A Dissertation

by

JOSE ERNESTO RAMIREZ CALDERON

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

DOCTOR OF PHILOSOPHY

| | |
|---|---|
| Chair of Committee, | V. Jorge Leon |
| Co-Chair of Committee, | F. Barry Lawrence |
| Committee Members, | Amarnath Banerjee |
| | Erick Moreno-Centeno |
| Head of Department, | Mark Lawley |

May 2018

Major Subject: Industrial Engineering

ABSTRACT

This dissertation focuses on binary linear programming problems (BLP) affected by uncertainties preventing the implementation of the solutions exactly as prescribed. This type of uncertainty is termed implementation uncertainty and occurs due to model fidelity limitations.

This dissertation presents a model of binary variables under implementation uncertainty and develops a methodology to solve BLPs under this type of uncertainty consisting in a robust formulation (RBIU). The RBIU identifies solutions that satisfy given levels of optimality and feasibility for any realizations of the uncertainty. A solution methodology of the RBIU consists of an equivalent linear programming model.

Robust solutions tend to be conservative in the sense that they sacrifice optimality to achieve the given level of feasibility. This dissertation presents two methodologies to control the conservatism of the RBIU solutions. The first method consists in controlling the feasibility relaxation level and selecting the solutions bounding the value of the objective function. The second method is an extension of a well-known method in the field of robust optimization and consists in the development of a cardinality-constrained robust BLP under implementation uncertainty (CBIU) that controls the conservatism by bounding the maximum number of variables under uncertainty with different implemented and prescribed values. The proposed concepts of robustness are applied to the knapsack problem, assignment problem and shortest path problem (SPP) under implementation uncertainty to identify their solutions immune to uncertainty and to show how particular problem structures permit to identify different important theoretical and practical properties. This work examines the properties of the robust counterparts including configurations of the control parameters, complexity and the development of solutions algorithms.

This dissertation includes experimental studies to show how the proposed concepts of robustness permit to solve BLPs under implementation uncertainty and identify solutions protected against this type of uncertainty. The results of the experiments illustrate the sensitivity of the deterministic solutions to implementation uncertainty, the performance of the proposed solution methods and the different levels of the conservatism of the robust solutions. A case study involving information of a distribution company illustrates the application of the SPP under implementation uncertainty to a real problem.

DEDICATION

To my wife, our little angel, my parents and my siblings for their unconditional support.

## ACKNOWLEDGMENTS

Foremost, I want to thank God for his uncountable blessings and for letting me pursue my dream.

My most sincere appreciation to Dr. Leon for his incredible support throughout my graduate program. The completion of this work would not have been possible without Dr. Leon's wise guidance and patience. I am also grateful to my other committee members, Dr. Barry Lawrence, Dr. Erick Moreno and Dr. Amarnath Banerjee. All have contributed significantly to my academic growth and their advise has lifted the level of this research.

I am very thankful with the Fulbright-LASPAU program for supporting me; I am very proud of being part of their selected group. I would like to also extend my gratitude to the Industrial Distribution Program and the Global Supply Chain Laboratory, especially to Dr. Esther Rodriguez and Dr. Barry Lawrence, for contributing enormously to my professional development. I am also thankful to my professors, colleagues and friends at the Industrial and Systems Engineering Department, Dr. Adolfo Escobedo, Dr. Daniel Jornada, Dr. Jorge Samayoa and Erin Roady; all of them made my experience more valuable.

Last but not least, I must thank the most important people in my life, my family. This work would not have been possible without their support. To my parents, Hector and Francis, and my siblings, Cildri and Luis, who have supported me since the first moment I decided to pursue this goal. I have no words to express my gratitude to my wife, Ana Lucia, who has walked with me throughout this path; she has been my inspiration and motivation, my strength in the moments of weakness, and my relief during the hard times. Finally, thanks to our baby Jacob for being my motivation during the final, and the most complicated, stage of this experience.

May God bless y'all and may the force be with you.

CONTRIBUTORS AND FUNDING SOURCES

## Contributors

This work was supported by the dissertation committee consisting of Dr. Jorge Leon as the academic advisor, Dr. Amarnath Banerjee and Erick Moreno-Centeno of the Department of Industrial and Systems Engineering, and Dr. Barry Lawrence of the Department of Engineering Technologies and Industrial Distribution.

The data analyzed for Chapter 5 was provided by Texas A&M's Global Supply Chain Laboratory.

All other work conducted for the dissertation was completed by the student independently.

## Funding Sources

Graduate study was supported by a Graduate Assistant position at the Department of Engineering Technologies and Industrial Distribution from Texas A&M University.

NOMENCLATURE

| | |
|---|---|
| AP | Assignment Problem |
| BLP | Binary Linear Programming Problem |
| CBIU | Cardinality-constrained Robust Binary Linear Programming Under Implementation Uncertainty |
| CRKP | Cardinality-constrained Robust Knapsack Problem Under Implementation Uncertainty |
| CRPM | Cardinality-constrained Robust Minimum Weighted Bipartite Perfect Matching Problem Under Implementation Uncertainty |
| CRSPP | Cardinality-constrained Robust Shortest Path Problem Under Implementation Uncertainty |
| DP | Dynamic Programming |
| KP | Knapsack Problem |
| MILP | Mixed-Integer Linear Programming Problem |
| PM | Minimum Weighted Bipartite Perfect Matching Problem |
| RBIU | Robust Binary Linear Programing Under Implementation Uncertainty |
| RBMP | Equivalent Mixed-Integer Linear Reformulation of a Robust Binary Linear Programming Under Implementation Uncertainty |
| RKP | Robust Knapsack Problem Under Implementation Uncertainty |
| RPM | Robust Minimum Weighted Bipartite Perfect Matching Problem Under Implementation Uncertainty |

| | |
|---|---|
| RSPP | Robust Shortest Path Problem Under Implementation Uncertainty |
| SPP | Shortest Path Problem |

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

# 1. INTRODUCTION AND LITERATURE REVIEW

This chapter discusses the motivations, contributions and research questions focus of this dissertation. It also presents a review of the related literature and introduces basic notation common through all the subsequent chapters. The chapter ends detailing the organization of this dissertation.

## 1.1 Motivation

Optimization problems may be impacted by uncertainties in the values of the model parameters or the implemented values of the decision variables. Data or parameter uncertainty refers to the case when data take values different to the nominal values considering during the modeling phase. The second type of uncertainty, here denoted as *implementation uncertainty*, refers to the type of uncertainty that prevents the implementation of the solutions exactly as computed. Implementation uncertainty occurs due to a number of reasons including model fidelity limitations result of unknown or intrinsic characteristics of the modeled system, estimation errors, unavailable information, lack of precision or simply limited time available to model the problem in hand.

Uncertainties may impact the level of feasibility or optimality of the solutions of optimization problems. For instance, in the knapsack problem, the use of estimates of the values of the capacity and profits during the modeling phase may produce changes in their values at the time of implementation leading to a solution that does not remain feasible or reduces the total profit significantly. On the other hand, the implementation may take too much time, and some of the prescribed items may not be available anymore and cannot be selected leading to a reduction of the total profit; similarly, changes in policies or priorities may force the selection of items not prescribed leading an excess of the given capacity making the prescribed solution infeasible.

There may exist situations when feasibility cannot be guaranteed due to uncertainty, for instance, when the problem contains equality constraints. It may be possible that the decision maker is willing to accept a certain level of feasibility relaxation because he assumes that certain level of infeasibility is manageable or produces meaningful solutions. For instance, in the knapsack problem, the decision maker may consider that the cost of increasing the given capacity is smaller than the improvement in the profit. Similarly, in an assignment problem, one may be willing to accept that more than one task is assigned to a resource such that all the tasks can be completed, even if that requires more time for completion. It is of interest to find solutions of optimization problems satisfying certain levels of optimality and feasibility when impacted by uncertainty.

Different approaches have been developed to protect the solutions of optimization problems against uncertainties; for instance, stochastic optimization (e.g. Dantzig, 1955; Beale, 1955; Wets, 1966, 1974, 1983) and robust optimization (e.g. Soyster, 1973; Mulvey et al., 1995; Bertsimas and Sim, 2004; Ben-Tal et al., 2009). Stochastic optimization uses random variables to model uncertainty and seeks for solutions that remain optimal and feasible with high probability. Characteristics of stochastic optimization include that we should know the associated probability distributions of the uncertain elements, or at least one should be able to identify the family of distributions to which the true one belongs. Additionally, in stochastic optimization there may exist realizations of the uncertainty where the given levels of optimality or feasibility are not satisfied (see Ben-Tal et al., 2009).

On the other hand, robust optimization seeks for solutions that satisfy given levels of optimality and feasibility for any realization of the uncertainty; these solutions are termed robust solutions (Mulvey et al., 1995). Literature in robust optimization is mostly focused in data uncertainty (see Soyster, 1973; Kouvelis and Yu, 1997; Ben-Tal et al., 2009; Bertsimas et al., 2011; Gabrel et al., 2014). In contrast, only a few work consider implementation uncertainty explicitly in part because in some cases data and implementation uncertainty have been proven equivalent (see Ben-Tal and Nemirovski, 2002; Ben-Tal et al., 2009). The existing models of implementation uncertainty account for real variables only and extending these models to binary variables may not be possible.

To the best of our knowledge, there does not exist previous work in the field of robust optimization studying binary linear programming problems (BLP) under implementation uncertainty nor applications of this type of problem. Existing work in the field of optimization focuses on the knapsack problem (KP) under data uncertainty only (e.g. Steinberg and Parks, 1979; Yu, 1996; Bertsimas and Sim, 2003) and does not consider the issue that options may not be available at the time of implementation or options not initially prescribed may be forced to be selected, for instance, as a result of policy changes. Similarly, in the field of robust optimization, the assignment problem (AP) has been addressed when weights are uncertain (e.g. Kouvelis and Yu, 1997; Katriel et al., 2008) and do not account for unexpected changes in the prescribed assignments at the time of the implementation. Finally, the shortest path problem (SPP) has been studied in robust optimization when impacted by parameter uncertainty only (e.g. Kouvelis and Yu, 1997; Yu and Yang, 1998; Bertsimas and Sim, 2003).

Motivated by the theoretical significance of BLPs under implementation uncertainty and the practical importance of solutions immune to this type of uncertainty in applications of these type of problems, this dissertation aims to model implementation uncertainty in binary variables and develop a methodology to solve BLPs under this type of uncertainty.

## 1.2 Research Objectives and Contributions

This section describes the research objectives and contributions of this dissertation to the body of knowledge.

### 1.2.1 Research Objective

The main objective of this dissertation is to develop a methodology to solve BLPs under implementation uncertainty. SSpecifically, this research aims to (i) present a model of implementation uncertainty in binary variables; (ii) formulate a BLP under implementation uncertainty; (iii) develop a methodology to solve BLPs under this type of uncertainty consisting in a robust formulation (RBIU); (iv) develop methodologies to address the conservatism of the solutions of the RBIU; (v) apply the proposed concepts of robustness to well-known BLPs; and (vi) identify the characteristics of the problems that make the robust solutions more attractive.

### 1.2.2 Contribution of the Research

The main contributions of this dissertation to the academic body of knowledge are the model of implementation uncertainty in binary variables and the development of a methodology to solve BLPs under this type of uncertainty. This dissertation accomplishes these contributions by studying the following research questions:

(a) How can implementation uncertainty be modeled in binary variables?

(b) How can a BLP under implementation uncertainty be formulated?

(c) How can a BLP under implementation uncertainty be solved?

(d) How can the conservatism of the RBIU be addressed?

(e) How do the proposed concepts of robustness apply to well-known BLPs under implementation uncertainty and what are the properties of these particular problems?

(f) What are the characteristics of the problem that make robust solutions more attractive?

Question (a) is associated with one of the main contributions of this research and is presented in Chapter 2. This dissertation presents a model of implementation uncertainty in binary variables that represents the existing uncertainty in the implemented value of the decision variables. Binary variables under this type of uncertainty are termed uncertain variables, and variables not impacted by uncertainty are termed certain variables. Although this work focuses on linear programming, this model of implementation uncertainty holds for nonlinear programming models as well. The proposed model of uncertainty allows the development of a model of the solutions space by defining an uncertain set containing all possible implementation vectors due to uncertainty. Properties of the solution space permit the development of the properties of the robust formulation and robust solutions.

Question (b), dealt in Chapter 2, is motivated by the impact of implementation uncertainty in the solutions of optimization problems. This work develops a robust formulation for BLPs under implementation uncertainty (RBIU). The RBIU is founded by the definitions of measures of objective and feasibility robustness level. The measure of objective robustness computes for the worst case value of the objective function, and the measure of feasibility robustness computes for the maximum deviation from feasibility. The RBIU seeks for the solution that minimizes the worst case value of the objective function while satisfying the given level of feasibility for all realizations of the uncertainty. Properties of the RBIU include nonlinearity and the existence of multiple solutions.

Question (c) is due to the nonlinearity and complexity of the RBIU. Chapter 2 presents a solution methodology to the RBIU consisting of a linear reformulation (RMBP). The RMBP is a mixed-integer linear programming problem (MILP) with the number of binary variables equals to the total number of certain variables. Particular problem structures may permit the development of algorithms to solve the RBIU using other solution methods such as dynamic programming; Chapters 3 and 5 show algorithms to solve the robust counterparts of the KP and the SPP, respectively.

Question (d) aims to address the characteristic of robust solutions that tend to be conservative in the sense that they sacrifice optimality to satisfy the level of feasibility (e.g.

Ben-Tal and Nemirovski, 1998; Bertsimas and Sim, 2004). A very conservative solution provide higher protection of the feasibility and degradation of the objective value concerning the given optimization problem; however, the objective value produced by a conservative solution may be too high, assuming a minimization problem. On the other hand, a less conservative solution may produce a better objective function value but it may sacrifice the level of feasibility, or its objective value may degrade when impacted by uncertainty. For instance, consider the knapsack problem representing the loading of a vehicle with the most valuable cargo; if there exists the possibility that some items not initially selected may be included later due to changes in their priority, a conservative solution may be more attractive because it allows the inclusion of more high priority items although the value of the cargo is reduced. For the same problem, a too conservative solution, such as the worst case scenario, may have a low probability to occur, or may never happen, leading to too much empty space in the vehicle and a significant reduction on the value of the cargo; a less conservative solution may be more attractive since it improves the utilization of the space and the total value of the selected items. One may be interested in controlling the level of the conservatism depending on the characteristics of the problem in hand.

Solutions of the RBIU are not exempt from conservatism, and this dissertation provides several methods for reducing it. The first methodology consists of the control of the feasibility relaxation level and the selection of the solutions bounding the value of the objective function. Feasibility relaxation may reduce conservatism by improving the worst case value of the objective function, while the selection of the bounding solutions permits to control the level of conservatism among the set of robust solutions obtained from the resolution of the RBIU with a fixed feasibility level.

The second method consists in the development of a cardinality-constrained robust formulation for a BLP under implementation uncertainty (CBIU). This formulation is based on the model of parameter uncertainty developed by Bertsimas and Sim (2004).

The CBIU includes a control parameter to bound the maximum number of uncertain variables impacted by uncertainty. The CBIU also incorporates a control parameter to allow a feasibility relaxation; the simultaneous use of the two control parameters may result in a possible higher reduction of the level of conservatism. This dissertation provides a solution methodology of the CBIU in the form of an equivalent MILP. The CBIU assumes that the number of uncertain variables with different prescribed and implemented values is less than or equal to the maximum value defined by the control parameter; in reality, there may exist more variables with different implemented values. When the value of the control parameter is exceeded, the CBIU does not guarantee that the given levels of feasibility or optimality are satisfied. Feasibility relaxation permits a certain level of infeasibility, and the RBIU guarantees that such level of infeasibility is protected and it will not be violated; in contrast, CBIU cannot protect the given level of feasibility relaxation but permits a higher reduction of the conservatism. This work estimates the probability that the CBIU loses protection against uncertainty due to a violation of the assumption of the maximum number of variables impacted by uncertainty. The CBIU for BLPs with particular problem structures is examined.

Question (e) comprises the application of the proposed concepts of robustness to well-known BLPs. The first problem corresponds to the KP under implementation uncertainty. This work applies the proposed concepts of robustness to develop the robust KP under implementation uncertainty (RKP). The structural properties of the RKP permit the use dynamic programming to identify robust solutions. Chapter 3 also presents the cardinality-constrained robust KP under implementation uncertainty (CRKP), its equivalent linear reformulation and probability bounds.

The second application is presented in Chapter 4 and consists of the AP under implementation uncertainty. This problem is studied in the context of the equivalent minimum weight bipartite perfect matching problem (PM). The proposed concepts of robustness are used to develop the robust AP/PM under implementation uncertainty (RPM). Different configurations of the control parameters of the RPM permit the identification of conditions for the existence of robust solutions and configurations to generate solutions with different characteristics, such as solutions defining matchings or connected bipartite graphs. The corresponding cardinality-constrained robust formulation (CRPM) is demonstrated to be equivalent to the RPM.

Question (e) is completed in Chapter 5 by applying the proposed concepts of robustness to the shortest path problem (SPP) under implementation uncertainty (RSPP). The structural properties of the RSPP permit the development of a solution algorithm based on Dijkstra's algorithm for the SPP making possible to solve the RSPP using dynamic programming instead of mathematical programming methods. Similarly to the AP, the cardinality-constrained robust formulation is shown to be equivalent to the RSPP and can be treated as such.

Question (f) is associated with the results of the experimental studies included in this dissertation for each application. The experimental studies aim to show how the proposed concepts of robustness permit to identify solutions to BLPs under implementation uncertainty that are protected against this type of uncertainty. The results of these experimental studies demonstrate the sensitivity of the deterministic solutions against implementation uncertainty in terms of their feasibility level. These results also show that robust solutions tend to sacrifice the objective function value to guarantee the feasibility level; the higher the sensitivity of the problem to uncertainty the more significant the degeneration of optimality. Based on the performance of the deterministic solutions, these experimental studies uncover conditions of the problems that make the robust solutions more attractive for each application. Some of these conditions are tied to real problem applications; moreover, the SPP under implementation uncertainty is illustrated in the context of a real problem involving transportation operations of a distribution company. The results provide alternative routes and recommendations of when to use the robust solutions based on the average annual profit produced by the deterministic and robust solutions.

## 1.3 Literature Review

This section presents an overview of the existing literature on robust optimization, and the KP, the AP and the SPP under uncertainty.

### 1.3.1 Background on Robust Optimization

The study of optimization under uncertainty dates back to the establishment of modern decision theory in the 1950s with the development of the first stochastic optimization models. For instance, Dantzig (1955) presents a special class of two-stage linear programming problems in which the decisions in the first stage are made to meet the uncertain demands with known distribution occurring in the second stage. Similarly, Beale (1955) study the minimization of convex functions with coefficients of the constraints given by random variables. Recent work in the field of stochastic optimization include Collet and Rennard (2007); Duchi et al. (2011); Kingma and Ba (2014), among others. The interested reader is referred to Birge (1997); Birge and Louveaux (2011); Uryasev and Pardalos (2013) for more information of stochastic optimization models and solution algorithms.

Robust optimization provides a different approach for optimization under uncertainty. Unlike stochastic optimization, robust optimization provides solutions that are feasible and optimal for any realization of uncertainty. Soyster (1973) presents the first known work in robust optimization; the author's model produces solutions that are feasible for any realization of the data within convex sets. Mulvey et al. (1995) introduced the term *robust optimization* for the first time and termed *robust solutions* to solutions that satisfy the given levels of optimality and feasibility. The authors provide a model that identify robust solutions for all the scenarios of the input data; Kouvelis and Yu (1997) extends these concepts of robustness to discrete optimization models.

A characteristic of robust solutions is that they tend to be conservative in the sense that they sacrifice optimality to satisfy the given level of feasibility. For instance, the model in Soyster (1973) is considered too conservative from this perspective (Bertsimas and Sim, 2004). Different authors have addressed this issue by modeling uncertainty using different representations. For instance, El Ghaoui and Lebret (1997); El Ghaoui et al. (1998); Ben-Tal and Nemirovski (1998, 1999, 2002) propose a less conservative model by using ellipsoidal sets to describe data uncertainty; Bertsimas and Sim (2003, 2004) control conservatism by bounding the maximum number of uncertain coefficients changing in each constraint simultaneously; and (Kouvelis and Yu, 1997) address this issue by proposing less conservative measures of robustness for data uncertainty.

In the context of discrete optimization, robust optimization models have been developed to account for data uncertainty only. Kouvelis and Yu (1997) develop a robust approach for discrete optimization by seeking solutions that minimize the worst case value of the objective function within a set of scenarios. Lin et al. (2004) developed robust formulations for MILPs with bounded data uncertainty. Bertsimas and Sim (2003) develop a cardinality-constrained robust formulation for discrete optimization and network flow problems with uncertain data, and Chen and Lin (2010) propose an algorithm to solve the robust design of network flow problems with uncertain demand. Li et al. (2011) present a comprehensive description of these work.

Work in the field of robust optimization accounting for implementation uncertainty can be found in Ben-Tal and Nemirovski (2000, 2002); Ben-Tal et al. (2009). Ben-Tal et al. denote this type of uncertainty as *implementation errors* and provide two forms of modeling it on real decision variables; furthermore, they show that these forms of implementation errors are equivalent to artificial data uncertainties and can be treated as such. Jornada and Leon (2016) apply these forms of implementation uncertainty to real variables in the context of multi-objective optimization problems.

The interested reader is referred to Beyer and Sendhoff (2007); Ben-Tal et al. (2009); Bertsimas et al. (2011); Gabrel et al. (2014) for comprehensive studies of robust optimization models.

### 1.3.2 Background on the Knapsack Problem

The KP is a well-known problem in the field of integer programming. This problem consists of a set of items possessing weights and profits; the objective is to select a subset of items such that the total profit is maximum and the total weight of the selected items does not exceed a given capacity (e.g. Bertsimas and Weismantel, 2005). Early publications of this problem include Dantzig (1957) and Bellman (1957). Applications of the KP include the capital budget problem (e.g. Weingartner and Ness, 1967), and the loading of a plane or ship with the most valuable cargo given the capacity constraint (e.g. Shih, 1979). Bartholdi III (2008) presents other applications of this problem.

Work of the KP under uncertainty includes the work in Steinberg and Parks (1979) studying the KP with random profits and develop a dynamic solution methodology replacing the real-values by a preference ordering of the distributions of the selected items. Similarly, Henig (1990) develop a methodology combining dynamic programming and a search procedure to solve the KP where the items possess normally distributed profits. On the other hand, Yu (1996) presents the first known work of robust optimization applied to the KP where a set of scenarios define the profits. Bertsimas and Sim (2003, 2004) study this problem where the profits are given and the weights are independently distributed following a symmetric distribution; the authors develop a cardinality-constrained robust formulation to control the maximum number of items with uncertain weights.

The interested reader is referred to Kouvelis and Yu (1997); Schrijver (1998); Wolsey (1998); Bazaraa et al. (2011) for further information about the KP, solution algorithms, applications of this problem and the KP under uncertainty.

### 1.3.3 Background on the Assignment Problem

The AP is described as the problem of assigning exactly one resource for every job and every resource performs exactly one job; the objective is to find the assignments with the minimum total weight (Bazaraa et al., 2011). Other versions of the AP include the quadratic assignment problem (e.g. Loiola et al., 2007) and the semi assignment problem (e.g. Kennington and Wang, 1992). Applications of the AP include personnel selection, vehicle assignment and scheduling problems (e.g. Kuhn, 1955; Fisher and Jaikumar, 1981).

The weights associated with every assignment defines an instance of the AP; the majority of the work on this problem under uncertainty focuses on the case where the weights are uncertain. For instance, Katriel et al. (2008) use a stochastic optimization approach to address the changes in the nominal value of the weights. On the other hand, Kouvelis and Yu (1997); Aissi et al. (2005); Deı et al. (2006) use robust optimization to deal with the value of the weights belonging to a set of scenarios. Further information on this topic can be found in Kasperski (2008); Aissi et al. (2009); Kasperski and Zieliński (2016).

### 1.3.4 Background on the Shortest Path Problem

The SPP is a well-known problem in the optimization and graph theory fields. Given a digraph with each arc possessing a nonnegative weight, the SPP consists of finding the path from a source node to a destination node with the minimum total weight. Early publications of this problem include Kruskal (1956) discussing the problem of finding the shortest spanning subtree on a graph; Ford Jr (1956) presenting the network flow problem, and Loberman and Weinberger (1957) studying the problem of connecting terminals with the minimum wire length. The first solutions algorithms for this problem are presented in Bellman (1958), Dijkstra (1959) and Dantzig (1960); Glover et al. (1985a,b) develop a more efficient algorithm, and Fredman and Tarjan (1987) present an improved implementation of Dijkstra's algorithm using Fibonacci heaps.

Applications of the SPP include vehicle routing problems (e.g. Desrochers and Soumis, 1988; Toth and Vigo, 2002; Feillet et al., 2004), driving directions on GPS systems (e.g. Shibuya, 2002; Goldberg and Harrelson, 2005), design of railway networks (e.g. Nachtigall, 1995; Schulz et al., 2002), machine learning (e.g. Borgwardt and Kriegel, 2005; Bunescu and Mooney, 2005), and telecommunications among others (e.g. Montemanni et al., 2004). The interested reader is referred to Deo and Pang (1984); Cherkassky et al. (1996) for further information and classification of the shortest path problems and solution algorithms.

The SPP under parameter uncertainty has been studied in stochastic programming considering that weights of the arcs follow a random distribution (e.g. Polychronopoulos et al., 1993), or change probabilistically according to a Markov chain (e.g. Orda et al., 1993). Further stochastic programming models and solution algorithms for the SPP under parameter uncertainty can be found in Bertsekas and Tsitsiklis (1991); Ji (2005); Hutson and Shier (2009). In the field of robust optimization Kouvelis and Yu (1997); Yu and Yang (1998) present a robust formulation for the SPP when the weight of the arcs belong to a set of different scenarios. The authors show that the problem is NP-complete for more restricted layered networks of width 2 and 2 scenarios; Yu and Yang developed a pseudo-polynomial to solve the problem in general networks with a bounded number of scenarios. Montemanni and Gambardella (2004) present an exact algorithm for Kouvelis and Yu's robust shortest path model by considering that the uncertain weights belong to a given interval; Montemanni et al. (2004) uses a branch and bound methodology to improve this algorithm. Bertsimas and Sim (2003) develop a cardinality-constrained robust SPP with the weights belonging to a given interval. Vincke (1999); Roy (2005, 2010); Gabrel et al. (2013) present other robust models and solutions algorithms for the SPP with weights belonging to an interval or to a set of scenarios.

## 1.4 Preliminaries and Notation

This section presents preliminaries and introduces basic notation used through all the chapters of this dissertation.

Let $x \in \mathbb{R}^n$ be a binary decision vector with $n$ binary decision variables $x_i, i = 1, ..., n$. Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a function defined as $f(x) = \sum_{i=1}^{n} c_i x_i$, with $c_i \in \mathbb{R}, \forall i$. Let $g_j : \mathbb{R}^n \rightarrow \mathbb{R}$ be a function defined as $g_j(x) = \sum_{i=1}^{n} a_{ij} x_i$ with $a_{ij} \in \mathbb{R}, \forall i, j = 1, ..., m$ defining the left-hand-side, and $b_j \in \mathbb{R}$ defining the right-hand-side of the $j$-th constraint $\forall j$.

Let $X$ be the feasible set defined as $X = \{x \in \mathbb{R}^n : g_j(x) \leq b_j, \forall j; x_i \in \{0, 1\}, \forall i\}$. A BLP can be formulated as follows:

$$\min_{x} \{f(x) : x \in X\} . \tag{1.1}$$

A prescribed vector $\hat{x}$ denotes an instance of $x$ formed by the given prescribed values $\hat{x}_i$ of the variables $x_i, \forall i$. Similarly, $\tilde{x}$ denotes the implemented vector formed by the implemented values $\tilde{x}_i, \forall i$.

Variables under implementation uncertainty are termed *uncertain variables* and variables not affected by uncertainty are termed *certain variables*. The following assumptions are considered through all this dissertation:

**Assumption 1.4.1.** *It is known which variables are uncertain variables.*

**Assumption 1.4.2.** *There exists at least one certain variable.*

Through all this dissertation, the original BLP formulation is denoted as the *deterministic model* or the *deterministic BLP*. Similarly, the solutions of the deterministic BLP are denoted as the *deterministic solutions*.

## 1.5 Organization of the Dissertation

This dissertation is organized in chapters as follows.

Chapter 2 presents the proposed model of implementation uncertainty in binary variables. This model of permits the development of a model of the solution space containing all the possible realizations of the implementation uncertainty; the solution space permits to identify essential properties of the robust formulation and robust solutions. This chapter introduces the methodology to solve BLPs under implementation uncertainty. This methodology consists in the formulation a robust BLP under this type of uncertainty that seeks for solutions satisfying the given levels of optimality and feasibility. The corresponding measures of robustness define the levels of optimality and feasibility of the robust solutions. This chapter also presents a linear reformulation of the RBIU that permits to identify robust solutions by applying existing algorithms to solve MILPs. This dissertation proposes a methodology consisting of the control of the feasibility relaxation level and the selection of the solutions bounding the objective function value that allows the decision maker to control the level of conservatism of the robust solutions. Conservatism is also addressed with a cardinality-constrained methodology that permits the decision maker to control the conservatism through the control of the maximum number of the uncertain variables impacted by uncertainty and the control of the feasibility relaxation level.

Chapter 3 focuses in the KP under implementation uncertainty. The proposed concepts of robustness are applied to develop a robust KP under implementation uncertainty. Similarly, the proposed solution is applied to identify the solutions of the RKP. Moreover, the particular structural properties of this problem permit the use dynamic programming to identify the robust solutions. This chapter also shows the cardinality-constrained KP under implementation uncertainty and its equivalent linear reformulation. The performance of the deterministic solutions of the KP and robust solutions of the RKP and CRKP is evaluated in terms of their objective value and feasibility level. Similarly, the dynamic programming and MILP solution method are compared in terms of their running time. This chapter includes a discussion of the characteristics of the KP that make deterministic solutions more sensitive to uncertainty and robust solutions more attractive.

Chapter 4 presents the AP under implementation uncertainty. This chapter describes the equivalent PM and applies the proposed concepts of robustness to formulate the robust PM under implementation uncertainty. Different configurations of the RPM permit to identify conditions for the existence of robust solutions and solutions that define matchings or connected bipartite graphs. The cardinality-constrained PM under implementation uncertainty and the RPM are equivalent due to the structural properties of the PM. The experimental study evaluates the performance of the deterministic and robust solutions in terms of the objective function value and feasibility level when producing matchings and connected bipartite graphs. The experiments also illustrate the difference in the lower and upper bound of the objective function value as a control of the level of conservatism.

Chapter 5 is devoted to the SPP under implementation uncertainty. The robust SPP under this type of uncertainty is presented. Different configurations of the control parameters permit to identify properties of the RSPP including its equivalence to a reduced SPP free of uncertainty. This chapter presents a solution algorithm of the RSPP based on Dijkstra's algorithm for SPP. The cardinality-constrained robust SPP under implementation uncertainty is shown to be equivalent to the RSPP and can be treated as such. An experimental study aims to evaluate the performance of the deterministic and robust solutions and the performance of the solutions methodologies. Recommendations on the conditions that make the robust solutions more attractive are included. This chapter ends presenting a case study for a real application of the SPP under implementation uncertainty.

Chapter 6 presents the conclusions and summary of the contributions of this dissertation, as well a discussion of related future research and extensions.

# 2. ROBUST BINARY LINEAR PROGRAMMING UNDER IMPLEMENTATION UNCERTAINTY

## 2.1 Introduction

This chapter presents a model of implementation uncertainty in binary variables and a methodology to solve BLPs under this type of uncertainty. This chapter includes the development of a solution methodology to the RBIU and methodologies to address the conservatism of the robust solutions.

Implementation uncertainty refers to the type of uncertainty that prevents the implementation of the prescribed solution exactly as computed. In other words, there exists a possibility that the implemented values of the decision variables are different to the prescribed ones. Implementation uncertainty occurs due model fidelity limitations result of unknown or intrinsic characteristics of the modeled system, estimation errors, unavailable or limited information, lack of precision or limited time available to model the problem in hand. In the context of binary variables, implementation uncertainty can be interpreted as the possibility that the binary decisions in a prescribed solution switches to the opposite value when implemented. For instance, consider the solution of a facility location problem prescribing siting a natural gas electricity generation plant in city A, but at the time of the implementation new environmental regulations prohibit building this type of plant in this city; therefore, the implemented value has to switch to do not sit the plant in city A. This example also illustrates how data uncertainty could not straightforward be used to model the uncertainty at hand.

Uncertainties may affect the optimality and feasibility of the solutions of an optimization problem. One may ask, how can I protect the solutions of optimization of optimization problems against uncertainties? Robust optimization approaches seek to answer this question by identifying solutions that satisfy the given levels of optimality and feasibility for any realization of the uncertainty; such solutions are termed *robust solutions* (Mulvey et al., 1995). For instance, Soyster (1973) considers perturbations in the coefficients of the constraints using convex sets; the resulting model produces solutions that are feasible for any realization of the data within the convex sets.

Existing work in the field of robust optimization accounting for implementation uncertainty is very limited. For instance, Ben-Tal et al. (2009) propose two forms of modeling this type of uncertainty on real decision variables: 1) *additive implementation errors* refers to the case when a random value is added to the prescribed value, and 2) *multiplicative implementation error* refers to the case when the random value multiplies the prescribed value; furthermore, they show that these forms of implementation errors are equivalent to artificial data uncertainties and can be treated as such. However, extending these representations to binary variables may not be straightforward; for instance, when the prescribed value is zero, multiplicative implementation errors cannot model changes in the implemented values independent of the random value. This work presents a model of implementation uncertainty in binary variables that allows the development of a robust formulation for BLPs under this type of uncertainty.

The proposed model of implementation uncertainty in binary variables describes the existing uncertainty in the implemented value of this type of variable given its prescribed value. The proposed model consists of defining the probability that the implemented value is equal or different to the prescribed value; this probability is assumed to be unknown. Although this dissertation focuses on linear programming, this model of implementation uncertainty may hold for nonlinear binary problems as well.

The development of the RBIU is founded by measures of optimality and feasibility robustness levels. This work proposes a measure of the objective robustness level that seeks for the worst case value of the objective function among all the possible solutions result of implementation uncertainty, guaranteeing that the implemented solutions do not worsen the given value. This measure is conservative since it sacrifices too much optimality, and although there exist less conservative measures such as the maximum deviation from optimality or the maximum percentage deviation (Kouvelis and Yu, 1997), the use of the worst-case value enables a linear reformulation leading to a more tractable robust formulation. Similarly to Mulvey et al. (1995); Jornada and Leon (2016), this work proposes a measure of the feasibility robustness level that measures the maximum deviation from the feasible region when binary variables are impacted by implementation uncertainty. This measure of the feasibility robustness level permits to control of the conservatism of the robust solutions by controlling the maximum allowed deviation from feasibility.

The RBIU seeks solutions that minimize the worst case value of the objective function while satisfying the given deviation from the feasible region. Constraints forming the RBIU require searching for a maximum value among an exponential number of binary vectors, making the RBIU a nonlinear combinatorial problem. Moreover, this work demonstrates that RBIU is NP-Complete. This work also proves the existence of multiple solutions for the RBIU given by every vector within the uncertain set; this result supports the characteristic of robust formulations producing solutions that satisfy the given levels of optimality and feasibility for any realization of the uncertainty, and supports the development of a methodology to address the conservatism. Given the complexity of the RBIU, a solutions methodology to the RBIU is developed. This methodology consists in a linear reformulation of the robust feasible region. A characteristic of the RMBP is a reduction of the number of binary variables as a function of the number of variables under implementation uncertainty; the higher the number of uncertain variables the smaller the number of discrete variables in the RMBP.

The RBIU solutions are not exempt from conservatism and this chapter provides several methods to reduce it. This work develops a methodology to address conservatism by controlling the feasibility relaxation level and selecting the solutions bounding the value of the objective function. The RBIU allows the decision maker to relax the feasibility level leading to a possible reduction of the sacrifice of the optimality. A feasibility relaxation is possible through an increment in the maximum permitted deviation from feasibility. Another methodology to select the level of conservatism of the solutions consists in computing the solutions that produce the lower and upper bounds of the value of the objective function among the set of robust solutions. The lower bound solution is the least conservative, but its value may degrade when impacted by uncertainty; on the other hand, the upper bound solution is the most conservative by being the worst case value, but its value may improve when impacted by uncertainty. This dissertation provides another approach to address conservatism by developing a cardinality-constrained robust BLP under implementation uncertainty based on Bertsimas and Sim (2004). The CBIU contains two parameters to control the maximum number of uncertain variables that possess different prescribed and implemented values and to control the feasibility relaxation level. A solution methodology and properties of the CBIU are presented in this chapter.

The remainder of this chapter is organized as follows. Section 2.2 presents the model of implementation uncertainty for binary variables and the development of the solution space. Section 2.3 presents the development of the RBIU and its properties. Section 2.4 presents the development of the solution methodology. Section 2.5 presents the methodologies to address the conservatism of the roust solutions. Section 2.6 presents concluding remarks for this chapter.

## 2.2 Implementation Uncertainty in Binary Variables

Consider a binary variable $x$, an instance $\hat{x}_i$ of $x_i$ and its implemented value $\tilde{x}_i$. Let $P(\tilde{x}_i|\hat{x}_i)$ be the conditional probability that the implemented value of the decision variable $x_i$ is $\tilde{x}_i$ given that its prescribed value is $\hat{x}_i$.

Consider the following probabilities:

$$P(\tilde{x}_i = \hat{x}_i | \hat{x}_i = 0) = p_i, \qquad P(\tilde{x}_i = 1 - \hat{x}_i | \hat{x}_i = 0) = 1 - p_i$$
$$P(\tilde{x}_i = \hat{x}_i | \hat{x}_i = 1) = q_i, \qquad P(\tilde{x}_i = 1 - \hat{x}_i | \hat{x}_i = 1) = 1 - q_i, \tag{2.1}$$

with $0 \leq p_i, q_i \leq 1$, and probabilities $p_i$ and $q_i$ independent of each other $\forall i$. The probabilities $p_i$ and $q_i$ provide the probability of implementing the prescribed values, while their complements provide the probability that the implemented values are different than the prescribed values. A binary variable under implementation uncertainty may possess different prescribed and implemented values; an uncertain variable can be defined as follows:

**Definition 2.2.1.** *A binary variable $x_i$ is under implementation uncertainty if $p_i < 1$ or $q_i < 1$.*

The following assumption is considered through this dissertation:

**Assumption 2.2.1.** *For the uncertain variables the probabilities $p_i$ and $q_i$ are unknown.*

Without loss of generality, the decision vector $x$ is decomposed into two vectors $x_C$ and $x_U$, where $x_C$ is composed of the certain variables $x_1, ..., x_c$, and $x_U$ is composed of the uncertain variables $x_{c+1}, ..., x_n$; for convenience define $C = \{1, ..., c\}$ as the set of indices of the certain variables in $x_C$ with $c < n$, and $U = \{c+1, ..., n\}$ as the set of indices of uncertain variables in $x_U$.

There exist an *uncertain set* $\mathcal{U}(x_C)$ containing all the possible implementation vectors due to implementation uncertainty. This set models the solution space for the BLP under implementation uncertainty. The uncertain set is defined as follows:

**Definition 2.2.2.** *Given a decision vector $x = (x_C, x_U)^T$, the uncertain set is defined as $\mathcal{U}(x_C) = \big\{y = (y_C, y_U)^T : y_C = x_C; y_i \in \{0,1\}, \forall i\big\}$.*

The set $\mathcal{U}(x_C)$ possesses the following properties:

**Property 2.2.1.** $|\mathcal{U}(x_C)| = 2^{|U|} = 2^{(n-c)}$.

**Property 2.2.2.** $x \in \mathcal{U}(x_C)$.

**Property 2.2.3.** *Given binary vectors $x, y \in \mathbb{R}^n$, if $x_C = y_C$, then $\mathcal{U}(x_C) = \mathcal{U}(y_C)$.*

The first property states that there are $2^{|U|}$ vectors in $\mathcal{U}(x_C)$ because of the $2^{|U|}$ different outcomes of $x_U$. The second property states that $x$ may be implemented exactly as computed even when it is under implementation uncertainty. The third property states that vectors with the same certain values have equal uncertain sets.

### 2.3 Problem Formulation and Analysis

Consider the BLP in (1.1). A prescribed solution $\hat{x}$ for the BLP is assumed to be feasible and optimal; however, the implemented solution $\tilde{x}$ may be different due to implementation uncertainty. It can be observed that given the definition of $\mathcal{U}(x_C)$, $\tilde{x}$ may be infeasible or suboptimal because the uncertain variables can take any value at the time of the implementation. Therefore it is of interest to find solutions that guarantee certain levels of optimality and feasibility when impacted by implementation uncertainty. Before this can be done, it is necessary to formalize the concepts of the level of optimality and level of feasibility.

#### 2.3.1 Measures of Robustness

The objective robustness level, $\gamma : \mathbb{R}^n \to \mathbb{R}$, is a measure of the degree to which a solution is robust with respect to the objective function degradation when affected by implementation uncertainty. $\gamma$ is defined as follows:

**Definition 2.3.1.** *The objective robustness level $\gamma$ is defined as:*

$$\gamma(x) = \max_{y \in \mathcal{U}(x_C)} \{f(y)\}. \tag{2.2}$$

$\gamma$ is the worst case value of the objective function among all the elements of $\mathcal{U}(x_C)$ guaranteeing that the objective value of the implemented solution will not worsen when affected by implementation uncertainties. Similarly, the feasibility robustness level, $\delta_j : \mathbb{R}^n \to \mathbb{R}$, is a measure of the degree to which a solution is robust with respect to the violation of constraint $j$ when affected by implementation uncertainty. $\delta_j$ is defined as follows:

**Definition 2.3.2.** *The feasibility robustness level $\delta_j$ is defined as follows:*

$$\delta_j(x) = \max_{y \in \mathcal{U}(x_C)} \{g_j(y) - b_j, 0\}, \quad \forall j. \tag{2.3}$$

The value of $\delta_j$ is positive if there exists a vector $y \in \mathcal{U}(x_C)$ such that $g_j(y)$ exceeds $b_j$, and zero otherwise. The value of $\delta_j$ can be bounded by a given parameter $\delta_j^{max}$ such that a solution $x$ is feasibility robust if the excess of $g_j$ over $b_j$ is not greater than $\delta_j^{max}$ for all the elements in $\mathcal{U}(x_C)$. Appropriate scalarization factors may be used in (2.3) to make comparisons between different constraints significant in practice.

### 2.3.2 Robust Formulation of a BLP Under Implementation Uncertainty

Let $\mathcal{X}$ be a feasible region named *robust feasible region*; $\mathcal{X}$ is defined by the following set of constraints:

$$\max_{y \in \mathcal{U}(x_C)} \{f(y)\} \leq \gamma(x) \tag{2.4}$$

$$\max_{y \in \mathcal{U}(x_C)} \{g_j(y)\} - \delta_j(x) \leq b_j, \quad \forall j \tag{2.5}$$

$$\gamma(x) \text{ is unrestricted} \tag{2.6}$$

$$0 \leq \delta_j(x) \leq \delta_j^{max}, \quad \forall j \tag{2.7}$$

$$x_i \in \{0, 1\}, \quad \forall i. \tag{2.8}$$

Constraint (2.4) is named *objective robustness constraint* and it ensures that the Definition 2.3.1 is satisfied. The value of $\gamma$ is unrestricted (2.6) since it depends of the coefficients of the objective function $f$. Similarly, constraints (2.5) are named *feasibility robustness constraints*; they together with constraints (2.7) ensure that the Definition 2.3.2 is satisfied and the maximum excess of $g_j$ over $b_j$ is at most $\delta_j^{max}$. Constraints (2.8) are the binary constraints for the decision variables $x_i$.

It is possible now to formulate the *robust BLP under implementation uncertainty* (RBIU) as the following single objective mixed-binary optimization problem:

$$\min_{x} \{\gamma(x) : x, \gamma, \delta_1, ..., \delta_m \in \mathcal{X}\}. \tag{2.9}$$

Constraints (2.4) and (2.5) depend on the selected vector $x$, and require to search for the maximum value among an exponential number of binary vectors in $\mathcal{U}(x_C)$; this makes the constraints nonlinear, and therefore the RBIU is a nonlinear mixed-integer formulation. Claim 2.3.1 states the complexity of RBIU.

**Claim 2.3.1.** *The RBIU is NP-Complete.*

*Proof.* To prove Claim 2.3.1, it is necessary to prove that: 1) the RBIU is in NP, and 2) an NP-Complete problem can be polynomially reducible to the RBIU. These results are shown in Lemmas 2.3.1 and 2.3.2, respectively. Then Claim 2.3.1 holds. □

**Lemma 2.3.1.** *The RBIU is in NP.*

*Proof.* Given an instance of RBIU with parameters $c_i, a_{ij}, b_j$ and a solution $x^*, \gamma^*, \delta_j^*$ with $\delta_j^* \leq \delta_j^{max}, \forall j$, the verification of a *yes* answer takes polynomial time. Indeed, the verification of the expressions $\max_{\{y_i \in \mathcal{U}(x_C^*)\}} \left\{\sum_{i\in C} c_i x_i^* + \sum_{i \in U} c_i y_i\right\} \leq \gamma^*$ and $\max_{\{y_i \in \mathcal{U}(x_C^*)\}} \left\{\sum_{i\in C} a_{ij} x_i^* + \sum_{i \in U} a_{ij} y_i\right\} \leq b_j + \delta_j^*$ for all $j$ runs in $\mathcal{O}(mn^2)$ time. □

RBIU is reformulated and solved as a mixed-binary linear problem (RMBP) in Section 2.4. MILPs are considered NP-complete (see Gary and Johnson, 2002).

**Lemma 2.3.2.** *The RMBP is polynomially reducible to the RBIU (RMBP $\leq_P$ RBIU).*

*Proof.* To prove that the RMBP is polynomially reducible to the RBIU, it suffices to show that an instance of the RMBP can be solved as a instance of the RBIU. RBIU and RMBP are equivalent (see Theorem 2.4.1); therefore RMBP $\leq_P$ RBIU. □

These results complete the proof of Claim 2.3.1.

The remaining of this section presents structural properties of RBIU. Theorem 2.3.1 establishes the characteristics of the robust solutions.

**Theorem 2.3.1.** *Let $x^* = (x_C^*, x_U^*)$ be an optimal solution of the RBIU; then any vector $y \in \mathcal{U}(x_C^*)$ is also optimal for the RBIU.*

*Proof.* Let $y$ be any vector in $\mathcal{U}(x_C^*)$. From the Property 2.2.3, it follows that $\mathcal{U}(y_C) = \mathcal{U}(x_C^*)$. From Definitions 2.3.1 and 2.3.2, it follows that $\gamma(y) = \gamma(x^*)$ and $\delta_j(y) = \delta_j(x^*)$. Then, $y \in \mathcal{X}$ and yields the same value of $\gamma$. Therefore, every vector $y \in \mathcal{U}(x_C^*)$ is also optimal. □

The set $\mathcal{U}(x_C^*)$ is called here the *robust-optimal solution set*.

**Corollary 2.3.1.** *If $\mathcal{U}(x_C^*) \neq \emptyset$, then RBIU has at least $2^{|U|}$ optimal solutions.*

*Proof.* This result follows from Property 2.2.1 and Theorem 2.3.1. □

**Corollary 2.3.2.** *To find the robust-optimal solutions set, it suffices to find one solution in the set.*

*Proof.* This result follows from Theorem 2.3.1 and Definition 2.2.2. □

The proposed measures of robustness and RBIU consider a BLP where the objective is to minimize the function $f$ subject to inequality constraints of the form $g_j(x) \leq b_j$. When maximizing $f(x)$, it can be rewritten as the minimization of $-f(x)$ and $\max_x\{f(x)\} = -\min_x\{-f(x)\}$. Similarly, a constraint $g_j(x) \geq b_j$ can be rewritten as $-g_j(x) \leq -b_j$. On the other hand, the model proposed cannot directly handle equality constraints since $g_j(x) = b_j$ is infeasible whether $g_j$ exceeds $b_j$ or $b_j$ exceeds $g_j$. An equality constraint can be approached by representing it as to two inequalities, i.e. $g_j(x) \leq b_j$ and $-g_j(x) \leq -b_j$; then it is possible to write the corresponding feasibility robustness constraints as follows:

$$\max_{y \in \mathcal{U}(x_C)} \{g_j(y)\} - \delta_j^L \leq b_j \tag{2.10}$$

$$\max_{y \in \mathcal{U}(x_C)} \{-g_j(y)\} - \delta_j^G \leq -b_j, \tag{2.11}$$

with $0 \leq \delta_j^L \leq \delta_j^{Lmax}$ and $0 \leq \delta_j^G \leq \delta_j^{Gmax}$.

The variables $\delta_j^L$ and $\delta_j^G$ measure the infeasibility level of the equality constraint; variable $\delta_j^L$ measures the excess of $g_j$ over $b_j$; conversely, $\delta_j^G$ measures the excess of $b_j$ over $g_j$. When $\delta_j^L = 0$, RBIU produce solutions satisfying the constraint $g_j(x) \leq b_j$; similarly, when $\delta_j^G = 0$, RBIU produce solutions satisfying the constraint $g_j(x) \geq b_j$. Clearly, the equality is satisfied when $\delta_j^L = 0$ and $\delta_j^G = 0$ simultaneously.

## 2.4 Solution Methodology

This section presents the solution approach to solve RBIU using an equivalent mixed-binary linear formulation. Advantages of solving the linear formulation include: 1) a reduction in the number of binary variables, and 2) linearity. As will be discussed in detail later, the total number of binary variables is reduced because the equivalent problem only depends on the certain variables. Linearity is convenient because of the wealth of knowledge in the literature when compared to non-linear optimization models. Interested readers are referred to Wolsey (1998); Schrijver (1998) for a review on existing solution approaches.

Given a prescribed solution $\hat{x}$ for the BLP, the implemented value $\tilde{x}_i, i \in U$ of an uncertain variable may affect the feasibility of the $j$-th constraint if one of the following conditions is satisfied:

1. $\tilde{x}_i = 1$ given $\hat{x}_i = 0$ and $a_{ij} \geq 0$, or

2. $\tilde{x}_i = 0$ given $\hat{x}_i = 1$ and $a_{ij} < 0$.

These conditions describe an increment in the contribution of $a_{ij}x_i$ to the value of $g_j(x)$. Clearly, any increment in $g_j(x)$ may lead to $g_j(x) > b_j$ making the $j$-th constraint infeasible. An increment in the value of $f(x)$ can be described similarly.

**Lemma 2.4.1.** *Expressions* $(a_{ij} + |a_{ij}|)/2$ *and* $(c_i + |c_i|)/2$ *provide the maximum contribution of the terms* $a_{ij}x_i$ *and* $c_ix_i$ *to* $g_j(x)$ *and* $f(x)$ *for any value of* $x_i$*, respectively.*

*Proof.* If $a_{ij} \geq 0$, then $|a_{ij}| = a_{ij}$ and $(a_{ij} + |a_{ij}|)/2 = (a_{ij} + a_{ij})/2 = a_{ij} \geq a_{ij}x_i$. Similarly, if $a_{ij} < 0$, then $|a_{ij}| = -a_{ij}$ and $(a_{ij} + |a_{ij}|)/2 = (a_{ij} - a_{ij})/2 = 0 \geq a_{ij}x_i$. Therefore, $(a_{ij} + |a_{ij}|)/2 \geq a_{ij}x_i$ for any value of $x_i$. Similar proof can be done for the expression $(c_i + |c_i|)/2$. □

Result in Lemma 2.4.1 allow the linearization of the measures of robustness.

Let $\mathcal{X}'$ be a feasible region defined by the following constraints:

$$\sum_{i \in C} c_i x_i + \sum_{i \in U} \left( \frac{c_i + |c_i|}{2} \right) \leq \gamma(x) \tag{2.12}$$

$$\sum_{i \in C} a_{ij} x_i + \sum_{i \in U} \left( \frac{a_{ij} + |a_{ij}|}{2} \right) - \delta_j(x) \leq b_j, \quad \forall j \tag{2.13}$$

(2.6), (2.7), (2.8)

Constraint (2.12) is named *maximum-contribution objective robustness* constraint, and constraints (2.13) are named *maximum-constribution feasibility robustness* constraints. The RMBP can be formulated as follows:

$$\min_{x} \left\{ \gamma(x) : x, \gamma, \delta_1, ..., \delta_m \in \mathcal{X}' \right\}. \tag{2.14}$$

The RMBP formulation contains: 1) a single objective $\gamma$, 2) $n - |U| = |C|$ binary variables $x_i$, 3) one unconstrained variable $\gamma$, 4) $m$ nonnegative variables $\delta_j$ and 5) $2m+1$ constraints. The relation between the RBIU and RMBP is described in Theorem 2.4.1.

**Theorem 2.4.1.** *The RBIU and RMBP are equivalent.*

The proof of Theorem 2.4.1 consists of proving the equality of the sets of solutions of the RBIU and RMBP formulations. The following lemmas provide the necessary results to complete this proof.

**Lemma 2.4.2.** *Let $x^*$, $\gamma^*$ and $\delta_j^*, \forall j$ be a feasible solution of the RMBP. Then $x^*$, $\gamma^*$ and $\delta_j^*, \forall j$ is also a feasible solution of the RBIU.*

*Proof.* Given that $x^*$, $\gamma^*$ and $\delta_j^*, \forall j$ are feasible, they satisfy constraints (2.6), (2.7) and (2.8). By Lemma 2.4.1, $a_{ij} y_i \leq (a_{ij} + |a_{ij}|)/2$ and $c_i y_i \leq (c_i + |c_i|)/2$ for all $i \in U$, and by Property 2.2.3 $x_C^* = y_C$ for all $y \in \mathcal{U}(x_C^*)$. Then:

$$\begin{aligned}
\max_{y \in \mathcal{U}(x_C^*)} \left\{ \sum_{i \in C} c_i y_i + \sum_{i \in U} c_i y_i \right\} &= \sum_{i \in C} c_i x_i^* + \max_{y \in \mathcal{U}(x_C^*)} \left\{ \sum_{i \in U} c_i y_i \right\} \\
&\leq \sum_{i \in C} c_i x_i^* + \sum_{i \in U} \left( \frac{c_i + |c_i|}{2} \right) \leq \gamma^*, \\
\max_{y \in \mathcal{U}(x_C^*)} \left\{ \sum_{i \in C} a_{ij} y_i + \sum_{i \in U} a_{ij} y_i \right\} - \delta_j^* &= \sum_{i \in C} a_{ij} x_i^* + \max_{y \in \mathcal{U}(x_C^*)} \left\{ \sum_{i \in U} a_{ij} y_i \right\} - \delta_j^* \\
&\leq \sum_{i \in C} a_{ij} x_i^* + \sum_{i \in U} \left( \frac{a_{ij} + |a_{ij}|}{2} \right) - \delta_j^* \leq b_j, \forall j.
\end{aligned} \tag{2.15}$$

Therefore, $x^*$, $\gamma^*$ and $\delta_j^*, \forall j$ are a feasible solution of the RBIU. □

**Lemma 2.4.3.** *Let $x^*$, $\gamma^*$ and $\delta_j^*, \forall j$ be a feasible solution of the RBIU. Then $x^*$, $\gamma^*$ and $\delta_j^*, \forall j$ is also a feasible solution of the RMBP.*

*Proof.* Given that $x^*$, $\gamma^*$ and $\delta_j^*, \forall j$ are feasible, they satisfy constraints (2.6), (2.7) and (2.8). By Lemma 2.4.1, $\max_{y \in \mathcal{U}(x_C)} \left\{\sum_{i \in U} a_{ij} x_i\right\} = \sum_{i \in U} ((a_{ij} + |a_{ij}|)/2)$, and since $x^*$ and $\delta_j^*$ satisfy the constraints (2.5), $x^*$, $\gamma^*$ and $\delta_j^*$ satisfy the constraints $\sum_{i \in C} a_{ij} x_i^* + \sum_{i \in U} ((a_{ij} + |a_{ij}|)/2) - \delta_j^* \leq b_j, \forall j$. Therefore, $x^*$, $\gamma^*$ and $\delta_j^*, \forall j$ satisfy constraints (2.13). A proof that $x^*$, $\gamma^*$ and $\delta_j^*, \forall j$ satisfy constraint (2.12) follows a similar rationale. Therefore, $x^*$, $\gamma^*$ and $\delta_j^*, \forall j$ are feasible solution of the RMBP. □

**Lemma 2.4.4.** *Let $x^*$, $\gamma^*$ and $\delta_j^*, \forall j$ be an optimal solution of the RMBP. Then the optimal value $\gamma^* = \gamma(x^*)$ of RMBP is also the optimal value of the RBIU.*

*Proof.* Assume $\gamma^*$ is not the optimal value of the RBIU; then there should exist a different value $\gamma' = \gamma(x^*)$ such that $\gamma'$ is the optimal value of the RBIU and $\gamma' < \gamma^*$. By Lemma 2.4.3, $\gamma'$ satisfies constraint (2.12); however, $\gamma^*$ is the optimal value of the RMBP and $\gamma^*$ is the minimum value of the right-hand-side of the constraint (2.12). Then it follows that $\gamma^* \leq \gamma'$, which makes the initial assumption false. Therefore, $\gamma^*$ is the optimal value of the RBIU. □

**Lemma 2.4.5.** *Let $x^*$, $\gamma^*$ and $\delta_j^*, \forall j$ be an optimal solution of the RBIU. Then the optimal value $\gamma^* = \gamma(x^*)$ of the RBIU is also the optimal value of the RMBP.*

*Proof.* Assume $\gamma^*$ is not the optimal value of the RMBP; then there should exist a different value $\gamma' = \gamma(x^*)$ such that $\gamma'$ is the optimal value of the RMBP and $\gamma' < \gamma^*$. By Lemma 2.4.4, $\gamma'$ is also optimal for the RBIU. However, $\gamma^*$ is optimal for the RBIU and it follows that $\gamma^* \leq \gamma'$, which makes the initial assumption false. Therefore, $\gamma^*$ is optimal for the RMBP. □

Using the results in Lemmas 2.4.2, 2.4.3, 2.4.4 and 2.4.5, the proof of Theorem 2.4.1 is as follows:

*Proof.* Let $X^{'}$ be the set of solutions of the RBIU, and let $X^{''}$ be the set of solutions of the RMBP. By Lemma 2.4.2, a solution $x^{*}, \gamma^{*}, \delta_j^{*} \in X^{'}$ also belongs to $X^{''}$, then $X^{'} \subseteq X^{''}$. Similarly, by Lemma 2.4.3, a solution $x^{*}, \gamma^{*}, \delta_j^{*} \in X^{''}$ also belongs to $X^{'}$, then $X^{''} \subseteq X^{'}$. It follows that $X^{'} = X^{''}$, and by Lemmas 2.4.4 and 2.4.5 their optimal values coincide; therefore, the RBIU and RMBP are equivalent. □

**Corollary 2.4.1.** *Let $x^{*} = (x_C^{*}, x_U^{*})$ be an optimal solution of the RMBP. Then every vector $y \in \mathcal{U}(x_C^{*})$ is also an optimal solution of the RMBP.*

*Proof.* This result follows from Theorems 2.3.1 and 2.4.1. □

Note that the RMBP replaces all the terms corresponding to the uncertain variables with constant values given by the expressions for the maximum contribution in Lemma 2.4.1; therefore, the RMBP does not depend on the uncertain variables. When the number of uncertain variables $|U|$ is equal to $n$, the RMBP does not contain any decision variable and it is not possible to make any decision. Given the equivalence between the RBIU and RMBP, this paper assumes that the RBIU contains at least one certain variable, i.e. $1 \leq c < n$.

Appendix A presents a polynomial-time transformation of RBIU into RMBP.

### 2.5 Conservatism of the RBIU Solutions

Robust solutions tend to be conservative in the sense that they sacrifice optimality to satisfy the given level of feasibility (Ben-Tal and Nemirovski, 1998; Bertsimas and Sim, 2004). Solutions of the RBIU are not exempt from conservatism; this section presents several methodologies to reduce the level of conservatism of the robust solutions.

### 2.5.1 Feasibility Relaxation and Bounding Solutions Method

This section presents a methodology to address conservatism consisting in the control of the feasibility relaxation level and the selection of the solutions that bound the value of the objective function. The methodology to select the solutions providing the lower and upper bounds of the value of the objective function is presented next.

Theorem 2.3.1 shows that the RBIU produces a robust-optimal solution set $\mathcal{U}(x_C^*)$. Each solution within the robust-optimal solution set may produce a different objective function value. Although the value of each solution depends on the specific problem instance, this work shows a method to compute the solutions within the robust-optimal solution providing the maximum and minimum objective function value; i.e., the upper and lower bound values of the objective function. Propositions 2.5.1 and 2.5.2 provide these bounds for the value of $f(y)$, with $y \in \mathcal{U}(x_C^*)$.

**Proposition 2.5.1.** *Let $x^* = (x_C^*, x_U^*)$ be an optimal solution of the RBIU, and $x^{UB} = (x_C^*, x_U^{UB})$, where $x_i$ in $x_U^{UB}$ is set as follows:*

$$x_i^* = \begin{cases} 1, & \textit{if } c_i \geq 0 \\ 0, & \textit{if } c_i < 0. \end{cases} \tag{2.16}$$

*Then $f(x^{UB}) \geq f(y), \forall y \in \mathcal{U}(x_C^*)$; hence, $f(x^{UB})$ is an upper bound for $f(y), y \in \mathcal{U}(x_C^*)$.*

*Proof.* By expression (2.16), if $c_i \geq 0$ for $x_i$ in $x_U^{UB}$ then $x_i = 1$ and $c_i x_i = c_i \geq c_i y_i$; similarly, if $c_i < 0$ then $x_i = 0$ and $c_i x_i = 0 \geq c_i y_i$. Consider a solution $y \in \mathcal{U}(x_C^*)$ such that $x^{UB} \neq y$. Given that $x_C^* = y_C^*$, then $f(x^{UB}) = \sum_{x_i \in x^{UB}} c_i x_i = \sum_{x_i \in x_C^*} c_i x_i + \sum_{x_i \in x_U^{UB}} c_i x_i \geq \sum_{x_i \in x_C^*} c_i x_i + \sum_{y_i \in y_U} c_i y_i = \sum_{y_i \in y} c_i y_i = f(y)$. Therefore, $f(x^{UB}) \geq f(y), \forall y \in \mathcal{U}(x_C^*)$. $\square$

**Proposition 2.5.2.** *Let $x^* = (x_C^*, x_U^*)$ be an optimal solution of the RBIU, and $x^{LB} = (x_C^*, x_U^{LB})$, where $x_i$ in $x_U^{LB}$ is set as follows:*

$$x_i^* = \begin{cases} 0, & \text{if } c_i \geq 0 \\ 1, & \text{if } c_i < 0. \end{cases} \tag{2.17}$$

*Then $f(x^{LB}) \leq f(y), \forall y \in \mathcal{U}(x_C^*)$; hence, $f(x^{LB})$ is a lower bound for $f(y), y \in \mathcal{U}(x_C^*)$.*

*Proof.* Proof of Proposition 2.5.2 is similar to the proof of Proposition 2.5.1. □

The upper bound solution equals the worst case value of the objective function; this solution is the most conservative among the solutions within the robust-optimal solution set. The upper bound solution improves the value of $f$ for any implemented value of the uncertain variables. This solution can be considered as the *pessimistic case* where the decision maker believes there exists a high chance that uncertainties will occur during implementation; therefore, the decision maker seeks for more protection against any loss in the value of the objective function. On the other hand, the lower bound solution is the least conservative among the solutions within the robust-optimal solution set. The lower bound solution suffers degradation of the value of $f$ for any implemented value of the uncertain variables. This solution can be considered as the *optimistic case* where the decision maker believes there exists a low chance uncertainties will occur during implementation; therefore, the decision maker seeks for a better value of $f$ while taking the risk that it may deteriorate.

The decision maker controls the level of conservatism of the prescribed solution by computing the solutions described in Propositions 2.5.1 and 2.5.2.

**Claim 2.5.1.** *The upper and lower bounds can be computed in $\mathcal{O}(n)$ time.*

*Proof.* For every variable $x_i$ in the vector $x_U$, the decision consists in evaluating expressions (2.16) or (2.17). Therefore, the complete process requires exactly $|U|$ steps with $0 \leq |U| \leq n$. $\square$

Selecting the bounding solutions permits to control the level of conservatism among the robust-optimal solution set; however, this methodology does not improve the worst case value of the objective function, which is the main cause of the conservatism of the robust solutions. On the other hand, feasibility relaxation permits to improve the worst case value of the objective function by trading off the feasibility level concerning the deterministic BLP.

RBIU also provides control on the degree of conservatism by conveniently setting the parameters $\delta_j^{max}$. By increasing this parameter the decision maker increases the amount of constraint relaxation, and an associated improvement in the worst case value of the objective function may be possible. An increment in the value of $\delta_j^{max}$ may lead to solutions that are not feasible for the BLP; therefore, a reduction of conservatism is achieved by sacrificing feasibility with respect to the deterministic problem. In many practical applications, solutions are required to be feasible with respect to the deterministic BLP; such solutions can be obtained by setting each parameter $\delta_j^{max}$ to value zero for all $j$. By Theorem 2.3.1, it follows that $\mathcal{U}(x_C^*) \subseteq X$ when $\delta_j^{max} = 0, \forall j$.

When using feasibility relaxation, the decision maker is optimistic by assuming that the possible improvement in the objective function is greater than the cost of violating feasibility with respect to the deterministic BLP. For instance, in the knapsack problem the cost of increasing the given capacity may be lower than the possible increment in the profit due to the extra capacity. The optimistic level of the decision maker depends on his selected level of the constraint relaxation; a pessimistic decision maker may consider that no constraint relaxation is possible or there is no value in doing so.

The use of these two methods simultaneously may result in a possible higher reduction of the level of the conservatism. Feasibility relaxation permits to reduce the worst case value of the objective function leading to an overall reduction of the conservatism; the selection of the bounding solutions allows to control the level of the conservatism among the robust-optimal solution set obtained from the resolution of the RBIU with a fixed feasibility level.

### 2.5.2 Cardinality-Constrained Robust Formulation

A different methodology to address the conservatism of the solutions of the RBIU consists of the development of a CBIU. Similar to Bertsimas and Sim Bertsimas and Sim (2004), this formulation constrains the maximum number of uncertain variables that are impacted by uncertainty simultaneously. The CBIU accounts for uncertainty impacting discrete variables, which impacts the entire column of the uncertain variables including constraints and the objective function simultaneously. Moreover, the interval of uncertainty in the proposed model is asymmetric and discrete due to the nature of the decision variables. The CBIU is founded by the expressions in Lemma 2.4.1 that permits to represent the implementation uncertainty in binary variables through a linear expression. Similarly to the RBIU, the CBIU contains parameters to control the feasibility relaxation level as described in Section 2.5.1. The use of the two controls of conservatism may lead to a high reduction of the conservatism but a high probability of losing protection. The development of the CBIU is presented next.

#### *2.5.2.1 Development of the Cardinality-Constrained Robust Formulation for a BLP Under Implementation*

Consider the expressions in Lemma 2.4.1 for maximum contribution of $a_{ij}x_i$ and $c_ix_i$. Let $\Gamma$ be an integer parameter with $1 \leq \Gamma \leq |U|$ such that $\Gamma$ represents the maximum number of uncertain variables with different prescribed and implemented values.

Let $\mathcal{X}''$ be a robust feasible region defined by the following constraints:

$$\sum_{i \in C} c_i x_i + \max_{\{S_0 : S_0 \subseteq U, |S_0| \leq \Gamma\}} \left\{ \sum_{i \in S_0} \left( \frac{c_i + |c_i|}{2} \right) + \sum_{i \in U \setminus S_0} c_i x_i \right\} \leq \gamma(x) \tag{2.18}$$

$$\sum_{i \in C} a_{ij} x_i + \max_{\{S_j : S_j \subseteq U, |S_j| \leq \Gamma\}} \left\{ \sum_{i \in S_j} \left( \frac{a_{ij} + |a_{ij}|}{2} \right) + \sum_{i \in U \setminus S_j} a_{ij} x_i \right\} - \delta_j(x) \leq b_j, \forall j \tag{2.19}$$

(2.6), (2.7), (2.8)

The CBIU is formulated as follows:

$$\min_x \left\{ \gamma(x) : x, \gamma, \delta_1, ..., \delta_m \in \mathcal{X}'' \right\}. \tag{2.20}$$

The CBIU possesses an alternative control parameter of the conservatism, $\delta_j^{max}$, that controls the level of feasibility relaxation as described in Section 2.5.1. The incorporation of the control of feasibility relaxation results in a possible greater reduction of the conservatism of the solutions, but there exists a higher probability of losing protection and feasibility with respect to the deterministic BLP. The probability of the CBIU losing protection is estimated in Section 2.5.2.2.

Formulation (2.20) considers every combination of at most $\Gamma$ uncertain variables with different prescribed and implemented values, identifies the combination that produces the maximum contribution to $f(x)$ and $g_j(x)$, and protects the given levels of feasibility and optimality against these maximum contributions. The selection of the sets $S_0$ and $S_j$ in the CBIU are nonlinear combinatorial problems making the CBIU a nonlinear combinatorial problem as well. Theorem 2.5.1 addresses the nonlinearity of the CBIU and describes an equivalent linear reformulation.

**Theorem 2.5.1.** *The following mixed-binary linear programming problem is equivalent to the CBIU.*

$$
\begin{aligned}
\min \quad & \gamma \\
\textit{s.t.} \quad & \sum_{i \in C} c_i x_i + \Gamma v_0 + u_{00} + \sum_{i \in U} u_{i0} \le \gamma \\
& v_0 + u_{i0} \ge \frac{c_i + |c_i|}{2} - c_i x_i, & \forall i \in U \\
& u_{00} \ge \sum_{i \in U} c_i x_i \\
& \sum_{i \in C} a_{ij} x_i + \Gamma v_j + u_{0j} + \sum_{i \in U} u_{ij} - \delta_j \le b_j, & \forall j \qquad (2.21) \\
& v_j + u_{ij} \ge \frac{a_{ij} + |a_{ij}|}{2} - a_{ij} x_i, & \forall i \in U, j \\
& u_{0j} \ge \sum_{i \in U} a_{ij} x_i, & \forall j \\
& u_{i0}, u_{ij}, v_0, v_j \ge 0, & \forall i \in U, j \\
& (2.6), (2.7), (2.8)
\end{aligned}
$$

Proof of Theorem 2.5.1 is based on demonstrating that the selection of the sets $S_0$ and $S_j$ is equivalent to linear formulations that can be substituted into the CBIU by using concepts of duality.

The following definitions and propositions provide the necessary results to complete this proof.

**Definition 2.5.1.** *Given a prescribed vector $\hat{x}$ and the value of $\Gamma$, the protection function $\beta_0(\hat{x}, \Gamma)$ of the optimality robustness constraint against at most $\Gamma$ uncertain variables with different prescribed and implemented values is defined as follows:*

$$\beta_0(x, \Gamma) = \max_{\{S_0 : S_0 \subseteq U, |S_0| \le \Gamma\}} \left\{ \sum_{i \in S_0} \left( \frac{c_i + |c_i|}{2} \right) + \sum_{i \in U \setminus S_0} c_i x_i \right\}. \qquad (2.22)$$

**Proposition 2.5.3.** *The protection function $\beta_0(\hat{x}, \Gamma)$ equals the value of the objective function of the following linear program:*

$$\begin{aligned} \beta_0(\hat{x}, \Gamma) = \max \quad & \sum_{i \in U} \left( \frac{c_i + |c_i|}{2} \right) z_{i0} + \sum_{i \in U} c_i \hat{x}_i (1 - z_{i0}) \\ \textit{s.t.} \quad & \sum_{i \in U} z_{i0} \leq \Gamma \\ & z_{i0} \in \{0, 1\}, \quad \forall i \in U. \end{aligned} \tag{2.23}$$

*Proof.* Since $c_i \hat{x}_i \leq (c_i + |c_i|)/2, \forall i$ the optimal solution of problem (2.23) clearly consists of at most $\Gamma$ variables $z_{i0}$ with value 1. This is equivalent to the selection of the subset $S_0$ such that $S_0 \subseteq U, |S_0| \leq \Gamma$, and maximizes the value of $\sum_{i \in S_0} ((c_i + |c_i|)/2) + \sum_{i \in U \setminus S_0} c_i x_i$. □

**Definition 2.5.2.** *Given a prescribed vector $\hat{x}$ and the value of $\Gamma$, the protection function $\beta_j(\hat{x}, \Gamma)$ of the $j$-th feasibility robustness constraint against at most $\Gamma$ uncertain variables with different prescribed and implemented values is defined as follows:*

$$\beta_j(x, \Gamma) = \max_{\{S_j : S_j \subseteq U, |S_j| \leq \Gamma\}} \left\{ \sum_{i \in S_j} \left( \frac{a_{ij} + |a_{ij}|}{2} \right) + \sum_{i \in U \setminus S_j} a_{ij} x_i \right\}. \tag{2.24}$$

**Proposition 2.5.4.** *The protection function $\beta_j(\hat{x}, \Gamma)$ equals the value of the objective function of the following linear program:*

$$\begin{aligned} \beta_j(\hat{x}, \Gamma) = \max \quad & \sum_{i \in U} \left( \frac{a_{ij} + |a_{ij}|}{2} \right) z_{ij} + \sum_{i \in U} a_{ij} \hat{x}_i (1 - z_{ij}) \\ \textit{s.t.} \quad & \sum_{i \in U} z_{ij} \leq \Gamma \\ & z_{i1} \in \{0, 1\}, \quad \forall i \in U. \end{aligned} \tag{2.25}$$

*Proof.* Proof of Proposition 2.5.4 is similar to proof of Proposition 2.5.3. □

The proof of Theorem 2.5.1 is completed as follows.

*Proof.* Given that the constraint matrix is totally unimodular, it is possible to relax the constraint $z_{i0} \in \{0,1\}, \forall i \in U$ in formulation (2.23) with $0 \leq z_{i0} \leq 1, \forall i \in U$.

The dual problem of formulation(2.23) is the following:

$$
\begin{aligned}
\min \quad & \Gamma v_0 + u_{00} + \sum_{i \in U} u_{i0} \\
\text{s.t.} \quad & v_0 + u_{i0} \geq \frac{c_i + |c_i|}{2} - c_i \hat{x}_i, \quad \forall i \in U \\
& u_{00} \geq \sum_{i \in U} c_i \hat{x}_i \\
& v_0, u_{i0} \geq 0, \qquad\qquad\qquad \forall i \in U \\
& u_{00} \text{ is unrestricted.}
\end{aligned}
\tag{2.26}
$$

Problem (2.23) is feasible and bounded for all $\Gamma \in [0, |U|]$, by strong duality the dual problem (2.26) is also feasible and bounded and their objective values coincide. By Proposition 2.5.3, the value of $\beta_0(\hat{x}, \Gamma)$ is equal to the objective value of the dual problem (2.26). Therefore, it is possible to substitute the dual problem (2.26) into CBIU. A similar process follows for the function $\beta_j(\hat{x}, \Gamma)$, and its equivalent linear and dual problems. Therefore the CBIU is equivalent to the CMBP. □

The CMBP contains: 1) $n$ binary variables, 2) $m+2$ unrestricted variables, 3) $mn+m+n+1$ nonnegative variables, and 4) $2|U|m+5m+|U|+n+3$ constraints. While the CMBP in Theorem 2.5.1 assumes that there exist at most $\Gamma$ uncertain variables with different prescribed and implemented values, Corollary 2.5.1 assumes that there exist exactly $\Gamma$ uncertain variables with different prescribed and implemented values.

**Corollary 2.5.1.** *A cardinality-constrained robust formulation with exact $\Gamma$ uncertain variables with different prescribed and implemented values is equivalent to CMBP formulation with the variables $v_0$ and $v_j$ unrestricted.*

*Proof.* Consider constraints $\sum_{i \in U} z_{i0} \leq \Gamma$ and $\sum_{i \in U} z_{ij} \leq \Gamma$, $\forall j$ in Propositions 2.5.3 and 2.5.4 and replace them by $\sum_{i \in U} z_{i0} = \Gamma$ and $\sum_{i \in U} z_{ij} = \Gamma$ respectively. Then, the associated variables $v_0$ and $v_j$ in their corresponding dual problems become unrestricted. The rest of the proof is similar to the proof of Theorem 2.5.1. □

**Proposition 2.5.5.** *CBIU is equivalent to RMBP if* $\Gamma = |U|$.

*Proof.* With $\Gamma = |U|$ it follows that $|S_0| = |S_j| = |U|, \forall j$; then $S_0 = S_j = U, \forall j$. Therefore the constraints (2.18) and (2.19) in CBIU can be simplified as follows:

$$\begin{aligned} &\sum_{i \in C} c_i x_i + \sum_{i \in U} \left( \frac{c_i + |c_i|}{2} \right) \leq \gamma \\ &\sum_{i \in C} a_{ij} x_i + \sum_{i \in U} \left( \frac{a_{ij} + |a_{ij}|}{2} \right) - \delta_j \leq b_j, \quad \forall j, \end{aligned} \tag{2.27}$$

which correspond to the constraints (2.12) and (2.13) in the feasible region $\mathcal{X}'$ of RMBP.

□

Unlike the RBIU, CBIU does not sacrifice feasibility to improve optimality but it sacrifices protection of the solutions instead. The following section presents estimates of the probabilities of losing protection against uncertainty when using CBIU and presents upper bounds for such probabilities.

#### 2.5.2.2 *Probability bounds*

CBIU solutions satisfy the given optimality and feasibility levels if there exist at most $\Gamma$ uncertain variables with different implemented and prescribed values. In reality, there may exist more than $\Gamma$ uncertain variables with different prescribed and implemented values; therefore, there exist probabilities that the levels of optimality and feasibility are not satisfied. To estimate such probabilities, Assumption 2.2.1 will be relaxed and it will be assumed that $p_i = p, \forall i$ and $q_i = q, \forall i$, with $p$ and $q$ known.

Let $\eta_0$ and $\eta_1$ be two independent random variables such that $\eta_0$ measures the number of uncertain variables with prescribed value 0 and implemented value 1, and $\eta_1$ measures the number of uncertain variables with prescribed value 1 and the implemented value 0. Then the probability that there exist exactly $\Gamma$ uncertain variables with different prescribed and implemented values can be computed as follows:

$$P(\eta_0 + \eta_1 = \Gamma) = \sum_{i=0}^{\Gamma} \binom{U_0}{i} (1-p)^i p^{U_0 - i} \binom{U_1}{\Gamma - i} (1-q)^{\Gamma - i} q^{U_1 - \Gamma + i}, \tag{2.28}$$

where $U_0$ and $U_1$ are the number of uncertain variables whose prescribed values are 0 and 1, respectively, and $U_0 + U_1 = |U|$. Distribution (2.28) represents the sum of two independent binomial random variables $\eta_0$ and $\eta_1$.

**Theorem 2.5.2.** *Given a CBIU solution $x^*, \gamma^*$ and $\delta_j^*$, the upper bound of the probability that the solution robustness or any of the $j$-th model robustness constraints is not satisfied is given by:*

$$\begin{aligned} P\left(\left\{\sum_{i=1}^{n} c_i x_i^* > \gamma^*\right\} \cup \left\{\sum_{i=1}^{n} a_{ij} x_i^* > b_j + \delta_j^*\right\}\right) \\ \leq 1 - \sum_{\ell=0}^{\Gamma} \left( \sum_{i=0}^{\ell} \binom{U_0}{i} (1-p)^i p^{U_0 - i} \binom{U_1}{\ell - i} (1-q)^{\ell - i} q^{U_1 - \ell + i} \right). \end{aligned} \tag{2.29}$$

*Proof.* Given that the solution and model robustness constraints are protected for at most $\Gamma$ uncertain variables with different prescribed and implemented values, it can be assumed that they may become infeasible if there exist at least one one more uncertain variable with such characteristic; this is $\eta_0 + \eta_1 > \Gamma$ in expression (2.28). Therefore, expression (2.29) holds. □

Note that this probability bound depends alone on the number of uncertain variables with different implemented and prescribed values, and the number of uncertain variables with prescribed value zero or one. By not depending on other elements such as the parameters of the problem or the values of $\delta_j^{max}$, this probability may be loose for certain problems.

When $p = q$, the distribution of $\eta_0 + \eta_1$ is a binomial distribution with parameters $1-p$ and $|U|$, i.e. $\eta_0 + \eta_1 \sim B(|U|, 1-p)$. Furthermore, with very little information available, one may assume that for all the uncertain variables the implemented values are equal or different to the prescribed value with the same probability, independent of the prescribed value. The assumption is modeled by making $p = 1/2$ and the probability bound for this special case is shown in Corollary 2.5.2.

**Corollary 2.5.2.** *Given a CBIU solution $x^*, \gamma^*$ and $\delta_j^*$, and $\eta_0 + \eta_1 \sim B(|U|, 1/2)$, the upper bound of the probability that the solution robustness or any $j$-th model robustness constraint is not satisfied is given by:*

$$P\left(\left\{\sum_{i=1}^{n} c_i x_i^* > \gamma^*\right\} \cup \left\{\sum_{i=1}^{n} a_{ij} x_i^* > b_j + \delta_j^*\right\}\right) \leq 1 - \frac{1}{2^{|U|}} \sum_{\ell=0}^{\Gamma} \binom{|U|}{\ell}. \qquad (2.30)$$

*Proof.* This proof is similar to proof of Theorem 2.5.2 with $\eta_0 + \eta_1 \sim B(|U|, 1/2)$. □

When the assumption that $p = 1/2$ is not satisfied, Theorem 2.5.2 is applied.

#### *2.5.2.3 Properties of the CBIU for Certain Problem Structures*

This section shows particular cases where the CBIU formulation shown in Section 2.5.2.1 does not work as intended. Results in Lemmas 2.5.1 and 2.5.2, and Corollaries 2.5.3, 2.5.4, 2.5.5 and 2.5.6 show that, due to certain structural properties of the BLP, there may not be possible to control the level of conservatism using a cardinality-constrained robust formulation. Moreover, Corollaries 2.5.4 and 2.5.6 state that for a particular BLP the CBIU and RBIU are equivalent independent of $\Gamma$.

**Lemma 2.5.1.** *The cardinality-constrained robust counterpart of a constraint of the form* $\sum_{i=1}^{n} x_i \geq b$ *is independent of the value of* $\Gamma$.

*Proof.* The cardinality-constrained robust counterpart of a constraint of the form $\sum_{i=1}^{n} x_i \geq b$ is the following:

$$\sum_{i \in C} x_i - \max_{\{S: S \subseteq U, |S| \leq \Gamma\}} \left\{ - \sum_{i \in U \setminus S} x_i \right\} + \delta \geq b. \tag{2.31}$$

The maximum value of the term $-\sum_{i \in U \setminus S} x_i$ in (2.31) is zero independent of the value of $\Gamma$, with $\Gamma \geq 1$. Then $\sum_{i=1}^{n} x_i \geq b$ can be simplified to $\sum_{i \in C} x_i + \delta \geq b$, and it does not depend of $\Gamma$. □

**Corollary 2.5.3.** *The CBIU is independent of* $\Gamma$ *if the BLP consists of* $c_i \geq 0, \forall i$ *and constraints* $\sum_{i=1}^{n} x_i \geq b_j, \forall j$.

*Proof.* By Lemma 2.5.1, the cardinality-constrained robust counterpart of each constraint $j$ is independent of $\Gamma$. On the other hand, the cardinality-constrained robust counterpart of the objective function is as shown in (2.18). Since $c_i \geq 0$ then the maximum value of the term $\sum_{i \in S_0} ((c_i + |c_i|)/2) + \sum_{i \in U \setminus S_0} c_i x_i$ is given by $\sum_{i \in S_0} ((c_i + |c_i|)/2) + \sum_{i \in U \setminus S_0} c_i = \sum_{i \in U} c_i$ for any value of $\Gamma$. Therefore, the CBIU is independent of $\Gamma$. □

**Corollary 2.5.4.** *The CBIU is equivalent to RBIU independent of* $\Gamma$ *if the BLP consists of* $c_i \geq 0, \forall i$ *and constraints* $\sum_{i=1}^{n} x_i \geq b_j, \forall j$.

*Proof.* By Lemma 2.5.1, the cardinality-constrained robust counterpart of each constraint $j$ is independent of $\Gamma$, and each constraint can be simplified to $\sum_{i=1}^{n} x_i + \delta_j \geq b_j, \forall j$. By Corollary 2.5.4, the term $\sum_{i \in S_0} ((c_i + |c_i|)/2) + \sum_{i \in U \setminus S_0} c_i x_i$ is equivalent to $\sum_{i \in U} c_i = \sum_{i \in U} ((c_i + |c_i|)/2)$; constraint (2.18) can be rewritten as $\sum_{i \in C} c_i x_i + \sum_{i \in U} ((c_i + |c_i|)/2)$ for any value of $\Gamma$. Therefore, the reduced expressions of the cardinality-constrained robust formulation corresponds to the RBIU formulation independent of the value of $\Gamma$. □

**Lemma 2.5.2.** *The cardinality-constrained robust counterpart of a constraint of the form* $\sum_{i=1}^{n} x_i = b$ *is independent of the value of* $\Gamma$.

*Proof.* The cardinality-constrained robust counterpart of a constraint of the form $\sum_{i=1}^{n} x_i = b$ is the following:

$$\sum_{i \in C} x_i - \max_{\{S' : S' \subseteq U, |S'| \leq \Gamma\}} \left\{ - \sum_{i \in U \setminus S'} x_i \right\} + \delta^G \geq b \tag{2.32}$$

$$\sum_{i \in C} x_i + \max_{\{S'' : S'' \subseteq U, |S''| \leq \Gamma\}} \left\{ |S''| + \sum_{i \in U \setminus S''} x_i \right\} - \delta^L \leq b. \tag{2.33}$$

By Lemma 2.5.1, constraint (2.32) is independent of $\Gamma$. On the other hand, the maximum value of the term $|S''| + \sum_{i \in U \setminus S''} x_i$ is given by $|S''| + |U \setminus S''| = |U|$ independent of $\Gamma$, with $\Gamma \geq 1$; then constraint (2.33) can be simplified to $\sum_{i \in C} x_i + |U| - \delta \leq b$, and it does not depend of $\Gamma$. $\square$

**Corollary 2.5.5.** *The CBIU is independent of* $\Gamma$ *if the BLP consists of* $c_i \geq 0, \forall i$ *and constraints* $\sum_{i=1}^{n} x_i = b_j, \forall j$.

*Proof.* Proof of Corollary 2.5.5 is similar to proof of Corollary 2.5.3. $\square$

**Corollary 2.5.6.** *The CBIU is equivalent to RBIU independent of* $\Gamma$ *if the BLP consists of* $c_i \geq 0, \forall i$ *and constraints* $\sum_{i=1}^{n} x_i = b_j, \forall j$.

*Proof.* Proof of Corollary 2.5.6 is similar to proof of Corollary 2.5.4. $\square$

## 2.6 Summary and Conclusions

The work presented in this chapter represents the first attempt in the field of robust optimization to model a BLP under implementation uncertainty. The model of binary variables under implementation uncertainty is the groundwork for the development of the measures of robustness and the development of a model to solve BLPs under this type of uncertainty.

The robust formulation for a BLP under implementation uncertainty permits to identify solutions that satisfy the given level of feasibility when minimize the degradation of the objective function value for any realization of the uncertain variables. This chapter also presents a solution methodology to solve the RBIU.

The level of the conservatism of the RBIU can be controlled through a relaxation of the feasibility level and the selection of the bounding solutions among the robust-optimal solution set. The cardinality-constrained robust BLP under implementation uncertainty allows the control of the conservatism by bounding the maximum number of uncertain variables with different prescribed and implemented values and a relaxation of the feasibility level.

The proposed concepts of robustness can be applied to well-known BLPs under implementation uncertainty. Particular problem characteristics may make the robust solutions more attractive from a theoretical or practical standpoint. Applications of these concepts are presented in the subsequent chapters.

# 3. ROBUST KNAPSACK PROBLEM UNDER IMPLEMENTATION UNCERTAINTY

## 3.1 Introduction

This section presents an application of the proposed concepts of robustness presented in Chapter 2 to the KP under implementation uncertainty. Properties of the robust formulations and robust solutions are developed. This chapter includes an experimental study to evaluate the performance of the deterministic and robust solutions in terms of the objective value and feasibility level.

The KP under implementation uncertainty considers that some options may not be available at the time of implementation or options not initially considered may be forced to be selected, for instance, as a result of policy changes. Consider for example the problem of loading a plane with the most valuable cargo; a missing item may not be available to be loaded, or changes in priority may lead to loading an item not prescribed. These changes in the implemented values may lead to an excess of the initial capacity or high degradation of the objective values. Existing work in the field of robust optimization accounts only for the KP under data uncertainty (e.g. Yu, 1996; Bertsimas and Sim, 2003, 2004). This work presents the first attempt to study this problem under implementation uncertainty.

The concepts of robustness developed in Chapter 2 are applied to the KP under implementation uncertainty. This work formulates the corresponding robust KP under implementation uncertainty and its corresponding linear reformulation as a solution methodology. Due to its structural properties, the RKP can be transformed to an equivalent KP free of uncertainty to which existing dynamic programming algorithms can be applied. The cardinality-constrained robust KP under implementation uncertainty and its properties are also presented in this chapter.

This chapter presents an experimental study of the KP under implementation uncertainty. This study aims to compare the performance of the deterministic and robust solutions in terms of the objective function value and feasibility level. This work develops measures of performance to evaluate the solutions. The measure of the objective function performance computes the average value of the feasible solutions among all the uncertain set; the measure of feasibility performance computes the proportion of the vectors in the uncertain set that satisfies the given level of feasibility. The results of the experimental study show the characteristics of the KP that make the deterministic solutions more sensitive to uncertainty and make the robust solutions more appropriate.

The remainder of this chapter is organized as follows. Section 3.2 presents the development of the robust formulations RKP and demonstrates its equivalence to a KP free of uncertainty. Section 3.3 shows the corresponding CRKP, its equivalent linear reformulation and their properties. Section 3.4 presents the experimental study. Section 3.5 presents concluding remarks for the chapter.

## 3.2 Robust Knapsack Problem Under Implementation Uncertainty

The deterministic KP can be described as follows: given a set $N$ containing $n$ different items, each item possesses a profit $c_i \geq 0$ and weight $a_i \geq 0$ for $i = 1, ..., n$, and a maximum capacity $b \geq 0$. The objective of the KP is to select a subset of items such that the total profit $f(x)$ is maximum and the total weight does not exceed $b$. The KP can be formulated as follows:

$$\begin{aligned} \max f(x) = & \sum_{i=1}^{n} c_i x_i \\ \text{s.t.} & \sum_{i=1}^{n} a_i x_i \leq b \\ & x_i \in \{0, 1\}, \quad i = 1, ..., n. \end{aligned} \tag{3.1}$$

The binary decision variable $x_i$ is equal to 1 if the $i$-th item is selected, and the value of $x_i$ is equal to 0 otherwise.

In practice, there exist a possibility that some of the options selected by the deterministic solutions of the KP formulation may not be available or options not initially considered may be forced to be selected at the time of the implementation. In any of these situations, the implemented values of the decision variables are different to the prescribed values.

Definition 2.2.1 of a binary variable under implementation uncertainty and definition 2.2.2 of the uncertain set can be applied to model this situation. Moreover, the KP under this type of uncertainty can be solved by using the robust formulation developed in Section 2.3.2. The development of the robust KP under implementation uncertainty is presented next.

Without loss of generality, the objective function of the formulation (3.1) will be treated as $-\min\{-f(x)\} = -\min\{-\sum_{i=1}^{n} c_i x_i\}$.

The measure of objective robustness level in Definition 2.3.1 corresponding to the KP is the following:

$$\gamma(x) = \max_{y \in \mathcal{U}(x_C)} \left\{ -\sum_{i=1}^{n} c_i y_i \right\}. \tag{3.2}$$

Similarly, the measure of feasibility robustness level in Definition 2.3.2 corresponding to the KP is the following:

$$\delta(x) = \max_{y \in \mathcal{U}(x_C)} \left\{ \sum_{i=1}^{n} a_i y_i - b, 0 \right\}. \tag{3.3}$$

with $\delta(x) \leq \delta^{max}$.

In the context of the KP, the parameter $\delta^{max}$ represents the extra capacity that the decision maker is accepted to consider in order to reduce the conservatism of the robust solution.

Considering the measures of the level of robustness, the robust knapsack problem under implementation uncertainty (RKP) can be formulated as follows:

$$\min \quad \gamma(x) \tag{3.4}$$

$$\text{s.t.} \quad \max_{y \in \mathcal{U}(x_C)} \left\{ \sum_{i=1}^{n} -c_i y_i \right\} \leq \gamma \tag{3.5}$$

$$\max_{y \in \mathcal{U}(x_C)} \left\{ \sum_{i=1}^{n} a_i y_i \right\} - \delta \leq b \tag{3.6}$$

$$x_i \in \{0, 1\}, \quad \forall i \tag{3.7}$$

$$\gamma \leq 0 \tag{3.8}$$

$$0 \leq \delta \leq \delta^{max}. \tag{3.9}$$

The objective function (3.4) seeks to minimize the worst-case value of the total profit. Constraint (3.5) is the objective robustness constraint guaranteeing that definition of $\gamma$ is satisfied. Similarly, constraint (3.6) is the feasibility robustness constraint and together with constraint (3.9) guarantee that definition of $\delta$ is satisfied. Constraints (3.7) are the binary constraints for the decision variables $x_i, \forall i$.

Note that $\max_{y \in \mathcal{U}(x_C)} \{\sum_{i=1}^{n} a_i y_i\} - \delta^{max} \leq \max_{y \in \mathcal{U}(x_C)} \{\sum_{i=1}^{n} a_i y_i\} - \delta$, and constraints (3.6) and (3.9) can be combined into a single constraint $\max_{y \in \mathcal{U}(x_C)} \{\sum_{i=1}^{n} a_i y_i\} - \delta^{max} \leq b.$. The equivalent linear reformulation corresponding to the RKP is the following:

$$\min \quad \gamma \tag{3.10}$$

$$\text{s.t.} \quad -\sum_{i \in C} c_i x_i \leq \gamma \tag{3.11}$$

$$\sum_{i \in C} a_i x_i + \sum_{i \in U} a_i - \delta^{max} \leq b \tag{3.12}$$

(3.7), (3.8).

Results in Theorem 2.3.1 and Corollary 2.3.1 hold for the RKP.

Robust formulations tend to increase the computational complexity of the deterministic model. Wolsey (1998) shows that the KP is NP-Complete and presents a dynamic programming approach to solve this problem in pseudo-polynomial time when the coefficients $a_i, \forall i$ and $b$ are positive integers; the dynamic programming algorithm runs in $\mathcal{O}(nb)$ time.

When the parameters of the constraint $a_i, \forall i$ satisfy this condition, the RKP can be solved using the Wolsey's dynamic programming approach by modeling the RKP as an equivalent KP consisting of variables not under implementation uncertainty only. The development of the equivalent KP is shown as follows:

Let $b' = b + \delta^{max} - \sum_{i \in U} a_i$, and let $f'(x) = -\sum_{i \in C} c_i x_i$. The equivalent linear reformulation of the RKP can be reformulated as follows:

$$\begin{aligned} \min \quad & f'(x) \\ \text{s.t.} \quad & \sum_{i \in C} a_i x_i \leq b' \\ & x_i \in \{0,1\}, \quad i \in C. \end{aligned} \tag{3.13}$$

This formulation is a deterministic KP and can be treated as such; therefore, it can be solved in pseudo-polynomial time using dynamic programming as shown in Wolsey (1998).

Appendix B presents an implementation of the dynamic programming approach.

### 3.3 Cardinality-Constrained Robust Formulation of the KP

Similarly, the cardinality-constrained robust optimization concept presented in Section 2.5.2 can be applied to the KP under implementation uncertainty.

The cardinality-constrained robust KP under implementation uncertainty (CRKP) is the following:

$$
\begin{aligned}
\min \quad & \gamma && (3.14)\\
\text{s.t.} \quad & -\sum_{i\in C} c_i x_i + \max_{\{S_0 : S_0 \subseteq U, |S_0| \le \Gamma\}} \left\{ -\sum_{i\in U\setminus S_0} c_i x_i \right\} \le \gamma && (3.15)\\
& \sum_{i\in C} a_i x_i + \max_{\{S_1 : S_1 \subseteq U, |S_1| \le \Gamma\}} \left\{ \sum_{i\in S_1} a_i + \sum_{i\in U\setminus S_1} a_i x_i \right\} - \delta^{max} \le b && (3.16)\\
& (3.7), (3.8).
\end{aligned}
$$

And the equivalent linear reformulation as shown in Section 2.5.2 is the following:

$$
\begin{aligned}
\min \quad & \gamma \\
\text{s.t.} \quad & -\sum_{i\in D} c_i x_i + \Gamma v_0 + u_{00} + \sum_{i\in U} u_{i0} \le \gamma \\
& \sum_{i\in D} a_i x_i + \Gamma v_1 + u_{0j} + \sum_{i\in U} u_{i1} - \delta^{max} \le b \\
& v_0 + u_{i0} \ge c_i x_i, \quad \forall i \in U \\
& u_{00} \ge -\sum_{i\in U} c_i x_i \\
& v_1 + u_{i1} \ge a_i - a_i x_i, \quad \forall i \in U \\
& u_{01} \ge \sum_{i\in U} a_i x_i \\
& u_{i0}, u_{i1}, v_0, v_1 \ge 0, \quad \forall i \in U \\
& u_{00}, u_{01} \text{ are unrestricted} \\
& (3.7), (3.8).
\end{aligned} \qquad (3.17)
$$

In contrast to the RKP, the CRKP cannot be modeled as an equivalent KP. Existing methods to solve cardinality-constrained robust formulations address problems with uncertainty impacting to the objective function only Bertsimas and Sim (2003). Implementation uncertainty impacts the objective function and constraint of the KP simultaneously; therefore, the existing solutions methods cannot be applied to the CRKP. The CRKP can be solved using existing MILP algorithms or commercial software applied to its linear reformulation.

Result in Theorem 2.5.2 can be also apply to the CRKP. The corresponding expression for the CRKP is the following:

$$P\left(\left\{-\sum_{i=1}^{n} c_i x_i^* > \gamma^*\right\} \cup \left\{\sum_{i=1}^{n} a_i x_i^* > b + \delta^{max}\right\}\right)$$
$$\leq 1 - \sum_{\ell=0}^{\Gamma}\left(\sum_{i=0}^{\ell}\binom{U_0}{i}(1-p)^i p^{U_0-i}\binom{U_1}{\ell-i}(1-q)^{\ell-i}q^{U_1-\ell+i}\right) \quad (3.18)$$

## 3.4 Experimental Study

This section presents the experimental study for the KP under implementation uncertainty. The objective of this study is to illustrate the sensitivity of the deterministic and robust solutions under uncertainty, the different levels of the conservatism of the RKP and CRKP, and the tightness of the probability bounds in (3.18).

### 3.4.1 Performance Measures

The solutions' performance is evaluated based on the objective function performance and the level of feasibility in the face of implementation uncertainty. In evaluating these performances, one needs to account for the fact that solutions under implementation uncertainty produce a uncertain set $\mathcal{U}(x_C)$, where $x_C$ is the deterministic component of the solution vector. By the result in Theorem 2.3.1, $\mathcal{U}(x_C)$ provides also the robust-optimal solution set.

Let $F$ represent the feasible region of a given problem, including the constraint relaxation defined by $\delta_j^{max}$, and let $\mathcal{F} \subseteq \mathcal{U}(x_C) = \{x \in \mathcal{U}(x_C) : x \in F \cap \mathcal{U}(x_C)\}$ be the set of all the solutions in $\mathcal{U}(x_C)$ that are feasible with respect to the given feasible region. The feasibility ratio $h(x)$ is defined as the proportion of the elements of $\mathcal{U}(x_C)$ that are feasible with respect to $F$; $h(x)$ can be defined as follows:

**Definition 3.4.1.** *The feasibility ratio $h(x)$ is defined as:*

$$h(x) = \frac{|\mathcal{F}|}{|\mathcal{U}(x_C)|}. \tag{3.19}$$

The objective performance ratio $\bar{f}(x)$ measures the average objective value of the feasible solutions in $\mathcal{U}(x_C)$.

**Definition 3.4.2.** *The average objective value of the feasible solutions $\bar{f}(x)$ is defined as follows:*

$$\bar{f}(x) = \frac{\sum_{x \in \mathcal{F}} f(x)}{|\mathcal{F}|}. \tag{3.20}$$

The difference between the objective performance of the solution sets associated with two solutions $x$ and $y$ is measured by the loss of the objective performance ratio $l(x, y)$ of $y$ with respect to $x$; $l(x, y)$ is defined as follows, assuming a minimization objective function:

**Definition 3.4.3.** *The loss of the objective performance ratio $l(x, y)$ is defined as follows:*

$$l(x, y) = \frac{\bar{f}(y) - \bar{f}(x)}{\bar{f}(x)}; \tag{3.21}$$

The larger the value of $l(x, y)$, the worse is the objective performance of $y$ compared to $x$.

### 3.4.2 Test Problem Generation

Test knapsack problems are generated with the values of $c_i$ and $a_i$ randomly generated using a uniform distribution between 1 and 1000. The right hand side of the knapsack constraint is calculated as $b = \alpha \sum a_i$, with constant $\alpha = 0.75, 0.5$ and 0.25 for low, medium and high levels of sensitivity to implementation uncertainty, respectively. The experimental study consists of 30 problems with 20 variables for each level of $\alpha$. Note that, decreasing values of $b$ result in solutions with increasing number of variables at 0, and solutions with more variables prescribed at 0 tend to be more sensitive to implementation uncertainty (i.e. more prone to become infeasible), because there are more chances of $\hat{x}_i = 0$ and $\tilde{x}_i = 1$ resulting in an increase in the value of $\sum_{i=1}^{n} a_i x_i$, and consequently in a possible violation of the knapsack constraint.

Each problem instance is solved for every $|U| = 1, ..., 19$ and $\Gamma = 1, ..., |U|$. Solutions $x^D$, $x^R$ and $x^{CC}$ are obtained by solving KP, RKP and CRKP, respectively. In turn, the corresponding robust-optimal solution sets $\mathcal{U}(x_C^D)$, $\mathcal{U}(x_C^R)$ and $\mathcal{U}(x_C^{CC})$, feasible solution sets $\mathcal{F}(x^D)$, $\mathcal{F}(x^R)$ and $\mathcal{F}(x^{CC})$, and performance measures are obtained via complete enumeration. Recall that $\mathcal{U}$ contains all possible implemented outcomes; therefore, the performance obtained represent the actual performance of a given solution in the face of implementation uncertainty. It is assumed that the implemented value of the uncertain variables can be different or equal to the prescribed value with same probability for any prescribed value, i.e. $p_i = q_i = 0.5, \forall i$.

### 3.4.3 Performance Results

#### *3.4.3.1 RPK and CRPK Performance Results*

Figure 3.1 shows the loss in objective performance ratio of RPK and CRPK ($\Gamma = 1, 5$ and $10$) solutions versus increasing levels of implementation uncertainty (i.e. the number of uncertain variables $|U|$) for high, medium and low sensitivity knapsack problems. In this graph, a higher value represents a worse performance in objective compared to the solution of the KP. As expected, in all cases tested the objective performance ratio of robust solutions is worse than the solution of the KP.

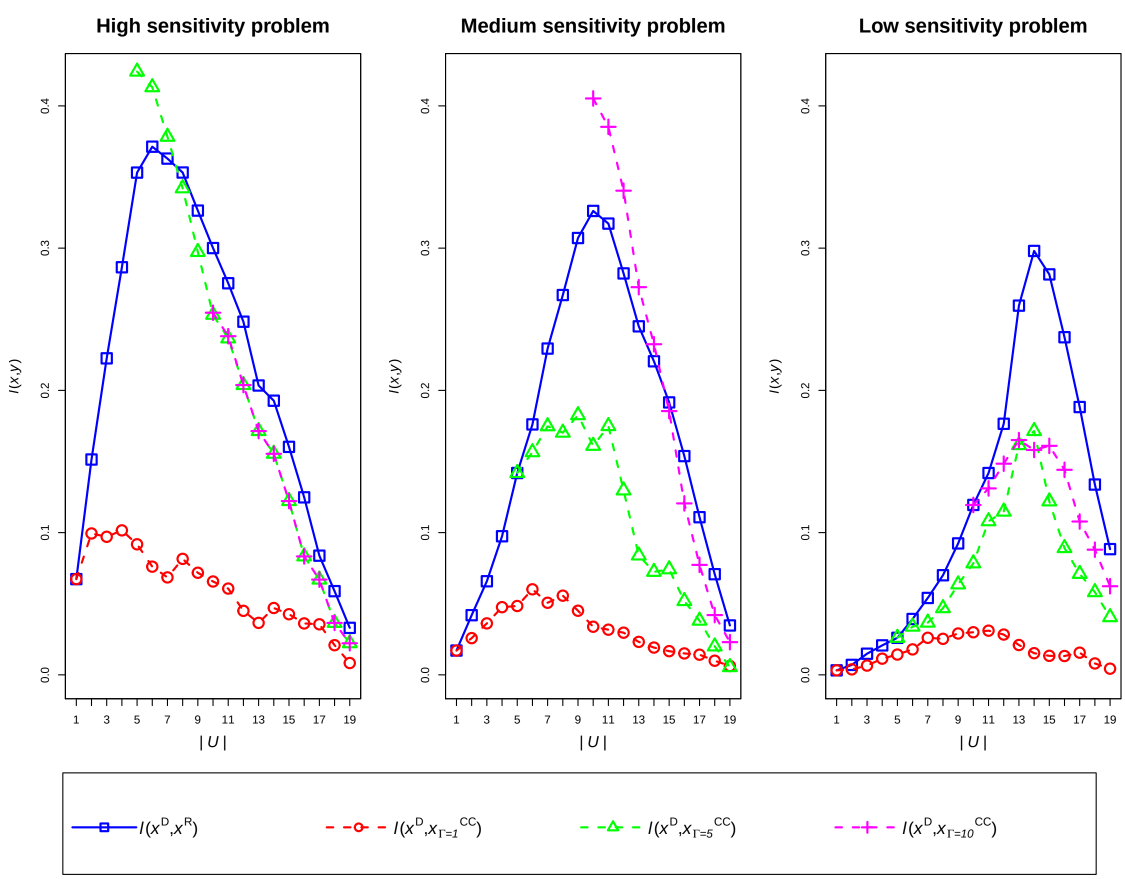



Figure 3.1: Loss of the objective performance ratio $l(x, y)$.

For brevity, the following discussions refer to the medium sensitivity problem plot in Figure 3.1 (the discussion would be similar for the other two cases). RPK solutions perform similar to the solution of the KP with a loss of 0.02 at $|U| = 1$ degrading quickly as the number of uncertain variables start increasing to a loss of around 0.33 at $|U| = 10$; interestingly, the performance starts improving for $|U| > 10$. It can be observed that for this problem set RPK becomes infeasible at around $|U| = 10$; hence for each problem, the last feasible solution is fixed as the solution for the remaining values of $|U|$. The apparent improvement in the loss of the objective performance ratio obtained after the solution is fixed suggests that the difference between robust and solutions of the KP tend to diminish as the level of uncertainty increases and the level of protection of RPK solutions decrease.

It can be observed that CRKP solutions tend to perform better than RPK solutions in terms of objective value. This is because, by design and at the expense of feasibility losses, CRKP produces more optimistic solutions by assuming that only a limited number of variables will be simultaneously affected by uncertainty. On the other hand, RKP solutions are the most conservative solutions assuming all uncertain variables may change their values for the worst, and in our test, also protecting the solutions from becoming infeasible with respect to the KP (i.e. $\delta_{max} = 0$). As a result, RKP solutions attempt to tradeoff objective value for assurance of feasibility during implementation. CRKP solutions tend to behave as the RKP solutions as the value of $\Gamma$ increases; this effect appears to be stronger when the problems are more sensitive to implementation uncertainty.

Figure 3.2 displays the feasible ratio versus the level of uncertainty (i.e. the number of uncertain variables). In this graph, a higher value indicates better performance in terms of feasibility. It can be observed that, when compared with the deterministic case, the loss in objective performance ratio discussed so far is accompanied by better protection against the infeasibility caused by implementation uncertainty; RKP solutions tend to perform the best in terms of feasibility for all levels of uncertainty; thus justifying the loss of the objective performance ratio observed earlier. As expected, CRKP tends to have better feasibility performance as the value of $\Gamma$ increases and rapidly have similar performance as RKP. The experimental results confirm that the degradation in feasibility is significant for problems that are more sensitive to implementation uncertainty and small for low sensitivity problems.

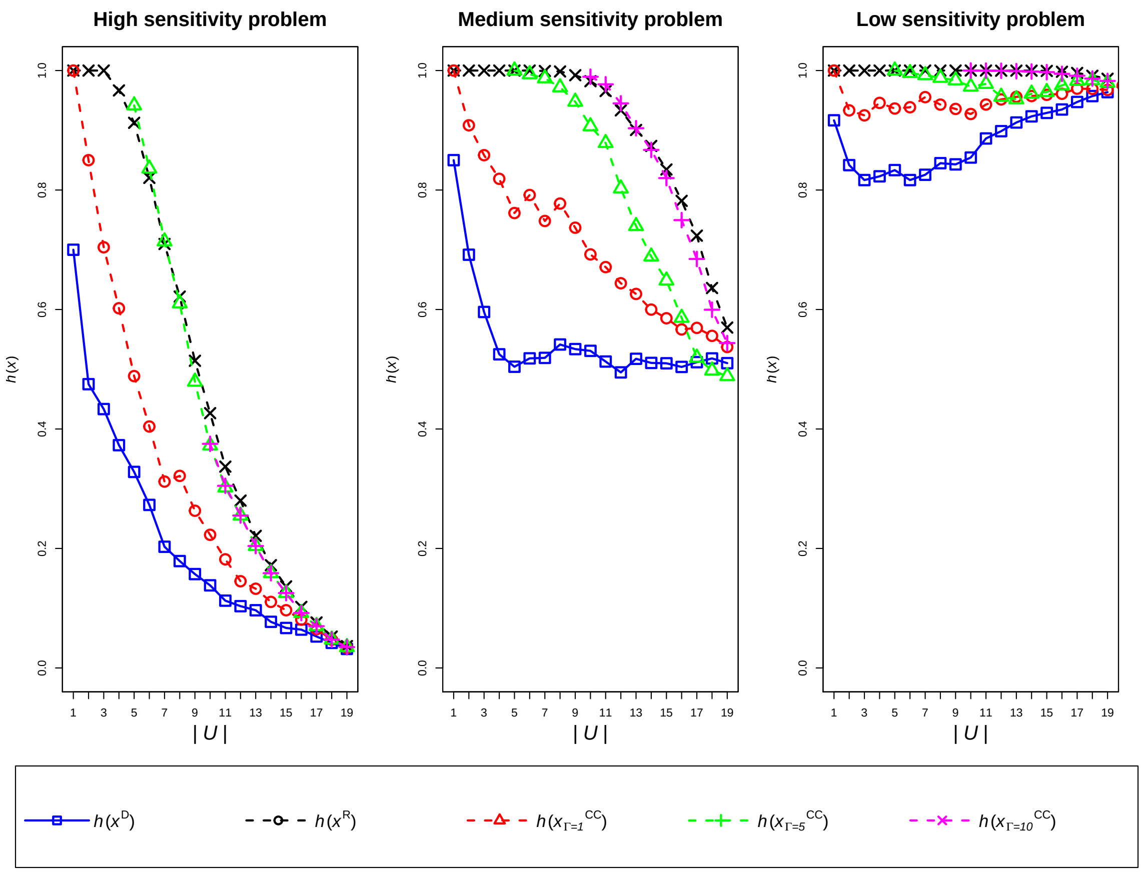



Figure 3.2: Feasible performance ratio $h(x)$.

The experimental study suggests that the benefits of using robust solutions can be dramatic. For instance, for highly sensitive problems with $|U| = 2$, using RKP will result in a 15% loss in objective performance ratio but ensuring feasibility; while the solution of the KP may become infeasible 50% of the time. Another example is for medium sensitivity problems with $|U| = 5$, using CRKP with $\Gamma = 1$ the loss in the objective function is less than 5% with a feasibility of about 75%; while the solution of the KP will only have a feasibility performance of about 50%.

In summary, the experimental results suggest that in some situations solving the knapsack problem using RKP and CRKP may yield solutions that tend to maintain high levels of feasibility with acceptable losses in the objective function performance. Following are some practical recommendations based on the experimental results:

- The solutions of the KP are appropriate when the decision maker is optimistic and believes the problem possesses low sensitivity to uncertainties, or such uncertainties are unlikely to occur. In this case, the decision maker considers that a small infeasibility is acceptable and prefers a better objective value. This is supported by the plots of the loss of objective performance and feasible ratio for low sensitivity problems; it can be observed that the loss of feasibility of the solutions of the KP is low, and although, RKP and CRKP produce better feasibility level, their loss of objective performance ratio is high (optimality plot).

- RKP solutions are appropriate when feasibility is very important, the decision maker is pessimistic, or considers that the problem possesses high sensitivity to uncertainties. Besides, this formulation is appropriate when infeasibilities cannot be accepted even at the expense of objective value degradation. This behavior is observed in Figure 3.2 where the feasibility ratio of RKP solutions is higher than the feasibility ratio of the KP and CRKP solutions.

- CRKP is preferable when additional control in the level of conservativism of the solutions is desired. In this case, the decision maker considers that the worst-case scenario (all uncertain variables changing) is unlikely. As a result, the solutions can lead to a better loss of the objective performance ratio while still providing some protection against uncertainties. This is supported by the plot of optimality that shows CRKP producing better objective values than RKP (the loss of the objective performance ratio of CRKP solutions is smaller than the loss of the objective performance ratio of RKP solutions); however, the feasibility plot shows that CRKP solutions produce smaller feasible ratio than RKP solutions, but better than the feasible ratio of the solution of the KP.

- If possible, use robust solutions when the number of uncertain variables is away, preferably smaller, from the minimum number of uncertain variables where RKP cannot find feasible solutions. In these cases, the losses in objective function performance tend to be smaller, and the feasibility performance better than the solution of the KP.

#### *3.4.3.2 CRKP Probability Bounds Performance*

The experimental study tests the upper bound probability shown in (3.18). Figure 3.3 displays the experimental results and the theoretical values; the graphs show the plots for a few values of $\Gamma$, and only show the portions of the curve where CRKP is feasible. The test shows the mean probability obtained after completely enumerating all possible outcomes of implementation uncertainty. It can be observed that the theoretical bound is more accurate when the problems are highly sensitive to implementation uncertainty, and are loose when the problems have medium or low sensitivity.

#### *3.4.3.3 RKP Solution Methodologies Performance*

The experimental study also evaluates the performance of the solution methodologies for the RKP discussed in Section 3.2, namely the use of the equivalent MILP and the dynamic programming (DP) approach, in terms of their runtime.

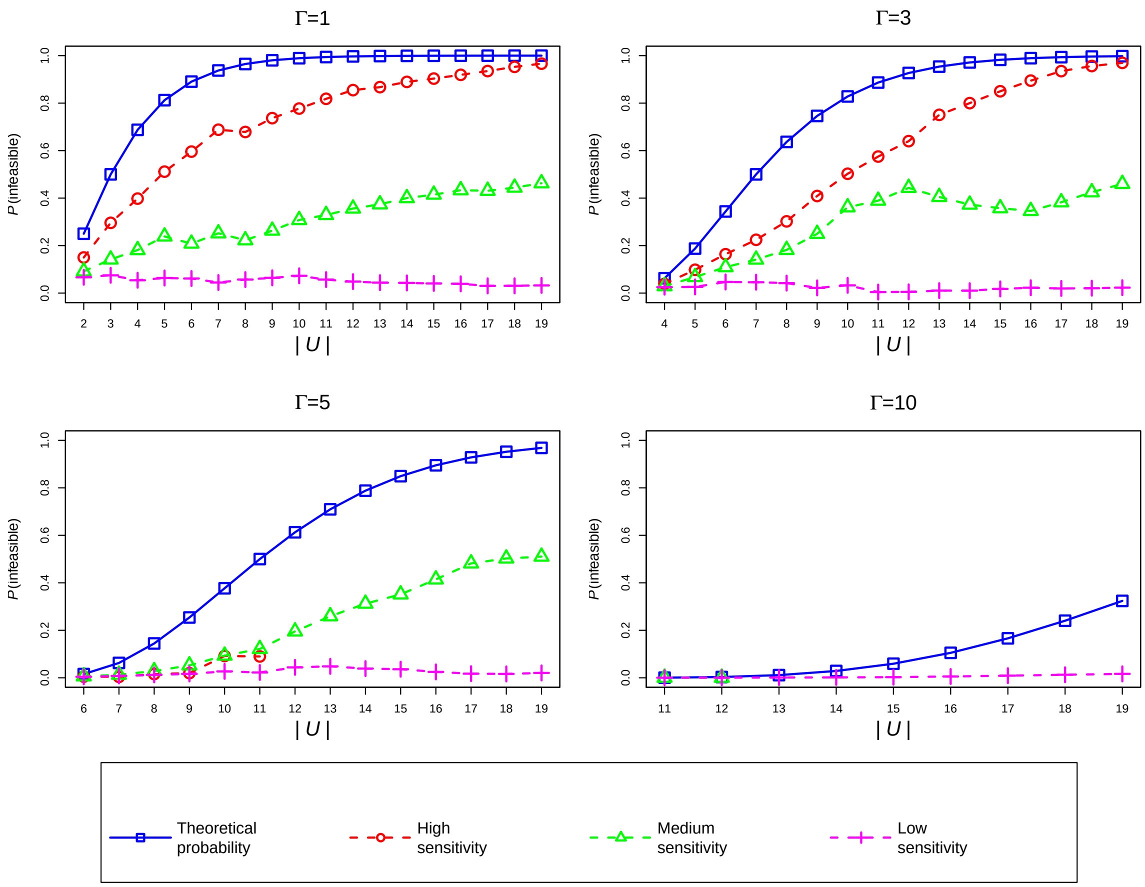



Figure 3.3: Probability bounds of CRKP solutions.

The KP are generated as shown in Section 3.4.2; the values of $a_i, \forall i$ and $b$ are rounded such that the constraint contains integer parameters only. 30 problems with 2000 variables and high sensitivity to uncertainty are tested. The performance of the solution methodologies is also evaluated in terms of the number of uncertain variables with $|U| = 50, 100, 200, 300, 400, ..., 1500$. The average runtime of the 30 problems for every value of $|U|$ is computed; the results are shown in Figure 3.4. The MILP is solved using CPLEX 12.6 and Java; the DP algorithm is implemented in Java.

Results in Figure 3.4 show that the MILP approach provides a significantly runtime when the number of uncertain variables is low, and it tends to decrease as the number of

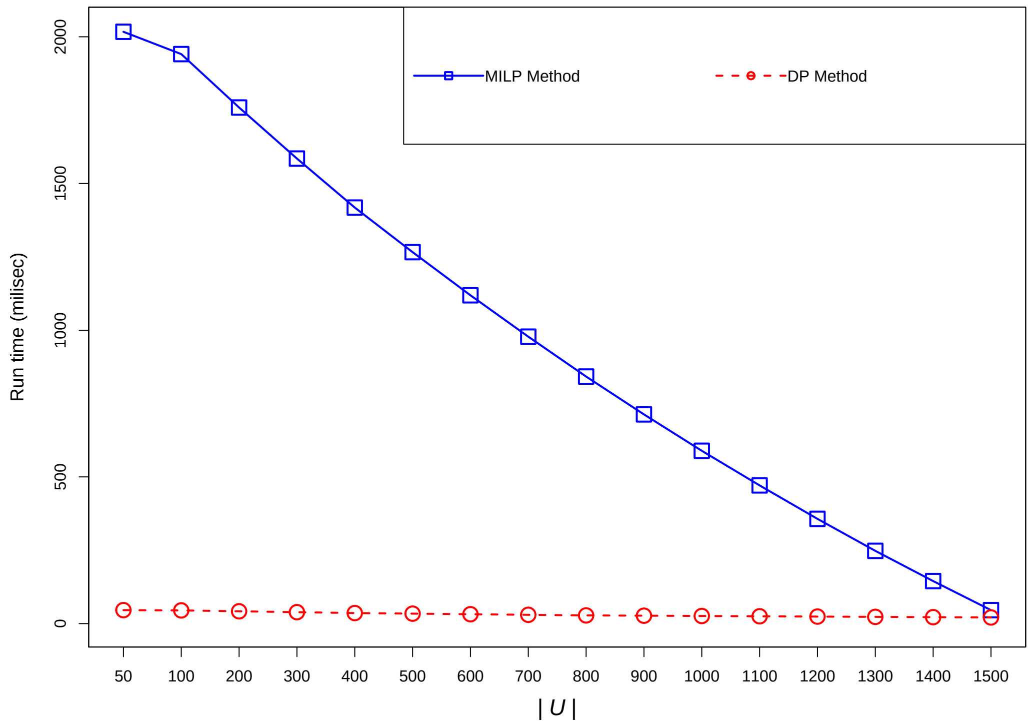



Figure 3.4: Average runtime of the MILP and DP solution approaches.

uncertain variables increases. The runtime of the two approaches is similar for a large number of uncertain variables due to the reduced number of binary variables forming the RKP when the number of uncertain variables is large. The effect of reducing the number of certain variables is less significant for the DP approach than to the MILP.

## 3.5 Summary and Conclusions

This chapter presents a robust formulation for the KP under implementation uncertainty such that it identifies solutions satisfying a certain level of feasibility while the reduction of the profit is minimized. The KP particular problem structures permit to identify an equivalent KP free of uncertainty such that existing dynamic programming solution approaches can be used to determine the robust solutions.

Experimental results show that the applicability of the deterministic solution can easily become infeasible in the presence of uncertainties, while robust solutions guarantee their applicability at the expense of optimality. The results show how the sensitivity of the KP to uncertainty impacts in the performance of the robust solutions making them more attractive for problems with significant sensitivity to implementation uncertainty. For problems with low sensitivity, the loss of optimality is justified when one is interested in protecting the feasibility for all realizations of the uncertainty.

This experimental study shows that computing the robust solutions using a dynamic programming approach is more efficient for large-scale instances of the problem. The performance of the dynamic programming and linear programming methods is similar for problems containing a small number of options. Some practical guidelines are offered for the applications of the proposed methods.

# 4. ROBUST ASSIGNMENT PROBLEM UNDER IMPLEMENTATION UNCERTAINTY

## 4.1 Introduction

This section presents the application of the proposed concepts of robustness presented in Chapter 2 to the AP when this is impacted by implementation uncertainty. Significant theoretical and practical results associated with the robust formulation and robust solutions for this problem are also shown.

The AP under implementation uncertainty considers unexpected changes in the prescribed assignments at the time of the implementation. For instance, last minute changes in schedules may lead to task without being completed, or vehicle assignments may not be possible due to breakdowns in the trucks. These unexpected changes may impact the cost of the assignments or a reduction of the completion level of the assigned tasks. The AP can be also studied in the context of graph theory as the PM. The PM under implementation uncertainty considers that certain edges may not be available during the implementation.

Existing work in robust optimization studying the AP accounts for parameter uncertainty only (e.g. Kouvelis and Yu, 1997; Aissi et al., 2005; Deı et al., 2006). This dissertation attempts to address the AP under implementation uncertainty by applying the concepts of robustness proposed in Chapter 2. This chapter shows the RPM and its corresponding linear reformulation. This work develops different configurations of the controls parameters of the RPM that permit to identify the maximum number of uncertain variables allowing the identification of robust solutions and the identification of solutions defining matchings (not perfect) or connected bipartite graphs. This chapter presents the cardinality-constrained robust PM and demonstrates that the CRPM is equivalent to the RPM and can be treated as such.

This chapter presents an experimental study to evaluate the performance of the deterministic and robust solutions. To evaluate the performance of the solutions, the performance measures defined in Chapter 3 are used. The feasibility level computes the proportion of the solutions among the uncertain set that define matchings and connected bipartite graphs. Properties of the measures of feasibility for different configurations of the RPM are also shown.

The remainder of this chapter is organized as follows. Section 4.2 introduces the PM as an equivalent AP. Section 4.3 presents the development of the RPM and its properties. Section crpmsec presents the CRPM and the demonstration of its equivalence with the RPM. Section 4.5 presents the experimental study of the deterministic and robust solutions of the PM under uncertainty. Section 4.6 presents concluding remarks for the chapter.

## 4.2 Minimum Weighted Bipartite Perfect Matching Problem

A *graph* is an ordered pair $G = (V, E)$ consisting of a nonempty set $V$ of *vertices*, and a set $E$ of pairs of vertices called *edges*. For an edge $e = (i, j)$, vertices $i$ and $j$ are the *endpoints* of $e$, and $e$ is said *incident to* and *connect* $i$ and $j$. A graph $G$ is named *bipartite* if the set of vertices $V$ can be divided into two disjoint sets $I$ and $J$ such that every edge possesses one endpoint in $I$ and another one in $J$. A bipartite graph is *complete* if every vertex in $I$ is connected to every vertex in $J$.

A set $M \subseteq E$ is called a *matching* if $\forall i \in V$ there exists at most one edge $e \in M$ incident to $i$. A vertex is *matched* if it is endpoint of one edge in $M$; otherwise, the vertex is *unmatched*. A matching is *perfect* if all vertices are matched; in a bipartite graph, $|I| = |J|$ is a necessary condition for the existence of a perfect matching.

Given a bipartite graph, and given a weight $c_{ij} \geq 0$ for each edge $e = (i, j)$, *the minimum weight bipartite perfect matching problem* (PM) consists of identifying the perfect matching $M$ with minimum total weight. Interested readers are referred to Bondy and

Murty (1976); Wolsey (1998) and West et al. (2001) for more information about the PM problem.

Consider a bipartite graph $G$, and the two vertex partitions $I = \{1, ..., n\}$ and $J = \{1, ..., n\}$. Let the binary decision variable $x_{ij}$ be defined as follows:

$$x_{ij} = \begin{cases} 1, & \text{if the edge } (i,j) \in M \\ 0, & \text{otherwise} \end{cases} \quad , \forall i \in I, j \in J. \tag{4.1}$$

The PM can be formulated as a binary linear programming problem as follows:

$$\min f(x) = \sum_{(i,j) \in I \times J} c_{ij} x_{ij} \tag{4.2}$$

$$\text{s.t.} \sum_{j \in J} x_{ij} = 1, \qquad \forall i \in I \tag{4.3}$$

$$\sum_{i \in I} x_{ij} = 1, \qquad \forall j \in J \tag{4.4}$$

$$x_{ij} \in \{0, 1\}, \qquad \forall i \in I, j \in J. \tag{4.5}$$

The objective function (4.2) represents the total weight of all the edges in $M$. Constraints (4.3) ensure that every vertex in $I$ is connected to exactly one vertex in $J$, and constraints (4.4) ensure that every vertex in $J$ is connected to exactly one vertex in $I$; constraints (4.5) are the binary constraints for the decision variables.

As it was mentioned in the introduction, the PM is equivalent to the assignment problem where the weight $c_{ij}$ corresponds to the cost of assigning the resource $i \in I$ to the job $j \in J$ and the objective is to minimize the cost of all the assignments; constraints (4.3) ensure that every resource performs one job, and constraints (4.4) ensure that every job is assigned to one resource.

### 4.3 Robust PM Under Implementation Uncertainty

#### 4.3.1 Model Development

Some of the edges forming the matching $M$ may not be available to be selected or some edges might be forced into the matching at the time of implementing the solution of the PM. In other words, variables $x_{ij}$ may be affected by implementation uncertainty. Definitions 2.2.1 and 2.2.2 of a binary variable under implementation uncertainty and the uncertain set can be applied to this problem.

Let $U \subset E$ be the set of edges $(i,j)$ such that the associated binary variable $x_{ij}$ is an uncertain variable, and assume there exists at least one uncertain variable, i.e. $|U| > 0$. Given a prescribed solution $\hat{x}$, the implemented solution $\tilde{x}$ may be any vector in $\mathcal{U}(\hat{x}_C)$ due to implementation uncertainty; however, $\tilde{x}$ may not remain optimal nor feasible.

To measure the level of objective robustness, the measure $\gamma$ can be used as defined in (3.1). However, the definition of $\delta$ as defined in (2.3.1) measures infeasibility as the excess of the left-hand-side over the right-hand-side; therefore, it cannot be directly applied in constraints (4.3) and (4.4) since they are equality constraints (i.e. an excess of the right-hand-side over the left-hand-side is also infeasible). This situation can be resolved by rewriting constraints (4.3) and (4.4) as inequality constraints as follows:

$$\sum_{j \in J} x_{ij} \leq 1, \forall i \in I \tag{4.6}$$

$$-\sum_{j \in J} x_{ij} \leq -1, \forall i \in I \tag{4.7}$$

$$\sum_{i \in I} x_{ij} \leq 1, \forall j \in J \tag{4.8}$$

$$-\sum_{i \in I} x_{ij} \leq -1, \forall j \in J. \tag{4.9}$$

Constraints (4.6) and (4.8) ensure that every vertex in $I$ and $J$ have at most one incident edge, respectively; in other words, these constraints may lead to matchings that are not perfect. The measures of feasibility robustness corresponding to constraints (4.6) and (4.8) are $\delta_i^{IL}$ and $\delta_j^{JL}$, respectively; they measure the excess of edges incident to vertices in $I$ and $J$. On the other hand, constraints (4.7) and (4.9) ensure that every vertex in $I$ and $J$ have at least one incident edge, respectively; in other words, each vertex in one partition is connected to at least one other in the other partition; i.e., connected bipartite graphs. The measures of feasibility robustness corresponding to constraints (4.7) and (4.9) are $\delta_i^{IG}$ and $\delta_j^{JG}$, respectively; they measure the deficit of edges incident to vertices in $I$ and $J$.

The robust PM (RPM) can be formulated as follows:

$$\min \quad \gamma \tag{4.10}$$

$$\text{s.t.} \max_{y \in \mathcal{U}(x_C)} \left\{ \sum_{(i,j) \in I \times J} c_{ij} x_{ij} \right\} \leq \gamma \tag{4.11}$$

$$\max_{y \in \mathcal{U}(x_C)} \left\{ \sum_{j \in J} y_{ij} \right\} - \delta_i^{IL} \leq 1, \qquad \forall i \in I \tag{4.12}$$

$$\max_{y \in \mathcal{U}(x_C)} \left\{ \sum_{j \in J} -y_{ij} \right\} - \delta_i^{IG} \leq -1, \qquad \forall i \in I \tag{4.13}$$

$$\max_{y \in \mathcal{U}(x_C)} \left\{ \sum_{i \in I} y_{ij} \right\} - \delta_j^{JL} \leq 1, \qquad \forall j \in J \tag{4.14}$$

$$\max_{y \in \mathcal{U}(x_C)} \left\{ \sum_{i \in I} -y_{ij} \right\} - \delta_j^{JG} \leq -1, \qquad \forall j \in J \tag{4.15}$$

$$\gamma \text{ is unrestricted} \tag{4.16}$$

$$0 \leq \delta_i^{IL} \leq \delta_i^{ILmax}, \qquad \forall i \in I \tag{4.17}$$

$$0 \leq \delta_i^{IG} \leq \delta_i^{IGmax}, \qquad \forall i \in I \tag{4.18}$$

$$0 \leq \delta_j^{JL} \leq \delta_j^{JLmax}, \qquad \forall j \in J \qquad (4.19)$$

$$0 \leq \delta_j^{JG} \leq \delta_j^{JGmax}, \qquad \forall j \in J \qquad (4.20)$$

$$x_{ij} \in \{0,1\}, \qquad \forall i \in I, j \in J. \qquad (4.21)$$

The objective function (4.10) is the objective robustness level; constraint (4.11) is the objective robustness constraint associated to the objective function (4.2); constraints (4.12) and (4.13) are the feasibility robustness constraints associated to equality constraints (4.3); similarly, constraints (4.14) and (4.15) are the feasibility robustness constraints associated to equality constraints (4.4); constraints (4.17) and (4.19) bound the acceptable excess of incident edges for all elements in $\mathcal{U}(x_C)$, and constraints (4.18) and (4.20) bound the acceptable deficit of incident edges for all elements in $\mathcal{U}(x_C)$. Constraints (4.21) represents the binary constraints of variables $x_{ij}$.

Let $|U_i^{'}|$ and $|U_j^{''}|$ be the number of uncertain variables in constraints (4.12)-(4.13) and (4.14)-(4.15), respectively, with $0 \leq |U_i^{'}| \leq |U|$, $0 \leq |U_j^{''}| \leq |U|$ for all $i \in I, j \in J$ and $\sum_{i \in I} |U_i^{'}| = \sum_{j \in J} |U_j^{''}| = |U|$. The equivalent linear reformulation of the RPM as shown Section 2.4 is the following:

$$\min \quad \gamma \qquad (4.22)$$

$$\text{s.t.} \quad \sum_{(i,j) \in (I \times J) \setminus U} c_{ij} x_{ij} + \sum_{(i,j) \in U} c_{ij} \leq \gamma \qquad (4.23)$$

$$\sum_{\{j \in J : (i,j) \notin U\}} x_{ij} - \delta_i^{IL} \leq 1 - |U_i^{'}| \qquad \forall i \in I \qquad (4.24)$$

$$\sum_{\{j \in J : (i,j) \notin U\}} x_{ij} + \delta_i^{IG} \geq 1 \qquad \forall i \in I \qquad (4.25)$$

$$\sum_{\{i \in I : (i,j) \notin U\}} x_{ij} - \delta_j^{JL} \leq 1 - |U_j^{''}| \qquad \forall j \in J \qquad (4.26)$$

$$\sum_{\{i \in I:(i,j) \notin U\}} x_{ij} + \delta_j^{JG} \geq 1 \qquad \forall j \in J \tag{4.27}$$

$$(4.16), (4.17), (4.18), (4.19), (4.20), (4.21).$$

Constraint (4.23) is the maximum-contribution objective robustness constraint associated with constraint (4.11). Constraints (4.24) and (4.25) are the maximum-contribution feasibility robustness constraints associated with constraints (4.12) and (4.13), respectively. Similarly, constraints (4.26) and (4.27) are the maximum-contribution feasibility robustness constraints associated with constraints (4.14) and (4.15) respectively.

To be able to identify robust solutions of the RPM, the following assumption is considered:

**Assumption 4.3.1.** *Each constraint* (4.23)*,* (4.24)*,* (4.25)*,* (4.26) *and* (4.27) *contains at least one certain variable.*

Note that if this assumption is not satisfied, there do not exist any value to be fixed and the feasibility of these constraints cannot be guaranteed. The following section show several properties of the RPM.

### 4.3.2 Properties of the RPM

This section presents properties of the RPM. The following results assume that Assumption 4.3.1 is satisfied. Lemma 4.3.1 provides a lower bound on the proportion of uncertain variables such that RPM has no feasible solutions.

**Lemma 4.3.1.** *RPM solutions do not exist if the proportion of uncertain variables is greater than* $1 - 1/n$*.*

*Proof.* Each variable $x_{ij}$ is present in exactly two constraints; once in the set of constraints (4.3), and once in constraints (4.4). Given that the number of constraints is $2n$ and each constraint should contain at least one certain variable to guarantee their feasibility, then there should exist at least $n$ certain variables. Since the number of decision variables is $n^2$, the minimum proportion of certain variables is $1/n$. If the proportion of uncertain variables is greater than $1 - 1/n$ then there exists at least one constraint with no certain variables; therefore, RPM solutions do not exist. $\square$

The behavior of the maximum proportion of the number of uncertain variables is exponential; the greater the size of the partitions the greater the level of uncertainty that RPM can handle in terms of the number of uncertain variables.

The following lemmas provide properties of the control parameters $\delta_i^{ILmax}$ and $\delta_i^{IGmax}, \forall i$, and $\delta_j^{JLmax}$ and $\delta_j^{JGmax}, \forall j$. Different values of these control parameter define different configurations of the RPM.

**Lemma 4.3.2.** *If there exist solutions of the RPM with $\delta_i^{IG} \geq 1, \forall i$ and $\delta_j^{JG} \geq 1, \forall j$, these solutions consist of all the certain variables equal to zero, i.e. $x_{ij} = 0$, $\forall (i,j) \in (I \times J) \backslash U$.*

*Proof.* Given that $\delta_i^{IG} \geq 1, \forall i$, $\delta_j^{JG} \geq 1, \forall j$, constraints (4.25) and (4.27) in the equivalent linear formulation are feasible with $\sum_{\{j \in J : (i,j) \notin U\}} x_{ij} \geq 0$ and $\sum_{\{i \in I : (i,j) \notin U\}} x_{ij} \geq 0$, respectively. Given that the objective is to minimize, the value of $\gamma$ can be minimized by setting at value zero as many certain variables as possible; then it follows that $\sum_{\{j \in J : (i,j) \notin U\}} x_{ij} = 0$ and $\sum_{\{i \in I : (i,j) \notin U\}} x_{ij} = 0$. Therefore, the solutions of the RPM consist of all the certain variables equal to zero. $\square$

Given that the right-hand-side of constraints (4.25), (4.27) and decision variables are integer, variables $\delta_i^{IG}$ and $\delta_j^{JG}$ are integer as well. Therefore, $\delta_i^{IG} < 1$ implies $\delta_i^{IG} = 0, \forall i$; similarly, $\delta_j^{JG} < 1$ implies $\delta_j^{JG} = 0, \forall j$.

**Lemma 4.3.3.** *If there exist solutions of the RPM with $\delta_i^{IG} = 0, \forall i$ and $\delta_j^{JG} = 0, \forall j$, these solutions contain exactly one certain variable with value 1 in each constraint* (4.24)*,* (4.25)*,* (4.26) *and* (4.27).

*Proof.* Given that $\delta_i^{IG} = 0$ and $\delta_j^{JG} = 0$ and decision variables are binary, constraints (4.25) are feasible with $\sum_{\{j \in J:(i,j) \notin U\}} x_{ij} \geq 1$ and constraints (4.27) are feasible with $\sum_{\{i \in I:(i,j) \notin U\}} x_{ij} \geq 1$. Given that the objective is to minimize, the value of $\gamma$ can be minimized by setting at value one as many certain variables as possible; then it follows that exactly one certain variable $x_{ij}$ with a value 1 is necessary to satisfy each constraint (4.25) and (4.27). Therefore, the solutions of the RPM consist of exactly one certain variable with value 1 in each constraint (4.24), (4.25), (4.26) and (4.27). □

Proofs of Lemmas 4.3.2 and 4.3.3 assume that $\delta_i^{IL}$ and $\delta_j^{JL}$ can take any nonnegative value such that constraints (4.24) and (4.26) are feasible for any value of $\sum_{\{j \in J:(i,j) \notin U\}} x_{ij}$ and $\sum_{\{i \in I:(i,j) \notin U\}} x_{ij}$, respectively. Lemma 4.3.2 considers that $\delta_i^{IL} \geq |U_i^{'}| - 1, \forall i$ to satisfy constraints (4.24) when $\delta_i^{IG} \geq 1$. On the other hand, Lemma 4.3.3 considers that $\delta_i^{IL} \geq |U_i^{'}|, \forall i$ to satisfy constraints (4.24) when $\delta_i^{IG} = 0, \forall i$. One can conclude that by setting $\delta_i^{ILmax} \geq |U_i^{'}|, \forall i$, constraints (4.24) are satisfied for any value of $\delta_i^{IG}$. Similarly, constraints (4.26) with $\delta_j^{JLmax} \geq |U_j^{''}|, \forall j$ are satisfied for any value of $\delta_j^{JG}$.

Solutions of the RPM as described in Lemma 4.3.2 define matchings not necessarily perfect; these solutions can be achieved by making $\delta_i^{IGmax} \geq 1$ and $\delta_j^{JGmax} \geq 1$. On the other hand, Solutions of the RPM as described in Lemma 4.3.3 define connected bipartite graphs; these solutions can be achieved by making $\delta_i^{IGmax} = 0$ and $\delta_j^{JGmax} = 0$. From a practical perspective, solutions providing connected bipartite graphs are more appropriate; although they do not produce a match, they guarantee the connectivity of the bipartite network. In the context of the assignment problem, connectivity guarantee that every job is assigned to at least one resource and that they will be completed somehow.

## 4.4 Cardinality-Constrained Robust Formulation of the PM

This section presents the cardinality-constrained formulation of the PM (CRPM) and the proof of its equivalence with the RPM.

The CRPM as shown in Section 2.5.2 can be formulated as follows:

$$\min \quad \gamma \tag{4.28}$$

$$\text{s.t.} \sum_{(i,j)\in(I\times J)\setminus U} c_{ij}x_{ij} + \max_{\{S_0:S_0\subseteq U,|S_0|\leq\Gamma\}} \left\{ \sum_{(i,j)\in S_0} \left( \frac{c_{ij}+|c_{ij}|}{2} \right) + \sum_{(i,j)\in U\setminus S_0} c_{ij}x_{ij} \right\} \leq \gamma \tag{4.29}$$

$$\sum_{\{j\in J:(i,j)\notin U\}} x_{ij} + \max_{\left\{S_i':S_i'\subseteq U_i',|S_i'|\leq\Gamma\right\}} \left\{ |S_i'| + \sum_{(i,j)\in U_i'\setminus S_i'} x_{ij} \right\} - \delta_i^{IL} \leq 1, \forall i \in I \tag{4.30}$$

$$\sum_{\{j\in J:(i,j)\notin U\}} x_{ij} - \max_{\left\{S_i'':S_i''\subseteq U_i'',|S_i''|\leq\Gamma\right\}} \left\{ - \sum_{(i,j)\in U_i''\setminus S_i''} x_{ij} \right\} - \delta_i^{IG} \geq 1, \forall i \in I \tag{4.31}$$

$$\sum_{\{i\in I:(i,j)\notin U\}} x_{ij} + \max_{\left\{S_j''':S_j'''\subseteq U_j',|S_j'''|\leq\Gamma\right\}} \left\{ |S_j'''| + \sum_{(i,j)\in U_j'\setminus S_j'''} x_{ij} \right\} - \delta_j^{JL} \leq 1, \forall j \in J \tag{4.32}$$

$$\sum_{\{i\in I:(i,j)\notin U\}} x_{ij} - \max_{\left\{S_j'''':S_j''''\subseteq U_j'',|S_j''''|\leq\Gamma\right\}} \left\{ - \sum_{(i,j)\in U_j''\setminus S_j''''} x_{ij} \right\} - \delta_j^{JG} \geq 1, \forall j \in J \tag{4.33}$$

(4.16), (4.17), (4.18), (4.19), (4.20), (4.21).

Corollary 4.4.1 is the result of the structural properties of the PM and the results shown in Section 2.5.2.3 applied to the CRPM.

**Corollary 4.4.1.** *The CRPM is equivalent to the RPM.*

*Proof.* Consider constraints (4.30); the maximum value of the term $|S_i'| + \sum_{(i,j)\in U_i'\setminus S_i'} x_{ij}$ for any value of $\Gamma$ is given by $x_{ij} = 1, \forall (i,j) \in U_i' \setminus S_i'$. Therefore $|S_i'| + \sum_{(i,j)\in U_i'\setminus S_i'} x_{ij} = |S_i'| + |U_i' \setminus S_i'| = |U_i'|$ and $\max_{\left\{S_i':S_i'\subseteq U_i',|S_i'|\leq\Gamma\right\}} \left\{ |S_i'| + \sum_{(i,j)\in U_i'\setminus S_i'} x_{ij} \right\} = |U_i'|$, independent of $\Gamma$. Constraint (4.30) can be simplified to $\sum_{\{j\in J:(i,j)\notin U\}} x_{ij}+|U_i^{'}|-\delta_i^{IL}\leq 1,\forall i\in I$, which is equivalent to constraint (4.24). Constraint (4.32) can proof to be equivalent to constraint (4.26) in a similar form.

In constraints (4.31), the maximum value of $-\sum_{(i,j)\in U_i''\setminus S_i''} x_{ij}$ for any value of $\Gamma$ is given by $x_{ij}=0,\forall(i,j)\in U_i^{''}\setminus S_i^{''}$; then $\max_{\{S_i'':S_i''\subseteq U_i'',|S_i''|\leq\Gamma\}}\left\{-\sum_{(i,j)\in U_i''\setminus S_i''} x_{ij}\right\}=0$. Constraint (4.31) can be simplified to $\sum_{\{j\in J:(i,j)\notin U\}} x_{ij}-\delta_i^{IG}\geq 1,\forall i\in I$, which is equivalent to constraint (4.25). Constraint (4.33) can proof to be equivalent to constraint (4.27) in a similar form.

Given that $c_{ij}\geq 0,\forall(i,j)\in I\times J$, the maximum value of the term $\sum_{(i,j)\in S_0}\left(\frac{c_{ij}+|c_{ij}|}{2}\right)+\sum_{(i,j)\in U\setminus S_0} c_{ij}x_{ij}$ is given by $x_{ij}=1,\forall(i,j)\in U\setminus S_0$ for any value of $\Gamma$. Therefore $\sum_{(i,j)\in S_0}\left(\frac{c_{ij}+|c_{ij}|}{2}\right)+\sum_{(i,j)\in U\setminus S_0} c_{ij}x_{ij}=\sum_{(i,j)\in S_0}\left(\frac{c_{ij}+|c_{ij}|}{2}\right)+\sum_{(i,j)\in U\setminus S_0} c_{ij}=\sum_{(i,j)\in U} c_{ij}$. Constraint (4.29) can be simplified to $\sum_{(i,j)\in(I\times J)\setminus U} c_{ij}x_{ij}+\sum_{(i,j)\in U} c_{ij}\leq\gamma$, which is equivalent to constraint (4.23).

Therefore, the CRPM is equivalent to the linear reformulation of the RPM. □

By Theorem 2.4.1 and Corollary 4.4.1, all the properties and results developed for the RPM apply to the CRPM.

## 4.5 Experimental Study

This section presents the experimental study for the PM under implementation uncertainty. The objective of this study is to illustrate the performance of the deterministic and robust solutions under uncertainty.

### 4.5.1 Performance Measures

We consider the performance measures $h(x)$ and $l(x,y)$ defined in Section 3.4.1 to measure the performance of the deterministic and robust solutions of the PM under implementation uncertainty.

The feasibility ration $h(x)$ exhibits particular properties when used to measure the feasibility level of connected bipartite graphs or perfect matchings. The following sections show these properties.

#### *4.5.1.1 Feasibility Ratio for Connected Bipartite Graphs*

As it was mentioned in Section 4.3.2, an RPM with $\delta_i^{IG} < 1, \forall i$ and $\delta_j^{JG} < 1, \forall j$, and $\delta_i^{IL} \geq |U_i^{'}|, \forall i$ and $\delta_j^{JL} \geq |U_j^{''}|, \forall j$ produces connected bipartite graphs. Let $h'(x)$ be the feasibility ratio $h(x)$ such that $h'(x)$ considers feasible solutions as solutions producing connected bipartite graphs. The following lemmas present properties of $h'(x)$.

**Lemma 4.5.1.** *Let $x'$ be a RPM solution satisfying conditions in Lemma 4.3.2. Then $h'(x') < 1$.*

*Proof.* By Lemma 4.3.2, $x'$ consists of all the certain variables fixed at value 0. Given that $\mathcal{U}(x'_C)$ contains all combinations of the uncertain variables, $\mathcal{U}(x'_C)$ contains a vector $y$ with all certain variables equal to 0, and $y$ consists of all variables with value 0. Therefore, $\mathcal{F}$ contains at least one element do not producing a connected bipartite graph; then, $|\mathcal{F}| < |\mathcal{U}(x'_C)|$ and $h'(x') < 1$. □

**Lemma 4.5.2.** *Let $x^R$ be a RPM solution satisfying conditions in Lemma 4.3.3. Then $h'(x^R) = 1$.*

*Proof.* By Lemma 4.3.3, $x^R$ contains exactly one certain variable at value 1 for each constraint and these values are fixed. Therefore, Lemma 4.5.2 holds. □

**Corollary 4.5.1.** $h'(x') < h'(x^R)$.

*Proof.* Corollary 4.5.1 holds from Lemmas 4.5.1 and 4.5.2. □

**Lemma 4.5.3.** *Let $x^D$ be the deterministic solution of the PM. Then $h'(x^D) \leq 1$.*

*Proof.* Due to constraints (4.3) and (4.4), $x^D$ is a perfect matching. If at least one of its variables with value 1 is uncertain, by implementation uncertainty it may change to 0; therefore, $h'(x^D) < 1$. On the other hand, if all the variables with value 1 in $x^D$ are certain, then their values are fixed and $h'(x^D) = 1$. Therefore, $h'(x^D) \leq 1$. □

**Corollary 4.5.2.** $h'(x^D) \leq h'(x^R)$.

*Proof.* Corollary 4.5.2 holds from Lemmas 4.5.2 and 4.5.3. □

*4.5.1.2 Feasibility Ratio for Perfect Matchings*

Given a RPM solution $x^*$, each element $y \in \mathcal{U}(x^*_C)$ produces a different value of $f$. Given that $c_{ij} \geq 0$, the solution $x^{LB}$ producing the lower bound of $f$ is formed as $x^{LB} = (x^*_C, x^{LB}_U)$, with every uncertain variable $x_{ij}$ in vector $x^{LB}_U$ defined as $x_{ij} = 0, \forall (i,j) \in U$ (see Proposition 2.5.2).

**Lemma 4.5.4.** *Let $x^{LB}$ be the RPM solution providing the lower bound of $f$ among the elements of $\mathcal{U}(x^{LB}_C)$, and $x^{LB}$ satisfies Lemma 4.3.3. Then $x^{LB}$ defines a perfect matching.*

*Proof.* By Lemma 4.3.3, there exists exactly one certain variable with the value of one in each constraint and $x_{ij} = 0, \forall (i,j) \in U$. Therefore, $x^{LB}$ defines a perfect matching. □

Consider the PM formulation and let $h''(x)$ be the feasible ratio $h(x)$ considering feasible solutions of the PM; i.e., perfect matchings. The following lemmas present properties of $h''(x)$.

**Lemma 4.5.5.** $h''(x^{LB}) = 1/\left(2^{|U|}\right)$.

*Proof.* By Lemma 4.5.4, the solution producing the lower bound of $f$ is a perfect matching. Since all other elements in $\mathcal{U}(x^{LB}_C)$ contains at least one uncertain variable with value 0, only $x^{LB}$ is a perfect matching, i.e. $|\mathcal{F}| = 1$. Therefore, $h''(x^{LB}) = 1/\left(2^{|U|}\right)$. □

**Lemma 4.5.6.** $h''(x^D) \geq 1/\left(2^{|U|}\right)$.

*Proof.* Due to constraints (4.3) and (4.4), $x^D$ is a perfect matching. If $x^D$ contains at least two uncertain variables in each constraint and one of them takes a value 1, then there may exist more than one combination of the uncertain variables such that only one of them takes a value 1. Therefore, there may exist more than one solution that is a perfect matching, i.e. $|\mathcal{F}| \geq 1$. □

**Lemma 4.5.7.** $h''(x^D) \leq U_{min}/\left(2^{|K|}\right)$*, where* $U_{min} = \min\left\{|U_i'|, |U_j''|\right\}$.

*Proof.* Since $U_{min}$ is the minimum number of uncertain variables among all the constraints, there may exist at most $U_{min}$ combinations of the uncertain variables with exactly one uncertain variable with value one. Therefore, $h''(x^D) \leq U_{min}/\left(2^{|K|}\right)$. □

Corollary 4.5.3 states that the feasibility level of $x''$ is better than $x^D$ with respect to perfect matchings; this is as expected because to protect against uncertainty, the RPM solutions provide redundant edges for each vertex and under implementation uncertainty the number of edges for every vertex is at least one. On the other hand, PM solutions seek for one edge for each vertex and several uncertainty conditions may lead to multiple combinations providing exactly one edge.

**Corollary 4.5.3.** $h''(x^{LB}) \leq h''(x^D)$.

*Proof.* Corollary 4.5.3 follows from Lemmas 4.5.5 and 4.5.6. □

### 4.5.2 Test Problem Generation

An instance of the PM is given by the costs $c_{ij}$. This experiment study considers two sets of assignments problems with the following characteristics:

- Set I (low variance): cost $c_{ij}$ uniformly distributed between $[450, 550]$.

- Set II (high variance): cost $c_{ij}$ uniformly distributed between $[1, 1000]$.

Set I consists of costs with low variance, while Set II consists of costs with high variance. The objective is to observe how the variance of the costs affects the performance of the robust solutions.

Each set of problems consists of 30 problems with $n = 5$ vertices in each set $I$ and $J$. Each problem is solved for every $|U| = 1, ..., 20$ (the maximum number of uncertain variables for $n = 5$ according to Lemma 4.3.1). Solutions $x^D$ and $x^R$ are obtained by solving the PM and RPM formulations, respectively; $x^R$ is an RPM solution satisfying Lemma 4.3.3. The corresponding robust-optimal solutions sets $\mathcal{U}(x^D)$, $\mathcal{U}(x^R)$, feasible solutions sets $\mathcal{F}(x^D)$ and $\mathcal{F}(x^R)$, and performance measures are obtained via complete enumeration. It is assumed that the implemented value of each uncertain variable can be different or equal to the prescribed value with same probability independent of the prescribed value, i.e. $p_i = q_i = 0.5, \forall i$.

### 4.5.3 Performance Results

Figure 4.1 shows the loss in objective value of the solutions of the RPM versus increasing number of uncertain variables $|U|$ for the two sets of problems. High values in this graph represents a worse performance in the value of the objective function compared to the deterministic solution. It can be observed that the objective value of solutions of the RPM degrade as the uncertainty increases (increment in the number of uncertain variables $|U|$). For example, for low variance problems with one uncertain variable, the performance of the RPM is similar to the PM (degradation is almost zero), while for the same set of problems with 15 uncertain variables the degradation of the value of the objective function is around 30%. The robust solutions for problems with low variance perform slightly better than robust solutions for problems with high variance; although, their behavior is similar.

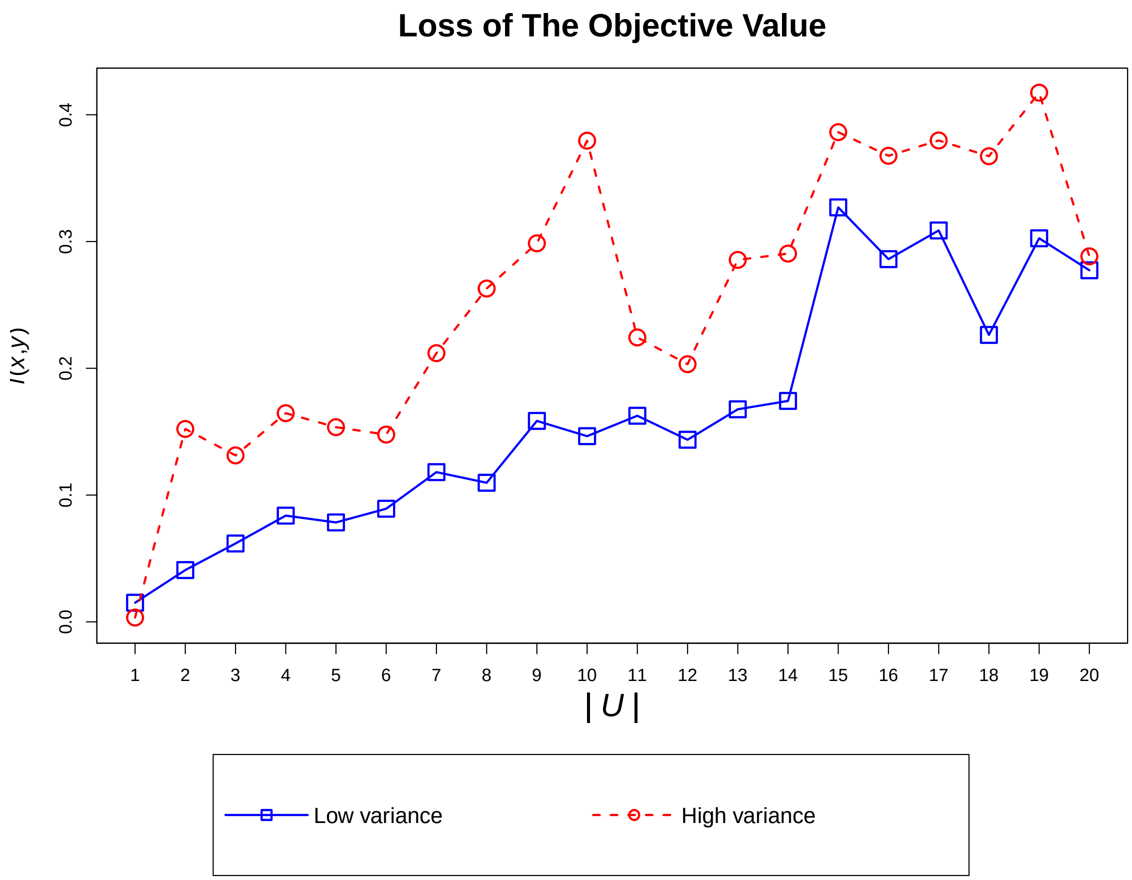



Figure 4.1: Loss of the objective performance ratio $l(x, y)$.

Given that variance of the cost values does not affect the constraints, there do not exist significant differences in the feasibility performance of the two sets of problems. Therefore, Figure 4.2 only shows the results for the low variance set of problem.

Figure 4.2 shows the feasibility ratios for perfect matchings, $h'(x^D)$ and $h'(x^R)$, and the feasibility ratios for connected bipartite graphs, $h''(x^D)$ and $h''(x^R)$. High values represent a better feasible performance. With respect to perfect matchings, Figure 4.2 shows that solutions of the PM and RPM perform similarly; the small improvement in the performance of $x^D$ over $x^R$ is not significant. This plot verifies the result described in Corollary 4.5.3 that deterministic solutions produce better feasibility performance than the robust

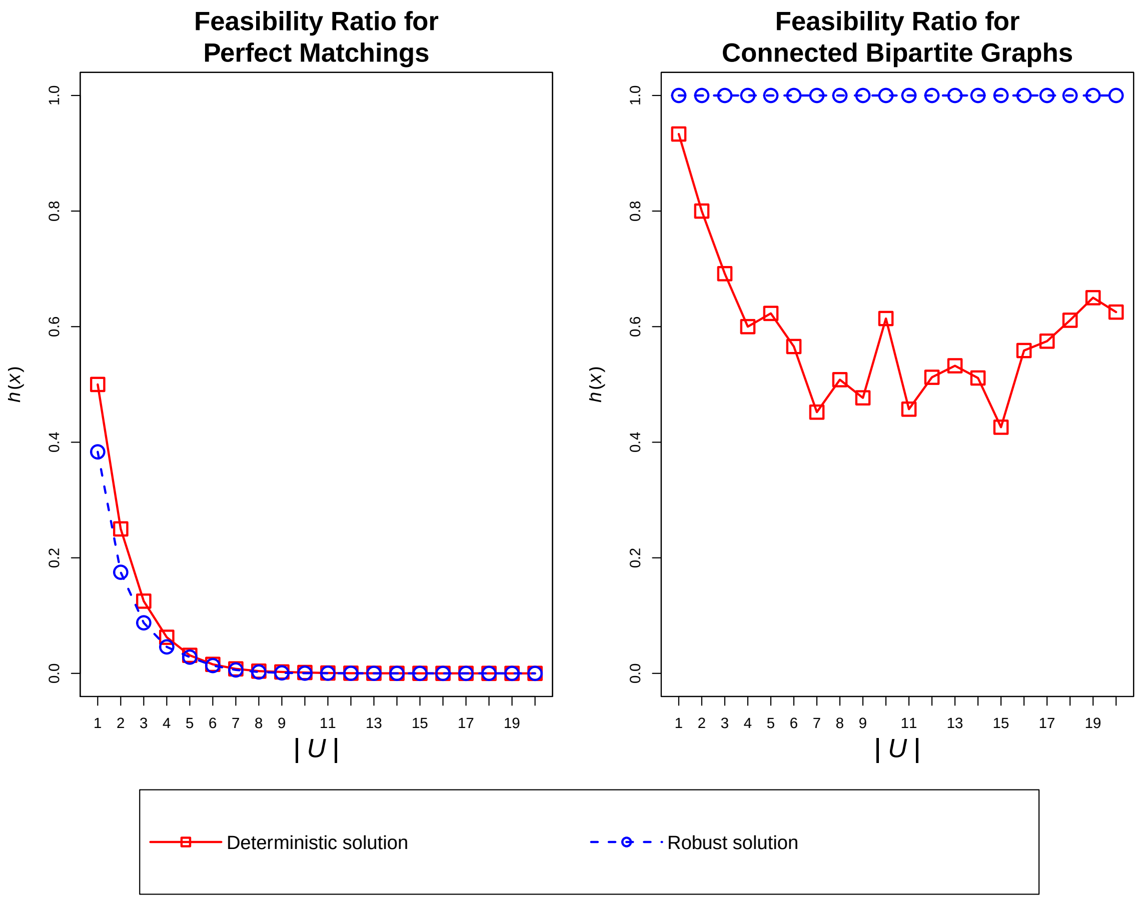



Figure 4.2: Feasibility performance ratio with respect to perfect matchings $h''(x)$ and connected bipartite graphs $h'(x)$.

solutions with respect to perfect matchings. Notice that this result is not surprising given that the equality constraints in the perfect matching formulation will be violated if any prescribed variable changes its value at the time of implementation. On the other hand, with respect to connected bipartite graphs, it can be observed that RPM solutions guarantee a 100% feasibility for any occurrence of uncertain variables, while the performance of the solutions of the PM degrades as the number of uncertain variables increases. For instance, the feasibility ratio of the solution of the PM is 93% for $|U| = 1$ and decreases to almost 40% for $|U| = 15$. This plot verifies the results in Corollary 4.5.2 that robust solutions produce better feasibility ratio than the deterministic solutions with respect to connected bipartite graphs. In summary, these results suggest the value of robust solutions in situations where feasibility is important, and connected bipartite solutions may be acceptable when a perfect matching is affected by implementation uncertainties.

Let $x^{LB}$ be the solution producing the lower bound of the value of $f$ among the elements of $\mathcal{U}(x_C)$, let $x^{UB}$ be the solutions producing the upper bound of the value of $f$, and let $x^{AVG}$ be the solutions producing the average value of the objective function computed by total enumeration. The losses of optimality of these solutions with respect to the deterministic solution $l(x^D, x^{LB})$, $l(x^D, x^{UB})$ and $l(x^D, x^{AVG})$ are shown in Figure 4.3.

It can be observed that the solution producing the lower bound of the objective function performs better than the other two, while the solution providing the upper bound value of the objective function delivers the worst performance. For instance, for the set of problems with low variance, the lower bound provides a small increment in the cost even for $|U| = 20$ uncertain variables, where the maximum increment is only 6%; for the same set of problems, the upper bound increases the cost from a minimum percentage of 21% with one uncertain variable to a maximum value of 430% with $|U| = 20$ uncertain variables; the solution providing the average value of the objective function produces a minimum cost increment of 10% to a maximum value of 217%. These results together with Lemma 4.5.4 show that the lower bound solution provides the smallest increment in the loss of the value of the objective function while it is simultaneously a perfect matching; these characteristics of the solution producing the lower bound of the objective function makes it the most attractive RPM solution.

One can conclude that RPM solutions are more appropriate when changes in the prescribed values likely to happen since they guarantee the existence of connected bipartite graphs, which are meaningful solutions from the assignment problem point of view (see Figure 4.2). The performance of RPM solutions for perfect matchings is similar to the

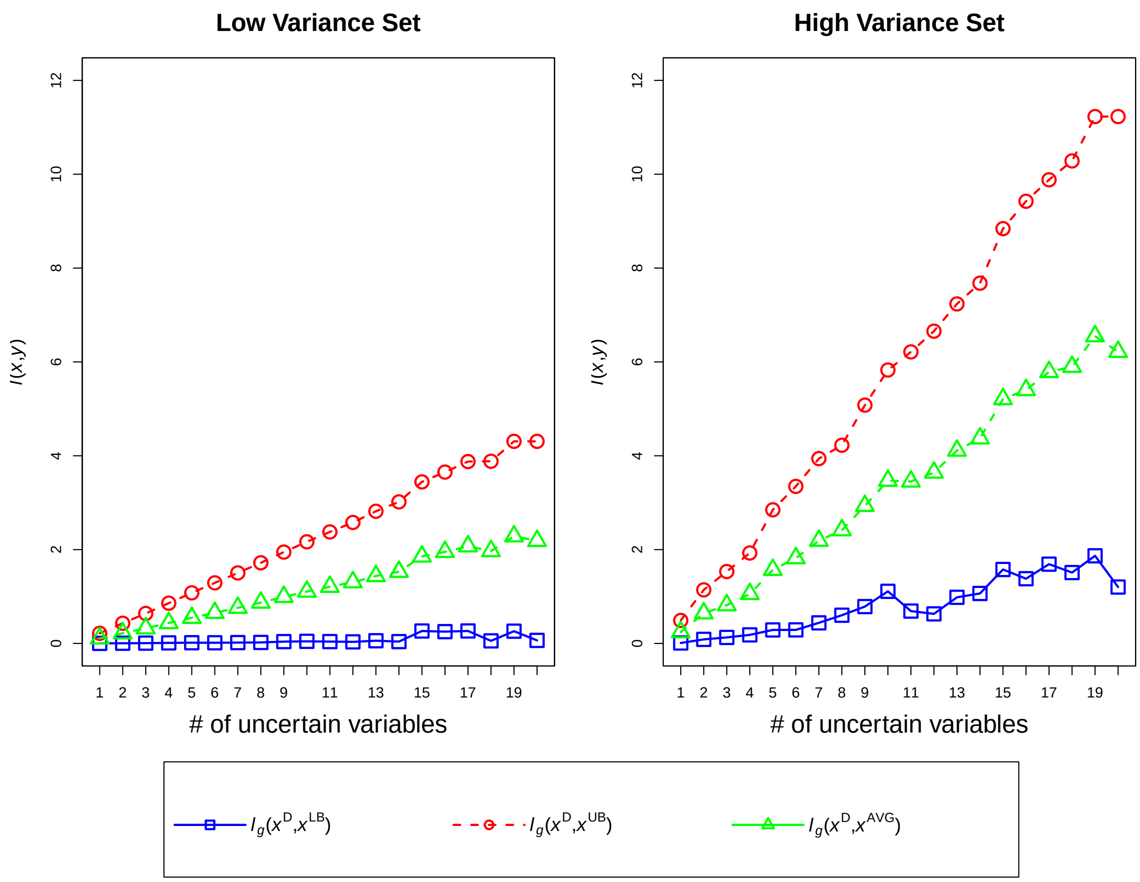



Figure 4.3: Loss of the objective performance ratio $l(x, y)$ for lower bound, upper bound and average value solutions.

RPM solutions (see Figure 4.2); although, there is a sacrifice in the cost due to the extra edges (see Figure 4.1). Furthermore, using the RPM solutions producing the lower bound of the objective value guarantee a small loss of optimality for problems whose costs possess low variance, and offers the minimum optimality degradation among other solutions in the robust-optimal solution set (see Figure 4.3).

In summary, the experimental results suggest that depending on the interest of the decision maker, robust solutions may be more attractive than the deterministic solutions. Following are some practical recommendations based on the experimental results:

- If the decision maker is interested in obtaining perfect matching, robust solutions do not provide any advantage given that their feasibility ratio behaves similar (see Figure 4.2) and robust solutions offer a smaller objective function value.

- In contrast, if the decision maker is interested in obtaining connected bipartite graphs, robust solutions are more attractive given that their feasibility ratio does not decrease when impacted by uncertainty (see Figure 4.2).

- The use of the robust solution providing the lower bound of the value of the objective function among the robust-optimal solution set provides a less conservative value than other solutions within the set or robust solutions (see Figure 4.3).

### 4.6 Summary and Conclusions

This chapter presents a robust formulation for the PM under implementation uncertainty such that it identifies solutions satisfying a given level of feasibility while the degradation of the cost is minimum. This work shows configurations of the RPM that produce solutions defining matchings or connected bipartite graphs for any realization of the uncertainty. The CRPM is shown to be equivalent to the RPM; the configurations of the RPM are defined by the control parameters. The CRPM is shown to be equivalent to the RPM and the control parameters define the configurations of the RPM. Therefore, for the RPM the only method to control the conservatism is the selection of the least conservative solutions from the robust-optimal solution set, which is illustrated in the experimental study.

The results of the experimental study show that the higher the variance of the costs of the assignments the greater the loss of optimality. This results also show that the deterministic and robust solutions posses a similar feasibility level when considering matchings. On the other hand, the feasibility level of the deterministic solution is significantly degraded when considering connected bipartite graphs. Finally, the results of the experiments show that the difference between of upper and lower bound of the objective value increases as the variance of the costs of the assignments increases as well.

# 5. ROBUST SHORTEST PATH PROBLEM IMPLEMENTATION UNCERTAINTY

## 5.1 Introduction

This chapter presents the SPP under implementation uncertainty. The impact of uncertainty is addressed by applying the proposed concepts of robustness. Significant theoretical and practical results associated with the robust formulation and robust solutions for this problem are also shown.

The SPP under implementation uncertainty considers that some arcs of the network may not be available at the time of implementing the solution. The impediment to moving through certain arcs may be the result of conditions not considered initially such as disasters, traffic accidents, etc. For example, Section 5.6 presents the case study of a transportation network where accidents or traffic conditions block roads impeding to reach the destination location or waiting times to continue traveling may be too long. Unexpected changes in the arcs selected may lead to a disconnected path between the source and destination nodes, or to a too costly path connecting these nodes. Existing work in robust optimization studying the SPP accounts for parameter uncertainty only (e.g. Kouvelis and Yu, 1997; Yu and Yang, 1998; Bertsimas and Sim, 2003). The SPP under implementation uncertainty is addressed by applying the proposed concepts of robustness.

The proposed concepts of robustness in Chapter 2 permit to develop the corresponding RSPP and its equivalent linear reformulation. This work identifies properties of the control parameters of the RSPP including the configuration of the control parameters to obtain an RSPP equivalent to a reduce SPP free of uncertainty, and to obtain robust solutions connecting every node destination of an arc associated to an uncertain variable with the destination node of the deterministic SPP. The equivalent linear reformulation can be used to identify the set of robust solutions. An algorithm to solve the RSPP based on the work in Dijkstra (1959); Fredman and Tarjan (1987) is presented. The proposed algorithm can be applied to directed and undirected graphs that do not contain cycles with negative arc weights. This work proofs the correctness of the proposed algorithm and discusses the its complexity. Similar to the case of the AP, the cardinality-constrained robust SPP under implementation uncertainty is shown to be equivalent to the RSPP.

An experimental study of the SPP under implementation uncertainty aims to evaluate and compare the performance of the deterministic and robust solutions. The performance measure of the objective function value computes the cost of the minimum path among all the possible solutions paths connecting the source and destination nodes among the uncertain set. The performance measure of feasibility level computes the proportion of the paths in the uncertain set that connect the source and destination nodes. The results of the experimental study show the characteristics of the network that makes the deterministic solutions more sensitive to implementation uncertainty and the robust solutions more attractive.

This chapter ends presenting a case study for a real application of the SPP under implementation. The case study involves a distribution company moving products in the East of United States. The results of the case study determine alternative routes that eliminate the possible impediment of traveling through the initial deterministic path due to accidents or other traffic conditions.

The remainder of this chapter is organized as follows. Section 5.2 formulates the deterministic SPP. Section 5.3 develops the RSPP and presents its properties. Section 5.4 presents an algorithm to identify the solutions of the RSPP. Section 5.6 presents the experimental study, and Section 5.8 presents concluding remarks for the chapter.

## 5.2 Deterministic Shortest Path Problem

Consider a digraph $G = (N, A)$, where $N$ is the set of nodes and $A$ is the set of arcs. Let $s \in N$ be the source node and $d \in N$ be the destination node, and let $c_{ij} \geq 0$ be the cost of arc $(i, j) \in A$. The objective of the SPP is to find the path from $s$ to $d$ such that its total cost is minimum.

Let $x_{ij}$ be a binary variable such that $x_{ij} = 1$ if arc $(i, j)$ is included in the path and $x_{ij} = 0$ otherwise. Wolsey (1998) formulates the SPP from $s$ to $d$ as follows:

$$\min \sum_{(i,j)\in A} c_{ij}x_{ij} \tag{5.1}$$

$$\text{s.t.} \sum_{(s,i)\in A, i\neq s} x_{si} - \sum_{(i,s)\in A, i\neq s} x_{is} = 1 \tag{5.2}$$

$$\sum_{(d,i)\in A, i\neq d} x_{di} - \sum_{(i,d)\in A, i\neq d} x_{id} = -1 \tag{5.3}$$

$$\sum_{(j,i)\in A, i\neq j} x_{ji} - \sum_{(i,j)\in A, i\neq j} x_{ij} = 0, \forall j \in N \setminus s, d \tag{5.4}$$

$$x_{ij} \in \{0, 1\}, \forall (i, j) \in A \tag{5.5}$$

The objective function (5.1) seeks to minimize the total cost of the path from node $s$ to node $d$. Constraint 5.2 guarantees that there exists exactly one arc whose source is node $s$; therefore, it guarantees that there is a path from $s$ to at least another node. Similarly, constraint (5.3) guarantees that there exists exactly one arc whose destination is node $d$; therefore, it guarantees that there is a path from any node to $d$. Constraint (5.4) are the flow conservation constraint for transshipment nodes guaranteeing that the in-degree and out-degree of these nodes are equal; if a node is destination of an arc in the path, then there is another arc whose destination is the transshipment node such that the path from $s$ to $d$ continues. Constraint (5.5) is the binary constraint for the decision variables.

The deterministic solution of the SPP avoids redundant arcs to reduce the total cost of the path connecting $s$ and $d$; therefore, the deterministic solution defines a single path from $s$ to $d$ whose cost is minimum. This path may be disconnected if at least one of the constraints (5.2), (5.3) and (5.4) are violated; this is an arc forming the path is not available at the time of implementation making impossible to travel from the source node to the destination node. This situation is addressed in the following section.

## 5.3 Robust Shortest Path Problem Under Implementation Uncertainty

### 5.3.1 Model Development

Under implementation uncertainty, existing arcs may not be available at the time of the implementation and the vehicle may not be able to travel through them, or the vehicle may be forced to move through arcs not initially considered. Due to implementation uncertainty, an existing path may not be able to connect the source and destination nodes. It is of interest to find robust solutions to the SPP under implementation uncertainty such that they guarantee the existence of a path connecting the source and destination nodes for any occurrence of the uncertainty. To identify robust solutions, the following assumption is considered:

**Assumption 5.3.1.** *For every node $i \in N$, there exists at least one arc associated to a certain variable whose source is $i$ and at least one arc associated to a certain variable whose destination is $i$.*

Assumption 5.3.1 is somewhat relaxed for the source and destination nodes as follows:

**Assumption 5.3.2.** *For the source node $s$, there exists at least one arc associated to a certain variable whose source is $s$. For the destination node $d$, there exists at least one arc associated to a certain variable whose destination is $d$.*

The equality constraints (5.13), (5.14) and (5.15) of the SPP are rewritten as inequalities as follows:

$$\sum_{(s,i)\in A, i\neq s} x_{si} - \sum_{(i,s)\in A, i\neq s} x_{is} \leq 1 \tag{5.6}$$

$$-\sum_{(s,i)\in A, i\neq s} x_{si} + \sum_{(i,s)\in A, i\neq s} x_{is} \leq -1 \tag{5.7}$$

$$\sum_{(d,i)\in A, i\neq d} x_{di} - \sum_{(i,d)\in A, i\neq d} x_{id} \leq -1 \tag{5.8}$$

$$-\sum_{(d,i)\in A, i\neq d} x_{di} - \sum_{(i,d)\in A, i\neq d} x_{id} \leq 1 \tag{5.9}$$

$$\sum_{(j,i)\in A, i\neq j} x_{ji} - \sum_{(i,j)\in A, i\neq j} x_{ij} \leq 0, \forall j \in N \setminus s, d \tag{5.10}$$

$$-\sum_{(j,i)\in A, i\neq j} x_{ji} + \sum_{(i,j)\in A, i\neq j} x_{ij} \leq 0, \forall j \in N \setminus s, d \tag{5.11}$$

Consider the Definitions 2.2.1 and 2.2.2 of a binary variable under implementation uncertainty and the uncertain set, respectively, and consider the measures of robustness in Definitions 2.3.1 and 2.3.2. The RSPP can be formulated as follows:

$$\min \quad \gamma \tag{5.12}$$

$$\text{s.t.} \max_{y\in\mathcal{U}(x_c)} \left\{ \sum_{(i,j)\in A} c_{ij} y_{ij} \right\} \leq \gamma \tag{5.13}$$

$$\max_{y\in\mathcal{U}(x_c)} \left\{ \sum_{(s,i)\in A, i\neq s} y_{si} - \sum_{(i,s)\in A, i\neq s} y_{is} \right\} - \delta_s^L \leq 1 \tag{5.14}$$

$$\max_{y\in\mathcal{U}(x_c)} \left\{ \sum_{(s,i)\in A, i\neq s} -y_{si} + \sum_{(i,s)\in A, i\neq s} y_{is} \right\} - \delta_s^G \leq -1 \tag{5.15}$$

$$\max_{y\in\mathcal{U}(x_c)} \left\{ \sum_{(d,i)\in A, i\neq d} y_{di} - \sum_{(i,d)\in A, i\neq d} y_{id} \right\} - \delta_d^L \leq -1 \tag{5.16}$$

$$\max_{y \in \mathcal{U}(x_c)} \left\{ \sum_{(d,i) \in A, i \neq d} -y_{di} + \sum_{(i,j) \in A, i \neq d} y_{id} \right\} - \delta_d^G \leq 1 \tag{5.17}$$

$$\max_{y \in \mathcal{U}(x_c)} \left\{ \sum_{(j,i) \in A, i \neq j} y_{ji} - \sum_{(i,j) \in A, i \neq j} y_{ij} \right\} - \delta_j^L \leq 0, \forall j \in N \setminus \{s, d\} \tag{5.18}$$

$$\max_{y \in \mathcal{U}(x_c)} \left\{ \sum_{(j,i) \in A, i \neq j} -y_{ji} + \sum_{(i,j) \in A, i \neq j} y_{ij} \right\} - \delta_j^G \leq 0, \forall j \in N \setminus \{s, d\} \tag{5.19}$$

$$\gamma \text{ is unrestricted} \tag{5.20}$$

$$0 \leq \delta_j^L \leq \delta_j^{L_{max}}, \forall j \in N \tag{5.21}$$

$$0 \leq \delta_j^L \leq \delta_j^{G_{max}}, \forall i \in N \tag{5.22}$$

$$x_{ij} \in \{0, 1\}, \forall (i, j) \in A \tag{5.23}$$

The objective function (5.12) seeks to minimize the worst-case value of the cost of the shortest path from $s$ to $d$ when affected by implementation uncertainty. Constraint (5.13) is the objective robustness constraint that guarantees that $\gamma$ will be greater than the worst-case cost. Constraints (5.14) and (5.15) are the feasibility robustness constraints associated with the source node $s$ that guarantees that there exists one arc with node $s$ as the source. Constraints (5.16) and (5.17) are the feasibility robustness constraints associated with the destination node $d$ that guarantees that there exists one arc with node $d$ as the destination. Constraints (5.18) and (5.19) are the robustness flow conservation constraints for the transshipment nodes and guarantee the flow through the graph. Constraints (5.21) and (5.22) are the maximum infeasibility constraints for $\delta_j$ variables, and constraints (5.23) are the binary constraints for the decision variables. Section 5.3.2 shows how to set up the values of $\delta_d, \delta_s, \delta_j$ to identify robust solutions with different characteristics.

Let $U \subset A$ be the set of arcs under implementation uncertainty; let $U'_j$ be the set of arcs under implementation uncertainty whose source is node $j$, and let $U''_j$ be the set of arcs under implementation uncertainty whose destination is node $j$. The equivalent linear formulation to the RSPP as show in Section 2.4 is the following:

$$\min \quad \gamma \tag{5.24}$$

$$\text{s.t.} \sum_{(i,j)\in A\setminus U} c_{ij}x_{ij} + \sum_{(i,j)\in U} c_{ij} \leq \gamma \tag{5.25}$$

$$\sum_{(s,i)\in A\setminus U, i\neq s} x_{si} + |U'_s| - \sum_{(i,s)\in A\setminus U, i\neq s} x_{is} - \delta_s^L \leq 1 \tag{5.26}$$

$$-\sum_{(s,i)\in A\setminus U, i\neq s} x_{si} + \sum_{(i,s)\in A\setminus U, i\neq s} x_{is} + |U''_s| - \delta_s^G \leq -1 \tag{5.27}$$

$$\sum_{(d,i)\in A\setminus U, i\neq d} x_{di} + |U'_d| - \sum_{(i,d)\in A\setminus U, i\neq d} x_{id} - \delta_d^L \leq -1 \tag{5.28}$$

$$-\sum_{(d,i)\in A\setminus U, i\neq d} x_{di} + \sum_{(i,d)\in A\setminus U, i\neq d} x_{id} + |U''_d| - \delta_d^G \leq 1 \tag{5.29}$$

$$\sum_{(j,i)\in A\setminus U, i\neq j} x_{ji} + |U'_j| - \sum_{(i,j)\in A\setminus U, i\neq j} x_{ij} - \delta_j^L \leq 0, \forall j \in N \setminus \{s,d\} \tag{5.30}$$

$$-\sum_{(j,i)\in A\setminus U, i\neq j} x_{ji} + \sum_{(i,j)\in A\setminus U, i\neq j} x_{ij} + |U''_j| - \delta_j^G \leq 0, \forall j \in N \setminus \{s,d\} \tag{5.31}$$

(5.20), (5.21), (5.22), (5.23)

The objective function (5.24) is equivalent to the objective (5.12). Constraints (5.25) are the equivalent linear objective robustness constraint, while constraints (5.26), (5.27), (5.28), (5.29), (5.30) and (5.31) are the equivalent linear feasibility robustness constraints.

### 5.3.2 Properties of the RSPP

The following results present properties of the maximum infeasibility levels, i.e., $\delta_j^{L_{max}}$ and $\delta_j^{G_{max}}$ for the RSPP.

Lemmas 5.3.1 and 5.3.2 show configurations of the RSPP using the control parameters $\delta_s^{L_{max}}$, $\delta_s^{G_{max}}$, $\delta_d^{L_{max}}$, $\delta_d^{G_{max}}$, $\delta_j^{L_{max}}$, $\delta_j^{G_{max}}$ for which robust solutions do not exist due to the structural properties of the RSPP. These results define the lower bounds for the values of the control parameter such that robust solutions may exist.

**Lemma 5.3.1.** *RSPP solutions do not exist if* $\exists j \in N \setminus \{s,d\}$ *such that* $\delta_j^{L_{max}} < |U'_j|$ *and* $\delta_j^{G_{max}} < |U''_j|$.

*Proof.* Assume there exist a solution $x^*$ of the RSPP, and assume $\exists j \in N \setminus \{s,d\}$ such that $\delta_j^{L_{max}} < |U'_j|$ and $\delta_j^{G_{max}} < |U''_j|$. Given that $x^*$ is solution of the RSPP, it satisfies constraints (5.30) and (5.31). Since $\delta_j^{L_{max}} < |U'_j|$, then $\sum_{(j,i)\in A\setminus U, i\neq j} x^*_{ji} - \sum_{(i,j)\in A\setminus U, i\neq j} x^*_{ij} \leq \delta_j^L - |U'_j| < 0$. Similarly, since $\delta_j^{G_{max}} < |U''_j|$, then $-\sum_{(j,i)\in A\setminus U, i\neq j} x^*_{ji} + \sum_{(i,j)\in A\setminus U, i\neq j} x^*_{ij} \leq \delta_j^G - |U''_j| < 0$. However, $\sum_{(j,i)\in A\setminus U, i\neq j} x^*_{ji} - \sum_{(i,j)\in A\setminus U, i\neq j} x^*_{ij} < 0$ and $-\sum_{(j,i)\in A\setminus U, i\neq j} x^*_{ji} + \sum_{(i,j)\in A\setminus U, i\neq j} x^*_{ij} < 0$ is a contradiction. Therefore, $x^*$ cannot be a solution of the RSPP. □

**Lemma 5.3.2.** *RSPP solutions do not exist if* $\delta_s^{L_{max}} < |U'_s|$ *and* $\delta_s^{G_{max}} < |U''_s|$, *or* $\delta_d^{L_{max}} < |U'_d|$ *and* $\delta_d^{G_{max}} < |U''_d|$.

*Proof.* Proof of Lemma 5.3.2 is similar to the proof of Lemma 5.3.1. □

Lemma 5.3.3 presents the result that makes the RSPP equivalent to an SPP over a graph formed only by the arcs associated with certain variables; therefore, the resulting RSPP can be treated as an SPP.

**Lemma 5.3.3.** *A RSPP with* $\delta_j^{L_{max}} = |U'_j|$ *and* $\delta_j^{G_{max}} = |U''_j|, \forall j \in N$ *is equivalent to a SPP over* $G'$, *where* $G' = (N, A \setminus U)$.

*Proof.* Consider the linear reformulation of the RSPP with $\delta_j^{L_{max}} = |U'_j|$ and $\delta_j^{G_{max}} = |U''_j|, \forall j \in N$, the resulting formulation is as follows:

$$\min \quad \gamma \tag{5.32}$$

$$\text{s.t.} \sum_{(i,j)\in A\setminus U} c_{ij}x_{ij} + \sum_{(i,j)\in U} c_{ij} \leq \gamma \tag{5.33}$$

$$\sum_{(s,i)\in A\setminus U, i\neq s} x_{si} - \sum_{(i,s)\in A\setminus U, i\neq s} x_{is} \leq 1 \tag{5.34}$$

$$-\sum_{(s,i)\in A\setminus U, i\neq s} x_{si} + \sum_{(i,s)\in A\setminus U, i\neq s} x_{is} \leq -1 \tag{5.35}$$

$$\sum_{(d,i)\in A\setminus U, i\neq d} x_{di} - \sum_{(i,d)\in A\setminus U, i\neq d} x_{id} \leq -1 \tag{5.36}$$

$$-\sum_{(d,i)\in A\setminus U, i\neq d} x_{di} + \sum_{(i,d)\in A\setminus U, i\neq d} x_{id} \leq 1 \tag{5.37}$$

$$\sum_{(j,i)\in A\setminus U, i\neq j} x_{ji} - \sum_{(i,j)\in A\setminus U, i\neq j} x_{ij} \leq 0, \forall j \in N \setminus \{s,d\} \tag{5.38}$$

$$-\sum_{(j,i)\in A\setminus U, i\neq j} x_{ji} + \sum_{(i,j)\in A\setminus U, i\neq j} x_{ij} \leq 0, \forall j \in N \setminus \{s,d\} \tag{5.39}$$

(5.20), (5.21), (5.22), (5.23)

The objective function and constraint (5.33) are equivalent to an objective function $\min \sum_{(i,j)\in A\setminus U} c_{ij}x_{ij} + \sum_{(i,j)\in U} c_{ij}$. Since $\sum_{(i,j)\in U} c_{ij}$ is a constant value, the objective function can be simplified to $\min \sum_{(i,j)\in A\setminus U} c_{ij}x_{ij}$ and add the given value of $\sum_{(i,j)\in U} c_{ij}$ after solving the problem. Constraints (5.34) and (5.35), (5.36) and (5.37), and (5.38) and (5.39) are equivalent to $\sum_{(s,i)\in A\setminus U, i\neq s} x_{si} - \sum_{(i,s)\in A\setminus U, i\neq s} x_{is} = 1$, $\sum_{(d,i)\in A\setminus U, i\neq s} x_{di} - \sum_{(i,d)\in A\setminus U, i\neq s} x_{id} = -1$ and $\sum_{(j,i)\in A\setminus U, i\neq s} x_{ji} - \sum_{(i,j)\in A\setminus U, i\neq s} x_{ij} = 0$, respectively. Given that $\delta_j^{L_{max}} = |U'_j|$ and $\delta_j^{G_{max}} = |U''_j|, \forall j \in N$, constraints (5.21) and (5.22) become redundant. Therefore, the RSPP is equivalent to a SPP over $G'$. □

Lemma 5.3.4 states the configuration of the RSPP such that if solutions exist all the certain variables are equal to zero; i.e., robust solutions do not define a path from $s$ to $d$.

**Lemma 5.3.4.** *Solutions of a RSPP with* $\delta_j^{L_{max}} \geq |U'_j| + 1$ *and* $\delta_j^{G_{max}} \geq |U''_j| + 1, \forall j \in N$ *consist of all certain variables equal to zero; i.e.,* $x_{ij} = 0, \forall (i,j) \in A \setminus U$.

*Proof.* Assume the solution of the RSPP, $x^*$, $\gamma^*$ and $\delta_j^L*, \delta_j^G*$ with $x_{ij}^* = 0 \forall (i,j) \in A \setminus U$ and $\delta_j^L* \leq |U_j'| + 1$ and $\delta_j^G* \leq |U_j''| + 1, \forall j$ is not optimal. Therefore, there may exist an optimal solution of the RSPP $x'$, $\gamma'$ and $\delta_j^{L'}, \delta_j^{G'}$ such that $\exists x_{ij}' = 1$ for $(i,j) \in A \setminus U$. Since $\gamma'$ is optimal, then $\gamma' \leq \gamma^*$. However, $\sum_{(i,j)\in A\setminus U} c_{ij} x_{ij}^* + \sum_{(i,j)\in U} c_{ij} = 0 \leq \gamma^*$, and $1 \leq \sum_{(i,j)\in A\setminus U} c_{ij} x_{ij}' + \sum_{(i,j)\in U} c_{ij} \leq \gamma'$; these results leads to $\gamma^* < \gamma'$, which is a contradiction. Therefore, the solution of the RSPP, $x^*$, $\gamma^*$ and $\delta_j^L*, \delta_j^G*$ with $x_{ij}^* = 0 \forall (i,j) \in A \setminus U$ and $\delta_j^L* \leq |U_j'| + 1$ and $\delta_j^G* \leq |U_j''| + 1, \forall j$ is optimal. $\square$

Robust solutions described in Lemma 5.3.4 do not define any path whatsoever since all the certain variables are equal to zero. On the other hand, robust solutions described in Lemma 5.3.3 produce a single path from $s$ to $d$ formed only by arcs associated with certain variables. Although the solutions described in Lemma 5.3.3 guarantee a path from $s$ to $d$, there may exist a different path formed with arcs under implementation uncertainty available to travel through them such that the cost of such path is smaller than the cost of the path formed by arcs not under implementation uncertainty. Therefore, it is also of interest to find the shortest paths from nodes reachable with an arc under implementation uncertainty to the destination node. Lemma 5.3.5 describes robust solutions that permit to identify such shortest paths.

**Lemma 5.3.5.** *Let $M$ be a large number. RSPP solutions with $\delta_s^{L_{max}} = M$, $\delta_s^{G_{max}} \leq |U_s''|$, $\delta_d^{L_{max}} \leq |U_d'|$, $\delta_d^{G_{max}} = M$, $\delta_j^{L_{max}} \leq |U_j'|$ and $\delta_j^{G_{max}} < |U_j''|$ define the shortest path between $s$ and $d$, and the shortest paths between every node $i$ such that $\exists (j,i)$ with $(j,i) \in U$ and $d$.*

*Proof.* $\delta_s^{G_{max}} \leq |U_s''|$ leads to $|U_s''| - \delta_s^G \geq 0$, then $-\sum_{(s,i)\in A\setminus U, i\neq s} x_{si} + \sum_{(i,s)\in A\setminus U, i\neq s} x_{is} \leq -\sum_{(s,i)\in A\setminus U, i\neq s} x_{si} + \sum_{(i,s)\in A\setminus U, i\neq s} x_{is} + |U_s''| - \delta_s^G$ and by constraint (5.27) we have that $-\sum_{(s,i)\in A\setminus U, i\neq s} x_{si} + \sum_{(i,s)\in A\setminus U, i\neq s} x_{is} \leq -1$. Constraint (5.27) is feasible with $\sum_{(s,i)\in A\setminus U, i\neq s} x_{si} \geq 1$, and $\delta_s^{L_{max}}$ equals to a large number makes constraint (5.26) feasible for any value of $\sum_{(s,i)\in A\setminus U, i\neq s} x_{si}$; therefore, there exists at least one arc not under implementation uncertainty whose source is $s$.

Similarly, $\delta_d^{L_{max}} \leq |U'_d|$ leads to $\sum_{(i,d)\in A\setminus U, i\neq s} x_{id} \geq 1$, which guarantee that there exists at least one arc not under implementation uncertainty whose destination is $d$. $\delta_j^{G_{max}} < |U''_j|$ with $|U''_j| = 0$ leads to $\delta_j^G = 0$, and with $\delta_j^{L_{max}} \leq |U'_j|$ make the constraints (5.30) and (5.31) equivalent to flow conservation constraints with arcs not under implementation uncertainty only. On the other hand, $\delta_j^{G_{max}} < |U''_j|$ with $|U''_j| > 0$ leads to $-\sum_{(j,i)\in A\setminus U, i\neq j} x_{ji} + \sum_{(i,j)\in A\setminus U, i\neq j} x_{ij} < -\sum_{(j,i)\in A\setminus U, i\neq j} x_{ji} + \sum_{(i,j)\in A\setminus U, i\neq j} x_{ij} + |U''_j| - \delta_j^G$; therefore $-\sum_{(j,i)\in A\setminus U, i\neq j} x_{ji} + \sum_{(i,j)\in A\setminus U, i\neq j} x_{ij} < 0$ and there exists at least one arc not under implementation uncertainty with node $j$ as its source. These constraints together guarantee the existence of the shortest paths. □

Figure 5.1 show examples of the robust solutions described in Lemma 5.3.5. Graphs in Figure 5.1 possess different densities based on the percentage of all the possible arcs that may be included; a graph is consider sparse if it contains 10% of all the possible arcs, non-dense-non-sparse if contains 50% of all the possible arcs, and dense if it contains 90% of all the arcs.

### 5.4 Robust Dynamic Shortest Path Algorithm

This section proposes an algorithm to find the robust solutions of the RSPP based on Dijkstra's shortest path algorithm. This algorithm is named *robust dynamic shortest path algorithm* (RDA). The purpose of the RDA is to identify robust solutions described in Lemmas 5.3.3 and 5.3.5 without solving the nonlinear or linear formulations.

The proposed algorithm assumes that the values of the control parameters $\delta_s^{L_{max}}$, $\delta_s^{G_{max}}$, $\delta_d^{L_{max}}$, $\delta_d^{G_{max}}$, $\delta_j^{L_{max}}$, $\delta_j^{G_{max}}$ are set up as described in Lemmas 5.3.3 or 5.3.5. The RDA does not consider the arcs under implementation uncertainty; RDA only identifies the values of the uncertain variables and assumes a value of zero for all the uncertain variables.

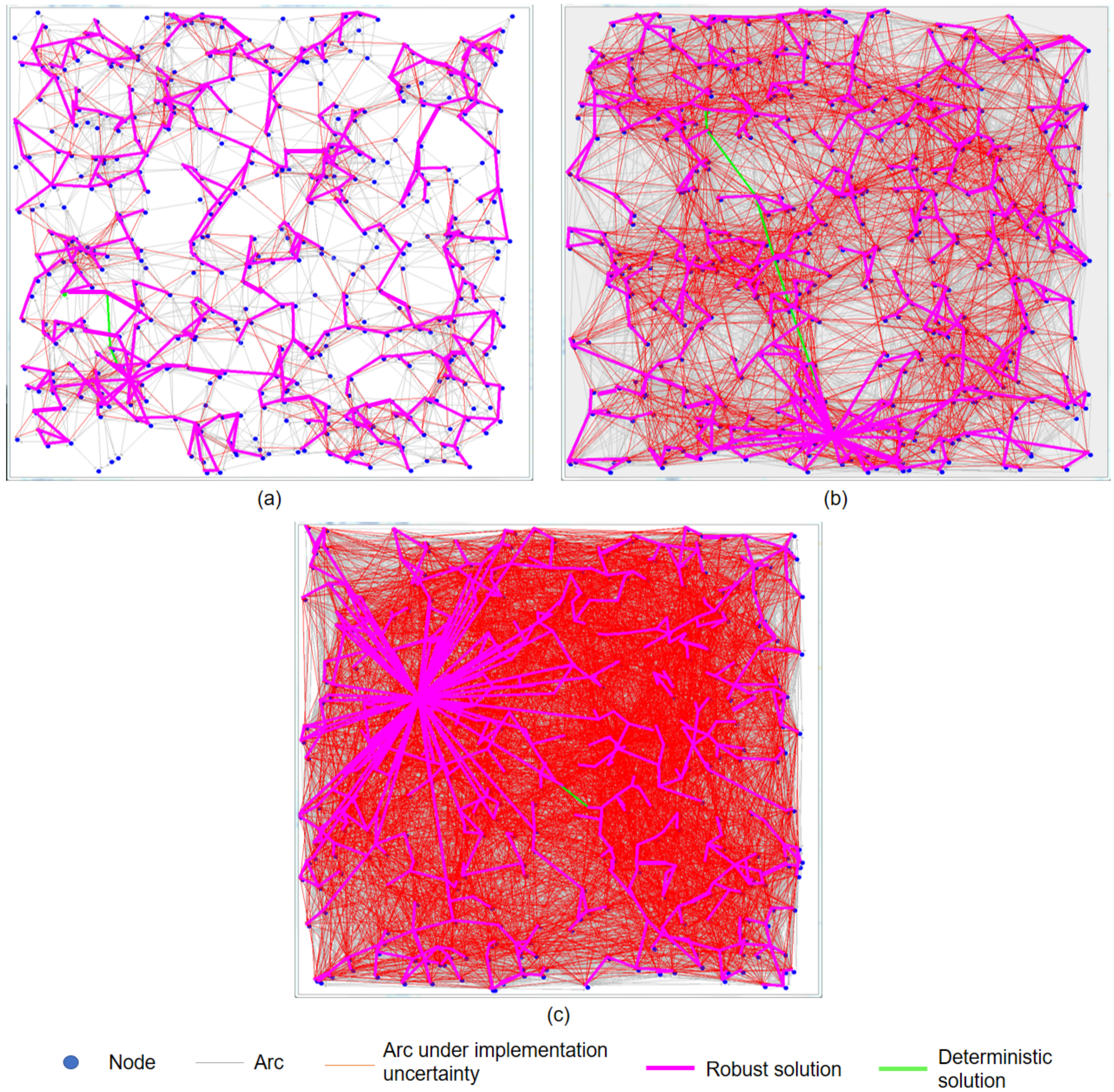



Figure 5.1: Robust solutions as described in Lemma 5.3.5. (a) Sparse graph. (b) Non-dense-non-sparse graph. (c) Dense graph.

Therefore, the RDA identifies the solution providing the lower bound of the objective function described in Proposition 2.5.2, i.e., the optimistic solution.

Given a digraph with arcs possessing nonnegative weights, Dijkstra (1959) presents an algorithm to find the shortest path between a source node and every other node. To form the shortest paths, Dijkstra's algorithm divides nodes into two sets containing the visited and non-visited nodes; initially, the set of visited nodes contains the source node only. In every iteration, the algorithm searches for a pair of nodes in different sets with the minimum distance between them, it moves the node in the non-visited set to another set, updates the distances, and repeat the process until the non-visited set is empty. Dijkstra's algorithm is able to identify the shortest path on a digraph with cycles containing nonnegative weights.

The RSPP is a modified SPP such that the path connecting the source and destination nodes is formed only by arcs not under implementation uncertainty. Additionally, the robust solutions connect every node destination of an arc under implementation uncertainty with the destination node of the problem. It is possible to develop an algorithm to identify the robust solutions provided by the formulation of the RSPP. Given a digraph $G = (N, A)$, a source node $s$, a destination node $d$, a nonnegative weight $c_{ij}$ for every arc $(i, j) \in A$, and the set of arcs under implementation uncertainty $U$, the RDA identifies the shortest paths formed only with arcs not under implementation uncertainty between the destination node $d$, the source node $s$ and every node $i \in V$ such that $\exists (j, i) \in U$.

The RDA follows Dijkstra's concept of dividing nodes into sets containing the visited and non-visited nodes; the RDA requires to know the set of arcs under implementation uncertainty and that Assumptions 5.3.1 and 5.3.2 are satisfied. Initially, the RDA includes the destination node in the set of visited nodes. In every iteration, the RDA searches for a pair of nodes in different sets connected through arcs not under implementation uncertainty and minimum weight; then it moves the node in the non-visited set to the visited set, updates the distances, and repeat the process until the set of non-visited nodes is empty. The RDA is described in Algorithm 1.

Lines 2 to 8 in Algorithm 1 initialize the process; lines 2 and 6 create a set of vertices $Q$, which are considered the non-visited nodes; line 4 makes the cost for every node a large number, while line 5 sets the next node in the optimal path for every node undefined; finally, line 11 make the cost of $d$ equals zero. Lines 9 to 19 implement the searching for

**Algorithm 1** RDA.

**input:** a digraph $G = (N, A)$, a set of arcs under implementation uncertainty $U$, a source node $s$ and a destination node $d$

**output:** a precedence list, next, describing the shortest paths

```
1:  function SolveRSPP(G = (N, A), U, s, d)
2:      create node set Q                          ▷ Set of all the nodes that must be connected to d
3:      for all nodes v in G do
4:          dist[v] ← INFINITY                     ▷ Unknown distance from d to v
5:          next[v] ← UNDEFINED                    ▷ Set the next node in the path from v to d
6:          add v to Q                             ▷ All nodes initially in Q
7:      end for
8:      dist[d] ← 0                                ▷ Cost from the destination node to the destination node
9:      while Q is not empty do
10:         u ← node in Q with minimum dist[u]     ▷ Node with the smallest cost
11:         Remove u from Q
12:         for all v in Q such that the arc (v, u) is in A \ U do ▷ Adjacent nodes connect through
        arcs not under implementation uncertainty
13:             aux ← dist[u]+ Cost(u, v)
14:             if aux < dist[v] then              ▷ If a shortest path is found
15:                 dist[v] ← aux                  ▷ Update the cost to v
16:                 next[v] ← u                    ▷ Make u the next node in the optimal path from v to d
17:             end if
18:         end for
19:     end while
20:     return next
21: end function
```

the shortest paths formed by arcs not under implementation uncertainty; this process is repeated until every node in $Q$ is connected to $d$. Line 10 seeks for the node $u$ in $Q$ with the minimum cost; during the first time the node $u$ is always the node $d$ since its cost was previously set to zero. Line 11 removes $u$ from set $Q$, which is equivalent to make the node $u$ a visited node. Lines 12 to 18 search for every node $v \in Q$ adjacent to $u$ that is connected by an arc not under implementation uncertainty whose destination is $u$. Line 13 computes the total cost from the destination node $d$ to $v$; if the updated cost is less than the actual cost (line 14), then line 15 updates the cost of $v$ with the minimum cost. Line 16 makes $u$ the next node in the shortest path from $v$ to $d$.

The shortest paths defined by robust solutions in Lemma 5.3.5 can be read by following the sequences of nodes described in the data structure next[]; Algorithm 1 can be terminated after line 11 when $Q$ does not contain any node destination of an arc not under implementation uncertainty neither contains the source node $s$. Robust solutions in Lemma 5.3.3 can be read similarly; Algorithm 1 can be terminated after line 11 if $u = s$.

The proof of correctness of the RDA can be constructed by induction; the objective is to show that dist$[u]$ provides the shortest path from $u$ to $d$, with $u$ being any node in $G$. The proof is as follows:

*Proof.* Let $\ell(u)$ be the shortest path from $u$ to $d$ and labeled nodes removing from $Q$ as visited nodes.

Basis: Let $d$ be a visited node, then dist$[d] = \ell(d) = 0$. This is true because of line 8 in Algorithm 1.

Inductive step: For each visited node $v$, the shortest path from $v$ to $d$ is dist$[v] = \ell(v)$. If the non-visited node $u$ is visited, then dist$[u] = \ell(u)$.

Let the arc $(u, v)$ with $v$ visited node be selected to form the path from $u$ to $d$ and make $u$ a visited node, then dist$[u]$ $=$ dist$[v]$ $+$ Cost$(u, v)$ $= \ell(v) +$ Cost$(u, v)$ is the shortest path from $u$ to $d$. Assume there exists a shorter path that considers the arc $(u, y)$ with node $y \in Q$ to form the path from $u$ to $d$; also, assume the arc $(y, x)$ with $x$ visited node to form the path from $y$ to $d$. Since $u$ is selected from $Q$ in line 10, it means that the dist$[u]$ $\leq$ dist$[y]$; otherwise, $y$ would be selected. Given that Cost$(u, y)$ is nonnegative, then dist$[u] = \ell(v) +$ Cost$(u, v) \leq$ dist$[y]$ $+$ Cost$(u, y) = \ell(x) +$ Cost$(y, x)$ $+$ Cost$(u, y)$. This result is a contradiction and there can be no such node $y$ that leads to a shortest path from $u$ to $d$. Therefore, our assumption is false and dist$[u] = \ell(u)$.

This result completes the proof of the correctness of the RDA; therefore, dist$[v]$ provides the shortest path from $v$ to $d$. □

**Algorithm 2** Modified RDA.

**input:** a digraph $G = (N, A)$, a set of arcs under implementation uncertainty $U$, a source node $s$ and a destination node $d$

**output:** a precedence list, next, describing the shortest paths

1: **function** SOLVERSPP($G = (N, A), U, s, d$)<br>
2:     create node set $Q$ ▷ Set of all the nodes that must be connected to $d$<br>
3:     **for all** nodes $v$ in $G$ **do**<br>
4:         dist$[v] \leftarrow$ INFINITY ▷ Unknown distance from $d$ to $v$<br>
5:         next$[v] \leftarrow$ UNDEFINED ▷ Set the next node in the path from $v$ to $d$<br>
6:         $Q$.add_priority$(v, dist[v])$ ▷ Add nodes to $Q$ with a priority based on their dist$[v]$ value<br>
7:     **end for**<br>
8:     dist$[d] \leftarrow 0$ ▷ Cost from $d$ to $d$<br>
9:     **while** $Q$ is not empty **do**<br>
10:         $u \leftarrow Q$.extract_min() ▷ Remove and return the node with smallest priority<br>
11:         Remove $u$ from $Q$<br>
12:         **for all** $v$ in $Q$ such that the arc $(v, u)$ is in $A \setminus U$ **do**<br>
13:             $aux \leftarrow$ dist$[u]$+ Cost$(u, v)$<br>
14:             **if** $aux <$ dist$[v]$ **then** ▷ If a shortest path is found<br>
15:                 dist$[v] \leftarrow aux$ ▷ Update the cost to $v$<br>
16:                 next$[v] \leftarrow u$ ▷ Make $u$ the next node in the optimal path from $v$ to $d$<br>
17:                 $Q$.decrease_priority$(v, aux)$ ▷ Update priority of node $v$<br>
18:             **end if**<br>
19:         **end for**<br>
20:     **end while**<br>
21:     **return** next<br>
22: **end function**

Algorithm 2 presents a modification of the RDA using a basic queue to reduce the computational time of the operations add_priority, extract_priority and decrease_priority; a Fibonacci heap structure, presented in Fredman and Tarjan (1987), offers optimal implementation for those three operations leading to a running time of $\mathcal{O}(|N| \log |N| + |A|)$. The reason is that the inner loop in lines 9 to 20 is executed $\mathcal{O}(|N| + |A|)$ time and the decrease_priority operation using a Fibonacci heap structure runs in $\mathcal{O}(1)$, which leads to the $\mathcal{O}(|N| \log |N| + |A|)$ time.

## 5.5 Cardinality-Constrained Robust Formulation of the SPP

This section shows the cardinality-constrained robust SPP under implementation uncertainty (CRSPP) based on the concepts presented in Chapter 2.

The CRSPP can be formulated as follows:

$$\min \quad \gamma \tag{5.40}$$

$$\text{s.t.} \sum_{(i,j)\in A\setminus U} c_{ij}x_{ij} + \max_{\{S_0: S_0\subseteq U, |S_0|\leq\Gamma\}} \left\{ \sum_{(i,j)\in S_0} \left(\frac{c_{ij}+|c_{ij}|}{2}\right) + \sum_{(i,j)\in U\setminus S_0} c_{ij}x_{ij} \right\} \leq \gamma \tag{5.41}$$

$$\sum_{(s,i)\in A\setminus U, i\neq s} x_{si} - \sum_{(i,s)\in A\setminus U, i\neq s} x_{is} + \max_{\{S'_s: S'_s\subseteq U'_s, |S'_s|\leq\Gamma\}} \left\{ |S'_s| + \sum_{(s,i)\in U'_s\setminus S'_s, i\neq s} x_{si} - \sum_{(i,s)\in U'_s\setminus S'_s, i\neq s} x_{is} \right\} - \delta_s^L \leq 1 \tag{5.42}$$

$$- \sum_{(s,i)\in A\setminus U, i\neq s} x_{si} + \sum_{(i,s)\in A\setminus U, i\neq s} x_{is} + \max_{\{S''_s: S''_s\subseteq U''_s, |S''_s|\leq\Gamma\}} \left\{ |S''_s| - \sum_{(s,i)\in U''_s\setminus S''_s, i\neq s} x_{si} + \sum_{(i,s)\in U''_s\setminus S''_s, i\neq s} x_{is} \right\} - \delta_s^G \leq -1 \tag{5.43}$$

$$\sum_{(d,i)\in A\setminus U, i\neq d} x_{di} - \sum_{(i,d)\in A\setminus U, i\neq d} x_{id} + \max_{\{S'_d: S'_d\subseteq U'_d, |S'_d|\leq\Gamma\}} \left\{ |S'_d| + \sum_{(d,i)\in U'_d\setminus S'_d, i\neq d} x_{di} - \sum_{(i,d)\in U'_d\setminus S'_d, i\neq d} x_{id} \right\} - \delta_d^L \leq -1 \tag{5.44}$$

$$- \sum_{(d,i)\in A\setminus U, i\neq d} x_{di} + \sum_{(i,d)\in A\setminus U, i\neq d} x_{id} +$$

$$\max_{\left\{S''_d : S''_d \subseteq U''_d, |S''_d| \leq \Gamma\right\}} \left\{ |S''_d| - \sum_{(d,i) \in U''_d \setminus S''_d, i \neq d} x_{di} + \sum_{(i,d) \in U''_d \setminus S''_d, i \neq d} x_{id} \right\} - \delta^G_d \leq 1 \tag{5.45}$$

$$\sum_{(j,i) \in A \setminus U, i \neq j} x_{ji} - \sum_{(i,j) \in A \setminus U, i \neq j} x_{ij} + \max_{\left\{S'_j : S'_j \subseteq U'_j, |S'_j| \leq \Gamma\right\}} \left\{ |S'_j| + \sum_{(j,i) \in U'_j \setminus S'_j, i \neq j} x_{ji} - \sum_{(i,j) \in U'_j \setminus S'_j, i \neq j} x_{ij} \right\} - \delta^L_j \leq 0, \quad \forall j \in N \setminus \{s, d\} \tag{5.46}$$

$$- \sum_{(j,i) \in A \setminus U, i \neq j} x_{ji} + \sum_{(i,j) \in A \setminus U, i \neq j} x_{ij} + \max_{\left\{S''_j : S''_j \subseteq U''_j, |S''_j| \leq \Gamma\right\}} \left\{ |S''_j| - \sum_{(j,i) \in U''_j \setminus S''_j, i \neq j} x_{ji} + \sum_{(i,j) \in U''_j \setminus S''_j, i \neq j} x_{ij} \right\} - \delta^G_j \leq 0, \quad \forall j \in N \setminus \{s, d\} \tag{5.47}$$

(5.20), (5.21), (5.22), (5.23)

The CRSPP attempts to reduce conservatism by bounding the maximum number of uncertain variables with different prescribed and implemented values. However, CRSPP can be proved to be equivalent to the RSPP due to the structural properties of the SPP. Corollary 5.5.1 shows this result.

**Corollary 5.5.1.** *The CRSPP is equivalent to the RSPP.*

*Proof.* Given $c_{ij} > 0, \forall (i,j) \in A$, Corollary 5.5.1 follows from results in Corollary 2.5.5 applied to SPP. □

From Corollary 5.5.1, the CRSPP can be treated as RSPP, including the use of algorithm developed in Section 5.4 to identify its robust solutions.

## 5.6 Experimental Study

This experimental study evaluates the performance of the deterministic and robust solutions in terms of their practical feasibility and objective value. This study aims to illustrate the situations where the use of the robust formulation is more appropriate than the deterministic one.

### 5.6.1 Performance Measures

Deterministic and robust solutions define at least one path; moreover, implementation uncertainty may define more paths or may disrupt the existing ones. From a practical perspective, it is expected that solutions define at least one path connecting the source and destination nodes; this concept is formalized next.

Let $\mathcal{C}$ be the set of all the paths in $G$ connecting $s$ and $d$. Let $\mathcal{P}(x)$ be the set of paths defined by the solution $x$. The practical feasibility of $x$ can be defined as follows:

**Definition 5.6.1.** *A solution $x$ is practical feasible if $\mathcal{P}(x) \cap \mathcal{C} \neq \emptyset$.*

Definition 5.6.1 states that a solution is practical feasible if it defines at least one path connecting the source and destination nodes.

Implementation uncertainty may impact the practical feasibility of the solution. Let $\mathcal{F}'(x)$ be the set of the practical feasible solutions in $\mathcal{U}(x_C)$; $\mathcal{F}'(x)$ is defined as follows:

**Definition 5.6.2.** *The set of the practical feasible solutions in $\mathcal{U}(x_C)$ is defined as $\mathcal{F}'(x) = \{x \in \mathcal{U}(x_C) : \mathcal{P}(x) \cap X \neq \emptyset\}$.*

Given a solution $x$, its feasibility level $h'(x)$ is defined as the proportion of the total number of practical feasible solutions among all possible implementation vectors result of implementation uncertainty.

**Definition 5.6.3.** *The feasibility ratio $h'(x)$ can be defined as follows:*

$$h'(x) = \frac{|\mathcal{F}'(x)|}{|\mathcal{U}(x_C)|} \tag{5.48}$$

The greater the feasibility level, better the protection of the solution against implementation uncertainty.

Given a practical feasible solution $x$, the cost $f^-(x)$ of the shortest path in $\mathcal{P}(x) \cap X$ connecting $s$ and $d$ is defined as follows:

$$f^-(x) = \min_{y \in \mathcal{P}(x) \cap X} \{f(y)\} \tag{5.49}$$

The decision of computing the cost of the shortest path in $\mathcal{P}(x) \cap X$ to evaluate the performance of the solutions is driven by the objective of determining how good the solution can be. For instance, consider that the deterministic solution produces the shortest path among $X$, say $p_1$; assume that due to implementation uncertainty $x$ also define a second path $p_2$ connecting $s$ and $d$. Assume a different feasible solution $y$, which produces a feasible path, say $p_3$, with greater cost than $p_1$; assume also that due to implementation uncertainty $p_2$ is also defined by $y$ and the cost of $p_2$ is greater than the cost of $p_3$. Note that the worst case of the two solutions is the same; therefore, under the maximum cost one can conclude that their performance is the same. However, by evaluating their performance according to the minimum value, clearly, the deterministic solution performs better; therefore, this measure of performance is fair with the solutions that provide a shorter path.

Given a solution $x$, its average objective value $\bar{f}'(x)$ is defined as the average minimum cost of the shortest paths in $\mathcal{P}(x) \cap X$.

**Definition 5.6.4.** *The average objective value $\bar{f}'(x)$ is defined as follows:*

$$\bar{f}'(x) = \frac{\sum_{y \in \mathcal{F}'(x)} f^-(y)}{|\mathcal{F}'(x)|} \tag{5.50}$$

The lower the value of $\bar{f}'(x)$, better the solution performs in terms of the objective value.

Similarly to the loss of the objective performance ratio in Definition 3.4.3, we define $l'(x, y)$ for the SPP as follows:

**Definition 5.6.5.** *Given two solutions $x$ and $y$, the loss of the objective performance ratio, $l'(x, y)$, of $y$ with respect to $x$ is given as follows, assuming a minimization objective function:*

$$l'(x, y) = \frac{\bar{f}'(y) - \bar{f}'(x)}{\bar{f}'(x)} \tag{5.51}$$

The larger the value of $l'(x, y)$, the worse is the objective performance of $y$ compared to $x$.

### 5.6.2 Test Problem Generation

The test problems considered in this study are generated as follows:

1. The generation of a directed graph $G = (N, A)$ using Algorithm 3; the parameter $n$ controls the number of nodes, and parameter $b \in (0, 1)$ controls the density of the arcs, the greater the value of $b$, more dense the graph is; the parameter $d$ controls the distance between the source and destination nodes, when $d = 1$ they are near to each other, when $d = 2$ they are located in a middle distance, and when $d = 3$ locates them far from each other. The subroutine compute_maximum_distance computes the maximum distance between any pair of nodes in $N$. Source and destination nodes are considered near to each other when the distance between them is less than 25% of the maximum distance between any pair of nodes; they are considered far from each other if the distance between them is greater than 75% of the maximum distance, and they are considered in a medium distance otherwise. The subroutine

locate_source_destination locates the source and destination nodes randomly such that their distance satisfies the configuration given by the parameter $d$. The density of the graph is defined as the percentage of all the possible arcs that may be included in $G$; the parameter $b$ controls such percentage. A graph is consider sparse if $b = 0.1$, non-sparse-non-dense if $b = 0.5$ and dense if $b = 0.9$.

**Algorithm 3** Generation of a graph $G = (N, A)$

**input:** number of nodes $n$, density $b$, distance between source and destination nodes $d$
**output:** directed graph $G = (N, A)$

```
function GRAPHGENERATION(n,b)
    for 2 to n − 1 do
        Coordinate X ← Random number between 1 and 1000        ▷ Generate the coordinate X
        Coordinate Y ← Random number between 1 and 1000        ▷ Generate the coordinate Y
        if Coordinates X and Y do not exist already then       ▷ Verify that coordinates are unique
            Add coordinates X and Y to set of nodes N          ▷ Add the new node and its coordinates
        end if
    end for
    maximum distance ← compute_maximum_distance(N)             ▷ Compute the maximum distance
    N ← locate_source_destination(d, maximum distance)         ▷ Locate the source and destination nodes
    for all nodes i in N do
        for all nodes j in N do
            if i ≠ j and arc (j, i) not in A then
                if Random[0, 1) > (1 − b) then
                    Add arc (i, j) to A                        ▷ Add the selected arc to the set of arcs
                end if
            end if
        end for
    end for
    Return G = (N, A)
end function
```

2. The determination of the set of arcs under implementation uncertainty $U$ considering that the arcs under uncertainty are uniformly randomly selected, or they are clustered near to the source node, destination node or in the middle of these two nodes. Algorithm 4 shows how to compute $U$ when the arcs under implementation uncertainty are randomly uniformly select among all elements in $A$; the parameter

$p$ controls the percentage of arcs under implementation uncertainty to be selected. The subroutine verify_uncertain verifies that Assumption 5.3.1 and 5.3.2 are satisfied when the current arc $u$ is considered into the set $U$.

---

**Algorithm 4** Selection of the arcs under implementation uncertainty $U$

---

**input:** acyclic graph $G = (N, A)$ and percentage of arcs under implementation uncertainty $p$
**output:** a set of arcs under implementation uncertainty $U$

```
function GenerateUncertainSet(G = (N, A), p)
    while |U| < p × |A| do          ▷ Repeat until the maximum number of arcs under implementation uncertainty is achieved
        u ← Random number between 1 and |A|          ▷ Select one element of A
        if verify_uncertain(G = (N, A), U, u) is False then
            Add u to set U          ▷ Add arcs u to the set U
        end if
    end while
    Return U
end function
```

---

Algorithm 5 shows how to compute $U$ when most of the arcs in this set are clustered; the parameter $c$ determines whether the arcs under uncertainty are clustered near the source node ($c = 1$), near the destination node ($c = 3$), or in the middle of the source and destination nodes ($c = 2$). The subroutine sort_nodes_near_to sorts ascending the nodes based on their cost to the objective node; if $c = 1$ the objective node is the source node, if $c = 3$ the objective node is the destination node, and if $c = 2$ the objective node is the midpoint between the source and destination nodes.

Given a graph $G$ generated as described in the step 1, different scenarios are defined depending of the distribution of the arcs under implementation uncertainty among the graph as follows:

- **Scenario 1:** The arcs under implementation uncertainty uniformly randomly distributed among the graph.

**Algorithm 5** Selection of the arcs under implementation uncertainty $U$ in clusters

**input:** acyclic graph $G = (N, A)$, a percentage of arcs under implementation uncertainty $p$, and cluster $c = 1, 2, 3$
**output:** a set of arcs under implementation uncertainty $U$

```
function GENERATEUNCERTAINSET(G = (N, A), p, c)
    sorted_nodes = sort_nodes_near_to(N, c)  ▷ Sort ascending nodes based on cost to the objective node
    i = 0
    while i < |N| and |U| < p × |A| do  ▷ Repat until the maximum number of arcs under uncertainty
        j = 0
        while j < |N| and |U| < p × |A| do  ▷ Repat until the maximum number of arcs under uncertainty
            if node i is adjacent to sorted_nodes[j] then  ▷ Search adjacent nodes to the closest node
                select an arc u = (i, sorted_nodes[j])  ▷ Select an arc whose destination is the closest node
                if Random number between [0,1) > 0.5 and verify_uncertain(G = (N, A), U, u) is False then
                    Add u to set U  ▷ Add arcs u to the set U
                end if
            end if
            increment j  ▷ Select next node in sorted_nodes[j]
        end while
        increment i  ▷ Select next node
    end while
    Return U
end function
```

- **Scenario 2:** Most of the arcs under implementation uncertainty are clustered near to the source node.
- **Scenario 3:** Most of the arcs under implementation uncertainty are clustered near to the destination node.
- **Scenario 4:** Most of the arcs under implementation uncertainty are clustered in the middle of the source and destination nodes.

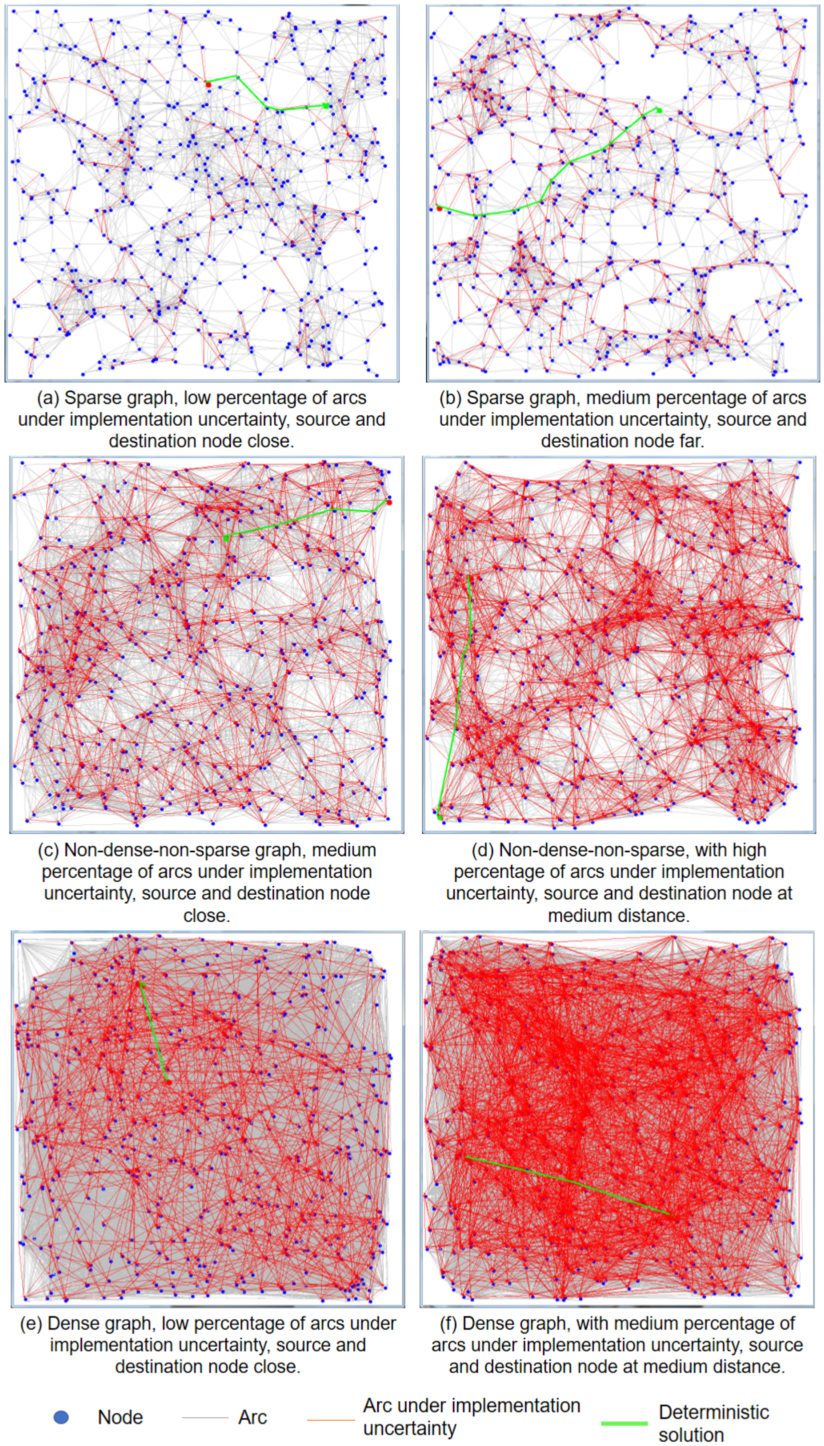
Each scenario is tested with the source and destination located near, far and at a medium distance from each other, and when the graph is sparse, dense or non-sparse-non-dense. Figures 5.2 shows different examples of graphs with arcs under implementation uncertainty uniformly randomly distributed, different percentages of arcs under uncertainty, and different distances between the source and destination nodes. The deterministic shortest path is shown but the robust solutions are omitted for clarity of the graphs.

Figure 5.3 show examples of different graphs with the arcs under implementation uncertainty clustered in different areas. These graphs represent examples of scenarios 2, 3, and 4, respectively.

The scenarios described above may represent real situations. For instance, in a distribution network with the source or destination points located near to the coast, the roads connecting them with the rest of the network may be highly affected when a hurricane occurs; this situation may be represented by a scenario with the arcs under implementation uncertainty clustered near to the source or destination nodes. Similarly, graphs with a low percentage of arcs under uncertainty may represent transportation networks with a small number of bridges, few areas prone to flooding, and so on.

On the other hand, graphs with a high number of arcs under uncertainty may represent areas where disasters or traffic accidents may block many roads with high probability. For example, traveling through a big metropolis, such as Houston, during peak hours, or when the city is affected by a disaster such as the hurricane Harvey. In the same way, the density of the graph may represent the connectivity of the different regions or the type of roads considered; for example, if only highways are considered, the density may be low given that only a few roads are considered. In contrast, if any road may be included, then the road network may be represented for a more dense graph which considers urban roads, detours and so on. Scenarios with source and destination nodes near to each other may represent the transportation of loads locally, for example, between south and north Houston areas; while these nodes located far from each other may represent the travel between two cities, for example, traveling from San Antonio to Dallas, Texas.

For each scenario and different configuration, there are generated 30 problems with different graphs $G$ consisting of 500 nodes, and different set of arcs under implementation




(a) Sparse graph, low percentage of arcs under implementation uncertainty, source and destination node close.

(b) Sparse graph, medium percentage of arcs under implementation uncertainty, source and destination node far.

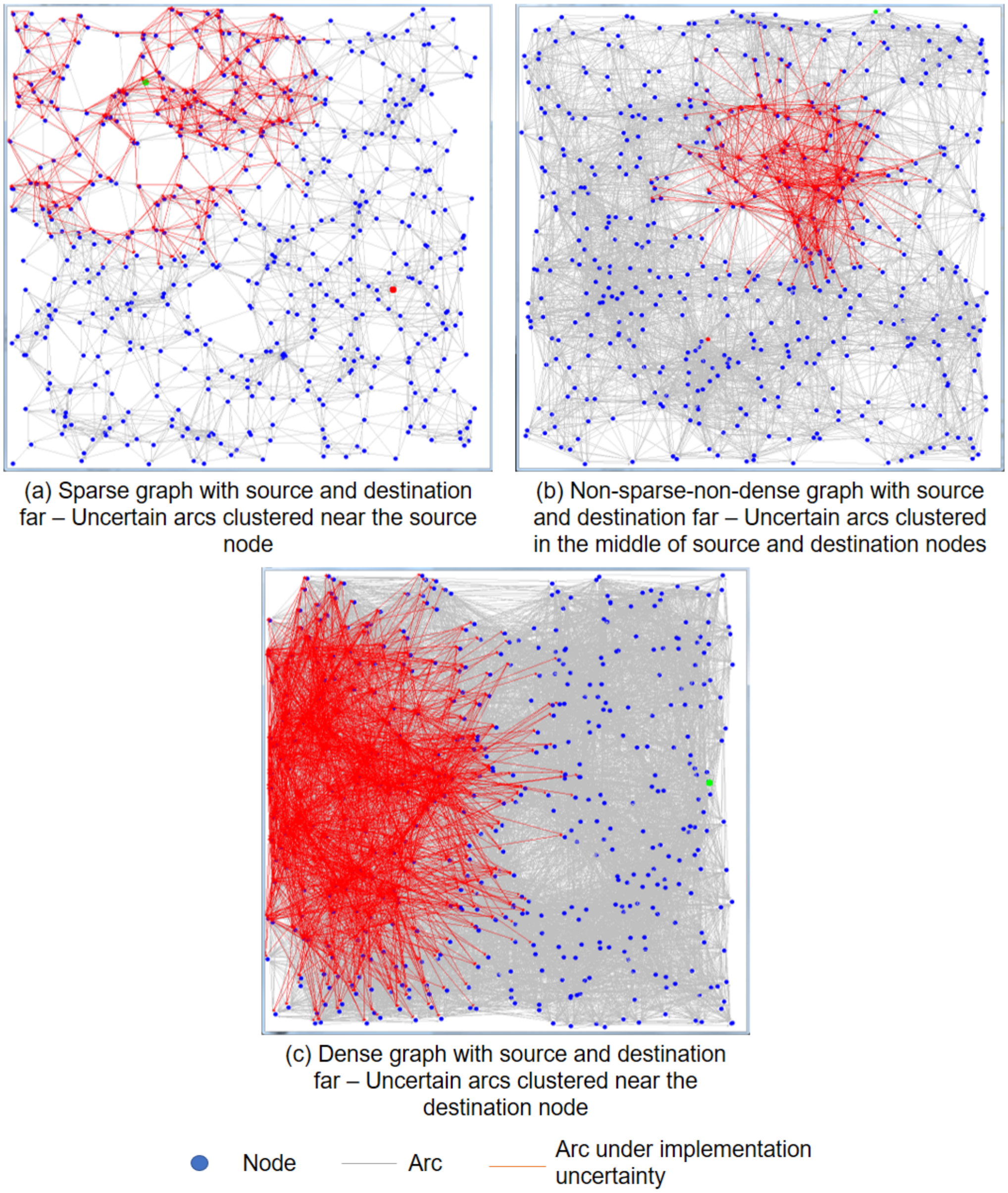
(c) Non-dense-non-sparse graph, medium percentage of arcs under implementation uncertainty, source and destination node close.

(d) Non-dense-non-sparse, with high percentage of arcs under implementation uncertainty, source and destination node at medium distance.

(e) Dense graph, low percentage of arcs under implementation uncertainty, source and destination node close.

(f) Dense graph, with medium percentage of arcs under implementation uncertainty, source and destination node at medium distance.

Figure 5.2: Graphs with arcs under implementation uncertainty uniformly distributed.




(a) Sparse graph with source and destination far – Uncertain arcs clustered near the source node

(b) Non-sparse-non-dense graph with source and destination far – Uncertain arcs clustered in the middle of source and destination nodes

(c) Dense graph with source and destination far – Uncertain arcs clustered near the destination node

Figure 5.3: Graphs with arcs under implementation uncertainty clustered in different areas.

uncertainty $U$. The percentage of arcs under uncertainty ranges from 5% to 35%, in jumps of 5%. For each problem, the deterministic solution of the SPP, $x^D$, and the robust solution $x^R$ as described in Lemma 5.3.5 are computed. In turn, the corresponding uncertain sets

$\mathcal{U}(x_C^D)$ and $\mathcal{U}(x_C^R)$, sets $\mathcal{F}'(x^D)$ and $\mathcal{F}'(x^R)$, and measures of performances are obtained by a random sample with $2000$ sample points. It is assumed that the implemented value of the uncertain variables is equal or different to the prescribed value with the same probability.

### 5.6.3 Performance Results

Figure 5.4 shows the feasibility level for the different scenarios. The results of the experiments show that the feasibility level of the deterministic solutions is more sensitive to implementation uncertainty when the arcs under uncertainty are clustered in any region due to arcs forming the shortest path become uncertain with high probability; therefore, the shortest path possess a high probability to become disconnected. The probability that arcs forming the shortest path are impacted by uncertainty is reduced when the arcs under uncertainty are uniformly randomly distributed among the graph; therefore, there is an improvement in the feasibility level when arcs under uncertainty are selected in this way. Similarly, the feasibility level of the deterministic solutions is more affected when the graph is sparse. The reason is that with fewer arcs in the graph, the probability that arcs in the shortest path are contained in the set of arcs under uncertainty $U$ increases. On the other hand, the feasibility level is less affected by uncertainty when the graph contains more arcs due to a decrease of the probability aforementioned; therefore, dense graphs provide better feasibility level. When the arcs under implementation uncertainty are clustered, the feasibility level is more affected as closer these two nodes are; the reason is that as closer they are, there is more probability that arcs forming the shortest path are impacted by uncertainty given that they are clustered near the shortest path; as farther they are, the arcs under uncertainty are more distributed among arcs not contained in the shortest path. This behavior is different when the arcs under implementation uncertainty are uniformly distributed; in this case, the closer the nodes are the better the feasibility level given that arcs are distributed among all the network and the probability that arcs in the shortest path become uncertain is small. When the arcs are uniformly distributed, the farther the nodes are there is more probability that the arcs under uncertainty contain arcs into the shortest path given that there are more arcs forming it.

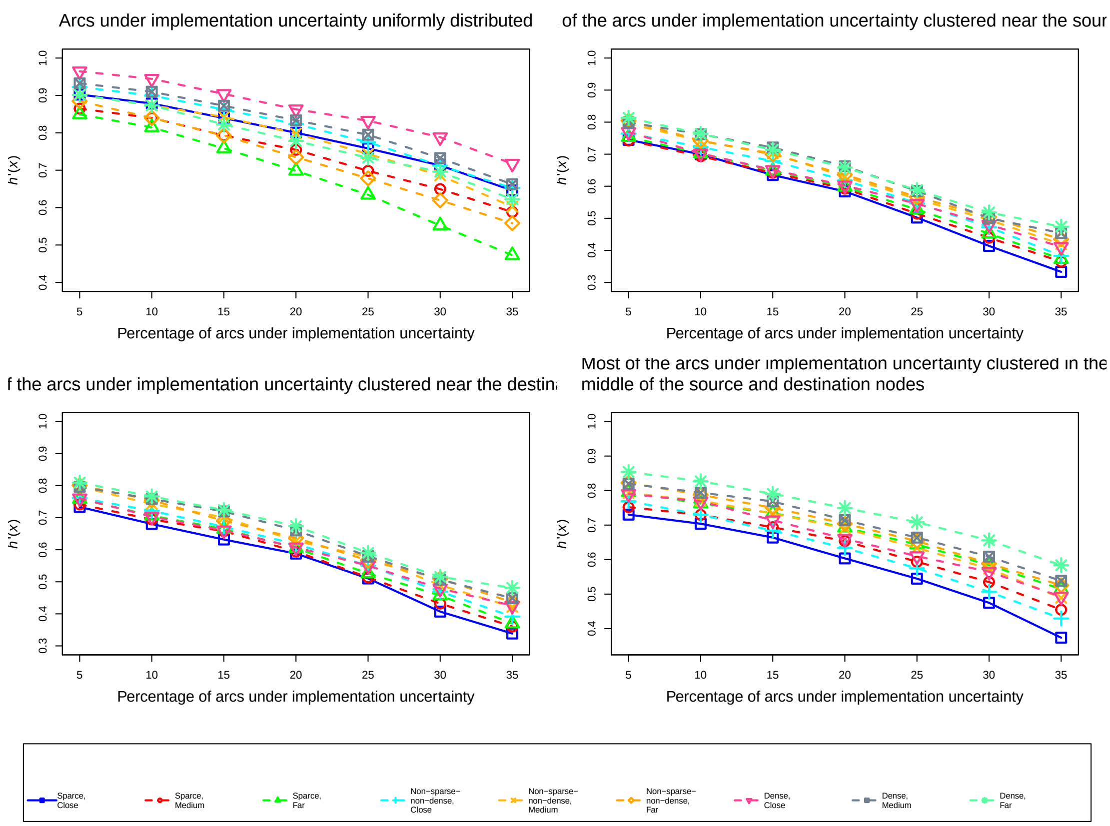



Figure 5.4: Feasibility performance ratio of the deterministic solutions $h'(x^D)$.

On the other hand, the robust solutions guarantee feasibility for any realization of the uncertainty; however, there exists an increment of the cost to guarantee such feasibility level. Figure 5.5 shows the loss of the objective value of the robust solutions against the deterministic solutions. As more arcs are under implementation uncertainty, there exists a high probability that arcs possessing low costs become uncertain. Therefore, the robust solutions should form the feasible path by considering arcs with high cost or should form the shortest path by considering a larger number of arcs. The increment of the cost of the robust solutions increases when the arcs under uncertainty are clustered in any part of the graph in comparison to the scenario when these arcs are uniformly randomly distributed; this increment of the cost is due to the robust solutions forming the feasible path with a large number of arcs than the deterministic solution does. For all scenarios, the loss of objective value is not significantly impacted by the density of the graph; though sparse graphs provide a slightly higher cost, and dense graphs provide the best value. Similarly, the distance between the source and destination nodes does not impact significantly the increment in the cost. However, when the arcs under implementation uncertainty are uniformly distributed, the closer the two nodes are, the smaller the cost increment is; on the other hand, when the arcs under uncertainty are clustered, the closer the two nodes are the higher the cost increment is.

Experimental results also show that in less than 1% of the realizations of the uncertainty, the cost of a path connecting the source and destination nodes result of the uncertainty affecting the deterministic solution possesses a higher cost than the cost of the path defined by the robust solutions; although, this difference of the cost is not significant. This phenomenon is due to the deterministic path connecting the source and destination nodes by using arcs not initially considered due to implementation uncertainty; although these arcs permit to reach the destination node, their cost is higher than other arcs, including the ones used by the robust solution.

From a practical perspective, the robust solutions are more appropriate when there exists a priority to reach the destination, even at an increment of the cost. For example, in the case of a disaster, there may be roads connecting two points that are not available, which can be modeled using implementation uncertainty. In this case, first aid supplies are transported from suppliers to regions affected by the disaster; it is a priority to reach the

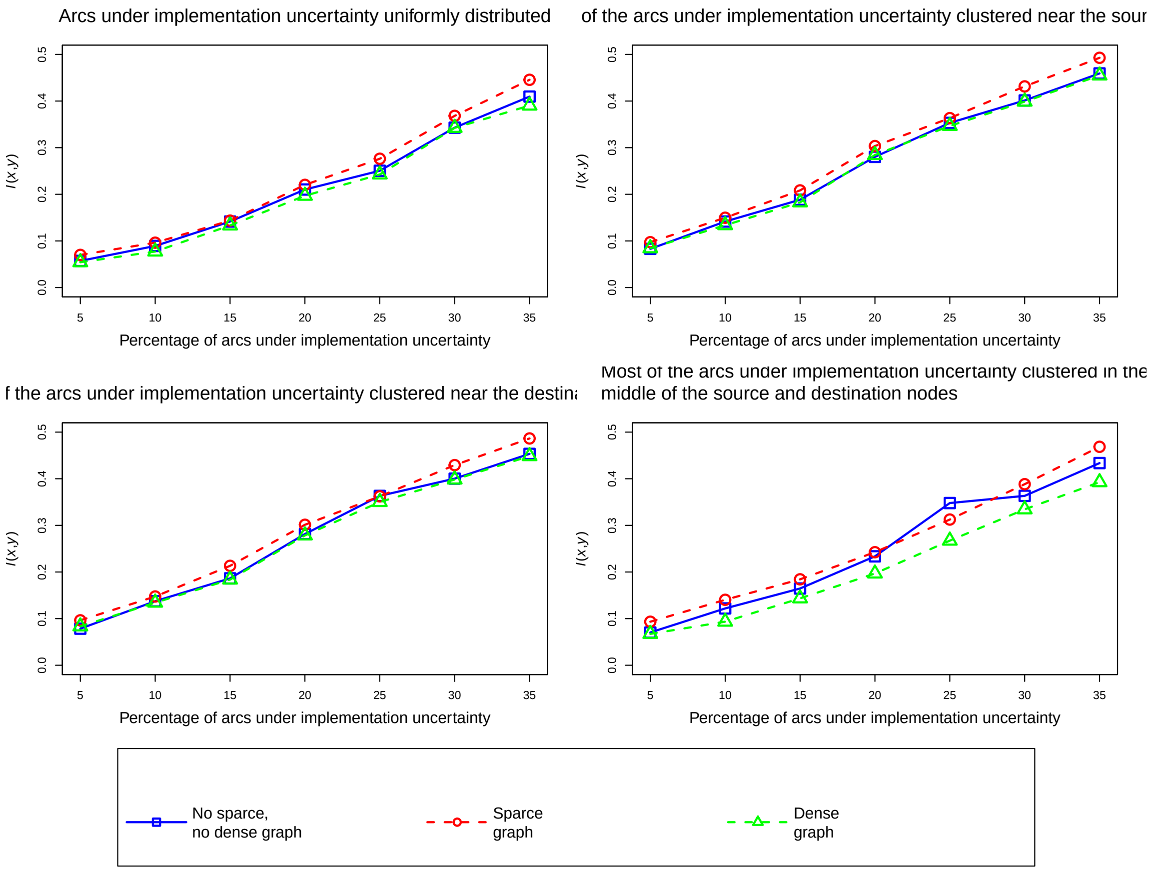



Figure 5.5: Loss of the objective performance ratio $l'(x^D, x^R)$.

destination point, even with an increment of the traveled distance since the regions cannot wait until the road is available again. A similar scenario can be seen when deployed troops need supplies and the communication network is destroyed; the priority is to find a different path to guarantee that the soldiers receive the supplies or they are extracted. Similarly, retailers such as Amazon may accept an increment in their transportation cost to satisfy customers, especially to prime members. Not complying with the delivery date affects Amazon's customer service level and their promise of service with this type of customers. Another industry interested in completing the delivery may be the food industry, especially companies distributing perishable products.

On the other hand, with a small percentage of arcs under implementation uncertainty, the increment of the cost of the robust solutions may not justify the improvement in the feasibility level, if there does not exist a high priority to reach the destination node, or there exists an option to wait until the solution can be implemented. Going back to the Amazon example, when serving regular customers Amazon may wait until the delivery is completed using the initial route since they usually provide a wide window of time to complete these deliveries and the customers are aware of it. Similarly, the delivery of low cost-not perishable products may not justify the increment in the transportation costs since they may not be a priority.

The experimental study also evaluates the performance of the two discussed methodologies to solve the RSPP in terms of their runtime; the equivalent MILP formulation and Algorithm 1. The MILP is solved using CPLEX 12.6 and Java; Algorithm 1 is implemented in Java. Table 5.1 shows the average runtime for both solutions methods when applied to digraphs with arcs under implementation uncertainty uniformly distributed, and origin and destination at a medium distance. The times are measured in milliseconds.

| | MILP | | | Algorithm 1 | | |
|---|---|---|---|---|---|---|
| Percentage of arcs under implementation uncertainty | Sparse | Non-sparse-non-dense | Dense | Sparse | Non-sparse-non-dense | Dense |
| 5% | 117,730 | 183,474 | 243,593 | 980 | 1,783 | 2,971 |
| 10 | 104,780 | 168,796 | 219,234 | 912 | 1,693 | 2,793 |
| 15 | 94,302 | 153,604 | 199,503 | 857 | 1,558 | 2,681 |
| 20 | 85,815 | 136,708 | 177,557 | 806 | 1,449 | 2,467 |
| 25 | 76,375 | 127,138 | 163,353 | 757 | 1,376 | 2,294 |
| 30 | 69,501 | 115,696 | 151,918 | 712 | 1,239 | 2,065 |
| 35 | 62,551 | 107,597 | 136,726 | 669 | 1,152 | 1,941 |

Table 5.1: Runtime of the MILP and Algorithm 1 (milliseconds).

Results in Table 5.1 show that the runtime of both solutions is impacted by the density of the graph and the percentage of uncertain variables. Both of these factors impact the number or arcs considered in the graph reducing the number of variables, in the case of the MILP, and loops in Algorithm 1. As expected, Algorithm 1 possesses a better the performance than the MILP.

## 5.7 Case Study

This section presents a case study of a distribution company that moves its products in the east of United States. The road network may be impacted by accidents and weather conditions that prevent the use of road sections. The developed concepts of robustness in the context of the SPP are applied to identify the robust routes to move their products. An analysis of the profit produced by the deterministic and robust solutions presents the attractiveness of the concepts of robustness.

### 5.7.1 Problem Description

A distribution company covers the east side of the United States. The company moves its products among California, Oregon, Idaho, Utah, Nevada, Alaska, Colorado, Washington, Wyoming, Arizona, Montana and North Dakota. Figure 5.6 shows the map with the different locations served by this company. There are 126 locations in total distributed among the states aforementioned.

There exist many source-destination pairs considered in the company's operations; however, this case study focuses on those pairs that comprise more than the 65% of their operations. These pairs and the total number of trips per week between them are:

- **Scenario 1:** Santa Fe Springs, CA to Salt Lake City, UT; total of 54 trips.
- **Scenario 2:** Santa Fe Springs, CA to Billings, MT; total of 37 trips.
- **Scenario 3:** Santa Fe Springs, CA to Upton, WY; total of 33 trips.


Figure 5.6: Locations in United States considered in the case study.

- **Scenario 4:** Redmon, OR to Chula Vista, CA; total of 27 trips.

These source and destination nodes of the four scenarios are represented in Figure 5.6. The company is interested in visiting these locations, but they also want to include other locations served by them into the route such that they can serve more than one order on the same trip.

Due to the nature of their products, the company is interested in reducing the stopping times of the vehicles and deliver them the fastest possible. Primarily, the company would like to reduce the times that their fleet is not moving. Uncertainty such as accidents impacts the traffic conditions, and the company is interested in identifying solutions that permit to achieve the destination a soon as possible for any realization of the uncertain conditions.

### 5.7.2 Solution Approach

This case study is addressed by applying the concepts of robustness developed in previous sections. The modeling of the problem is described as follows:

1. Generate a graph $G$ using every mark in Figure 5.6 as the nodes. By considering the nodes in Figure 5.6, the solutions allow visiting the rest of the locations on the map.

2. The set of arcs is generated by connecting every pair of nodes using two arcs in opposite directions. The weight of each arc is given by the distance between each pair of locations according to the information retrieved from Google Maps.

3. The set of arcs under implementation uncertainty is generated by considering the routes crossing areas with a high number of accidents. The traffic data of accidents is obtained from the National Highway Traffic Safety Administration (NHTSA) (U.S. Department of Transportation, 2018). Figure 5.7 shows an example of the data retrieved from the NHTSA; this figure shows the locations with accidents reported during 2016 in the states of California, Nevada, and Utah. Similar information is obtained for the other states served by the company.

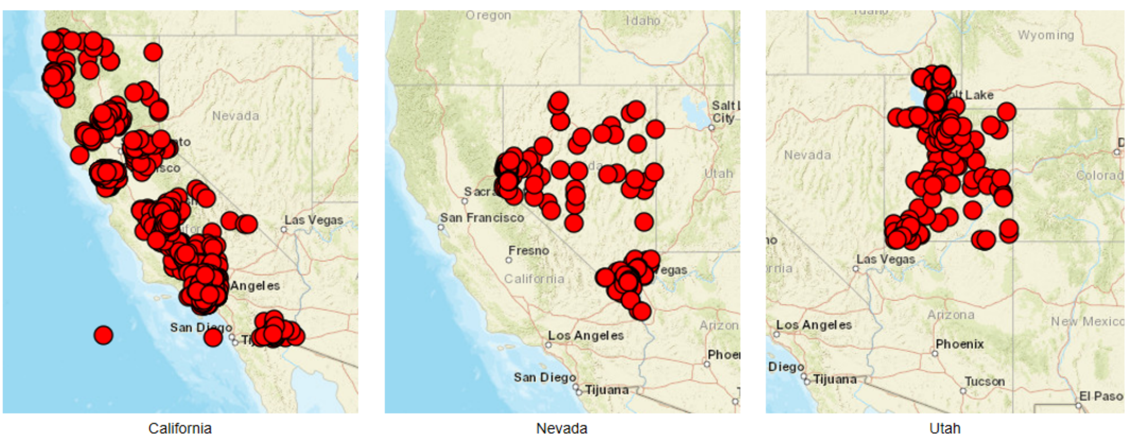



Figure 5.7: Example of accidents reported during 2016 in California, Nevada and Utah.

Based on the information from the NHTSA, the arcs under implementation uncertainty for each of the scenarios described above are considered as follows:

- Santa Fe Springs, CA to Salt Lake City, UT; arcs under uncertainty are clustered near to these two locations.
- Santa Fe Springs, CA to Billings, MT; arcs under uncertainty are clustered near to the source point and near to Salt Lake City, UT.
- Santa Fe Springs, CA to Upton, WY; arcs under uncertainty are clustered near to the source point.
- Redmon, OR to Chula Vista, CA; arcs under uncertainty are clustered around Los Angeles, CA.

For each of the scenarios, the SPP and RSPP are solved, and their performance is evaluated using the feasibility performance ratio in Definition 5.6.3 and the loss of objective performance ratio in Definition 5.6.4.

### 5.7.3 Financial Analysis

The deterministic and robust solutions of the case study can be analyzed from a financially perspective by computing the annual profit of these solutions. Information on the total number of trips per week and the average profit per trip is available; therefore, the average annual profit $AP$ of all the trips without uncertainty can be computed as follows:

$$AP = 52 \times (\#\text{trips per week}) \times (\text{average profit per trip}) \tag{5.52}$$

Where the term $52 \times (\#\text{trips per week})$ provides the total number of trips during one year. For instance, if the average profit per trip is $1,000, and there are 50 trips per week, then the average annual profit $AP$ is $2,600,000.

It can be assumed that under uncertainty the feasibility level of the deterministic solution $h(x_D)$ provides the percentage of the annual trips that will be completed during one year. Therefore, the average annual profit of the deterministic solution $AP_D$ can be computed as the average annual profit of the proportion of trips completed; $AP_D$ is computed as follows:

$$AP_D = AP \times h(x_D) \tag{5.53}$$

For instance, if the average annual profit $AP$ is $1,000,000 and the feasibility level $h(x_D) = 0.8$, then the average annual profit of the deterministic solution $AP_D$ is the 80% of the average annual profit; i.e., $AP_D = \$800,000$.

On the other hand, the robust solution $x_R$ permits to complete all the trips with an increment of the traveled distance with respect to the deterministic solution $x_D$. The increment of the traveling distance of the robust solution with respect to the deterministic solution $\Delta d(x_D, x_R)$ can be computed as follows:

$$\Delta d(x_D, x_R) = d(x_R) - d(x_D) = (1 + l(x_D, x_R)) \times d(x_D) \tag{5.54}$$

Where $d(x_R)$ is the total distance produced by the robust solution, $d(x_D)$ is the total distance produced by the deterministic solution, and $l(x_D, x_R)$ is the loss of objective performance ratio. The total annual profit of the robust solution $AP_R$ is computed as the average annual profit $AP$ minus the cost of the extra distance of the robust solution using the cost per mile $c_{pm}$; $AP_R$ can be computed as follows:

$$AP_R = AP - 52 \times (\text{\#trips per week}) \times \Delta d(x_D, x_R) \times c_{pm} \tag{5.55}$$

For instance, assume an increment of 30% of the distance and $d(x_D) = 200$ miles; if the cost per mile is $\$1.5$, there are 50 trips per wek, and $AP = \$1,200,000$, the average annual profit of the robust solution $AP_R$ using equation (5.55) is as follows:

$$AP_R = 1,200,000 - 52 \times 50 \times (1 + 0.3) \times 200 \times 1.5 = \$186,000 \tag{5.56}$$

Note that the $AP_R$ may be a negative value if the extra distance or the cost per mile are high values.

### 5.7.4 Performance Results

Table 5.2 shows the feasibility level of the deterministic solution for each of the scenarios. The results show that the feasibility levels of the scenarios with Santa Fe Springs, CA as the source are impacted by accidents or other traffic conditions around the Santa Fe Springs and San Bernardino areas. The traffic conditions around Salt Lake City, UT also impact the probability of disconnecting the path; therefore, the scenario with this location as the destination, and the scenario with Billings, MT as the destination posses lower feasibility level than the other scenarios. The deterministic route from Santa Fe Springs, CA to Upton, WY is not impacted by the traffic around Salt Lake City since the route deviates to Wyoming before reaching this location; therefore, its feasibility level is slightly improved. On the other hand, the deterministic solution from Redmon, OR to Chula Vista, CA is highly impacted by the traffic in Los Angeles since this location is part of the path and there is a high probability of unavailable roads around it.

| Scenario | Feasibility performance ratio $h(x_D)$ |
|---|---|
| 1 | 0.7653 |
| 2 | 0.782492 |
| 3 | 0.845743 |
| 4 | 0.702147 |

Table 5.2: Feasibility performance ratio of each scenario.

From the data provided by the company, the average profit per trip is $1,500. Using the total number of trips per week for each scenario and equation (5.52), the average annual profit $AP$ is $11,778,000. The average annual profit of the deterministic solution of each scenario can be computing using equation (5.53) and information in Table 5.2. Table 5.3 shows the average annual profit of the deterministic solutions for each scenario.

| Scenario | $AP_D$ |
|---|---|
| 1 | $3,223,443.60 |
| 2 | $2,258,271.91 |
| 3 | $2,176,942.48 |
| 4 | $1,478,721.58 |

Table 5.3: Annual profit of the deterministic solutions.

The total annual profit provided by the deterministic solution is $9,137,379.58.

On the other hand, the robust solutions are feasible for any realization of the traffic conditions and it is assumed that all the trips can be completed; however, there is an increase in the traveled distance to achieve such feasibility level. Table 5.4 shows the proportion of the increment in the traveled distance of the robust solutions in comparison to the deterministic solution. The scenario with Santa Fe Springs, CA and Salt Lake City, UT is the one whose robust solution possesses the highest increment in the traveled distance; this increase is due to the robust path trying to avoid the areas with high traffic near the source and the destination. The rest of the scenarios with Santa Fe Springs as the source possess small increments in the traveled distance since they avoid Salt Lake City, but the deviations are smaller than trying to avoid all the traffic around this area. The scenario from Redmon, OR to Chula Vista, CA behaves similarly to the last two mentioned scenarios.

Using the data provided by the company, the cost per mile is estimated at $1.5. The average annual profit of the robust solution for each scenario can be computing using

| Scenario | Loss of the objective performance ratio $l(x_D, x_R)$ |
|---|---|
| 1 | 0.31736 |
| 2 | 0.18746 |
| 3 | 0.153872 |
| 4 | 0.178593 |

Table 5.4: Loss of the objective performance ratio of each scenario in the case study.

equation (5.55). Table 5.5 shows the average annual profit of the robust solutions for each scenario.

| Scenario | $AP_R$ |
|---|---|
| 1 | $3,583,691.94 |
| 2 | $2,210,820.07 |
| 3 | $2,086,046.04 |
| 4 | $1,728,378.67 |

Table 5.5: Annual profit of the robust solutions.

The total annual profit of the robust solutions is $9,608,936.72.

Based on previous results, it is possible to conclude the following:

- The company may increase their profit in $609,905.43 by using the robust solutions for scenarios 1 and 4, and the deterministic solutions for scenarios 2 and 3. The cost of the extra distance for scenarios 2 and 3 is greater than the profit of the extra trips completed by the robust solution with respect to the deterministic one.

- If the company implements the robust solutions for all scenarios, the increment of the annual profit is $471,557.15.

- If the profit per trip is reduced more than 18%, the deterministic solutions provide a better overall profit than the robust ones. Figure 5.8 shows the profit of the deterministic and robust solutions for different values of the profit per trip.

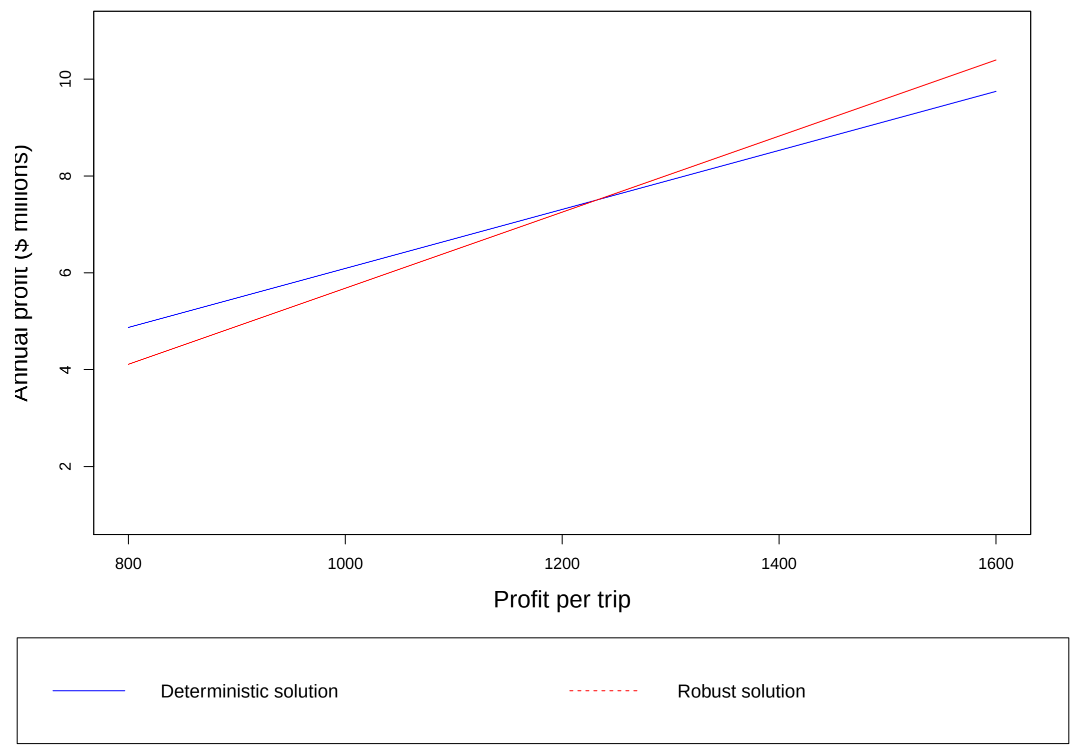



Figure 5.8: Annual profit for the deterministic and robust solutions for different values of the profit per trip.

## 5.8 Summary and Conclusions

This chapter presents the first attempt to study the shortest path problem when there may not be possible to travel through certain arcs considered in the shortest path, or one may be forced to travel through arcs not initially considered. The study of this problem is useful for transportation companies, distributors or any institution that requires moving between two points and disruptions or blocs in the communications network may exist.

This study addresses implementation uncertainty by developing a robust formulation that seeks for a path from the source node to the destination node formed by arcs not under implementation uncertainty. This work presents a solution algorithm based on Dijkstra's algorithm; the computational complexity of the proposed algorithm is the same as Dijkstra's.

The experimental study considers different scenarios representing real situations and shows that the increment in the cost of the robust solutions to guarantee a path from the source to the destination node is justified if there exists a priority in reaching the destination. Examples considering priority in the deliveries such as Amazon prime, first aid supplies, and supplies to troops are discussed. When there does not exist a high priority in reaching the destination node, this study suggests that the increment in the cost may not be justified and the deterministic shortest path may be more appropriate to use.

The case study presented in this chapter illustrates how the concepts of the SPP under implementation uncertainty and the robust concepts for this type of problem can be applied to a real problem. The results of the case study show how a company may be financially benefited by implementing robust solutions. Recommendations for each scenario are included.

# 6. CONCLUSIONS AND FUTURE RESEARCH

This chapter summarizes the main contributions from this dissertation. Finally, it provides directions on possible extensions and future research.

## 6.1 Summary

This dissertation is motivated by the limited work accounting for implementation uncertainty in binary variables and the practical applications fo this type of problem. This dissertation aimed to develop a model of this type of uncertainty and models to identify solutions immune to implementation uncertainty. The contributions of this research to the academic body of knowledge can be summarized as follows:

- The model of implementation uncertainty in binary variables presented in Chapter 2 models the existing uncertainty in the implemented value of a binary variable under this type of uncertainty. This model of implementation uncertainty allows the development of a solution space, the cornerstone of the robust formulation that permits the definition of the measures of robustness levels. Properties of the solution space permit the development of the properties of the robust formulations and robust solutions.

- Chapter 2 presents the development of a methodology to solve BLPs under implementation uncertainty that identifies solutions immune to this type of uncertainty. A characteristic of the proposed robust model is that it allows the development of a solution method based on a linear reformulation; therefore, the difference in the complexity of the deterministic and robust models may not be significant. Moreover, the RMBP permits to obtain the robust solutions by applying existing MILP solution algorithms. Some particular problem structures permit the development of more efficient solution algorithms. For instance, the use of dynamic programming to solve the RKP and the Dijkstra's based algorithm for the RSPP.

- The development of solution methodologies to address conservatism of the robust solutions. These methods permit to reduce conservatism by sacrificing feasibility with respect to the deterministic BLP, accepting a probability of degradation of the objective function value, or accepting a probability of losing protection against uncertainty. Not only the proposed methodologies reduce conservatism, but they also allow the decision maker a control of the accepted level of the conservatism.

- The application of the proposed concepts of robustness to the KP, the AP and the SSP under implementation uncertainty. These applications illustrate how the proposed concepts can be applied to problems with particular structures, and how these structures define characteristics of the robust counterparts and the robust solutions. These characteristics include the configurations of the control parameters to identify robust solutions with different properties or the development of more efficient algorithms. The experimental studies included for each application identify the characteristics of the problems and instances of the problems that make the robust solutions more attractive in terms of protection of the feasibility and degradation of the objective function value.

- A case study shows that the robust solutions may be attractive for real applications by providing a better profit than the deterministic solutions when impacted by uncertainty.

Based on the contributions of this work, it is possible to conclude that this dissertation provides the theoretical concepts necessary to address implementation uncertainty in binary variables and solve BLPs under this type of uncertainty. Furthermore, this dissertation also provides the theoretical results to solve the robust formulation and identify the set of robust solutions. Similarly, the theoretical results to address conservatism of the robust solutions of the RBIU are presented.

From a practical perspective, the proposed RBIU provides solutions that guarantee their applicability or practical meaning for any realization of the uncertainty. Similarly, the proposed methodology to solve the RBIU can make use of existing algorithms and software to solve MILPs. By applying the proposed concepts of robustness to well-known BLPs, this dissertation develops properties based on the particular problem structures, including more efficient solution algorithms. Finally, recommendations of the conditions that make robust solutions more attractive from a practical standpoint are also presented. This recommendations are also presented in the context of a case study for the SPP under implementation uncertainty.

### 6.2 Future Research

This work opens opportunities for extensions and future research. Extensions of this work include:

- The extension of the proposed concepts of implementation uncertainty to integer programming models. To the best of our knowledge, there does not exist previous work addressing this type of problem under implementation uncertainty. Integer programming models under implementation uncertainty may be useful in practical applications such as production scheduling problems and transportation problems, among others.

- The development of less conservative measures of robustness or robust formulations based on the model of implementation uncertainty for binary variables. For instance, the use of maximum deviation from optimality as a measure of optimality robustness. The use of different measures may open the opportunity to develop other solution methods.

Opportunities for future research include:

- The development of robust models accounting for data and implementation uncertainty simultaneously. These models may provide an opportunity to address a wide range of problems with higher model fidelity. For instance, networks with uncertain arc weight and uncertain availability of the nodes. Such networks may represent communication networks where the speed of the information varies, and some antennas may not be available to receive or transmit the information.

- The application of the proposed concepts of robustness to other BLPs and real problems. For instance, the set covering problem when installing emergency service centers and one of them cannot serve to an assigned region due to destruction in the transportation network or destruction of the center when a disaster impacts the area. Similarly, the shortest path problem with arcs blocked due to accidents or disasters preventing to travel from the source to the destination node.

## REFERENCES


Aissi, H., Bazgan, C., and Vanderpooten, D. Complexity of the min–max and min–max regret assignment problems. *Operations research letters*, 33(6):634–640, 2005.

Aissi, H., Bazgan, C., and Vanderpooten, D. Min–max and min–max regret versions of combinatorial optimization problems: A survey. *European journal of operational research*, 197(2):427–438, 2009.

Bartholdi III, J. J. The knapsack problem. In *Building Intuition*, pages 19–31. Springer, 2008.

Bazaraa, M. S., Jarvis, J. J., and Sherali, H. D. *Linear programming and network flows*. John Wiley & Sons, 2011.

Beale, E. M. On minimizing a convex function subject to linear inequalities. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 173–184, 1955.

Bellman, R. *Dynamic programming*. Princeton university press, 1957.

Bellman, R. On a routing problem. *Quarterly of applied mathematics*, 16(1):87–90, 1958.

Ben-Tal, A. and Nemirovski, A. Robust convex optimization. *Mathematics of operations research*, 23(4):769–805, 1998.

Ben-Tal, A. and Nemirovski, A. Robust solutions of uncertain linear programs. *Operations research letters*, 25(1):1–13, 1999.

Ben-Tal, A. and Nemirovski, A. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical programming*, 88(3):411–424, 2000.

Ben-Tal, A. and Nemirovski, A. Robust optimization–methodology and applications. *Mathematical Programming*, 92(3):453–480, 2002.

Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. *Robust optimization*. Princeton University Press, 2009.

Bertsekas, D. P. and Tsitsiklis, J. N. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16(3):580–595, 1991.

Bertsimas, D. and Sim, M. Robust discrete optimization and network flows. *Mathematical programming*, 98(1-3):49–71, 2003.

Bertsimas, D. and Sim, M. The price of robustness. *Operations research*, 52(1):35–53, 2004.

Bertsimas, D. and Weismantel, R. *Optimization over integers*, volume 13. Dynamic Ideas Belmont, 2005.

Bertsimas, D., Brown, D. B., and Caramanis, C. Theory and applications of robust optimization. *SIAM review*, 53(3):464–501, 2011.

Beyer, H.-G. and Sendhoff, B. Robust optimization–a comprehensive survey. *Computer methods in applied mechanics and engineering*, 196(33):3190–3218, 2007.

Birge, J. R. State-of-the-art-survey-stochastic programming: computation and applications. *INFORMS journal on computing*, 9(2):111–133, 1997.

Birge, J. R. and Louveaux, F. *Introduction to stochastic programming*. Springer Science & Business Media, 2011.

Bondy, J. A. and Murty, U. S. R. *Graph theory with applications*, volume 290. Citeseer, 1976.

Borgwardt, K. M. and Kriegel, H.-P. Shortest-path kernels on graphs. In *Data Mining, Fifth IEEE International Conference on*, pages 8–pp. IEEE, 2005.

Bunescu, R. C. and Mooney, R. J. A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 724–731. Association for Computational Linguistics, 2005.

Chen, S.-G. and Lin, Y.-K. An approximate algorithm for the robust design in a stochastic-flow network. *Communications in StatisticsâĂŤTheory and Methods*, 39(13):2440–

2454, 2010.

Cherkassky, B. V., Goldberg, A. V., and Radzik, T. Shortest paths algorithms: Theory and experimental evaluation. *Mathematical programming*, 73(2):129–174, 1996.

Collet, P. and Rennard, J.-P. Stochastic optimization algorithms. *arXiv preprint arXiv:0704.3780*, 2007.

Dantzig, G. B. Linear programming under uncertainty. *Management science*, 1(3-4): 197–206, 1955.

Dantzig, G. B. Discrete-variable extremum problems. *Operations research*, 5(2):266–288, 1957.

Dantzig, G. B. On the shortest route through a network. *Management Science*, 6(2): 187–190, 1960.

Deı, V. G., Woeginger, G. J., et al. On the robust assignment problem under a fixed number of cost scenarios. *Operations Research Letters*, 34(2):175–179, 2006.

Deo, N. and Pang, C.-Y. Shortest-path algorithms: Taxonomy and annotation. *Networks*, 14(2):275–323, 1984.

Desrochers, M. and Soumis, F. A generalized permanent labelling algorithm for the shortest path problem with time windows. *INFOR: Information Systems and Operational Research*, 26(3):191–212, 1988.

Dijkstra, E. W. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

El Ghaoui, L. and Lebret, H. Robust solutions to least-squares problems with uncertain data. *SIAM Journal on Matrix Analysis and Applications*, 18(4):1035–1064, 1997.

El Ghaoui, L., Oustry, F., and Lebret, H. Robust solutions to uncertain semidefinite programs. *SIAM Journal on Optimization*, 9(1):33–52, 1998.

Feillet, D., Dejax, P., Gendreau, M., and Gueguen, C. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks*, 44(3):216–229, 2004.

Fisher, M. L. and Jaikumar, R. A generalized assignment heuristic for vehicle routing. *Networks*, 11(2):109–124, 1981.

Ford Jr, L. R. Network flow theory. Technical report, RAND CORP SANTA MONICA CA, 1956.

Fredman, M. L. and Tarjan, R. E. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)*, 34(3):596–615, 1987.

Gabrel, V., Murat, C., and Wu, L. New models for the robust shortest path problem: complexity, resolution and generalization. *Annals of Operations Research*, pages 1–24, 2013.

Gabrel, V., Murat, C., and Thiele, A. Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235(3):471–483, 2014.

Gary, M. R. and Johnson, D. S. *Computers and Intractability: A Guide to the Theory of NP-completeness*, volume 29. WH Freeman and Company, New York, 2002.

Glover, F., Klingman, D., and Phillips, N. A new polynomially bounded shortest path algorithm. *Operations Research*, 33(1):65–73, 1985a.

Glover, F., Klingman, D. D., Phillips, N. V., and Schneider, R. F. New polynomial shortest path algorithms and their computational attributes. *Management Science*, 31(9):1106–1128, 1985b.

Goldberg, A. V. and Harrelson, C. Computing the shortest path: A search meets graph theory. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 156–165. Society for Industrial and Applied Mathematics, 2005.

Henig, M. I. Risk criteria in a stochastic knapsack problem. *Operations research*, 38(5):

820–825, 1990.

Hutson, K. R. and Shier, D. R. Extended dominance and a stochastic shortest path problem. *Computers & Operations Research*, 36(2):584–596, 2009.

Ji, X. Models and algorithm for stochastic shortest path problem. *Applied Mathematics and Computation*, 170(1):503–514, 2005.

Jornada, D. and Leon, V. J. Biobjective robust optimization over the efficient set for pareto set reduction. *European Journal of Operational Research*, 2016.

Kasperski, A. Discrete optimization with interval data. *Studies in fuzziness and soft computing*, 228, 2008.

Kasperski, A. and Zieliński, P. Robust discrete optimization under discrete and interval uncertainty: A survey. In *Robustness Analysis in Decision Aiding, Optimization, and Analytics*, pages 113–143. Springer, 2016.

Katriel, I., Kenyon-Mathieu, C., and Upfal, E. Commitment under uncertainty: Two-stage stochastic matching problems. *Theoretical Computer Science*, 408(2-3):213–223, 2008.

Kennington, J. and Wang, Z. A shortest augmenting path algorithm for the semi-assignment problem. *Operations Research*, 40(1):178–187, 1992.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kouvelis, P. and Yu, G. *Robust discrete optimization and its applications*, volume 14. Springer Science & Business Media, 1997.

Kruskal, J. B. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956.

Kuhn, H. W. The hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 2(1-2):83–97, 1955.

Li, Z., Ding, R., and Floudas, C. A. A comparative theoretical and computational study on robust counterpart optimization: I. robust linear optimization and robust mixed integer linear optimization. *Industrial & engineering chemistry research*, 50(18):10567–10603, 2011.

Lin, X., Janak, S. L., and Floudas, C. A. A new robust optimization approach for scheduling under uncertainty:: I. bounded uncertainty. *Computers & chemical engineering*, 28 (6):1069–1085, 2004.

Loberman, H. and Weinberger, A. Formal procedures for connecting terminals with a minimum total wire length. *Journal of the ACM (JACM)*, 4(4):428–437, 1957.

Loiola, E. M., de Abreu, N. M. M., Boaventura-Netto, P. O., Hahn, P., and Querido, T. A survey for the quadratic assignment problem. *European journal of operational research*, 176(2):657–690, 2007.

Montemanni, R. and Gambardella, L. M. An exact algorithm for the robust shortest path problem with interval data. *Computers & Operations Research*, 31(10):1667–1680, 2004.

Montemanni, R., Gambardella, L. M., and Donati, A. V. A branch and bound algorithm for the robust shortest path problem with interval data. *Operations Research Letters*, 32 (3):225–232, 2004.

Mulvey, J. M., Vanderbei, R. J., and Zenios, S. A. Robust optimization of large-scale systems. *Operations research*, 43(2):264–281, 1995.

Nachtigall, K. Time depending shortest-path problems with applications to railway networks. *European Journal of Operational Research*, 83(1):154–166, 1995.

Orda, A., Rom, R., and Sidi, M. Minimum delay routing in stochastic networks. *IEEE/ACM Transactions on Networking (TON)*, 1(2):187–198, 1993.

Polychronopoulos, G. H., Tsitsiklis, J. N., et al. Stochastic shortest path problems with recourse. 1993.

Roy, B. A propos de robustesse en recherche opérationnelle et aide à la décision. *Flexibilité et robustesse en ordonnancement*, pages 35–50, 2005.

Roy, B. Robustness in operational research and decision aiding: A multi-faceted issue. *European Journal of Operational Research*, 200(3):629–638, 2010.

Schrijver, A. *Theory of linear and integer programming*. John Wiley & Sons, 1998.

Schulz, F., Wagner, D., and Zaroliagis, C. Using multi-level graphs for timetable information in railway systems. *Algorithm engineering and experiments*, pages 43–59, 2002.

Shibuya, T. Shortest path search system, March 12 2002. US Patent 6,356,911.

Shih, W. A branch and bound method for the multiconstraint zero-one knapsack problem. *Journal of the Operational Research Society*, pages 369–378, 1979.

Soyster, A. L. Technical note-convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations research*, 21(5):1154–1157, 1973.

Steinberg, E. and Parks, M. S. A preference order dynamic program for a knapsack problem with stochastic rewards. *Journal of the Operational Research Society*, 30(2):141–147, 1979.

U.S. Department of Transportation. National highway traffic safety administration, 2018. URL `http://www.one.nhtsa.gov`.

Toth, P. and Vigo, D. *The vehicle routing problem*. SIAM, 2002.

Uryasev, S. and Pardalos, P. M. *Stochastic optimization: algorithms and applications*, volume 54. Springer Science & Business Media, 2013.

Vincke, P. Robust solutions and methods in decision-aid. *Journal of multicriteria decision analysis*, 8(3):181, 1999.

Weingartner, H. M. and Ness, D. N. Methods for the solution of the multidimensional 0/1 knapsack problem. *Operations Research*, 15(1):83–103, 1967.

West, D. B. et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.

Wets, R. J.-B. Programming under uncertainty: the equivalent convex program. *SIAM*

*Journal on Applied Mathematics*, 14(1):89–105, 1966.

Wets, R. J.-B. Stochastic programs with fixed recourse: The equivalent deterministic program. *SIAM review*, 16(3):309–339, 1974.

Wets, R. J.-B. Solving stochastic programss with simple recourse. *Stochastics: An International Journal of Probability and Stochastic Processes*, 10(3-4):219–242, 1983.

Wolsey, L. A. *Integer programming*, volume 42. Wiley New York, 1998.

Yu, G. On the max-min 0-1 knapsack problem with robust optimization applications. *Operations Research*, 44(2):407–415, 1996.

Yu, G. and Yang, J. On the robust shortest path problem. *Computers & Operations Research*, 25(6):457–468, 1998.

## APPENDIX A

## TRANSFORMATION ALGORITHM

Given an RBIU formulation with $m$ model robustness constraints and decision vectors $x$ with $n$ binary variables. Algorithm 6 shows the steps to transform RBIU into RMBP formulation. Consider $a_{i0} = c_i, \forall i$, $\delta_0 = 0$ and $b_0 = \gamma$.

**Algorithm 6** Transformation of RBIU to RMBP algorithm

**input:** RBIU formulation
**output:** RMBP formulation

**function** TRANFORMATION(RBIU formulation)
    add objective $\min \gamma$ to RMBP formulation
    **for** $j = 0$ to $m$ **do**
        **for** $i = 1$ to $n$ **do**
            **if** $x_i$ is a certain variable **then**
                add term $a_i x_i$ to constraint $j$ of RMBP formulation
            **else**
                add term $\left(\frac{a_{ij}+|a_{ij}|}{2}\right)$ to constraint $j$ of RMBP formulation
            **end if**
        **end for**
        add term $-\delta_j$ to constraint $j$ of RMBP formulation
        add inequality $\leq$ to constraint $j$ of RMBP formulation
        add right-hand-side $b_j$ to constraint $j$ of RMBP formulation
        add constraint $j$ to RMBP formulation
    **end for**
    **for** $i = 1$ to $n$ **do**
        add constraint $x_i \in \{0, 1\}$ to RMBP formulation
    **end for**
    **for** $j = 1$ to $m$ **do**
        add constraint $\delta_j \geq 0$ to RMBP formulation
    **end for**
    **Return** RMBP formulation
**end function**

Algorithm 6 requires to verify and modify the values of coefficients of each variable $x_i$ with $i = 1, ..., n$, for each constraint $j = 1, ..., m + 1$. Therefore, Algorithm 6 runs in a polynomial $\mathcal{O}(mn)$ time.

## APPENDIX B

## DYNAMIC PROGRAMMING ALGORITHM FOR THE KNAPSACK PROBLEM

Algorithm 7 presents an implementation of the dynamic programming solution for the KP as formulated in (3.1). It is assumed that the values of $a_i, \forall i$ and $b$ are nonnegative integer.

---

**Algorithm 7** Dynamic programming solutions for the KP.

---

**input:** A KP with parameters $a_i, c_i \forall i$ and $b$.
**output:** A subset of items minimizing the total cost.

**function** KNAPSACK($c_i, a_i, b, n$)
    **for** $j$ from 0 to $b$ **do**
        $m[0, j] = 0$
    **end for**
    **for** $i$ from 1 to $n$ **do**
        **for** $j$ from 0 to $b$ **do**
            **if** $w[i] \leq j$ **then**
                $m[i, j] = \max(m[i-1, j], m[i-1, j-w[i]] + v[i])$
            **else**
                $m[i, j] = m[i-1, j]$
            **end if**
        **end for**
    **end for**
    **Return** $m[n, b]$
**end function**

---

Algorithm 7 runs in $\mathcal{O}(nb)$ time (Wolsey, 1998).