# Min-max and min-max regret versions of some combinatorial optimization problems: a survey

Hassene Aissi, Cristina Bazgan, Daniel Vanderpooten

# Min-max and min-max regret versions of some combinatorial optimization problems : a survey

Hassene Aissi*, Cristina Bazgan*, and Daniel Vanderpooten*

**Résumé**

Les critères min-max et min-max regret sont souvent utilisés en vue d'obtenir des solutions robustes. Après avoir motivé l'utilisation de ces critères, nous présentons des résultats généraux. Ensuite, nous donnons des résultats de complexité des versions min-max et min-max regret de quelques problèmes d'optimisation combinatoire : plus court chemin, arbre couvrant, affectation, coupe, $s - t$ coupe, sac à dos. Comme la plupart de ces problèmes sont *NP*-difficiles, nous présentons des algorithmes d'approximation ainsi que des algorithmes exacts de résolution.

**Mots-clefs :** Min-max, min-max regret, optimisation combinatoire, complexité, approximation, analyse de robustesse.

**Abstract**

Min-max and min-max regret criteria are commonly used to define robust solutions. After motivating the use of these criteria, we present general results. Then, we survey complexity results for the min-max and min-max regret versions of some combinatorial optimization problems: shortest path, spanning tree, assignment, cut, s-t cut, knapsack. Since most of these problems are *NP*-hard, we also investigate the approximability of these problems. Furthermore, we present algorithms to solve these problems to optimality.

**Key words :** Min-max, min-max regret, combinatorial optimization, complexity, approximation, robustness analysis.

*LAMSADE, Université Paris-Dauphine, 75775 Paris cedex 16, France. {aissi,bazgan,vdp}@lamsade.dauphine.fr

# 1 Introduction

The definition of an instance of a combinatorial optimization problem requires to specify parameters, in particular coefficients of the objective function, which may be uncertain or imprecise. Uncertainty/imprecision can be structured through the concept of *scenario* which corresponds to an assignment of plausible values to model parameters. There exist two natural ways of describing the set of all possible scenarios. In the *discrete scenario case*, the scenario set is described explicitly. In the *interval scenario case*, each numerical parameter can take any value between a lower and an upper bound. The min-max and min-max regret criteria, stemming from decision theory, are often used to obtain solutions hedging against parameters variations. The min-max criterion aims at constructing solutions having the best possible performance in the worst case. The min-max regret criterion, less conservative, aims at obtaining a solution minimizing the maximum deviation, over all possible scenarios, between the value of the solution and the optimal value of the corresponding scenario.

The study of these criteria is motivated by practical applications where an anticipation of the worst case is crucial. For instance, consider the sensor placement problem when designing contaminant warning systems for water distribution networks [16, 47]. A key deployment issue is identifying where the sensors should be placed in order to maximize the level of protection. Quantifying the protection level using the expected impact of a contamination event is not completely satisfactory since the issue of how to guard against potentially high-impact events, with low probability, is only partially handled. Consequently, standard approaches, such as deterministic or stochastic approaches, will fail to protect against exceptional high-impact events (earthquakes, hurricanes, 9/11-style attacks,…). In addition, reliable estimation of contamination event probabilities is extremely difficult. Min-max and min-max regret criteria are appropriate in this context by focusing on a subset of high-impact contamination events, and placing sensors to minimize the impact of such events.

The purpose of this paper is to review the existing literature on the min-max and min-max regret versions of combinatorial optimization problems with an emphasis on the complexity, the approximation and the exact resolution of these problems, both for the discrete and interval scenario cases.

The rest of the paper is organized as follows. Section 2 introduces, illustrates and motivates the min-max and min-max regret criteria. General results for these two criteria are presented in section 3. Section 4 provides complexity results for the min-max and min-max regret versions of various combinatorial optimization problems. Since most of these problems are *NP*-hard, section 5 describes the approximability of these problems. Section 6 describes exact procedures to solve min-max and min-max regret versions in the discrete and interval scenario cases. Conclusions are provided in the final section.

# 2 Presentation and motivations

We consider in this paper the class $\mathcal{C}$ of 0-1 problems with a linear objective function defined as:

$$\begin{cases} \min(\text{or } \max) \sum_{i=1}^{n} c_i x_i & c_i \in \mathbb{N} \\ x \in X \subset \{0,1\}^n \end{cases}$$

This class encompasses a large variety of classical combinatorial problems, some of which are polynomial-time solvable (shortest path problem, minimum spanning tree, ...) and others are *NP*-difficult (knapsack, set covering, ...).

In the following, all the definitions and results are presented for minimization problems $\mathcal{P} \in \mathcal{C}$. In general, they also hold with minor modifications for maximization problems, except for some cases that will be explicitly mentioned.

## 2.1 Min-max, min-max regret versions

In the discrete scenario case, the min-max or min-max regret version associated to a minimization problem $\mathcal{P} \in \mathcal{C}$ has as input a finite set $S$ of scenarios where each scenario $s \in S$ is represented by a vector $c^s = (c_1^s, \ldots, c_n^s)$.

In the interval scenario case, each coefficient $c_i$ can take any value in the interval $[\underline{c}_i, \overline{c}_i]$. In this case, the scenario set $S$ is the cartesian product of the intervals $[\underline{c}_i, \overline{c}_i]$, $i = 1, \ldots, n$. An *extreme scenario* is a scenario where $c_i = \underline{c}_i$ or $\overline{c}_i$, $i = 1, \ldots, n$.

We denote by $val(x, s) = \sum_{i=1}^{n} c_i^s x_i$ the value of solution $x \in X$ under scenario $s \in S$, by $x_s^*$ an optimal solution under scenario $s$, and by $val_s^* = val(x_s^*, s)$ the corresponding optimal value.

The min-max version corresponding to $\mathcal{P}$ consists of finding a solution having the best worst case value across all scenarios, which can be stated as:

$$\min_{x \in X} \max_{s \in S} val(x, s) \tag{1}$$

This version is denoted by DISCRETE MIN-MAX $\mathcal{P}$ in the discrete scenario case, and by INTERVAL MIN-MAX $\mathcal{P}$ in the interval scenario case.

Given a solution $x \in X$, its *regret*, $R(x, s)$, under scenario $s \in S$ is defined as $R(x, s) = val(x, s) - val_s^*$. The *maximum regret* $R_{max}(x)$ of solution $x$ is then defined as $R_{max}(x) = \max_{s \in S} R(x, s)$.

The min-max regret version corresponding to $\mathcal{P}$ consists of finding a solution minimizing its maximum regret, which can be stated as:

$$\min_{x \in X} R_{max}(x) = \min_{x \in X} \max_{s \in S} (val(x, s) - val_s^*) \tag{2}$$

This version is denoted by DISCRETE MIN-MAX REGRET $\mathcal{P}$ in the discrete scenario case, and by INTERVAL MIN-MAX REGRET $\mathcal{P}$ in the interval scenario case.

For maximization problems, corresponding max-min and min-max regret versions can be defined.

## 2.2 An illustrative example

In order to illustrate the previous definitions and to show the interest of using the min-max and min-max regret criteria, we consider a capital budgeting problem with uncertainty or imprecision on the expected profits. Suppose we wish to invest a capital $b$ and we have identified $n$ investment opportunities. Investment $i$ requires a cash outflow $w_i$ at present time and yields an expected profit $p_i$ at some term, $i = 1, \ldots, n$. Due to various exogenous factors (evolution of the market, inflation conditions, ...), the profits are evaluated with uncertainty/imprecision.

The capital budgeting problem is a knapsack problem where we look for a subset $I \subseteq \{1, \ldots, n\}$ of items such that $\sum_{i \in I} w_i \leq b$ which maximizes the total expected profit, i.e., $\sum_{i \in I} p_i$.

Depending on the type of uncertainty or imprecision, we use either discrete or interval scenarios. When the profits of investments are influenced by the occurrence of well-defined future events (e.g., different levels of market reactions, public decisions of constructing or not a facility which would impact on the investment projects), it is natural to define discrete scenarios corresponding to each event. When the evaluation of the profit is simply imprecise, a definition using interval scenarios is more appropriate.

As an illustration, consider a small size capital budgeting problem where profit uncertainty or imprecision is modelled with three discrete scenarios or with interval scenarios. Numerical values are given in Table 1 with $b = 12$ and $n = 6$.

| $i$ | $w_i$ | $p_i^1$ | $p_i^2$ | $p_i^3$ | $\underline{p}_i$ | $\overline{p}_i$ |
|---|---|---|---|---|---|---|
| 1 | 3 | 4 | 3 | 3 | 3 | 5 |
| 2 | 5 | 8 | 4 | 6 | 2 | 6 |
| 3 | 2 | 5 | 3 | 3 | 2 | 5 |
| 4 | 4 | 3 | 2 | 4 | 2 | 3 |
| 5 | 5 | 2 | 8 | 2 | 3 | 9 |
| 6 | 3 | 4 | 6 | 2 | 1 | 7 |

Table 1: Cash outflows and profits of the investments

| 1 | 2 | 3 | Optimal solution |
|---|---|---|---|
| 17 | 10 | 12 | scenario 1: (1,1,1,0,0,0) |
| 11 | 17 | 7 | scenario 2: (0,0,1,0,1,1) |
| 16 | 9 | 13 | scenario 3: (0,1,1,1,0,0) |
| 15 | 12 | 12 | max-min: (0,1,0,1,0,1) |
| 15 | 15 | 11 | min-max regret: (0,1,1,0,1,0) |

Table 2: Optimal values and solutions (discrete scenario case)

In the discrete scenario case, we observe in Table 2 that the optimal solutions of the three scenarios are not completely satisfactory since they give low profits in some scenarios. An appropriate solution should behave well under any variation of the future profits. In this example, optimal solutions to max-min and min-max regret versions are more acceptable solutions since their performances are more stable. A risk-averse decision maker would favor the max-min solution that guarantees at least 12 in all scenarios. If the decision maker is willing to accept a small degradation of the performance in the worst case scenario with an increase of the average performance over all scenarios, then the optimal solution of the min-max regret version is more appropriate than the optimal solution of the min-max version.

In the interval scenario case, the optimal solution to the max-min version is obtained by considering only the worst-case scenario (see section 3.2) and corresponds to $x_1 = x_3 = x_5 = 1$; $x_2 = x_4 = x_6 = 0$ with a worst value 8. The obtained solution only depends on one scenario and, in particular, is completely independent of the interval upper bound values. An optimal solution of the min-max regret version is given by $x_1 = x_5 = x_6 = 1$; $x_2 = x_3 = x_4 = 0$ with a worst value and a maximum regret 7. Unlike for the max-min criterion, the optimal min-max regret solution does depend on the interval lower and upper bound values. In particular, when an interval upper bound increases, the maximum regret of each feasible solution either increases or remains stable. Obviously, this may impact on the resulting optimal solution. For instance, if we increase $\overline{p}_3$ from 5 to 8, the new min-max regret optimal solution becomes $x_3 = x_5 = x_6 = 1$; $x_1 = x_2 = x_4 = 0$ with a worst value 6 and a maximum regret 8. Observe, in the previous example, that improving the prospects of investment 3 leads to include this investment in the new optimal solution.

## 2.3 Relevance and limits of the min-max (regret) criteria

When uncertainty or imprecision on the parameter values of decision aiding models is a crucial issue, decision makers may not feel confident using results derived from parameters taking precise values. *Robustness analysis* is a theoretical framework that enables the decision maker to take into account uncertainty or imprecision in order to produce decisions that will behave reasonably under any likely input data (see, e.g., Roy [42], Vincke [45] for general contexts, Kouvelis and Yu [30] in combinatorial optimization, and Mulvey, Verderbel, and Zenios [39] in mathematical programming). Different criteria can be used to select among robust decisions. The min-max and min-max regret criteria are often used to obtain conservative decisions to hedge against variations of the input data [41]. We briefly review situations where each criterion is appropriate.

The min-max criterion is suited for non-repetitive decisions (construction of a high voltage line, highway, . . . ) and for decision environments where precautionary measures are needed (nuclear accidents, public health). In such cases, classical approaches like deterministic or stochastic optimization are not relevant. This criterion is also appropriate in competitive situations [41] or when the decision maker must reach a pre-defined goal (sales, inventory, . . . ) under any variation of the input data.

The min-max regret criterion is suitable in situations where the decision maker may feel regret if he/she makes a wrong decision. He/she thus takes this anticipated regret into account when deciding. For instance, in finance, an investor may observe not only his own portfolio performance but also returns on other stocks or portfolios in which he was able to invest but decided not to [20]. Therefore, it seems very natural to assume that the investor may feel joy/disappointment if his own portfolio outperformed/underperformed some benchmark portfolio or portfolios. The min-max regret criterion reflects such a behavior and is less conservative than the min-max criterion since it considers missed opportunities. It has also been shown that, in the presence of uncertainty on prices and yields, the behavior of some economical agents (farmers) could sometimes be better predicted using a min-max regret criterion, rather than a classical profit maximization criterion [29].

Maximum regret can also serve as an indicator of how much the performance of a decision can be improved if all uncertainties/imprecisions could be resolved [30]. The min-max regret criterion is relevant in situations where the decision maker is evaluated ex-post.

Another justification of the min-max regret criterion is as follows. In uncertain/imprecise decision contexts, a reasonable objective is to find a solution with performances as close as possible from the optimal values under all scenarios. This amounts to setting a threshold $\varepsilon$ and looking for a solution $x \in X$ such that $val(x,s) - val_s^* \leq \varepsilon$, for all $s \in S$. Equivalently, $x$ should satisfy $R_{max}(x) \leq \varepsilon$. Then, looking for such a solution with $\varepsilon$ as small as possible is equivalent to determining a min-max regret solution.

Min-max and min-max regret criteria are simple to use since they do not require any additional information unlike other approaches based on probability or fuzzy set theory. Furthermore, they are often considered as reference criteria and a starting point in robustness analysis. However, the min-max and min-max regret criteria are sometimes inappropriate. In some situations, these criteria are too pessimistic for decision makers who are willing to accept some degree of risk. In addition, they attach a great importance to worst case scenarios which are sometimes unlikely to occur. In the discrete scenario case, this difficulty could be handled, at the modelling stage, by including only relevant scenarios in the scenario set. In the interval scenario case, min-max and min-max regret optimal solutions are obtained for extreme scenarios, as will be shown in section 3.2. Clearly not all these scenarios are likely to happen and this limits the applicability of these criteria. This difficulty becomes all the more important than the intervals get larger.

Some approaches have been designed so as to reduce these drawbacks. In the discrete scenario case, Daskin, Hesse, and ReVelle [19] propose a model called the $\alpha$-reliable min-max regret model that identifies a solution that minimizes the maximum regret with respect to a selected subset of scenarios whose probability of occurrence is at least some user-defined value $\alpha$. Kalai, Aloulou, Vallin, and Vanderpooten [25] propose another approach called lexicographic $\alpha$-robustness which, instead of focusing on the worst case, considers all scenarios in lexicographic order, from the worst to the best, and also includes a tolerance threshold $\alpha$ in order not to discriminate among solutions with similar values. In the interval scenario case, Bertsimas and Sim [15] propose a robustness model with a scenario set restricted to scenarios where the number of parameters that are pushed to their upper bounds is limited by a user-specified parameter. The robust version obtained using this model has the same approximation complexity as the classical version. This is a clear advantage over the classical robustness criteria since their robust counterpart are generally *NP*-hard. The main difficulty is, however, the definition of this technical parameter whose value may impact on the resulting solution.

# 3 General results on min-max (regret) versions

We present in this section general results that give an insight on the nature and difficulty of min-max (regret) versions. They also serve as a basis for further results or algorithms presented in the next sections.

## 3.1 Discrete scenario case

### 3.1.1 Relationships between min-max (regret) and classical versions

We investigate, in this part, the quality of the solutions obtained by solving $\mathcal{P}$ for specific scenarios.

A first attempt to solve DISCRETE MIN-MAX (REGRET) $\mathcal{P}$ is to construct an optimal solution $x_s^*$ for each scenario $s \in S$, compute $\max_{s\in S} val(x_s^*, s)$ (or $R_{max}(x_s^*)$), and then, select the best of the obtained candidate solutions. However, this solution can be very bad, since the gap between its value and the optimal value may increase exponentially in the size of the instance as we can see in the following example. Consider for example a pathological instance of DISCRETE MIN-MAX (REGRET) SHORTEST PATH. Figure 1 depicts a graph $G = (V, A)$ with $2n$ vertices where each arc is valued by costs on 2 scenarios. The optimal solution in the first scenario is given by path $1, n+1, \ldots, 2n-1, 2n$ with values $(0, 2^n - 1)$, a worst value $2^n - 1$ and a maximum regret $2^n - 1$, whereas the optimal solution in the second scenario is given by path $1, 2, \ldots, n, 2n$ with values $(2^n - 1, 0)$, a worst value $2^n - 1$, and a maximum regret $2^n - 1$. However, the optimal solution of the min-max and min-max regret versions is given by path $1, 2n$ with a worst value 1 and a maximum regret 1. Thus the gap between the best value (regret) among the optimal solutions of each scenario and the optimum of min-max (min-max regret) is $2^n - 2$.

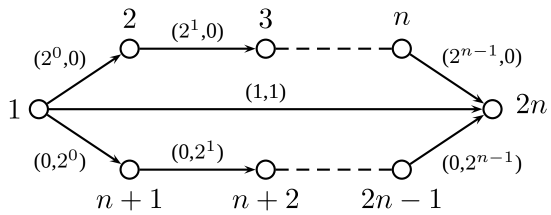


Figure 1: A critical instance of DISCRETE MIN-MAX (REGRET) SHORTEST PATH where optimal solutions in each scenario are very bad

The previous example shows that by solving $\mathcal{P}$ for all scenarios in *S*, we cannot guarantee the performance of the obtained solution. Another idea is to solve problem $\mathcal{P}$ for a fictitious scenario in the hope that the obtained solution has a good performance. The following proposition considers the *median* scenario, i.e. with costs corresponding to the average value over all scenarios.

**Proposition 1 ([30])** *Consider an instance $I$ of* DISCRETE MIN-MAX $\mathcal{P}$ *(or* DISCRETE MIN-MAX REGRET $\mathcal{P}$*) with $k$ scenarios where each scenario $s \in S$ is represented by $(c_1^s, \ldots, c_n^s)$ and where $\mathcal{P}$ is a minimization problem. Consider also an instance $I'$ of $\mathcal{P}$ where each coefficient of the objective function is defined by $c_i' = \sum_{s=1}^{k} \frac{c_i^s}{k}$, $i = 1, \ldots, n$. Then $x'$ an optimal solution of $I'$ is such that* $\max_{s \in S} val(x', s) \leq k.opt(I)$ *(or $R_{max}(x') \leq k.opt(I)$) where $opt(I)$ denotes the optimal value of instance $I$.*

**Proof :** Consider an instance $I$ of DISCRETE MIN-MAX REGRET $\mathcal{P}$. Let $x'$ be an optimal solution of $I'$. We first show that $L = \sum_{s \in S} \frac{1}{k}(val(x', s) - val_s^*)$ is a lower bound of $opt(I)$.

$$L = \min_{x \in X} \frac{1}{k} \sum_{s \in S} (val(x, s) - val_s^*) \leq \min_{x \in X} \frac{1}{k} k \max_{s \in S} (val(x, s) - val_s^*) = opt(I)$$

Moreover, $U = \max_{s\in S}(val(x', s) - val^*_s)$ is clearly an upper bound of $opt(I)$ and we get

$$\min_{x\in X}\max_{s\in S}(val(x,s) - val^*_s) \leq \max_{s\in S}(val(x',s) - val^*_s) \leq \sum_{s\in S}(val(x',s) - val^*_s)$$

$$= kL \leq k.opt(I)$$

The proof for DISCRETE MIN-MAX $\mathcal{P}$ is exactly the same except that we take $L = \sum_{s\in S}\frac{1}{k}val(x', s)$ and $U = \max_{s\in S} val(x', s)$ as lower and upper bounds of $opt(I)$, and we remove $val^*_s$ everywhere in the proof. $\square$

Considering a maximization problem $\mathcal{P}$, we can define lower and upper bounds $L$ and $U$ similarly. For the min-max regret version, we also obtain a $k$-approximate solution in this way. However, the previous result does not hold for the max-min version since the ratio $\frac{U}{L}$ is not bounded. In particular, for the knapsack problem, this ratio can be exponential in the size of the input, as noticed in [49].

An alternative idea is to solve problem $\mathcal{P}$ on a fictitious scenario, called *pessimistic* scenario, with costs corresponding to the worst values over all scenarios.

**Proposition 2** *Consider an instance $I$ of* DISCRETE MIN-MAX $\mathcal{P}$ *with $k$ scenarios where each scenario $s \in S$ is represented by $(c^s_1, \ldots, c^s_n)$ and where $\mathcal{P}$ is a minimization problem. Consider also an instance $I'$ of $\mathcal{P}$ where each coefficient of the objective function is defined by $c'_i = \max_{s\in S} c^s_i$, $i = 1, \ldots, n$. Then $x'$ an optimal solution of $I'$ is such that $\max_{s\in S} val(x', s) \leq k.opt(I)$ where $opt(I)$ denotes the optimal value of instance $I$.*

**Proof :** Let $x^*$ denote an optimal solution of instance $I$. We have

$$\max_{s\in S}\sum_{i=1}^{n} c^s_i x'_i \leq \sum_{i=1}^{n} c'_i x'_i \leq \sum_{i=1}^{n} c'_i x^*_i \leq \sum_{i=1}^{n}\sum_{s\in S} c^s_i x^*_i$$

$$= \sum_{s\in S}\sum_{i=1}^{n} c^s_i x^*_i \leq k.\max_{s\in S}\sum_{i=1}^{n} c^s_i x^*_i = k.opt(I)$$

$\square$

The same result cannot be extended to the min-max regret version. Figure 2 illustrates a critical instance of DISCRETE MIN-MAX REGRET SHORTEST PATH with two scenarios. The optimal solution for the first and second scenario is the same and corresponds to path $1, n-1, n$. Thus, this path is the min-max regret optimal solution with a maximum regret 0. However, the optimal solution for the pessimistic scenario is given by path $1, 2, \ldots, n$ with maximum regret $2^n$.

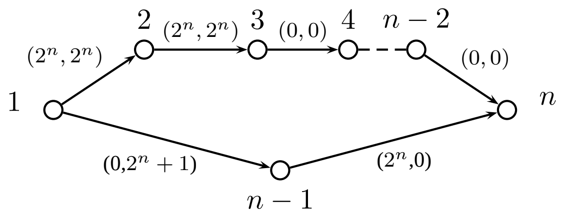


Figure 2: A critical instance of DISCRETE MIN-MAX REGRET SHORTEST PATH where the optimal solution in the pessimistic scenario is very bad

Extension of Proposition 2 to a maximization problem $\mathcal{P}$ does not hold neither for DISCRETE MAX-MIN $\mathcal{P}$ nor for DISCRETE MIN-MAX REGRET $\mathcal{P}$.

Quite interestingly, however, solving $\mathcal{P}$ for the pessimistic scenario leads to optimal solutions for the specific class of bottleneck problems defined as

$$\begin{cases} \min \max_{i=1,\dots,n} c_i x_i & c_i \in \mathbb{N} \\ x \in X \subset \{0,1\}^n \end{cases}$$

This class encompasses bottleneck versions of classical optimization problems (shortest path, spanning tree, ...). The min-max and min-max regret bottleneck versions are defined by (1) and (2) respectively, with $val(x,s) = \max_{i=1,\dots,n} c_i^s x_i$.

These versions are denoted DISCRETE MIN-MAX (REGRET) BOTTLENECK $\mathcal{P}$ in the discrete scenario case, and INTERVAL MIN-MAX (REGRET) BOTTLENECK $\mathcal{P}$ in the interval scenario case.

**Proposition 3** *Consider an instance $I$ of* DISCRETE MIN-MAX BOTTLENECK $\mathcal{P}$ *with $k$ scenarios where each scenario $s \in S$ is represented by $(c_1^s, \dots, c_n^s)$ and where $\mathcal{P}$ is a minimization problem. Consider also an instance $I'$ of $\mathcal{P}$ where each coefficient of the objective function is defined by $c_i' = \max_{s\in S} c_i^s$, $i = 1, \dots, n$. Then $x'$ an optimal solution of $I'$ is also an optimal solution of $I$.*

**Proof :**

$$\min_{x\in X} \max_{s\in S} val(x,s) = \min_{x\in X} \max_{s\in S} \max_{i=1,\dots,n} c_i^s x_i = \min_{x\in X} \max_{i=1,\dots,n} \max_{s\in S} c_i^s x_i = \min_{x\in X} \max_{i=1,\dots,n} c_i' x_i$$

□

The next proposition gives a reduction from DISCRETE MIN-MAX REGRET BOTTLENECK $\mathcal{P}$ to DISCRETE MIN-MAX BOTTLENECK $\mathcal{P}$.

**Proposition 4** *Consider an instance $I$ of* DISCRETE MIN-MAX REGRET BOTTLENECK $\mathcal{P}$ *with $k$ scenarios where each scenario $s \in S$ is represented by $(c_1^s, \ldots, c_n^s)$ and where $\mathcal{P}$ is a minimization problem. Consider also an instance $\widetilde{I}$ of* DISCRETE MIN-MAX BOTTLENECK $\mathcal{P}$ *where each coefficient of the objective function is defined by $\widetilde{c}_i^s = \max\{c_i^s - val_s^*, 0\}$, $i = 1, \ldots, n$. Then $\widetilde{x}$, an optimal solution of $\widetilde{I}$, is also an optimal solution of $I$.*

**Proof :**

$$\min_{x \in X} \max_{s \in S}(val(x,s) - val_s^*) = \min_{x \in X} \max_{s \in S}( \max_{i=1,\ldots,n} c_i^s x_i - val_s^*) = \min_{x \in X} \max_{s \in S} \max_{i=1,\ldots,n} \widetilde{c}_i^s x_i$$

□

The previous propositions are summarized in the following corollary.

**Corollary 1** *Given $\mathcal{P}$ an optimization problem,* DISCRETE MIN-MAX BOTTLENECK $\mathcal{P}$ *and* DISCRETE MIN-MAX REGRET BOTTLENECK $\mathcal{P}$ *reduce to $\mathcal{P}$.*

### 3.1.2 Relationships between min-max (regret) and multi-objective versions

It is natural to consider scenarios as objective functions. This leads us to investigate relationships between min-max (regret) and multi-objective versions.

The multi-objective version associated to $\mathcal{P} \in \mathcal{C}$, denoted by MULTI-OBJECTIVE $\mathcal{P}$, has for input $k$ objective functions where the $h$th objective function has coefficients $c_1^h, \ldots, c_n^h$. We denote by $val(x,h) = \sum_{i=1}^n c_i^h x_i$ the value of solution $x \in X$ on criterion $h$, and assume w.l.o.g. that all criteria are to be minimized. Given two feasible solutions $x$ and $y$, we say that $x$ dominates $y$ if $val(x,h) \leq val(y,h)$ for $h = 1, \ldots, k$ with at least one strict inequality. The problem consists of finding the set $E$ of efficient solutions. A feasible solution $x$ is *efficient* if there is no other feasible solution $y$ that dominates $x$. In general MULTI-OBJECTIVE $\mathcal{P}$ is intractable in the sense that it admits instances for which the size of $E$ is exponential in the size of the input (see, e.g., Ehrgott [21]).

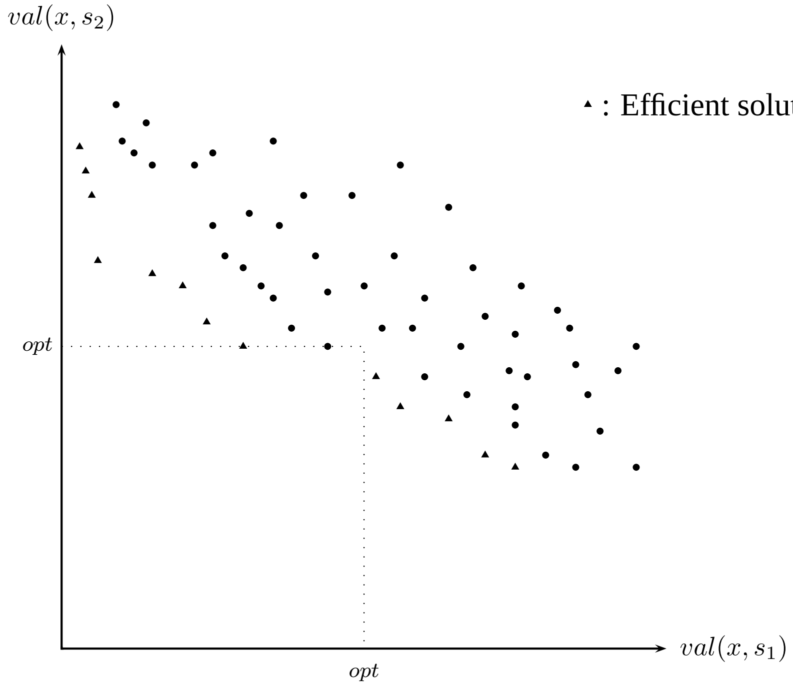


Figure 3: Relationships between min-max and multi-objective versions

**Proposition 5** *Given a minimization problem $\mathcal{P}$, at least one optimal solution for* DISCRETE MIN-MAX $\mathcal{P}$ *is necessarily an efficient solution.*

**Proof:** If $x \in X$ dominates $y \in X$ then $\max_{s \in S} val(x, s) \leq \max_{s \in S} val(y, s)$. Therefore, we obtain an optimal solution for DISCRETE MIN-MAX $\mathcal{P}$ by taking, among the efficient solutions, one that has a minimum $\max_{s \in S} val(x, s)$, see Figure 3. □

**Proposition 6** *Given a minimization problem $\mathcal{P}$, at least one optimal solution for* DISCRETE MIN-MAX REGRET $\mathcal{P}$ *is necessarily an efficient solution.*

**Proof:** If $x \in X$ dominates $y \in X$ then $val(x, s) \leq val(y, s)$, for each $s \in S$, and thus $R_{max}(x) \leq R_{max}(y)$. Therefore, we obtain an optimal solution for DISCRETE MIN-MAX REGRET $\mathcal{P}$ by taking, among the efficient solutions, a solution $x$ that has a minimum $R_{max}(x)$, see Figure 4. □

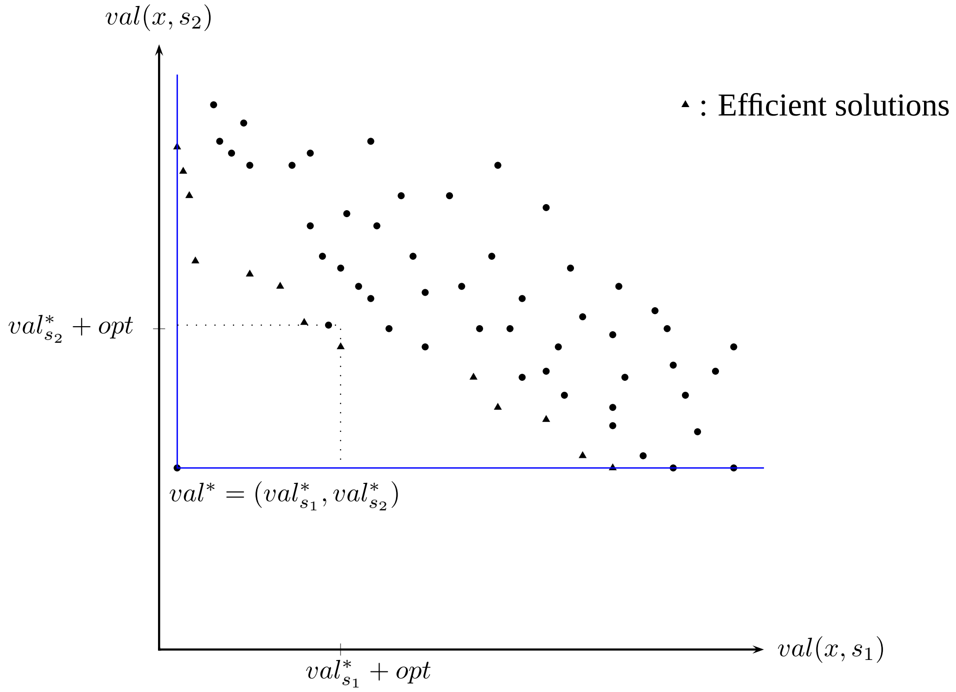


Figure 4: Relationships between min-max regret and multi-objective versions

Observe that if DISCRETE MIN-MAX (REGRET) $\mathcal{P}$ admit several optimal solutions, some of them may not be efficient, but at least one is efficient (see Figures 3 and 4).

## 3.2 Interval scenario case

For a minimization problem $\mathcal{P} \in \mathcal{C}$, solving INTERVAL MIN-MAX $\mathcal{P}$ is equivalent to solving $\mathcal{P}$ on the scenario $\overline{c} = (\overline{c}_1, \ldots, \overline{c}_n)$ where the values of all coefficients $c_i$ are set to their upper bounds (for a maximization problem, consider only scenario $\underline{c} = (\underline{c}_1, \ldots, \underline{c}_n)$). Therefore, the rest of the section is devoted to INTERVAL MIN-MAX REGRET $\mathcal{P}$.

In order to compute the maximum regret of a solution $x \in X$, we only need to consider its worst scenario $c^-(x)$. Yaman, Karaşan and Pinar [48] propose an approach to construct this scenario for INTERVAL MIN-MAX REGRET SPANNING TREE and characterize the optimal solution. These approaches can, however, be generalized to any problem $\mathcal{P} \in \mathcal{C}$.

**Proposition 7** *Given a minimization problem $\mathcal{P}$, the regret of a solution $x \in X$ is maximized for scenario $c^-(x)$ defined as follows:*

$$c_i^-(x) = \begin{cases} \overline{c}_i & \text{if } x_i = 1, \\ \underline{c}_i & \text{if } x_i = 0, \end{cases} \qquad i = 1, \ldots, n$$

**Proof :** Given a solution $x \in X$, let $I(x) = \{i \in 1, \ldots, n : x_i = 1\}$. For any scenario $s \in S$, we have

$$\begin{aligned} R(x, s) = val(x, s) - val(x^*_s, s) &= \sum_{i \in I(x) \setminus I(x^*_s)} c^s_i - \sum_{i \in I(x^*_s) \setminus I(x)} c^s_i \\ &\leq \sum_{i \in I(x) \setminus I(x^*_s)} \overline{c}_i - \sum_{i \in I(x^*_s) \setminus I(x)} \underline{c}_i \\ &= val(x, c^-(x)) - val(x^*_s, c^-(x)) \\ &\leq val(x, c^-(x)) - val(x^*_{c^-(x)}, c^-(x)) \\ &= R(x, c^-(x)) \end{aligned}$$

□

The following proposition shows that an optimal solution of INTERVAL MIN-MAX REGRET $\mathcal{P}$ is an optimal solution of $\mathcal{P}$ for one of the extreme scenarios.

**Proposition 8** *Given a minimization problem $\mathcal{P}$, an optimal solution $x^*$ of* INTERVAL MIN-MAX REGRET $\mathcal{P}$ *corresponds to an optimal solution of $\mathcal{P}$ for at least one extreme scenario, in particular its most favorable scenario $c^+(x^*)$ defined as follows:*

$$c^+_i(x^*) = \begin{cases} \underline{c}_i & \text{if } x^*_i = 1, \\ \overline{c}_i & \text{if } x^*_i = 0, \end{cases} \qquad i = 1, \ldots, n$$

**Proof :** Let $x^*$ denote an optimal solution of INTERVAL MIN-MAX REGRET $\mathcal{P}$ and $I(x) = \{i \in 1, \ldots, n : x_i = 1\}$. For any $s \in S$, we have

$$\begin{aligned} val(x^*, s) - val(x^*_{c^+(x^*)}, s) = &\sum_{i \in I(x^*) \setminus I(x^*_{c^+(x^*)})} c^s_i - \sum_{i \in I(x^*_{c^+(x^*)}) \setminus I(x^*)} c^s_i \\ \geq &\sum_{i \in I(x^*) \setminus I(x^*_{c^+(x^*)})} \underline{c}_i - \sum_{i \in I(x^*_{c^+(x^*)}) \setminus I(x^*)} \overline{c}_i \\ = &\; val(x^*, c^+(x^*)) - val(x^*_{c^+(x^*)}, c^+(x^*)) \end{aligned}$$

Suppose that $x^*$ is not an optimal solution of $\mathcal{P}$ for its most favorable scenario $c^+(x^*)$. Then the previous expression is strictly positive. Consequently, we have

$$val(x^*, s) - val^*_s > val(x^*_{c^+(x^*)}, s) - val^*_s, \text{ for all } s \in S$$

thus

$$\max_{s \in S} \{val(x^*, s) - val^*_s\} > \max_{s \in S} \{val(x^*_{c^+(x^*)}, s) - val^*_s\}$$

This contradicts the optimality of $x^*$ for INTERVAL MIN-MAX REGRET $\mathcal{P}$. □

The two previous propositions suggest the following direct procedure to solve exactly the min-max regret version. First construct an optimal solution $x_s^*$ for each extreme scenario $s$ and compute $R_{max}(x_s^*)$ using Proposition 7. Then, select among these solutions one having the minimum maximum regret. This procedure is in general impracticable since the number of extreme scenarios is up to $2^n$. Nevertheless, if the number $u \leq n$ of uncertain/imprecise coefficients $c_i$, corresponding to non degenerate intervals, is constant or bounded by the logarithm of a polynomial function in the size of the input, then INTERVAL MIN-MAX REGRET $\mathcal{P}$ has the same complexity as $\mathcal{P}$, as noticed by Averbakh and Lebedev [14].

As for the discrete scenario case, we briefly investigate the quality of the solutions obtained by solving $\mathcal{P}$ on a specific scenario.

A first idea, is to consider the worst scenario $\overline{c} = (\overline{c}_1, \ldots, \overline{c}_n)$, that is to take an optimal min-max solution as a candidate solution for the min-max regret version.

An optimal solution of INTERVAL MIN-MAX $\mathcal{P}$ has, in practice, a maximum regret close to the optimal value of the min-max regret version but, in the worst case, its performance is very bad. Figure 5 presents a critical example for SHORTEST PATH. The optimal solution of the min-max version is given by path $1, n$ with a worst value and a maximum regret $2^n$. On the other hand, the optimal solution of the min-max regret version is given by path $1, 2, \ldots, n$ with a worst value $2^n + 1$ and a maximum regret 1.

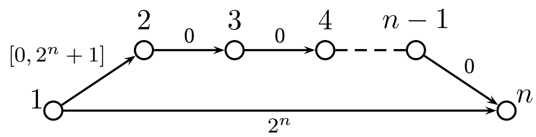


Figure 5: A critical instance of INTERVAL MIN-MAX REGRET SHORTEST PATH where the optimal min-max solution is very bad

Kasperski and Zieliński [28] show that, by solving problem $\mathcal{P}$ on a particular scenario, we can obtain a good upper bound of the optimal value of INTERVAL MIN-MAX REGRET $\mathcal{P}$.

**Proposition 9 ([28])** *Given an instance $I$ of* INTERVAL MIN-MAX REGRET $\mathcal{P}$*, consider an instance $I'$ of $\mathcal{P}$ where each coefficient of the objective function is defined by $c'_i = \frac{1}{2}(\underline{c}_i + \overline{c}_i)$. Then $x'$ an optimal solution of $I'$ has $R_{max}(x') \leq 2opt(I)$ where $opt(I)$ denotes the optimal value of instance $I$.*

For the class of bottleneck problems, as for the discrete scenario case (see Propositions 3 and 4), an optimal solution of INTERVAL MIN-MAX REGRET BOTTLENECK $\mathcal{P}$ can be obtained by solving BOTTLENECK $\mathcal{P}$ on a carefully selected scenario. Let $s_i$ denote the scenario where $c_i = \overline{c}_i$ and $c_j = \underline{c}_j$ for $j \neq i$, and $i = 1, \ldots, n$.

**Proposition 10** **([9])** *Given an instance $I$ of* INTERVAL MIN-MAX REGRET BOTTLENECK $\mathcal{P}$*, consider an instance $I'$ of* BOTTLENECK $\mathcal{P}$ *where each coefficient of the objective function is defined by $c'_i = \max\{\overline{c}_i - val^*_{s_i}, 0\}$. Then $x'$ an optimal solution of $I'$ is also an optimal solution of $I$.*

# 4 Complexity of min-max and min-max regret versions

Complexity of the min-max (regret) versions has been studied extensively during the last decade. Clearly, the min-max version of problem $\mathcal{P}$ is at least as hard as $\mathcal{P}$ since $\mathcal{P}$ is a particular case of DISCRETE MIN-MAX $\mathcal{P}$ when the scenario set contains only one scenario and $\mathcal{P}$ is a particular case of INTERVAL MIN-MAX $\mathcal{P}$ when all coefficients are degenerate. Similarly, the min-max regret version of $\mathcal{P}$ is at least as difficult as $\mathcal{P}$ since $\mathcal{P}$ reduces to DISCRETE (INTERVAL) MIN-MAX REGRET $\mathcal{P}$ when the scenario set is reduced to one scenario. For this reason, in the following, we consider only min-max (regret) versions of polynomial or pseudo-polynomial time solvable problems, for which we could hope to preserve the complexity.

For the discrete scenario case, Kouvelis and Yu [30] give complexity results for several combinatorial optimization problems, including SHORTEST PATH, SPANNING TREE, ASSIGNMENT, and KNAPSACK. In general, these versions are shown to be harder than the classical versions. More precisely, if the number of scenarios is non-constant (the number of scenarios is part of the input), these problems become strongly *NP*-hard, even when the classical problems are solvable in polynomial time. On the other hand, for a constant number of scenarios, it is only partially known if these problems are strongly or weakly *NP*-hard. Indeed, the reductions described in [30] to prove the *NP*-hardness are based on transformations from the partition problem which is known to be weakly *NP*-hard [23]. These reductions give no indications on the precise status of these problems. Most of the problems which are known to be weakly *NP*-hard are those for which there exists a pseudo-polynomial algorithm based on dynamic programming [30] (MIN-MAX (REGRET) SHORTEST PATH, MAX-MIN KNAPSACK, MIN-MAX REGRET KNAPSACK, MIN-MAX (REGRET) SPANNING TREE on grid graphs, ...)

As an example for this family of algorithms, we describe a pseudo-polynomial algorithm to solve DISCRETE MAX-MIN KNAPSACK. The standard procedure to solve the classical knapsack problem is based on dynamic programming. We give an extension to the multi-scenario case which is very similar to the procedure presented in [22] by Erlebach, Kellerer and Pferschy to solve multi-objective knapsack problem. This procedure is easier and more efficient than the one originally presented by Yu [49].

Consider a knapsack instance where each item $i$ has a weight $w_i$ and profit $p_i^s$, for $i = 1, \ldots, n$ and $s = 1, \ldots, k$, and a capacity $b$.

Let $W_i(v_1, \ldots, v_k)$ denote the minimum weight of any subset of items among the first $i$ items with profit $v_s$ in each scenario $s \in S$. The initial condition of the algorithm is given by setting $W_0(0, \ldots, 0) = 0$ and $W_0(v_1, \ldots, v_k) = b + 1$ for all other combinations. The recursive relation is given by:

If $v_s \geq p_i^s$ for all $s \in S$, then

$$W_i(v_1, \ldots, v_k) = \min\{W_{i-1}(v_1, \ldots, v_k), W_{i-1}(v_1 - p_i^1, \ldots, v_k - p_i^k) + w_i\}$$

else
$$W_i(v_1, \ldots, v_k) = W_{i-1}(v_1, \ldots, v_k)$$

for $i = 1, \ldots, n$.

Each entry of $W_n$ satisfying $W_n(v_1, \ldots, v_k) \leq b$ corresponds to a feasible solution with profit $v_s$ in scenario $s$. The set of items leading to a feasible solution can be determined easily using standard bookkeeping techniques. The feasible solutions are collected and an optimal solution to DISCRETE MAX-MIN KNAPSACK is obtained by picking a feasible solution minimizing $\max_{s \in S} v_s$. Since, $v_s \in \{0, \ldots, \sum_{i=1}^n p_i^s\}$, the running time of this approach is given by going through the complete profit space for every item that is $O(n(\max_{s \in S} \sum_{i=1}^n p_i^s)^k)$. The algorithm presented by Yu [49] has a running time $O(nb(\max_{s \in S} \sum_{i=1}^n p_i^s)^k)$.

SPANNING TREE is not known to have a dynamic programming scheme but can be solved, for instance, using Kruskal's algorithm or Prim's algorithm. A specific pseudo-polynomial algorithm is given in [3] for DISCRETE MIN-MAX SPANNING TREE using an extension of the matrix tree theorem to the multi-scenario case.

Aissi, Bazgan, and Vanderpooten investigate in [2] the complexity of min-max and min-max regret versions of two closely related polynomial-time solvable problems: CUT and $s{-}t$ CUT. For a constant number of scenarios, the complexity status of these problems is widely contrasted. More precisely, DISCRETE MIN-MAX (REGRET) CUT are polynomial, whereas DISCRETE MIN-MAX (REGRET) $s - t$ CUT are strongly *NP*-hard even for two scenarios. It is interesting to observe that $s - t$ CUT is the first polynomial-time solvable problem, whose min-max and min-max regret versions are strongly *NP*-hard.

Network location models are characterized by two types of parameters (weights of nodes and lengths of edges) and by a location site (on nodes or on edges). Kouvelis and Yu [30] show that DISCRETE MIN-MAX (REGRET) 1-MEDIAN can be solved in polynomial time for trees with scenarios both on node weights and edge lengths. If the location sites are limited to nodes, DISCRETE MIN-MAX (REGRET) 1-MEDIAN can be solved in polynomial time using an exhaustive search algorithm.

The complexity of the min-max (regret) versions of scheduling problems has also been investigated. Daniels and Kouvelis [18] show that DISCRETE MIN-MAX (REGRET) $1||\sum w_j C_j$ are *NP*-hard. Aloulou and Della Croce [6] show that DISCRETE MIN-MAX

$1|prec|f_{max}$ is polynomial-time solvable whereas DISCRETE MIN-MAX $1||\sum U_j$ is *NP*-hard.

In the interval scenario case, extensive research has been devoted for studying the complexity of min-max regret versions of various optimization problems including SHORTEST PATH [14], SPANNING TREE [8, 14], ASSIGNMENT [1], CUT and $s-t$ CUT [2]. In general, all these problems are shown to be strongly *NP*-hard.

For network locations problems, if edge lengths are uncertain/imprecise, INTERVAL MIN-MAX REGRET 1-MEDIAN is shown to be strongly *NP*-hard for general graphs [11] but polynomial-time solvable on trees [17]. However, if vertex weights are uncertain/imprecise, this problem is polynomial-time solvable [13]. For 1-CENTER, a closely related problem, INTERVAL MIN-MAX REGRET 1-CENTER is shown to be strongly *NP*-hard for general graphs if edge lengths are uncertain/imprecise but polynomial-time solvable on trees [13]. Nevertheless, if vertex weights are uncertain/imprecise, this problem is polynomial-time solvable [12].

For scheduling problems, Kasperski [27] shows that INTERVAL MIN-MAX REGRET $1|prec|f_{max}$ is polynomial-time solvable. Lebedev and Averbakh [31] show that INTERVAL MIN-MAX REGRET $1||\sum w_j C_j$ is *NP*-hard.

Tables 3-5 summarize the complexity of min-max (regret) versions of several classical combinatorial optimization problems. For all studied problems, min-max and min-max regret versions have the same complexity in the discrete scenario case. In general, there is no reduction between the min-max and the min-max regret versions or vice versa. However, in all known reductions establishing the *NP*-hardness of min-max versions [30, 1, 2], the obtained instances of min-max versions have been designed so as to get an optimal value on each scenario equal to zero. This way, these reductions also imply the *NP*-hardness of the min-max regret associated versions. Comparing now the complexity of the min-max regret version in the discrete scenario case and in the interval scenario case, we observe some slight differences. As shown by Averbakh [10], there even exists a simple combinatorial optimization problem, the selection problem, whose min-max regret version is solvable in polynomial time in the interval scenario case whereas it is *NP*-hard even for two scenarios.

Table 3: Complexity of the min-max (regret) versions of classical combinatorial problems (constant number of scenarios)

| Problems | Constant | |
|---|---|---|
| | Min-max | Min-max regret |
| SHORTEST PATH | *NP*-hard, pseudo-poly [30] | *NP*-hard, pseudo-poly [30] |
| SPANNING TREE | *NP*-hard [30], pseudo-poly [3] | *NP*-hard [30], pseudo-poly [3] |
| ASSIGNMENT | *NP*-hard [30] | *NP*-hard [30] |
| KNAPSACK | *NP*-hard, pseudo-poly [30] | *NP*-hard, pseudo-poly [30] |
| CUT | polynomial [7] | polynomial [2] |
| $s-t$ CUT | strongly *NP*-hard [2] | strongly *NP*-hard [2] |

Table 4: Complexity of the min-max (regret) versions of classical combinatorial problems (non-constant number of scenarios)

| Problems | Non-constant | |
|---|---|---|
| | Min-max | Min-max regret |
| SHORTEST PATH | strongly *NP*-hard [30] | strongly *NP*-hard [30] |
| SPANNING TREE | strongly *NP*-hard [30] | strongly *NP*-hard [4] |
| ASSIGNMENT | strongly *NP*-hard [1] | strongly *NP*-hard [1] |
| KNAPSACK | strongly *NP*-hard [30] | strongly *NP*-hard [4] |
| CUT | strongly *NP*-hard [2] | strongly *NP*-hard [2] |
| $s-t$ CUT | strongly *NP*-hard [2] | strongly *NP*-hard [2] |

# 5 Approximation of the min-max (regret) versions

The approximation of the min-max (regret) versions of classical combinatorial optimization problems has received great attention recently. Aissi, Bazgan, and Vanderpooten initiated in [4] a systematic study of this question in the discrete scenario case. More precisely, they investigate the relationships between min-max, min-max regret and multi-objective versions, and show the existence, in the case of a constant number of scenarios, of fully polynomial-time approximation schemes (fptas) for min-max (regret) versions of several classical optimization problems. They also study the case of a non-constant number of scenarios. In this setting, they provide non-approximability results for min-max (regret) versions of shortest path and spanning tree. In [3] they adopt an alternative

Table 5: Complexity of min-max regret versions of classical combinatorial problems (interval scenario case)

| Problems | Min-max regret |
|---|---|
| SHORTEST PATH | strongly *NP*-hard [14] |
| SPANNING TREE | strongly *NP*-hard [14] |
| ASSIGNMENT | strongly *NP*-hard [1] |
| KNAPSACK | *NP*-hard |
| CUT | polynomial [2] |
| $s-t$ CUT | strongly *NP*-hard [2] |

perspective and develop a general approximation scheme, using the scaling technique, which can be applied to min-max (regret) versions of some problems, provided that some conditions are satisfied. The advantage of this second approach is that the resulting fptas usually have much better running times than those derived using multi-objective fptas for the multi-objective versions.

The interval scenario case has been much less studied. We only report one general result due to Kasperski and Zieliński [28].

## 5.1 Discrete scenario case

As a first general result, Proposition 1 gives a $k$-approximation algorithm for DISCRETE MIN-MAX (REGRET) $\mathcal{P}$ when the underlying problem $\mathcal{P}$ is polynomial-time solvable. In some cases, however, we can derive fptas for some problems.

### 5.1.1 Relationships between min-max (regret) and multi-objective versions

Exploiting the results mentioned in section 3.1.2 concerning the relationships with the multi-objective version, we can derive approximation algorithms, with a noticeable difference between the min-max and the min-max regret versions.

**Min-max version**

**Proposition 11** **([4])** *Given a minimization problem $\mathcal{P}$, for any function $f : \mathbb{N} \rightarrow (1, \infty)$, if* MULTI-OBJECTIVE $\mathcal{P}$ *has a polynomial-time $f(n)$-approximation algorithm, then* DISCRETE MIN-MAX $\mathcal{P}$ *has a polynomial-time $f(n)$-approximation algorithm.*

**Proof :** Let $F$ be an $f(n)$-approximation of the set of efficient solutions. Since at least one optimal solution $x^*$ for DISCRETE MIN-MAX $\mathcal{P}$ is efficient as shown in Proposition 5, there exists a solution $y \in F$ such that $val(y,s) \leq f(n)val(x^*,s)$, for all $s \in S$. Consider among set $F$ a solution $z$ that has a minimum $\max_{s\in S} val(z,s)$. Thus, $\max_{s\in S} val(z,s) \leq \max_{s\in S} val(y,s) \leq f(n) \max_{s\in S} val(x^*,s)$. □

**Corollary 2 ([4])** *For a constant number of scenarios,* DISCRETE MIN-MAX SHORTEST PATH*,* DISCRETE MIN-MAX SPANNING TREE*, and* DISCRETE MAX-MIN KNAPSACK *have an fptas.*

**Proof :** For a constant number of objectives, multi-objective versions of SHORTEST PATH, SPANNING TREE, and KNAPSACK have an fptas as shown by Papadimitriou and Yannakakis in [40]. □

**Min-max regret version**

As shown in Proposition 6, at least one optimal solution $x^*$ for DISCRETE MIN-MAX REGRET $\mathcal{P}$ is efficient. Unfortunately, given $F$ an $f(n)$-approximation of the set of efficient solutions, a solution $x \in F$ with a minimum $R_{max}(x)$ is not necessarily an $f(n)$-approximation for the optimum value since the minimum maximum regret could be very small compared with the error that was allowed in $F$.

However, it is sometimes possible to derive an approximation algorithm working directly in the regret space instead of the criterion space. This way, an fptas is obtained in [4] for DISCRETE MIN-MAX REGRET SHORTEST PATH using a dynamic programming algorithm that computes at each stage the set of efficient vectors of regrets.

### 5.1.2 A general approximation scheme in the discrete scenario case

A standard approach for designing fptas for *NP*-hard problems is to use the *scaling technique* [43] in order to transform an input instance into an instance that is easy to solve. Then an optimal solution for this new instance is computed using a pseudo-polynomial algorithm. This solution will serve as an approximate solution. This general scheme usually relies on the determination of a lower bound that is polynomially related to the maximum value of the original instance. However, for the min-max (regret) versions, we need to adapt this scheme in order to obtain an fptas.

**Proposition 12 ([3])** *Given a problem* DISCRETE MIN-MAX (REGRET) $\mathcal{P}$*, if*

1. *for any instance $I$, a lower and an upper bound $L$ and $U$ of the optimal value $opt$ can be computed in time $p(|I|)$, such that $U \leq q(|I|)L$, where $p$ and $q$ are two polynomials with $q$ non decreasing and $q(|I|) \geq 1$, and*

2. *there exists an algorithm that finds for any instance $I$ an optimal solution in time $r(|I|, U)$ where $r$ is a non decreasing polynomial,*

*then* DISCRETE MIN-MAX (REGRET) $\mathcal{P}$ *has an fptas.*

We discuss now the two conditions of the previous theorem. The first condition can be satisfied easily for polynomial-time solvable problems as shown in Proposition 1 where we exhibit bounds $L$ and $U$ such that $U \leq kL$. Furthermore, if $\mathcal{P}$ is polynomially approximable, then the first condition of Proposition 12 can also be satisfied for DISCRETE MIN-MAX $\mathcal{P}$ using Proposition 1. More precisely, if $\mathcal{P}$ is $f(n)$-approximable where $f(n)$ is a polynomial, given an instance $I$ of DISCRETE MIN-MAX $\mathcal{P}$, let $x'$ be an $f(|I|/k)$-approximate solution in $I'$ (defined as in the proof of Proposition 1), then we have $L = \frac{1}{f(|I|/k)} \sum_{s \in S} \frac{1}{k} val(x', s)$ and $U = \max_{s \in S} val(x', s)$, and thus $U \leq kf(|I|/k)L$.

The second condition of Proposition 12 can be weakened for DISCRETE MIN-MAX $\mathcal{P}$ by requiring only a pseudo-polynomial algorithm, that is an algorithm polynomial in $|I|$ and $max(I) = \max_{i,s} c_i^s$. Indeed, knowing an upper bound $U$, we can eliminate any variable $x_i$ such that $c_i^s > U$ on at least one scenario $s \in S$. Condition 2 is then satisfied by applying the pseudo-polynomial algorithm on this modified instance.

Min-max (regret) versions of some problems, like SHORTEST PATH, KNAPSACK, admit pseudo-polynomial time algorithms based on dynamic programming [30]. For some dynamic programming formulations, we can easily obtain algorithms satisfying condition 2, by discarding partial solutions with value more than $U$ on at least one scenario. For other problems, like SPANNING TREE, which are not known to admit pseudo-polynomial algorithms based on dynamic programming, specific algorithms are required [3].

**Corollary 3 ([3])** *For a constant number of scenarios,* DISCRETE MIN-MAX (REGRET) SHORTEST PATH, DISCRETE MIN-MAX (REGRET) SPANNING TREE *have an fptas.*

The resulting fptas obtained using Corollary 3 are much more efficient than those obtained using Corollary 2.

Table 6 summarizes approximation results in the discrete scenario case for several combinatorial optimization problems.

Table 6: Approximation of the min-max and min-max regret versions in the discrete scenario case

| Problems | Constant | | Non-constant | |
|---|---|---|---|---|
| | Min-max | Min-max regret | Min-max | Min-max regret |
| SHORTEST PATH | fptas [46] | fptas [4, 3] | not $(2-\varepsilon)$ apx. [4] | not $(2-\varepsilon)$ apx. [4] |
| SPANNING TREE | fptas [4, 3] | fptas [4, 3] | not $(\frac{3}{2}-\varepsilon)$ apx. [4] | not $(\frac{3}{2}-\varepsilon)$ apx. [4] |
| KNAPSACK | fptas [4] | non apx. [4] | non apx. [4] | non apx. [4] |

### 5.2 Interval scenario case

In this case, most of the min-max regret versions of classical combinatorial optimization problems become strongly *NP*-hard (see Table 5). Kasperski and Zieliński [28] give a 2-approximation algorithm for the min-max regret versions of polynomial-time solvable problems as a direct consequence of Proposition 9.

## 6 Resolution of the min-max (regret) versions

We review in this section exact procedures for solving the min-max and min-max regret versions of classical combinatorial optimization problems.

### 6.1 Discrete scenario case

Kouvelis and Yu [30] implement a branch-and-bound procedure to solve the min-max (regret) versions of several combinatorial optimization problems. A lower bound on each node of the arborescence is calculated using surrogate relaxation as explained in the following.

Consider DISCRETE MIN-MAX REGRET $\mathcal{P}$ which can be stated as:

$$\begin{cases} \min y \\ y \geq val(x,s) - val_s^*, \ \forall s \in S \\ x \in X, y \geq 0 \end{cases} \tag{3}$$

Let $\mu = (\mu_1, \ldots, \mu_k)$ be a multiplier vector such that $\mu_s \geq 0$, $\forall s \in S$, and $\sum_{s\in S} \mu_s = 1$. The surrogate relaxation of (3) is formulated as follows:

$$\begin{cases} L(\mu) = \min \sum_{s\in S} \mu_s(val(x,s) - val_s^*) \\ x \in X \end{cases} \tag{4}$$

The efficiency of the relaxation procedure rests on finding a vector $\mu^*$ providing the best lower bound $L(\mu^*) \leq opt$ where $opt$ is the optimum value of the input considered. A sub-gradient procedure is used in [30] to find the tightest one. The later is based on solving, at each iteration, problem (4) with a multiplier vector $\mu = (\mu_1, \ldots, \mu_k)$ and the result is exploited in order to find a new multiplier vector that improves the bound given by the surrogate relaxation. Furthermore, the optimal solution $x(\mu)$ of (4) is a feasible solution of (3) and thus provides an upper bound $U(\mu) = \max_{s\in S}(val(x(\mu), s) - val^*_s)$ on the optimal value. The best upper bound obtained so far during the execution of the sub-gradient procedure is saved.

The Best-Node First search strategy is adopted in the branch-and-bound procedure presented in [30]. The choice of the branching variable is guided by the surrogate relaxation.

DISCRETE MIN-MAX $\mathcal{P}$ can be solved in a similar way by removing $val^*_s$ in (3) and (4).

## 6.2 Interval scenario case

In the interval scenario case, Inuiguchi and Sakawa [24] study the min-max regret version of linear programs. Their algorithm is based on a relaxation procedure developed initially by Shimizu and Aiyochi [44]. Karasan, Pinar and Yaman [26, 48] formulate the min-max regret versions of SHORTEST PATH and SPANNING TREE as mixed integer linear programs. A generalization of this formulation to problems with zero duality gap is presented in section 6.2.1. Montemanni and Gambardella [36, 37] propose branch-and-bound algorithms for solving these problems. Recently, Aissi, Vanderpooten and Vanpeperstraete [5], Montemanni [34], Montemanni and Gambardella [38], and Montemanni, Barta and Gambardella [35] proposed a relaxation procedure for solving the min-max regret versions of problems from $\mathcal{C}$.

In this section, we give a formulation of the min-max regret versions of problems with a zero duality gap (section 6.2.1) and we present a general relaxation procedure to solve the min-max regret versions of polynomial-time solvable or *NP*-hard problems (section 6.2.2).

### 6.2.1 Formulation of the min-max regret versions of problems with a zero duality gap

Consider a minimization problem $\mathcal{P} \in \mathcal{C}$ with a zero duality gap and where the feasible solution set is defined by $X = \{x : Ax \geq b,\ x \in \{0,1\}^n\}$. The maximum regret

$R_{max}(x)$ of a solution $x$ is given as follows:

$$R_{max}(x) = \max_{s \in S, y \in X}\{c^s x - c^s y\} = c^-(x)x - c^-(x)x^*_{c^-(x)} = \overline{c}x - \min_{z \in X} c^-(x)z$$

where $\overline{c} = (\overline{c}_1, \ldots, \overline{c}_n)$ and $x^*_s$ is an optimal solution for scenario $s$.

Therefore, INTERVAL MIN-MAX REGRET $\mathcal{P}$ can be written as follows:

$$\min_{x \in X} R_{max}(x) = \min_{x \in X}\{\overline{c}x - \min_{z \in X} c^-(x)z\} \tag{5}$$

Since $\mathcal{P}$ has a zero duality gap, we can replace in (5) the problem $\min_{z \in X} c^-(x)z$ by its dual:

$$\max_{y \in Y} b^t y$$

where $Y = \{y : A^t y \leq c^-(x),\ y \geq 0\}$. Consequently, INTERVAL MIN-MAX REGRET $\mathcal{P}$ can be formulated as follows:

$$\min_{x \in X,\ y \in Y}\{\overline{c}x - b^t y\}$$

In the following, we illustrate this approach with the assignment problem. A linear programming formulation of this problem is given as follows:

$$\begin{cases} \min \sum_{i,j=1}^n c_{ij}x_{ij} \\ \text{s.t.} \\ \sum_{j=1}^n x_{ij} = 1;\ i = 1, \ldots, n \\ \sum_{i=1}^n x_{ij} = 1;\ j = 1, \ldots, n \\ x_{ij} \geq 0;\ i, j = 1, \ldots, n \end{cases} \tag{6}$$

The dual of (6) is given by

$$\begin{cases} \max \sum_{i=1}^n (u_i + v_j) \\ \text{s.t.} \\ u_i + v_j \leq c_{ij};\ \ i, j = 1, \ldots, n \\ u_i, v_j \gtrless 0;\ i, j = 1, \ldots, n \end{cases}$$

Thus, a formulation of INTERVAL MIN-MAX REGRET ASSIGNMENT is

$$\begin{cases} \min \sum_{i,j=1}^n \overline{c}_{ij}x_{ij} - \sum_{i=1}^n (u_i + v_j) \\ \text{s.t.} \\ \sum_{j=1}^n x_{ij} = 1;\ i = 1, \ldots, n \\ \sum_{i=1}^n x_{ij} = 1;\ j = 1, \ldots, n \\ u_i + v_j \leq \underline{c}_{ij} + (\overline{c}_{ij} - \underline{c}_{ij})x_{ij};\ \ i, j = 1, \ldots, n \\ x_{ij} \in \{0, 1\};\ i, j = 1, \ldots, n \\ u_i, v_j \gtrless 0;\ i, j = 1, \ldots, n \end{cases}$$

### 6.2.2 A relaxation procedure for the min-max regret versions

A general linear formulation of min-max regret versions is given by:

$$\textit{MMR}\begin{cases}\min r\\ \text{s.t.}\\ r \geq c^s x - c^s x^*_s, \quad \forall s \in S\\ x \in X, r \geq 0\end{cases}$$

where constraint $r \geq c^s x - c^s x^*_s$ is referred to as a cut of type 1.

From Proposition 7, we know that, in the interval scenario case, we can restrict the scenario set $S$ to the set of extreme scenarios. However, since the number of extreme scenarios is up to $2^n$, formulation *MMR* is exponential.

In order to handle this difficulty, a standard approach is to use a relaxation procedure which starts with an initial subset of scenarios $S^0$ and iteratively takes into account one additional scenario, by adding a cut of type 1, until we can establish that the current solution is optimal.

Considering the following formulation, defined at iteration $h$ with a current subset of scenarios $S^h$:

$$\textit{MMR}^h\begin{cases}\min r\\ \text{s.t.}\\ r \geq c^s x - c^s x^*_s, \quad \forall s \in S^h\\ x \in X, r \geq 0\end{cases}$$

we denote by $r^h$ and $x^h$ the optimal value and an optimal solution of *MMR*$^h$ at step $h$.

The details of the relaxation procedure, originally proposed in [24, 32, 33] for solving min-max regret versions of linear programming problems, are given as follows:

**Algorithm 1** Relaxation procedure

1: Initialisation. $LB \leftarrow 0$, $S^0 \leftarrow \emptyset$, $h \leftarrow 0$, and construct $x^0 \in X$.
2: Compute $\widehat{c}^h$ and $R_{max}(x^h)$ by solving $\max_{s \in S, x \in X}\{c^s x^h - c^s x\}$.
3: **if** $R_{max}(x^h) \leq LB$ **then**
4: END: $x^h$ is an optimal solution.
5: $S^{h+1} \leftarrow S^h \cup \{\widehat{c}^h\}$.
6: $h \leftarrow h+1$, compute $r^h$ and $x^h$ by solving *MMR*$^h$, $LB \leftarrow r^h$ and go to step 2.

In the linear programming case, step 2 is the most computationally demanding part of the algorithm since it requires solving a quadratic programming problem. On the other hand, step 6 requires only solving a linear programming problem.

This algorithm can be extended to $0$-$1$ linear programming problems. In this case, we do not need to solve a quadratic programming problem at step 2 since we know, from Proposition 7, that we can take as scenario $\widehat{c}^h$ the worst case scenario for for $x^h$, $c^-(x^h)$. Hence, step 2 of Algorithm 1 can be replaced by:

2: $\widehat{c}^h \leftarrow c^-(x^h)$, compute an optimal solution $x^*_{\widehat{c}^h}$ of $\min_{x \in X} \widehat{c}^h x$,
$R_{max}(x^h) \leftarrow \widehat{c}^h x^h - \widehat{c}^h x^*_{\widehat{c}^h}$.

Now step 6 becomes computationally expensive since *MMR*$^h$ is a mixed integer programming problem whose difficulty increases with the number of added cuts. This explains why a straightforward application of this algorithm to solve min-max regret versions of $0$-$1$ linear programming problems is not satisfactory. We can improve significantly the performance of the algorithm by using alternative cuts, called cuts of type 2, which were found independently in [5] and [34, 38]. These cuts are defined as follows:

$$r \geq \overline{c}x - c^-(x)x^*_{c^-(x^h)}$$

These cuts can be derived easily for a $0$-$1$ linear programming problem, since $R_{max}(x) \geq \overline{c}x - c^-(x)y$ for any $x, y \in X$, and in particular for $y = x^*_{c^-(x^h)}$.

**Proposition 13** *Given a minimization problem $\mathcal{P} \in \mathcal{C}$, cuts of type 2 dominate cuts of type 1.*

**Proof :** We need to prove that for any $x, x^h \in X$, we have $\overline{c}x - c^-(x)x^*_{c^-(x^h)} \geq c^-(x^h)x - c^-(x^h)x^*_{c^-(x^h)}$. Thus, we consider the following difference:

$$\overline{c}x - c^-(x)x^*_{c^-(x^h)} - c^-(x^h)x + c^-(x^h)x^*_{c^-(x^h)} = \sum_{i=1}^{m} (\overline{c}_i - \underline{c}_i)(x_i - x_i x^*_{c^-(x^h),i} - x_i x_i^h + x_i^h x^*_{c^-(x^h),i})$$

If $x_i^h = 1$, we have $(\overline{c}_i - \underline{c}_i)(x^*_{c^-(x^h),i} - x_i x^*_{c^-(x^h),i}) \geq 0$ for $i = 1, \ldots, n$. If $x_i^h = 0$, we have $(\overline{c}_i - \underline{c}_i)(x_i - x_i x^*_{c^-(x^h),i}) \geq 0$ for $i = 1, \ldots, n$. □

# 7 Conclusions

We reviewed in this paper motivations, complexity, approximation, and exact resolution for the min-max and min-max regret versions of several combinatorial optimization problems. We conclude this survey by listing some open questions.

For all studied problems, min-max and min-max regret versions have the same complexity in the discrete scenario case. As observed at the end of section 4, there is no general reduction between the min-max and the min-max regret versions or vice versa, except for bottleneck problems. The construction of such a reduction would be interesting since only one algorithm (or *NP*-hardness proof) would be sufficient for both versions.

Almost all problems discussed in this paper have a clear complexity status, see, e.g., Tables 3-5. However, a few *NP*-hard problems still need to be precisely classified (weakly or strongly *NP*-hard). As a first example, INTERVAL MIN-MAX REGRET KNAPSACK is clearly *NP*-hard, but its precise status is not known. A second example is DISCRETE MIN-MAX (REGRET) ASSIGNMENT for a constant number of scenarios. The original proofs of the *NP*-hardness of these problems are obtained in [30] using a reduction from partition, which has a pseudo-polynomial time algorithm [23]. Therefore, it remains an open question whether these problems have pseudo-polynomial time algorithms or are strongly *NP*-hard.

Approximation results, which are quite recent, should be studied more thoroughly. In particular, for a non-constant number of scenarios, there is a gap between the general $k$-approximation result given at the beginning of section 5.1 and specific negative results indicated in Table 6. In the interval scenario case, finding better approximation algorithms than the general 2-approximation algorithm referred in section 5.2, for the min-max regret versions of specific polynomial-time solvable problems, is also an interesting research perspective.

Finally, the efficient exact resolution of the min-max and min-max regret versions of combinatorial optimization problems remains a computationally challenging question.

# References

[1] H. Aissi, C. Bazgan, and D. Vanderpooten. Complexity of the min-max and min-max regret assignment problem. *Operations Research Letters*, 33(6):634–640, 2005.

[2] H. Aissi, C. Bazgan, and D. Vanderpooten. Min-max (regret) versions of cut problems. In *Proceedings of the 16th International Symposium on Algorithms and Computation (ISAAC 2005), Sanya, Hainan, China*, LNCS 3827, pages 789–798. Springer-Verlag, 2005.

[3] H. Aissi, C. Bazgan, and D. Vanderpooten. Approximating min-max (regret) versions of some polynomial problems. In *Proceedings of the 12th Annual International Computing and Combinatorics Conference (COCOON 2006), Taipei, Taiwan*, LNCS 4112, pages 428–438. Springer-Verlag, 2006.

[4] H. Aissi, C. Bazgan, and D. Vanderpooten. Approximation of min-max and min-max regret versions of some combinatorial optimization problems. *European Journal of Operational Research*, 179(2):281–290, 2007.

[5] H. Aissi, D. Vanderpooten, and J.M. Vanpeperstraete. Robust approaches for the data association problem. In *Proceedings of the 8th International Conference on Information Fusion (FUSION 2005)*, Philadelphia, USA, 2005.

[6] M.A. Aloulou and F. Della Croce. On the complexity of single machine scheduling problems under scenario-based uncertainty. Annales du LAMSADE N°4-5, pages 1–14, 2005.

[7] A. Armon and U. Zwick. Multicriteria global minimum cuts. In *Proceedings of the 15th International Symposium on Algorithms and Complexity (ISAAC 2004), Hong Kong, China*, LNCS 3341, pages 65–76. Springer-Verlag, 2004.

[8] I. D. Aron and P. Van Hentenryck. On the complexity of the robust spanning tree with interval data. *Operations Research Letters*, 32(1):36–40, 2004.

[9] I. Averbakh. Minmax regret solutions for minimax optimization problems with uncertainty. *Operations Research Letters*, 27:57–65, 2000.

[10] I. Averbakh. On the complexity of a class of combinatorial optimization problems with uncertainty. *Mathematical Programming*, 90:263–272, 2001.

[11] I. Averbakh. Complexity of robust single facilitylocation problems on networks with uncertain edge lengths. *Discrete Applied Mathematics*, 127:505–522, 2003.

[12] I. Averbakh and O. Berman. Minimax regret p-center location on a network with demand uncertainty. *Location Science*, 5(4):247–254, 1997.

[13] I. Averbakh and O. Berman. An improved algorithm for the min-max regret median problem on a tree. *Networks*, 41(2):97–103, 2003.

[14] I. Averbakh and V. Lebedev. Interval data min-max regret network optimization problems. *Discrete Applied Mathematics*, 138(3):289–301, 2004.

[15] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming series B*, 98:49–71, 2003.

[16] R. D. Carr, H. J. Greenberg, W. E. Hart, G. Konjevod E. Lauer, H. Lin, T. Morrison, and C. A. Phillips. Robust optimization of contaminant sensor placement for community water systems. *Mathematical Programming, Serie B*, 107(1–2):337–356, 2006.

[17] B. Chen and C. S. Lin. Min-max regret robust 1-median location on a tree. *Networks*, 31:93–103, 1998.

[18] R. L. Daniels and P. Kouvelis. Robust scheduling to hedge against processing time uncertainty in single-stage production. *Management Science*, 41(2):363–376, 1995.

[19] M.S. Daskin, S.M. Hesse, and C.S. ReVelle. $\alpha$-reliable p-minimax regret: a new model for strategic facility location modeling. *Location Science*, 5:227–246, 1997.

[20] A. Dodonova and Y. Khoroshilov. Applications of regret theory to asset pricing. Technical report, University of Ottawa, 2005.

[21] M. Ehrgott. *Multicriteria Optimization*. Springer, 2nd edition, 2005.

[22] T. Erlebach, H. Kellerer, and U. Pferschy. Approximating multiobjective knapsack problems. *Management Science*, 48(12):1603–1612, 2002.

[23] M. Garey and D. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*. Freeman, San Francisco, 1979.

[24] M. Inuiguchi and M. Sakawa. Minimax regret solution to linear programming problems with interval objective function. *European Journal of Operational Research*, 86(3):526–536, 1995.

[25] R. Kalai, M.A. Aloulou, P. Vallin, and D. Vanderpooten. Robust 1-median location problem on a tree. In *Proceedings of Euro Conference for Young OR Researchers and Practitioners (ORP3 2005), Valencia, Spain*, 2005.

[26] O. E. Karaşan, M. C. Pinar, and H. Yaman. The robust shortest path problem with interval data. 2001. Available from http://www.optimization-online.org.

[27] A. Kasperski. Minimizing maximal regret in single machine sequencing problem with maximum lateness criterion. *Operations Research Letters*, 33(4):431–436, 2005.

[28] A. Kasperski and P. Zieliński. An approximation algorithm for interval data min-max regret combinatorial optimization problems. *Information Processing Letters*, 97(5):177–180, 2006.

[29] A. O. Kazakçi, S. Rozakis, and D. Vanderpooten. Energy crop supply in france: a min-max regret approach. *Journal of the Operational Research Society*. To appear, 2007.

[30] P. Kouvelis and G. Yu. *Robust discrete optimization and its applications*. Kluwer Academic Publishers, Boston, 1997.

[31] V. Lebedev and I. Averbakh. Complexity of minimizing the total flow time with interval data and minmax regret criterion. *Discrete Applied Mathematics*, 154:2167–2177, 2006.

[32] M. E. Mausser and M. Laguna. A heuristic to minimax absolute regret for linear programs with interval objective function coefficients. *European Journal of Operational Research*, 117(1):157–174, 1998.

[33] M. E. Mausser and M. Laguna. A new mixed integer formulation for the maximum regret problem. *International Transaction on Operations Research*, 5(5):398–403, 1998.

[34] R. Montemanni. A Benders decomposition approach for the robust spanning tree with interval data. *European Journal of Operational Research*, 174(3):1479–1490, 2006.

[35] R. Montemanni, J. Barta, and L. Gambardella. An exact algorithm for the robust travelling salesman problem with interval data. In *Proceedings of ODYSSEUS 2006*, pages 256–258, 2006.

[36] R. Montemanni and L. Gambardella. A branch and bound algorithm for the robust shortest path problem with interval data. *Operations Research Letters*, 32(3):225–232, 2005.

[37] R. Montemanni and L. Gambardella. A branch and bound algorithm for the robust spanning tree problem with interval data. *European Journal of Operational Research*, 161(3):771–779, 2005.

[38] R. Montemanni and L. Gambardella. The robust shortest path problem with interval data via Benders decomposition. *4OR*, 3(4):315–328, 2005.

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

[40] C. H. Papadimitriou and M. Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *IEEE Symposium on Foundations of Computer Science (FOCS 2000), Redondo Beach, California, USA*, pages 86–92, 2000.

[41] J. Rosenhead, M. Elton, and S.K. Gupta. Robustness and optimality as criteria for strategic decisions. *Operational Research Quarterly*, 23(4):413–431, 1972.

[42] B. Roy. A missing link in OR-DA: robustness analysis. *Foundations of Computing and Decision Sciences*, 23(3):141–160, 1998.

[43] S. Sahni. General techniques for combinatorial approximation. *Operations Research*, 25(6):920–936, 1977.

[44] K. Shimizu and E. Aiyoshi. Necessary conditions for min-max problems and algorithms by a relaxation procedure. *IEEE Transactions on Automatic Control*, 25(1):62–66, 1980.

[45] P. Vincke. Robust solutions and methods in decision-aid. *Journal of Multicriteria Decision Analysis*, 8:181–187, 1999.

[46] A. Warburton. Approximation of Pareto optima in multiple-objective, shortest-path problems. *Operations Research*, 35(1):70–79, 1987.

[47] J. P. Watson, W. E. Hart, and R. Murray. Formulation and optimization of robust sensor placement problems for contaminant warning systems. Technical report, 2006.

[48] H. Yaman, O. E. Karaşan, and M. C. Pinar. The robust spanning tree problem with interval data. *Operations Research Letters*, 29(1):31–40, 2001.

[49] G. Yu. On the max-min 0-1 knapsack problem with robust optimization applications. *Operations Research*, 44(2):407–415, 1996.