Université Libre de Bruxelles
Faculté des Sciences

Advances in robust combinatorial optimization and linear programming

Martha Salazar Neumann

Supervised by

Prof. Martine Labbé

Année académique 2009-2010

Thèse présentée en vue de l'obtention du titre de Docteur en Sciences

Acknowledgments

I would like to express my special gratitude to my supervisor Prof. Martine Labbé, for her expertise, hard work and attention to details which helped me enormously in the realisation of my work. I am especially grateful for her friendly support and encouragements.

My acknowledgments also go to:

Prof. Philippe Vincke who proposed me the exciting topic of this thesis. He gave me the opportunity to know several people working in robustness.

Prof. Gilles Savard and Prof. Charles Audet of the École Polytechnique de Montréal who provided the resources to construct the source-code of one of the algorithms proposed in this thesis.

The research group on robustness organized by LAMSADE labs of the Université Paris Dauphine for the interesting seminars, discussions and comments who helped me to broaden my views on robust optimization and improve my results.

Prof. Marc Pirlot who took the time to read some of my preliminary and final manuscripts and enhanced my work with his comments and advises.

My colleagues from the G.O.M labs and S.M.G. labs, in special Vincent Ho Lee for his continuous technical assistance and computer-support and Daniele Catanzaro who helped me to revise the final version of the manuscript.

Prof.Jean Doyen who employed me as an assistant at the course of General Mathematics at the Faculty of Medicine of the Université Libre de Bruxelles. This position allowed me to finance the years of research necessary to write my thesis. His support and friendship and the encouragements of all my colleagues were a buoy in difficult moments.

Prof. Francis Buekenhout and Prof. Dimitri Leemans who allowed me to start the DEA at the Mathematics Department of the ULB. They provided me with knowledges on combinatorial geometry which helped me to have another view on the problems of my thesis.

All my family and friends in Mexico and Belgium without whose love, continuous support and encouragements, I would not have finished this thesis.

Table of Contents

List of Figures

Introduction

The construction of models that protect against uncertainty in the data, such as variability of the information, imprecision, lack of knowledge, is one of the major challenges in robust optimization.

Uncertainty affects several areas, ranging from transportation to telecommunications, to finance and to inventory management. For example, the problem of maximizing the future value of an investment portfolio is a classical case where uncertainty arises in finance and since profitability rates are uncertain, we face a linear program with uncertain objective function. In telecommunication networks, the model of a network is generally made using a weighted digraph in which the costs of arcs are associated with the transmission delays. Since such transmission delays are in general uncertain, a number of shortest path problems with uncertainty in the arc costs have to be solved to route the communications. We refer the reader to Roy (2007) for a larger view on robust optimization.

Uncertainty may concern different parts of an optimization problem, namely: the coefficients of the objective function, constraints or both. Moreover, the uncertainty set can be modeled in several ways, such as compact and convex subsets of $\mathbb{R}^n$, polytopes, Cartesian products of intervals, ellipsoids, and so on.

One of the possible approaches to face optimization problems where uncertainty occurs is to consider a worst case analysis, such as that proposed by Kouvelis and Yu (1997). The motivation for this approach is that we would like to find a solution that is relatively good in each of the possible realizations of the parameters (called scenarii) including the worst case scenario. More precisely, this consists in looking for the best solution in the worst case situation. An example of such an approach is, the problem of finding a solution with the *smallest worst case regret.* In this case, solving a problem under uncertainty is translated into finding a solution that, in terms of the objective function value, deviates the least from the optimal solution in all cases. Such a solution is called *minimax regret solution*.

Under interval uncertainty, the minimax regret versions of many polynomially solvable combinatorial optimization problems are NP-hard, see Aissi (2005). Hence, in solving these problems, the challenge to reduce the solution space becomes an important issue. In this context, to know when an element of the problem, represented by a variable, is always or never part of an optimal solution for all realization of data, (*1-persistent* and *0-persistent* variables respectively), constitutes a way to reduce the size of the problem. For instance, in project management, a project can be modeled by a directed acyclic graph where arcs represent activities. The arc lengths denote times to complete individual activities and the longest path from the start node s to the end node t provides the time necessary to complete the whole project. If the task activity times are uncertain, the challenge is to determine a set of activities that will always or never lie on the longest path. This information is useful to preprocess the problem by removing the 0-persistent variables and setting the 1-persistent variables equal to 1. One of the main goals of the thesis is to investigate these issues for some combinatorial optimization problems under uncertainty.

We also investigate the minimax regret version of linear programming problems under polyhedral uncertainty in the objective function coefficients and we give an algorithm to find an exact solution. We study the geometry of minimax regret version of linear programming problems under interval uncertainty in the objective function coefficients, we test our algorithm in this case and we give conditions under which we can discard the variables that take the value zero in each optimal solution (0-persistent variables).

In Part I we study combinatorial optimization problems under uncertainty. Specifically, we consider the minimax regret versions of the p-elements, the minimum spanning tree and the shortest path problems. We investigate the problem of finding the 1-persistent and 0-persistent variables for such problems under interval uncertainty and we give preprocessing procedures that reduce considerably the size of the minimax regret formulations.

In Chapter 1 we give a short literature survey concerning robust optimization for mixed integer programming. We introduce the notions of *uncertain combinatorial optimization problem*, *robust solution*, 0-persistent, 1-persistent variables and we illustrate these notions with the p-elements problem under interval uncertainty.

In Chapter 2 we study the minimum spanning tree problem under compact and convex uncertainty. We present localization results for scenario yielding the largest regret for a tree and in the case of interval uncertainty we give characterizations of 1-persistent and 0-persistent edges leading to recognition algorithms.

In Chapter 3 we study the uncertain versions of the *shortest path problem* that consists in finding a path of minimum weight connecting two specified nodes 1 and m, and the *single-source shortest path problem*, i.e., the problem of finding shortest paths from a fixed node 1 to every nodes of the graph. We consider both of them on finite directed graphs where arcs lengths are nonnegative intervals. In the context of the uncertain shortest path problem, we give sufficient conditions for a node to be never on a shortest path from 1 to m and sufficient conditions for an arc to be always or never on a shortest path from 1 to m.

For the uncertain single-source shortest path problem, we give sufficient conditions (i) for an arc (k, r) to be always on an optimal solution and (ii) for an arc to never be on an optimal solution. Based on these results, we present polynomial time recognition algorithms that we use to preprocess a given graph with interval uncertainty prior to the solution of the minimax regret problem. In order to test our preprocessing procedure, we propose a mixed integer programming formulation to solve the minimax regret single-source shortest path problem. We show by numerical experiments that such preprocessing procedures vastly reduce the time needed to compute a minimax regret solution.

In Part II we deal with the minimax regret linear programming problem under uncertainty in the objective function.

In Chapter 4 we investigate the properties of the minimax regret linear programming problem under compact and convex uncertainty and under polyhedral uncertainty. We give an alternative proof to the one given by Averbakh and Lebedev (2005) for the NP-hardness of the *maximum regret problem* under interval uncertainty. We present special cases when the maximum regret and the minimax regret problems are polynomially solvable. Under polyhedral uncertainty, we give an algorithm to find an exact solution to the minimax regret problem and we discuss the numerical results obtained on a large number of randomly generated instances.

Chapter 5, is devoted to the relations between the continuous 1-center problem in location theory and the minimax regret linear programming problem when the objective function coefficients are subject to interval uncertainty. Specifically, we describe the underlying geometry of this last problem, we generalize some results in location theory and we give conditions under which some variables can be removed. Finally, we test these conditions on randomly generated instances and present the conclusions.

Part I

Robust combinatorial optimization

Chapter 1

Robustness: the minimax models and definitions

1.1 Introduction

The input data of many combinatorial problems may be not known with certainty in real applications. Consider for instance, the design of a communication network where routing delays on links are uncertain. It is desirable to construct a configuration that protects the network against the worst possible scenario.

Combinatorial optimization under uncertainty deals with these kinds of problems and to know when a decision variable is 1-persistent or 0-persistent, i.e., when an element of the problem, represented by a variable is always or never part of an optimal solution for all realization of data, constitutes a useful information. In the case of interval uncertainty, these entities can be used for instance to preprocess a given uncertain combinatorial problem removing the 0-persistent variables and setting the 1-persistent ones equal to 1. We illustrate these notions with the p-elements problem under interval uncertainty and we give characterizations of the 0-persistent and 1-persistent variables by providing polynomial algorithms to find such variables.

1.2 Problem definition and notation

Combinatorial optimization problems is a class of optimization problems where the variables are binary.

Consider the class of Combinatorial Optimization Problems **COP**:

$$\text{minimize} \ \sum_{i \in I} c_i x_i,$$

$$\text{subject to} \ \ x \in X \subseteq \{0,1\}^n,$$

where $I = \{1, \ldots, n\}$. Assume that the coefficients of the objective function are uncertain. We shall call this class of problems *Uncertain Combinatorial Optimization Problems* **UCOP** and we shall denote by **c** the vector of uncertain objective function coefficients.

In this thesis we model uncertainty using the concept of *scenario*. A scenario is an assignment of possible values to each uncertain parameter of the problem. We denote by S the set of all possible scenarii and by D the set of all possible values for the uncertain coefficients, which we shall call *uncertainty set*. For each $s \in S$, we shall denote by $c^s = (c_1^s, \ldots c_n^s) \in D$ the vector corresponding to scenario s, where c_i^s is the value of the coefficient i in that scenario. In this case, to each $c^s \in D$ corresponds a deterministic combinatorial optimization problem:

$$\text{minimize} \ \sum_{i \in I} c_i^s x_i,$$

$$\text{subject to} \ \ x \in X \subseteq \{0,1\}^n.$$

We denote by $X^*(s)$ the set of all optimal solutions of the problem for the scenario s and by x^{*s} a generic element of $X^*(s)$.

To evaluate a solution $x \in X$ under the scenario s, we use a function $f : X \times D \to \mathbb{R} : (x, c^s) \to f(x, c^s)$. In this work, we assume that this function is linear,

$$f(x, c^s) = \sum_{i \in I} c_i^s x_i.$$

An optimal solution x^{*s} corresponding to scenario s satisfies

$$f(x^{*s}, c^s) = \min_{x \in X} f(x, c^s).$$

There are several ways to face a UCOP. Kouvelis and Yu (1997) present the state of the art in robust discrete optimization, various definitions of robustness in optimization problems under uncertainty, and a large list of open problems. Specifically, the authors suggest the following three definitions of robustness (robust solutions).

Definition 1.1 *We call the* minimax problem*, the problem to find a solution* $x_a \in X$ *such that*

$$\max_{c^s \in D} f(x_a, c^s) = \min_{x \in X} \max_{c^s \in D} f(x, c^s),$$

and its optimal solution x_a *a* minimax solution*.*

Definition 1.2 *Consider the function*

$$R_{max} : X \to \mathbb{R} \ : \ x \to R_{max}(x)$$

$$R_{max}(x) = \max_{c^s \in D} \Big(f(x, c^s) - f(x^{*s}, c^s) \Big).$$

Given a solution $x \in X$*, we call the value* $R_{max}(x)$ *the* maximum regret of x*. The optimization problem called* minimax regret problem *consists in finding a solution* $y \in X$ *such that*

$$R_{max}(y) = \min_{x \in X} R_{max}(x),$$

where y *is called a* minimax regret solution *of the uncertain combinatorial problem.*

Definition 1.3 *The* relative robust solution x_r *is such that*

$$\max_{c^s \in D} \frac{f(x_r, c^s) - f(x^{*s}, c^s)}{f(x^{*s}, c^s)} = \min_{x \in X} \max_{c^s \in D} \frac{f(x, c^s) - f(x^{*s}, c^s)}{f(x^{*s}, c^s)}.$$

Definition 1.4 *A* minimax worst scenario *for a solution* x *is a scenario in which* $f(x, c^s)$ *is maximum.*

Definition 1.5 *A* worst minimax regret scenario *for a solution* x *is a scenario in which* $f(x, c^s) - f(x^{*s}, c^s)$ *is maximum.*

Bertsimas et al. (2005) study the "persistency" of a decision variable, i.e., the probability that it is part of an optimal solution and give a model to compute the persistency of a decision variable in discrete optimization problems under probabilistic information on the objective coefficients. Inspired by this idea, we shall introduce the following definitions.

Definition 1.6 *A variable* x_i *is* 0-persistent *if* $x_i = 0$ *for all optimal solutions* x *of the problem and for all scenarii.*

Definition 1.7 *A variable* x_i *is* 1-persistent *if* $x_i = 1$ *for all optimal solutions of the problem and for all scenarii.*

The following definitions can be found in Yaman et al. (2001) in the context of the robust minimum spanning tree problem.

Definition 1.8 *A solution* $x \in X$ *is a* weak solution *if it is an optimal solution for some scenario.*

Definition 1.9 *A variable* x_i *is a* weak variable *if there exists a weak solution* x *for which* $x_i = 1$.

Definition 1.10 *A variable* x_i *is a* strong variable *if, for each scenario, there exists an optimal solution for which* $x_i = 1$.

In the last years considerable research has been developed in robust optimization and several papers have focused on mixed integer programming and on combinatorial optimization under uncertainty. In order to give an idea of the different techniques to face a robust optimization problem and the several ways under which uncertainty can be modeled, we shall give a short literature survey concerning robust optimization for mixed integer programming. Literature surveys concerning robust versions of three combinatorial problems, the p-elements problem, the minimum spanning tree and the shortest path problems, shall be given in Chapters 1 to 3. Complexity results and approximation algorithms for other robust combinatorial optimization problems can be found in Aissi (2005). For a survey concerning minimax and minimax regret combinatorial optimization problems, see Aissi et al. (2009).

1.3 Robust optimization for mixed integer programming

Let c, l, u be n-vectors, let A be an $m \times n$ matrix, and b be an m-vector. We consider the following deterministic mixed integer programming on a set of n variables, the k first of which are integer:

$$\begin{aligned} &\text{minimize} \;\; cx \\ &\text{subject to} \;\; Ax \leq b \\ &l \leq x \leq u \\ &x_i \in \mathbb{Z} \qquad i = 1, \ldots, k. \end{aligned}$$

Bertsimas and Sim (2003) propose a robust formulation of mixed integer programming problems for the case where both, the data in the constraints and the coefficients of the objective function are subject to uncertainty. When

ROBUST MIXED INTEGER PROGRAMMING ELLIPSOIDAL AND INTERVAL UNCERTAINTY SETS		
Minimax regret		
	Algorithms & heuristics	Theory & complexity
c with interval uncertainty	Averbakh (2000) Kasperski and Zielinski (2004) Aissi (2005)	Averbakh (2000) Kasperski and Zielinski (2004) Aissi (2005)
Other robust counterparts		
	Algorithms & heuristics	Theory & complexity
c with ellipsoidal uncertainty	Bertsimas and Sim (2004a)	Bertsimas and Sim (2004a)
A,b,c with interval uncertainty	Bertsimas and Sim (2003)	Bertsimas and Sim (2003)
c with interval uncertainty	Bertsimas and Sim (2003) Atamturk (2006) Averbakh (2000)	Bertsimas and Sim (2003) Atamturk (2006) Averbakh (2000)

Table 1.1: Robust mixed integer programming with ellipsoidal or interval uncertainty sets.

only the coefficients of the objective function are uncertain and the problem is a 0-1 optimization problem on n variables, Bertsimas and Sim (2003) consider a UCOP where each $\mathbf{c}_j \in [c_j, c_j + d_j]$, $d_j \geq 0$, and propose the following robust formulation,

$$\begin{aligned} \text{minimize} \quad & cx + \max_{\{J : J \subseteq I, |J| \leq \Gamma\}} \sum_{j \in J} d_j x_j \qquad \text{(RF)} \\ \text{subject to} \quad & x \in X \subseteq \{0,1\}^n, \end{aligned}$$

where $I = \{1, \ldots, n\}$. In this model a robust solution, is a solution that minimizes the maximum cost when at most Γ of the objective function coefficients are allowed to change. The parameter Γ controls the level of robustness in the objective function. Additionally Bertsimas and Sim (2003) propose an algorithm for the special case of the robust network flow. It solves the robust counterpart by solving a polynomial number of nominal minimum cost flow problems in a modified network.

Atamturk (2006) introduces alternative formulations to robust mixed 0-1 programming with interval uncertain objective coefficients

$$\min_{\psi,x}\{\psi : \psi \geq cx, x \in X \text{ for all } c \in B\},$$

where $X \subseteq \{0,1\}^n$ and $B = \{c \in \mathbb{R}^n : \pi c \leq \pi_0,\ a \leq c \leq a+g\}$, is a rational polytope defined by bounds a, $a+g$ and $\pi c \leq \pi_0$ is a "budget constraint" representing the allowed uncertainty in the objective coefficients. They present computational experiments for the proposed formulations. Averbakh (2000) proposes a general approach for finding minimax regret solutions for a class of combinatorial problems with interval uncertain objective function coefficients. The approach is based on reducing a problem with uncertainty to a number of problems without uncertainty. Kasperski and Zielinski (2004) consider a similar class of problems and present a polynomial time approximation algorithm for them.

Bertsimas and Sim (2004a) deal with robust discrete optimization under ellipsoidal uncertainty sets. It is shown that the robust counterpart of a discrete optimization problem with correlated objective function data is NP-hard even though the original problem is polynomially solvable. For uncorrelated data and identically distributed data, it is proved that the robust counterpart retains the complexity of the original problem. Table 1.1 presents a classification of the different works on robust mixed integer programming.

1.4 Combinatorial problems with interval uncertainty

In this section we shall show that, under interval uncertainty for the minimax and the minimax regret versions of combinatorial optimization problems, it is possible to discard the 0-persistent variables and set the 1-persistent ones equal to 1. This fact is a direct consequence of the following two results.

Theorem 1.1 *[Aissi (2005)] Let x be a minimax regret solution for a UCOP and suppose that $D = \prod_{i\in I} \left[\underline{c_i}, \overline{c_i}\right]$, where $I = \{1\dots n\}$. Then x is a weak solution.*

Proof. Assume that $x \in X$ is not weak. Let x' be another solution which is optimal when coefficients $c^{s'}$ are such that

$$c_i^{s'} = \begin{cases} \overline{c}_i \text{ if } x_i = 0, \\ \underline{c}_i \text{ if } x_i = 1. \end{cases}$$

Then, for any scenario, it holds that

$$f(x,c^s)-f(x',c^s)=\sum_{i\in I}c_i^s x_i-\sum_{i\in I}c_i^s x_i' =$$

$$\sum_{i\in\{j\in I|x_j=1,\ x_j'=0\}} c_i^s x_i - \sum_{i\in\{j\in I|x_j'=1,\ x_j=0\}} c_i^s x_i' \geq$$

$$\sum_{i\in\{j\in I|x_j=1,\ x_j'=0\}} \underline{c}_i x_i - \sum_{i\in\{j\in I|x_j'=1,\ x_j=0\}} \overline{c}_i x_i' =$$

$$\sum_{i\in I}c_i^{s'} x_i-\sum_{i\in I}c_i^{s'} x_i' = f(x,c^{s'})-f(x',c^{s'}).$$

Since x is not weak and x' is optimal with coefficient vector $c^{s'}$, we have

$$\sum_{i\in I}c_i^{s'} x_i = f(x,c^{s'}) > f(x',c^{s'}) = \sum_{i\in I}c_i^{s'} x_i'$$

and thus $f(x,c^s) > f(x',c^s)$ for all scenarii s. Now consider the worst minimax regret scenario s_0 for the solution x'. Then, we have

$$R_{max}(x')=\sum_{i\in I}c_i^{s_0}x_i'-\sum_{i\in I}c_i^{s_0}x_i^{*s_0} < \sum_{i\in I}c_i^{s_0}x_i-\sum_{i\in I}c_i^{s_0}x_i^{*s_0} \leq R_{max}(x).$$

Hence, x cannot be a minimax regret solution. □

A corresponding result can be derived for the minimax solutions. The proof of it, being straightforward, is omitted.

Theorem 1.2 *Consider a UCOP where* $D=\prod_{i\in I}\left[\underline{c_i},\overline{c_i}\right]$. *Let* s' *be the scenario for which the coefficients* $c_i^{s'}=\overline{c}_i$ *for all* $i\in\{1,\ldots,n\}$. *The set of all optimal solutions* $X^*(s')$ *of the problem*

$$\begin{array}{ll} \textit{minimize} & c^{s'}x \\ \textit{subject to} & x\in X\subseteq\{0,1\}^n, \end{array}$$

coincides with the set of minimax solutions. In particular, every minimax solution is a weak solution.

In the case of interval uncertainty, Theorems 1.1 and 1.2 imply that we can restrict to search for minimax and minimax regret solutions to the set of weak solutions. Hence, we can discard the 0-persistent variables and set the 1-persistent ones equal to 1.

1.5 The p-elements problem: 1-persistent and 0-persistent variables

In this section, we shall consider the p-elements problem, i.e., given m elements each one endowed with a real number representing its cost, the problem consists in determining p elements out of m with the smallest sum of costs. An optimal solution to this problem is given by a set of p elements with the smallest associated costs and can be solved in $O(m)$ time in the deterministic case, see Averbakh (2001) or Cormen et al. (1991). It is trivial when the data are fixed, but is not so when the data are not known with certainty. The minimax regret p-elements problem has been studied by Averbakh (2001) in the case where the associated costs c_i take any value in an interval $\left[\underline{c_i}, \overline{c_i}\right]$. He has presented an algorithm that solves this problem in $O((\min\{p, m-p\})^2 m)$ time. In the same paper, Averbakh (2001) shows that the problem is NP-hard in the case of an arbitrary finite set of scenarii. To the best of our knowledge, there are only two minimax regret combinatorial optimization problem that has been proved to be polynomial in the case of interval uncertainty, the minimax regret interval p-elements problem and minimax regret min cut interval problem, see Aissi et al. (2008).

1.5.1 Definition of the problem

Let $\mathcal{E}$ be a finite set of cardinality m. The p-elements problem consists in picking p elements out of $\mathcal{E}$ with the smallest sum of costs. In this problem, for a fixed $p < m =\mid \mathcal{E} \mid$, we define

$$\mathcal{X}_p(\mathcal{E}) = \{W \subset 2^{\mathcal{E}} : |W| = p\},$$

as the set of all subsets of $\mathcal{E}$ with cardinality p. That is, $\mathcal{X}_p(\mathcal{E})$ is the set of all feasible subsets of $\mathcal{E}$. We can identify $\mathcal{X}_p(\mathcal{E})$ with the subset of vectors of dimension m with exactly p coordinates equal to 1, i.e.,

$$X = \{x \in \{0, 1\}^m : \sum_{i \in I} x_i = p\}.$$

where I is the set of indices $I = \{1, \dots m\}$. If c_i represents the cost associated to element i, the problem can be formulated as follows

$$\min_{x \in X} \sum_{i \in I} c_i x_i.$$

We call this version of the p-elements problem, where the costs are fixed, the deterministic case. However, in the following, we shall consider the case in which the costs are not known with certainty. Let S be the set of scenarii for the values of the elements of $\mathcal{E}$ and let D be the uncertainty set.

1.5.2 The p-elements problem: interval uncertainty

In this subsection we shall consider the p-elements problem under interval uncertainty. We shall give characterizations of the 0-persistent and 1-persistent variables and an algorithm to find such variables. But first we shall give some other definitions.

Definition 1.11 *Assume that* $D = \prod_{i=1}^{m} [\underline{c_i}, \overline{c_i}]$. *Let*

$$E^-(k) = \{i \in \mathcal{E} : \ \bar{c}_i < \underline{c}_k\}$$

$$E^+(k) = \{i \in \mathcal{E} : \ \underline{c}_i > \bar{c}_k\}$$

$$E^=(k) = \{i \in \mathcal{E} : i \neq k \text{ and } \exists s \in S \text{ such that } c_i^s = c_k^s\}.$$

Remark that $E^=(k) \cup E^-(k) \cup E^+(k) \cup \{k\} = \mathcal{E}$.

Theorem 1.3 *Suppose that* $D = \prod_{i=1}^{m} [\underline{c_i}, \overline{c_i}]$. *The element k is 1-persistent if and only if* $\mid E^=(k) \cup E^-(k) \mid < p$.

Proof. Let k be an element of $\mathcal{E}$ such that, $\mid E^=(k) \cup E^-(k) \mid < p$, then for all $s \in S$, all minimum p-elements sets contain an element of $E^+(k) \cup \{k\}$, and then we must choose the element k. Thus for all $s \in S$, and for all $x^{*s} \in X^*(s)$, $k \in x_s^*$, thus k is a 1-persistent element. Now if k is such that $|E^=(k) \cup E^-(k)| \geq p$, by definition of $E^=(k)$ and $E^-(k)$ there exists a scenario $s' \in S$ such that $c_j^{s'} \leq c_k^{s'}$ for all $j \in E^=(k) \cup E^-(k)$. Hence under the scenario s' there exists a minimum p-elements set which does not contain the element k and then k is not a 1-persistent element. □

Theorem 1.4 *Suppose that* $D = \prod_{i=1}^{m} [\underline{c_i}, \overline{c_i}]$. *The element k is 0-persistent if and only if* $\mid E^-(k) \mid \geq p$.

Proof. If k is a 0-persistent element, for all $s \in S$, and for all $x^{*s} \in X^*(s)$, $k \notin x^{*s}$. This implies that for all $s \in S$, $x^{*s} \subseteq E^-(k)$, hence $\mid E^-(k) \mid \geq p$. Now suppose that k is such that $\mid E^-(k) \mid \geq p$. By definition of $E^-(k)$, we know that for all $s \in S$ all minimum p-elements sets of $\mathcal{E}$ are included in $E^-(k)$, thus do not contain the element k and k is a 0-persistent element. □

The last two theorems allow us to derive algorithms to find all the 1-persistent and 0-persistent elements.

Algorithm to find the 0-persistent elements

1. For each $k \in \mathcal{E}$ obtain the set $E^-(k) = \{i \in \mathcal{E} : \ \overline{c}_i < \underline{c}_k\}$.
2. Compute $\mid E^-(k) \mid$.
3. If $\mid E^-(k) \mid \geq p$, k is a 0-persistent element, otherwise k is not 0-persistent.

Algorithm to find the 1-persistent elements

1. For each $k \in \mathcal{E}$ obtain the set $E^-(k) = \{i \in \mathcal{E} : \ \overline{c}_i < \underline{c}_k\}$.
2. Obtain the set $E^=(k) = \{i \in \mathcal{E} : i \neq k \ and \ \exists s \in S$ such that $c_i^s = c_k^s\}$
3. Compute $\mid E^=(k) \cup E^-(k) \mid$.
4. If $\mid E^=(k) \cup E^-(k) \mid < p$, k is a 1-persistent element, otherwise k is not 1-persistent.

For each $k \in \mathcal{E}$, the problem of finding the sets $E^-(k)$ and $E^=(k)$ can be solved in $O(m)$ time. Since we have m elements, the above algorithms find the 1-persistent variables and the 0 -persistent variables in $O(m^2)$ time each one of them. Since the complexity of Averbakh's algorithm is $O((\min\{p, m-p\})^2 m)$, if p is of the same order of m, the presented preprocessing can be useful.

Chapter 2

The robust spanning tree problem: compact and convex uncertainty

2.1 Introduction

In this chapter, we consider the uncertain minimum spanning tree problem where edge costs belong to a compact and convex subset of $\mathbb{R}^m$. Theoretical results about this problem are given. We establish localization results for scenarii yielding the largest regret for a tree. Characterizations of 1-persistent and 0-persistent edges for the spanning tree problem under interval uncertainty are also given and polynomial time recognition algorithms are proposed. Such characterizations are based on the topology of the graph combined with the structure of the uncertainty set. These results have been developed with the goal to reduce the time to compute a robust minimum spanning tree and have been published in Salazar-Neumann (2007a).

In the literature, several papers about the robust versions of the minimum spanning tree problem can be found. Kouvelis and Yu (1997) prove that in the case of discrete uncertainty, the minimax minimum spanning tree problem is NP-hard for a bounded number of scenarii and strongly NP-hard for an unbounded number of scenarii and the minimax regret minimum spanning tree is NP-hard for a bounded number of scenarii. Aissi (2005) shows that this last problem is strongly NP-hard for an unbounded number of scenarii, see Table 2.1.

The minimax minimum spanning tree problem with interval uncertainty is polynomial, see Yaman et al. (2001), while the minimax regret minimum spanning tree problem with interval uncertainty is NP- complete, see Aron

THE MINIMUM SPANNING TREE PROBLEM DISCRETE UNCERTAINTY SET		
Minimax regret		
	Algorithms & heuristics	Theory and complexity
MST U bounded	Aissi (2005)	NP-hard Kouvelis and Yu (1997)
MST U unbounded		strongly NP-hard Kouvelis and Yu (1997) Aissi (2005)
Minimax		
	Algorithms & heuristics	Theory and complexity
MST U bounded	Aissi (2005) Kouvelis and Yu (1997)	NP-hard Kouvelis and Yu (1997)
MST U unbounded	Kouvelis and Yu (1997)	strongly NP-hard Kouvelis and Yu (1997)

Table 2.1: The minimum spanning tree problem: discrete uncertainty set.

and Hentenryck (2004). Averbakh and Lebedev (2004), prove that this problem is NP-hard even if all intervals of uncertainty are equal to $[0, 1]$ and they prove that this problem is polynomially solvable in the case where the number of edges with uncertain lengths is fixed or is bounded by the logarithm of a polynomial function of the total number of edges, see Table 2.2.

Recently, Montemanni (2006), Montemanni and Gambardella (2005a) and Aissi (2005) present relaxation procedures to solve minimax regret combinatorial problems in the case of interval uncertainty. Yaman et al. (2001), give a mixed integer programming formulation for the minimax regret spanning tree problem with interval uncertainty and show that characterizations of strong edges and 0-persistent edges can be useful for pre-processing this mixed integer programming. Montemanni and Gambardella (2005b) provide a branch and bound algorithm to solve the minimax regret spanning tree problem under interval uncertainty.

2.2 Notation and definitions

Let G be a finite, connected and undirected graph, we denote by $V(G)$ and $E(G)$ the set of vertices and edges respectively. We suppose that $|V(G)| = n$,

THE MINIMUM SPANNING TREE PROBLEM INTERVAL UNCERTAINTY	
Minimax regret	
Algorithms & heuristics	Theory and complexity
Branch and bound Montemanni and Gambardella (2005b) Montemanni (2006) Yaman et al. (2001) Kasperski and Zielinski (2004) approximation algorithm	NP-complete Aron and Hentenryck (2004) Averbakh and Lebedev (2004) polynomial cases
Minimax	
Algorithms & heuristics	Theory and complexity
algorithm Yaman et al. (2001)	polynomial Yaman et al. (2001)

Table 2.2: The minimum spanning tree problem: interval uncertainty set.

$|E(G)| = m$. Consider the following combinatorial problem called *minimum spanning tree problem* Given G and cost c_i associated with each edge $i \in E(G)$, we want to find a connected subgraph G' of G which contains all its vertices and whose cost $\sum_{i \in E(G')} c_i$, is minimum. It is clear that such a subgraph is a *tree*, that is to say, is connected and acyclic. Since G' is assumed to be connected, and its cost is minimum, none of its edges can be removed without destroying its connectivity. A subgraph of G, which contains all of its vertices is called a *spanning subgraph.* A spanning subgraph of G which is a a tree is called a *spanning tree* of G. A minimum spanning tree can be obtained, in the deterministic case, in time $O(min\{n^2, mlogn\})$ using Prim's algorithm or in time $O(mlogm)$ using Kruskal's algorithm, see Cormen et al. (1991).

We denote by $\mathcal{T}(G)$ the set of spanning trees of G. For $T \in \mathcal{T}(G)$, we denote by $E(T)$ the set of edges of T. Let S be the set of scenarii for the costs of edges of G and let D be the uncertainty set. We shall assume that D is a compact and convex subset of $\mathbb{R}^n$.

We denote by $\mathcal{T}^*(G, s)$ the set of minimum spanning trees of G for the scenario s and by T^{*s} a generic element of $\mathcal{T}^*(G, s)$. For a scenario $s \in S$, we denote by $s(k, t)$ the scenario for which $(c_1^{s(k,t)}, \ldots c_k^{s(k,t)}, \ldots c_m^{s(k,t)}) = c^s + te_k$, where $t \in \mathbb{R}$ and e_k is the kth canonical vector in $\mathbb{R}^m$.

If T is a tree of G and D is a compact and convex subset of $\mathbb{R}^m$, then for each tree the minimax best and worst scenarii are on the border of D. Since the worst scenario for a tree T is a solution of the problem

$$\max_{c^s \in D} \sum_{i \in E(T)} c_i^s$$

then if D is a polytope, we can find the minimax worst scenario for a tree in polynomial time.

Consider the functions

$$r : \mathcal{T}(G) \times D \to \mathbb{R} \ : \ (T, c^s) \to r(T, c^s)$$

$$r(T, c^s) = \max_{T' \in \mathcal{T}(G)} \Big(f(T, c^s) - f(T', c^s) \Big) = f(T, c^s) - f(T^{*s}, c^s).$$

For any T in $\mathcal{T}(G)$, we shall call the value

$$R_{max}(T) = \max_{c^s \in D} \Big(f(T, c^s) - f(T^{*s}, c^s) \Big),$$

the maximum regret for the tree T. If $\tilde{T} \in \mathcal{T}(G)$ is such that

$$R_{max}(\tilde{T}) = \min_{T \in \mathcal{T}(G)} R_{max}(T),$$

then we call $\tilde{T}$ a minimax regret minimum spanning tree.

We observe that for a fixed tree $T \in \mathcal{T}(G)$ the function r becomes the function

$$r_T : D \to \mathbb{R} \ : \ c^s \to r_T(c^s)$$

$$r_T(c^s) = f(T, c^s) - f(T^{*s}, c^s).$$

Thus we can also write the maximum regret of T as

$$R_{max}(T) = \max_{c^s \in D} r_T(c^s).$$

2.3 The worst minimax regret scenarii for a tree

In order to give the location of the worst and best minimax regret scenarii for a tree (Theorem 2.3), we shall first study the properties of the functions $c^s \to f(T^{*s}, c^s)$ and r_T.

Theorem 2.1 *If T^{*s} denotes an optimal tree of G for the scenario s then the functions $c^s \to f(T^{*s}, c^s)$, and r_T are continuous and piecewise linear over D.*

Proof. The function $c^s \to f(T^{*s}, c^s)$ is such that

$$f(T^{*s}, c^s) = \min_{T \in \mathcal{T}(G)} f(T, c^s)$$

since the number of spanning trees of G is finite, this function is the minimum of a finite number of linear functions and then is continuous. For a fixed T the function $c^s \to f(T, c^s)$ is linear thus r_T is the difference of a linear function and a piecewise linear function. Hence r_T is continuous and piecewise linear. □

In order to study the differentiability of $c^s \to f(T^{*s}, c^s)$ and r_T consider the following lemma.

Lemma 2.1 *Let $t > 0$, and let $c^s \in D$ such that $c^{s(k,-t)} \in D$. If $T_0 \in \mathcal{T}^*(G, s)$ and $k \in E(T_0)$, then $T_0 \in \mathcal{T}^*(G, s(k, -t))$. Moreover $k \in E(T)$ for all $T \in \mathcal{T}^*(G, s(k, -t))$.*

Proof. We have $f(T_0, c^{s(k,-t)}) = f(T_0, c^s) - t \leq f(T, c^s) - t \leq f(T, c^{s(k,-t)})$ for all $T \in \mathcal{T}(G)$. This implies that $T_0 \in \mathcal{T}^*(G, s(k, -t))$. If there exists $T_1 \in \mathcal{T}^*(G, s(k, -t))$ such that $k \notin E(T_1)$, then

$$f(T_1, c^{s(k,-t)}) = f(T_1, c^s) \geq f(T_0, c^s) = f(T_0, c^{s(k,-t)}) + t$$

Therefore $T_1 \notin \mathcal{T}^*(G, s(k, -t))$ and we have a contradiction. □

Now we shall define the subset Ω of D for which all neighborhood $B(c^s, \delta) = \{c \in \mathbb{R}^m : |c - c^s| < \delta\}$ of $c^s \in D$ contains points $c^{s_1}, c^{s_2} \in B(c^s, \delta)$ such that $T^{*s_1} \in \mathcal{T}^*(G, s_1)$ but T^{*s_1} is not optimal on the scenario s_2 and $T^{*s_2} \in \mathcal{T}^*(G, s_2)$ but T^{*s_2} is not optimal on the scenario s_1.

Definition 2.1 *Let* $\Omega = \bigcup_{k=1}^{m} \Omega(k)$ *where*

$$\begin{aligned}
\Omega(k) = \{c^s \in D : \exists \quad & T_1, T_2 \in \mathcal{T}^*(G, s), \quad \text{and} \quad t > 0 \quad \text{such that} \\
(c^{s(k,t)} \in D, \quad & T_1 \in \mathcal{T}^*(G, s(k, t)), \quad \text{and} \quad T_2 \notin \mathcal{T}^*(G, s(k, t))) \text{ or} \\
(c^{s(k,-t)} \in D, \quad & T_2 \in \mathcal{T}^*(G, s(k, -t)) \quad \text{and} \quad T_1 \notin \mathcal{T}^*(G, s(k, -t)))\}.
\end{aligned}$$

We remark that, by Lemma 2.1, $k \notin E(T_1)$ and $k \in E(T_2)$. For each $k \in E(G)$ we can also describe $\Omega(k)$ as follows

$$\Omega(k) = \{c^s \in D : \exists\, T_1, T_2 \in \mathcal{T}^*(G,s) \text{ and } j \in E(G) :$$

$$k \notin E(T_1),\ k \in E(T_2),\ j \in E(T_1),\ j \notin E(T_2) \text{ and } c^s_k = c^s_j\}$$

We observe that if $\Omega = \emptyset$, then for all $c^s \in D$ there exists a neighborhood $B(c^s,t)$ of c^s, such that for all $c^{s_o} \in B(c^s,t) \bigcap D$, $\mathcal{T}^*(G,s) = \mathcal{T}^*(G,s_o)$ and this implies that there exists a minimum spanning tree for all realizations of edges costs (a permanent tree for S, see Yaman et al. (2001)) .

Theorem 2.2 *For all $T \in \mathcal{T}(G)$ the functions $c^s \to f(T^{*s},c^s)$ and r_T are differentiable over $D \setminus \Omega$. Moreover for all $c^s \in \Omega$ both functions are not differentiable in c^s.*

Proof. If $c^s \in D \setminus \Omega$ there exists a neighborhood $B(c^s,t)$ of c^s such that for all $c^{s_o} \in B(c^s,t) \bigcap D$, $\mathcal{T}^*(G,s) = \mathcal{T}^*(G,s_o)$, then $c^s \to f(T^{*s},c^s)$ and r_T are differentiable over $B(c^s,t) \bigcap D$ and then over $D \setminus \Omega$.

In order to prove that for all $c^s \in \Omega$, $c^s \to f(T^{*s},c^s)$ is not differentiable on c^s we shall prove that $\frac{\partial f(T^{*s},c^s)}{\partial c_k}$ does not exist. By definition

$$\frac{\partial f(T^{*s},c^s)}{\partial c_k} =$$

$$\lim_{h \to 0} \frac{f(T^{*s(k,h)}, c^s + he_k) - f(T^{*s},c^s)}{h}$$

Let $k \in I$ such that $\Omega(k) \neq \emptyset$ and let $T_1, T_2 \in \mathcal{T}^*(G,s)$ and $j \in E(G)$ such that $k \notin E(T_1)$, $k \in E(T_2)$, $j \in E(T_1)$, $j \notin E(T_2)$ and $c^s_k = c^s_j$. Then if $h < 0$ by Lemma 2.1, $T_2 \in \mathcal{T}^*(G, s(k,h))$ and

$$\lim_{h \to 0^-} \frac{f(T_2, c^s + he_k) - f(T_2,c^s)}{h} =$$

$$\lim_{h \to 0^-} \frac{h}{h} = 1.$$

If $h > 0$, since $T_1 \in \mathcal{T}^*(G,s)$ and $k \notin E(T_1)$ then

$$\lim_{h \to 0^+} \frac{f(T^{*s(k,h)}, c^s + he_k) - f(T^{*s},c^s)}{h} =$$

$$\lim_{h \to 0^+} \frac{f(T_1, c^s + he_k) - f(T_1,c^s)}{h} = 0$$

then $c^s \to f(T^{*s},c^s)$ and r_T are not differentiable over Ω. $\square$

Lemma 2.2 *If D is convex then r_T is convex and $c^s \to f(T^{*s}, c^s)$ is a concave function.*

Proof. Let $c^{s_1}, c^{s_2} \in D$, as D is convex, for all $\lambda \in [0,1]$ if $c^{s_0} = \lambda c^{s_1} + (1-\lambda)c^{s_2}$ then $c^{s_0} \in D$ and then we have

$$r_T(c^{s_0}) = f(T, c^{s_0}) - f(T^{*s_0}, c^{s_0}) = \sum_{i\in E(T)} c_i^{s_0} - \sum_{i\in E(T^{*s_0})} c_i^{s_0} =$$

$$\sum_{i\in E(T)} (\lambda c_i^{s_1} + (1-\lambda)c_i^{s_2}) - \sum_{i\in E(T^{*s_0})} (\lambda c_i^{s_1} + (1-\lambda)c_i^{s_2}) =$$

$$\lambda\left(\sum_{i\in E(T)} c_i^{s_1} - \sum_{i\in E(T^{*s_0})} c_i^{s_1}\right) + (1-\lambda)\left(\sum_{i\in E(T)} c_i^{s_2} - \sum_{i\in E(T^{*s_0})} c_i^{s_2}\right) =$$

$$\lambda(f(T, c^{s_1}) - f(T^{*s_0}, c^{s_1})) + (1-\lambda)(f(T, c^{s_2}) - f(T^{*s_0}, c^{s_2})) \leq$$

$$\lambda\left(f(T, c^{s_1}) - f(T^{*s_1}, c^{s_1})\right) + (1-\lambda)\left(f(T, c^{s_2}) - f(T^{*s_2}, c^{s_2})\right) =$$

$$\lambda r_T(c^{s_1}) + (1-\lambda) r_T(c^{s_2})$$

hence r_T is convex.

$$f(T^{*s_o}, c^{s_o}) = \sum_{i\in E(T^{*s_0})} c_i^{s_0} = \sum_{i\in E(T^{*s_0})} (\lambda c_i^{s_1} + (1-\lambda)c_i^{s_2}) =$$

$$\sum_{i\in E(T^{*s_0})} \lambda c_i^{s_1} + \sum_{i\in E(T^{*s_0})} (1-\lambda)c_i^{s_2} \geq \lambda f(T^{*s_1}, c^{s_1}) + (1-\lambda) f(T^{*s_2}, c^{s_2})$$

and $c^s \to f(T^{*s}, c^s)$ is concave. □

Next theorem gives the location of the worst and best minimax regret scenarii for a tree when the set of scenarii is a compact and convex subset of $\mathbb{R}^n$.

Theorem 2.3 *Let T a tree of G and D a compact and convex subset of $\mathbb{R}^m$, then the data instances that correspond to the worst and best minimax regret scenarii for T are on the boundary ∂D of D and on $\partial D \bigcup \Omega$ respectively.*

Proof. Straightforward, because r_T is a piecewise linear and convex function defined in a compact and convex set. □

2.4 Interval uncertainty: 1-persistent and 0-persistent edges

In this section we consider the minimum spanning tree problem under interval uncertainty, i.e., $D = \prod_{i=1}^{m} [\underline{c_i}, \overline{c_i}]$. We provide conditions for an edge to be always or never on a minimum spanning tree (1-persistent and 0-persistent edges respectively) for all realization of data. We shall first give some definitions.

Definition 2.2 *Let*

$$B(k) = \{(c_1^s \dots c_k^s \dots c_m^s) \in D : \exists\ j \in E(G),\ j \neq k \text{ such that } c_k^s = c_j^s\}$$

$$E^-(k) = \{i \in E(G) :\ \overline{c}_i < \underline{c}_k\}$$

$$E^=(k) = \{i \in E(G) : i \neq k \text{ and } \exists s \in S \text{ such that } c_i^s = c_k^s\}$$

Definition 2.3 *Let E' a nonempty subset of $E(G)$. The subgraph of G whose vertex set is the set of ends of edges in E' and whose edge set is E' is called the subgraph of G induced by E' and is denoted by $G[E']$.*

Remark 2.1 *We can remark that $G[E^-(k)]$ is not necessarily a connected subgraph of G, thus for all $s \in S$, $G[E(T^{*s}) \cap E^-(k)]$ is a spanning acyclic subgraph of $G[E^-(k)]$ and each connected component of $G[E(T^{*s}) \cap E^-(k)]$ is a spanning tree of a connected component of $G[E^-(k)]$, thus for all $s \in S$, $G[E(T^{*s}) \cap E^-(k)]$ and $G[E^-(k)]$ have the same number of connected components and for all $s, s' \in S$, $|E(T^{*s}) \cap E^-(k)| = |E(T^{*s'}) \cap E^-(k)|$.*

We remark that if $\Omega(k) = \emptyset$ then k is 1-persistent or k is 0-persistent, if $\Omega(k) \neq \emptyset$ then by definition k is a weak edge.

Theorem 2.4 *Suppose that D is a Cartesian product of intervals. Then*

1. *If $E^-(k) = \emptyset$ then k is weak.*
2. *If $B(k) = \emptyset$ and $E^-(k) = \emptyset$ then k is 1-persistent.*
3. *If $B(k) = \emptyset$ and there exists $s \in S$ such that $k \in E(T^{*s})$ then k is 1-persistent.*
4. *If $B(k) = \emptyset$ and there exists $s \in S$ such that $k \notin E(T^{*s})$ then k is 0-persistent.*

Proof. If $E^{-}(k) = \emptyset$ then there exists $s \in S$ such that when Kruskal's algorithm sorts edges in non-decreasing order of their costs, edge k is in the first place, then edge k is in a minimum spanning tree of G for the scenario $s \in S$ and then k is a weak edge. If $B(k) = \emptyset$ and $E^{-}(k) = \emptyset$, for all $s \in S$ the edge k is in the first place, then k is in all minimum spanning tree of G for all scenario $s \in S$, so k is a 1-persistent edge.

If $B(k) = \emptyset$ and there exists $s \in S$ such that $k \in E(T^{*s})$, then adding edge k to $E(T^{*s}) \cap E^{-}(k)$ at the point it was encountered would not have introduced a cycle, then k does not have two extremities in the same connected component of the subgraph of G induced by edges in $E(T^{*s}) \cap E^{-}(k)$. By Remark 2.1 for all $s' \in S$, k does not have two extremities in the same connected component of the subgraph of G induced by edges in $E(T^{*s'}) \cap E^{-}(k)$. Since $B(k) = \emptyset$, for all $s' \in S$ Kruskal's algorithm adds edge k to $E(T^{*s'}) \cap E^{-}(k)$, and then k is a 1-persistent edge.

If $B(k) = \emptyset$ and there exists $s \in S$ such that $k \notin E(T^{*s})$, then adding edge k to $E(T^{*s}) \cap E^{-}(k)$ at the point it was encountered would have introduced a cycle, then k joins two vertices in the same connected component of the subgraph of G induced by edges of $E(T^{*s}) \cap E^{-}(k)$.Then by Remark 2.1, for all $s' \in S$, adding k to $E(T^{*s'}) \cap E^{-}(k)$ we introduce a cycle, and then k is 0-persistent. □

The following result provides a characterization of the 0-persistent edges.

Theorem 2.5 *Suppose that D is a Cartesian product of intervals. Edge k is 0-persistent if and only if k is not a cut edge of the subgraph of G induced by the edges in $E^{-}(k) \cup \{k\}$.*

Proof. If k is 0-persistent for all $s \in S$ and for all $T^{*s} \in \mathcal{T}^{*}(G, s)$, $k \notin E(T^{*s})$. This implies that for all $s \in S$ adding edge k to $E(T^{*s}) \cap E^{-}(k)$ at the point it was encountered would have introduced a cycle, so k is not a cut edge of $G[E^{-}(k) \cup \{k\}]$. Now if k is not a cut edge of $G[E^{-}(k) \cup \{k\}]$, k is contained in a cycle of $G[E^{-}(k) \cup \{k\}]$. By definition of $E^{-}(k)$ we know that for all $s \in S$ each minimum spanning acyclic subgraph of $G[E^{-}(k) \cup \{k\}]$ does not contain edge k, so for all $s \in S$, $k \notin E(T^{*s})$ thus k is 0-persistent. □

The following theorem gives a characterization of the 1-persistent edges.

Theorem 2.6 *Suppose that D is a Cartesian product of intervals. Edge k is 1-persistent if and only if k is a cut edge of the subgraph of G induced by the edges in $E^{=}(k) \cup E^{-}(k) \cup \{k\}$.*

Proof. If k is a cut edge of $G[E^{=}(k) \cup E^{-}(k) \cup \{k\}]$, then for all $s \in S$, Kruskal's algorithm must select edge k to form a spanning tree of G and then for all $s \in S$ and for all $T^{*s} \in \mathcal{T}^*(G,s)$, $k \in T^{*s}$, thus k is a 1-persistent edge. Now if k is not a cut edge of $G[E^{=}(k) \cup E^{-}(k) \cup \{k\}]$ then k is contained in a cycle of $G[E^{=}(k) \cup E^{-}(k) \cup \{k\}]$. By definition of $E^{-}(k)$ and $E^{=}(k)$ there exists a scenario $s' \in S$ such that $c_j^{s'} \leq c_k^{s'}$ for all $j \in E^{=}(k) \cup E^{-}(k)$. So there exists a minimum spanning tree $T^{*s'}$ of G which does not contain edge k and then k is not a 1-persistent edge. □

The last theorems allow us to derive algorithms to find all the 0-persistent and 1-persistent edges.

Algorithm to find the 0-persistent edges

1. For each $k \in E(G)$ obtain the set $E^{-}(k) = \{i \in E(G) : \; \overline{c}_i < \underline{c}_k\}$.
2. Construct the subgraph of G induced by edges in $E^{-}(k) \cup \{k\}$.
3. Verify if k is not a cut edge of the subgraph $G[E^{-}(k) \cup \{k\}]$.
4. If the answer is yes, k is a 0-persistent edge, otherwise k is not 0-persistent.

Algorithm to find the 1-persistent edges

1. For each $k \in E(G)$ obtain the set $E^{-}(k) = \{i \in E(G) : \; \overline{c}_i < \underline{c}_k\}$.
2. Obtain the set $E^{=}(k) = \{i \in E(G) : \exists s \in S \text{ such that } c_i^s = c_k^s\}$.
3. Construct the subgraph of G induced by edges in $E^{=}(k) \cup E^{-}(k) \cup \{k\}$.
4. Verify if k is a cut edge of the subgraph $G[E^{=}(k) \cup E^{-}(k) \cup \{k\}]$.
5. If the answer is yes, k is a 1-persistent edge, otherwise k is not 1-persistent.

A consequence of Theorems 1.1 and 1.2, is that for the minimax and minimax regret minimum spanning tree problems under interval uncertainty, we can preprocess the problem removing the 0-persistent variables and setting the 1-persistent ones equal to 1.

For each $k \in E(G)$, the problem to find the sets $E^{-}(k)$ and $E^{=}(k)$ can be solved in $O(m)$ time. Since we have n edges, in the case of interval uncertainty, the above algorithms find the 1-persistent variables and the 0 -persistent variables in $O(m^2)$ time each one of them. Since the minimax regret minimum spanning tree problem is NP-hard, the presented algorithms can be useful for pre-processing the mixed integer programming presented by Yaman et al. (2001).

Chapter 3

Two robust path problems: interval uncertainty

3.1 Introduction

In this chapter, we consider the uncertain versions of the *shortest path problem*, that consists in finding a path of minimum length connecting two specified nodes 1 and m, and the *single-source shortest path problem*, i.e., the problem of finding shortest paths from a fixed node 1 to every nodes of the graph. We consider both of them on finite directed graphs where arc lengths belong to nonnegative intervals. We allow cycles on the graphs and *degenerated arc lengths*, i.e., the lower and upper bounds of some intervals may have the same value. These problems have important applications in transportation and telecommunications, where it is not easy to estimate arc costs exactly. Other applications may be found in Montemanni et al. (2004).

The first problem that we shall consider is the uncertain shortest path problem on a finite directed graph under interval arc length uncertainties. Kouvelis and Yu (1997) have studied this problem under discrete uncertainty and proved that the minimax and minimax regret shortest path problems are NP-complete for a bounded number of scenarii. Moreover the authors proved that the problem becomes strongly NP-hard for an unbounded number of scenarii, see Table 3.1. In this chapter, we model data uncertainty by treating the arc lengths as non-negative intervals, i.e. each arc length can take any value in its interval.

Averbakh and Lebedev (2004) proved that the minimax regret problem under interval uncertainty is NP-hard even if the network is directed, acyclic and has a layered structure and showed that this problem is polynomially

THE SHORTEST PATH PROBLEMS DISCRETE UNCERTAINTY SET		
Minimax regret		
	Algorithms & heuristics	Theory and complexity
SPP U bounded	Kouvelis and Yu (1997) Aissi (2005)	NP-complete Kouvelis and Yu (1997)
SPP U unbounded	Kouvelis and Yu (1997)	strongly NP-hard Kouvelis and Yu (1997)
Minimax		
	Algorithms & heuristics	Theory and complexity
SPP U bounded	Kouvelis and Yu (1997)	NP-complete Kouvelis and Yu (1997)
SPP U unbounded	Kouvelis and Yu (1997)	strongly NP-hard Kouvelis and Yu (1997)

Table 3.1: The shortest path problems: discrete uncertainty set.

solvable in the case where the number of edges with uncertain lengths is fixed or is bounded by the logarithm of a polynomial function of the total number of edges. Independently, Zielinski (2004) showed that this problem is NP-hard and remains NP-hard even when a graph is restricted to be directed, acyclic, planar and regular of degree three. Thus in solving this problem, reducing the solution space becomes an important issue.

Karasan et al. (2001) studied the minimax regret shortest path problem with interval uncertainty. The authors considered acyclic directed graphs under nonnegative and non-degenerated interval arc length uncertainties and proposed a mixed integer programming formulation with preprocessing to solve this problem. Giving a sufficient condition for an arc to never be on a shortest path from 1 to m, they presented a polynomial procedure to determine if an arc is of such type. The preprocessing consists of removing those arcs which are never in the shortest paths. Computational results showed that preprocessing in such kinds of graphs is efficient.

Montemanni et al. (2004) provided a branch and bound algorithm to solve the minimax regret path problem under interval uncertainties but they did not implement the preprocessing proposed by Karasan et al. (2001) because in practice it can be used only for acyclic layered graphs with small

THE SHORTEST PATH PROBLEMS INTERVAL UNCERTAINTY	
Minimax regret	
Algorithms & heuristics	Theory and complexity
Branch and bound Montemanni et al. (2004) Montemanni and Gambardella (2004) Montemanni and Gambardella (2005a) Karasan et al. (2001) Kasperski and Zielinski (2004) Kasperski and Zielinski (2006) approximation algorithm	NP-hard Zielinski (2004) polynomial cases Averbakh and Lebedev (2004)
Minimax	
Algorithms & heuristics	Theory and complexity
Karasan et al. (2001)	polynomial Karasan et al. (2001)

Table 3.2: The shortest path problems: interval uncertainty set.

width. Montemanni and Gambardella (2004) presented an exact algorithm to solve the minimax regret path problem under interval uncertainty. Kasperski and Zielinski (2006) examined the minimax regret shortest path problem in series-parallel multi-digraphs with interval uncertainty and showed that this problem is NP-hard. The authors presented a pseudo-polynomial algorithm to solve the problem, see Table 3.2.

In this chapter we study the problem of detecting nodes and arcs that are always or never on a shortest path from 1 to m. We consider this problem on a larger class of directed graphs i.e., we allow cycles on the graph and degenerated arc lengths. In particular, we extend the results given in Karasan et al. (2001). We give sufficient conditions for a node to be never on a shortest path from 1 to m (0-persistent nodes) and sufficient conditions for an arc to be always or never on a shortest path from 1 to m (1-persistent and 0-persistent arcs, respectively). These conditions allow us to give polynomial time algorithms to find 1-persistent arcs and 0-persistent nodes and arcs. We propose a preprocessing procedure consisting of removing the 0-persistent nodes and prove by means of computational experiments that our procedure vastly reduces the overall time to compute a solution of the minimax regret shortest path problem with interval uncertainty.

The second problem that we consider is the minimax regret version of the single-source shortest path problem. We propose a mixed integer programming formulation for this problem. We give sufficient conditions for an arc (k, r) to be always (for all realization of data) on all shortest paths from 1 to r (T 1-persistent arcs) and a characterization of arcs that are never on a shortest paths from 1 to another node of G (T 0-persistent arcs).

We present polynomial time algorithms to find T 1-persistent and T 0-persistent arcs. Finally we present numerical results that show that the preprocessing consisting of removing the T 0-persistent arcs, greatly decreases the time needed to compute a minimax regret solution. These last results have been published in Salazar-Neumann (2007b).

3.2 Notation and definitions

As in Bondy and Murty (1976), we introduce here some definitions that we shall prove useful throughout the rest of the chapter. Let G be a finite directed graph. We denote by $V(G)$ and $A(G)$ the set of nodes and arcs of G respectively. We suppose that $|V(G)| = m$ and $|A(G)| = n$. We denote by (i, j) the arc from node i to node j. If A' is a nonempty subset of $A(G)$, the subgraph of G with edge set $A(G) \setminus A'$, denoted as $G - A'$ represents the subgraph obtained from G by deleting arcs in A'. In a similar way, the graph obtained from G by adding a set of arcs A' is denoted by $G + A'$. If $A' = \{(i, j)\}$ we write $G - (i, j)$ and $G + (i, j)$ instead of $G - \{(i, j)\}$ and $G + \{(i, j)\}$. Similarly, if G' is a subgraph of G and $(i, j) \in A(G')$, we denote by $G' - (i, j)$ the subgraph obtained from G' by deleting arc (i, j). If H_1 and H_2 are two subgraphs of G, the union $H_1 \cup H_2$ is the subgraph with vertex set $V(H_1) \cup V(H_2)$ and arc set $A(H_1) \cup A(H_2)$.

Given a digraph G and nonnegative lengths l_{ij} associated with each arc $(i, j) \in A(G)$, the deterministic shortest path problem consists of finding a shortest directed path, denoted as $(1, m)$-*path*, connecting two specified nodes, the origin node 1 and the destination node m in G. The length of a path is the sum of the lengths of its arcs. An efficient $O(|V(G)|^2)$ algorithm to solve this problem was given by Dijkstra. It finds not only a shortest $(1, m)$-path, but shortest paths from 1 to all nodes of G.

In order to construct the uncertain version of this problem, we shall introduce the following concepts and notations. Let S be the set of scenarii for the lengths of arcs of G and let D be the uncertainty set. We denote by l_{ij}^s the nonnegative length of arc (i, j) in the scenario s and we shall assume that D is a Cartesian product of intervals, that is to say, each l_{ij}^s can take an arbitrary value in the interval $[\underline{l}_{ij}, \bar{l}_{ij}]$. We denote by $\underline{s}$ the scenario for which for all $(i, j) \in A(G)$, $l_{ij}^{\underline{s}} = \underline{l}_{ij}$ and by $\overline{s}$ the scenario for which for all $(i, j) \in A(G)$, $l_{ij}^{\overline{s}} = \bar{l}_{ij}$.

We denote by $P_1(G)$ the set of all the *1-paths* of G connecting the node 1 with another node of G, by $P_1^*(G, s)$ the set of all shortest 1-paths of G under the scenario s, by $P_{1k}(G)$ the set of all $(1, k)-$paths of G and by $P_{1k}^*(G, s)$, the set of all the shortest $(1, k)-$paths of G under the scenario s. For a path $p \in P_{1k}(G)$ we denote by $A(p)$ the set of arcs of p. To evaluate a path $p \in P_{1k}(G)$ under the scenario s we use a function

$$f : P_{1k}(G) \times D \to \mathbb{R} : (p, l^s) \to f(p, l^s),$$

defined as

$$f(p, l^s) = \sum_{(i,j)\in A(G)} l_{ij}^s x_{ij},$$

where

$$x_{ij} = \begin{cases} 1 \text{ if } & (i, j) \in A(p), \\ 0 \text{ otherwise.} \end{cases}$$

For the sake of simplicity, we shall denote by l_p^s the length $\sum_{(i,j)\in A(G)} l_{ij}^s x_{ij}$ of path p in the scenario s. In the case where we want to find a shortest directed path connecting two specified nodes 1 and m, we define

$$f(p^{*s}, l^s) = l_{p^{*s}}^s = \min_{p\in P_{1m}(G)} l_p^s.$$

In the context of the uncertain version of shortest path problem the following four definitions prove useful, see Karasan et al. (2001).

Definition 3.1 *A path is said to be a* weak path *if it is a shortest path from* 1 *to* m *for at least one realization of arc lengths.*

Definition 3.2 *An arc is a* weak arc *if it lies on some weak path.*

Definition 3.3 *An arc is a* strong arc *if it lies on at least a shortest path from 1 to* m *for all scenarii.*

Definition 3.4 *An arc is a* 0-persistent arc *if it never lies on a shortest path from 1 to* m *for all scenarii.*

Karasan et al. (2001) call the 0-persistent arcs "non-weak arcs". Now, we introduce the following definitions.

Definition 3.5 *An arc is a* 1-persistent arc *if it lies on all shortest path from 1 to m for all scenarii.*

Definition 3.6 *A node is a* 0-persistent node *if it never lies on a shortest path from 1 to m for all scenarii.*

We call a feasible solution of the deterministic single-source shortest path problem a *spanning tree*, an optimal solution a *shortest path spanning tree* of G and a shortest path from node 1 to another node of G a *shortest 1-path.*

We denote by $\mathcal{T}^1(G)$ the set of all the spanning trees of G, by $\mathcal{T}_1^*(G,s)$ the set of all shortest path spanning trees of G under the scenario s and by T_1^{*s} a generic element of $\mathcal{T}_1^*(G,s)$. For a spanning tree $T \in \mathcal{T}^1(G)$ we denote by $A(T)$ the set of arcs of T. Given a $T \in \mathcal{T}^1(G)$, we denote by $p_{1k}(T)$ the $(1,k)$-path such that $A(p_{1k}(T)) \subset A(T)$. To evaluate a spanning tree $T \in \mathcal{T}^1(G)$ under the scenario s we use the function

$$f : \mathcal{T}^1(G) \times D \to \mathbb{R} : (T, l^s) \to f(T, l^s),$$

defined as

$$f(T, l^s) = \sum_{k \in V(G)} l^s_{p_{1k}(T)} = \sum_{k \in V(G)} \sum_{(i,j) \in A(G)} l^s_{ij} x^k_{ij},$$

where

$$x^k_{ij} = \begin{cases} 1 \text{ if } \ (i,j) \in A(p_{1k}(T)), \\ 0 \text{ otherwise.} \end{cases}$$

For the sake of simplicity we shall denote by l^s_T the value $f(T, l^s)$ of a spanning tree T in the scenario s, and we define

$$l^s_{T_1^{*s}} = f(T_1^{*s}, l^s) = \min_{T \in \mathcal{T}^1(G)} f(T, l^s) = \min_{T \in \mathcal{T}^1(G)} l^s_T.$$

This model coincides with the model of uncertain combinatorial optimization problems UCOP considered in Chapter 1, in which to each data instance $c^s \in D$ corresponds a deterministic combinatorial optimization problem:

$$\text{minimize} \quad \sum_{i \in I} c_i^s x_i,$$

$$\text{subject to} \quad x \in X \subseteq \{0,1\}^{|I|}.$$

Such a correspondence is obtained as follows

$$I = A(G) \times V(G) \quad \text{and} \quad D = \prod_{(i,j) \in A(G)} [\underline{l}_{ij}, \bar{l}_{ij}].$$

Definition 3.7 *A spanning tree T is said to be a weak spanning tree if it is a shortest path spanning tree for at least one realization of arc lengths.*

Definition 3.8 *An arc is a T-weak arc if it lies on some weak spanning tree.*

Definition 3.9 *An arc (i,j) is a T 1-persistent arc if it lies on all shortest path spanning trees for all scenarii.*

Definition 3.10 *An arc (i,j) is a T 0-persistent arc if it never lies on a shortest path spanning tree for all scenarii.*

3.3 The uncertain shortest path problem: 1-persistent arcs and 0-persistent arcs and nodes

We shall consider the minimax regret shortest path problem with interval uncertainty, defined as follows. Given a finite digraph G, assume that the lengths of the arcs are uncertain and D is a Cartesian product of nonnegative intervals, $D = \prod_{i \in A(G)} [\underline{l}_{ij}, \bar{l}_{ij}]$. The minimax regret shortest path problem consists in finding a path $\tilde{p} \in P_{1m}(G)$ such that

$$R_{max}(\tilde{p}) = \min_{p \in P_{1m}(G)} R_{max}(p) = \min_{p \in P_{1m}(G)} \max_{s \in S} \left(l_p^s - \min_{q \in P_{1m}(G)} l_q^s \right).$$

This definition coincides with the definition of the minimax regret problem considered in Chapter 1.

Karasan et al. (2001) showed that the worst scenario for a path $p \in P_{1m}(G)$ is the scenario in which the lengths of all arcs on p are at upper bound and the lengths of all other arcs at their lower bound. So if we define the vector $y \in \{0,1\}^n$ as

$$y_{ij} = \begin{cases} 1 \text{ if } \ (i,j) \in A(p), \\ 0 \text{ otherwise.} \end{cases}$$

The length of arc (i,j) under the worst scenario for p is defined as $l_{ij} = \underline{l}_{ij} + (\bar{l}_{ij} - \underline{l}_{ij})y_{ij}$. In the same paper authors give the following mixed integer programming formulation to solve such problem.

$$\min \sum_{(i,j)\in A(G)} \bar{l}_{ij} y_{ij} - x_m$$

$$\text{s.t.} \quad x_j \le x_i + \underline{l}_{ij} + (\bar{l}_{ij} - \underline{l}_{ij})y_{ij}, \quad \forall (i,j) \in A(G)$$

$$-\sum_{i\in\Gamma^-(j)} y_{ij} + \sum_{k\in\Gamma^+(j)} y_{jk} = b_j, \quad j = 1,2\ldots,m,$$

$$x_1 = 0,$$

$$y_{ij} \in \{0,1\}, \quad \forall (i,j) \in A(G),$$

$$x_j \ge 0, \quad j = 1,2\ldots,m,$$

where

$$\Gamma^-(j) = \{i \in V(G) : (i,j) \in A(G)\},$$

$$\Gamma^+(j) = \{k \in V(G) : (j,k) \in A(G)\}$$

and

$$b_j = \begin{cases} 1 & \text{if} \quad j = 1 \\ 0 & \text{if} \quad j \ne 1, m \\ -1 & \text{if} \quad j = m \end{cases}$$

The first constraint ensures that x_j is the length of the shortest path from node 1 to node j in the graph under the scenario defined by y, the second one that y defines a path from node 1 to node m and the third avoids an unbounded solution.

In this section we shall suppose that $G(V,A)$ is a finite directed graph, with origin node 1 and destination node m and that each node $j \in V(G)$ is reachable from node 1.

We shall present a preprocessing procedure that reduces the solution time taken to solve this formulation. In order to construct our preprocessing algorithm we present a sufficient condition easy to test for an arc to be 1-persistent, but we give the following characterization of the 1-persistent arcs. The proof of it, being straightforward, is omitted.

Proposition 3.1 *Consider an arc (k,r) such that on the graph $G-(k,r)$ node m is reachable from node 1. For each $s \in S$ let $p^{*s} \in P^*_{1m}(G,s)$. Thus arc (k,r) is a 1-persistent arc of G if and only if for all $q \in P_{1m}(G-(k,r))$ and for all $s \in S$, $l^s_q > l^s_{p^{*s}}$.*

Theorem 3.1 *Let p be a shortest path from 1 to m under scenario $\overline{s}$. Let $(k,r) \in A(p)$, and let s' be the scenario such that, $l^{s'}_{ij} = \bar{l}_{ij}$ if $(i,j) \in A(p)$, and $l^{s'}_{ij} = \underline{l}_{ij}$ if $(i,j) \notin A(p)$. If in $G-(k,r)$, node m is reachable from node 1, let $q \in P^*_{1m}(G-(k,r),s')$ be a shortest path from 1 to m under the scenario s'. If $l^{s'}_q > l^{s'}_p$ then the arc (k,r) is 1-persistent.*

Proof. Let p be a shortest path from 1 to m under the scenario $\overline{s}$ and let $(k,r) \in A(p)$. Consider the scenario s' such that $l^{s'}_{ij} = \bar{l}_{ij}$ if $(i,j) \in A(p)$, and $l^{s'}_{ij} = \underline{l}_{ij}$ if $(i,j) \notin A(p)$. Let q a shortest path of $G-(k,r)$ from 1 to m under the scenario s'. If $l^{s'}_q > l^{s'}_p$ then for all $p' \in P_{1m}(G-(k,r))$ $l^{s'}_{p'} \geq l^{s'}_q > l^{s'}_p$ and then for all $p' \in P_{1m}(G-(k,r))$

$$\sum_{(i,j)\in A(p')\setminus A(p)} \underline{l}_{ij} = \sum_{(i,j)\in A(p')\setminus A(p)} l^{s'}_{ij} > \sum_{(i,j)\in A(p)\setminus A(p')} l^{s'}_{ij} = \sum_{(i,j)\in A(p)\setminus A(p')} \bar{l}_{ij}.$$

Then, for all $s \in S$

$$\sum_{(i,j)\in A(p')\setminus A(p)} l^{s}_{ij} \geq \sum_{(i,j)\in A(p')\setminus A(p)} \underline{l}_{ij} > \sum_{(i,j)\in A(p)\setminus A(p')} \bar{l}_{ij} \geq \sum_{(i,j)\in A(p)\setminus A(p')} l^{s}_{ij}$$

which implies that for all $s \in S$ and for all $p' \in P_{1m}(G-(k,r))$, we have $l^s_{p'} > l^s_p$ and for $p^{*s} \in P^*_{1m}(G,s)$, $l^s_{p'} > l^s_p \geq l^s_{p^{*s}}$. Hence, (k,r) is a 1-persistent arc. □

Theorem 3.1 allows us to derive a polynomial time algorithm to find 1-persistent arcs.

Algorithm to find 1-persistent arcs

1. Apply Dijkstra algorithm to G under scenario $\overline{s}$ to obtain p, a shortest path from 1 to m.

2. Choose an arc $(k,r) \in A(p)$.

3. Construct the scenario s' for which $l^{s'}_{ij} = \bar{l}_{ij}$ if $(i,j) \in A(p)$, and $l^{s'}_{ij} = \underline{l}_{ij}$ if $(i,j) \notin A(p)$.

4. Apply Dijkstra to $G-(k,r)$ under scenario s'. If there exists a shortest path q from 1 to m, go to step 5. Otherwise (k,r) is 1-persistent.

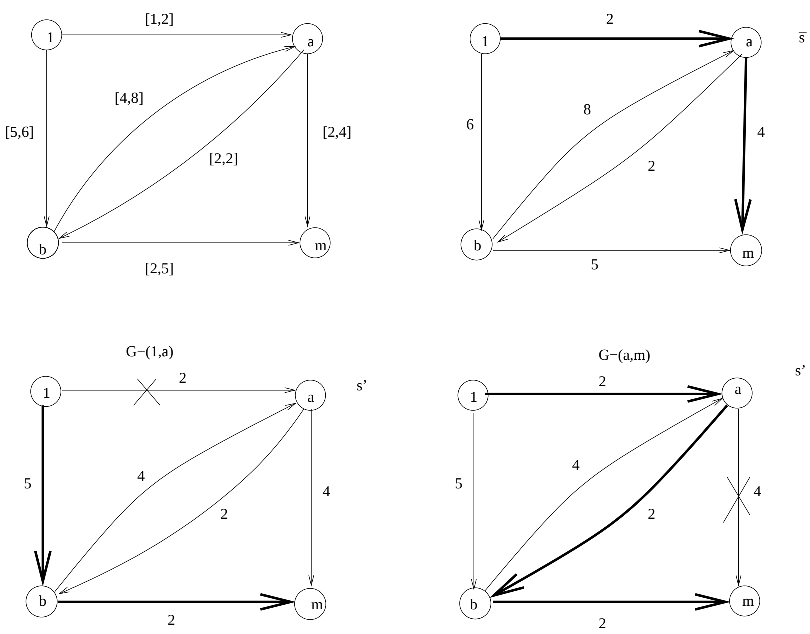

Figure 3.1: 1-persistent arcs algorithm

5. Compute $l_q^{s'}$ and $l_p^{s'}$.

6. If $l_q^{s'} > l_p^{s'}$ then arc (k, r) is a 1-persistent arc.

7. Go to step 2.

Example 3.1 *To clarify the above procedure, we apply it on the first graph given in Figure 3.1. We set the lengths of all arcs to their upper bounds and we find a shortest path p from node 1 to node m, we represent such path with bold lines. We construct the scenario s' setting the arcs $(1, a)$ and (a, m) to their upper bounds and the lengths of the remaining arcs to their lower bounds. We delete arc $(1, a) \in A(p)$ and we apply Dijkstra to $G - (1, a)$ under the scenario s'. Since there exists a shortest path q from 1 to m, we compute $l_q^{s'} = 7$ and $l_p^{s'} = 6$. Since $l_q^{s'} > l_p^{s'}$ thus arc $(1, a)$ is a 1-persistent arc. Then we take the second arc (a, m) of p, we delete arc $(a, m) \in A(p)$ and we apply Dijkstra to $G - (a, m)$ under the scenario s'. As there exists a shortest path q from 1 to m, we compute $l_q^{s'} = 6$ and $l_p^{s'} = 6$. Since $l_q^{s'} = l_p^{s'}$ we cannot conclude that arc (a, m) is 1-persistent.*

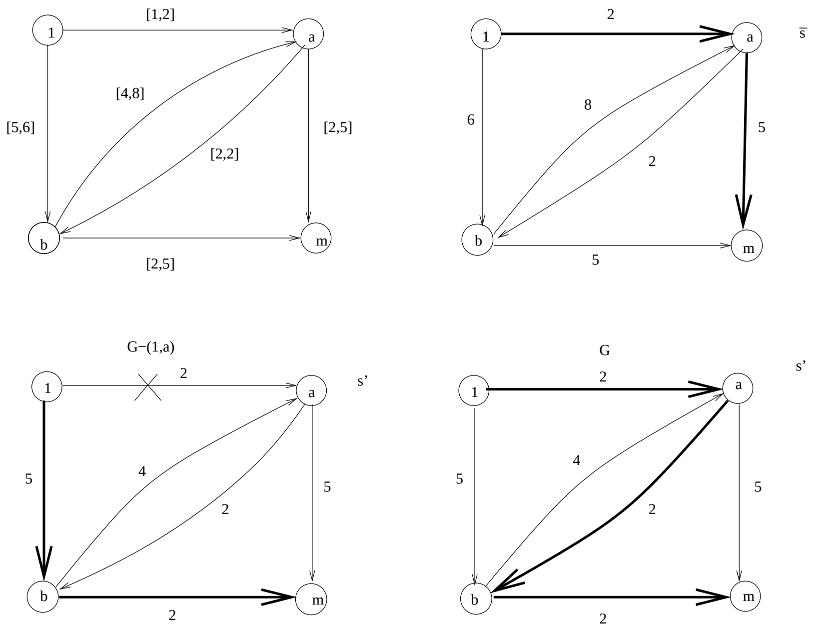

Figure 3.2: Non necessary condition to 1-persistent arcs

Example 3.2 *In order to show that the condition of Theorem 3.1 is a sufficient but not a necessary condition, we will change to 5 the upper bound of arc (a, m), see Figure 3.2. We set the lengths of all arcs to their upper bounds and we find a shortest path p from node 1 to node m, we represent this path with bold lines. We construct scenario s' by setting the arcs $(1, a)$ and (a, m) to their upper bounds and the lengths of the remaining arcs to their lower bounds. We delete arc $(1, a) \in A(p)$ and we apply Dijkstra to $G - (1, a)$ under the scenario s'. As there exists a shortest path q from 1 to m, we compute $l_q^{s'} = 7$ and $l_p^{s'} = 7$. Then $l_q^{s'} = l_p^{s'}$ however arc $(1, a)$ is a 1-persistent arc.*

From the last theorem, if we replace the arc $(k, r) \in A(p)$ for a node $k \in V(p)$ we can obtain a very similar sufficient condition for a node to be 1-persistent. The proof of it, being analogous, is omitted.

Theorem 3.2 *Let p be a shortest path from 1 to m under scenario $\overline{s}$. Let $k \in V(p)$, and let s' be the scenario such that, $l_{ij}^{s'} = \overline{l}_{ij}$ if $(i, j) \in A(p)$, and $l_{ij}^{s'} = \underline{l}_{ij}$ if $(i, j) \notin A(p)$. If in $G - k$, node m is reachable from node 1, let $q \in P_{1m}^{*}(G - k, s')$ be a shortest path from 1 to m under the scenario s'. If $l_q^{s'} > l_p^{s'}$ then the node k is 1-persistent.*

The following theorem gives a sufficient condition for a node to be 0-persistent.

Theorem 3.3 *Let q be a shortest path from 1 to m under scenario $\overline{s}$. Let $k \in V(G) \setminus V(q)$. Consider the shortest paths p_1 and p_2 from 1 to k and from k to m respectively under scenario $\underline{s}$. If $l^{\underline{s}}_{p_1} + l^{\underline{s}}_{p_2} > l^{\overline{s}}_{q}$, then node k is 0-persistent.*

Proof. Consider the paths p_1, p_2 and q described above. Since all $p' \in P^*_{1m}(G, \underline{s})$ is such that $l^{\underline{s}}_{p'} \leq l^{\overline{s}}_{q}$, if $l^{\underline{s}}_{p_1} + l^{\underline{s}}_{p_2} > l^{\overline{s}}_{q}$, then $p_1 \cup p_2$ is not a shortest path from 1 to m under the scenario $\underline{s}$. Since p_1 and p_2 are shortest paths from 1 to k and from k to m respectively, under scenario $\underline{s}$, any path p from 1 to m that contains node k is such that $l^{\underline{s}}_{p} \geq l^{\underline{s}}_{p_1} + l^{\underline{s}}_{p_2}$. Then for all $s \in S$ and all $q^{*s} \in P^*_{1m}(G, s)$

$$l^{s}_{p} \geq l^{\underline{s}}_{p} \geq l^{\underline{s}}_{p_1} + l^{\underline{s}}_{p_2} > l^{\overline{s}}_{q} \geq l^{s}_{q^{*s}}.$$

And this implies that for all scenario $s \in S$ no one path from 1 to m that contains node k is a shortest path from 1 to m under s, and then k is a 0-persistent node. □

Algorithm to find 0-persistent nodes

1. Apply Dijkstra algorithm to G under scenario $\overline{s}$ to obtain shortest paths from 1 to each node of G. In particular, a shortest path q, from 1 to m.

2. Choose a node $k \in V(G) \setminus V(q)$.

3. Consider the shortest path p_1, from 1 to k under scenario $\underline{s}$.

4. Apply Dijkstra algorithm to G under scenario $\underline{s}$ to obtain a shortest path p_2, from k to m.

5. Compute $l^{\underline{s}}_{p_1}$, $l^{\underline{s}}_{p_2}$ and $l^{\overline{s}}_{q}$.

6. If $l^{\underline{s}}_{p_1} + l^{\underline{s}}_{p_2} > l^{\overline{s}}_{q}$. then node k is 0-persistent.

7. Go to step 2.

In the following section, we shall give a sufficient condition for an arc to be 0-persistent and a polynomial algorithm to detect some of such arcs.

3.4 The uncertain single-source shortest path problem: T 1-persistent and T 0-persistent arcs

In this section we shall consider the minimax regret single source shortest path problem with interval uncertainty. In order to give a formulation to this problem we shall first consider the following formulation for the deterministic single source shortest path problem.

$$\min \sum_{r\in\{2\dots m\}} \sum_{(i,j)\in A(G)} l_{ij} y_{ij}^r$$

$$\text{s.t.} \quad -\sum_{i\in\Gamma^-(j)} y_{ij}^r + \sum_{k\in\Gamma^+(j)} y_{jk}^r = b_j^r \quad j = 1\dots,m \quad r = 2\dots,m$$

$$y_{ij}^r \in \{0,1\} \quad \forall (i,j) \in A(G) \quad r = 2\dots,m$$

where $\Gamma^-(j) = \{i \in V(G) : (i,j) \in A(G)\}$, $\Gamma^+(j) = \{k \in V(G) : (j,k) \in A(G)\}$ and for all $j = 1\dots,m$ and $r = 2\dots,m$

$$b_j^r = \begin{cases} 1 & \text{if} \quad j = 1 \\ 0 & \text{if} \quad j \neq 1, r \\ -1 & \text{if} \quad j = r. \end{cases}$$

Hence, we propose the following mixed integer programming formulation to solve the minimax regret single source shortest path problem with interval uncertainty.

$$\min \sum_{r\in\{2\dots m\}} \Big(\sum_{(i,j)\in A(G)} \bar{l}_{ij} y_{ij}^r - x_r \Big)$$

$$\text{s.t.} \quad x_j \le x_i + \underline{l}_{ij} + (\bar{l}_{ij} - \underline{l}_{ij}) y_{ij}, \quad \forall (i,j) \in A(G)$$

$$-\sum_{i\in\Gamma^-(j)} y_{ij}^r + \sum_{k\in\Gamma^+(j)} y_{jk}^r = b_j^r, \quad j = 1, 2\dots,m, \quad r = 2\dots,m,$$

$$y_{ij}^r \le y_{ij}, \quad \forall (i,j) \in A(G), \quad r = 2\dots,m,$$

$$\sum_{(i,j)\in A(G)} y_{ij} = n - 1,$$

$$x_1 = 0,$$

$$y_{ij}^r \in \{0,1\}, \quad \forall (i,j) \in A(G), \quad r = 2\dots,m,$$

$$y_{ij} \in \{0,1\}, \quad \forall (i,j) \in A(G),$$

$$x_j \geq 0, \quad j = 1, 2 \ldots, m,$$

where $\Gamma^-(j)$, $\Gamma^+(j)$ and b_j^r are as in the last formulation.

The first constraint ensures that x_j is the length of the shortest path from node 1 to node j in the graph under the scenario defined by y. The second constraint ensures that y^r defines a path from node 1 to node r, the third and fourth constraints ensure that y defines a tree. Finally, the fifth constraint avoids an unbounded solution.

We shall suppose that $G(V, A)$ is a finite directed graph, with origin node 1 and each node $j \in V(G)$ is reachable from node 1. The following proposition is a direct consequence of Theorem 3.1 and gives a sufficient condition easy to test for an arc to be T 1-persistent.

Proposition 3.2 *Let T be a shortest path spanning tree under the scenario $\overline{s}$, and let $(k, r) \in A(T)$. Let $p \in P_{1r}(G)$ such that $A(p) \subset A(T)$. Let s' be the scenario for which $l_{ij}^{s'} = \overline{l}_{ij}$ if $(i, j) \in A(p)$, and $l_{ij}^{s'} = \underline{l}_{ij}$ if $(i, j) \notin A(p)$. If on the graph $G - (k, r)$ node r is reachable from node 1, let $q \in P_{1r}^*(G - (k, r), s')$ be a shortest path from 1 to r under the scenario s'. If $l_q^{s'} > l_p^{s'}$ then arc (k, r) is a T 1-persistent arc.*

The last result allows us to derive a polynomial time algorithm to find T 1-persistent arcs.

Algorithm to find T 1-persistent arcs

1. Apply Dijkstra algorithm to G under scenario $\overline{s}$ to obtain T.
2. Choose an arc $(k, r) \in A(T)$.
3. We consider the path $p \in P_{1r}(G)$ such that $A(p) \subset A(T)$.
4. Construct the scenario s' for which $l_{ij}^{s'} = \overline{l}_{ij}$ if $(i, j) \in A(p)$, and $l_{ij}^{s'} = \underline{l}_{ij}$ if $(i, j) \notin A(p)$.
5. Apply Dijkstra to $G - (k, r)$ under scenario s'. If there exists a shortest path q from 1 to r, go to step 6. Otherwise (k, r) is T 1-persistent.
6. Compute $l_q^{s'}$ and $l_p^{s'}$.
7. If $l_q^{s'} > l_p^{s'}$ then arc (k, r) is a T 1-persistent arc.
8. Go to step 2.

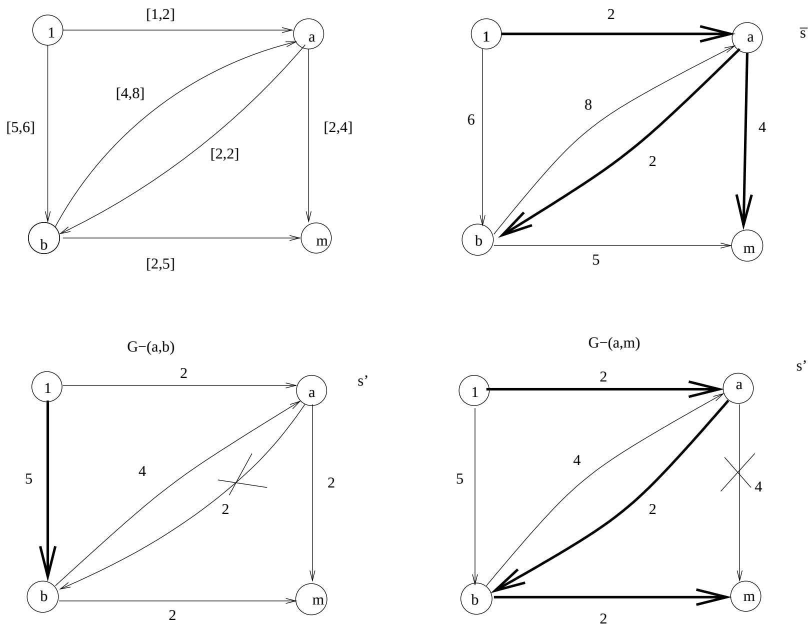

Figure 3.3: T 1-persistent arcs algorithm

Example 3.3 *To clarify the above procedure, we apply it on the first graph given in Figure 3.3. We set the lengths of all arcs to their upper bounds and we find a shortest path spanning tree. We represent this tree T with bold lines. We choose arc (a, b) and we construct scenario s' setting the arcs $(1, a)$, and (a, b) to their upper bounds and the lengths of the remaining arcs to their lower bounds. We delete arc $(a, b) \in A(p)$ and we apply Dijkstra to $G-(a, b)$ under the scenario s'. Since there exists a shortest path q from 1 to b, we compute $l_q^{s'} = 5$ and $l_p^{s'} = 4$. As $l_q^{s'} > l_p^{s'}$ then arc (a, b) is a T 1-persistent arc. Then we take a second arc (a, m) of T, we construct the scenario s' setting the arcs $(1, a)$, and (a, m) to their upper bounds and the lengths of the remaining arcs to their lower bounds. We delete arc $(a, m) \in A(T)$ and we apply Dijkstra to $G-(a, m)$ under the scenario s'. As there exists a shortest path q from 1 to m, we compute $l_q^{s'} = 6$ and $l_p^{s'} = 6$. As $l_q^{s'} = l_p^{s'}$ then we cannot conclude that arc (a, m) is T 1-persistent. Similarly, we continue with arc $(1, a)$.*

The following proposition gives a characterization of the T 0-persistent arcs.

Proposition 3.3 *For each $s \in S$, let $p^{*s} \in P^*_{1k}(G, s)$. The arc (k, r) is a T 0-persistent arc of G if and only if for each $s \in S$ all $q \in P^*_{1r}(G, s)$, are such* *that* $l_q^s < l_{p^{*s}}^s + \underline{l}_{kr}$.

Proof. Let (k,r) be a T 0-persistent arc of G, then for all $p^{*s} \in P_{1k}^*(G,s)$, $p^{*s} + (k,r)$ is never a shortest path from 1 to r, thus for all $s \in S$, if $q \in P_{1r}^*(G,s)$, we have $l_q^s < l_{p^{*s}}^s + \underline{l}_{kr}$. Now, if for all $s \in S$, all $q \in P_{1r}^*(G,s)$ is such that $l_q^s < l_{p^{*s}}^s + \underline{l}_{kr}$, then for all $p \in P_{1k}(G)$ $l_q^s < l_{p^{*s}}^s + \underline{l}_{kr} \leq l_p^s + l_{kr}^s$, then each path from 1 to r that uses arc (k,r) is never a shortest path from 1 to r for the scenario $s \in S$, and then arc (k,r) is T 0-persistent. □

Karasan et al. (2001) showed that the problem of deciding whether an arc is never on shortest path is NP-complete, the proof consists in modifying the proof of NP-completeness given by Chanas and Zielinski (2003) in the case of the longest path problem. The following theorem gives a sufficient condition easy to test for an arc to be a T 0-persistent.

Theorem 3.4 *Let* T *be a shortest path spanning tree under the scenario* $\underline{s}$ *and let* $(k,r) \in A(G) \setminus A(T)$*. Let* $p \in P_{1k}(G)$ *such that* $A(p) \subset A(T)$*. If for* $q \in P_{1r}^*(G,\overline{s})$ $l_q^{\overline{s}} < l_p^{\underline{s}} + \underline{l}_{kr}$*, then* (k,r) *is a T 0-persistent arc.*

Proof. Let T be a shortest path spanning tree under the scenario $\underline{s}$ and let $p \in P_{1k}(G)$ such that $A(p) \subset A(T)$, then p is a shortest path from 1 to k under the scenario $\underline{s}$. If for $(k,r) \in A(G) \setminus A(T)$, and $q \in P_{1r}^*(G,\overline{s})$, $l_q^{\overline{s}} < l_p^{\underline{s}} + \underline{l}_{kr}$ then for all scenarii s, if $q^{*s} \in P_{1r}^*(G,s)$ and $p^{*s} \in P_{1k}^*(G,s)$,

$$l_{q^{*s}}^s \leq l_q^{\overline{s}} < l_p^{\underline{s}} + \underline{l}_{kr} \leq l_{p^{*s}}^s + \underline{l}_{kr},$$

and Proposition 3.3 implies that (k,r) is a T 0-persistent arc. □

Since all the T 0-persistent arcs are 0-persistent, the last results allow us to derive polynomial time algorithms to find T 0-persistent and 0-persistent arcs.

Algorithm to find T 0-persistent and 0-persistent arcs.

1. Delete all arcs of the form $(j,1)$.
2. Apply Dijkstra algorithm to G under scenario $\underline{s}$ to obtain T.
3. Choose an arc $(k,r) \in A(G) \setminus A(T)$.
4. Consider the path $p \in P_{1k}(G)$, such that $A(p) \subset A(T)$.
5. Apply Dijkstra to G under scenario $\overline{s}$ to obtain q a shortest path from 1 to r.

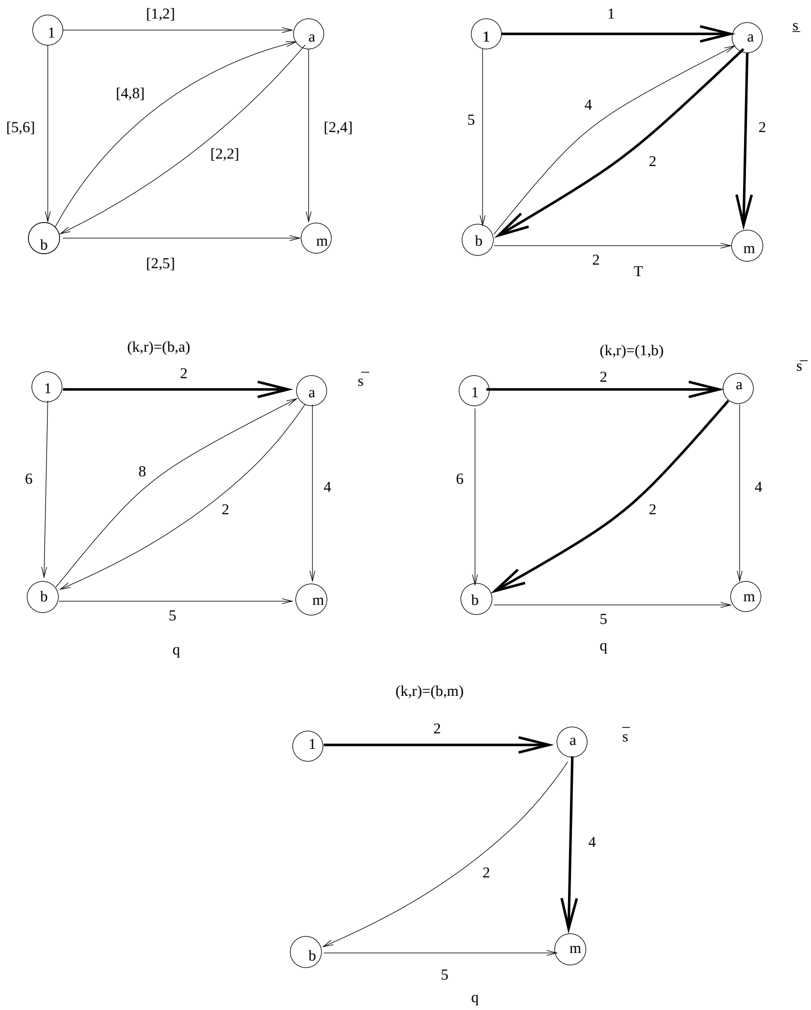

Figure 3.4: T 0-persistent and 0-persistent arcs algorithm

6. Compute $l_q^{\overline{s}}$.
7. If $l_q^{\overline{s}} < l_p^{\underline{s}} + \underline{l}_{kr}$ then arc (k,r) is T 0-persistent and then 0-persistent. Otherwise, go to step 3.
8. Delete arc (k,r) and go to step 2.

Example 3.4 *To clarify the above procedure, we apply it on the first graph given in Figure 3.4. We set the lengths of all arcs to their lower bounds and we find a shortest path spanning tree from node 1 to all nodes of G. We represent this tree T with bold lines. We choose arc $(b,a) \notin A(T)$ and we consider a path $p \in P_{1b}(G)$ such that $A(p) \subset A(T)$. We apply Dijkstra to G*

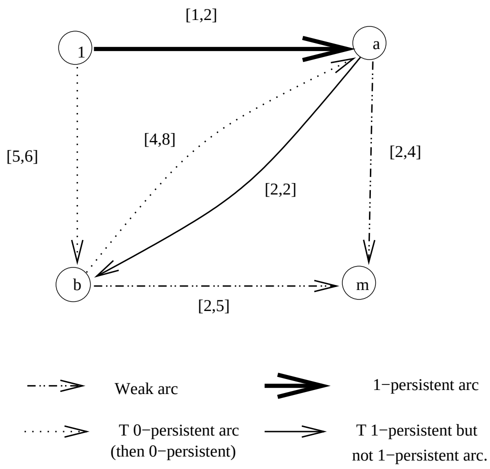

Figure 3.5: Classification of arcs

under scenario $\overline{s}$ to obtain a shortest path q from 1 to a, we compute $l_q^{\overline{s}} = 2$, $l_p^{\underline{s}} = 3$ and $\underline{l}_{ba} = 4$. Since $l_q^{\overline{s}} < l_p^{\underline{s}} + \underline{l}_{ba}$, (b,a) is T 0-persistent and thus 0-persistent. We delete arc (b,a) and we take a second arc $(1,b) \notin A(T)$. We consider a path $p \in P_{11}(G)$ such that $A(p) \subset A(T)$ and a shortest path q from 1 to b under the scenario $\overline{s}$. We compute $l_q^{\overline{s}} = 4$, $l_p^{\underline{s}} = 0$ and $\underline{l}_{1b} = 5$. Since $l_q^{\overline{s}} < l_p^{\underline{s}} + \underline{l}_{1b}$, arc $(1,b)$ is T 0-persistent and we delete it. Finally we choose arc $(b,m) \notin A(T)$ and we consider a path $p \in P_{1b}(G)$ such that $A(p) \subset A(T)$. We consider a shortest path q from 1 to m under the scenario $\overline{s}$. We compute $l_q^{\overline{s}} = 6$, $l_p^{\underline{s}} = 3$ and $\underline{l}_{bm} = 2$. Since $l_q^{\overline{s}} \geq l_p^{\underline{s}} + \underline{l}_{ba}$ we cannot conclude that arc (b,m) is T 0-persistent. The classification of arcs of this graph is given in Figure 3.5.

3.5 Numerical results

In order to reduce the computing time used to solve the minimax regret shortest path and the minimax regret single source shortest path problems with interval uncertainty we propose two different preprocessing procedures that consist of deleting the 0-persistent nodes for the first problem and the T 0-persistent arcs for the second problem.

For the minimax regret shortest path, we use the formulation given by Karasan et al. (2001) coded in Mosel and XPRESS to solve the mixed integer programming. We first solve the problem with all the nodes in the graph.

Then we use the 0-persistent node algorithm given in Section 3.3. to remove the 0-persistent nodes from the problem, and finally we solve it one more time. The detection of 0-persistent arcs is very time consuming, thus we do not include it in our preprocessing.

For the minimax regret single source shortest path problem, we use the formulation proposed in Section 3.4 coded in Mosel and XPRESS to solve the mixed integer programming part. We first solve the problem with all the arcs in the graph. Then we use the T 0-persistent arc algorithm given in Section 3.4. to remove the T 0-persistent arcs and finally we resolve it one more time.

Their performances are tested on randomly generated instances, and we use the cpu time as criteria to compare their performances. All the computational experiments were made on a 2 CPU Pentium 4 with 3GHz station under Linux 2.6.22.7-85.fc7 with 2 GB of RAM. The preprocessing algorithms were coded in C (compiler gcc) and use the LEDA-4.2 library to solve the shortest path and the single source shortest path problems (Dijkstra algorithm).

We generate the input data as follows. We first generate a random graph without self-loops nor parallel arcs, with the graph generator of LEDA-4.2. A graph in this model consists of m nodes and n random edges. A random edge is generated by selecting a random element (v, w) from the set C of all $m(m-1)$ pairs of distinct nodes. Upon selection of a pair (v, w) from C the pair is removed from C. The second part of the input data are arc lengths, i.e., upper and lower bounds. We generate this part of input as Karasan et al. (2001), considering two cases, that is to say, we first generate a random base case scenario from an uniform distribution between numbers 1 and 20 or between 1 and 100, that is $U(1, 20)$ or $U(1, 100)$ for a given arc. We denote by c_{ij}^0, the value of the base case scenario for arc (i, j). Then, the lower bounds $\underline{l}_{ij}$ are randomly generated from a uniform distribution $U((1-d)c_{ij}^0), (1+d)c_{ij}^0)$ where d is a pre-specified number $(0 < d < 1)$. Then, the upper bounds $\overline{l}_{ij}$ are generated from $U(\underline{l}_{ij} + 1, (1+d)c_{ij}^0)$.

For the minimax regret shortest path problem with interval uncertainty, we investigate the behavior of the 0-persistent node algorithm used as preprocessing. Table 3.3 through Table 3.8 show the computational results for graphs with 150 through 1000 nodes, with different number of arcs, with deviation parameter 0.3, 0.6, and 0,9 and with uniform distribution $U(1, 20)$ for Tables 3.3 to 3.5 and $U(1, 100)$ for Tables 3.6 to 3.8. A series contains 10 randomly generated problems with the same number of nodes, percentage deviation d and number of arcs. We give for each series, the mean value μ and the standard deviation σ of the number of 0-persistent nodes, of the computing times in seconds taken by the corresponding preprocessing, (given in the column labeled *prep*) and of the solution times of the problem before and after the preprocessing, (given in columns labeled *cpu1* and *cpu2* respectively). The time reductions are given in the column labeled *reduct* and corresponds to the difference $\mu(cpu1) - (\mu(cpu2) + \mu(prep))$ divided by $\mu(cpu1)$.

Table 3.3 through Table 3.8 show that when the density of the graph increases, the percentage of the number of 0-persistent nodes decreases for d equal to 0.6, and 0,9. Contrary to the case when $d = 0.3$ where the percentage of 0-persistent nodes seems to be relatively stable for uniform distributions $U(1, 20)$ and $U(1, 100)$. While the deviation parameter d increases, the number of 0-persistent nodes decreases. This can be explained as follows. In the case when d is higher we have less empty intersections between intervals and thus less 0-persistent nodes. If we compare the results between the base case $U(1, 20)$ and $U(1, 100)$, the percentage of 0-persistent nodes is higher in the last case. In conclusion, it appears that the percentage of the 0-persistent nodes depends on the deviation parameter and then on the interval lengths.

Table 3.3 through Table 3.8 show that, on average, we obtain a time reduction of 45.69 % for the base case $U(1, 20)$ and 51.97 % for base case $U(1, 100)$. Hence, our preprocessing can vastly decrease the computing time to solve the minimax regret shortest path problem.

For the minimax regret single source shortest path problem with interval uncertainty, we investigate the behavior of the number of the T 1-persistent and T 0-persistent arcs on digraphs of big size and the cpu time reductions. The number of arcs, the mean value μ and standard deviation σ of the number of T 1-persistent and of the T 0-persistent arcs in the graph are reported in order to compare the numbers (given in the column labeled *T1-p* and labeled *T0-p* respectively). Computing times in seconds spent by the corresponding preprocessing are given in the column labeled *prep*. The columns labeled *cpu1* and *cpu2* corresponds to the solution times of the problem before and after the preprocessing respectively. The time reductions are given in the column labeled *reduct* and corresponds to the difference $\mu(cpu1)-(\mu(cpu2)+\mu(prep))$ divided by $\mu(cpu1)$.

Tables 3.9 and 3.10 present the behavior of the algorithm on problems with 150 through 300 nodes, with number of arcs equal to 1000 through 5000 with deviation parameter 0.3, 0.6, and 0,9, and with the base case $U(1, 20)$ and $U(1, 100)$ respectively.

Tables 3.9 and 3.10 show that when the density of the graph increases, the percentage of the number of T 0-persistent arcs increases as well for d equal to 0.3, 0.6, and 0,9. Contrary to the percentage of T 1-persistent arcs, that seems to decrease significantly. While the deviation parameter increases, the number of T 1-persistent and T 0-persistent arcs decreases. If we compare the results between the base case $U(1, 20)$ and $U(1, 100)$, the percentage of T 0-persistent and T 1-persistent arcs is higher in the last case.

In conclusion, it appears that also for this problem, the percentage of the T 1-persistent and T 0-persistent arcs depends on the deviation parameter and then on the interval lengths. We obtain a time reduction of 62.14 % for the base case $U(1, 20)$ and 91.62 % for base case $U(1, 100)$. Then the T 0-persistent arc algorithm used as preprocessing can dramatically decrease the computer time to solve the minimax regret single source shortest path problem.

node	d	arcs	0-pers nodes $\mu \pm \sigma$	prep $\mu \pm \sigma$	cpu1 $\mu \pm \sigma$	cpu2 $\mu \pm \sigma$	reduct μ
150	0.3	1000	140.20 ±7.42	0.01 ±0.01	0.18 ±0.09	0.04 ±0.02	70.14 %
150	0.3	5000	128.00 ±22.25	0.03 ±0.01	0.80 ±0.42	0.09 ±0.09	85.08 %
150	0.3	15000	128.20 ±17.97	0.13 ±0.05	1.86 ±0.61	0.09 ±0.06	88.29 %
210	0.3	1000	195.70 ±14.70	0.04 ±0.01	0.31 ±0.15	0.15 ±0.04	41.45 %
210	0.3	5000	189.50 ±22.65	0.18 ±0.07	2.88 ±2.39	0.22 ±0.20	86.04 %
210	0.3	15000	177.00 ±20.29	0.53 ±0.19	7.99 ±2.78	0.29 ±0.12	89.72 %
210	0.3	30000	185.40 ±22.88	0.39 ±0.14	5.24 ±2.04	0.15 ±0.12	89.78 %
300	0.3	1000	270.10 ±19.89	0.03 ±0.01	0.29 ±0.10	0.12 ±0.02	46.71 %
300	0.3	5000	275.90 ±21.78	0.08 ±0.03	1.32 ±0.83	0.15 ±0.04	82.51 %
300	0.3	15000	260.40 ±33.25	0.30 ±0.11	4.19 ±1.93	0.21 ±0.12	87.64 %
300	0.3	30000	243.90 ±55.68	0.59 ±0.21	6.26 ±2.12	0.39 ±0.54	84.27 %
300	0.3	50000	266.70 ±23.67	0.93 ±0.33	11.20 ±2.28	0.22 ±0.11	89.70 %
1000	0.3	10000	960.30 ±47.23	1.12 ±0.40	5.19 ±1.64	1.22 ±0.10	54.89 %
1000	0.3	50000	940.50 ±53.94	4.04 ±1.42	21.30 ±8.85	1.25 ±0.09	75.16 %
1000	0.3	250000	897.40 ±110.27	15.69 ±5.52	54.12 ±4.95	2.05 ±1.76	67.22 %
Average reduction 75.91 %							

Table 3.3: 0-persistent nodes algorithm for base case $U(1, 20)$ for $d = 0.3$.

node	d	arcs	0-pers nodes $\mu \pm \sigma$	prep $\mu \pm \sigma$	cpu1 $\mu \pm \sigma$	cpu2 $\mu \pm \sigma$	reduct μ
150	0.6	1000	107.80 ±24.00	0.01 ±0.01	0.23 ±0.11	0.07 ±0.04	64.30 %
150	0.6	5000	112.60 ±22.95	0.03 ±0.01	0.75 ±0.30	0.10 ±0.06	82.88 %
150	0.6	15000	91.00 ±60.83	0.13 ±0.05	2.48 ±0.85	0.95 ±1.18	56.46 %
210	0.6	1000	174.30 ±30.59	0.05 ±0.02	0.63 ±0.33	0.17 ±0.09	64.41 %
210	0.6	5000	130.50 ±77.33	0.15 ±0.06	2.75 ±1.57	1.06 ±1.62	55.99 %
210	0.6	15000	136.40 ±59.23	0.51 ±0.20	6.44 ±1.95	1.43 ±1.86	69.93 %
210	0.6	30000	83.60 ±58.47	0.39 ±0.14	5.46 ±0.75	2.47 ±2.14	47.67 %
300	0.6	1000	251.89 ±30.82	0.03 ±0.01	0.30 ±0.13	0.13 ±0.03	46.84 %
300	0.6	5000	161.70 ±88.45	0.08 ±0.03	1.61 ±0.77	0.82 ±1.33	43.90 %
300	0.6	15000	199.30 ±87.15	0.30 ±0.11	3.88 ±1.90	0.93 ±1.65	68.34 %
300	0.6	30000	162.30 ±108.48	0.59 ±0.21	10.22 ±5.54	3.23 ±4.39	62.60 %
300	0.6	50000	181.80 ±112.40	0.93 ±0.33	8.70 ±1.11	2.71 ±3.96	58.11 %
1000	0.6	10000	835.40 ±111.98	1.12 ±0.39	6.10 ±1.22	1.36 ±0.15	59.38 %
1000	0.6	50000	717.10 ±188.72	4.00 ±1.41	23.22 ±11.20	2.86 ±2.50	70.47 %
1000	0.6	250000	315.70 ±340.30	15.91 ±5.61	70.82 ±9.97	45.66 ±31.97	13.07 %
Average reduction 57.62 %							

Table 3.4: 0-persistent nodes for base case $U(1,20)$ for $d = 0.6$.

node	d	arcs	0-pers nodes $\mu \pm \sigma$	prep $\mu \pm \sigma$	cpu1 $\mu \pm \sigma$	cpu2 $\mu \pm \sigma$	reduct μ
150	0.9	1000	85.40 $\pm$55.79	0.01 $\pm$0.01	0.25 $\pm$0.21	0.16 $\pm$0.24	29.09 %
150	0.9	5000	0.67 $\pm$0.58	0.02 $\pm$0.02	0.83 $\pm$0.30	0.82 $\pm$0.31	-2.25 %
150	0.9	15000	0.00 $\pm$0.00	0.13 $\pm$0.05	3.63 $\pm$1.37	3.67 $\pm$1.41	-4.69 %
210	0.9	1000	95.60 $\pm$56.89	0.02 $\pm$0.01	0.38 $\pm$0.14	0.24 $\pm$0.18	32.94 %
210	0.9	5000	22.10 $\pm$61.60	0.05 $\pm$0.02	0.93 $\pm$0.17	0.86 $\pm$0.31	2.46 %
210	0.9	15000	28.10 $\pm$59.95	0.21 $\pm$0.08	3.70 $\pm$1.44	3.38 $\pm$1.97	3.10 %
210	0.9	30000	0.90 $\pm$2.85	0.39 $\pm$0.14	4.96 $\pm$0.66	4.93 $\pm$0.79	-7.32 %
300	0.9	1000	112.89 $\pm$83.41	0.03 $\pm$0.01	0.40 $\pm$0.16	0.32 $\pm$0.12	11.89 %
300	0.9	5000	28.20 $\pm$37.54	0.08 $\pm$0.03	1.56 $\pm$1.03	1.43 $\pm$1.15	3.51 %
300	0.9	15000	12.60 $\pm$36.01	0.30 $\pm$0.11	4.37 $\pm$1.80	3.77 $\pm$1.27	6.80 %
300	0.9	30000	0.00 $\pm$0.00	0.59 $\pm$0.21	8.91 $\pm$4.79	8.58 $\pm$3.91	-2.94 %
300	0.9	50000	0.00 $\pm$0.00	0.93 $\pm$0.33	11.37 $\pm$1.47	11.32 $\pm$1.42	-7.73 %
1000	0.9	10000	243.80 $\pm$282.95	1.13 $\pm$0.40	8.13 $\pm$2.80	5.85 $\pm$3.64	14.12 %
1000	0.9	50000	6.30 $\pm$15.16	4.01 $\pm$1.41	30.40 $\pm$10.47	29.57 $\pm$10.23	-10.49 %
1000	0.9	250000	0.20 $\pm$0.63	15.69 $\pm$5.51	109.80 $\pm$18.14	110.67 $\pm$18.66	-15.08 %
Average reduction 3.56 %							

Table 3.5: 0-persistent nodes for base case $U(1,20)$ for $d = 0.9$.

node	d	arcs	0-pers nodes $\mu \pm \sigma$	prep $\mu \pm \sigma$	cpu1 $\mu \pm \sigma$	cpu2 $\mu \pm \sigma$	reduct μ
150	0.3	1000	135.00 ±10.21	0.01 ±0.01	0.20 ±0.10	0.05 ±0.02	70.08 %
150	0.3	5000	135.80 ±7.80	0.03 ±0.01	0.57 ±0.22	0.06 ±0.03	83.12 %
150	0.3	15000	140.90 ±6.79	0.13 ±0.05	1.28 ±0.27	0.05 ±0.02	86.06 %
210	0.3	1000	197.56 ±9.77	0.02 ±0.01	0.16 ±0.10	0.06 ±0.01	50.18 %
210	0.3	5000	197.90 ±9.28	0.05 ±0.02	0.66 ±0.35	0.07 ±0.02	81.06 %
210	0.3	15000	193.00 ±9.01	0.20 ±0.07	1.75 ±0.52	0.09 ±0.02	83.47 %
210	0.3	30000	189.20 ±14.29	0.39 ±0.14	4.04 ±0.67	0.12 ±0.06	87.32 %
300	0.3	1000	278.75 ±16.83	0.04 ±0.02	0.30 ±0.16	0.15 ±0.04	32.53 %
300	0.3	5000	283.40 ±12.90	0.08 ±0.03	0.86 ±0.50	0.13 ±0.02	75.72 %
300	0.3	15000	284.80 ±10.88	0.31 ±0.11	1.52 ±0.45	0.13 ±0.02	71.03 %
300	0.3	30000	277.80 ±25.84	0.59 ±0.21	3.76 ±1.15	0.17 ±0.10	79.78 %
300	0.3	50000	285.10 ±13.19	0.93 ±0.33	7.27 ±1.37	0.14 ±0.03	85.29 %
1000	0.3	10000	972.70 ±25.87	1.13 ±0.40	3.96 ±1.58	1.18 ±0.04	41.64 %
1000	0.3	50000	915.10 ±84.31	4.00 ±1.41	22.33 ±13.51	1.35 ±0.33	76.02 %
1000	0.3	250000	919.80 ±70.66	15.65 ±5.50	53.89 ±4.93	1.49 ±0.45	68.21 %
Average reduction 71.43 %							

Table 3.6: 0-persistent nodes for base case $U(1, 100)$ for $d = 0.3$.

node	d	arcs	0-pers nodes $\mu \pm \sigma$	prep $\mu \pm \sigma$	cpu1 $\mu \pm \sigma$	cpu2 $\mu \pm \sigma$	reduct μ
150	0.6	1000	130.10 ±17.54	0.01 ±0.01	0.24 ±0.13	0.06 ±0.03	68.27 %
150	0.6	5000	120.10 ±31.79	0.03 ±0.01	0.46 ±0.25	0.10 ±0.10	72.65 %
150	0.6	15000	119.00 ±22.82	0.13 ±0.05	1.69 ±0.56	0.15 ±0.12	83.56 %
210	0.6	1000	190.60 ±11.00	0.02 ±0.01	0.25 ±0.10	0.08 ±0.02	61.31 %
210	0.6	5000	156.20 ±57.36	0.05 ±0.02	0.76 ±0.34	0.21 ±0.25	65.65 %
210	0.6	15000	173.20 ±23.18	0.20 ±0.07	1.72 ±0.36	0.13 ±0.07	80.94 %
210	0.6	30000	118.20 ±49.10	0.39 ±0.14	4.49 ±0.98	0.87 ±0.77	71.94 %
300	0.6	1000	278.40 ±15.77	0.03 ±0.01	0.24 ±0.11	0.11 ±0.02	40.03 %
300	0.6	5000	187.10 ±99.69	0.08 ±0.03	1.23 ±0.50	0.39 ±0.35	61.44 %
300	0.6	15000	241.40 ±64.62	0.31 ±0.11	2.37 ±1.05	0.34 ±0.46	72.64 %
300	0.6	30000	231.10 ±55.17	0.60 ±0.21	4.64 ±0.96	0.38 ±0.38	79.01 %
300	0.6	50000	217.70 ±87.84	0.93 ±0.33	7.88 ±1.75	1.14 ±1.95	73.69 %
1000	0.6	10000	838.80 ±159.16	1.14 ±0.40	5.40 ±2.30	1.44 ±0.40	52.19 %
1000	0.6	50000	636.30 ±259.91	4.02 ±1.41	22.34 ±8.34	4.68 ±5.66	61.09 %
1000	0.6	250000	453.00 ±364.97	15.74 ±5.53	60.48 ±4.27	30.34 ±29.87	23.81 %
Average reduction 64.55 %							

Table 3.7: 0-persistent nodes for base case $U(1, 100)$ for $d = 0.6$.

node	d	arcs	0-pers nodes $\mu \pm \sigma$	prep $\mu \pm \sigma$	cpu1 $\mu \pm \sigma$	cpu2 $\mu \pm \sigma$	reduct μ
150	0.9	1000	98.60 ±44.34	0.01 ±0.01	0.25 ±0.11	0.10 ±0.10	56.55 %
150	0.9	5000	22.30 ±26.64	0.03 ±0.01	0.76 ±0.42	0.63 ±0.46	13.96 %
150	0.9	15000	48.90 ±54.41	0.13 ±0.05	1.91 ±0.64	1.16 ±0.98	32.34 %
210	0.9	1000	134.80 ±71.99	0.02 ±0.01	0.27 ±0.19	0.15 ±0.12	38.85 %
210	0.9	5000	54.70 ±55.75	0.05 ±0.02	0.71 ±0.26	0.44 ±0.28	31.31 %
210	0.9	15000	69.20 ±78.29	0.20 ±0.07	1.98 ±0.77	1.11 ±0.88	33.99 %
210	0.9	30000	22.10 ±60.53	0.39 ±0.14	4.53 ±1.69	4.33 ±2.33	-4.34 %
300	0.9	1000	215.60 ±57.55	0.03 ±0.01	0.35 ±0.13	0.18 ±0.07	40.50 %
300	0.9	5000	66.20 ±98.14	0.09 ±0.03	1.20 ±0.61	1.02 ±0.72	8.16 %
300	0.9	15000	61.80 ±79.63	0.31 ±0.11	2.10 ±0.66	1.50 ±0.98	14.13 %
300	0.9	30000	66.30 ±73.19	0.59 ±0.21	4.48 ±0.93	2.79 ±1.77	24.64 %
300	0.9	50000	114.90 ±110.62	0.93 ±0.33	7.74 ±1.93	4.06 ±4.13	35.56 %
1000	0.9	10000	262.10 ±325.49	1.15 ±0.40	6.87 ±2.16	5.52 ±3.38	2.90 %
1000	0.9	50000	20.70 ±30.92	4.05 ±1.42	25.10 ±5.33	24.13 ±6.03	-12.25 %
1000	0.9	250000	0.50 ±1.58	15.78 ±5.55	95.70 ±15.75	96.57 ±15.54	-17.39 %
Average reduction 19.93 %							

Table 3.8: 0-persistent nodes for base case $U(1, 100)$ for $d = 0.9$.

node	d	arcs	T 1-p μ $\pm\sigma$	T 0-p μ $\pm\sigma$	prep μ $\pm\sigma$	cpu1 μ $\pm\sigma$	cpu2 μ $\pm\sigma$	reduct μ
150	0.3	1000	49.90 ±27.07	730.60 ±142.92	0.12 ±0.08	145.49 ±144.17	5.80 ±1.73	95.92 %
150	0.3	3000	37.10 ±7.22	2475.60 ±45.12	0.44 ±0.16	413.44 ±275.79	107.23 ±130.88	73.96 %
210	0.3	3000	53.60 ±7.00	2312.90 ±68.67	0.84 ±0.30	1289.67 ±1215.30	245.39 ±465.09	80.91 %
210	0.3	5000	41.80 ±4.29	4158.10 ±36.16	1.37 ±0.48	1395.26 ±378.36	1102.28 ±1184.96	20.90 %
300	0.3	3000	80.00 ±12.29	2086.70 ±130.70	3.49 ±1.25	9606.14 ±14935.25	4318.03 ±11316.86	55.01 %
150	0.6	1000	34.80 ±7.18	512.80 ±52.52	0.18 ±0.09	395.11 ±455.38	105.12 ±106.96	73.35 %
150	0.6	3000	15.40 ±4.22	2130.30 ±90.96	0.44 ±0.16	2393.54 ±2017.30	2147.44 ±3918.79	10.26 %
210	0.6	3000	19.00 ±4.06	1805.20 ±112.25	0.85 ±0.30	31001.67 ±51141.26	5148.69 ±5099.80	83.39 %
210	0.6	5000	14.00 ±6.07	3393.00 ±295.58	1.39 ±0.49	25770.30 ±32469.95	6855.08 ±7657.60	73.39 %
300	0.6	3000	32.10 ±6.90	1555.10 ±149.10	3.25 ±1.50	26077.07 ±20217.62	12325.37 ±17585.84	52.72 %
150	0.9	1000	11.90 ±6.67	372.90 ±230.40	0.14 ±0.08	1639.65 ±1290.91	592.94 ±538.35	63.83 %
Average reduction 62.14 %								

Table 3.9: T 0-persistent arcs for base case $U(1, 20)$ for $d = 0.3, 0.6, 0.9$

node	d	arcs	T 1-p μ $\pm\sigma$	T 0-p μ $\pm\sigma$	prep μ $\pm\sigma$	cpu1 μ $\pm\sigma$	cpu2 μ $\pm\sigma$	reduct μ
150	0.3	1000	81.00 ±29.54	776.10 ±81.36	0.14 ±0.08	75.45 ±49.43	3.39 ±0.46	95.32 %
150	0.3	3000	73.80 ±7.15	2715.30 ±19.22	0.46 ±0.16	281.12 ±124.58	3.96 ±0.38	98.43 %
210	0.3	3000	105.60 ±18.47	2587.30 ±57.27	1.16 ±0.64	715.24 ±376.20	15.54 ±8.71	97.66 %
210	0.3	5000	98.30 ±18.20	4575.00 ±55.11	1.73 ±0.94	1549.75 ±870.85	17.02 ±12.86	98.79 %
300	0.3	3000	138.10 ±49.59	2450.10 ±221.13	1.48 ±0.78	1189.85 ±674.11	41.52 ±37.37	96.39 %
150	0.6	1000	39.00 ±22.07	670.70 ±177.97	0.12 ±0.08	167.15 ±118.54	21.17 ±16.34	87.26 %
150	0.6	3000	41.10 ±12.60	2520.70 ±98.81	0.45 ±0.16	698.79 ±424.09	38.81 ±38.47	94.38 %
210	0.6	3000	37.20 ±14.63	2183.70 ±151.64	0.84 ±0.30	1642.36 ±1045.07	223.60 ±137.78	86.33 %
210	0.6	5000	39.60 ±20.91	4167.00 ±273.62	1.74 ±0.96	6296.83 ±11600.60	335.82 ±358.38	94.64 %
300	0.6	3000	59.00 ±17.31	1958.70 ±128.93	2.08 ±1.13	12191.55 ±12131.10	1160.61 ±935.60	90.46 %
150	0.9	1000	24.00 ±10.50	515.70 ±180.25	0.14 ±0.08	1379.07 ±1756.24	345.32 ±635.51	74.95 %
150	0.9	3000	5.50 ±4.48	1859.90 ±309.68	0.44 ±0.16	18890.67 ±24551.49	2845.62 ±3159.62	84.93 %
Average reduction 91.62 %								

Table 3.10: T 0-persistent arcs for base case $U(1, 100)$ for $d = 0.3, 0.6, 0.9$

Part II

Robust linear programming

Chapter 4

Linear programs under uncertainty in the objective function coefficients

4.1 Introduction

Uncertainty in linear programming may concern coefficients of the objective function, constraints or both. In this thesis we focus on linear programming under uncertainty in the objective function coefficients. We consider the minimax regret approach and we model the uncertainty set, first as a compact and convex subset of $\mathbb{R}^n$, then as a polytope, and finally as a Cartesian product of intervals.

A variety of business applications can be modeled as linear programs under uncertainty on the objective function coefficients, see Kouvelis and Yu (1997). As an example, Vallin (2007) presents an investment problem under budgetary constraint and supposes that the profitability rates of the investments are uncertain.

In Section 4.2, we define the minimax regret problem associated to a uncertain objective linear programming problem, and we give some definitions and notations. In Section 4.3, we present a literature survey on uncertain linear programming. In Section 4.4, we study the case when the uncertainty set is compact and convex and give some properties of the problem. In Section 4.5, we formulate the problem under polyhedral and under interval uncertainty. In Section 4.6 we provide an alternative proof of the NP-hardness of the minimax regret linear programming problem with interval uncertain objective function coefficients, to the one given by Averbakh and Lebedev (2005). In Section 4.7, we present special cases when the maximum regret and the minimax regret problems are polynomially solvable. In Section 4.8, we propose an algorithm to find an exact solution when the set of possible values for the uncertain coefficients is a polytope. Finally, we discuss its numerical results in the cases of polyhedral and interval uncertainty.

4.2 Problem definition and notation

Consider the class of linear programs **LP**

$$\begin{aligned} \text{minimize} \quad & cx \\ \text{s.t.} \quad & Ax \geq b \\ & x \geq 0. \end{aligned} \tag{4.1}$$

Assume that the coefficients of the objective function are uncertain. We shall call this class of problems *Uncertain Objective Linear Programming Problems* (UOLPP) and we shall denote by $\mathbf{c}$, the n dimensional vector of uncertain objective function coefficients. We assume that the matrix $A \in \mathbb{R}^{m \times n}$ and the vector $b \in \mathbb{R}^m$ are fixed and we suppose that the polyhedron $P = \{x \in \mathbb{R}^n : Ax \geq b, \ x \geq 0\}$ is nonempty and bounded. We denote by $V(P)$ the set of vertices of P.

We shall model uncertainty using the concept of *scenario*. A scenario is an assignment of possible values to each uncertain coefficient of the problem. We denote by S the set of all scenarii and by D the set of all possible values for the uncertain coefficients, which we shall call *uncertainty set*. For each $s \in S$, we shall denote by $c^s = (c_1^s, \dots c_n^s) \in D$ the vector that corresponds to scenario s, where c_i^s is the value of the coefficient i in that scenario.

In this case, to each $c^s \in D$ corresponds a deterministic linear programming problem

$$\begin{aligned} \text{minimize} \quad & c^s x \\ \text{s.t.} \quad & Ax \geq b \\ & x \geq 0. \end{aligned} \tag{4.2}$$

We denote by $\mathcal{X}^{*s}(P)$ the set of all optimal solutions to the problem for the scenario s and by x^{*s} a generic element of $\mathcal{X}^{*s}(P)$.

One possible approach to face a UOLPP is to consider a worst case analysis. This consists in looking for the best solution in the worst case situation.

In this approach there are no probabilities associated with the scenarii. Kouvelis and Yu (1997) suggest various definitions of robustness in linear programming under uncertainty. In particular, the problem to find a solution $x_a \in P$ that minimize the maximum total cost,

$$\max_{c^s \in D} c^s x_a = \min_{x \in P} \max_{c^s \in D} c^s x,$$

is called *minimax problem*. Another approach is to find a solution with the smallest worst case relative regret, that is, a solution $x_r \in P$ that, in terms of the objective function value, have the least worst percentage deviation from the optimal solution in all cases, i.e.,

$$\max_{c^s \in D} \frac{c^s x_r - c^s x^{*s}}{c^s x^{*s}} = \min_{x \in P} \max_{c^s \in D} \frac{c^s x - c^s x^{*s}}{c^s x^{*s}}.$$

An optimal solution to that problem will be called a *minimax relative regret solution*.

In this chapter, we consider the problem of finding a solution with the smallest worst case regret, that is, a solution that, in terms of the objective function value, deviates the least from the optimal solution in all cases. More precisely, consider the function

$$R_{max} : P \to \mathbb{R} \; : \; x \to R_{max}(x),$$

where

$$R_{max}(x) = \max_{c^s \in D} \max_{x' \in P} (c^s x - c^s x') = \max_{c^s \in D} (c^s x - c^s x^{*s}).$$

Given a solution $x \in P$, we call the value $R_{max}(x)$ the *maximum regret* of x. The optimization problem called *minimax regret problem* consists in finding $y \in P$ such that

$$R_{max}(y) = \min_{x \in P} R_{max}(x),$$

where y is called a *minimax regret solution* to the uncertain objective linear programming problem.

A *minimax worst scenario* for a solution x is a scenario in which $c^s x$, the cost of this solution, is maximum. A *worst minimax regret scenario* for a solution x is a scenario in which $c^s x - c^s x^{*s}$, i.e., the difference between the cost of the solution x and the cost of an optimal solution in this scenario, is maximum.

For a scenario $s \in S$, we denote by $s(k,t)$ the scenario for which the coefficients of the objective function are such that $(c_1^{s(k,t)}, \dots c_k^{s(k,t)}, \dots c_n^{s(k,t)}) = c^s + te_k$, where $t \in \mathbb{R}$ and e_k is the kth canonical vector in $\mathbb{R}^n$.

4.3 Literature survey

In the last years, considerable research has been devoted to continuous optimization under uncertainty, in particular on robust linear optimization. In order to give an idea of the different techniques useful to face a uncertain linear programming problem, we shall present here a survey of the literature. Specifically, we discuss several ways under which uncertainty can be modeled, complexity results, and approximation algorithms.

4.3.1 Robust counterparts of linear programs under uncertainty

Soyster (1973) proposed a linear optimization model to face uncertainty in the constraint coefficients of a linear program whose objective function must be maximized. Specifically, the author considered the model,

$$\begin{aligned} &\text{maximize} \;\; cx \\ &\text{subject to} \;\; \sum_{j}^{n} \mathbf{A}^j x_j \leq b \;\; \forall \mathbf{A}^j \in K_j, \quad j = 1, \dots n \\ &x \geq 0, \end{aligned}$$

where $\mathbf{A}^j$ is the jth column vector of the matrix $\mathbf{A}$ of uncertain coefficients and the uncertainty sets K_j are convex. This model constructs a solution that is feasible for all possible realizations of data that belong to a convex set. However, this approach finds solutions that are considered as over-conservative by several authors, because it provides a maximum protection level against infeasibilities, see for example Bertsimas and Sim (2004b).

To address this over-conservatism of the robust solutions, Ben-Tal and Nemirovsky (1999) Ben-Tal and Nemirovsky (1998) Ben-Tal and Nemirovsky (2000) and independently El-Ghaoui and Lebret (1997) El-Ghaoui et al. (1998) proposed models under the assumption that the coefficients of the constraint matrix vary in ellipsoidal uncertainty sets and presented a number of algorithms to solve such linear optimization problems.

Ben-Tal et al. (2004) considered a uncertain linear program with uncertainty in the coefficients of the constraints and in the objective function. In this model, values of a subset of variables must be determined before the realization of the uncertain parameters ("non-adjustable variables"), while the

Minimax regret linear optimization		
	Algorithms & heuristics	Theory and complexity
$c \in D$	Inuiguchi and Sakawa (1995)IU Inuiguchi and Sakawa (1996)IU Mausser and Laguna (1999)IU Mausser and Laguna (1998)IU △ algorithm PU △ algorithm IU △ preprocessing IU	strongly NP-hard Averbakh and Lebedev (2005) △ properties CCU △ properties IU △ polynomial cases PU △ polynomial cases IU
$c \in D$ finite	reducible to LP	polynomial Mausser and Laguna (1999) Mausser and Laguna (1998)
Other robust counterparts		
	Algorithms & heuristics	Theory and complexity
$A \in D$ $b \in D$	Ben-Tal and Nemirovsky (1998) Ben-Tal and Nemirovsky (1999) Ben-Tal and Nemirovsky (2000) El-Ghaoui and Lebret (1997) El-Ghaoui et al. (1998) Bertsimas and Sim (2004b) Soyster (1973)	Ben-Tal and Nemirovsky (1998) Ben-Tal and Nemirovsky (1999) Ben-Tal and Nemirovsky (2000) El-Ghaoui and Lebret (1997) El-Ghaoui et al. (1998) Bertsimas and Sim (2004b) Soyster (1973)
$A, b,$ $c \in D$	Bertsimas and Sim (2004b)	Ben-Tal et al. (2004) Bertsimas and Sim (2004b)
$A \in D$	* LP algorithm Soyster (1973) Ben-Tal and Nemirovsky (1998) Ben-Tal and Nemirovsky (1999) Ben-Tal and Nemirovsky (2000) El-Ghaoui and Lebret (1997) El-Ghaoui et al. (1998) Bertsimas and Sim (2004b)	* polynomial Soyster (1973) Ben-Tal and Nemirovsky (1998) Ben-Tal and Nemirovsky (1999) Ben-Tal and Nemirovsky (2000) El-Ghaoui and Lebret (1997) El-Ghaoui et al. (1998) Bertsimas and Sim (2004b)
$c \in D$	Bertsimas and Sim (2004b)	* Soyster ⇒polynomial Soyster (1973) Bertsimas and Sim (2004b)
$c \in D$ finite	* reducible to LP	*polynomial
$A \in D$ finite	* reducible to LP	* polynomial

The following notation mean *: Absolute robustness, △: In this work, IU: interval uncertainty, PU: polyhedral uncertainty, CCU: compact and convex uncertainty.

Table 4.1: Linear optimization under uncertainty.

values of the other variables can be chosen after the realization ("adjustable variables").

Bertsimas and Sim (2004b) presented an approach to solve linear optimization problems under interval uncertainty affecting the coefficients of the constraints and the coefficients of the objective. This situation reduces to the case when the objective function is not subject to uncertainty. The reduction consists in maximizing z and including the constraint $z - \mathbf{c}x \leq 0$ into $\mathbf{A}x \leq b$. The robust counterpart given in Bertsimas and Sim (2004b) is

$$\begin{aligned} &\text{maximize} \;\; cx && (4.3)\\ &\text{subject to} \sum_j a_{ij}x_j \\ &+ \max_{\{S_i \cup \{t_i\}: S_i \subseteq J_i, |S_i| = \lfloor \Gamma_i \rfloor, t_i \in J_i - S_i\}} \{\sum_{j \in S_i} \hat{a}_{ij} y_j + (\Gamma_i - \lfloor \Gamma_i \rfloor)\hat{a}_{it_i} y_{t_i}\} \leq b_i \;\; \forall i && (*)\\ &-y_j \leq x_j \leq y_j \;\; \forall j \\ &l \leq x \leq u \\ &y \geq 0. \end{aligned}$$

In this model J_i represent the set of indices corresponding to the uncertain coefficients appearing in the i-th constraint, denoted by $\mathbf{a}_{ij}$. The authors assume that such coefficients $\mathbf{a}_{ij}$ take values according to a symmetric distribution in the interval $[a_{ij} - \hat{a}_{ij}, a_{ij} + \hat{a}_{ij}]$. For every i, the parameter Γ_i, not necessarily integer, takes values in the interval $[0, |J_i|]$. The model controls the level of conservatism in the solution by supposing that only a subset of the coefficients of cardinality at most Γ_i are allowed to change for each constraint. In order to explain the role of the constraint (*), we first give the formulation corresponding to the maximum protection level, that is, when there are $|J_i|$ uncertain coefficients affecting the i-th constraint. Consider the model

$$\begin{aligned} &\text{maximize} \;\; cx && (4.4)\\ &\text{subject to} \sum_j a_{ij}x_j + \sum_{j \in J_i} \hat{a}_{ij} y_j \leq b_i \;\; \forall i \\ &-y_j \leq x_j \leq y_j \;\; \forall j \in J_i \\ &l \leq x \leq u \end{aligned}$$

$$y \geq 0,$$

since $\hat{a}_{ij} \geq 0 \ \ \forall i$ and $\forall j \in J_i$, the second constraint to problem (4.4) implies that

$$a_{ij}x_j + \hat{a}_{ij}x_j \leq a_{ij}x_j + \hat{a}_{ij}y_j \quad \forall i \ \forall j \in J_i$$

and

$$a_{ij}x_j - \hat{a}_{ij}x_j \leq a_{ij}x_j + \hat{a}_{ij}y_j \quad \forall i \ \forall j \in J_i.$$

Since $\mathbf{a}_{ij} \in [a_{ij} - \hat{a}_{ij}, a_{ij} + \hat{a}_{ij}]$,

$$a_{ij} - \hat{a}_{ij} \leq \mathbf{a}_{ij} \leq a_{ij} + \hat{a}_{ij},$$

thus if $x_j \geq 0$,

$$a_{ij}x_j - \hat{a}_{ij}x_j \leq \mathbf{a}_{ij}x_j \leq a_{ij}x_j + \hat{a}_{ij}x_j,$$

otherwise

$$a_{ij}x_j - \hat{a}_{ij}x_j \geq \mathbf{a}_{ij}x_j \geq a_{ij}x_j + \hat{a}_{ij}x_j,$$

in both of cases that implies that $\mathbf{a}_{ij}x_j \leq a_{ij}x_j + \hat{a}_{ij}y_j$ for all $i, j \in J_i$, and

$$\sum_j \mathbf{a}_{ij}x_j \leq \sum_j a_{ij}x_j + \sum_{j \in J_i} \hat{a}_{ij}y_j \leq b_i \quad \forall i,$$

thus the first constraint of problem (4.4) protects against all constraints violations. Now if we suppose that at most Γ_i coefficients are allowed to change for each constraint, we have the following formulation,

$$\begin{aligned}
&\text{maximize} \ \ cx \\
\text{s.t.} \ &\sum_j a_{ij}x_j \ + \sum_{j \in S_i} \hat{a}_{ij}y_j + (\Gamma_i - \lfloor \Gamma_i \rfloor)\hat{a}_{it_i}y_{t_i} \leq b_i \\
&\forall S_i \cup \{t_i\} : S_i \subseteq J_i, |S_i| = \lfloor \Gamma_i \rfloor, t_i \in J_i - S_i, \ \forall i \\
&-y_j \leq x_j \leq y_j \ \ \forall j \\
&l \leq x \leq u \\
&y \geq 0.
\end{aligned}$$

and we can replace the first set of constraints of this last problem by

$$\sum_j a_{ij}x_j + \max_{\{S_i \cup \{t_i\} : S_i \subseteq J_i, |S_i| = \lfloor \Gamma_i \rfloor, t_i \in J_i - S_i\}} \{\sum_{j \in S_i} \hat{a}_{ij}y_j + (\Gamma_i - \lfloor \Gamma_i \rfloor)\hat{a}_{it_i}y_{t_i}\} \leq b_i \quad \forall i$$

that protects against constraints violations caused by data perturbations. The authors proved that this approach leads to a robust formulation with the same complexity as the original problem and thus can by directly applied to optimization problems with 0-1 variables. Table 4.1 shows a classification of all these results and the results obtained in Chapter 4 and 5.

4.3.2 Linear programs under interval uncertainty in the objective function coefficients: the minimax regret model.

The problem of minimizing the maximum regret for linear programs under interval uncertainty affecting the coefficients of the objective function has been first addressed by Inuiguchi and Sakawa (1995). They proposed an enumerative approach that requires finding all solutions that are optimal for any extreme cost vector. Inuiguchi and Sakawa (1996), made use of a branch-and-bound procedure instead of enumeration. Mausser and Laguna (1998) proposed a new formulation for this problem and a solution algorithm. Specifically, at each iteration, it solves a linear program to generate a candidate solution and a mixed integer program (MIP) to find the corresponding maximum regret. Computational results are presented. Mausser and Laguna (1999) presented a heuristic for the MIP and discuss its performance.

Averbakh and Lebedev (2005) proved that the minimax regret linear programming problem with interval uncertain objective function coefficients is strongly NP-hard.

4.4 Linear problems under convex and compact uncertainty

Consider the minimax regret version of a UOLPP where, the uncertainty set D is a convex and compact set in $\mathbb{R}^n$. In this context, we can define the function

$$r : P \times D \to \mathbb{R} \ : \ (x, c^s) \to r(x, c^s)$$
$$r(x, c^s) = \max_{x' \in P}(c^s x - c^s x') = c^s x - c^s x^{*s},$$

and we observe that for a fixed $x \in P$ the function r becomes a function $r_x \ : \ D \to \mathbb{R} \ : \ c^s \to r_x(c^s)$, where $r_x(c^s) = c^s x - c^s x^{*s}$. Then we can also write the maximum regret of x as

$$R_{max}(x) = \max_{c^s \in D} r_x(c^s).$$

So, in order to study the maximum regret function R_{max}, we shall first analyze the properties of the function r_x.

4.4.1 Properties

In order to localize the worst and best minimax regret scenarii for a fixed $x \in P$ we shall study the properties of the function r_x in the case of a compact and convex uncertainty set.

Lemma 4.1 *The function r_x is continuous and piecewise linear.*

Proof. The function $c^s \to c^s x^{*s}$ is such that

$$c^s x^{*s} = \min_{x' \in V(P)} c^s x',$$

since the number of vertices of the polyhedron P is finite, this function is the minimum of a finite number of linear functions and is thus piecewise linear. Since

$$r_x(c^s) = c^s x - c^s x^{*s} = c^s x - \min_{x' \in V(P)} c^s x',$$

thus r_x is the difference of a linear function and a piecewise linear function, hence r_x is continuous and piecewise linear. □

Lemma 4.2 *The function r_x is convex.*

Proof. Let $c^{s_1}, c^{s_2} \in D$. Since D is convex, for all $\lambda \in [0,1]$, $c^{s_0} = \lambda c^{s_1} + (1-\lambda)c^{s_2} \in D$ and thus

$$r_x(c^{s_0}) = (\lambda c^{s_1} + (1-\lambda)c^{s_2})x - (\lambda c^{s_1} + (1-\lambda)c^{s_2})x^{*s_0} =$$

$$\lambda(c^{s_1}x - c^{s_1}x^{*s_0}) + (1-\lambda)(c^{s_2}x - c^{s_2}x^{*s_0}) \leq$$

$$\lambda(c^{s_1}x - c^{s_1}x^{*s_1}) + (1-\lambda)(c^{s_2}x - c^{s_2}x^{*s_2}) = \lambda r_x(c^{s_1}) + (1-\lambda) r_x(c^{s_2}),$$

hence r_x is convex. □

In order to study the differentiability of the function r_x, we consider the following definition.

Definition 4.1 *Let* $\Omega = \bigcup_{k=1}^{n} \Omega(k)$ *where*

$$\begin{array}{rcllll} \Omega(k) & = & \{ \; c^s \in D: & \exists\; y, w \in \mathcal{X}^{*s}(P) & \textit{and} \;\; t \geq 0 \textit{ such that} & \\ & & (c^{s(k,t)} \in D, & y \in \mathcal{X}^{*s(k,t)}(P) & \textit{and } w \notin \mathcal{X}^{*s(k,t)}(P)) & \textit{or} \\ & & (c^{s(k,-t)} \in D, & w \in \mathcal{X}^{*s(k,-t)}(P) & \textit{and } y \notin \mathcal{X}^{*s(k,-t)}(P)) \; \}. & \end{array}$$

The set Ω can be geometrically interpreted as the set of instance data $c^s \in D$ for which $\mathcal{X}^{*s}(P)$ is a face of dimension at least one of the polytope P.

In the following lemma we prove that Ω is exactly the set of instance data where the function r_x is not differentiable.

Lemma 4.3 *For all $x \in P$ the function r_x is differentiable over $D \setminus \Omega$. Moreover for all $c^{s_o} \in \Omega$, r_x is not differentiable on c^{s_o}.*

Proof. If $c^s \in D \setminus \Omega$, then there exists a neighborhood $B(c^s, t) = \{ c \in \mathbb{R}^n : \mid c - c^s \mid < t\}$ of c^s such that for all $c^{s_o} \in B(c^s, t) \bigcap D$, $\mathcal{X}^{*s}(P) = \mathcal{X}^{*s_o}(P)$, then r_x is differentiable over $B(c^s, t) \bigcap D$ and then over $D \setminus \Omega$.

Let $k \in \{1, \ldots, n\}$ such that $\Omega(k) \neq \emptyset$, then there exist $c^s \in \Omega(k)$ and $y, w \in \mathcal{X}^{*s}(P)$ such that for all $t \geq 0$, if $c^{s(k,t)} \in D$ then $y \in \mathcal{X}^{*s(k,t)}(P)$ and $w \notin \mathcal{X}^{*s(k,t)}(P)$ and if $c^{s(k,-t)} \in D$, $w \in \mathcal{X}^{*s(k,-t)}(P)$ and $y \notin \mathcal{X}^{*s(k,-t)}(P)$. Under these conditions we observe that $y_k \neq w_k$.

In order to prove that for all $c^s \in \Omega(k)$, r_x is not differentiable on c^s we shall prove that $\frac{\partial r_x}{\partial c_k}(c^s)$ does not exist.

By definition

$$\frac{\partial r_x}{\partial c_k}(c^s) \;=\; \lim_{h \to 0} \frac{r_x(c^{s(k,h)}) - r_x(c^s)}{h}.$$

If $h < 0$ and if $c^{s(k,h)} \in D$ then $w \in X^*_{s(k,h)}(P)$ and $y \notin \mathcal{X}^{*s(k,h)}(P)$. Then,

$$\lim_{h \to 0^-} \frac{r_x(c^{s(k,h)}) - r_x(c^s)}{h} = \lim_{h \to 0^-} \frac{(c^{s(k,h)}x - c^{s(k,h)}x^{*s(k,h)}) - (c^s x - c^s x^{*s})}{h}.$$

Since $c^{s(k,h)}w = c^s w + h w_k$ and $c^{s(k,h)}x = c^s x + h x_k$,

$$\lim_{h \to 0^-} \frac{(c^s x + h x_k) - (c^s w + h w_k) - (c^s x - c^s w)}{h} \lim_{h \to 0^-} \frac{h x_k - h w_k}{h} = x_k - w_k.$$

If $h > 0$ and if $c^{s(k,h)} \in D$ then $y \in \mathcal{X}^{*s(k,h)}(P)$ and

$$\lim_{h \to 0^+} \frac{r_x(c^{s(k,h)}) - r_x(c^s)}{h} = \lim_{h \to 0^+} \frac{(c^{s(k,h)}x - c^{s(k,h)}x^{*s(k,h)}) - (c^s x - c^s x^{*s})}{h}.$$

Since $c^{s(k,h)}y = c^s y + h y_k$ and $c^{s(k,h)}x = c^s x + h x_k$, then

$$\lim_{h \to 0^+} \frac{(c^s x + h x_k) - (c^s y + h y_k) - (c^s x - c^s y)}{h} \lim_{h \to 0^+} \frac{h x_k - h y_k}{h} = x_k - y_k,$$

and this implies that $\frac{\partial r_x}{\partial c_k}(c^s)$ does not exist and thus r_x is not differentiable over Ω. □

Next theorem gives the location of the worst and best minimax regret scenarii for $x \in P$ when the set of scenarii is compact and convex. In the case where D is a Cartesian product of intervals the problem was studied by Inuiguchi and Sakawa (1995).

Theorem 4.1 *Let $x \in P$, if D is a compact and convex subset of $\mathbb{R}^n$ of dimension at least one, then the data instances that correspond to the worst and best minimax regret scenarii for x are on the boundary ∂D of D and on $\partial D \cup \Omega$ respectively.*

Proof. This property holds true since, because r_x is a piecewise linear and convex function defined in a compact and convex set. □

Corollary 4.1 *If D is a polytope, the maximum regret of $x \in P$ can be calculated as*

$$R_{max}(x) = \max_{c^s \in V(D)} r_x(c^s) = \max_{c^s \in V(D)} \{c^s x - c^s x^{*s}\} = \max_{c^s \in V(D)} \{c^s x - \min_{x' \in V(P)} c^s x'\}.$$

Proof. By Theorem 4.1, the maximum regret of a $x \in P$ is

$$R_{max}(x) = \max_{c^s \in \partial D} r_x(c^s) = \max_{c^s \in \partial D} \{c^s x - c^s x^{*s}\} = \max_{c^s \in \partial D} \{c^s x - \min_{x' \in V(P)} c^s x'\}.$$

If D is a polytope, the maximum value can always be reached in a vertex of D, hence

$$R_{max}(x) = \max_{c^s \in V(D)} \{c^s x - \min_{x' \in V(P)} c^s x'\},$$

see Figure 4.1 □

Let us now study the function R_{max} that gives the maximum regret of a point $x \in P$.

Lemma 4.4 *If D is a convex and compact subset of $\mathbb{R}^n$, the function R_{max} is convex.*

Proof. Let $P = \{x \in \mathbb{R}^n : Ax \geq b,\ x \geq 0\}$, a nonempty polytope, since P is a convex set, for all $\lambda \in [0,1]$ and all $x, y \in P$, $\lambda x + (1-\lambda)y \in P$ and by definition we have the following equality

$$R_{max}(\lambda x + (1-\lambda)y) = \max_{c^s \in D} \{c^s(\lambda x + (1-\lambda)y) - c^s x^{*s}\}.$$

Let $c^{s_1} \in D$ such that

$$R_{max}(\lambda x + (1-\lambda)y) = c^{s_1}(\lambda x + (1-\lambda)y) - c^{s_1} x^{*s_1} = \lambda c^{s_1} x + (1-\lambda) c^{s_1} y - c^{s_1} x^{*s_1},$$

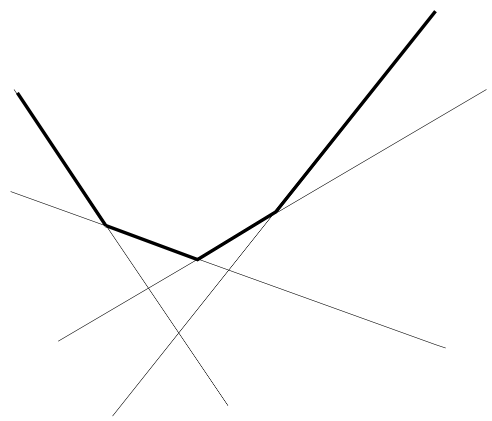

Figure 4.1: The function R_{max}

since $c^{s_1}x^{*s_1} = \lambda c^{s_1}x^{*s_1} + (1-\lambda)c^{s_1}x^{*s_1}$

$$R_{max}(\lambda x + (1-\lambda)y) = \lambda c^{s_1}x + (1-\lambda)c^{s_1}y - \lambda c^{s_1}x^{*s_1} - (1-\lambda)c^{s_1}x^{*s_1} =$$

$$\lambda(c^{s_1}x - c^{s_1}x^{*s_1}) + (1-\lambda)(c^{s_1}y - c^{s_1}x^{*s_1}) \leq \lambda R_{max}(x) + (1-\lambda)R_{max}(y),$$

hence R_{max} is a convex function. □

From the last lemma, it follows that a point that minimizes the maximum regret can be located in the interior of the polytope P. We observe that due to Corollary 4.1, in the case where D is a polytope the proof of the last lemma is easier. Since the number of vertices of D is finite, by Lemmas 4.1 and 4.2, R_{max} is the maximum of a finite number of convex and piecewise linear functions. Hence, R_{max} is convex and piecewise linear function.

4.5 Linear problems under polyhedral uncertainty

In this part we consider the case when the uncertainty set is a nonempty polytope of the form $D = \{c \in \mathbb{R}^n : c^tE \leq e^t, c \geq 0\}$, where $E \in \mathbb{R}^{n\times p}$ and $e \in \mathbb{R}^p$. We give an exact algorithm to find an optimal solution to the minimax regret problem in the case of polyhedral uncertainty. Finally, we present numerical experiments and we discuss the performance of the algorithm.

4.5.1 Formulation

In the case when the uncertainty set D is a polytope, the minimax regret problem associated to the UOLPP, can be formulated as a bilevel programming problem.

Bilevel MinMaxR problem

$$\begin{aligned}
\min_{x} \quad & \sum_{j=1}^{n} c_j^s(x_j - y_j) \qquad (4.5)\\
\text{s.t.} \quad & Ax \geq b\\
& x \geq 0\\
& c^s, y \in Arg\max_{c^s, y} \ \sum_{j=1}^{n} c_j^s(x_j - y_j)\\
& \qquad \text{s.t.} \ \ Ay \geq b\\
& \qquad y \geq 0\\
& \qquad c^s \in D.
\end{aligned}$$

The second level of this problem is the maximum regret problem.

$$\begin{aligned}
\max_{c^s, y} \quad & \sum_{j=1}^{n} c_j^s(x_j - y_j) \qquad (4.6)\\
\text{st} \quad & Ay \geq b\\
& y \geq 0\\
& c^s \in D.
\end{aligned}$$

The minimax regret and the maximum regret problems can be reformulated as follows

$$\begin{aligned}
& \min \quad z \qquad (4.7)\\
\text{s.t.} \quad & c^s(x - y) \leq z, \quad \forall\, y \in P, \ \forall c^s \in D,\\
& Ax \geq b\\
& x \geq 0,
\end{aligned}$$

and

$$\begin{aligned}
& \min \quad z \qquad (4.8)\\
\text{s.t.} \quad & c^s(x - y) \leq z, \quad \forall\, y \in P, \ \forall c^s \in D,
\end{aligned}$$

respectively. Consequences of Corollary 4.1 are the following formulations to the maximum regret and minimax regret problems under polyhedral uncertainty.

MaxR polyhedral problem

$$\begin{aligned} &\min z \\ \text{s.t.} \quad & c^s x - z \leq c^s y, \quad \forall\, y \in V(P),\ \forall c^s \in V(D). \end{aligned} \tag{4.9}$$

MinMaxR polyhedral problem

$$\begin{aligned} &\min z \\ \text{s.t.} \quad & c^s(x-y) \leq z \quad \forall\, y \in V(P)\ \forall c^s \in V(D) \qquad (**) \\ & Ax \geq b \\ & x \geq 0, \end{aligned} \tag{4.10}$$

where $P = \{x \in \mathbb{R}^n : Ax \geq b,\ x \geq 0\}$ is a polytope, $V(P)$ is the set of vertices of P and $V(D)$ is the set of vertices of the polytope $D = \{c \in \mathbb{R}^n : c^t E \leq e^t, c \geq 0\}$.

4.5.2 Linear problems under interval uncertainty

If we assume that D is a Cartesian product of intervals, i.e., $D = \prod_{i=1}^{n} \left[\underline{c_i}, \overline{c_i}\right]$ and $\underline{c_i} \leq \overline{c_i}$, the maximum regret and the minimax regret problems under interval uncertainty can be formulated in the following way.

MaxR interval problem

$$\begin{aligned} &\min z \\ \text{s.t.} \quad & c^s x - z \leq c^s y, \quad \forall\, y \in V(P), \\ & \forall c^s = (c_1^s, \ldots, c_j^s, \ldots, c_n^s), \text{ with } c_j^s \in \{\underline{c}_j, \overline{c}_j\},\ j = 1 \ldots n. \end{aligned} \tag{4.11}$$

and the minimax regret problem as

MinMaxR interval problem

$$\begin{aligned} &\min z \\ \text{s.t.} \quad & c^s(x-y) \leq z, \quad \forall\, y \in V(P), \\ & \forall c^s = (c_1^s, \ldots c_j^s, \ldots, c_n^s), \text{ with } c_j^s \in \{\underline{c}_j, \overline{c}_j\},\ j = 1 \ldots n, \\ & Ax \geq b, \\ & x \geq 0. \end{aligned} \tag{4.12}$$

We denote by $\underline{s}$ the scenario for which for all $i = 1 \ldots n$, $c_i^{\underline{s}} = \underline{c}_i$ and by $\overline{s}$ the scenario for which for all $i = 1 \ldots n$, $c_i^{\overline{s}} = \overline{c}_i$.

4.6 Complexity

The following definition can be found for instance, in Bertsimas and Tsitsiklis (1997).

Definition 4.2 *Given a polyhedron $P \subset \mathbb{R}^n$, the separation problem is to:*
a) Either decide that $x \in P$, or
b) Find a vector d such that $d'x < d'y$ for all $y \in P$.

In problems with exponentially many constraints, if we can solve the separation problem for a family of polyhedra in polynomial time then we can also solve the linear optimization problem in polynomial time, see e.g. Bertsimas and Tsitsiklis (1997). In the context of our problem this means that if we can solve the maximum regret problem in polynomial time then we can also solve the minimax regret problem in polynomial time.

The minimax regret problem under interval uncertainty was shown to be strongly NP-hard, see Averbakh and Lebedev (2005). Hence, the maximum regret problem under interval uncertainty and the minimax regret and the maximum regret problems under polyhedral uncertainty are implicitly strongly NP-hard. The following proof to the NP-hardness of the maximum regret problem under interval uncertainty can be seen as an alternative though weaker result. The proof of this result allowed us to identify some polynomial cases that we shall present in the following section. We first present decision versions of the maximum regret under interval uncertainty and the knapsack problems.

MaxR interval problem: Let $K' \in \mathbb{R}$, $\underline{c}_i, \overline{c}_i \in \mathbb{R}$ for each $i \in \{1, \ldots, n\}$ such that $\underline{c}_i \leq \overline{c}_i$, a $m \times n$ matrix A , $b \in \mathbb{R}^m$ and $x \in \mathbb{R}^n_+$ such that $Ax \geq b$. Does there exist $c \in \mathbb{R}^n$ such that $\underline{c}_i \leq c_i \leq \overline{c}_i$ for all $i \in \{1, \ldots, n\}$ and $y \in \mathbb{R}^n_+$ such that $Ay \geq b$ and $c(x - y) \geq K'$?

Knapsack problem: Given positive integers a_j, $w_j \in \mathbb{Z}^+$ for each $j \in \{1, \ldots, q\}$ and $B, K \in \mathbb{Z}^+$ such that $\sum_{j=1}^{q} a_j \geq B$, does there exist $z \in \{0,1\}^q$ such that $\sum_{j=1}^{q} w_j z_j \leq K$ and $\sum_{j=1}^{q} a_j z_j \geq B$?.

Theorem 4.2 *The maximum regret problem is NP-hard.*

Proof. We prove that the 0-1 knapsack problem, which is NP-complete, reduces to MaxR interval problem.

Given any instance of the knapsack problem **KP**, we can construct the following instance of the MaxR interval problem.

For $n = q+1,\ A = \begin{bmatrix} 1 & \cdots & 1 \\ & -I_{q+1} & \end{bmatrix}$ and $b = \begin{bmatrix} B \\ -a_1 \\ \cdot \\ \cdot \\ -a_q \\ -B \end{bmatrix}$, if there exists $y \in \mathbb{R}^n_+$ such that $Ay \geq b$, then $\sum_{j=1}^{q+1} y_j \geq B$, $0 \leq y_j \leq a_j$ for $j = 1, \ldots, q$ and $0 \leq y_{q+1} \leq B$.

Set $\underline{c}_j = 0$ for $j \in \{1, \ldots, q\}$, $\underline{c}_{q+1} = M$ (with M $>> \sum_{j=1}^{q} w_j$),

$\overline{c}_j = w_j$ for $j \in \{1, \ldots, q\}$, $\overline{c}_{q+1} = M$, $x_j = 1$ for $j \in \{1, \ldots, q\}$, $x_{q+1} = B$ and

$$K' = \sum_{j=1}^{q} \overline{c}_j - K + BM.$$

We call this special MaxR interval problem **MaxR-KP** and we can rewrite it as follows,

$$\max_{c,y} \sum_{j=1}^{q} c_j(1 - y_j) + M(B - y_{q+1})$$

$$\text{st} \quad \sum_{j=1}^{q+1} y_j \geq B$$

$$0 \leq y_j \leq a_j \quad for\ j \in \{1, \ldots, q\}$$

$$0 \leq y_{q+1} \leq B$$

$$y \geq 0$$

$$c_j \in [0, \overline{c}_j]$$

Let us now show that the knapsack problem has a solution if and only if there exists a feasible solution to MaxR-KP with value at least K'. Suppose that the knapsack problem has a solution $\overline{z}$, i.e., there exist $\overline{z} \in \{0,1\}^q$ such that $\sum_{j=1}^{q} w_j \overline{z}_j \leq K$ and $\sum_{j=1}^{q} a_j \overline{z}_j \geq B$. For $j \in \{1, \ldots, q\}$ consider the solution $(\tilde{y}, \tilde{c}) = (\tilde{y_1}, \ldots, \tilde{y_{q+1}}, \tilde{c_1}, \ldots, \tilde{c_{q+1}})$ such that

$\tilde{y}_j = \begin{cases} a_j & \text{if} \quad \overline{z}_j = 1 \\ 0 & \text{if} \quad \overline{z}_j = 0. \end{cases}$ i.e. $\tilde{y}_j = a_j \overline{z}_j$,
$\tilde{y}_{q+1} = 0$ and
$\tilde{c}_j = \begin{cases} \overline{c}_j & \text{if} \quad \overline{z}_j = 0 \\ 0 & \text{if} \quad \overline{z}_j = 1. \end{cases}$

Since $\overline{z}$ is a feasible solution for KP, $\sum_{j=1}^{q+1} \tilde{y}_j = \sum_{j=1}^{q} \tilde{y}_j = \sum_{j=1}^{q} a_j \overline{z}_j \geq B$. By construction, for all $j \in \{1, \ldots, q\}$, $0 \leq \tilde{y}_j \leq a_j$ and $0 \leq \tilde{y}_{q+1} \leq B$ then $(\tilde{y}, \tilde{c})$ is a feasible solution to MaxR-KP. Moreover,

$$\sum_{j=1}^{q} \tilde{c}_j (1 - \tilde{y}_j) + M(B - \tilde{y}_{q+1}) = \sum_{j=1}^{q} \tilde{c}_j (1 - a_j \overline{z}_j) + MB =$$

$$\sum_{j=1}^{q} \overline{c}_j (1 - \overline{z}_j) + MB = \sum_{j=1}^{q} \overline{c}_j + MB - \sum_{j=1}^{q} \overline{c}_j \overline{z}_j =$$

$$\sum_{j=1}^{q} \overline{c}_j + MB - \sum_{j=1}^{q} w_j \overline{z}_j \geq \sum_{j=1}^{q} \overline{c}_j + MB - K = K'.$$

Suppose now that there exists an optimal solution (y^*, c^*) to MaxR-KP with value at least K'. Let us now show that we can assume that such a solution (y^*, c^*) satisfies the following six conditions.

1. $c_j^* = \begin{cases} \overline{c}_j & \text{if} \quad 0 \leq y_j^* < 1 \\ 0 & \text{if} \quad 1 \leq y_j^* \leq a_j. \end{cases}$ this is straightforward.

2. $y_{q+1}^* = 0$.

   If $y_{q+1}^* > 0$ and for some $j \in \{1, \ldots, q\}$, $y_j^* < a_j$, we can construct another solution (y', c') such that $y'_{q+1} = y_{q+1}^* - \epsilon > 0$ and $y'_j = y_j^* + \epsilon < a_j$.

$$\sum_{j=1}^{q} c'_j (1 - y'_j) + M(B - y'_{q+1}) = \sum_{j=1}^{q} c_j^* (1 - y_j^* - \epsilon) + M(B - y_{q+1}^* + \epsilon) >$$

$$\sum_{j=1}^{q} c_j^* (1 - y_j^*) + M(B - y_{q+1}^*)$$

and then y' has a larger regret.

If all solutions (y^*, c^*) to MaxR-KP are such that $y^*_{q+1} > 0$ and $y^*_j = a_j$ for all $j \in \{1, \ldots, q\}$, any other the solution y' such that $y'_j = y^*_j$ for all $j \in \{1, \ldots, q\}$ and $y'_{q+1} = 0$ is feasible because $\sum_{j=1}^{q+1} y'_j = \sum_{j=1}^{q} a_j \geq B$ and y' has a larger regret.

3. Without loss of generality, we may assume that for all $j \in \{1, \ldots, q\}$ such that $y^*_j \geq 1$, $y^*_j = a_j$.

   If solution y^* is such that there exists $k \in \{1, \ldots, q\}$ for which $1 \leq y^*_k < a_k$, then we can construct a solution y' such that $y'_k = a_k$, and for all $j \in \{1, \ldots, q\}$ such that $j \neq k$, $y'_j = y^*_j$. Since $c^*_k(1 - y^*_k) = 0 = c'_k(1 - y'_k)$ the solution y' has the same regret than y^*.

4. There do not exists an optimal solution (y^*, c^*) such that for some $j \in \{1, \ldots, q\}$, $0 < y^*_j < 1$ and for all $k \in \{1, \ldots, q\}$ such that $k \neq j$, $y^*_k = a_k$.

   In order to prove that, we first observe that $\sum_{i=1}^{q} y^*_i \neq B$ because for all $i \in I$, a_i and B are integers. Then if $\sum_{i=1}^{q} y^*_i > B$, let $\epsilon > 0$ be such that $\sum_{i=1}^{q} y^*_i - \epsilon \geq B$ and let y' be such that $y'_j = y^*_j - \epsilon > 0$, and for all $k \neq j$, $y'_k = y^*_k$. Then $c'_j(1 - y'_j) = \bar{c}_j(1 - y'_j) = \bar{c}_j(1 - y^*_j + \epsilon) > \bar{c}_j(1 - y^*_j) = c^*_j(1 - y^*_j)$ and y' has a larger regret.

5. Without loss of generality, we may assume that the solution (y^*, c^*) is such that, there does not exist two variables, say y^*_j and y^*_k such that $0 < y^*_j < 1$ and $0 < y^*_k < 1$.

   If $\bar{c}_j < \bar{c}_k$, let $\epsilon > 0$ and y' be such that

$y'_j = y^*_j + \epsilon < 1$ and $y'_k = y^*_k - \epsilon \geq 0$, then

$$c'_j(1 - y'_j) = \overline{c}_j(1 - y'_j) = \overline{c}_j(1 - y^*_j - \epsilon) = \overline{c}_j(1 - y^*_j) - \overline{c}_j\epsilon$$

and

$$c'_k(1 - y'_k) = \overline{c}_k(1 - y'_k) = \overline{c}_k(1 - y^*_k + \epsilon) = \overline{c}_k(1 - y^*_k) + \overline{c}_k\epsilon$$

thus

$$c'_j(1 - y'_j) + c'_k(1 - y'_k) = \overline{c}_j(1 - y^*_j) - \overline{c}_j\epsilon + \overline{c}_k(1 - y^*_k) + \overline{c}_k\epsilon >$$

$$\overline{c}_j(1 - y^*_j) + \overline{c}_k(1 - y^*_k),$$

hence y' has a larger regret.

If $\overline{c}_j = \overline{c}_k$ suppose that $0 < y^*_j \leq y^*_k < 1$ and let $\epsilon = min\{\mid y^*_j - 0 \mid, |1 - y^*_k|\}$. Construct a solution y' such that $y'_j = y^*_j - \epsilon$, $y'_k = y^*_k + \epsilon$, and for all $i \in \{1, \ldots, q\}$ such that $i \neq j$, and $i \neq k$, $y'_i = y^*_i$. Then

$$c'_j(1 - y'_j) = \overline{c}_j(1 - y'_j) = \overline{c}_j(1 - y^*_j + \epsilon) = \overline{c}_j(1 - y^*_j) + \overline{c}_j\epsilon.$$

For the value of $c'_j(1 - y'_j) + c'_k(1 - y'_k)$ we have two cases.
If $y'_k = y^*_k + \epsilon < 1$

$$c'_k(1 - y'_k) = \overline{c}_k(1 - y'_k) = \overline{c}_k(1 - y^*_k - \epsilon) = \overline{c}_k(1 - y^*_k) - \overline{c}_k\epsilon$$

thus

$$c'_j(1 - y'_j) + c'_k(1 - y'_k) = \overline{c}_j(1 - y^*_j) + \overline{c}_j\epsilon + \overline{c}_k(1 - y^*_k) - \overline{c}_k\epsilon =$$

$$\overline{c}_j(1 - y^*_j) + \overline{c}_k(1 - y^*_k)$$

hence y' has the same regret than y^* and we can replace y^* for y'.
If $y'_k = y^*_k + \epsilon = 1$, $c'_k(1 - y'_k) = 0$, and then

$$c'_j(1 - y'_j) + c'_k(1 - y'_k) = \overline{c}_j(1 - y^*_j) + \overline{c}_j\epsilon + 0 =$$

$$\overline{c}_j(1 - y^*_j) + \overline{c}_j(1 - y^*_k) = \overline{c}_j(1 - y^*_j) + \overline{c}_k(1 - y^*_k) = c^*_j(1 - y^*_j) + c^*_k(1 - y^*_k)$$

thus y' has the same regret than y^* and we can replace y^* for y'.

6. There does not exist two variables, y_j^* and y_k^* such that $0 < y_j^* < 1$ and $1 \leq y_k^* < a_k$.

   Let $\epsilon > 0$ be such that $y_j' = y_j^* - \epsilon > 0$ and $y_k' = y_k^* + \epsilon < a_k$ then
   $0 < c_j^*(1 - y_j^*) = \overline{c}_j(1 - y_j^*) < \overline{c}_j(1 - y_j^* + \epsilon) = \overline{c}_j(1 - y_j') = c_j'(1 - y_j')$
   and $0 = c_k^*(1 - y_k^*) = c_k^*(1 - y_k^* - \epsilon) = c_k'(1 - y_k')$. thus y' has a larger regret.

Given a feasible solution (y^*, c^*) to MaxR-KP with value at least K' let $z \in \{0,1\}^q$ such that $z_j = \begin{cases} 0 & \text{if} \quad y_j^* = 0 \\ 1 & \text{if} \quad y_j^* = a_j \end{cases}$, then

$$K' \leq \sum_{j=1}^{q} c_j^*(1 - y_j^*) + MB = \sum_{j=1}^{q} \overline{c}_j(1 - z_j) + MB = \sum_{j=1}^{q} \overline{c}_j - \sum_{j=1}^{q} \overline{c}_j z_j + MB$$

since $K' = \sum_{j=1}^{q} \overline{c}_j - K + BM$ then $-K \leq -\sum_{j=1}^{q} \overline{c}_j z_j$, and since $\overline{c}_j = w_j$, then $\sum_{j=1}^{q} w_j z_j \leq K$.

Moreover $B \leq \sum_{i=1}^{q+1} y_i = \sum_{i=1}^{q} y_i = \sum_{i=1}^{q} a_j z_j$. Hence the knapsack problem has a solution.

Given any instance of the knapsack problem, the instance of MaxR interval problem can be constructed in polynomial time. We then conclude that the MaxR interval problem is NP-hard. □

4.7 Polynomial cases

In order to identify special cases when the maximum regret and minimax regret problems under polyhedral uncertainty are polynomial, consider the following sets.

$A_1 = \{i \in I : \forall c^s \in D \; c_i^s > 0\}$,
$A_2 = \{i \in I : \forall c^s \in D \; c_i^s < 0\}$,
$A_3 = \{i \in I : \exists c^{s_1}, c^{s_2} \in D$ such that $c_i^{s_1} \leq 0$ and $c_i^{s_2} \geq 0\}$.

Clearly $I = A_1 \cup A_2 \cup A_3$ and $A_i \cap A_j = \emptyset$ for $i \neq j$.

Proposition 4.1 *If D is a nonempty and bounded polyhedron, if $A_3 = \emptyset$ and the polyhedron $P = \{x \in \mathbb{R}^n : Ax \geq b,\ x \geq 0\}$ is an hyperectangle $P = \prod_{i=1}^{n} \left[\underline{y_i}, \overline{y_i}\right]$, then the minimax regret is equal to zero and the coordinates of the minimax regret solution x^* are*

$$\begin{cases} x_i^* = \underline{y_i} & \text{if} \quad i \in A_1 \\ x_i^* = \overline{y_i} & \text{if} \quad i \in A_2. \end{cases}$$

Proof. Under these assumptions, the maximum regret for any $x \in P$ is

$$R_{max}(x) = \max_{c^s \in D} \left\{ \sum_{i \in A_1} c_i^s (x_i - \underline{y_i}) + \sum_{i \in A_2} c_i^s (x_i - \overline{y_i}) \right\} \geq 0.$$

Since x^* is such that $R_{max}(x^*) = 0$ it is a minimax regret solution. □

4.7.1 Polynomial cases under interval uncertainty

In this section, we shall study some cases when the minimax regret and maximum regret problems under interval uncertainty are polynomial.

Consider a nonempty polytope $P = \{x \in \mathbb{R}^n : Ax \geq b,\ x \geq 0\}$ and the hyperectangle $Q = \prod_{i=1}^{n} \left[\underline{y_i}, \overline{y_i}\right]$ where $\overline{y_i}$ and $\underline{y_i}$ are optimal solutions to the problems

$$\begin{aligned} &\text{maximize} \quad y_i \\ &\text{subject to} \quad Ay \geq b \\ &\qquad y \geq 0, \end{aligned}$$

and

$$\begin{aligned} &\text{minimize} \quad y_i \\ &\text{subject to} \quad Ay \geq b \\ &\qquad y \geq 0, \end{aligned}$$

respectively. Hence $P \subseteq Q$, and we have the following polynomial cases.

Proposition 4.2 *If each vertex of* $P = \{x \in \mathbb{R}^n : Ax \geq b,\ x \geq 0\}$ *is a vertex of* $Q = \prod_{i=1}^{n} [\underline{y_i}, \overline{y_i}]$ *and* $D = \prod_{i=1}^{n} [\underline{c_i}, \overline{c_i}]$, *then the maximum regret is equal to*

$$R_{max}(x) = \sum_{i=1}^{n} \overline{c_i}(x_i - \underline{y_i}) \left(\frac{\overline{y_i}}{\overline{y_i} - \underline{y_i}} \right) + \underline{c_i}(x_i - \overline{y_i}) \left(\frac{-\underline{y_i}}{\overline{y_i} - \underline{y_i}} \right) +$$

$$- \min_{y \in P} \sum_{i=1}^{n} \left\{ \underline{c_i} \left(\frac{\overline{y_i} - x_i}{\overline{y_i} - \underline{y_i}} \right) + \overline{c_i} \left(\frac{x_i - \underline{y_i}}{\overline{y_i} - \underline{y_i}} \right) \right\} y_i$$

and the maximum regret and the minimax regret are polynomially solvable.

Proof. In the case when each vertex of $P = \{x \in \mathbb{R}^n : Ax \geq b,\ x \geq 0\}$ is a vertex of $Q = \prod_{i=1}^{n} [\underline{y_i}, \overline{y_i}]$, for a fixed $x \in P$ the component $c_i^s(x_i - y_i)$ of the maximum regret satisfies one of the following conditions:

1. If $y_i = \underline{y_i}$ the component $c_i^s(x_i - y_i)$ of the maximum regret is $c_i^s(x_i - \underline{y_i})$ and that is maximum for $c_i^s = \overline{c_i}$.

2. If $y_i = \overline{y_i}$ the component $c_i^s(x_i - y_i)$ of the maximum regret is $c_i^s(x_i - \overline{y_i})$ and that is maximum for $c_i^s = \underline{c_i}$.

Thus the maximum regret for a fixed $x \in P$ is

$$\max_{y \in P} \sum_{i=1}^{n} \left\{ \overline{c_i}(x_i - \underline{y_i}) \left(\frac{\overline{y_i} - y_i}{\overline{y_i} - \underline{y_i}} \right) + \underline{c_i}(x_i - \overline{y_i}) \left(\frac{y_i - \underline{y_i}}{\overline{y_i} - \underline{y_i}} \right) \right\} =$$

$$\sum_{i=1}^{n} \left\{ \overline{c_i}(x_i - \underline{y_i}) \left(\frac{\overline{y_i}}{\overline{y_i} - \underline{y_i}} \right) + \underline{c_i}(x_i - \overline{y_i}) \left(\frac{-\underline{y_i}}{\overline{y_i} - \underline{y_i}} \right) \right\} +$$

$$\max_{y \in P} \sum_{i=1}^{n} \left\{ \overline{c_i}(x_i - \underline{y_i}) \left(\frac{-y_i}{\overline{y_i} - \underline{y_i}} \right) + \underline{c_i}(x_i - \overline{y_i}) \left(\frac{y_i}{\overline{y_i} - \underline{y_i}} \right) \right\} =$$

$$\sum_{i=1}^{n} \left\{ \overline{c_i}(x_i - \underline{y_i}) \left(\frac{\overline{y_i}}{\overline{y_i} - \underline{y_i}} \right) + \underline{c_i}(x_i - \overline{y_i}) \left(\frac{-\underline{y_i}}{\overline{y_i} - \underline{y_i}} \right) \right\} +$$

$$- \min_{y \in P} \sum_{i=1}^{n} \left\{ \underline{c_i} \left(\frac{\overline{y_i} - x_i}{\overline{y_i} - \underline{y_i}} \right) + \overline{c_i} \left(\frac{x_i - \underline{y_i}}{\overline{y_i} - \underline{y_i}} \right) \right\} y_i,$$

and that can be calculated in polynomial time optimizing over P with cost coefficients equal to

$$\underline{c_i}\left(\frac{\overline{y_i}-x_i}{\overline{y_i}-\underline{y_i}}\right)+\overline{c_i}\left(\frac{x_i-\underline{y_i}}{\overline{y_i}-\underline{y_i}}\right)$$

hence we can also solve the MinMaxR interval problem in polynomial time. □

The following result is a particular case of Proposition 4.2, but it is interesting because implies that for interval UCOP problems the maximum regret for a fixed x can be calculated in polynomial time if and only if the corresponding COP problem is polynomial.

Corollary 4.2 *If all coordinates of each vertex of the polytope*

$$P=\{x\in\mathbb{R}^n : Ax\geq b,\ x\geq 0\}$$

are equal to 0 or 1, and the uncertainty set is

$$D=\prod_{i=1}^{n}\left[\underline{c_i},\overline{c_i}\right],$$

then the maximum regret for a fixed $x\in P$ is equal to

$$R_{max}(x)=\sum_{i=1}^{n}\overline{c_i}x_i-\min_{y\in P}\sum_{i=1}^{n}\left\{\underline{c_i}(1-x_i)+\overline{c_i}x_i\right\}y_i,$$

and the maximum regret and the minimax regret version can be calculated in polynomial time.

Proof. This is a direct consequence of Proposition 4.2 □

We study next the sub-case where the polyhedron P is an hyperectangle. In this case we can give explicitly the value of the minimax regret and the coordinates of the minimax regret solution.

Proposition 4.3 *If the polyhedron $P=\{x\in\mathbb{R}^n : Ax\geq b,\ x\geq 0\}$ is an hyperectangle $P=\prod_{i=1}^{n}\left[\underline{y_i},\overline{y_i}\right]$ and $D=\prod_{i=1}^{n}\left[\underline{c_i},\overline{c_i}\right]$, then the maximum regret of $x\in P$ is*

$$R_{max}(x)=\sum_{i\in A_1}\overline{c_i}(x_i-\underline{y_i})+\sum_{i\in A_2}\underline{c_i}(x_i-\overline{y_i})+ \tag{4.13}$$

$$\sum_{i\in A_3} \frac{-\overline{c_i}\underline{c_i}}{\overline{c_i}-\underline{c_i}}(\overline{y_i}-\underline{y_i}) + \max_{c^s\in D} \sum_{i\in A_3} \left(x_i - \frac{-\underline{c_i}\overline{y_i}+\overline{c_i}\underline{y_i}}{\overline{c_i}-\underline{c_i}} \right) c_i^s,$$

and can be calculated in polynomial time. The minimax regret is equal to

$$\min_{x\in P} R_{max}(x) = \sum_{i\in A_3} \frac{-\overline{c_i}\underline{c_i}}{\overline{c_i}-\underline{c_i}}(\overline{y_i}-\underline{y_i}),$$

and the coordinates of the minimax regret solution x^* *are*

$$\begin{cases} x_i^* = \underline{y_i} & \text{if } \ i \in A_1 = \{i \in I : \underline{c_i} > 0\} \\ x_i^* = \overline{y_i} & \text{if } \ i \in A_2 = \{i \in I : \overline{c_i} < 0\} \\ x_i^* = \frac{-\underline{c_i}\overline{y_i}+\overline{c_i}\underline{y_i}}{\overline{c_i}-\underline{c_i}} & \text{if } \ i \in A_3 = \{i \in I : \underline{c_i} \le 0 \text{ and } \overline{c_i} \ge 0\}. \end{cases}$$

Proof. In the case when the polyhedron $P = \{x \in \mathbb{R}^n : Ax \ge b,\ x \ge 0\}$ is the hyperectangle $P = \prod_{i=1}^n \left[\underline{y_i}, \overline{y_i}\right]$, for a fixed $x \in P$ a component $c_i^s(x_i - y_i)$ of the maximum regret satisfies one of the following conditions:

1. If $i \in A_1$, the component of the maximum regret is $c_i^s(x_i - \underline{y_i})$ and it is maximum for $c_i^s = \overline{c_i}$.

2. If $i \in A_2$ the component of the maximum regret is $c_i^s(x_i - \overline{y_i})$ and it is maximum for $c_i^s = \underline{c_i}$.

3. If $i \in A_3$ and $c_i^s = \overline{c_i}$ the component of the maximum regret is $\overline{c_i}(x_i - y_i)$ and it is maximum for $y_i = \underline{y_i}$.

   If $i \in A_3$ and $c_i^s = \underline{c_i}$ $i \in A_3$ the component of the maximum regret is $\underline{c_i}(x_i - y_i)$ and it is maximum for $y_i = \overline{y_i}$.

Then the maximum regret for a fixed $x \in P$ is

$$R_{max}(x) = \max_{y\in P} \max_{c^s\in D} \sum_{i=1}^n c_i^s(x_i - y_i) = \sum_{i\in A_1} \overline{c_i}(x_i - \underline{y_i}) + \sum_{i\in A_2} \underline{c_i}(x_i - \overline{y_i}) +$$

$$\max_{c^s\in D} \sum_{i\in A_3} \left\{ \overline{c_i}(x_i - \underline{y_i}) \left(\frac{c_i^s - \underline{c_i}}{\overline{c_i}-\underline{c_i}} \right) + \underline{c_i}(x_i - \overline{y_i}) \left(\frac{\overline{c_i} - c_i^s}{\overline{c_i}-\underline{c_i}} \right) \right\} =$$

$$\sum_{i\in A_1} \overline{c_i}(x_i - \underline{y_i}) + \sum_{i\in A_2} \underline{c_i}(x_i - \overline{y_i}) +$$

$$\sum_{i\in A_3}\left\{\overline{c_i}(x_i-\underline{y_i})\left(\frac{-\underline{c_i}}{\overline{c_i}-\underline{c_i}}\right)+\underline{c_i}(x_i-\overline{y_i})\left(\frac{\overline{c_i}}{\overline{c_i}-\underline{c_i}}\right)\right\}+$$

$$\max_{c^s\in D}\sum_{i\in A_3}\left\{\overline{c_i}(x_i-\underline{y_i})\left(\frac{c_i^s}{\overline{c_i}-\underline{c_i}}\right)-\underline{c_i}(x_i-\overline{y_i})\left(\frac{c_i^s}{\overline{c_i}-\underline{c_i}}\right)\right\}=$$

$$\sum_{i\in A_1}\overline{c_i}(x_i-\underline{y_i})+\sum_{i\in A_2}\underline{c_i}(x_i-\overline{y_i})+$$

$$\sum_{i\in A_3}\frac{-\overline{c_i}\underline{c_i}}{\overline{c_i}-\underline{c_i}}(\overline{y_i}-\underline{y_i})+\max_{c^s\in D}\sum_{i\in A_3}\left(x_i-\frac{-\underline{c_i}\overline{y_i}+\overline{c_i}\underline{y_i}}{\overline{c_i}-\underline{c_i}}\right)c_i^s,$$

and that can be calculated in polynomial time optimizing over $D=\prod_{i=1}^{n}\left[\underline{c_i},\overline{c_i}\right]$ with cost coefficients equal to $x_i-\frac{-\underline{c_i}\overline{y_i}+\overline{c_i}\underline{y_i}}{\overline{c_i}-\underline{c_i}}$ if $i\in A_3$ and 0 if $i\notin A_3$. We observe that $\sum_{i\in A_1}\overline{c_i}(x_i-\underline{y_i})\geq 0$, $\sum_{i\in A_2}\underline{c_i}(x_i-\overline{y_i})\geq 0$ and

$$\max_{c^s\in D}\sum_{i\in A_3}\left(x_i-\frac{-\underline{c_i}\overline{y_i}+\overline{c_i}\underline{y_i}}{\overline{c_i}-\underline{c_i}}\right)c_i^s\geq 0.$$

This last inequality holds because for $i\in A_3$ if $x_i-\frac{-\underline{c_i}\overline{y_i}+\overline{c_i}\underline{y_i}}{\overline{c_i}-\underline{c_i}}\geq 0$ the maximum over $c^s\in D$ is reached for $c_i^s=\overline{c_i}>0$ and if $x_i-\frac{-\underline{c_i}\overline{y_i}+\overline{c_i}\underline{y_i}}{\overline{c_i}-\underline{c_i}}<0$ the maximum is reached for $c_i^s=\underline{c_i}<0$.

Since $\frac{-\underline{c_i}\overline{y_i}+\overline{c_i}\underline{y_i}}{\overline{c_i}-\underline{c_i}}$ is a convex combination of $\underline{y_i}$ and $\overline{y_i}$ and $P=\prod_{i=1}^{n}\left[\underline{y_i},\overline{y_i}\right]$, then there exists a $x^*\in P$ such that

$$\begin{cases} x_i^*=\underline{y_i} & \text{if } \ i\in A_1=\{i\in I:\underline{c_i}>0\}\\ x_i^*=\overline{y_i} & \text{if } \ i\in A_2=\{i\in I:\overline{c_i}<0\}\\ x_i^*=\frac{-\underline{c_i}\overline{y_i}+\overline{c_i}\underline{y_i}}{\overline{c_i}-\underline{c_i}} & \text{if } \ i\in A_3=\{i\in I:\underline{c_i}\leq 0 \text{ and } \overline{c_i}\geq 0\},\end{cases}$$

hence the minimax regret in this case is equal to

$$\min_{x\in P}R_{max}(x)=\min_{x\in P}\left\{\sum_{i\in A_1}\overline{c_i}(x_i-\underline{y_i})+\sum_{i\in A_2}\underline{c_i}(x_i-\overline{y_i})\right\}+$$

$$\min_{x\in P}\left\{\sum_{i\in A_3}\frac{-\overline{c_i}\underline{c_i}}{\overline{c_i}-\underline{c_i}}(\overline{y_i}-\underline{y_i})+\max_{c^s\in D}\sum_{i\in A_3}\left(x_i-\frac{-\underline{c_i}\overline{y_i}+\overline{c_i}\underline{y_i}}{\overline{c_i}-\underline{c_i}}\right)c_i^s\right\}=$$

$$\sum_{i\in A_3} \frac{-\overline{c_i}\underline{c_i}}{\overline{c_i}-\underline{c_i}}(\overline{y_i}-\underline{y_i}) = R_{max}(x^*)$$

and can be calculated in polynomial time. □

In Section 4.4, we have shown that an optimal solution x^* to the MinMaxR problem can be located in the interior of P and in such case x^* is not a feasible solution to the minimax regret combinatorial problem.

Clearly, for a fixed $x \in P$ and each vertex v of P

$$\max_{c^s\in D} \sum_{i=1}^{n} c_i^s(x_i - v_i) \leq R_{max}(x)$$

and then optimizing over $D = \prod_{i=1}^{n} \left[\underline{c_i}, \overline{c_i}\right]$ each vertex v of P gives a lower bound for the maximum regret of $x \in P$. Then if $V' = \{y^1, y^2 \dots y^k\} \subset V(P)$ is a finite subset of vertices of P, it holds that

$$\max_{y\in V'} \max_{c^s\in D} \sum_{i=1}^{n} c_i^s(x_i - y_i) \leq R_{max}(x)$$

and this lower bound to the maximum regret of $x \in P$ can be calculated in polynomial time.

Considering the formulation (4.12) of the MinMaxR interval problem and a finite subset of constraints $c^{s_i}(x-y^i) \leq z$, indexed by I, where $c^{s_i} \in \{\underline{c}, \overline{c}\}^n$ and $y^i \in V(P)$, we can construct the following relaxed problem

$$\begin{array}{ll} \min & z \\ \text{s.t.} & Ax \geq b \\ & x \geq 0 \\ & c^{s_i}(x - y^i) \leq z \quad i \in I. \end{array}$$

Let $(\overline{x}, \overline{z})$ a optimal basic feasible solution to the last problem. Since for all $x \in P$

$$\max_{i\in I} c^{s_i}(x - y^i) \leq \max_{y\in V(P)} \max_{c^s\in V(D)} c^s(x - y) = R_{max}(x),$$

thus

$$\overline{z} = \max_{i\in I} c^{s_i}(\overline{x} - y^i) = \min_{x\in P} \max_{i\in I} c^{s_i}(x - y^i) \leq \min_{x\in P} R_{max}(x) \qquad (4.14)$$

and

$$\min_{x\in P} R_{max}(x) \leq R_{max}(\overline{x}).$$

Hence $\overline{z}$ and $R_{max}(\overline{x})$ are lower and upper bounds for the value of the Mini-Max interval problem respectively.

In the following section, we present an algorithm to solve the minimax regret problem under polyhedral uncertainty, our algorithm uses the lower bound of (4.14) at each iteration.

4.8 An exact algorithm

In the case of interval uncertainty, Mausser and Laguna (1998) give an exact algorithm to solve this problem and Mausser and Laguna (1999) present a heuristic to solve the minimax regret problem. In this section, we present an algorithm to find an exact solution to the minimax regret problem under polyhedral uncertainty, that is, a solver to the problem

$$\begin{aligned} \min \quad & z \\ \text{s.t.} \quad & c^s(x-y) \leq z, \quad \forall\, y \in P,\ \forall c^s \in D, \\ & Ax \geq b \\ & x \geq 0, \end{aligned} \tag{4.15}$$

in the case when the uncertainty set D is a polytope of the form, $D = \{c \in \mathbb{R}^n : c^t E \leq e^t, c \geq 0\}$, where $E \in \mathbb{R}^{n\times p}$ and $e \in \mathbb{R}^p$.

The algorithm at each iteration, solves at optimality a relaxed linear program to generate a candidate solution and then it computes a lower bound for the maximum regret of this candidate either to generate a new cut constraint or to conclude that this current solution is optimal for the minimax regret problem. To compute the lower bound for the maximum regret problem we use a branch and bound algorithm called CBA, given by Alarie et al. (2001), which makes use of concavity cuts to solve a disjoint bilinear programming problem. We adapt their implementation to our problem.

The disjoint bilinear programming problem considered by Alarie et al. (2001), is

$$\max_{w,u} \kappa^t w - u^t \mathcal{Q} w + u^t \delta \tag{4.16}$$

$$\begin{aligned} \text{st} \quad & \mathcal{A} w \leq \alpha \\ & u^t \mathcal{B} \leq \beta^t \\ & w \geq 0 \\ & u \geq 0. \end{aligned}$$

where

$$\kappa \in \mathbb{R}^{n_w};\ \alpha \in \mathbb{R}^{n_v};\ \mathcal{A} \in \mathbb{R}^{n_v \times n_w};\ \mathcal{Q} \in \mathbb{R}^{n_u \times n_w}$$

$$\delta \in \mathbb{R}^{n_u}; \beta \in \mathbb{R}^{n_y}; \mathcal{B} \in \mathbb{R}^{n_u \times n_y}.$$

The reduction:

In the case when $D = \{c \in \mathbb{R}^n : c^t E \leq e^t, c \geq 0\}$, where $E \in \mathbb{R}^{n \times p}$ and $e \in \mathbb{R}^p$, the maximum regret of $\overline{x}$ formulated as

$$\max_{c^s, y} \quad c^s(\overline{x} - y) \tag{4.17}$$

$$\text{s.t.} \quad Ay \geq b$$

$$y \geq 0$$

$$c^s \in D$$

can be reduced to the problem (4.16), for $\kappa = 0$, $w = y$, $u = c^s$, $\mathcal{Q} = I_n$ (the identity matrix of size n), $\delta = \overline{x}$, $\mathcal{A} = -A$, $\alpha = -b$, $\mathcal{B} = B$, $\beta = e$, $n_u = n_w = n$, $n_v = m$ and $n_y = p$.

In the case when $D = \prod_{j=1}^{n} \left[\underline{c_j}, \overline{c_j}\right]$ and $\underline{c_j} \leq \overline{c_j}$, without loss of generality, suppose that there does not exist $i \in \{1, \ldots, n\}$ such that $\underline{c_i} = \overline{c_i} = 0$, (in such case we can eliminate the term $c_i^s(\overline{x}_i - y_i)$ and the variables c_i^s and y_i to the problem (4.17)). The problem (4.17) can be reformulated as a particular case of problem (4.16), for $n_u = 2n$, $n_w = n$, $n_v = m$ and $n_y = 2n$, $\kappa = 0$, $w = y$, replacing the n variables c_j^s by $2n$ positive variables u_i, on the following way,

$$c_j^s = \begin{cases} u_j & \text{if } \underline{c}_j \geq 0, \\ -u_j & \text{if } \overline{c}_j \leq 0, \\ u_j - u_{n+j} & \text{if } \underline{c}_j < 0 < \overline{c}_j. \end{cases}$$

The matrix $\mathcal{Q} \in \mathbb{R}^{2n \times n}$ is constructed in such a way that for $j = 1, \ldots, n$ and $i = 1, \ldots, 2n$,

$$\mathcal{Q}_{ij} = \begin{cases} 1 & \text{if } i = j & \text{and } \underline{c}_j \geq 0, \\ -1 & \text{if } i = j & \text{and } \overline{c}_j \leq 0, \\ -1 & \text{if } i = n + j & \text{and } \underline{c}_j < 0 < \overline{c}_j, \\ 0 & \text{if } i = n + j & \text{and } \underline{c}_j \geq 0 \text{ or } \overline{c}_j \leq 0, \\ 0 & \text{if } i \neq j & \text{and } i \neq n + j, \end{cases}$$

and the matrix $\mathcal{B} \in \mathbb{R}^{2n \times 2n}$ is for $j = 1 \ldots 2n$ and $i = 1 \ldots 2n$, such that

$$\mathcal{B}_{ij} = \begin{cases} 1 & \text{if } \; i = j \leq n \quad\quad\quad\quad \text{and} \; (\underline{c}_j \geq 0 \text{ or } \underline{c}_j < 0 < \overline{c}_j) \\ -1 & \text{if } \; i = j \leq n \quad\quad\quad\quad \text{and} \; \overline{c}_j \leq 0, \\ 1 & \text{if } \; i = j > n \quad\quad\quad\quad \text{and} \; \underline{c}_{j-n} < 0 < \overline{c}_{j-n}, \\ 0 & \text{if } \; i = j > n \quad\quad\quad\quad \text{and} \; \underline{c}_{j-n} \geq 0 \text{ or } \overline{c}_{j-n} \leq 0, \\ 1 & \text{if } \; i = j - n, \; j \geq n+1 \;\; \text{and} \; \overline{c}_j \leq 0 \\ -1 & \text{if } \; i = j - n \; j \geq n+1 \;\; \text{and} \; \underline{c}_j \geq 0 \\ 0 & \quad\quad \text{otherwise} \end{cases}$$

$\delta = \overline{x}$, $\mathcal{A} = -A$, $\alpha = -b$, for $j = 1, \dots, n$, $\beta_j = \overline{c}_j$ and for $j = n+1 \dots 2n$ $\beta_j = -\underline{c}_{j-n}$. In both of the cases, we can use this branch and bound algorithm to solve the maximum regret problem.

We can see that in general it is impossible to generate and store the constraints $c^s(x - y) \leq z$ for all $c^s \in V(D)$ and for all vertex y of the polyhedron P, because the number of variables and vertices can be very large. Instead of dealing with all these constraints, we shall present an algorithm that combines the CBA algorithm with a relaxation procedure. The relaxation procedure is a well known technique to solve mixed integer problems of big size (see e.g. Minoux (1983)). We consider a finite subset of constraints of type $c^{s_i}(x - y^i) \leq z$ indexed by the set J, where $c^{s_i} \in V(D)$ and $y^i \in V(P)$ constructed in the following way. In the case of polyhedral but non interval uncertainty, we first fix the value of $|J|$, and then we choose randomly $|J|$ vectors $v^i \in \{-1, 1\}^{n_w}$. Then we solve each corresponding linear programming problem

$$\begin{aligned} &\min v^i c^s \\ \text{s.t.} \quad & c^s \in D \end{aligned} \tag{4.18}$$

to obtain solutions $c^{s_i} \in V(D)$. In the case of interval uncertainty, we choose randomly $|J|$ vectors $v^i \in \{0, 1\}^{n_w}$. If $v^i_j = 1$, we fix $c^{s_i}_j = \overline{c}_j$ and if $v^i_j = 0$, we set $c^{s_i}_j = \underline{c}_j$. We solve the corresponding linear programming problems

$$\begin{aligned} &\min c^{s_i} x \\ \text{s.t.} \quad & Ax \geq b \\ & x \geq 0 \end{aligned} \tag{4.19}$$

to obtain optimal solutions $y^i \in V(P)$ to each one of these problems and then we construct a subset of constraints $c^{s_i}(x - y^i) \leq z$, indexed by J. We form the relaxed problem

$$\begin{aligned} & \min z \qquad (4.20) \\ \text{s.t.} \quad & Ax \geq b \\ & x \geq 0 \\ & c^{s_i}(x - y^i) \leq z, \quad i \in J \end{aligned}$$

which we solve to optimality, finding an optimal basic feasible solution $(\overline{x}, \overline{z})$. Then we construct the maximum regret problem (4.17) associated to $\overline{x}$

$$\begin{aligned} \max_{c^s, y} \quad & c^s(\overline{x} - y) \\ \text{s.t.} \quad & Ay \geq b \\ & y \geq 0 \\ & c^s \in D. \end{aligned}$$

To solve the problem (4.17) is the most time consuming part of the algorithm, thus in order to accelerate it, we do not ask to solve the problem (4.17) to optimality at each iteration. We fix a parameter $\rho = 0.0001$. If CBA finds a current solution (c^{s_k}, y^k) such that the value $c^{s_k}(\overline{x} - y^k)$ is bigger that the value of the current relaxed problem $\overline{z}$ plus ρ one cut is found and CBA is stopped. We compare the value $c^{s_k}(\overline{x} - y^k)$ with $\overline{z}$ and we have two possibilities.

1. If $c^{s_k}(\overline{x} - y^k) \leq \overline{z}$ then $(\overline{x}, \overline{z})$ is a feasible solution to the original problem (4.15). Any other feasible solution (x, z) to the problem (4.15) is also feasible for the relaxed problem (4.20). Therefore by optimality of $(\overline{x}, \overline{z})$ for the problem (4.20) we have $\overline{z} \leq z$. Then $(\overline{x}, \overline{z})$ is an optimal solution to the original problem (4.15) and we can terminate the algorithm.

2. If $c^{s_k}(\overline{x} - y^k) > \overline{z}$, then $(\overline{x}, \overline{z})$ is infeasible for the problem (4.15) and we have found a violated constraint. Then, we add the constraint $c^{s_k}(x - y^k) \leq z$ to the relaxed problem (4.20) and continue similarly.

We observe that this algorithm terminates after a finite number of iterations because the maximum regret problem yields always a solution (c^{s_k}, y^k) such that $y^k \in V(P)$ and $c^{s_k} \in V(D)$. The different steps of the algorithm are as follows.

Algorithm to solve the MinMaxR polyhedral problem.

Input: P: polytope of constraints; D: the uncertainty set of coefficients, both of them of dimension n.

Output an optimal solution (x^*, z^*) to the minimax regret problem.

Data (polyhedral uncertainty case) A : a $n \times m$ matrix; E: a $n \times p$ matrix; b: a m dimensional vector; e: a p dimensional vector.

Data (interval uncertainty case) A : a $n \times m$ matrix; b: a m dimensional vector; $\underline{c}$: a n dimensional vector; $\overline{c}$: a n dimensional vector.

1. fix the value $|J|$.
2. fix the value $\rho = 0.001$.
3. in the case of non interval uncertainty choose vectors $v^1 \in \{-1\}^{n_w}$ and $v^2 \in \{1\}^{n_w}$. Otherwise go to 6.
4. construct the corresponding problem (4.18).
5. find $c^{s_i} \in V(D)$ solving the problem (4.18). Go to 7.
6. in the case of interval uncertainty choose randomly $|J|$ vectors $v^i \in \{0,1\}^{n_w}$. If $v^i_j = 1$ $c^{s_i}_j = \overline{c}_j$ and if $v^i_j = 0$ $c^{s_i}_j = \underline{c}_j$ and choose vectors $v^{|J|+1} \in \{-1\}^{n_w}$ and $v^{|J|+2} \in \{1\}^{n_w}$ and
7. construct the corresponding problem (4.19).
8. find $y^i \in V(P)$ solving the problem (4.19).
9. construct the problem (4.20).
10. find an optimal solution $(\overline{x}, \overline{z})$ to problem (4.20).
11. construct the maximum regret problem (4.17) associated to $\overline{x}$.
12. reduce the problem (4.17) to the problem (4.16)
13. run algorithm CBA, if CBA finds a current solution (c^{s_k}, y^k) such that $c^{s_k}(\overline{x} - y^k) > \overline{z} + \rho$, stop CBA.
14. if $c^{s_k}(\overline{x} - y^k) > \overline{z}$, add the cut $c^{s_k}(\overline{x} - y^k) \leq \overline{z}$ to (4.20) and go to 7.
15. if $c^{s_k}(\overline{x} - y^k) \leq \overline{z}$, we set $\overline{x} = x^*$, $\overline{z} = z^*$ and stop (x^* minimizes the maximum regret).
16. **Return** an optimal solution (x^*, z^*).

4.8.1 Numerical results

In this section, the algorithm has been tested in the case of polyhedral and interval uncertainty. Its performance is discussed on randomly generated instances. For the polyhedral uncertainty case the computational experiments were made on a Pentium II 400MHz station under Linux 2.2.16-SMP with 384 MB of RAM. In the interval uncertainty case, the experiments were made on a Pentium III 1 GHz station under Linux 2.4.20-60GB-SMP with 1 GB of RAM. The algorithm was coded in C (compiler gcc) and uses the CPLEX 7.0 library to solve the linear programs.

The mean value μ and the standard deviation σ of the computing time in seconds and of the number of iterations of 10 randomly generated problems of density parameter Δ are given in the columns labeled *time* and *Iter* respectively.

Polyhedral uncertainty

In Tables 4.2 to 4.5, the coefficient n_w is the dimension of the space, n_v the number of constraints defining the polytope P and n_y is the number of columns of the matrix $\mathcal{B}$. The number of initial constraints added at the first iteration is 2, we choose $v^1 = \{-1\}^{n_w}$ and $v^2 = \{1\}^{n_w}$. In this case, the problems were generated using a problem generator similar to the one proposed by Audet et al. (1999) and Alarie et al. (2001), i.e., to generate a minimax regret problem

$$\begin{aligned} \max_{c,y} \sum_{j=1}^{p} c_j(x_j - y_j) \\ \text{st } \; Ay \geq b \\ c^t E \leq e^t \\ y \geq 0 \end{aligned}$$

the elements of the vectors b and e are randomly chosen between -10 and 10. For each element of the matrix A and E a random number between 0 and 1 is generated. If the number is less than Δ, then the element is randomly chosen between -20 and 20, otherwise it is fixed to 0. Some entries are added to the matrix A in order to ensure that there is no empty line or column. Then, the additional constraints $\mathbf{1}^t x \leq n_w$ and $c^t\mathbf{1} \leq n_y$ ensure that the polyhedrons $P = \{x \in \mathbb{R}^n : Ax \geq b,\ x \geq 0\}$ and $D = \{c \in \mathbb{R}^n : c^t E \leq e^t, c \geq 0\}$ are bounded.

Table 4.2 presents the behavior of the algorithm on small instances. For larger problems, Tables 4.3 to 4.5 confirm that the difficulty increases with the

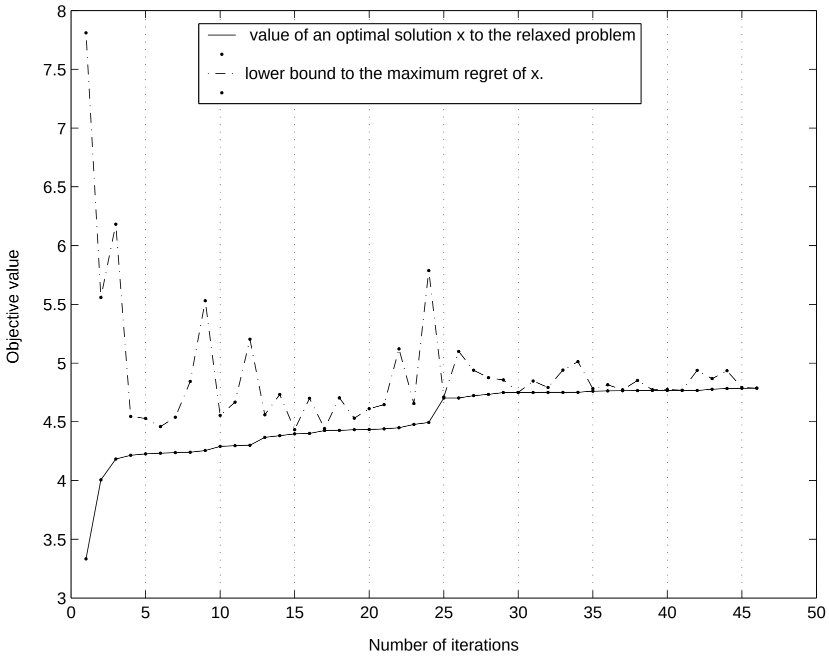

Figure 4.2: Convergence of the algorithm for $n_w = 50$ and matrix density 0.5

dim.	$n_w = 10$ $n_v = 20$ $n_y = 20$		$n_w = 20$ $n_v = 40$ $n_y = 40$	
Δ	time	Iter	time	Iter
0.05 μ	0.82	1.22	38.67	6.00
σ	0.85	0.67	69.17	9.15
0.1 μ	1.30	2.22	207.79	11.30
σ	1.53	0.97	396.99	8.76
0.3 μ	13.27	5.90	16.75	16.00
σ	23.05	3.03	13.56	7.35
0.5 μ	81.42	8.30	28.57	15.30
σ	114.34	4.00	32.62	9.24
0.7 μ	74.09	8.00	31.15	16.50
σ	191.04	3.68	39.99	9.63
0.9 μ	227.51	8.30	39.72	16.80
σ	227.94	2.67	28.45	7.10

Table 4.2: Polyhedral uncertainty. Computing time and number of iterations for $n_w = 10, 20$.

dim.		$n_w = 40$ $n_v = 80$ $n_y = 80$		$n_w = 50$ $n_v = 100$ $n_y = 100$	
Δ		time	Iter	time	Iter
0.05	μ	14.76	14.00	41.56	27.40
	σ	20.73	10.95	55.95	21.58
0.1	μ	21.08	18.70	91.70	28.60
	σ	8.57	5.89	140.37	14.81
0.3	μ	190.18	28.60	221.85	30.30
	σ	286.12	18.41	251.18	16.12
0.5	μ	275.21	29.30	345.37	26.20
	σ	213.17	12.70	217.15	9.02
0.7	μ	176.00	24.50	600.48	32.70
	σ	107.44	9.59	399.16	16.19
0.9	μ	187.40	20.10	441.84	27.90
	σ	138.79	10.27	270.38	11.34

Table 4.3: Polyhedral uncertainty. Computing time and number of iterations for $n_w = 40, 50$.

dimension n_w and with the density Δ. Figure (4.2) shows the convergence of the algorithm for a problem with $n_w = 50$, 100 constraints and matrix density equal to 0.5.

Interval uncertainty

In Tables 4.6 to 4.8, the algorithm is tested in the case of interval uncertainty. Its performance is discussed on randomly generated instances. The coefficient N gives the instance dimensions, $N = 3n_w$ where n_w is the dimension of the space and n_v is the number of constraints defining the polyhedron P. The number of initial constraints added to the departure are given in the row labeled *inic. const.* In this part, to generate a maximum regret problem we consider the model

$$\max_{c,y} \sum_{j=1}^{p} c_j(x_j - y_j)$$

$$\text{st} \quad Ay \geq b$$

$$y \geq 0$$

$$c_j \in \left[\underline{c}_j, \overline{c}_j\right],$$

dim.	$n_w = 60$ $n_v = 120$ $n_y = 120$		$n_w = 70$ $n_v = 140$ $n_y = 140$	
Δ	time	Iter	time	Iter
0.05 μ	40.34	34.30	58.13	41.20
σ	21.66	13.91	27.73	15.91
0.1 μ	421.79	47.10	960.06	64.30
σ	592.03	26.98	968.48	30.27
0.3 μ	517.10	37.80	1513.57	51.70
σ	656.39	20.82	906.27	20.49
0.5 μ	737.08	30.30	1274.54	31.40
σ	441.31	9.04	1014.09	12.21
0.7 μ	883.37	30.00	2027.39	36.90
σ	834.80	16.34	923.44	12.03
0.9 μ	1295.93	32.20	2460.89	37.00
σ	715.86	14.14	671.99	12.27

Table 4.4: Polyhedral uncertainty. Computing time and number of iterations for $n_w = 60, 70$.

dim.	$n_w = 100$ $n_v = 200$ $n_y = 200$		$n_w = 200$ $n_v = 400$ $n_y = 400$	
Δ	time	Iter	time	Iter
0.05 μ	685.00	66.64	15176.68	147.20
σ	1150.79	42.04	6412.55	40.34
0.1 μ	1685.31	62.80	27586.68	89.80
σ	1665.69	33.04	15662.59	29.59
0.3 μ	3645.35	50.90	47802.80	69.00
σ	4305.42	19.60	20362.99	22.01
0.5 μ	3138.10	36.80	47551.20	60.50
σ	1784.10	11.05	27097.19	21.04
0.7 μ	3724.14	34.00	5846.09	55.00
σ	2205.15	13.41	26.34	0.00
0.9 μ	6594.42	36.70	6054.65	30.00
σ	5180.63	14.61	11.59	0.00

Table 4.5: Polyhedral uncertainty. Computing time and number of iterations for $n_w = 100, 200$.

dim.	$N = 60 \quad n_w = 20 \quad n_v = 20$		$N = 90 \quad n_w = 30 \quad n_v = 30$	
inic. const.	$3n_w + 2$		$3n_w + 2$	
Δ	time	Iter	time	Iter
0.1 μ	4.42s	8.30	10.15s	12.50
σ	2.92s	6.18	4.13s	11.59
0.3 μ	7.97s	13.60	10.93s	9.70
σ	9.29s	8.76	3.19s	5.77
0.5 μ	41.33s	12.40	14.32s	10.90
σ	112.49s	7.18	5.79s	10.38
0.9 μ	6.38s	8.30	17.98s	9.80
σ	2.56s	6.24	7.09s	5.33

Table 4.6: Interval uncertainty. Computing time and number of iterations for $n_w = 20, 30$

where the elements of the vectors b, $\underline{c}$ and $\overline{c}$ are randomly chosen between -10 and 10 and then if there exists a j for which $\underline{c}_j > \overline{c}_j$ we change the sign of these elements in order to represent an real interval. For each element of the matrix A a random number between 0 and 1 is generated. If the number is less than Δ, then the element is randomly chosen between -20 and 20, otherwise it is fixed to 0. Some entries are added to the matrix A in order to ensure that there are no empty lines or columns. Then, the additional constraint $\mathbf{1}^t x \leq n_w$ assure that the polyhedron $P = \{x \in \mathbb{R}^n : Ax \geq b, \ x \geq 0\}$ is bounded.

In order to avoid the value $\overline{z} = 0$ at the first iteration, in these experiments, we add the constraints $c^{\underline{s}}(x - x^{*\underline{s}}) \leq z$ and $c^{\overline{s}}(x - x^{*\overline{s}}) \leq z$ to the relaxed problem (4.20).

Computing times in seconds are given in the columns labeled *time* and the number of iterations appears in the columns *Iter*.

Table 4.6 presents the behavior of the algorithm on small instances. For larger problems Tables 4.7 and 4.8 confirm that the difficulty increases with the dimension n_w and with the density Δ. In Table 4.6 to 4.7, we begin always with $3n_w + 2$ initial constraints. The above numerical results show that our algorithm is quite efficient for instances where $n_w \leq 200$. In the following chapter we shall present a preprocessing procedure which reduces considerably the performance of our algorithm in the case of interval uncertainty.

dim.	$N = 210 \quad n_w = 70 \quad n_v = 70$		$N = 300 \quad n_w = 100 \quad n_v = 100$	
inic. const.	$3n_w + 2$		$3n_w + 2$	
Δ	time	Iter	time	Iter
0.5 μ	94.40s	30.20	216.96s	16.10
σ	31.88s	25.80	25.36s	6.92
0.10 μ	90.27s	21.00	264.71s	29.20
σ	24.03s	18.01	72.57s	24.54
0.30 μ	154.51s	38.10	417.02s	38.80
σ	72.53s	37.54	173.29s	27.48
0.50 μ	154.90s	27.00	704.78s	61.40
σ	49.11s	18.62	387.63s	33.14
0.70 μ	209.57s	30.90	472.23s	22.10
σ	134.33s	28.76	130.15s	19.01
0.90 μ	173.53s	14.30	593.56s	27.60
σ	38.61s	7.17	242.26s	20.48

Table 4.7: Interval uncertainty. Computing time and number of iterations for $n_w = 70, 100$

dim.	$N = 600 \quad n_w = 200 \quad n_v = 200$	
inic. const.	$3n_w + 2$	
Δ	time	Iter
0.05 μ	2299.26s	80.10
σ	1047.16s	86.78
0.10 μ	3109.35s	93.00
σ	1385.90s	77.10
0.30 μ	3662.99s	44.33
σ	977.01s	32.22

Table 4.8: Interval uncertainty. Computing time and number of iterations for $n_w = 200$

Chapter 5

Robustness in linear programming and the center location problem

5.1 Introduction

In Chapter 4 we have shown that the minimax regret linear programming problem under uncertainty in the objective function coefficients is a convex piecewise linear optimization problem over a polytope. Since in location theory such kinds of models are frequently used, we present the relations between the 1-center problem in location theory and the minimax regret linear programming problem under interval uncertainty in the objective function coefficients. Such relations allow us to describe the underlying geometry of this last problem and to prove that we can eliminate the 0-persistent variables. Finally we give sufficient conditions for a variable to be 0-persistent and we test these conditions in randomly generated instances.

Given a set $\mathcal{A} = \{a^1, \ldots, a^m\}$ of m points or facilities in $\mathbb{R}^n$, the *1-center problem* in $\mathbb{R}^n$ under a norm or a gauge γ consists in looking for a point $x \in \mathbb{R}^n$ to minimize the maximum "distance" (symmetry is not necessarily required) defined by the given norm or gauge, to each of the given points. The "distance" function verifies the properties:

1. $\gamma(x-y) \geq 0 \ \forall x, y \in \mathbb{R}^n$.
2. $\gamma(x-y) = 0 \ \Leftrightarrow x = y$.
3. $\gamma(x-y) \leq \gamma(x-z) + \gamma(z-y) \ \forall x, y, z \in \mathbb{R}^n$.

The minimax objective function corresponds among others, to location problems of emergency services. This problem can be formulated as

$$\min_{x\in\mathbb{R}^n}\max_{a_j\in\mathcal{A}}\{\gamma(x-a_j)\}.$$

One of the goals of this chapter is to describe the underlying geometry of the minimax regret linear programming problem when the objective function coefficients are subject to interval uncertainty and to relate that problem with the 1-center problem. We shall generalize some results in location theory and in the case when the polytope is full dimensional and all the constraints are non redundant, we shall give conditions under which some variables can be eliminated from the problem.

5.2 Models and notations

In this chapter we shall assume that D is a Cartesian product of intervals, i.e., $D=\prod_{i=1}^{n}\left[\underline{c_i},\overline{c_i}\right]$ and $\underline{c_i}<\overline{c_i}$. We shall consider the following formulation to the minimax regret interval problem (4.12) established in Chapter 4,

MinMaxR interval problem

$$\begin{array}{ll}\min & z\\ \text{s.t.} & c^s(x-y)\le z,\quad \forall\, y\in V(P),\quad \forall c^s=(c^s_1,\dots c^s_j,\dots,c^s_n),\\ & \qquad\qquad\qquad\qquad\qquad\quad \text{with } c^s_j\in\{\underline{c}_j,\overline{c}_j\},\ j=1\dots n,\\ & Ax\ge b,\\ & x\ge 0.\end{array}$$

and the following reformulation, consequence of Corollary 4.1

$$\begin{array}{rc}& \min\ z\\ \text{s.t.} & c^s x-\min_{y\in P}c^s y\le z\ \ \forall c^s_j\in\{\underline{c_j},\overline{c_j}\},\ j=1,\dots,n,\\ & Ax\ge b,\\ & x\ge 0.\end{array}\tag{5.1}$$

Averbakh and Lebedev (2005) proves that the minimax regret linear programming problem with interval uncertain objective function coefficients is strongly NP-hard. However, when the dimension n is fixed the MinMaxR interval problem becomes polynomial. Indeed, the above linear problem (4.12) contains $n+1$ variables $x_1,\dots,x_n$ and z. Further, its number of constraints may be exponential but they can be separated in polynomial time. More precisely, finding an inequality $c^s x - z \leq c^s y$, that is violated by a current solution $(\overline{x}, \overline{z})$ amounts to solve the following 2^n linear programming problems:

$$\begin{aligned} z^* = \text{minimize} \ \ & cy \\ \text{subject to} \ \ & Ay \geq b \\ & x \geq 0, \end{aligned}$$

for $c \in \prod_{i=1}^n \{\underline{c_i}, \overline{c_i}\}$ and check each time whether $z^* < c\overline{x} - \overline{z}$.

In this chapter we shall also consider the following relaxation of the MinMaxR interval problem:

Relaxed MinMaxR interval problem

$$\begin{array}{lll} \min & z & \\ \text{s.t.} & c^s(x-y) \leq z & \forall\, y \in V(P) \\ & \forall c^s = (c^s_1, \ldots c^s_j, \ldots, c^s_n), & \text{with } c^s_j \in \{\underline{c}_j, \overline{c}_j\},\ j = 1 \ldots n. \end{array} \tag{5.2}$$

Let $\mathcal{A} = \{a^1, \ldots, a^m\}$ be a set of m points in $\mathbb{R}^n$ and let γ be a norm or a gauge. The 1-center problem can be reformulated as

1-center problem

$$\begin{aligned} \text{minimize} \quad & z \\ \text{subject to} \quad & \gamma(x - a_j) \leq z, \quad j = 1, \ldots, m, \\ & x \in \mathbb{R}^n. \end{aligned} \tag{5.3}$$

Introducing a polyhedral feasibility region $P = \{x \in \mathbb{R}^n : Ax \geq b,\ x \geq 0\}$, the 1-center problem under γ with the additional restriction that $x \in P$ can be formulated as the following linear program

Restricted 1-center problem

$$\begin{aligned} \min \quad & z \\ \text{subject to} \quad & \gamma(x - a_j) \leq z, \quad j = 1, \ldots, m, \\ & Ax \geq b, \\ & x \geq 0. \end{aligned} \tag{5.4}$$

5.3 Useful properties

In this section we present results concerning gauges, norms and convex optimization problems. Those results have been used to derive properties of location problems such as the continuous 1-center. They will also be used in the sequel of that chapter, namely to show links between the MinMaxR interval problem and the continuous 1-center.

Consider a set of k different linear functions g_i, $i = 1, 2, \ldots, k$, and let $g(x) = max\{g_1(x), g_2(x), \ldots g_k(x)\}$. The problem

$$\textbf{OL} \qquad \min_{x \in \mathbb{R}^n} g(x),$$

is a piecewise linear and convex optimization problem, that can be solved by linear programming methods. The OL problem with the additional restriction that $x \in P$, where $P = \{x \in \mathbb{R}^n : Ax \geq b,\ x \geq 0\}$, can also be formulated as the following linear program:

$$\begin{array}{lll} \textbf{ROL} & \min & z \\ & \text{s.t.} & g_i(x) \leq z \quad for\ i = 1, 2, \ldots k \\ & & Ax \geq b \\ & & x \geq 0. \end{array}$$

We denote by ∂P and $int(P)$, the boundary and the interior of P respectively, by $r^* = \min_{x \in \mathbb{R}^n} g(x)$ and by $r^*_P = \min_{x \in P} g(x)$ the optimal values of OL and ROL. Further,

$$\mathcal{X}^* = \{x \in \mathbb{R}^n : g(x) = r^*\}$$

$$\mathcal{X}^*_P = \{x \in \mathbb{R}^n : x \in P \text{ and } g(x) = r^*_P\}.$$

The following theorem is of interest, see e.g. Nickel and Schobel (1999).

Theorem 5.1 *If $\mathcal{X}^* \cap P = \emptyset$ then $\mathcal{X}^*_P \subseteq \partial P$. Moreover, for all $x \in \mathcal{X}^*_P$ there exists $y \in \mathcal{X}^*$ such that $\lambda x + (1 - \lambda)y \in \mathcal{X}^*_P$ if and only if $\lambda = 1$.*

Nickel (1998) observes that this result can be interpreted as a visibility property similar to one mentioned by Hansen et al. (1982). In order to put this more precisely we give the following concept of visibility developed by Goldman (1963). A point $x_2 \in P$ is said to be *visible from a point* $x_1 \notin P$ if the straight line segment connecting the two points contains no point of P except x_2. A point $x_2 \in P$ is said to be *visible from a set* C such that $C \cap P = \emptyset$ if x_2 is visible from some point in C, see Figure 5.1.

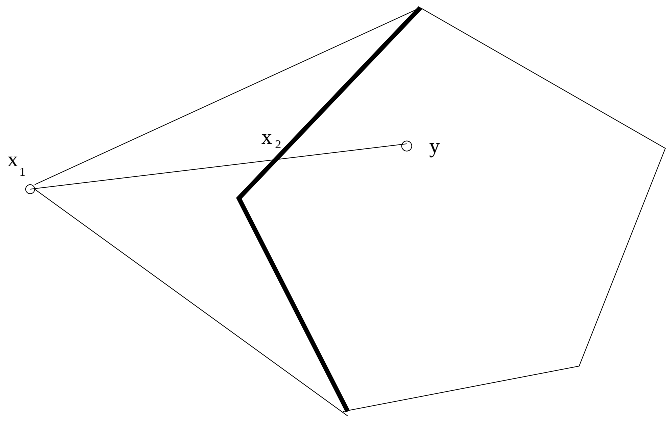

Figure 5.1: Visibility region from the point x_1

Then, the last theorem can be interpreted as follows. If there does not exist an optimal solution to the unrestricted problem OL that is optimal for ROL then all the optimal solutions of ROL are visible from $\mathcal{X}^*$.

Since the MinMaxR interval problem has the form of a ROL problem and Relaxed MinMaxR interval problem has the form of an OL problem, we have the following two corollaries:

Corollary 5.1 *If each optimal solution to the Relaxed MinMaxR interval problem is infeasible for the MinMaxR interval problem then the set of optimal solutions to the MinMaxR interval problem lies on the boundary of P. Moreover under these conditions each optimal solution to the MinMaxR interval is visible from the set of optimal solutions to the Relaxed MinMaxR interval problem.*

Corollary 5.2 *Suppose that there exists an optimal solution x^* to the MinMaxR interval problem such that $x^* \in int(P)$ then x^* is also an optimal solution of Relaxed MinMaxR interval.*

5.3.1 Polyhedral gauges and block norms

Let B_0 be a compact convex set in $\mathbb{R}^n$ containing the origin in its interior and let $x \in \mathbb{R}^n$. The *gauge* of x with respect to B_0 is defined as

$$\gamma(x) := inf\{\lambda > 0 : x \in \lambda \mathsf{B}_0\}.$$

The distance from x to y (symmetry is not necessarily required) is defined by $d(x, y) := \gamma(y - x)$. In the case when B_0 is a polytope with extreme points $V(\mathsf{B}_0) = \{\check{v}_1, \ldots, \check{v}_p\}$, we can write $\gamma(x)$ as

$$\gamma(x) = \min\{\sum_{j=1}^{p} \lambda_j : \; x = \sum_{j=1}^{p} \lambda_j \check{v}_j, \lambda_j \geq 0\}$$

and we shall call *n-dimensional diamond of radius* r_0 with center on the point a, the set of points $\mathsf{B}(a, r_0) = \{x \in \mathbb{R}^n : \gamma(x-a) \leq r_0\}$.

As $\gamma(x)$ is a convex function, if B_0 is symmetric with respect to the origin, γ defines a norm called *block norm* denoted by $||.||$ and B_0 is the corresponding unit ball, i.e.

$$\mathsf{B}_0 = \{x \in \mathbb{R}^n : \gamma(x) \leq 1\}.$$

5.3.2 Some properties of $\mathbb{R}^n$ under block norms and gauges

Consider the following property presented in Wendell and Hurter (1973).

Property 5.1 *Let $||.||$ be a given norm, and let $\{a^1, a^2, \ldots, a^m\}$ be an arbitrary finite set of points in $\mathbb{R}^n$. Let L be the smallest affine set containing the points $a^1, a^2, \ldots, a^m$. Let $x_1 \in \mathbb{R}^n \setminus L$. Then, there exists a point $x_0 \in L$ such that $||x_0 - a^k|| \leq ||x_1 - a^k||$ for $k = 1, \ldots, m$.*

Wendell and Hurter (1973) show that Property 5.1 guarantees that the Kuhn's convex-hull property is true for the 1-center problem under any norm. Let $Conv\{a^1, \ldots, a^m\}$ denote the convex hull of the points $a^1, \ldots, a^m$.

Theorem 5.2 *(Wendell and Hurter (1973)) Assume that for a norm $||.||$, Property 5.1 holds. Then there exists an optimal solution x^* of the 1-center problem*

$$\begin{aligned} &\textit{minimize} \quad z \\ &\textit{subject to} \;\; ||x - a^i|| \leq z, \qquad i = 1, \ldots, m, \\ &x = (x_1, \ldots, x_n) \in \mathbb{R}^n, \end{aligned}$$

such that $x^* \in Conv\{a^1, \ldots, a^m\}$.

Wendell and Hurter (1973) also prove that Property 5.1 holds true if the norm is Euclidean or if $dim(L) \leq 1$. Further, Hurter et al. (1975) show that, in general, Property 5.1 does not need to hold.

In order to show that, for the l_1 norm in $\mathbb{R}^n$, denoted by $||.||_1$, neither the Property 5.1 nor the Theorem 5.2 holds, consider the following example.

Example 5.1 *In* $\mathbb{R}^3$ *let* $a^1 = (10,0,0)$ $a^2 = (0,10,0)$ *and* $a^3 = (0,0,10)$. *For* $\overline{x} = (0,0,0)$ *and for each* a^i, $||\overline{x} - a^i||_1 = \sum_{j=1}^{3} |\overline{x}_j - a^i_j| = 10$. *The convex hull of these points is* $Conv\{a^1, a^2, a^3\} = \{x \in \mathbb{R}^3 : x_1 + x_2 + x_3 = 10,\ x_i \geq 0\}$ *and the smallest affine space containing the points* a^1, a^2, a^3, *is* $L = \{x \in \mathbb{R}^3 : x_1 + x_2 + x_3 = 10\}$. *If we compute an optimal solution to*

$$\begin{aligned} &\textit{minimize} \quad z \\ &\textit{subject to} \ \ ||x - a^i||_1 \leq z, \qquad i = 1, \ldots, 3, \\ &\qquad x_1 + x_2 + x_3 = 10, \\ &\qquad x_i \geq 0, \end{aligned}$$

we obtain the point $(\frac{10}{3}, \frac{10}{3}, \frac{10}{3})$ *whose objective function value is* $\frac{40}{3}$. *Hence neither Theorem 5.2 nor Property 5.1 holds.*

However, we can prove that there exists a class of polyhedral gauges in $\mathbb{R}^n$ that satisfies a restriction of Property 5.1, that we shall call Property 5.2, but first we introduce the following definitions.

Let $I' \subseteq I = \{1, \ldots, n\}$ be a subset of indices. An affine set Y is of *type* $H(I')$, if $Y = \{x \in \mathbb{R}^n : x_i = k_i \ if \ i \in I'\}$, where $k_i \in \mathbb{R}$ is a constant value associated to $i \in I'$. Hence, the *projection* of a point $y = (y_1, \ldots, y_n) \in \mathbb{R}^n$ onto Y is $Proj_Y(y) = \{x \in Y : x_j = y_j \ if \ j \notin I' \ and \ x_j = k_j \ if \ j \in I'\}$.

Let $\{e_1, e_2, \ldots e_n\}$ be the canonical basis of $\mathbb{R}^n$. Let B_0 be an non necessarily symmetric but convex polytope in $\mathbb{R}^n$ containing the origin in its interior and defined by the following set of extreme points, $V(\mathsf{B}_0) = \{\check{v}^1, \ldots, \check{v}^{2n}\}$, where $\check{v}^{2i-1} = \overline{w}_i e_i$ and $\check{v}^{2i} = \underline{w}_i e_i$, for all $i = 1 \ldots n$ and suppose that $\underline{w}_i < 0 < \overline{w}_i$, for all $i = 1 \ldots n$. Since B_0 contains the origin in its interior, we define the $[\underline{w}, \overline{w}]$*-polyhedral gauge* as

$$\gamma(x) = \min\{\sum_{j=1}^{2n} \lambda_j : \ x = \sum_{j=1}^{2n} \lambda_j \check{v}^j, \lambda_j \geq 0\},$$

$$\text{and } \gamma(x - y) = \sum_{i=1}^{n} \max_{\nu_i \in \{\frac{1}{\underline{w}_i}, \frac{1}{\overline{w}_i}\}} \nu_i (x_i - y_i).$$

Property 5.2 *Let* γ *be a given polyhedral gauge, and let* $\{a^1, \ldots, a^m\}$ *be a given set of points in* $\mathbb{R}^n$ *such that there exists a subset* $I' \subseteq I$ *of maximal cardinality and an affine set* Y *of type* $H(I')$ *such that each* $a^i \in Y$. *For each* $x \in \mathbb{R}^n$, *there exists a point* $y \in Y$ *such that* $\gamma(y - a^i) \leq \gamma(x - a^i)$ *for* $i = 1, \ldots, m$, *that is to say,* y *dominates* x *with respect to* $a^1, \ldots, a^m$.

Theorem 5.3 *Let γ be the $[\underline{w}, \overline{w}]$-polyhedral gauge defined by the following set of extreme points $V(\mathsf{B}_0) = \{\check{v}^1, \ldots, \check{v}^{2n}\}$, where $\check{v}^{2i-1} = \frac{1}{\overline{c}_i} e_i$, $\check{v}^{2i} = \frac{1}{\underline{c}_i} e_i$, and $\underline{c}_j < 0 < \overline{c}_j$ for all $j \in I$. Then γ satisfies Property 5.2.*

Proof. Let $x \in \mathbb{R}^n$, and consider

$$\gamma(x - a^i) = \sum_{j=1}^{n} \max_{c_j^s \in \{\underline{c}_j, \overline{c}_j\}} c_j^s (x_j - a_j^i).$$

Since $\underline{c}_j < 0 < \overline{c}_j$, for all $j \in I$,

$$\max_{c_j^s \in \{\underline{c}_j, \overline{c}_j\}} c_j^s (x_j - a_j^i) \geq 0.$$

Now, let y be the projection of x onto the affine set Y, i.e. $y_j = a_j^i$ for all $j \in I'$ and $i = 1, \ldots, m$ and $y_j = x_j$ for all $j \in I \backslash I'$.

Hence, for any a^i it holds that

$$\gamma(x - a^i) = \sum_{j \in I'} \max_{c_j^s \in \{\underline{c}_j, \overline{c}_j\}} c_j^s (x_j - a_j^i) + \sum_{j \in I \backslash I'} \max_{c_j^s \in \{\underline{c}_j, \overline{c}_j\}} c_j^s (x_j - a_j^i) \geq$$

$$\sum_{j \in I \backslash I'} \max_{c_j^s \in \{\underline{c}_j, \overline{c}_j\}} c_j^s (y_j - a_j^i) = \gamma(y - a^i).$$

□

However, the next example shows that Property 5.2 does not guarantees that the Kuhn's convex-hull property holds.

Example 5.2 *In $\mathbb{R}^4$ let $a^1 = (10, 0, 0, 0)$, $a^2 = (0, 10, 0, 0)$ and $a^3 = (0, 0, 10, 0)$. For $\overline{x} = (0, 0, 0, 0)$ and for each a^i, $||\overline{x} - a^i||_1 = \sum_{j=1}^{4} |\overline{x}_j - a_j^i| = 10$. The convex hull of these point is $Conv\{a^1, a^2, a^3\} = \{x \in \mathbb{R}^4 : x_1 + x_2 + x_3 = 10,\ x_4 = 0\ x_i \geq 0\}$ and the smallest affine set of type $H(I')$ containing the points a^1, a^2, a^3, is $Y = \{x \in \mathbb{R}^3 : x_4 = 0\}$. An optimal solution to*

$$\begin{aligned} \textit{minimize} \quad & z \\ \textit{subject to} \quad & ||x - a^i||_1 \leq z \qquad i = 1, \ldots, 3 \\ & x_1 + x_2 + x_3 = 10 \\ & x_4 = 0 \\ & x_i \geq 0 \end{aligned}$$

is the point $(\frac{10}{3}, \frac{10}{3}, \frac{10}{3}, 0)$ whose value is $\frac{40}{3}$, hence Property 5.2 holds but the Kuhn's convex-hull property does not hold.

5.4 The minimax regret linear problem when all uncertainty intervals contain the origin in their interior

In this section, we establish a link between 1-center problems and the minimax regret linear programming when all the uncertainty intervals contain the origin in its interior. We shall exploit this link to establish, a localization result for the set of optimal solutions to the MinMaxR interval problem.

5.4.1 The 1-center problem under the w-block norms

Let B_0 be a symmetric polytope in $\mathbb{R}^n$ defined by the set of its extreme points $V(\mathsf{B}_0) = \{\check{v}^1, \dots, \check{v}^{2n}\}$, where $\check{v}^{2i-1} = w_i e_i$ and $\check{v}^{2i} = -w_i e_i$. and $w_i > 0$ for all $i = 1, \dots, n$. Consider the w-block norm $||.||$ defined as

$$||x|| = \min\{\sum_{j=1}^{2n} \lambda_j : \ x = \sum_{j=1}^{2n} \lambda_j \check{v}^j, \lambda_j \geq 0\}.$$

$$||x - y|| = \sum_{j=1}^{n} \frac{1}{w_j} |x_j - y_j|.$$

The 1-center problem in $\mathbb{R}^n$ under the w-block norm consists in looking for a point $x \in \mathbb{R}^n$ which minimizes the largest distance defined by the w-block norm $||.||$ to the m given points $a^1, \dots, a^m \in \mathbb{R}^n$. This problem can be formulated as follows

$$\begin{aligned} &\text{minimize} \quad z \\ &\text{subject to} \quad \sum_{i=1}^{n} \frac{1}{w_i} |x_i - a_i^j| \leq z \qquad j = 1, \dots, m, \end{aligned}$$

and also as the following linear program

1-center problem under the w-block norm

$$\text{minimize} \quad z$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \nu_i x_i - z \leq \sum_{i=1}^{n} \nu_i a_i^j, \quad \forall \nu_i \in \{\frac{-1}{w_i}, \frac{1}{w_i}\}, \ \forall i = 1, \dots, n, \ \forall j = 1, \dots, m.$$

This linear problem has $2^n m$ constraints and $n+1$ variables $z, x_1, \dots, x_n$. Therefore, when the dimension n is fixed, it can be solved in $O(m)$ time by the algorithm given by Megiddo (1984). Ward and Wendell (1985) prove that the size of the 1-center problem under the w-block norm does not depend on the number of fixed points $\{a^1, \dots, a^m\}$. A simple way to show this is by writing this problem as

$$\text{minimize} \quad z$$
$$\text{s.t.} \quad \sum_{i=1}^{n} \nu_i x_i - z \le \min_{j\in\{1,\dots,m\}} \sum_{i=1}^{n} \nu_i a_i^j, \quad \forall \nu_i \in \{\frac{-1}{w_i}, \frac{1}{w_i}\}, \forall i = 1, \dots, n.$$

Then there are $n+1$ variables and 2^n constraints, regardless of the number of existing points $\{a^1, \dots, a^m\}$.

If we introduce a feasibility region $P = \{x \in \mathbb{R}^n : Ax \ge b,\ x \ge 0\}$, the 1-center problem under the w-block norm γ with the additional restriction that $x \in P$ can be formulated as the following linear program.

Restricted 1-center problem under the w-block norm

$$\begin{aligned} &\text{minimize} \quad z \\ \text{s.t.} \quad &\sum_{i=1}^{n} \nu_i x_i - z \le \min_{j\in\{1,\dots,m\}} \sum_{i=1}^{n} \nu_i a_i^j, \quad \forall \nu_i \in \{\frac{-1}{w_i}, \frac{1}{w_i}\},\ \forall i = 1, \dots, n. \\ &Ax \ge b, \\ &x \ge 0. \end{aligned}$$

Therefore it can solved efficiently when the dimension n is fixed.

Now consider the MinMaxR interval problem in which $\overline{c_i} > 0$ and $\overline{c_i} = -\underline{c_i}$ for $i = 1, \dots, n$,

$$\begin{aligned} &\text{minimize} \quad z \\ \text{s.t.} \quad &\sum_{i=1}^{n} c_i^s (x_i - y_i) \le z, \qquad \forall\ y \in V(P), \forall c_i^s \in \{-\overline{c_i}, \overline{c_i}\}, i = 1 \dots n, \\ &Ax \ge b, \\ &x \ge 0. \end{aligned} \tag{5.5}$$

Since we can also write it as

$$\begin{aligned} &\text{minimize} \quad z \\ \text{s.t.}\ &c^s x - z \le \min_{y\in V(P)} c^s y, \quad \forall c^s = (c_i^s, \dots, c_n^s),\ c_i \in \{-\overline{c}_i, \overline{c}_i\},\ i = 1 \dots n, \\ &Ax \ge b, \\ &x \ge 0 \end{aligned} \tag{5.6}$$

and its relaxation, the Relaxed MinMaxR interval problem as

$$\begin{aligned} &\text{minimize} \;\; z && (5.7)\\ &\text{s.t.} c^s x - z \leq \min_{y\in V(P)} c^s y, \quad \forall c^s = (c^s_i, \ldots, c^s_n),\; c^s_i \in \{-\overline{c}_i, \overline{c}_i\}, i = 1\ldots n, \end{aligned}$$

thus, when the dimension is fixed, this particular case of the MinMaxR interval problem can be viewed as the restricted 1-center problem in $\mathbb{R}^n$ under the block norm γ defined by the polytope B_0 (the unit ball) with extreme points $V(\mathsf{B}_0) = \{\check{v}^1, \ldots, \check{v}^{2n}\}$, where $\check{v}^{2i-1} = \frac{1}{\overline{c}_i} e_i$, $\check{v}^{2i} = -\frac{1}{\overline{c}_i} e_i$, and

$$||x - y|| = \sum_{i=1}^{n} \overline{c}_i |x_i - y_i| = \sum_{i=1}^{n} \max_{c^s_i \in \{-\overline{c}_i, \overline{c}_i\}} c^s_i (x_i - y_i).$$

We observe that

$$\sum_{i=1}^{n} \max_{c^s_i \in \{-\overline{c}_i, \overline{c}_i\}} c^s_i (x_i - y_i) = \sum_{j=1}^{2n} \lambda_j$$

and, for all $i \in \{1, \ldots, n\}$,

$$\text{if } x_i - y_j > 0, \; \textit{then} \;\; \lambda_{2i-1} = \overline{c}_i (x_i - y_i) \;\; \text{and} \;\; \lambda_{2i} = 0,$$

$$\text{if } x_i - y_j \leq 0, \; \textit{then} \;\; \lambda_{2i-1} = 0 \;\; \text{and} \;\; \lambda_{2i} = -\overline{c}_i (x_i - y_i).$$

The points $a^1, \ldots a^m$ are thus the vertices of the feasible region P, which are optimal for some scenario.

In the case of the l_1 norm, Francis et al. (1992) observe that the 1-center problem under rectilinear distances in two dimensions is equivalent to the *diamond covering problem*, which amounts to enclose m given points in the plane within a diamond of minimum radius. The radius of a diamond is equal to half the length of the line segment joining opposite vertices. Love et al. (1988) show that this problem is also equivalent to the problem of enclosing m known points in the plane within m diamonds of minimum radius, in such a way that the intersection of all the m diamonds is non-empty.

For fixed dimension, also in the case when $\overline{c_i} = -\underline{c_i}$, for all $i = 1 \ldots n$ and $\overline{c}_i > 0$, the MinMaxR interval problem (5.6) in $\mathbb{R}^n$ is equivalent to the problem of enclosing m known points in $\mathbb{R}^n$ within m n-dimensional diamonds of minimum radius, such that the intersection of all the m diamonds with $P = \{x \in \mathbb{R}^n : Ax \geq b, \; x \geq 0\}$ is nonempty. Similarly, the Relaxed MinMaxR interval problem (5.7) in $\mathbb{R}^n$ when $\overline{c_i} = -\underline{c_i} > 0$, for all $i = 1 \ldots n$, is equivalent to the problem of enclosing m points $a^1, \ldots a^m$ in $\mathbb{R}^n$ within

m n-dimensional diamonds of minimum radius, such that the intersection of all the m diamonds is non-empty and equivalent to enclose m given points within a diamond of minimum radius.

This is another way to understand why the MinMaxR interval problem and its relaxation satisfy the visibility property, that is to say, why if each optimal solution to the Relaxed MinMaxR interval problem (5.7) is infeasible for the MinMaxR interval problem (5.6), then all the optimal solutions of MinMaxR interval are visible from the set of optimal solutions of the Relaxed MinMaxR interval problem. This relation between the Relaxed MinMaxR interval problem and the 1-center under the w-block norm allows also to localize an optimal solution.

Proposition 5.1 *Suppose that there exists a subset of indices $I' \subseteq I$ of maximal cardinality and an affine set Y of type $H(I')$ such that for all $c^s \in D$ each optimal solution x^{*s} to the problem*

$$\begin{aligned} &\textit{minimize} \;\; c^s x \\ &\textit{s.t.} \;\; Ax \geq b \\ &x \geq 0, \end{aligned}$$

belongs to Y . Then there exists an optimal solution $\overline{x}$ to the Relaxed MinMaxR interval problem (5.7) such that $\overline{x} \in Y$.

Proof. This is a direct consequence of Theorem 5.3. □

This result implies that if all the optimal solutions to the UOLPP (defined in Chapter 4), for all $c^s \in D$ belong to $Y = \{x \in \mathbb{R}^n : x_i = k_i \; if \; i \in I'\}$, we can fix in the Relaxed MinMaxR interval problem (5.7) all the variables $x_i = k_i$ for which $i \in I'$.

5.4.2 The 1-center problem under polyhedral gauges

Consider the MinMaxR interval problem when all the intervals $\left[\underline{c_i}, \overline{c_i}\right]$ contain the origin in its interior. From formulation (5.1) it is clear that such problem can be viewed as the restricted 1-center problem in $\mathbb{R}^n$ under the $[\underline{w}, \overline{w}]$-polyhedral gauge γ defined by the polytope B_0 (the unit ball) with extreme points $V(\mathsf{B}_0) = \{\check{v}^1, \ldots, \check{v}^{2n}\}$, where $\check{v}^{2i-1} = \frac{1}{\overline{c}_i} e_i$, $\check{v}^{2i} = \frac{1}{\underline{c}_i} e_i$ and the points $a^1, \ldots a^m$ are the vertices of P, which are optimal for some scenario.

When the dimension is fixed, since the MinMaxR interval problem is polynomial, we may immediately conclude that the 1-center problem under polyhedral gauges is also polynomial.

Since the distance from y to x is defined (remember that symmetry is not necessarily required) by

$$d(y,x) = \gamma(x-y) = \sum_{i=1}^{n} \max_{c_i^s \in \{\underline{c}_i, \overline{c}_i\}} c_i^s(x_i - y_i),$$

we obtain that

$$\sum_{i=1}^{n} \max_{c_i^s \in \{\underline{c}_i, \overline{c}_i\}} c_i^s(x_i - y_i) = \sum_{j=1}^{2n} \lambda_j$$

where

$$\lambda_{2i-1} = \overline{c}_i(x_i - y_i) \text{ and } \lambda_{2i} = 0 \text{ if } x_i - y_i > 0,$$

$$\lambda_{2i-1} = 0 \text{ and } \lambda_{2i} = \underline{c}_i(x_i - y_i) \text{ if } x_i - y_i \leq 0,$$

for all $i \in \{1, \dots n\}$.

We remark that in this case, when all the intervals contain the origin in its interior the unit ball is bounded, see Figure 5.2.

Similarly to the w-block norm case, in fixed dimension, the particular MinMaxR (Relaxed MinMaxR) interval problem in $\mathbb{R}^n$ when $\underline{c_i} < 0 < \overline{c_i}$, for all $i = 1 \dots n$, is equivalent to the problem of enclosing m known points in $\mathbb{R}^n$ within m n-dimensional diamonds of minimum radius, such that the intersection of all the m diamonds and P (all the m diamonds) is nonempty.

On the contrary, the Relaxed MinMaxR interval problem with gauges cannot be viewed as finding a diamond of minimum radius which encloses m given points because the unit ball is not necessarily symmetric.

Also, as in the case of w-block norms, the set of optimal solutions to the Relaxed MinMaxR interval problem can have an empty intersection with P. Hence, given a set $\{a^1, \dots, a^m\}$ of m points in $\mathbb{R}^n$, the set of optimal solutions of the 1-center problem under the $[\underline{w}, \overline{w}]$-polyhedral gauge can have an empty intersection with $Conv\{a^1, \dots, a^m\}$, the convex hull of the points.

Finally Theorem 5.3 allows us to localize an optimal solution to the Relaxed MinMaxR interval problem on a hyperplane of type $H(I')$ in the case of gauges whose unit ball contains the origin in its interior.

In the following section we shall generalize these concepts in the case when the set of coefficients D does not necessarily contains the origin in its interior.

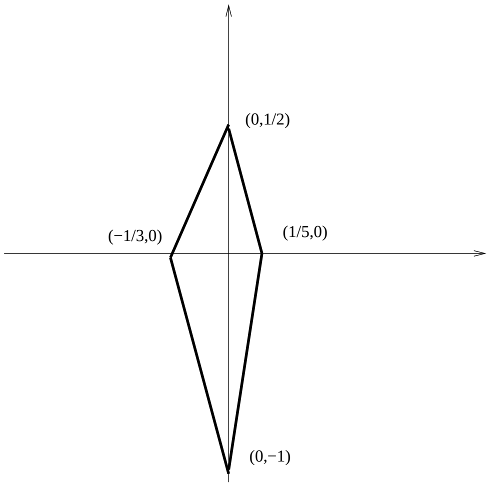

Figure 5.2: $\mathsf{B}(v, r_0)$ where $v = (0,0)$, $r_0 = 1$, $[\underline{c}_1, \overline{c}_1] = [-3, 5]$ and $[\underline{c}_2, \overline{c}_2] = [-1, 2]$

5.5 The minimax regret linear program when some uncertainty intervals do not contain the origin in its interior

Generalizing some results on location theory, we give here the geometrical description of the minimax regret problem under interval uncertainty, in the general case when D does not necessarily contains the origin in its interior.

5.5.1 1-center problem under the regret function

In this section we introduce the 1-center problem under the regret function and an application of this model. We shall also give the geometrical description of the MinMaxR problem under interval uncertainty in the objective function coefficients in the general case but first we need the following definitions.

Given a set of n intervals $\{[\underline{c_1}, \overline{c_1}], \ldots [\underline{c_n}, \overline{c_n}]\}$, such that $\underline{c_i} < \overline{c_i}$, for $i = 1, \ldots n$ a point $v \in \mathbb{R}^n$, and a scalar $r_0 \in \mathbb{R}$, consider the polyhedron $\mathcal{D}(v, r_0)$ defined as the intersection of the 2^n half-spaces $c_1^s x_1 + \ldots c_n^s x_n \leq r_0 + \sum_{i=1}^n c_i^s v_i$ where $c_i^s \in \{\underline{c_i}, \overline{c_i}\}$. We call this polyhedron the *generalized n-dimensional diamond of radius r_0 centered on the point v.*

We remark that in the case when there exists an index $i \in I$ for which the interval $[\underline{c_i}, \overline{c_i}]$ does not contain the origin in its interior, the n-dimensional diamond is unbounded, see Figure 5.3. Given the polyhedron $\mathcal{D}(\mathbf{0},1)$ associated to the intervals $\left[\underline{c_1},\overline{c_1}\right],\ldots\left[\underline{c_n},\overline{c_n}\right]$, and a point $x \in \mathbb{R}^n$, the *regret* of x with respect to $\mathcal{D}(\mathbf{0},1)$ is defined as

$$R(x) := \sum_{j=1}^{n} \max_{c_j^s \in \{\underline{c}_j, \overline{c}_j\}} c_j^s x_j,$$

and the regret from y to x is defined by

$$R(x-y) := \sum_{j=1}^{n} \max_{c_j^s \in \{\underline{c}_j, \overline{c}_j\}} c_j^s (x_j - y_j).$$

Let R the regret function associated with $\mathcal{D}(\mathbf{0},1)$. The *asymptotic cone* of R is defined as

$$\mathcal{D}(\mathbf{0},1)_\infty := \{x \in \mathbb{R}^n : \ R(x) \leq 0\}$$

and this set coincides with the closed convex cone determined by the extreme directions $\mathcal{D}(\mathbf{0},1)_\infty^{ext}$ of $\mathcal{D}(\mathbf{0},1)$ (see Hinojosa and Puerto (2003), in the case of gauges of closed convex sets containing the origin).

$$\mathcal{D}(\mathbf{0},1)_\infty := \{x \in \mathbb{R}^n : \ y + \lambda x \in \mathcal{D}(\mathbf{0},1) \ \forall y \in \mathcal{D}(\mathbf{0},1), \forall \lambda \geq 0\}.$$

This function verifies the following properties:

1. $R(x) = 0 \ iff \ x \in \partial\mathcal{D}(\mathbf{0},1)_\infty$.
2. $R(x) < 0 \ iff \ x \in \mathcal{D}(\mathbf{0},1)_\infty \setminus \partial\mathcal{D}(\mathbf{0},1)_\infty$.

One can easily see that in general, $R(x-y) \neq R(y-x)$, each point $x \in \mathcal{D}(\mathbf{0},1)$ satisfies $R(x) \leq 1$ and $\mathcal{D}(\mathbf{0},1)$ contains the origin. In general, for each $r_0 \in \mathbb{R}$, we can define $\mathcal{D}(v, r_0)$ as the set of points $x \in \mathbb{R}^n$ such that $R(x-v) \leq r_0$.

Let $I = \{1,\ldots n\}$, then for each pair of points x and y, there exist $\{\lambda_i : \lambda_i \in \mathbb{R}\}_{i\in I}$, such that the regret of the point x with respect to the y is

$$R(x-y) = \sum_{i=1}^{n} \max_{c_i^s \in \{\underline{c}_i, \overline{c}_i\}} c_i^s (x_i - y_i) = \sum_{j=1}^{2n} \lambda_j.$$

For $i \in I$,

$$\lambda_{2i-1} = \overline{c}_i(x_i - y_i) \ \text{ and } \ \lambda_{2i} = 0 \ \text{ if } x_i - y_i > 0,$$

$$\lambda_{2i-1} = 0 \ \text{ and } \ \lambda_{2i} = \underline{c}_i(x_i - y_i) \ \text{ if } x_i - y_i \leq 0.$$

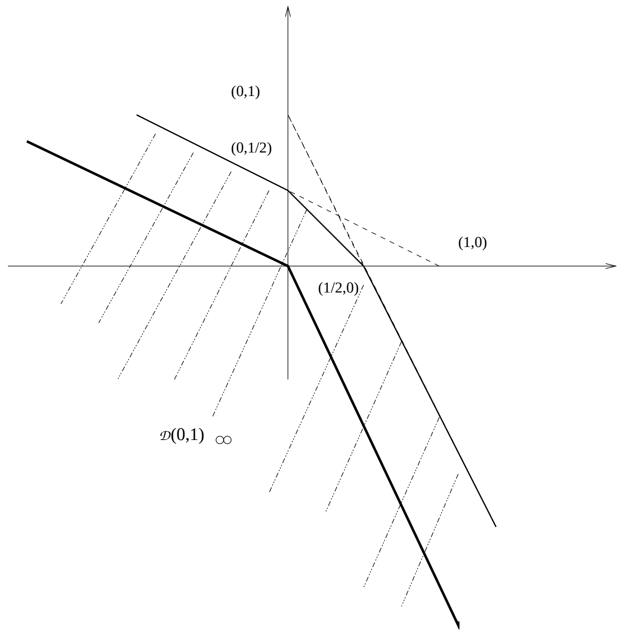

Figure 5.3: Region $\mathcal{D}(v, r_0)$ where $v = (0,0)$, $r = 1$ and $[\underline{c}_1, \overline{c}_1] = [\underline{c}_2, \overline{c}_2] = [1,2]$, and $\mathcal{D}(\mathbf{0}, 1)_\infty$

Example 5.3 *Consider the intervals* $\{[1,2], [1,2]\}$*. Then if* $v = (0,0)$ *and* $r_0 = 1$*,* $\mathcal{D}((0,0),1)$ *is the intersection of the half-spaces:* $\{(x_1,x_2) \in \mathbb{R}^2 : x_1 + x_2 \leq 1\}$*,* $\{(x_1,x_2) \in \mathbb{R}^2 : x_1 + 2x_2 \leq 1\}$*,* $\{(x_1,x_2) \in \mathbb{R}^2 : 2x_1 + x_2 \leq 1\}$*, and* $\{(x_1,x_2) \in \mathbb{R}^2 : 2x_1 + 2x_2 \leq 1\}$*. This situation is depicted in Figure 5.3.*

When the dimension is fixed, the most general case of the MinMaxR interval problem 4.12 (i.e. in which some uncertainty intervals may do not contain the origin in its interior) is equivalent to finding the minimum value $r_0 \in \mathbb{R}$ for which

$$\bigcap_{y \in V(P)} \mathcal{D}(y, r_0) \cap P \neq \emptyset$$

Example 5.4 *Consider the following problem*

$$\begin{aligned} & min \;\; z \\ s.t. \;\; & c^s x - z \leq \min_{y \in V(P)} c^s y, \quad \forall c^s \in [-1,1] \times [-3,-2] \\ & \frac{1}{4} x_1 - x_2 \geq 0 \\ & -x_1 - 4x_2 \geq -8 \end{aligned}$$

where $P = \{x = (x_1, x_2) \in \mathbb{R}^2 : \frac{1}{4}x_1 - x_2 \geq 0$ *and* $-x_1 - 4x_2 \geq -8\}$, *see Figure 5.4. The solution to the MinMaxR interval problem is the point* $(x_1, x_2) = (4, 1)$ *of value* $z = 2$.

Given a set $\mathcal{A} = \{a^1, \ldots, a^m\}$ of m facilities in $\mathbb{R}^n$, the *restricted 1-center problem under the regret function* consists in looking for a point $x \in P$ which minimizes the maximum regret to each of the given points. This problem can be formulated as

$$\min_{x \in P} \max_{a_j \in \mathcal{A}} \{R(x - a_j)\}.$$

An application of this model could be the location of a facility which is attractive for the demand points $\mathcal{A} = \{a^1, \ldots, a^m\}$ but repulsive for some fundamental directions. For example, such a direction may corresponds to a river with risk of overflowing and located in the exterior of the city. The repulsive fundamental directions correspond to the intervals for which $\overline{c}_i \leq 0$.

Clearly, when the dimension n is fixed, by formulation (5.1) the MinMaxR interval and the restricted 1-center problems under a regret function are equivalent in the sense that the demands points of the restricted 1-center correspond to the vertices of the polytope P, which are optimal for some scenario.

5.5.2 Robust information, 0-persistent variables and preprocessing

Now consider a UOLPP, where $P = \{x \in \mathbb{R}^n : Ax \geq b,\ x \geq 0\}$ and the uncertainty set of coefficients $D = \prod_{i=1}^n [\underline{c}_i, \overline{c}_i]$ does not necessarily contain the origin in its interior. Suppose that there exists a nonempty subset $I' \subseteq I$ and an affine set $Y = \{x \in \mathbb{R}^n : x_i = 0\ if\ i \in I'\}$, such that all the optimal solutions to the uncertain objective linear problem for all $c^s \in D$, lies onto Y. Then we have the following question: under what conditions does a solution of the MinMaxR interval problem exist that belongs to Y?. The last geometrical description will help us to give an answer to this question, but first we shall give some definitions and notations.

Definition 5.1 *Consider a UOLPP. A point* $p \in \{x \in \mathbb{R}^n : Ax \geq b,\ x \geq 0\}$ *is said to be a* **weak point** *if there exists at least* $c^s \in D$ *for which* p *is an optimal solution of the problem under the scenario* s*. We shall call* **weak vertex***, a vertex of* P *that is a weak point. We shall denote* W *the set of weak vertices of the* $UOLPP$*.*

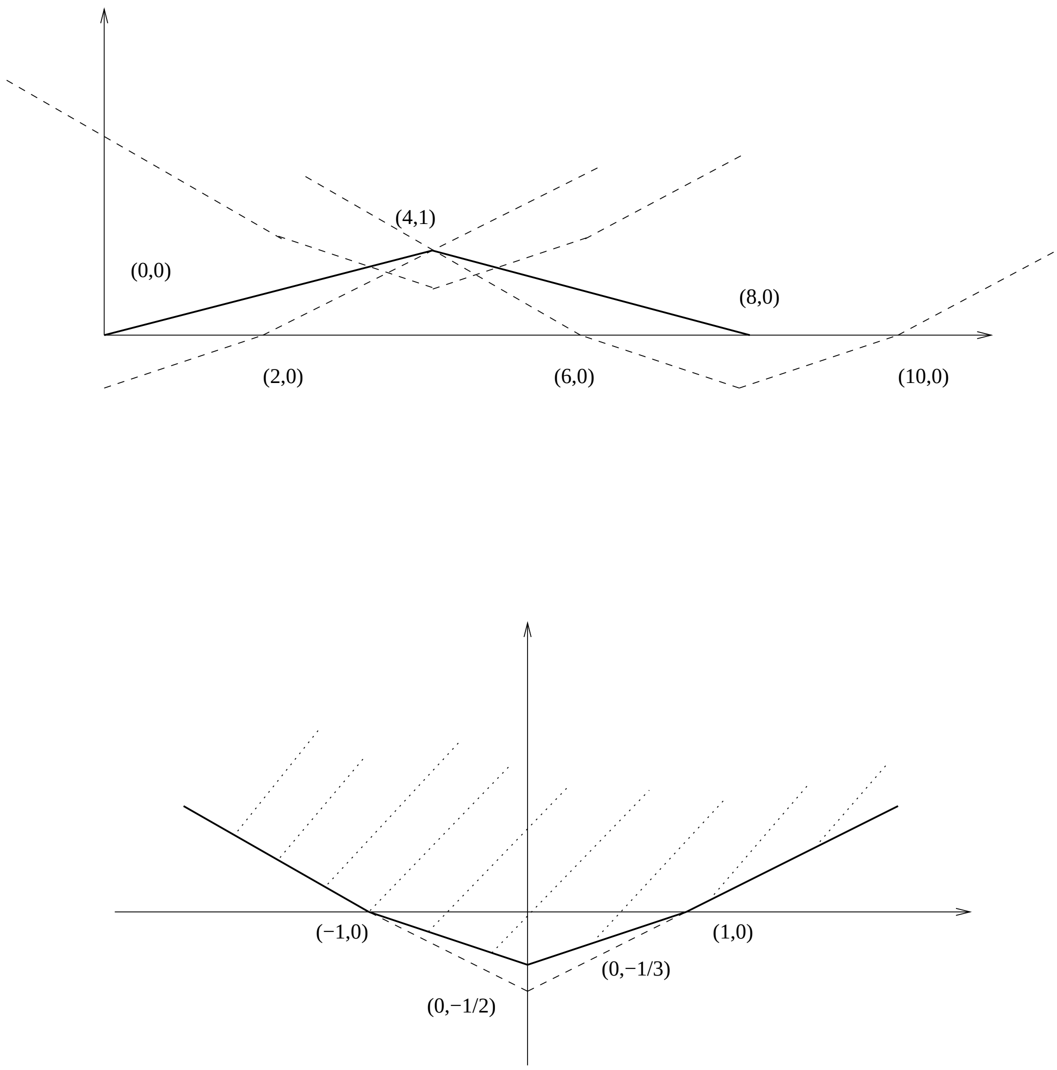

Figure 5.4: a) The polytope $P = \{x = (x_1, x_2) \in \mathbb{R}^2 : \frac{1}{4}x_1 - x_2 \geq 0$ and $-x_1 - 4x_2 \geq -8\}$ and b) the region $\mathcal{D}((0,0),1)$ where $R(x) \leq 1$ corresponding to the product of intervals $[-1,1] \times [-3,-2]$

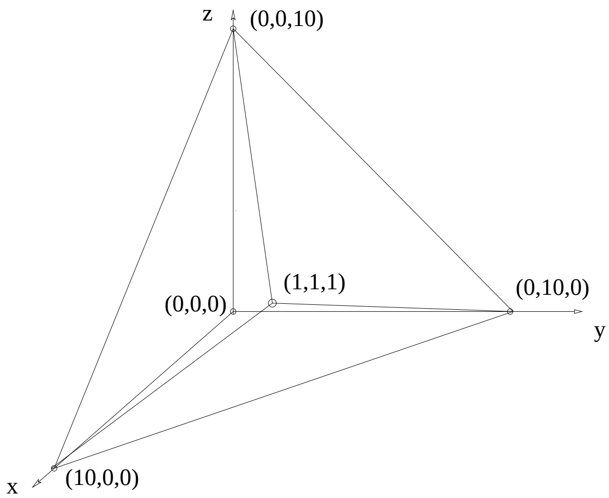

Figure 5.5: Polyhedron $P = \{x \in R^3 : x_1 + x_2 + 8x_3 \geq 10,\ x_1 + 8x_2 + x_3 \geq 10,\ 8x_1 + x_2 + x_3 \geq 10,\ x_1 + x_2 + x_3 \geq 10,\ x \geq 0\ \}$

Using this concept and formulation (5.1), we observe that, when the dimension is fixed, the most general case of the MinMaxR interval problem (i.e. in which some uncertainty intervals may not contain the origin in its interior) is also equivalent to finding the minimum value $r_0 \in \mathbb{R}$ for which

$$\bigcap_{\omega \in W} \mathcal{D}(\omega, r_0) \cap P \neq \emptyset$$

We observe that in the case when all the intervals $[\underline{c}_i, \overline{c}_i]$ contain the origin in its interior, $V(P)$ coincides with the set of weak vertices of the $UOLPP$. However, even in this case, the set of optimal solutions to the MinMaxR interval problem may have an empty intersection with the interior of P. This situation is illustrated by the following example.

Example 5.5 *Let P the polytope in $\mathbb{R}^3$, defined by the following inequalities: $x_1+x_2+x_3 \leq 10$, $x_1+x_2+8x_3 \geq 10$, $x_1+8x_2+x_3 \geq 10$, and $8x_1+x_2+x_3 \geq 10$. Suppose that $D = [-1,1] \times [-1,1] \times [-1,1]$. The set of weak vertices of P is $W = \{a^1 = (10,0,0), a^2 = (0,10,0), a^3 = (0,0,10), a^4 = (1,1,1)\} = V(P)$. The unique optimal solution to the MinMaxR interval problem*

$$\textit{minimize} \quad z$$

$$\textit{subject to} \quad c(x - a^i) \leq z \quad \forall\ c \in D \quad i = 1, \ldots, 3$$

$$x_1 + x_2 + x_3 \leq 10$$

$$x_1 + x_2 + 8x_3 \geq 10$$

$$x_1 + 8x_2 + x_3 \geq 10$$

$$8x_1 + x_2 + x_3 \geq 10$$

$$x_i \geq 0$$

is the point $\overline{x} = (1, 1, 1)$*, see Figure 5.5, and*

$$\max_{c \in D} \max_{a^i \in W} c(\overline{x} - a^i) = 11.$$

This situation can be explained by Corollary 5.1, because the unique optimal solution to the Relaxed MinMaxR interval problem is the point $(0, 0, 0)$*. Hence all optimal solution to the MinMaxR interval problem must be visible form this point and this must belong to the boundary of* P*.*

Definition 5.2 *A variable* x_k *is 0-persistent if the* k*th coordinate of all the optimal solutions* x^{*s} *to LP, for all* $c^s \in D$*, are equal to zero.*

Consider an optimal solution x^{*s} to LP corresponding to c^s. Variable x_k is thus a 0-persistent variable if and only if for all $c^s \in D$ and for all $x' \in P$ such that $x'_k > 0$, $c^s x' > c^s x^{*s}$.

Then in the case when $\prod_{i=1}^{n} [\underline{c}_i, \overline{c}_i]$ contains the origin in its interior and the polytope P is full dimensional, since $W = V(P)$, there is no 0-persistent variable.

Finally we shall give an answer to the main question of this section, that is to say, suppose that there exists a nonempty subset $I' \subseteq I$ and a affine set $Y = \{x \in \mathbb{R}^n : x_i = 0 \ if \ i \in I'\}$, such that all optimal solution to the uncertain objective linear problem for all $c^s \in D$, lies onto Y. Under what conditions does a solution to the MinMaxR interval problem exist that lies onto Y ?

The following theorem assures that if x_k is a 0-persistent variable, there exists an optimal solution x^* to the MinMaxR interval problem such that $x^*_k = 0$.

This result implies that we can preprocess the problem prior to the solution of the minimax regret linear programming problem. The preprocessing delete the 0-persistent variables in order to reduce the dimension of the problem.

Theorem 5.4 *Consider the UOLPP where $P = \{x \in \mathbb{R}^n : Ax \geq b,\ x \geq 0\}$ and the uncertainty set of coefficients is $D = \prod_{i=1}^{n}[\underline{c}_i, \overline{c}_i]$. Suppose that P is a full dimensional polytope and let x_k be a 0-persistent variable. Let $W = \{\omega^1, \ldots \omega^t\}$ be the set of weak vertices of UOLPP and consider the minimum value $r_0 \in \mathbb{R}$ such that*

$$\bigcap_{\omega \in W} \mathcal{D}(\omega, r_0) \cap P \neq \emptyset$$

then there exists a

$$v \in \bigcap_{\omega \in W} \mathcal{D}(\omega, r_0) \cap P$$

such that $v_k = 0$. That is to say, if x_k is a 0-persistent variable, there exists an optimal solution x^* to the MinMaxR interval problem such that $x_k^* = 0$.

Proof. Let $Y = \{x \in \mathbb{R}^n : x_k = 0\}$ and consider the minimum radius $r_0 \in \mathbb{R}$ such that

$$\bigcap_{\omega \in W} \mathcal{D}(\omega, r_0) \cap P \neq \emptyset,$$

and suppose that all

$$v \in \bigcap_{\omega \in W} \mathcal{D}(\omega, r_0) \cap P$$

are such that $v_k > 0$. That implies that all $y \in P \cap Y$, are such that

$$y \notin \bigcap_{\omega \in W} \mathcal{D}(\omega, r_0) \cap P$$

then

$$y \notin \bigcap_{\omega \in W} \mathcal{D}(\omega, r_0),$$

thus given a $y \in P \cap Y$, there exist a $\omega^j \in W$ such that $y \notin \mathcal{D}(\omega^j, r_0)$, and by definition of $\mathcal{D}(\omega^j, r_0)$, there exists a $c \in \prod_{i=1}^{n}\{\underline{c}_i, \overline{c}_i\}$ such that $c(y - \omega^j) > r_0$. Now, given a

$$v \in \bigcap_{\omega \in W} \mathcal{D}(\omega, r_0) \cap P,$$

for all $\omega \in W$ and for all $c^s \in \prod_{i=1}^{n}\{\underline{c}_i, \overline{c}_i\}$, $c^s(v - \omega) \leq r_0$. Then in particular,

$$c(v - \omega^j) \leq r_0 < c(y - \omega^j),$$

and for all $y \in P \cap Y$, there exists a $c \in \prod_{i=1}^{n}\{\underline{c}_i, \overline{c}_i\}$ such that $cv < cy$. Since $v_k \neq 0$, x_k is not a 0-persistent variable. □

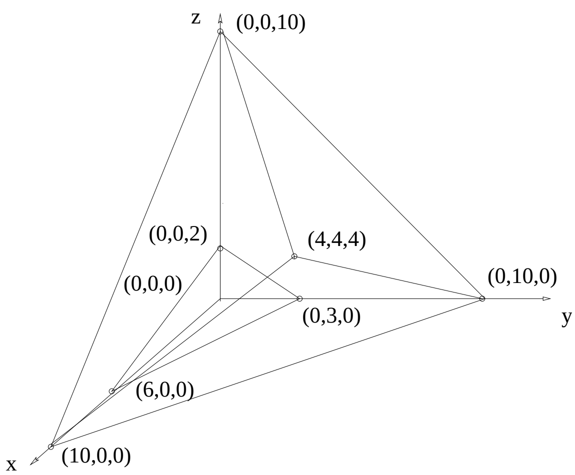

Figure 5.6: Polyhedron $P = \{x \in R^3 : -x_1 - 2x_2 - 2x_3 \geq -20,\ -2x_1 - x_2 - 2x_3 \geq -20,\ -2x_1 - 2x_2 - x_3 \geq -20,\ x_1 + 2x_2 + 3x_3 \geq 6,\ x \geq 0\ \}$

In order to illustrate the last result, we give the following example.

Example 5.6 *Consider the UOLPP associated to* $P = \{(x_1, x_2, x_3) \in R^3 : -x_1 - 2x_2 - 2x_3 \geq -20, -2x_1 - x_2 - 2x_3 \geq -20, -2x_1 - 2x_2 - x_3 \geq -20, x_1 + 2x_2 + 3x_3 \geq 6, x \geq 0\}$ *and* $D = [1, 2] \times [-4, 3] \times [-1, 1]$.

In this example the set of weak vertices of the UOLPP is

$$\begin{array}{ll} c^{s_1} = (1, -4, 1) & y^{*s_1}* = (0, 10, 0) \\ c^{s_2} = (1, -4, -1) & y^{*s_2} = (0, 10, 0) \\ c^{s_3} = (1, 3, 1) & y^{*s_3} = (0, 0, 2) \\ c^{s_4} = (1, 3, -1) & y^{*s_4} = (0, 0, 10) \\ c^{s_5} = (2, -4, -1) & y^{*s_5} = (0, 10, 0) \\ c^{s_6} = (2, 3, 1) & y^{*s_6} = (0, 0, 2) \\ c^{s_7} = (2, 3, -1) & y^{*s_7} = (0, 0, 10) \\ c^{s_8} = (2, -4, 1) & y^{*s_8} = (0, 10, 0). \end{array}$$

A solution of the MinMaxR interval problem is $(0, 5.55555, 4.444444)$ *with value* 22.22222*, and this solution lies on the hyperplane* $x_1 = 0$*, see Figure 5.6.*

Now consider the LP

$$\text{minimize} \quad c^s x$$

$$\text{subject to} \quad Ax \geq b$$

$$x \geq 0,$$

where the matrix A is a $m \times n$ matrix, $c^s \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$. We shall denote by A_i, for $i = 1, \ldots, m$ the i-th line vector of the matrix A and by A^j for $j = 1, \ldots, n$ the j-th column vector.

The next two results give sufficient conditions for a variable to be 0-persistent.

Lemma 5.1 *Consider the UOLPP where* $P = \{x \in \mathbb{R}^n : Ax \geq b,\ x \geq 0\}$ *and the uncertainty set of coefficients is* $D = \prod_{i=1}^{n} [\underline{c}_i, \overline{c}_i]$. *If there exists* $k \in \{1, \ldots, n\}$ *such that* $\underline{c}_k > 0$ *and* $A_{ik} \leq 0$ *for all* $i = 1, \ldots, m$ *then* x_k *is a 0-persistent variable.*

Proof. Let $x \in P$ such that $x_k > 0$ and $x' \in \mathbb{R}^n$ such that $x'_j = x_j$, for all $j \neq k$ and $x'_k = 0$. We can observe that for all $c^s \in D$, $c^s x' < c^s x$ and $Ax' = Ax - A^k x_k \geq Ax \geq b$, thus for all $c^s \in D$, $x_k^{*s} = 0$ and x_k is 0-persistent. □

Theorem 5.5 *Consider* $c^{s'} \in D$ *such that*

$$c_j^{s'} = \begin{cases} \overline{c}_j & \text{if } j \neq k, \\ \underline{c}_j & \text{if } j = k. \end{cases}$$

Let $y^{*\underline{s}}$ *and* $x^{*s'}$ *be optimal solutions to Problem (5.8) and (5.9) defined as follows:*

$$\begin{aligned} \text{minimize} \quad & \underline{c}y \\ \text{subject to} \quad & Ay \geq b \\ & y \geq 0, \end{aligned} \tag{5.8}$$

$$\begin{aligned} \text{minimize} \quad & c^{s'}x \\ \text{subject to} \quad & Ax \geq b \\ & x_i \geq 0, \ \text{for } i \neq k. \end{aligned} \tag{5.9}$$

If

$$\underline{c}y^{*\underline{s}} > c^{s'}x^{*s'},$$

then x_k *is a 0-persistent variable.*

Proof. We shall denote by y^{*s} and x^{*s} optimal solutions to the following problems:

$$\begin{aligned} \text{minimize} \quad & c^s y \\ \text{subject to} \quad & Ay \geq b \\ & y \geq 0, \end{aligned} \tag{5.10}$$

and

$$\begin{aligned} \text{minimize} \quad & c^s x \\ \text{subject to} \quad & Ax \geq b \\ & x_i \geq 0, \ \text{for } i \neq k \end{aligned} \tag{5.11}$$

respectively. We can observe that for all $s \in S$, all the optimal solutions y^{*s} to the problem (5.10) are such that $c^s y^{*s} \geq \underline{c} y^{*s} \geq \underline{c} y^{*\underline{s}}$, thus if $\underline{c} y^{*\underline{s}} > c^{s'} x^{*s'}$, then for $c^s = c^{s'}$, $x^{*s'}$ is non feasible for the problem (5.10) and then $x_k^{*s'} < 0$. This implies that for all $c^s \in D$, $c^{s'} x^{*s'} \geq c^s x^{*s'} \geq c^s x^{*s}$. Thus for all $c^s \in D$, we have

$$c^s y^{*s} \geq \underline{c} y^{*s} \geq \underline{c} y^{*\underline{s}} > c^{s'} x^{*s'} \geq c^s x^{*s'} \geq c^s x^{*s}.$$

Let us prove now that the inequality $c^s y^{*s} > c^s x^{*s}$, implies that all the optimal solutions y^{*s} to the problem (5.10) are such that $y_k^{*s} = 0$. If for $c^s \in D$, $c^s y^{*s} > c^s x^{*s}$, x^{*s} is not a feasible solution to the problem (5.10), and thus $x_k^{*s} < 0$. If we suppose that $y_k^{*s} > 0$ there exists a $\lambda \in [0,1]$, and a point $x' = \lambda x^{*s} + (1-\lambda) y^{*s}$, such that $x'_k = 0$ and

$$c^s x' = c^s(\lambda x^{*s} + (1-\lambda) y^{*s}) = \lambda c^s x^{*s} + (1-\lambda) c^s y^{*s} < \lambda c^s y^{*s} + (1-\lambda) c^s y^{*s} = c^s y^{*s}.$$

Since x' is a feasible solution to the problem (5.10), then y^{*s} is not optimal to the problem (5.10) and that is a contradiction. Since $c^s y^{*s} > c^s x^{*s}$, for all $c^s \in D$, then all optimal solutions to the problem (5.10) for all $c^s \in D$ are such that $y_k^{*s} = 0$, hence y_k is a 0-persistent variable. □

5.5.3 Numerical results

The condition of Theorem 5.5 has been tested on randomly generated instances. Except for the last table, all the computational experiments were made on a Pentium II 400MHz station under Linux 2.2.16-SMP with 384 MB of RAM. The algorithm uses the CPLEX 7.0 library to solve the linear

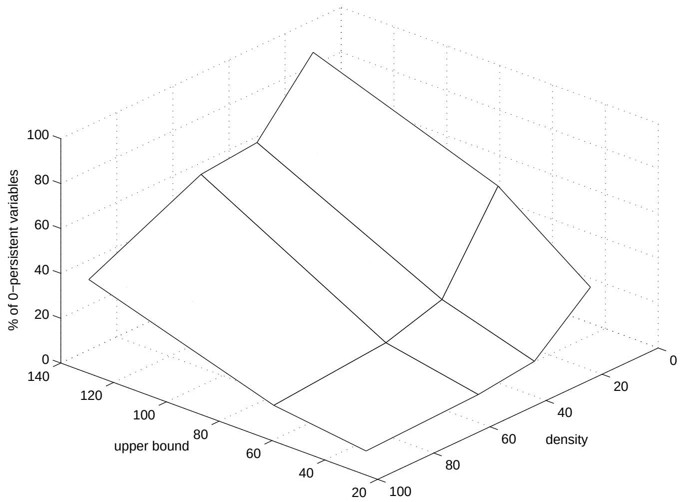

Figure 5.7: Percentage of 0-persistent variables, for $n = 70$

programs. We give, for each series, the mean value μ and the standard deviation σ of the number of 0-persistent variables that are detected. A series contains 10 randomly generated instances with same dimension n, number of constraints $Nconstr$ and density Δ of the matrix A.

To generate the instances, we use the following procedure. For Tables 5.1 to 5.3, The elements of the vector b, are randomly chosen between $-M$ and M, $\underline{c}$ and $\overline{c}$ are randomly chosen between 0 and M, where $M = 2n$ for Table 5.1, $M = n$ for Table 5.2 and $M = \frac{n}{2}$ for Table 5.3.

For each element of the matrix A, a random number between 0 and 1 is generated. If the number is less than Δ, then the element is randomly chosen between -20 and 20, otherwise it is fixed to 0. Some entries are added to the matrix A in order to ensure that there is no empty line or column. Then, the additional constraint $\mathbf{1}^t x \leq n$ guarantees that the polyhedron $P = \{x \in \mathbb{R}^n : Ax \geq b,\ x \geq 0\}$ is bounded.

Tables 5.1 to 5.3 show that, for positive intervals, when the density Δ of matrix A increases, in general, the percentage of the number of 0-persistent variables decreases. While the size of the interval $[0, M]$ increases, the number of 0-persistent variables increases. See Figure 5.7 for $n = 70$ and Figure 5.8 for $M = 2n$. In Figures 5.9 and 5.10, the elements of the vector b, are randomly chosen between $-2n$ and $2n$, $\underline{c}$ and $\overline{c}$ are randomly chosen between $-M$ and $2n$, for $M = n, \frac{n}{2}, \frac{n}{4}, \frac{n}{8}, \frac{n}{16}, \frac{n}{32}, \frac{n}{64}, \frac{n}{128}$ and the density is fixed to 0.3. Figures 5.9 and 5.10 show that, in this case, the percentage of the number of 0-persistent variables increases when M decreases.

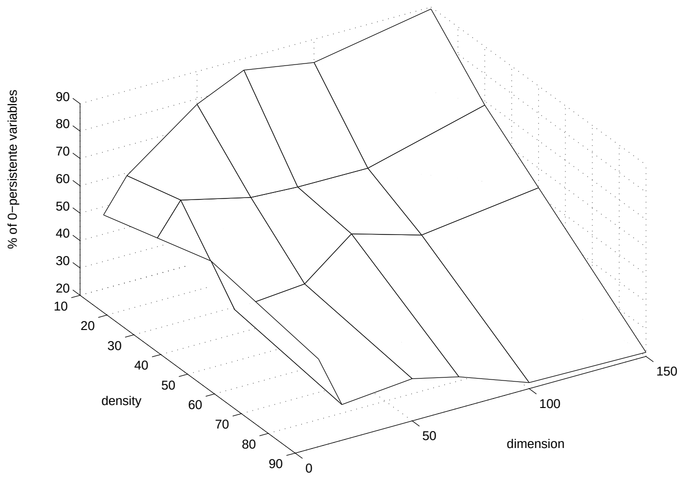

Figure 5.8: Percentage of 0-persistent variables, for $M = 2n$

In order to reduce the computing time used to solve a minimax regret linear programming problem under interval uncertainty we use our 0-persistent variables preprocessing procedure that consists of deleting the 0-persistent variables. We first solve the problem with all the variables. Then we use our preprocessing to remove the 0-persistent variables from the problem, and finally we solve it one more time.

The preprocessing has been tested on randomly generated instances. In this part, the experiments were made on a Pentium III 1 GHz station under Linux 2.4.20-60GB-SMP with 1 GB of RAM. The algorithm was coded in C (compiler gcc) and uses the CPLEX 7.0 library to solve the linear programs. In Table 5.4 we show the behavior of the number of 0-persistent variables, the size of the problem and cpu time reductions. The coefficient n_w is the dimension of the space, n_v the number of constraints defining the polytope P. The mean value μ and the standard deviation σ of the number of 0-persistent variables of 10 randomly generated problems of density parameter Δ are given in the column labeled *0-pers*. Computing times in seconds spent by the corresponding preprocessing are given in the column labeled *prep*. The columns labeled *cpu1* and *cpu2* corresponds to the solution times of the problem before and after the preprocessing respectively. The time reductions are given in the column labeled *t.r.* and corresponds to the difference $\mu(cpu1) - (\mu(cpu2) + \mu(prep))$ divided by $\mu(cpu1)$. The size reductions are given in the column labeled *s.r.*. These numerical experiments show that our preprocessing procedure vastly decrease the size of the problem and the computing time to solve it.

$n = dim$	$Nconstr$	density Δ	$\mu \pm \sigma$	reduct
10	10	10	4.70 ± 1.42	47.0 %
10	10	30	5.30 ± 2.50	53.0 %
10	10	50	5.90 ± 3.11	59.0 %
10	10	90	5.20 ± 2.74	52.0 %
20	20	10	11.80 ± 4.02	59.0 %
20	20	30	12.90 ± 3.38	64.5 %
20	20	50	7.80 ± 4.39	39.0 %
20	20	90	6.60 ± 3.81	33.0 %
50	50	10	39.00 ± 5.03	78.0 %
50	50	30	29.20 ± 9.45	58.4 %
50	50	50	20.60 ± 10.48	41.2 %
50	50	90	17.70 ± 8.99	35.4 %
70	70	10	60.10 ± 9.93	85.8 %
70	70	30	40.30 ± 13.37	57.5 %
70	70	50	38.40 ± 16.65	54.8 %
70	70	90	22.00 ± 13.27	31.4 %
100	100	10	81.50 ± 11.09	81.51 %
100	100	30	57.30 ± 17.23	57.3 %
100	100	50	47.30 ± 24.94	47.3 %
100	100	90	22.20 ± 10.17	22.2 %
150	150	10	134.00 ± 10.64	89.3 %
150	150	30	103.00 ± 18.92	68.6 %
150	150	50	79.30 ± 20.76	52.8 %

Table 5.1: 0-persistent variables $\underline{c}_j$ and $\overline{c}_j$ chosen between 0 and $2n$

$n = dim$	$Nconstr$	density Δ	$\mu \pm \sigma$	reduct
10	10	10	3.50 ± 2.17	35.00 %
10	10	30	3.70 ± 2.16	37.00 %
10	10	50	3.80 ± 1.69	38.00 %
10	10	90	3.50 ± 2.88	35.00 %
20	20	10	7.00 ± 4.37	35.00 %
20	20	30	6.60 ± 4.74	33.00 %
20	20	50	4.20 ± 4.32	21.00 %
20	20	90	2.90 ± 3.00	14.50 %
50	50	10	28.00 ± 10.20	56.00 %
50	50	30	12.20 ± 6.00	24.40 %
50	50	50	5.50 ± 4.30	11.00 %
50	50	90	2.90 ± 3.51	5.80 %
70	70	10	39.50 ± 12.64	56.43 %
70	70	30	12.30 ± 7.32	17.57 %
70	70	50	7.00 ± 6.27	10.00 %
70	70	90	3.90 ± 4.77	5.57 %
100	100	10	55.90 ± 11.96	55.90 %
100	100	30	22.10 ± 18.28	22.10 %
100	100	50	7.10 ± 8.95	7.10 %
100	100	90	1.20 ± 1.23	1.20 %
150	150	10	97.40 ± 18.73	64.93 %
150	150	30	43.30 ± 26.76	28.87 %
150	150	50	19.50 ± 16.22	13.00 %

Table 5.2: 0-persistent variables, $\underline{c}_j$ and $\overline{c}_j$ chosen between 0 and n

$n = dim$	$Nconstr$	density Δ	$\mu \pm \sigma$	reduct
10	10	10	2.10 ± 2.28	21.00 %
10	10	30	3.00 ± 1.63	30.00 %
10	10	50	2.60 ± 1.84	26.00 %
10	10	90	2.50 ± 2.92	25.00 %
20	20	10	4.60 ± 4.17	23.00 %
20	20	30	4.00 ± 2.05	20.00 %
20	20	50	2.00 ± 1.76	10.00 %
20	20	90	2.80 ± 3.19	14.00 %
50	50	10	13.50 ± 7.55	27.00 %
50	50	30	3.50 ± 3.10	7.00 %
50	50	50	1.50 ± 1.90	3.00 %
50	50	90	1.40 ± 1.35	2.80 %
70	70	10	18.50 ± 10.46	26.43 %
70	70	30	3.60 ± 3.27	5.14 %
70	70	50	1.40 ± 1.96	2.00 %
70	70	90	0.20 ± 0.42	0.29 %
100	100	10	15.60 ± 13.04	15.60 %
100	100	30	2.80 ± 2.15	2.80 %
100	100	50	0.20 ± 0.42	0.20 %
100	100	90	0.20 ± 0.63	0.20 %
150	150	10	35.00 ± 22.89	23.33 %
150	150	30	2.90 ± 3.96	1.93 %
150	150	50	0.80 ± 1.75	0.53 %

Table 5.3: 0-persistent variables, $\underline{c}_j$ and $\overline{c}_j$ chosen between 0 and $\frac{n}{2}$

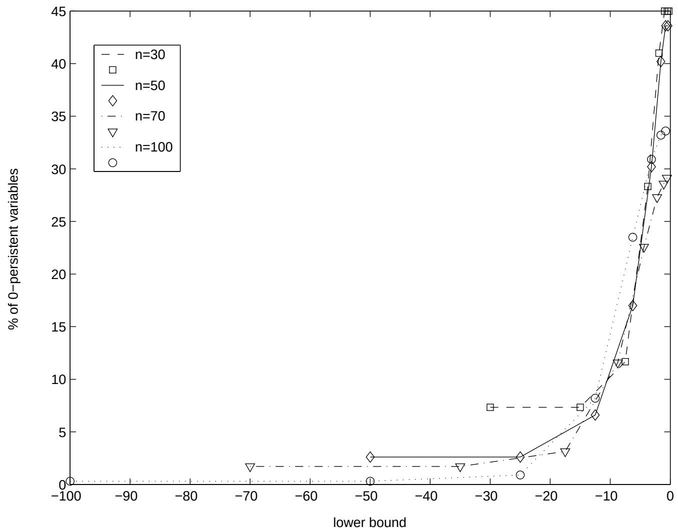

Figure 5.9: 0-persistent variables, $\underline{c}_j$ and $\overline{c}_j$ chosen between $-M$ and $2n$, $\Delta = 0.3$

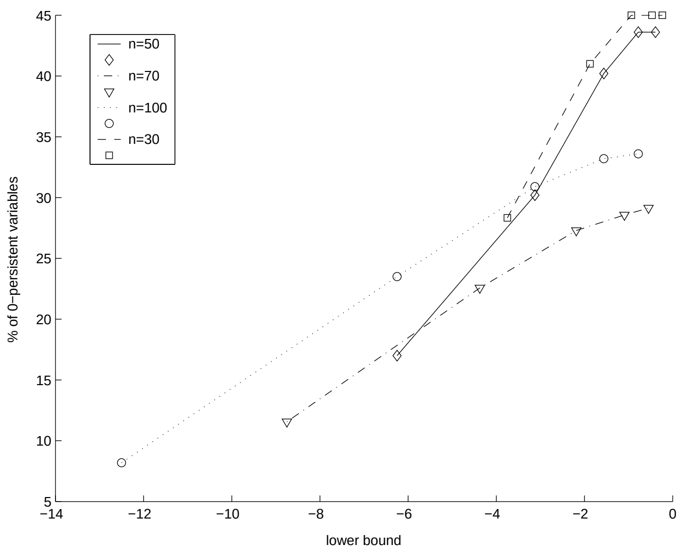

Figure 5.10: Detail of Figure 5.9 when M is close to 0

n_w n_v	Δ	0-pers $\mu \pm \sigma$	s.r. %	cpu1 $\mu \pm \sigma$	prep $\mu \pm \sigma$	cpu2 $\mu \pm \sigma$	t.r. %
50	30	25.40 ±12.56	50.80	300.09 ±204.76	2.16 ±0.09	86.73 ±83.64	70.38
50	70	23.80 ±11.82	47.60	781.82 ±822.19	3.74 ±0.17	258.94 ±398.90	66.40
100	30	59.20 ±24.15	59.20	2118.35 ±2045.58	12.78 ±0.21	845.02 ±858.25	59.51
100	70	30.40 ±12.93	30.40	3922.09 ±1726.87	27.40 ±0.79	3658.51 ±1507.27	6.02

Table 5.4: The minimax regret interval UOLPP algorithm and the performance of the preprocessing.

Conclusion and extensions

Under the classical definitions of robustness, given in Kouvelis and Yu (1997), the robust versions of most classical polynomial combinatorial optimization problems becomes NP-hard. Hence, investigating the ways in which the solution space can be reduced is an important issue. A way to reduce the size of these problems is to study when a robust solution is optimal for a scenario and to detect when a decision variable is always or never part of an optimal solution for all realization of data (1-persistent or 0-persistent variables respectively). One of the main goals of the first part of this thesis was to investigate these questions in some combinatorial optimization problems.

We considered the minimum spanning tree problem under compact and convex uncertainty. We presented localization results for scenarii yielding to the largest regret for a tree and in the case of interval uncertainty, we obtained characterizations of 1-persistent and 0-persistent edges and provided polynomial algorithms for detecting them.

We investigated the uncertain shortest path and the uncertain single-source shortest path problems, both of them on finite directed graphs where arcs lengths are nonnegative intervals. For the shortest path problem under interval uncertainty we obtained sufficient conditions for a node to be 0-persistent and for an arc to be 0- or 1- persistent. For the uncertain single-source shortest path problem, we presented sufficient conditions for an arc to be 1- or 0-persistent. Such conditions are based on the topology of the graph combined with the structure of the uncertainty set. Based on these results we presented polynomial time recognition algorithms that we used to preprocess a given graph with interval uncertainty prior to the solution of the minimax regret problem. In order to test our algorithms, we proposed a mixed integer programming formulation to solve the minimax regret single-source shortest path problem under interval uncertainty. We showed by numerical experiments that such preprocessing procedures greatly reduce the time to compute a minimax regret solution.

The choice of which element of the problem is most convenient to detect for persistency, depends on the combinatorial structure of the problem. We observed for instance, that in the shortest path problem under interval uncertainty it is much faster checking for 0-persistent nodes than 0-persistent arcs.

We also investigated the minimax regret linear programming problem under uncertainty in the objective function coefficients. We studied the problem under compact and convex uncertainty and we presented some of its properties. We presented an alternative proof to the one given in Averbakh and Lebedev (2005) about the NP-hardness of the maximum regret interval problem. We presented special cases when the maximum regret and the minimax regret problems are polynomial. We presented an exact algorithm to solve the minimax regret polyhedral problem. Numerical results were given, in the case of polyhedral and interval uncertainty.

We established a link between the restricted 1-center problem and the minimax regret linear programming problem under interval uncertainty in the objective function coefficients. We described the underlying geometry of this last problem and we derived an approach to the restricted 1-center problem based on the regret of a special class of unbounded closed convex sets. Moreover, we discussed an application of this model in location theory. In the case when the polytope is full dimensional and all the constraints are non redundant, we established conditions under which 0-persistent variables can be eliminated from the problem. We tested these condition on randomly generated instances and the numerical results showed that we can greatly reduce the size and the computing time to solve a minimax regret problem under interval uncertainty.

Future considerations in this area would involve the following aspects and questions.

- Consider a uncertain combinatorial optimization problem where $\mathbf{c} \in D$. Under what conditions over D and for which robust counterparts, a robust solution is a weak solution?
- The study of conditions for a variable to be 1- and 0-persistent for other combinatorial problems.
- Verifying the performance of the preprocessing that consists to detect 1-persistent arcs and nodes.
- In the case of UCOP under polyhedral uncertainty, a minimax regret solution is weak solution?

- The study of a link between the minimax regret linear programming problem under polyhedral uncertainty in the objective function coefficients and another 1-center problem, for example the 1-center problem under an special type of gauge.

- The study of the *positive persistent variables*, that is, the variables that are always positive, for all optimal solution and for all set of coefficients to the UOLPP.

- Verifying the performance of our minimax regret algorithm to solve 1-center problems under gauges.

- There is a class of problems for which the decision must resist to the repetitive change of conditions. We refer the reader to Roy (2007) for the motivation. Studying the properties of these problems, defining new robust counterparts and construct algorithms to solve it are also possible future research directions and warrant additional analysis.

Bibliography

H. Aissi. *Approximation et resolution des versions min-max et min-max regret de problemes d'optimisation combinatoire*. PhD thesis, Université Paris Dauphine, 2005.

H. Aissi, C. Bazgan, and D. Vanderpooten. Complexity of the min-max (regret) versions of min cut problems. *Discrete optimization*, 5:66–73, 2008.

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

S. Alarie, C. Audet, B. Jaumard, and G. Savard. Concavity cuts for disjoint bilinear programming. *Mathematical Programming*, 90:373–398, 2001.

I. Aron and P. Hentenryck. On the complexity of the robust spanning tree problem. *Operations Research Letters*, 32(1):36–40, 2004.

A. Atamturk. Strong formulations of robust mixed 0-1 programming. *Mathematical Programming*, series B, 108:235–250, 2006.

C. Audet, P. Hansen, B. Jaumard, and G. Savard. A symmetrical linear maxmin approach to disjoint bilinear programming. *Mathematical Programming*, 85:573–592, 1999.

I. Averbakh. Minmax regret solutions for minmax optimization problems with uncertainty. *Operations Research Letters*, 27(2):57–65, 2000.

I. Averbakh. On the complexity of a class of combinatorial optimization problems with uncertainty. *Mathematical Programming*, 90(2):263–272, 2001.

I. Averbakh and V. Lebedev. Interval data minmax regret network optimization problems. *Discrete Applied Mathematics*, 138(3):289–301, 2004.

I. Averbakh and V. Lebedev. On the complexity of minmax regret linear programming. *European Journal of Operations Research*, 160(1):227–231, 2005.

A. Ben-Tal and A. Nemirovsky. Robust convex optimization. *Operations Research*, 23:769–805, 1998.

A. Ben-Tal and A. Nemirovsky. Robust solutions of uncertain linear problems. *Operations Research Letters*, 25:1–13, 1999.

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

A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovsky. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99:351–376, 2004.

D. Bertsimas and M. Sim. Robust discrete optimization under ellipsoidal uncertainty sets. Technical report, Operations Research Center, MIT, 2004a.

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

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

D. Bertsimas and J.N. Tsitsiklis. *Introduction to Linear optimization*. Athena Scientific, Belmont, Massachusetts, 1997.

D. Bertsimas, K. Natarajan, and Ch. P. Teo. Persistence in discrete optimization under data uncertainty. Technical report, Operations Research Center, MIT, 2005.

J.A. Bondy and U.S.R. Murty. *Graph Theory with applications*. American Elsevier, New York, 1976.

S. Chanas and P. Zielinski. On the hardness of evaluating criticality of activities in a planar network with duration intervals. *Operations Research Letters*, 31(1):53–59, 2003.

E. Conde. A branch and bound algorithm for the minimax regret spanning arborescence. *Journal of Global Optimization*, 37:467–480, 2007.

T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. The MIT Press, Cambridge, Massachusetts and McGraw-Hill Book Company, Boston, 1991.

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

L. El-Ghaoui, F. Oustry, and H. Lebret. Robust solutions to uncertain semidefinite programs. *SIAM Journal on Optimization*, 9(1):32–52, 1998.

R.L. Francis, McGinnis L.F., and J.A. White. *Facility Layout and Location: an analytical approach*. Englewood Cliffs, New Jersey, Prentice-Hall, 1992.

A.P. Goldman. A theorem on convex programming. *Paper presented at M.A.A. Meeting, Annapolis*, 1963.

P. Hansen, D. Peeters, and J.F. Tisse. An algorithm for a constrainted weber problem. *Management Science*, 28(1):1285–1295, 1982.

Y. Hinojosa and J. Puerto. Single facility location problems with unbounded unit balls. *Mathematical Methods of Operations Research*, 58:87–104, 2003.

A.P. Hurter, M.K. Schaefer, and R.E. Wendell. Solutions of constrainted location problems. *Management Science*, 22(1):51–56, 1975.

M. Inuiguchi and M. Sakawa. Maximum regret analysis in linear programs with an interval objective function. *Proceedings of IWSCI*, pages 308–317, 1996.

M. Inuiguchi and M. Sakawa. Minimax regret solution to linear programming problems with interval objective function. *European Journal of Operations Research*, 86(3):526–536, 1995.

O.E. Karasan, M.C. Pinar, and H. Yaman. The robust shortest path problem with interval data. Technical report, Available from http://www.optimization-online.org., 2001.

A. Kasperski and P. Zielinski. The robust shortest path problem problem in series-parallel miltidigraphs with interval data. *Operations Research Letters*, 34(1):69–76, 2006.

A. Kasperski and P. Zielinski. An approximation algorithm for interval data minmax regret combinatorial problems. *Instytut Matematyki Politechnika Wroclawska, Wroclaw*, raport serii preprinty nr 044, 2004.

H. Konno. A cutting plane algorithm for solving bilinear programs. *Mathematical Programming*, 11:14–27, 1976.

P. Kouvelis and G. Yu. *Robust Discrete Optimization and its Applications.* Kluwer Academic Publishers, Boston, 1997.

R.F. Love, J.G. Morris, and G.O. Wesolowsky. *Facilities Location: Models and Methods.* North-Holland, New York, 1988.

H.E. Mausser and M. Laguna. A new mixed integer formulation for the maximum regret problem. *International Transactions in Operational Research*, 5(5):389–403, 1998.

H.E. Mausser and M. Laguna. A heuristic to minmax absolute regret for linear programs with interval objective function coefficients. *European Journal of Operations Research*, 117(1):157–174, 1999.

N. Megiddo. Linear programming in linear time when the dimension is fixed. *Journal of the Association for Computing Machinery*, 31:114–127, 1984.

H. Minkowski. *Gesammelte Abhandlugen*, volume Band 2. Chelsea Publishing Company, New York, 1967.

M. Minoux. *Programmaton mathématique, théorie et algorithmes.* Dounod, Paris, 1983.

R. Montemanni. A Benders decomposition approach for the robust spanning tree with interval data. *European Journal of Operations Research*, 174(3): 1479–1490, 2006.

R. Montemanni and L.M. Gambardella. The robust shortest path problem with interval data problem with interval data via Benders decomposition. *4OR*, 3:315–328, 2005a.

R. Montemanni and L.M. Gambardella. A branch and bound algorithm for the robust spanning tree problem with interval data. *European Journal of Operations Research*, 161(3):771–779, 2005b.

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

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

J. Mulvay, R. Vanderbei, and S. Zenios. Robust optimization of large scale systems. *Operations Research*, 43(2):264–281, 1995.

S. Nickel. Restricted center problems under polyhedral gauges. *European Journal of Operational Research*, 104:343–357, 1998.

S. Nickel and J. Puerto. *Location Theory: A unified approach*. Springer-Verlag, Berlin Heidelberg, 2005.

S. Nickel and A. Schobel. A geometric approach to global optimization. *Journal of Global Optimization*, 15:109–126, 1999.

B. Roy. La robustesse en recherche opérationnelle et aide à la décision : une préoccupation multi-facettes. *Annales du Lamsade, Université Paris Dauphine*, 7:209–235, 2007.

M. Salazar-Neumann. The robust minimum spanning tree problem: Compact and convex uncertainty. *Operations Research Letters*, 35(1):17–22, 2007a.

M. Salazar-Neumann. The robust shortest path and the single source shortest path problems: interval data. Technical report, Annales du Lamsade, Université Paris Dauphine, 2007b.

A.L. Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21:1154–1157, 1973.

P. Vallin. Recherche de conclusions robustes dans une problematique de placements financiers en environnement incertain. Technical report, Annales du Lamsade, Université Paris Dauphine, 2007.

J.E. Ward and R.E. Wendell. Using block norms for location modeling. *Operations Research*, 33:1074–1090, 1985.

R.E. Wendell and A.P. Hurter. Location theory, dominance, and convexity. *Operations Research*, 21,1:314–320, 1973.

H. Yaman, O.E. Karasan, and M.C. Pinar. The robust spanning tree problem with interval data. *Operations Research Letters*, 29(1):31–40, 2001.

P. Zielinski. The computational complexity of the relative robust shortest path problem with interval data. *European Journal of Operations Research*, 158:570–576, 2004.