Some tractable instances of interval data minmax regret problems

Bruno Escoffier [a,*] , Jérôme Monnot [a] , Olivier Spanjaard [b]

[a] *LAMSADE-CNRS, Université Paris Dauphine, Place du M^{al} de Lattre de Tassigny, F-75775 Paris Cedex 16, France*

[b] *LIP6, Université Pierre et Marie Curie, 4 Place Jussieu, F-75252 Paris Cedex 05, France*

Abstract

In this paper, we provide polynomial and pseudopolynomial algorithms for classes of particular instances of interval data minmax regret graph problems. These classes are defined using a parameter that measures the distance from well known solvable instances. Tractable cases occur when the parameter is bounded by a constant.

Key words: Robust optimization; Interval data; Shortest path; Spanning tree; Bipartite perfect matching

1. Introduction

In recent years there has been a growing interest in robust optimization problems [17]. Studies in this field concern problems where some parameters are not well known due to uncertainty or imprecision. Usually, in weighted graph optimization problems, the uncertain or imprecise parameters are the weights. In such a case, a set of scenarios is defined, with one scenario for each possible assignment of weights to the graph. Two approaches can be distinguished according to the way the set of scenarios is defined: the *interval model* where each weight is an interval and the set of scenarios is defined implicitly as the Cartesian product of all the intervals; the *discrete scenario model* where each weight is a vector, every component of which is a particular scenario. Intuitively, a *robust* solution is a solution that remains suitable whatever scenario finally occurs. Several criteria have been proposed to formalize this: the *minmax* criterion consists of evaluating a solution on the basis of its worst value over all scenarios, and the *minmax regret* criterion consists of evaluating a solution on the basis of its maximal deviation from the optimal value over all scenarios. We will mainly focus here on the *robust shortest path* problem (RSP for short), the *robust minimum spanning tree* problem (RST for short) and the *robust minimum weighted (bipartite) perfect matching* problem (R(B)PM for short), with the minmax regret criterion in the interval model. Although not crucial for our results, let us emphasize that we consider here RSP in directed graphs (while RST and R(B)PM deal of course with undirected graphs).

Formally, an interval data minmax regret network optimization problem can be defined as follows. Let $G = (V, E)$ be a given directed or undirected graph (in the sequel, unless otherwise stated, we denote $n = |V|$ and $m = |E|$). A feasible solution is a subset $\pi \subseteq E$ satisfying a given property Π (for example, being a path, a tree or a matching). Each edge $e \in E$ is valued by an interval $I_e = [l_e; u_e]$ of possible weights, where l_e and u_e are nonnegative integers. The set of scenarios is the Cartesian product $\mathcal{S} = \prod_{e\in E} I_e$. In other words, a scenario $s \in \mathcal{S}$ consists of assigning a weight $w_s(e) \in I_e$ for every $e \in E$. For any feasible solution π and any scenario $s \in \mathcal{S}$ of an instance $\mathcal{I} = (G, I_E)$ where $I_E = \{I_e : e \in E\}$, the *value of* π *under scenario* s is $w_s(\pi) = \sum_{e\in\pi} w_s(e)$ and its *regret under scenario* s is $R_s(\pi) = |w_s(\pi) - opt(s)|$, where $opt(s)$ is the value of an optimal solution for the standard instance valued by w_s (rigorously, we should write $R_s(\mathcal{I}, \pi)$ but we omit $\mathcal{I}$ when no confusion is possible). The max regret of solution π is defined by $R(\pi) = \max_{s\in\mathcal{S}} R_s(\pi)$. The aim of a minmax regret optimization problem is, given an instance $\mathcal{I} = (G, I_E)$, to find a feasible solution π^* minimizing $R(\pi^*)$. Note that, for a minimization problem, $R(\pi) = R_{s(\pi)}(\pi)$, where $s(\pi)$, called *worst case scenario for* π, is defined by $w_{s(\pi)}(e) = u_e$ if $e \in \pi$ and $w_{s(\pi)}(e) = l_e$ otherwise [3].

In this paper, we consider tractable instances of RSP and RST, that have been proved strongly NP-hard [4] in the general case, as well as tractable instances of RBPM, the restriction of which to complete bipartite graphs (known as the interval data minmax regret assignment problem) has

* Corresponding author.

Email addresses: escoffier@lamsade.dauphine.fr (Bruno Escoffier), monnot@lamsade.dauphine.fr (Jérôme Monnot), olivier.spanjaard@lip6.fr (Olivier Spanjaard).

been proved NP-hard [13]. For this purpose, as suggested by Guo *et al.* [11], we introduce parameters that measure the distance from well known solvable instances. For example, if all the intervals of an instance reduce to a single point –*degenerate* intervals–, then the robust optimization problem reduces to a standard optimization problem, and is therefore polynomially solvable provided that the standard version is polynomial. One can define the distance from this trivial case as the number k of non degenerate intervals. If this distance k is bounded by a constant, then the robust optimization problem is polynomially solvable by a brute force algorithm [4]. In this work, we focus on two kinds of parameters: some that measure the distance from special weight structures (instances for which the minmax regret is zero, instances with linearly ordered weights), and a one that measures the distance from a special graph structure (tree). The paper is organized as follows. The first two sections deal with the first kind of parameters: we show that RSP and RBPM are polynomially solvable when the minmax regret is bounded by a constant k (Section 2), as well as RST when the number of intersecting intervals in the instance is bounded by a constant k (Section 3). More precisely, following parameterized complexity terminology [8], the first two problems are in XP (problems solvable in $O(n^{f(k)})$ for some function f) while the third one is in FPT (problems solvable in $O(f(k)n^c)$ for some constant c). The next section deals with graphs close to be trees: we show that RSP and RBPM are pseudopolynomial for graphs with bounded treewidth and bounded degree (Section 4).

2. Upper bounded minmax regret

In this section, we investigate the hardness of solving an interval data minmax regret graph optimization problem when there exists a solution with bounded maximal regret. Note that studying instances where the optimum value is upper bounded is a classical way to understand the intrinsic difficulty of a combinatorial optimization problem (problems which become polynomially solvable in this case are called *simple*, see Paz and Moran [18]). Here, we first show that we can easily determine if there is a solution of maximal regret 0, *i.e.* a solution which is optimal under every possible scenario. Next, we show that for RSP and RBPM, we can extend this result to polynomially determine if there exists a solution of maximal regret at most k.

First, let us prove that the problem of the existence of a solution of maximal regret 0 can be easily solved for any interval data minmax regret graph optimization problem Π. We use a generic 2-approximation algorithm proposed by Kasperski and Zielinski [14]. For any instance $\mathcal{I}$ this algorithm outputs a solution π such that $R(\pi) \leq 2R(\pi^*)$ (where $R(\pi^*)$ is the minmax regret of $\mathcal{I}$). If $R(\pi^*) = 0$, then $R(\pi) = 0$, else since $R(\pi) \geq R(\pi^*)$, we have $R(\pi) > 0$. The expected result follows (Π being assumed to be polynomial). Now, by a reduction to the regret 0 case, we prove the following:

Proposition 1 *For RSP, the problem of determining if the minmax regret is at most k can be solved in time $O(n^2 m^k)$.*

Proof. Let $\mathcal{I} = (G, I_E)$ be an instance of RSP and denote by r its optimum regret. Let us remark that if there exists a degenerate interval $I_e = \{0\}$ in $\mathcal{I}$ with $e = (v_1, v_2)$, then one can merge nodes v_1 and v_2 and get an equivalent instance (possibly with multiedges). In particular, we can assume that $u_e > 0$ for any e. We construct m instances $\mathcal{I}_1, \dots, \mathcal{I}_m$ of RSP as follows: $\mathcal{I}_i$ is the same instance as $\mathcal{I}$ up to the interval $[l_i, u_i]$ associated in $\mathcal{I}$ to e_i which is transformed into $[\max\{l_i - 1; 0\}, u_i - 1]$ (we take $\max\{l_i - 1; 0\}$ to fit with the usual assumption that l_i and u_i are nonnegative). We claim that:

(i) $r_i^* \geq r - 1$ where r_i^* denotes the optimum regret of $\mathcal{I}_i$;

(ii) if $r_i^* = r - 1$ then any optimum solution for $\mathcal{I}_i$ is optimum for $\mathcal{I}$;

(iii) there exists at least one i such that $r_i^* = r - 1$ (if $r > 0$).

If the claims are true, then by applying this construction recursively to depth k, $\mathcal{I}$ has an optimum regret at most k if and only if (at least) one of the final instances has optimum regret 0 (if at some point, we find an interval reduced to $\{0\}$, we can merge the corresponding nodes). We get m^k instances; the generic 2-approximation algorithm is in $O(n^2)$ for RSP, and the complexity follows. Claims (i) and (ii) hold since the regret of any path π satisfies $R_i(\pi) \geq R(\pi) - 1$ (under any scenario, the value of any path has decreased by at most 1). For Claim (iii), consider an optimum solution $\pi^* = ((v_0, v_1), \cdots, (v_{p-1}, v_p))$ (where $v_0 = s$ and $v_p = t$) of $\mathcal{I}$, and its worst case scenario $s(\pi^*)$ in $\mathcal{I}$. We prove that there exists at least one edge $e_i \in \pi^*$ such that no shortest path in $s(\pi^*)$ contains this edge. Note that if this is true, then consider instance $\mathcal{I}_i$: in $s(\pi^*)$, the value of the shortest path is the same in $\mathcal{I}$ and in $\mathcal{I}_i$, hence the regret of π^* decreased by 1, and Claim (iii) is true. Then, assume that for any i, there exists a shortest path π^i (in $s(\pi^*)$) which contains (v_{i-1}, v_i). Let w_1^i be the value (in $s(\pi^*)$) of this path between s and v_{i-1} and w_2^i its value between v_i and t (hence $w_1^1 = w_2^p = 0$). Since π^* has regret r, we get ($s(\pi^*)$ is omitted for readability) that $w(\pi^i) = w_1^i + w_2^i + u_{(v_{i-1}, v_i)} = w(\pi^*) - r$. Summing up we obtain:

$$\begin{aligned}\sum_{i=1}^{p}(w_1^i + w_2^i) &= pw(\pi^*) - pr - \sum_{i=1}^{p} u_{(v_{i-1}, v_i)} \\ &= (p-1)w(\pi^*) - pr\end{aligned} \tag{1}$$

But remark that for each $i \in \{2, \cdots, p\}$ we can build a path of value $w_1^i + w_2^{i-1}$ (composed of the initial part of π^i from s to v_{i-1} and the final part of π^{i-1} from v_{i-1} to t). Then, since each of these paths has value at least $w(\pi^*) - r$:

$$\begin{aligned}\sum_{i=2}^{p}(w_1^i + w_2^{i-1}) &\geq (p-1)(w(\pi^*) - r) \\ &= (p-1)w(\pi^*) - pr + r\end{aligned} \tag{2}$$

But since $w_1^1 = w_2^p = 0$, Equations (1) and (2) are incompatible for $r > 0$. □

The central property, leading to Claim (iii), is that, in an optimum solution π^* for which $R(\pi^*) > 0$, there exists at least one edge that does not belong to any optimum solution (*i.e.* any shortest path) in $s(\pi^*)$. Actually, one can show that this property is also true for the interval data minmax regret perfect matching problem in bipartite graphs. For any instance $\mathcal{I} = (G, I_E)$ of R(B)PM, we assume that G has a perfect matching (in particular, the number n of vertices of G is even).

Proposition 2 *For RBPM, the problem of determining if the minmax regret is at most k can be solved in time $O(n^2 m^k)$.*

Proof. The proof is quite identical to the one of Proposition 1. Let $\mathcal{I} = (G, I_E)$ be an instance of RBPM where $G = (V, E)$ is a bipartite graph which admits a perfect matching and denote by r its optimum regret. W.l.o.g., assume that $l_e \geq k$ for any e. Actually, by adding any constant $c > 0$ to each interval I_e, we obtain an equivalent instance since all the perfect matchings have the same size. As previously, we build m instances $\mathcal{I}_1, \ldots, \mathcal{I}_m$ of RBPM where $\mathcal{I}_i$ is the same instance as $\mathcal{I}$ up to the interval $[l_i, u_i]$ associated in $\mathcal{I}$ to e_i which is transformed to $[l_i - 1, u_i - 1]$. Using the same notation than those given in Proposition 1, we claim that: (i) $r_i^* = R(\mathcal{I}_i) \geq r - 1$; (ii) if $r_i^* = r - 1$ then any optimum solution for $\mathcal{I}_i$ is optimum for $\mathcal{I}$; (iii) there exists at least one i such that $r_i^* = r - 1$ (if $r > 0$).

The proof of Claims (i) and (ii) is identical to the proof of Proposition 1. So, we only prove Claim (iii). Consider an optimum solution $\pi^* = \{e_1, \cdots, e_{\frac{n}{2}}\}$ of $\mathcal{I}$, and its worst case scenario $s(\pi^*)$ in $\mathcal{I}$. As previously, we prove that there exists at least one edge $e_i \in \pi^*$ such that no perfect matching with minimum weight in $s(\pi^*)$ contains this edge. Assume the reverse, and let π^i for $i = 1, \cdots, \frac{n}{2}$ be a perfect matching with minimum weight $w(\pi^*) - r$ which contains edge e_i in scenario $s(\pi^*)$ (note that possibly some π^i are identical). Then, in scenario $s(\pi^*)$ we have:

$$\sum_{i=1}^{\frac{n}{2}} w(\pi^i \setminus e_i) = \frac{n-2}{2} w(\pi^*) - \frac{n}{2} r \qquad (3)$$

On the other hand, the graph G' induced by $\cup_{i=1}^{\frac{n}{2}} \left(\pi^i \setminus e_i\right)$ is $(\frac{n}{2} - 1)$-regular (G' is considered as a multigraph, that is if an edge (x, y) appears p times in $\cup_{i=1}^{\frac{n}{2}} \left(\pi^i \setminus e_i\right)$, then there are p parallel edges between x and y in G'). Since G' is bipartite and $(\frac{n}{2} - 1)$-regular, G' can be decomposed into $(\frac{n}{2} - 1)$ matchings π'^i for $i = 1, \ldots, \frac{n}{2} - 1$ (by König's Theorem, [16]). These matchings π'^i are perfect in G and if π' is a matching of minimum weight in scenario $s(\pi^*)$ among the matchings π'^i for $i = 1, \ldots, \frac{n}{2} - 1$, then the value of π' satisfies:

$$\frac{n-2}{2} w(\pi') \leq \sum_{i=1}^{\frac{n}{2}} w(\pi^i \setminus e_i) \qquad (4)$$

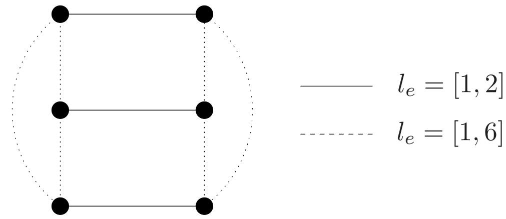

Fig. 1. An example for RPM in a non bipartite graph.

Using equality (3) and inequality (4) we obtain $w(\pi') \leq w(\pi^*) - (1 + \frac{2}{n}) r$, which is impossible for $r > 0$ since $w(\pi') \geq w(\pi^*) - r$.

By applying this construction recursively to depth k, we build m^k instances such that $\mathcal{I}$ has an optimum regret at most k iff (at least) one of the final instances has optimum regret 0. Since we supposed that $\forall e \in E, l_e \geq k$ for the initial instance, all the interval lower bounds in the final instances are non-negative. □

Our method seems to be quite general and may be fruitfully applied to other problems, but however not to all of them. Indeed, the property leading to Claim (iii) is no more true for some problems such as RST or RPM (in arbitrary graphs), and for them the question whether they are simple (according to the definition of [18]) or not remains open. Figure 1 illustrates why the property does not hold for RPM in a non bipartite graph. The solution π^* described by solid lines is the unique optimal solution for RPM. Its worst value is 6, its max regret is 2, and in its worst scenario $s(\pi^*)$, each edge of π^* belongs to a perfect matching of minimum weight.

3. Upper bounded number of interval intersections

As previously mentioned, RST and RSP are fixed parameter tractable (FPT) when the parameter is the number of non degenerate intervals (with a brute force algorithm). Minimum spanning trees have special properties that lead to another easy cost structure: when all intervals are disjoint ($I_e \cap I_f = \emptyset$ for any edges e and f), any minimum spanning tree under any scenario is an optimum solution for RST [1]. Indeed, Kruskal algorithm leads then to the same tree, independently of the scenario. This tree is optimal, and its regret is 0. Note that, on the other hand, even if all intervals are $[0, 1]$, RST is NP-hard [1,4]. Here, we show that RST is FPT when considering as a parameter the number of intervals that intersect at least one other interval. Although using brute force, the optimality of the algorithm is not obvious.

Proposition 3 *RST can be solved in time $O(2^k\ m \log m)$, where k is the number of intervals that intersect at least one other interval.*

Proof. Let $\mathcal{I} = (G, I_E)$ be an instance of RST where $G = (V, E)$ and $I_e = [l_e, u_e]$ for any $e \in E$. We define $J = \{I_{e_1} : \exists e_2 \neq e_1, I_{e_1} \cap I_{e_2} \neq \emptyset\}$, and we set $k = |J|$. Let $J' \subseteq J$.

We want to compute the best (in terms of regret) spanning tree π such that $\pi \cap E_J = E_{J'}$ (where E_J denotes the set of edges corresponding to intervals in J). Indeed, if we are able to do this, we only have to consider each possible $J' \subseteq J$, and take the best solution so computed.

If $E_{J'}$ contains a cycle, there is no tree π such that $\pi \cap E_J = E_{J'}$. If not, we proceed as follows: we remove from E the set $E_{J \setminus J'}$ and, considering $E_{J'}$ as part of the spanning tree, we complete it by applying Kruskal algorithm to the remaining graph (choosing any weight $w(e) \in [l_e, u_e]$ since the output does not depend on the value of an edge $e \notin J$). Let $\pi_{J'}$ be the obtained solution. In the sequel, we show that $\pi_{J'}$ is the best (in terms of regret) spanning tree π such that $\pi \cap E_J = E_{J'}$. Then, the global complexity of our algorithm is bounded by $2^k O(m \log m)$.

Let π be a spanning tree such that $\pi \cap E_J = E_{J'}$. We want to prove that $R(\pi_{J'}) \leq R(\pi)$. First, note that $\pi_{J'}$ and π agree on E_J. Then, under any scenario where $w(e) = u_e$ for $e \in E_{J'}$ and $w(e) = l_e$ for $e \in E_{J \setminus J'}$, Kruskal algorithm will produce the same optimum solution π^*. In particular π^* is optimal *both* in $s(\pi)$ and $s(\pi_{J'})$. However, π^* does not have the same value in these two scenarios. Then:

$$\begin{aligned} R(\pi_{J'}) - R(\pi) = {} & w_{s(\pi_{J'})}(\pi_{J'}) - w_{s(\pi_{J'})}(\pi^*) \\ & - \Big(w_{s(\pi)}(\pi) - w_{s(\pi)}(\pi^*)\Big) \end{aligned}$$

We upper bound this by considering each edge of the graph. If $\pi_{J'}$ and π agree on an edge e (either take it or not), then the difference is 0 for this edge, since this edge has the same value in $s(\pi)$ and $s(\pi_{J'})$, and since we refer to the same tree π^*. Note that this includes all edges in E_J. If $\pi_{J'}$ and π disagree on e, either:

- e is in $\pi_{J'} \setminus \pi$. If e is not in π^*, then in the regret it counts u_e for $\pi_{J'}$ (u_e for $\pi_{J'}$ and 0 for π^*) and 0 for π (0 for π and 0 for π^*). If e is in π^*, it counts 0 for $\pi_{J'}$ and $-l_e$ for π. The loss (in terms of regret) from $\pi_{J'}$ respect to π is therefore at most u_e;
- e is in $\pi \setminus \pi_{J'}$. If e is not in π^*, then it counts 0 for $\pi_{J'}$ and u_e for π. If e is in π^*, it counts $-l_e$ for $\pi_{J'}$ and 0 for π. Then, respect to π, $\pi_{J'}$ "wins" at least l_e.

Summing up these inequalities for all edges leads to:

$$R(\pi_{J'}) - R(\pi) \leq \sum_{e \in \pi_{J'} \setminus \pi} u_e - \sum_{e \in \pi \setminus \pi_{J'}} l_e \tag{5}$$

Now, recall that π and $\pi_{J'}$ agree on J, and that the intervals not in J do not intersect. Hence, whatever the value of edges not in J, $\pi_{J'}$ will have a better value than π. This is true in particular when the weight of each $e \notin J$ is fixed to u_e if e is in $\pi_{J'}$ and to l_e otherwise. This means that

$$\sum_{e \in \pi_{J'} \setminus \pi} u_e \leq \sum_{e \in \pi \setminus \pi_{J'}} l_e \tag{6}$$

Equations (5) and (6) lead to the result that $\pi_{J'}$ is the best tree π such that $\pi \cap J = J'$. □

Note that for RSP, making assumptions on interval intersections does not simplify the problem.

Proposition 4 *RSP is NP-hard even if there are no intersections between intervals.*

Proof. We simply modify the instances given in [15] showing that the problem is NP-hard in series-parallel graphs. We have a set of $n+1$ nodes $v_1, \ldots, v_{n+1}$. There is an edge from v_1 to v_{n+1}, and two edges e_i^1 and e_i^2 from v_i to v_{i+1}, $i = 1, \cdots, n$. Then a path from v_1 to v_{n+1} is either edge (v_1, v_{n+1}) or contains exactly one edge from v_i to v_{i+1}, $i = 1, \cdots, n$. Let M be greater than the largest number of the instance. We replace each edge e_i^1 (resp. e_i^2) by two consecutive edges (v_i, v_i^1) and (v_i^1, v_{i+1}) (resp. (v_i, v_i^2) and (v_i^2, v_{i+1})), where v_i^1 (resp. v_i^2) is a new vertex. Then, if $[l_i^1, u_i^1]$ and $[l_i^2, u_i^2]$ are the intervals of e_i^1 and e_i^2, we set

- the interval of (v_i, v_i^1) to $[4iM + l_i^1, 4iM + u_i^1]$,
- the one of (v_i^1, v_{i+1}) to $[(4i+3)M; (4i+3)M]$,
- the one of (v_i, v_i^2) to $[(4i+1)M + l_i^2, (4i+1)M + u_i^2]$,
- the one of (v_i^2, v_{i+1}) to $[(4i+2)M; (4i+2)M]$.

Moreover, we replace the interval $[l, u]$ of edge (v_1, v_{n+1}) by $[l + K, u + K]$, where $K = \sum_{i=1}^{n} (8i + 3)M = (3n + 4n(n+1))M$. Of course, there are no more intersection between intervals. Moreover, we have added a constant value (namely K) to any path from v_1 to v_n. Since regrets are not modified by this transformation, the hardness follows. □

4. Upper bounded treewidth and max degree

The *treewidth* of a graph can be seen as a measure of how far it is from being a tree (the treewidth of a tree is 1). Kasperski and Zielinski have recently shown that RSP is NP-hard in series-parallel graphs, but admits a pseudopolynomial algorithm in this case [15]. It is well-known that the treewidth of an (undirected) series-parallel graph is at most 2. A natural extension of their result is therefore to investigate complexity of RSP in graphs of bounded treewidth (more precisely, in graphs whose corresponding undirected simple graph has a bounded treewidth). Clearly, RSP is polynomially solvable in a graph G the treewidth of which is $k = 1$ (G is a tree), or the max degree of which is $\Delta \leq 2$ (G is a set of cycles and/or chains). However, it is NP-hard when $k = 2$ and $\Delta = 3$ (since there is a polynomial reduction from the partition problem involving an ESP graph -without multiedges- of max degree 3 [15]). Using a mathematical programming formulation, we show here its pseudopolynomiality for bounded k and Δ.

Proposition 5 *RSP can be solved in time $O((n+m)2^{\Delta(k+1)}((n-1)u_{\max})^{k+1})$ in graphs $G = (V, A)$ of treewidth k and max degree Δ, where $u_{\max} = \max_{(i,j) \in A} u_{ij}$.*

Proof. Let $G = (V, A)$ denote a directed graph with a source node s and a sink node t, and let $G' = (V, E)$ denote the simple undirected graph obtained from G by removing orientation of edges and by simplifying multiedges. Solving

RSP in G amounts to solving the following integer linear program (ILP) [12]:

$$\min \sum_{(i,j)\in A} u_{ij} y_{ij} - x_t \tag{7}$$

$$\text{s.t. } x_j \le x_i + l_{ij} + (u_{ij} - l_{ij}) y_{ij} \quad \forall (i,j) \in A, \tag{8}$$

$$\sum_{k:(j,k)\in A} y_{jk} - \sum_{i:(i,j)\in A} y_{ij} = \begin{cases} 1 & \text{if } j = s \\ \text{-1} & \text{if } j = t \\ 0 & \text{if not} \end{cases} \forall j \in V, \tag{9}$$

$$x_s = 0, y_{ij} \in \{0,1\} \ \forall (i,j) \in A, \ x_j \in \mathbb{N} \ \forall j \in V. \tag{10}$$

The *interaction graph* of an ILP includes a vertex for each variable of the program and an edge between two vertices if both corresponding variables appear in the same constraint. We now show that the program is solvable in pseudopolynomial time by applying a dynamic programming technique on a *tree decomposition* of the interaction graph $IG = (I, U)$, *i.e.* a labeled tree (T, L) such that (a) every node t of T is labeled by a non-empty subset $L(t)$ of V s.t. $\cup_{t\in T} L(t) = V$, (b) for every edge $\{i,j\} \in U$ there is a node t of T whose label $L(t)$ contains both i and j, (c) for every vertex $i \in I$ the nodes of T whose labels include i form a connected subtree of T. The *width* of a tree decomposition is $\max_{t\in T} |L(t)| - 1$. The *treewidth* of IG is the smallest k for which IG has a tree decomposition of width k. If the treewidth of a graph is bounded by a constant k, then a tree decomposition of treewidth at most k can be constructed in linear time (in the number of nodes) [7]. This tree decomposition can itself be converted in linear time in a *nice tree decomposition* of the same width, *i.e.* a rooted tree decomposition such that each node has at most two children, with four types of nodes t: *leaf nodes* with $|L(t)| = 1$, *join nodes* with two children t', t'' s.t. $L(t) = L(t') = L(t'')$, *introduce nodes* with one child t' s.t. $L(t) = L(t') \cup \{v\}$ for some $v \in V$, *forget nodes* with one child t' s.t. $L(t) = L(t') \setminus \{v\}$ for some $v \in V$. The proof of pseudopolynomiality of the approach is in three steps: (i) we show that if the max degree of G and the treewidth of G' are bounded by some constant, then the treewidth of IG is bounded by some constant; (ii) we show how to solve by dynamic programming an ILP whose IG has a bounded treewidth; (iii) we show that the previous approach is pseudopolynomial since variables x_j are upper bounded by $(n-1)u_{\max}$, where $u_{\max} = \max_{(i,j)\in A} u_{ij}$.

Proof of (i). Assume that G' has treewidth k and G has max degree Δ. Note that IG restricted to constraints (9) is the *line graph* of G, *i.e.*, the graph where each vertex represents an edge of G and any two vertices are adjacent iff their corresponding edges are incident. It can be shown that the treewidth of the line graph is at most $\Delta(k+1) - 1$ [2]. Assuming (T, L) is a tree decomposition of width k of G', the idea is to consider the labeled tree (T, L') where $L'(t)$ is the set of edges of G incident to some node in $L(t)$. Indeed, one can show that (T, L') is then a tree decomposition of the line graph [2]. We now show that $(T, L \cup L')$ is a tree decomposition of IG (where we identify a vertex or an edge of G with the corresponding variable in the ILP). For this purpose, one can consider the following partitions of I and U: $I = X \cup Y$, where $X = \{x_j : j \in V\}$ and $Y = \{y_{ij} : (i,j) \in A\}$, and $U = U_X \cup U_Y \cup U_{XY}$, where $U_X = \{[x_i, x_j] : (i,j) \in A\}$, $U_Y = \{[y_{jk}, y_{ij}] : (i,j) \in A, (j,k) \in A\}$ and $U_{XY} = \{[x_i, y_{ij}], [x_j, y_{ij}] : (i,j) \in A\}$. Condition (a) holds since $\cup_{t\in T} L(t) = X$ and $\cup_{t\in T} L'(t) = Y$. Conditions (b) and (c) hold for edges of U_X and for vertices in X since (T, L) is a tree decomposition of G'. They also hold for edges of U_Y and for vertices in Y since (T, L') is a tree decomposition of the line graph. Besides, condition (b) holds for edges of U_{XY} by construction of L'. Hence, $(T, L \cup L')$ is a tree decomposition of IG. Furthermore, the treewidth of IG is upper bounded by $\max_{t\in T} L(t) + \max_{t\in T} L'(t) - 1$ $= k + \Delta(k+1)$.

Proof of (ii). By using a method related to non-serial dynamic programming [6], we now show how to solve an ILP in the following general form:

$$(P) \begin{cases} \min \sum_{j=1}^{n} c_j x_j \\ \sum_{j=1}^{n} a_{ij} x_j \, \mathcal{R}_i \, b_i \text{ where } \mathcal{R}_i \in \{\le, =, \ge\} \quad \forall i \le m \\ x_j \in D_j \quad \forall j \le n \end{cases}$$

For this purpose, let us introduce the notion of subprogram of an ILP. For each node t of the nice tree decomposition T of the interaction graph of the ILP, $P(t)$ denotes the subprogram of P restricted to the variables whose indices belong to $D(t) = \bigcup_{t'} L(t')$ for $t' = t$ or t' a descendant of t:

$$(P(t)) \begin{cases} \min \sum_{j\in D(t)} c_j x_j \\ \sum_{j=1}^{n} a_{ij} x_j \, \mathcal{R}_i \, b_i \quad \forall i : [\forall j, \ (a_{ij} \ne 0 \Rightarrow j \in D(t))] \\ x_j \in D_j, \forall j \in D(t) \end{cases}$$

Given $t \in T$ and σ an assignment of values to variables of $L(t)$ (such that $\sigma(j) \in D_j$), we denote by $R_t(\sigma)$ the minimum value of a feasible solution x of $P(t)$ under the constraint $x_j = \sigma(j) \ \forall j \in L(t)$. One sets $R_t(\sigma) = +\infty$ if no feasible solution of $P(t)$ is compatible with σ. The dynamic programming algorithm consists of traversing the nice tree decomposition in a bottom up manner, and computing recursively the tables R_t for each $t \in T$, where table R_t has an entry $R_t(\sigma)$ for each possible assignment σ: let t be a leaf node, say $L(t) = \{j\}$, then $R_t(\sigma) = c_j \sigma(j)$; let t be a join node with two children t' and t'', then $R_t(\sigma) = R_{t'}(\sigma) + R_{t''}(\sigma) - \sum_{j\in L(t)} c_j \sigma(j)$; let t be an introduce node, say $L(t) = L(t') \cup \{j\}$, then $R_t(\sigma) = +\infty$ if σ violates a constraint of $P(t)$, otherwise $R_t(\sigma) = R_{t'}(\sigma_{t'}) + c_j \sigma(j)$ where $\sigma_{t'}$ denotes assignment σ restricted to the variables in $L(t')$; let t be a forget node, say $L(t) = L(t') \setminus \{j\}$, then $R_t(\sigma) = \min_{d_j \in D_j} \{R_{t'}(\sigma') : \sigma'(k) = \sigma(k) \ \forall k \ne j \text{ and } \sigma'(j) = d_j\}$. The optimum is $\min_\sigma R_r(\sigma)$ at the root node r of the nice tree decomposition.

Proof of (iii). We have $|I| = n + m$ since there are n x_i's and m y_{ij}'s in the ILP formulation of RSP. There are therefore $O(n+m)$ nodes in the nice tree decomposition. Noticing that a table R_t can be computed in time $O(2^{\Delta(k+1)}((n-1)u_{\max})^{k+1})$ since there are at most $\Delta(k+1)$ boolean variables and $k+1$ integer variables in $L(t)$, the result follows. □

This approach based on properties of the interaction graph of an ILP formulation is quite general, and can be also fruitfully applied to RBPM. As in Section 2, for any instance of RBPM, we assume that there exists a perfect matching.

Proposition 6 *RBPM can be solved in time* $O((n+m)2^{\Delta(k+1)}((n+1)u_{\max})^{k+1})$ *in graphs of treewidth* k *and max degree* Δ, *where* $u_{\max} = \max_{(i,j)\in E} u_{ij}$.

Proof. Let $G = (V_1 \cup V_2, E)$ denote an undirected bipartite graph, where V_1 and V_2 partition the set of vertices into two independent sets. Solving RBPM in G amounts to solving the following ILP [13]:

$$\min \sum_{[i,j]\in E} u_{ij}y_{ij} - \left(\sum_{j\in V_2} x_j - \sum_{i\in V_1} x_i\right) \tag{11}$$

$$\text{s.t. } x_j \leq x_i + l_{ij} + (u_{ij} - l_{ij})y_{ij} \quad \forall [i,j] \in E, \tag{12}$$

$$\sum_{j\in V_2} y_{ij} = 1 \quad \forall i \in V_1, \tag{13}$$

$$\sum_{i\in V_1} y_{ij} = 1 \quad \forall j \in V_2, \tag{14}$$

$$y_{ij} \in \{0,1\} \quad \forall [i,j] \in E, \tag{15}$$

$$x_i \in \mathbb{Z} \;\; \forall i \in V_1, \;\; x_j \in \mathbb{Z} \;\; \forall j \in V_2. \tag{16}$$

Variables x_i's and x_j's (resp. constraints (12)) represent potentials assigned to vertices of G (resp. constraints) in the dual version of the weighted perfect matching problem in bipartite graphs. More precisely, given a perfect matching characterized by a vector y of booleans, constraints (12) correspond to the dual version of the problem weighted according to the worst case scenario for y. Hence, $\sum_{j\in V_2} x_j - \sum_{i\in V_1} x_i$ takes the value of the best perfect matching in scenario $s(y)$. Actually, variables x_i's and x_j's are real variables in the formulation of Kasperski and Zielinski [13], which leads to a mixed integer model. However, there always exists integer potentials x_i's and x_j's that are optimal in the dual problem. Indeed, for instance, the primal dual algorithm of Ford and Fulkerson [10] builds an optimal dual solution, the potentials of which are integers within $\{-nu_{\max}, \ldots, u_{\max}\}$ by construction. Therefore the solution obtained by solving the above ILP remains optimal. The proof of Proposition 6 is then identical to the one of Proposition 5, constraints (12) (resp. (13) and (14)) playing the role of constraints (8) (resp. (9)), G the role of G', and $(n+1)u_{\max}$ the role of $(n-1)u_{\max}$. □

Note that we conjecture that RSP and RBPM can be pseudopolynomially solved in graphs with bounded treewidth (without any degree restriction). Besides, another extension of the result on series-parallel graphs, based on the notion of *reduction complexity* ([5]), can be found in the conference version of this article [9].

Acknowledgement: The very useful comments and suggestions of an anonymous referee are gratefully acknowledged.

References

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

[2] A. Atserias, On digraph coloring problems and treewidth duality, URL www.lsi.upc.es/˜atserias/ (2006).

[3] I. Averbakh, On the complexity of a class of combinatorial optimization problems with uncertainty, Mathematical Programming Ser. A 90 (2001) 263–272.

[4] I. Averbakh, V. Lebedev, Interval data minmax regret network optimization problems, Discrete Applied Mathematics 138 (2004) 289–301.

[5] W. W. Bein, J. Kamburowski, M. F. M. Stallmann, Optimal reduction of two-terminal directed acyclic graphs, SIAM J. on Computing 21 (6) (1992) 1112–1129.

[6] U. Bertele, F. Brioschi, Nonserial Dynamic Programming, Academic Press, 1972.

[7] H. L. Bodlaender, A linear-time algorithm for finding tree-decompositions of small treewidth, SIAM J. on Computing 25 (6) (1996) 1305–1317.

[8] R. G. Downey, M. R. Fellows, Parameterized Complexity, Springer, 1999.

[9] B. Escoffier, J. Monnot, O. Spanjaard, Some tractable instances of interval data minmax regret problems: Bounded distance from triviality, in: V. Geffert, J. Karhumäki, A. Bertoni, B. Preneel, P. Návrat, M. Bieliková (eds.), SOFSEM, vol. 4910 of Lecture Notes in Computer Science, Springer, 2008.

[10] M. Gondran, M. Minoux, Graphs and algorithms, John Wiley & Sons, 1984.

[11] J. Guo, F. Hüffner, R. Niedermeier, A structural view on parameterizing problems: Distance from triviality, in: IWPEC 2004, vol. 3162 of LNCS, 2004.

[12] O. E. Karasan, M. C. Pinar, H. Yaman, The robust shortest path problem with interval data, Tech. rep., Bilkent Univ., Dpt of Industrial Engineering (2001).

[13] A. Kasperski, P. Zielinski, Minimizing maximal regret in the linear assignment problems with interval costs, Tech. Rep. 007, Instytut Matematyki Wroclaw (2004).

[14] A. Kasperski, P. Zielinski, An approximation algorithm for interval data minmax regret combinatorial optimization problems, Inf. Proc. L. 97 (2006) 177–180.

[15] A. Kasperski, P. Zielinski, The robust shortest path problem in series-parallel multidigraphs with interval data, Operations Research Letters 34 (2006) 69–76.

[16] D. König, Über graphen und iher anwendung auf determinanten theorie und mengenlehre, Mathematische Annalen 77 (1916) 453–465.

[17] P. Kouvelis, G. Yu, Robust Discrete Optimization and Its Applications, Kluwer Academic Publishers, 1997.

[18] A. Paz, S. Moran, Non deterministic polynomial optimisation problems and their approximation, Theoretical Computer Science 95 (1981) 251–277.