Complexity of the min-max (regret) versions of min cut problems

Hassene Aissi Cristina Bazgan Daniel Vanderpooten

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

Abstract

This paper investigates the complexity of the min-max and min-max regret versions of the min s-t cut and min cut problems. Even if the underlying problems are closely related and both polynomial, the complexity of their min-max and min-max regret versions, for a constant number of scenarios, is quite contrasted since they are respectively strongly *NP*-hard and polynomial. However, for a non constant number of scenarios, these versions become strongly *NP*-hard for both problems. In the interval scenario case, min-max versions are trivially polynomial. Moreover, for min-max regret versions, we obtain the same contrasted result as for a constant number of scenarios: min-max regret min s-t cut is strongly *NP*-hard whereas min-max regret min cut is polynomial.

Keywords: min-max, min-max regret, complexity, min cut, min s-t cut.

1 Introduction

The definition of an instance of a combinatorial optimization problem requires to specify parameters, in particular objective function coefficients, 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. Each scenario s can be represented as a m-dimensional vector where m is the number of relevant numerical parameters. Kouvelis and Yu [10] proposed the min-max and min-max regret criteria, stemming from decision theory, to construct robust solutions hedging against parameters variations. In min-max optimization, the aim is to find a solution having the best worst case value across all scenarios. In min-max regret versions, it is required to find a feasible solution minimizing the maximum deviation, over all possible scenarios, of the value of the solution from the optimal value of the corresponding scenario. Two natural ways of describing the set of all possible scenarios S have been considered in the literature. In the *discrete scenario case*, S is described explicitly by the list of all vectors $s \in S$. In this case, we distinguish situations where the number of scenarios is constant from those where the number of scenarios is non constant. In the *interval scenario case*, each numerical parameter can take any value between a lower and upper bound, independently of the values of the other parameters. Thus, in this case, S is the cartesian product of the intervals of uncertainty for the parameters.

Complexity of the min-max (regret) versions has been studied extensively during the last decade. In the discrete scenario case, this complexity was investigated for several combinatorial optimization problems in [10]. In general, these versions are shown to be harder than the classical versions. For a constant number of scenarios, pseudo-polynomial algorithms, based on dynamic programming, are given in [10] for the min-max (regret) versions of shortest path, knapsack and minimum spanning tree for grid graphs. The latter result is extended to general graphs in [1]. When the number of scenarios is not constant, these versions usually become strongly *NP*-hard, even if the underlying problem is polynomial. 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 [5], minimum spanning tree [4, 5] and assignment [2].

We investigate in this paper the complexity of min-max and min-max regret versions of two closely related polynomial problems, min cut and min *s*-*t* cut. Quite interestingly, for a constant number of scenarios, the complexity status of these problems is widely contrasted. More precisely, min-max (regret) versions of min cut are polynomial whereas min-max (regret) versions of min *s*-*t* cut are strongly *NP*-hard even for two scenarios. For a non constant number of scenarios, min-max (regret) min cut become strongly *NP*-hard. These results were established by, or can be derived from, Armon and Zwick [3].

In the interval scenario case, min-max versions are trivially polynomial. Moreover, for min-max regret versions, we obtain the same contrasted result as for a constant number of scenarios: min-max regret cut is polynomial whereas min-max regret min *s*-*t* cut is strongly *NP*-hard.

After presenting preliminary concepts (Section 2), we investigate the complexity of min-max (regret) versions of min cut and min *s*-*t* cut in the discrete scenario case (Section 3), and in the interval scenario case (Section 4). Conclusions and perspectives are provided in a final section.

2 Preliminaries

Let us consider an instance of a 0-1 minimization problem Q with a linear objective function defined as:

$$\begin{cases} \min \sum_{i=1}^{m} c_i x_i & c_i \in \mathbb{Q}^+ \\ x \in X \subset \{0,1\}^m \end{cases}$$

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*-hard (knapsack, set covering, ...).

In the discrete scenario case, the min-max (regret) version associated to Q has as input a finite set of scenarios S where each scenario $s \in S$ is represented by a vector $(c_1^s, \ldots, c_m^s)$. 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 $[\underline{c}_i, \overline{c}_i]$, $i = 1, \ldots, m$.

We denote by $val(x, s) = \sum_{i=1}^{m} c_i^s x_i$ 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 optimal value under the scenario s.

The min-max optimization problem corresponding to Q, denoted by MIN-MAX Q, consists of finding a solution x having the best worst case value across all scenarios, which can be stated as:

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

This version is denoted by DISCRETE MIN-MAX Q in the discrete scenario case, and by INTERVAL MIN-MAX Q in the interval scenario case.

Given a solution $x \in X$, its *regret* under scenario $s \in S$ is defined as:

$$R(x, s) = val(x, s) - val_s^*$$

The *maximum regret* of solution x is then defined as $R_{max}(x) = \max_{s \in S} R(x, s)$.

The min-max regret optimization problem corresponding to Q, denoted by MIN-MAX REGRET Q, consists of finding a solution x minimizing its maximum regret $R_{max}(x)$ which can be stated as:

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

This version is denoted by DISCRETE MIN-MAX REGRET Q in the discrete scenario case, and by INTERVAL MIN-MAX REGRET Q in the interval scenario case.

In the interval scenario case, for a solution $x \in X$, we denote by $c^-(x)$ the worst scenario associated to x, where $c_i^-(x) = \overline{c}_i$ if $x_i = 1$ and $c_i^-(x) = \underline{c}_i$ if $x_i = 0$, $i = 1, \ldots, m$. Then we can establish easily that $R_{max}(x) = R(x, c^-(x))$, as shown e.g. in [15] in the specific context of the minimum spanning tree problem.

In this paper, we focus on the min-max (regret) versions of several minimum cut problems.

Given a graph $G = (V, E)$, a *cut* $C = (V_1, V_2)$ is the set of edges from E that have one endpoint in V_1 and one endpoint in V_2, where V_1 is a non-empty proper subset of V and $V_2 = V \setminus V_1$. The *value* of a cut is the number of edges in this cut when G is unweighted and the sum of the weights of the edges of this cut when G is weighted.

A bisection is a cut $C = (V_1, V_2)$ such that $|V_1| = |V_2|$. Given two specified vertices s and t, an s-t cut (respectively s-t bisection) is a cut (respectively bisection) $C = (V_1, V_2)$ such that $s \in V_1$ and $t \in V_2$.

The minimum cut problems for which we study min-max (regret) versions are:

MIN CUT
Input: A connected undirected graph $G = (V, E)$ with a nonnegative weight w_{ij} associated with each edge $(i, j) \in E$.
Output: A cut in G of minimum value.

MIN s-t CUT
Input: A connected undirected graph $G = (V, E)$ with a nonnegative weight w_{ij} associated with each edge $(i, j) \in E$, and two specified vertices $s, t \in V$.
Output: An s-t cut in G of minimum value.

Since both problems are minimization problems, we shall refer to their min-max versions omitting MIN from their names, while keeping it for their min-max regret versions to avoid any ambiguity. Thus, these versions will be named DISCRETE MIN-MAX CUT, DISCRETE MIN-MAX REGRET MIN CUT, DISCRETE MIN-MAX s-t CUT, DISCRETE MIN-MAX REGRET MIN s-t CUT in the discrete scenario case and for the interval scenario case, we replace DISCRETE by INTERVAL.

In order to prove one of our complexity results we use the following problem proved strongly *NP*-hard in [6].

Min s-t Bisection
Input: An unweighted graph $G = (V, E)$ with an even number of vertices, and two specified vertices $s, t \in V$.
Output: An s-t bisection in G of minimum value.

3 Discrete scenario case

We investigate the complexity of the min-max (regret) versions of min s-t cut (Section 3.1) and min cut (Section 3.2). Results for the min-max versions were established by, or can be easily derived from, Armon and Zwick [3]. Results for the min-max regret versions follow from the min-max case.

3.1 Min s-t cut

We first review the min-max version of the min s-t cut problem. Papadimitriou and Yannakakis [13, Th.6] showed that given a constant number $k \geq 2$ of bounds $b_1, \ldots, b_k$, the problem of deciding whether there exists or not an s-t cut C such that $val(C, s_i) \leq b_i$, for $i = 1, \ldots, k$, is strongly *NP*-hard. Armon and Zwick [3, Th.2] reduced this problem to the min-max version. Combining these results, we can state the following result.

Theorem 1 Discrete Min-Max s-t Cut *is strongly NP-hard even for two scenarios.*

Using this result, we show now that the min-max regret version is also strongly *NP*-hard.

Theorem 2 Discrete Min-Max Regret Min s-t Cut *is strongly NP-hard even for two scenarios.*

Proof: Consider an instance $G = (V, E)$ of Discrete Min-Max s-t Cut with the scenario set $S = \{s_1, s_2\}$. Let W be the total sum of the weights of all edges under all scenarios. We construct an instance G' of Discrete Min-Max Regret Min s-t Cut with the same scenario set. The graph G' is obtained from G by adding two new vertices s' and t' and edges (s', s) and (t', t). The weights of edges of G in S are kept in G'. Moreover, the weights of edges (s', s) and (t', t) are $w^1_{s's} = 0$, $w^1_{t't} = W$ under scenario s_1, and $w^2_{s's} = W$ $w^2_{t't} = 0$ under scenario s_2. Thus, the optimal values in G' under scenarios s_1 and s_2 are zero. Therefore, a solution is optimal in G if and only if it is optimal in G'. $\square$

3.2 Min cut

For a constant number of scenarios, Armon and Zwick [3] gave a polynomial-time algorithm for Discrete Min-Max Cut based essentially on the result of Nagamochi, Nishimura, and Ibaraki [12] for computing all α-approximate cuts in time $O(m^2n + mn^{2\alpha})$. A cut C in a graph G is called an α-approximate cut if $val(C) \leq \alpha\, opt$, where opt is the value of a minimum cut in G.

Theorem 3 ([3]) Discrete Min-Max Cut *is solvable in polynomial time for a constant number of scenarios.*

In a graph on n vertices and m edges and with k scenarios, Armon and Zwick's algorithm [3] constructs an optimal solution in $O(mn^{2k})$.

We show in the following that this algorithm can be modified in order to obtain a polynomial-time algorithm for Discrete Min-Max Regret Min Cut.

Theorem 4 Discrete Min-Max Regret Min Cut *is solvable in polynomial time for a constant number of scenarios.*

Proof: Consider an instance I of the problem given by graph $G = (V, E)$ on n vertices and m edges and a set of k scenarios S such that each edge $(i, j) \in E$ has a weight w^s_{ij} in scenario s. We construct, as before, an instance I' of Min Cut on the same graph, where $w'_{ij} = \sum_{s \in S} w^s_{ij}$. The algorithm consists firstly of computing all k-approximate cuts and secondly of choosing among these cuts one with a minimum maximum regret.

The first stage requires $O(mn^{2k})$ time using the algorithm presented in [12]. In the second stage, we first compute a minimum cut for each scenario, which can be performed in time $O(mn + n^2 \log m)$ using [11, 14]. Knowing, from [7, 8], that we have at most $O(n^{2k})$ k-approximate cuts in I', the complexity of the second stage is $O(mn^{2k})$. Thus, the overall running time of the algorithm is $O(mn^{2k})$.

We prove now the correctness of the algorithm. Let C^* be an optimal min-max regret cut in G. We show that for any cut C of G, we have $val'(C^*) \le k\, val'(C)$, where $val'(C)$ is the value of cut C in I'. In fact,

$$val'(C^*) = \sum_{s \in S} val(C^*, s) = \sum_{s \in S} (val(C^*, s) - val^*_s) + \sum_{s \in S} val^*_s \le$$

$$k \max_{s \in S} \{val(C^*, s) - val^*_s\} + \sum_{s \in S} val^*_s \le k \max_{s \in S} \{val(C, s) - val^*_s\} + \sum_{s \in S} val^*_s \le$$

$$k \sum_{s \in S} (val(C, s) - val^*_s) + \sum_{s \in S} val^*_s = k \sum_{s \in S} val(C, s) - (k-1) \sum_{s \in S} val^*_s \le k\, val'(C)$$

In particular, if C is a minimum cut in I', we obtain $val'(C^*) \le k\, opt(I')$. Thus all optimal solutions to Discrete Min-Max Regret Min Cut are among the k-approximate cuts in I'. □

The algorithms described above to solve min-max (regret) versions of the min cut problem are exponential in k. Armon and Zwick [3, Th.7] showed that given a non constant number k of bounds $b_1, \ldots, b_k$, the problem of deciding whether there exists or not a min cut C such that $val(C, s_i) \le b_i$, for $i = 1, \ldots, k$, is strongly NP-hard. Using again the reduction of [3, Th.2], from this problem to the min-max version, we can state the following result.

Theorem 5 Discrete Min-Max Cut *is strongly NP-hard for a non constant number of scenarios.*

We prove in the following that when k is not constant, the min-max regret version becomes also strongly NP-hard.

Theorem 6 DISCRETE MIN-MAX REGRET MIN CUT *is strongly NP-hard for a non constant number of scenarios.*

Proof: Consider an instance $G = (V, E)$ of DISCRETE MIN-MAX CUT with the scenario set $S = \{s_1, s_2, \ldots, s_k\}$. Let W be the total sum of the weights of all edges under all scenarios. We construct an instance G' of DISCRETE MIN-MAX REGRET MIN CUT with the same scenario set. The graph G' is obtained from G by adding two new vertices v_1 and v_2 and edges (v_1, v) and (v_2, v) for an arbitrarily chosen vertex v of G. The weights of edges of G in S are kept in G'. Moreover, $w^1_{v_1 v} = W$, $w^2_{v_2 v} = W$ and the weights of (v_1, v) and (v_2, v) in the other scenarios are 0. Thus, the optimal values in G' under each scenario are zero. Therefore, a solution is optimal in G if and only if it is optimal in G'. $\square$

4 Interval scenario case

We first state the polynomiality of the min-max cut problems (Section 4.1), then we establish the strong *NP*-hardness of INTERVAL MIN-MAX REGRET MIN s-t CUT (Section 4.2.1) and the polynomiality of INTERVAL MIN-MAX REGRET MIN CUT (Section 4.2.2).

4.1 Min-max versions

In the interval scenario case, the min-max version of a minimization problem corresponds to solving this problem in the worst-case scenario defined by the upper bounds of all intervals. Therefore, a minimization problem and its min-max version have the same complexity. INTERVAL MIN-MAX s-t CUT and INTERVAL MIN-MAX CUT are thus polynomial-time solvable.

4.2 Min-max regret versions

When the number $u \leq m$ of uncertain/imprecise parameters, corresponding to non-degenerate intervals, is small enough, then the problem becomes polynomial. More precisely, as shown by Averbakh and Lebedev [5] for general networks problems solvable in polynomial time, if u is fixed or bounded by the logarithm of a polynomial function of m, then the min-max regret version is also solvable in polynomial time (based on the fact that an optimal solution for the min-max regret version corresponds to one of the optimal solutions for the 2^u extreme scenarios, where extreme scenarios have values on each edge corresponding to either the lower or upper bound of its interval). This clearly applies to the min s-t cut and min cut problems.

4.2.1 Min s-t cut

We show now that INTERVAL MIN-MAX REGRET MIN s-t CUT is strongly *NP*-hard. For this purpose, we construct a reduction from the decision version of MIN s-t BISECTION.

Theorem 7 INTERVAL MIN-MAX REGRET MIN s-t CUT *is strongly NP-hard.*

Proof: Consider $G = (V, E)$ an instance of MIN s-t BISECTION with $|V| = 2n$, where $V = \{s = 1, \ldots, t = 2n\}$. We construct from G an instance $\widetilde{G} = (\widetilde{V}, \widetilde{E})$ of INTERVAL MIN-MAX REGRET MIN s-t CUT as illustrated in Figure 1. The vertex set is $\widetilde{V} = V \cup \{1', \ldots, 2n'\}$ $\cup\{1'', \ldots, 2n''\}$ $\cup\{1''', \ldots, 2n'''\}$ $\cup\{\widetilde{s}, 2n+1\}$, and $\widetilde{t} = t$.

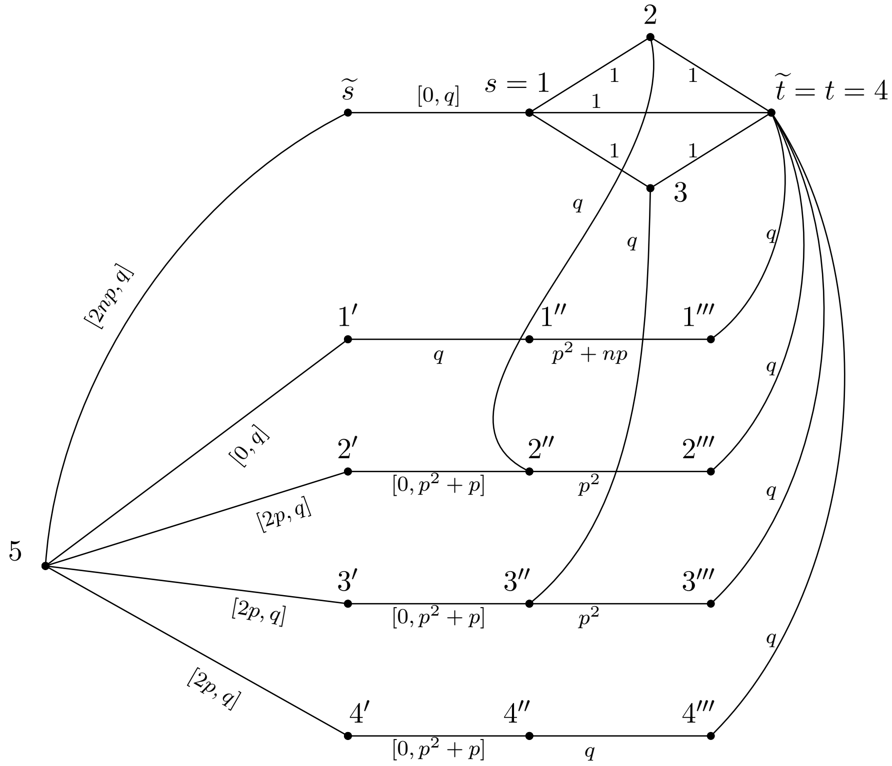

Figure 1: Interval Min-Max Regret Min s-t Cut instance resulting from a Min s-t Bisection instance

The edge set is $\widetilde{E} = E \cup \{(i', i''), (i'', i''') : i = 1, \ldots, 2n\} \cup \{(i, i'') : i = 2, \ldots, 2n-1\} \cup \{(2n+1, i') : i = 1, \ldots, 2n\} \cup \{(i''', t) : i = 1, \ldots, 2n\} \cup \{(\widetilde{s}, 2n+1), (\widetilde{s}, s)\}$.

Let p and q verifying, respectively, $p > n^2$ and $q > 4n(p+1)^2$. The weights are defined as follows :

- $\underline{w}_{ij} = \overline{w}_{ij} = 1$, for all $(i, j) \in E$;
- $\underline{w}_{i'i''} = \begin{cases} q & \text{for } i = 1 \\ 0 & \text{otherwise} \end{cases}$ and $\overline{w}_{i'i''} = \begin{cases} q & \text{for } i = 1 \\ p^2 + p & \text{otherwise} \end{cases}$
- $\underline{w}_{i''i'''} = \overline{w}_{i''i'''} = \begin{cases} p^2 + np & \text{for } i = 1 \\ p^2 & \text{for } i = 2, \ldots, 2n-1 \\ q & \text{for } i = 2n \end{cases}$
- $\underline{w}_{ii''} = \overline{w}_{ii''} = q$, for $i = 2, \ldots, 2n-1$;
- $\underline{w}_{(2n+1)i'} = \begin{cases} 0 & \text{for } i = 1 \\ 2p & \text{otherwise} \end{cases}$ and $\overline{w}_{(2n+1)i'} = q$, for $i = 1, \ldots, 2n$;
- $\underline{w}_{i'''t} = \overline{w}_{i'''t} = q$, for $i = 1, \ldots, 2n$;
- $\underline{w}_{\widetilde{s}(2n+1)} = 2np$ and $\overline{w}_{\widetilde{s}(2n+1)} = q$;
- $\underline{w}_{\widetilde{s}s} = 0$ and $\overline{w}_{\widetilde{s}s} = q$.

Clearly this transformation can be obtained in polynomial time.

We first establish the following property.

For any $\widetilde{s}$-$\widetilde{t}$ cut $\widetilde{C} = (\widetilde{V}_1, \widetilde{V}_2)$ in $\widetilde{G}$ not including any edge $(i,j) \in \widetilde{E}$ with $\overline{w}_{ij} = q$, a minimum $\widetilde{s}$-$\widetilde{t}$ cut $C^*_{w^-(\widetilde{C})}$ in, $w^-(\widetilde{C})$, the worst scenario associated to $\widetilde{C}$, has value $val(C^*_{w^-(\widetilde{C})}, w^-(\widetilde{C})) = 2p\min\{n, |V_2|\}$, where $V_2 = \widetilde{V}_2 \cap V$.

Indeed, consider such a cut $\widetilde{C} = (\widetilde{V}_1, \widetilde{V}_2)$ with $\widetilde{s} \in \widetilde{V}_1$, $\widetilde{t} \in \widetilde{V}_2$ and denote $V_1 = \widetilde{V}_1 \cap V$. Clearly, vertices $2n+1$, $1''$ and i', $i = 1, \dots, 2n$ belong to $\widetilde{V}_1$. Also, vertices $2n''$ and i''', $i = 1, \dots, 2n$ belong to $\widetilde{V}_2$. Moreover, i and i'' belong to the same part, $\widetilde{V}_1$ or $\widetilde{V}_2$. It follows that

$$val(\widetilde{C}, w^-(\widetilde{C})) = x + (n + |V_2|)p + 2np^2 \tag{1}$$

where x denotes the number of edges that have one endpoint in V_1 and one endpoint in V_2.

By construction, $C^*_{w^-(\widetilde{C})}$ necessarily cuts edge $(\widetilde{s}, s)$. Furthermore, there exist two cases:

1. If $|V_2| \leq n$ then $C^*_{w^-(\widetilde{C})} = (\widetilde{V}_1^*, \widetilde{V} \setminus \widetilde{V}_1^*)$, where $\widetilde{V}_1^* = \{\widetilde{s}, 2n+1\} \cup \{i' : i'' \in \widetilde{V}_1, i \neq 1\}$ and thus $val(C^*_{w^-(\widetilde{C})}, w^-(\widetilde{C})) = 2|V_2|p$.

2. If $|V_2| > n$ then $C^*_{w^-(\widetilde{C})} = (\{\widetilde{s}\}, \widetilde{V} \setminus \{\widetilde{s}\})$ and thus $val(C^*_{w^-(\widetilde{C})}, w^-(\widetilde{C})) = 2np$.

We claim that there exists an s-t bisection $C = (V_1, V_2)$ of value no more than v if and only if there exists an $\widetilde{s}$-$\widetilde{t}$ cut $\widetilde{C} = (\widetilde{V}_1, \widetilde{V}_2)$ in $\widetilde{G}$ with $R_{max}(\widetilde{C}) \leq v + 2np^2$.

$\Rightarrow$ Consider an s-t bisection $C = (V_1, V_2)$ in G of value $x \leq v$. We construct an $\widetilde{s}$-$\widetilde{t}$ cut $\widetilde{C}$ in $\widetilde{G}$ deduced from C as follows: $\widetilde{V}_1 = \{\widetilde{s}, 2n+1\} \cup \{1', \dots, 2n'\} \cup V_1 \cup \{i'' : i \in V_1\}$ and $\widetilde{V}_2 = \{1''', \dots, 2n'''\} \cup V_2 \cup \{i'' : i \in V_2\}$. It is easy to verify that $val(\widetilde{C}, w^-(\widetilde{C})) = x + 2n(p+p^2)$ and using the previous result, we have $R_{max}(\widetilde{C}) = x + 2np^2 \leq v + 2np^2$.
$\Leftarrow$ Consider an $\widetilde{s}$-$\widetilde{t}$ cut $\widetilde{C}$ in $\widetilde{G}$ with $R_{max}(\widetilde{C}) \leq v + 2np^2$. We first show that cut $\widetilde{C}$ does not cut any edge $(i,j) \in \widetilde{E}$ such that $\overline{w}_{ij} = q$. Otherwise, we would have $val(\widetilde{C}, w^-(\widetilde{C})) \geq q$. Moreover, since a minimum $\widetilde{s}$-$\widetilde{t}$ cut $C^*_{w^-(\widetilde{C})}$ in $w^-(\widetilde{C})$ does not cut any edge $(i,j) \in \widetilde{E}$ such that $\overline{w}_{ij} = q$, we have, using (1), $val(C^*_{w^-(\widetilde{C})}, w^-(\widetilde{C})) \leq n^2 + 3np + 2np^2 < 4np + 2np^2$. Therefore, we have $R_{max}(\widetilde{C}) > q - (4np + 2np^2) > 2np^2 + v$, a contradiction.

Thus $val(\widetilde{C}, w^-(\widetilde{C})) = y + 2np^2 + np + p|V_2|$ where y is the value of the cut induced by $\widetilde{C}$ in E. It follows that

$$R_{max}(\widetilde{C}) = \begin{cases} y + (n - |V_2|)p + 2np^2 & \text{if } |V_2| \leq n \\ y + (|V_2| - n)p + 2np^2 & \text{if } |V_2| > n \end{cases}$$

Consequently, since $R_{max}(\widetilde{C}) \leq v + 2np^2$, and $p > n^2 \geq v$, we have $|V_1| = n = |V_2|$ and $y \leq v$. $\square$

4.2.2 Min cut

We prove in this section that the min-max regret version of the min cut problem is polynomial in the interval scenario case.

Theorem 8 Interval Min-Max Regret Min Cut *is solvable in polynomial time in the interval scenario case.*

Proof: Consider an instance I of Interval Min-Max Regret Min Cut given by graph $G = (V, E)$ on n vertices and m edges. The weight w_{ij} of each edge $(i, j) \in E$ can take any value in the interval $[\underline{w}_{ij}, \overline{w}_{ij}]$. We construct an instance I' of Min Cut on the same graph, where $w'_{ij} = \overline{w}_{ij}$. The algorithm consists firstly of computing all the 2-approximate minimum cuts in I' and secondly of choosing among these cuts one with a minimum maximum regret.

The running time of the first stage is $O(mn^4)$ using the algorithm presented in [12]. For the second stage, the complexity for computing the maximum regret for any cut C corresponds to the complexity for computing a minimum cut in scenario $w^-(C)$, that is $O(mn + n^2 \log m)$ using [11, 14]. Knowing, from [7, 8], that we have at most $O(n^4)$ 2-approximate cuts in I', the complexity of the second stage is $O(mn^5 + n^6 \log m)$. Thus, the overall running time of the algorithm is $O(mn^5 + n^6 \log m)$.

We prove now the correctness of the algorithm. Let C^* be an optimal cut in I and $val'(C)$ denote the value of any cut C in I'. Then the following inequalities hold:

$$\begin{aligned} val'(C^*) &= R_{max}(C^*) + val^*_{w^-(C^*)} \\ &\leq R_{max}(C) + val(C, w^-(C^*)) \leq 2val'(C) \end{aligned}$$

In particular, if C is a minimum cut in I', we obtain $val'(C^*) \leq 2opt(I')$. Thus all optimal solutions to Interval Min-Max Regret Min Cut are among the 2-approximate cuts in I'. □

5 Conclusions

We reviewed in this paper positive and negative results concerning the complexity of min-max and min-max regret versions of the min cut and min s-t cut problems. Table 1 summarizes all the complexity results. Besides the fact that two closely related polynomial problems have widely contrasted complexity status for their min-max (regret) versions, it should be pointed out that, for a constant number of scenarios, min s-t cut is the first known polynomial problem whose min-max (regret) versions become strongly NP-hard whereas min cut is one of the few polynomial problems whose min-max (regret) versions remain polynomial.

Problem			**min cut**	**min s-t cut**
Discrete case	constant	min-max	polynomial [3]	strongly NP-hard
		min-max regret	polynomial	strongly NP-hard
	non constant	min-max	strongly NP-hard	strongly NP-hard
		min-max regret	strongly NP-hard	strongly NP-hard
Interval case		min-max	polynomial	polynomial
		min-max regret	polynomial	strongly NP-hard

Table 1: Complexity results of the min-max (regret) versions of min cut and min s-t cut

Now that the complexity status of these problems is clarified, it would be interesting to study their approximability. Observing that all the negative results are strong NP-hardness results, the best approximations that we could obtain for these problems are polynomial time approximation schemes. Moreover, we know two general results for the approximability of min-max (regret) versions of *polynomial-time solvable* problems: a k-approximation algorithm in the discrete scenario case by Kouvelis and Yu [10] and a 2-approximation algorithm in the interval scenario case by Kasperski and Zieliński [9]. It remains an open question whether the approximability of min-max (regret) versions of min cut and, above all, min s-t cut problems can be improved using specific algorithms.

Acknowledgements: We would like to thank the anonymous referees for their remarks that allowed us to improve the presentation of our paper. This work was partially supported by the ACI sécurité Informatique TADORNE grant (2004-2007).

References

[1] H. Aissi, C. Bazgan, and D. Vanderpooten. Approximation complexity of min-max (regret) versions of shortest path, spanning tree, and knapsack. In *Proc. of the 13th Annual European Symposium on Algorithms (ESA 2005)*, LNCS 3669, 862–873. Springer. Also appeared in *European Journal of Operational Research*, 179(2): 281-290, 2007.

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

[3] A. Armon and U. Zwick. Multicriteria global minimum cuts. In *Proc. of the 15th International Symposium on Algorithms and Complexity (ISAAC 2004)*, LNCS 3341, 65–76. Springer. Also appeared in *Algorithmica*, 46:15-26, 2006.

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

[5] I. Averbakh and V. Lebedev. Interval data min-max regret network optimization problems. *Discrete Applied Mathemathics*, 138:289–301, 2004.

[6] M. Garey and D. Johnson. *Computer and Intractability: A Guide to the theory of NP-completeness*. San Francisco, 1979.

[7] D. R. Karger. Global min-cuts in RNC, and the other ramifications of a simple min cut algorithm. In *Proc. of the 4th ACM-SIAM Symposium on Discrete Algorithms*, 21–30, 1993.

[8] D. R. Karger and C. Stein. A new approach to the minimum cut problem. *Journal of the ACM*, 43(4):601–640, 1996.

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

[10] P. Kouvelis and G. Yu. *Robust Discrete Optimization and its Applications*. Kluwer Academic Publishers, Boston, 1997.

[11] H. Nagamochi and T. Ibaraki. Linear time algorithm for finding a sparse k-connected spanning subgraph of a k-connected graph. *Algorithmica*, 7:583–596, 1992.

[12] H. Nagamochi, K. Nishimura, and T. Ibaraki. Computing all small cuts in an undirected network. *SIAM Journal on Discrete Mathematics*, 10(3):469–481, 1997.

[13] 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*, 86–92, 2000.

[14] M. Stoer and F. Wagner. A simple min cut algorithm. *Journal of th ACM*, 44(4):585–591, 1999.

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