J Glob Optim (2007) 37:467–480
DOI 10.1007/s10898-006-9074-4

ORIGINAL PAPER

A Branch and Bound algorithm for the minimax regret spanning arborescence

Eduardo Conde

Received: 24 January 2006 / Accepted: 27 July 2006 /
Published online: 6 September 2006
© Springer Science+Business Media B.V. 2006

Abstract The paper considers the problem of finding a spanning arborescence on a directed network whose arc costs are partially known. It is assumed that each arc cost can take on values from a known interval defining a possible economic scenario. In this context, the problem of finding the spanning arborescence which *better* approaches to that of minimum overall cost under each possible scenario is studied. The minimax regret criterion is proposed in order to obtain such a robust solution of the problem. As it is shown, the bounds on the optimal value of the minimax regret optimization problem obtained in a previous paper, can be used here in a Branch and Bound algorithm in order to give an optimal solution. The computational behavior of the algorithm is tested through numerical experiments.

Keywords Spanning arborescences · Robust optimization · Branch and Bound algorithms

1 Introduction

The minimax regret version of the minimum spanning tree on an undirected network with interval edge costs was first studied by Averbakh and Lebedev in [3]. This optimization model, that will be called from now MRST problem (*Minimax Regret Spanning Tree Problem*), considers as objective function to be minimized, the maximum regret for any spanning tree given that its edge costs vary in their respective interval estimates. These authors proved that the MRST problem is NP-hard even if the bounds of

This research has been supported by the Spanish Ministry of Education and Science and FEDER Grant No. **MTM2006-04393** and by the European Alfa Project, "Engineering System for Preparing and Making Decisions Under Multiple Criteria", **II-0321-FA**.

E. Conde
Facultad de Matemáticas, Universidad de Sevilla, Tarfia s/n 41012 Sevilla, Spain
e-mail: educon@us.es

all intervals belong to {0,1}. The NP-Hardness of MRST was independently established by Aron and Hentenryck in [1].

Yaman et al. proposed a mixed integer programming formulation in [9] for the MRST problem with interval estimates of the edge costs. They solved the integer formulation after a preprocessing phase where some edges are eliminated by a dominating condition. Branch and Bound algorithms for MRST have been proposed by Aron and Hentenryck [2] and Montemanni and Gambardella [8]. A Benders decomposition approach for MRST problem has been recently proposed by Montemanni in [7].

In [4] the directed case of the MRST problem is studied. The optimization problem that minimizes the maximum regret of a spanning arborescence (MRSA) is more general than the MRST problem and for this reason it is also NP-hard when interval estimates of the arc costs are considered. In [4] a heuristic algorithm is proposed and it is shown how the error associated to the approximate solution obtained by the scheme must be smaller than a given bound. In the present paper, these bounds are employed in order to give an exact algorithm for the MRSA problem.

The first section will introduce the notation used through this paper and also will sum up the main results about the bounding process developed in the paper [4]. After that, a Branch and Bound scheme will be set up in order to guarantee that an optimal solution for this robust optimization problem can be found. Finally we study the computational behavior of the resulting procedure through a numerical experiment.

2 Notation and preliminary results

In our optimization problem, the set of feasible scenarios, S, is identified by the Cartesian product of closed intervals of costs $[\omega_{ij}^-, \omega_{ij}^+]$, one for each arc $(i,j) \in E$ of a given directed network $D = (N, E)$. In order to save notation, we will denote by s, a particular scenario, that is, $\omega_{ij}^s \in [\omega_{ij}^-, \omega_{ij}^+]$ for each arc.

An arborescence rooted in a given node of N is a subgraph of D with $n-1$ arcs ($|N| = n$), so that, every node in N other than its root is connected to the root by a directed path of arcs in the subgraph. Let $\mathcal{A}$ be the set of spanning arborescences in D with the node 1 as their roots. For a fixed scenario s, the cost of an arborescence $X \in \mathcal{A}$ is given by

$$F^s(X) = \sum_{(i,j)\in X} \omega_{ij}^s.$$

By F^s we denote the optimal objective value of the Minimum Spanning Arborescence (MSA) problem defined by the scenario s, that is,

$$F^s = \min_{X\in\mathcal{A}} F^s(X) \quad \mathrm{MSA}(s).$$

As it is well-known the problem MSA(s) can be solved in polinomial time by Edmonds' algorithm [5].

The risk that the decision maker must support due to the choice of a specific spanning arborescence $X \in \mathcal{A}$ under the scenario s is given by $R^s(X) = F^s(X) - F^s$, this risk function is also known in the literature as *regret* value. The maximum regret over the set of scenarios $R(X) = \max\{R^s(X)\colon\ s \in S\}$ is the evaluation cost of $X \in \mathcal{A}$ under our decision criterion. Hence, our goal is to find a spanning arborescence that minimizes the maximum risk or regret under the set of scenarios (Minimax Regret Spanning Arborescence)

$$R^* = \min R(X) \quad \text{MRSA}$$
$$\text{st}: \; X \in \mathcal{A}.$$

As it was said in the introduction, the MRSA problem is NP-hard [3] therefore, the algorithms that can be proposed to seek optimal solutions must be computationally expensive. The framework of the Branch and Bound methods help us in the task of developing numerical algorithms to solve these type of problems. In what follows, we deal with the application of this technique to the MRSA problem using the following properties that can be found in [4].

Property 2.1 *If the network D is acyclic then, the* MRSA *problem is equivalent to a MSA problem with the arc costs $\overline{\omega}_{ij}$ given by*

$$\overline{\omega}_{ij} = \omega^+_{ij} - \min\{\omega^+_{ij}, \omega^-_{kj} : (k,j) \in E \setminus (i,j)\} \quad \forall (i,j) \in E.$$

For general, non necessarily acyclic, networks solving the MSA problem with arc costs $\overline{\omega}$, given in the property 2.1, can only guarantee an upper bound on the optimal objective of the MRSA problem.

Property 2.2 *Let β be the optimal objective value of the MSA problem on D with the set of arc costs given by $\overline{\omega}$ then, $\beta \geq R^*$, that is, β is an upper bound on the optimal value of the* MRSA *problem.*

In order to compute upper bounds on R^* one can also evaluate the objective function $R(X)$ of the MRSA problem over a set of promising solutions. The task of evaluating the maximum regret for a given spanning arborescence is, by itself, an optimization problem. The following definition is useful in order to have such an evaluation.

Definition 2.1 Let $X \in \mathcal{A}$ be an arbitrary spanning arborescence then, the worst case scenario for X is given by

$$\omega^{s^0(X)}_{ij} = \begin{cases} \omega^+_{ij} & \text{if } (i,j) \in X \\ \omega^-_{ij} & \text{if} (i,j) \notin X. \end{cases}$$

The next property guarantees that evaluating the maximum regret of a given solution is equivalent to solve the MSA problem defined by the corresponding worst scenario (MSA($s^0(X)$)).

Property 2.3 *Let $X \in \mathcal{A}$ be an arbitrary spanning arborescence then, it is verified*

$$R(X) = F^{s^0(X)}(X) - F^{s^0(X)}.$$

In order to have lower bounds on R^*, the optimal objective value of the MRSA problem, we can make use of the result given in Property 2.1, in fact, the determination of optimal solutions for a MSA problem in the acyclic case is extremely easy, just take the cheapest arc entering in each node other than node 1. With this idea in mind, it was defined in [4] the following arc costs.

Definition 2.2 Let $D_a = (N, E_a)$ be an acyclic spanning subgraph contained in D then, we define the following set of costs associated to the scenario set S,

$$\overline{\omega}_{ij}^{D_a} = \begin{cases} \omega_{ij}^{+} - \min\{\omega_{kj}^{-} : (k,j) \in E_a\} & \text{if } (i,j) \notin E_a \\ \omega_{ij}^{+} - \min\{\omega_{ij}^{+}, \omega_{kj}^{-} : (k,j) \in E_a \backslash (i,j)\} & \text{if } (i,j) \in E_a. \end{cases}$$

We denote by $\mathcal{A}(D_a)$ the set of spanning arborescences in D_a with the node 1 as their roots.

Remark 2.1 By a spanning subgraph it is meant a subgraph of D so that it contains at least one spanning arborescence of D rooted in 1, that is, $\mathcal{A}(D_a) \neq \emptyset$. This ensures, in particular, the finiteness of the coefficients $\overline{\omega}_{ij}^{D_a}$ given that, for each node j in $N \setminus \{1\}$, there exists at least one arc (k,j) in E_a entering in it, which implies the finiteness of $\min\{\omega_{kj}^{-} : (k,j) \in E_a\}$ for every $(i,j) \notin E_a$.

A method of finding lower bounds on R^* is given by the next result.

Property 2.4 *Let $D_a = (N, E_a)$ be an acyclic spanning subgraph contained in D and $\alpha(D_a)$ the optimal objective value of the MSA problem on D with arc costs given by $\overline{\omega}_{ij}^{D_a}$ then, $\alpha(D_a) \leq R^*$, that is, $\alpha(D_a)$ is a lower bound on the optimal objective of the* MRSA *problem.*

In [4] these bounds, in conjunction with some other properties, were used in the developing of a heuristic algorithm, now they are used on a Branch and Bound algorithm.

3 Branch and Bound Process

The properties given in the last section allow us to determine upper and lower bounds on the optimal objective value of the MRSA problem. Moreover, if we design a search procedure in which the original MRSA problem is split in a finite number of MRSA subproblems, we can use these bounds for pruning subproblems and, as we will see in this section, for eliminating infeasible subproblems.

We will use a typical splitting rule given by fixing a subset of arcs that must be contained in any spanning arborescence of the subproblem (binding arcs) and a subset of arcs that are forbidden for the feasible spanning arborescences.

Let F be the set of forbidden arcs and B, the set of binding (obligatory) arcs, the subproblems that will be used in the Branch and Bound process have the form

$$\begin{aligned} \min\ & R(X) \\ \text{st}:\ & X \in \mathcal{A} \qquad\qquad \text{MRSA } (F, B) \\ & B \subseteq X \subseteq E \backslash F. \end{aligned}$$

In order to make sense of the optimization subproblem MRSA (F, B) we will assume from now on, the following

Hypothesis 3.1 *The sets F and B have empty intersection, no arc in B has node 1 as its end vertex and there are neither two arcs in B with common end vertex nor vertices in $N \backslash \{1\}$ with zero inner degree in $E \backslash F$.*

It is obvious that, if Hypothesis 3.1 is not verified then, the MRSA subproblem associated to B and F has an empty feasible set. Moreover, this hypothesis can be easily tested and in case of failure of any of the conditions it imposes, the corresponding subproblem can be pruned out from the search tree in the Branch and Bound process.

We define $R_{FB}(X)$ as the maximum regret for the spanning arborescence $X \in \mathcal{A}$ respect to the set of interval costs

$$\begin{array}{llllll} \omega_{ij}^{+}(F,B) = +\infty & \text{if } (i,j) \in F & \text{and} & \omega_{ij}^{+}(F,B) = \omega_{ij}^{+} & \text{if } (i,j) \notin F, \\ \omega_{ij}^{-}(F,B) = -\infty & \text{if } (i,j) \in B & \text{and} & \omega_{ij}^{-}(F,B) = \omega_{ij}^{-} & \text{if } (i,j) \notin B. \end{array} \tag{1}$$

Property 3.1 *If there exists at least one spanning arborescence $Y \in \mathcal{A}$ such that $B \subseteq Y \subseteq E \setminus F$ then, for every $X \in \mathcal{A}$,*

$$R_{FB}(X) < +\infty \Leftrightarrow B \subseteq X \subseteq E \backslash F.$$

Proof Let $X \in \mathcal{A}$ such that there exists an arc $(i,j) \in B \backslash X$ then, $\omega_{ij}^{s^0_{FB}(X)} = \omega_{ij}^{-}(F,B) = -\infty$ which implies that $F^{s^0_{FB}(X)} = -\infty$, in fact, by definition

$$F^{s^0_{FB}(X)} \leq F^{s^0_{FB}(X)}(Y) = -\infty,$$

where $Y \in \mathcal{A}$ and $B \subseteq Y \subseteq E \setminus F$ by hypothesis.

Hence, using Property 2.3 and the definition of $\omega^{\pm}(F,B)$ one has that

$$R_{FB}(X) = F^{s^0_{FB}(X)}(X) - F^{s^0_{FB}(X)} = +\infty.$$

Therefore, given $X \in \mathcal{A}$ with $R_{FB}(X) < +\infty$ it is clear that $B \subseteq X$ which implies that every component of the vector $\omega^{s^0_{FB}(X)}$ must be finite or $+\infty$, never $-\infty$. From this fact, one has that, in particular, $F^{s^0_{FB}(X)} \leq F^{s^0_{FB}(X)}(Y) < +\infty$, that is, $R_{FB}(X) < +\infty$ implies $F^{s^0_{FB}(X)}(X) < +\infty$ and then $X \subset E \setminus F$.

The converse is trivial, from the fact that, if $B \subseteq X \subseteq E \setminus F$ then every component of the vector $\omega^{s^0_{FB}(X)}$ must be finite. □

Corollary 3.1 *If there exists at least one spanning arborescence $Y \in \mathcal{A}$ such that $B \subseteq Y \subseteq E \setminus F$ then, for every $X \in \mathcal{A}$,*

$$R_{FB}(X) < +\infty \Leftrightarrow R_{FB}(X) = R(X).$$

Proof It follows from the proof of Property 3.1 and the definition of $\omega^{\pm}(F,B)$. □

3.1 Upper bounding the subproblems

Now, we will upper bound the optimal objective value of each subproblem MRSA (F,B). First of all, we will state a result that can be used as a feasibility test for the subproblems MRSA (F,B).

Property 3.2 *Let β^0_{FB} be the optimal objective value for the MSA problem*

$$\beta^0_{FB} = \min_{X \in \mathcal{A}} \sum_{(k,p) \in X} \overline{\omega}_{kp}(F,B),$$

where, for all $(k,p) \in E$, we define

$$\overline{\omega}_{kp}(F,B) = \omega^{+}_{kp}(F,B) - \min\{\omega^{+}_{kp}(F,B), \omega^{-}_{jp}(F,B) : (j,p) \in E \setminus (k,p)\}.$$

Then, $\beta^0_{FB} < +\infty$ if and only if there exists an arborescence $Y \in \mathcal{A}$, such that $B \subseteq Y \subseteq E\backslash F$.

Proof Let us suppose that $\beta^0_{FB} < +\infty$ then, there must exist, at least an arborescence $Y \in \mathcal{A}$ such that

$$\sum_{(k,p)\in Y} \overline{\omega}_{kp}(F,B) < +\infty.$$

By its definition, $\overline{\omega}_{kp}(F,B) \geq 0$ for all $(k,p) \in E$, therefore

$$\overline{\omega}_{kp}(F,B) < +\infty \quad \forall (k,p) \in Y. \tag{2}$$

Now, let $(k,p) \in F$, following the definition of $\overline{\omega}_{kp}(F,B)$ one has that

$$\overline{\omega}_{kp}(F,B) = +\infty - \min\left\{+\infty, \omega^-_{jp}(F,B) : (j,p) \in E\backslash(k,p)\right\} = +\infty$$

where the last equality follows from the fact that there must exist at least one arc (j,p) in $E \backslash F$ (Hypothesis 3.1) which makes the above minimum to be finite or $-\infty$. Hence, $Y \subseteq E\backslash F$ according with (2).

In order to prove that $B \subseteq Y$, let us assume the opposite, that is, let $(j,p) \in B\backslash Y$. Since Y is an arborescence, there must exist an arc in Y with the vertex p as its endpoint, let (k,p) be such an arc. By its definition it follows,

$$\overline{\omega}_{kp}(F,B) = \omega^+_{kp} - \min\left\{\omega^+_{kp}, \omega^-_{ip}(F,B) : (i,p) \in E \backslash (k,p)\right\}.$$

In particular, one has that, since $\omega^-_{jp}(F,B) = -\infty$ and $\omega^+_{kp} < +\infty$ then, $\overline{\omega}_{kp}(F,B) = +\infty$ which leads to a logical contradiction with (2). In conclusion, the set of arcs B must be contained in the set of arcs defining the arborescence Y.

This concludes one of the implications that we must show, to prove the converse, let Y be an arborescence verifying the constraints imposed by the set of arcs F and B, that is, $B \subseteq Y \subseteq E\backslash F$. Given $(k,p) \in Y$ we may have one of the following situations. The first one is that (k,p) belongs to the set of binding arcs B. In that case,

$$\overline{\omega}_{kp}(F,B) = \omega^+_{kp} - \min\left\{\omega^+_{kp}, \omega^-_{jp} : (j,p) \in E\backslash(k,p)\right\},$$

where ω^+_{kp} is finite according with its definition and $\omega^-_{jp}(F,B) = \omega^-_{jp}$ for each $(j,p) \in E\backslash(k,p)$, it cannot be equal to $-\infty$ by Hypothesis 3.1, otherwise it must imply that B contains at least two arcs with the same endpoint. Hence, $\overline{\omega}_{kp}(F,B)$ must be finite.

The second case that one may have is that (k,p) belongs to $Y \backslash B$ then, since $\omega^+_{kp} < +\infty$, because $(k,p) \notin F$, and $\omega^-_{jp} > -\infty$ for all $(j,p) \in E\backslash(k,p)$ (otherwise, B would not be included in Y), one finally has that $\overline{\omega}_{kp}(F,B)$ must be finite.

Hence, in both cases we have shown the finiteness of the coefficients given by $\overline{\omega}_{kp}(F,B)$, which implies that

$$\sum_{(k,p)\in Y} \overline{\omega}_{kp}(F,B) < +\infty.$$

Finally, taking into account that $\overline{\omega}_{kp}(F,B)$'s are all nonnegative coefficients by its definition, we have that β^0_{FB} exists and is finite. □

Using the previous results one can finally have an upper bound on the minimax regret over the set of feasible spanning arborescences according with the constraints given by the sets of forbidden (F) and binding (B) arcs. This bound is given by β^0_{FB} as it is formally stated in the following.

Theorem 3.1 *Given a couple (F, B) of forbidden and binding arcs of E, β^0_{FB} is an upper bound for the optimal objective of* MRSA (F, B). *Moreover, $\beta^0_{FB} < +\infty$ if and only if the optimization problem* MRSA (F, B) *is feasible and, in this case, it is equivalent to the unrestricted* MRSA *problem with interval costs given by the bounds $\omega^{\pm}_{ij}(F, B)$ defined in (1).*

Proof First of all, note that the optimization subproblem MRSA (F, B) has a non-empty feasible set if and only if $\beta^0_{FB} < +\infty$ (Property 3.2). Hence, even in the case of an infeasible MRSA (F, B) problem, the bound given by β^0_{FB} is valid.

Let us now suppose that β^0_{FB} is finite. Then, there exist feasible solutions for the subproblem MRSA (F, B) (Property 3.2) and according with the property 3.1 and its corollary,

$$R_{FB}(X) = R(X) \quad \forall X \in \mathcal{A} : B \subseteq X \subseteq E \backslash F$$

whereas $R_{FB}(X) = +\infty$ on any $X \in \mathcal{A}$ which is infeasible for MRSA (F, B). Therefore, one has that

$$\begin{array}{lll} \min\ R_{FB}(X) = \min & R(X) \\ \text{st}:\ X \in \mathcal{A} \qquad \text{st}: & X \in \mathcal{A} \\ & B \subseteq X \subseteq E \backslash F. \end{array}$$

Following Property 2.2 it is known that β^0_{FB} is an upper bound on the optimal objective value of the left hand side optimization problem then, our result has been shown. □

Corollary 3.2 *Let $X_{FB} \in \mathcal{A}$ such that $\beta^0_{FB} = \sum_{(k,p)\in X_{FB}} \overline{\omega}_{kp}(F, B) < +\infty$, then one can refine the upper bound on the minimum of* MRSA *(F, B) by taking as upper bound*

$$\beta_{FB} = F^{s^0_{FB}(X_{FB})}(X_{FB}) - F^{s^0_{FB}(X_{FB})} = R(X_{FB}).$$

Proof First, one has by Property 4.1 of [4] that

$$R_{FB}(X_{FB}) \leq \sum_{(k,p)\in X_{FB}} \overline{\omega}_{kp}(F, B) = \beta^0_{FB}, \tag{3}$$

and $\beta^0_{FB} < +\infty$, therefore $R_{FB}(X_{FB}) < +\infty$. Hence, by Corollary 3.1 it follows that $R(X_{FB}) = R_{FB}(X_{FB})$. Finally, using Property 2.3 and the inequality (3) one has

$$R^* \leq R(X_{FB}) = F^{s^0_{FB}(X_{FB})}(X_{FB}) - F^{s^0_{FB}(X_{FB})} \leq \beta^0_{FB}.$$

□

3.2 Lower bounding the subproblems

In order to lower bound the subproblem MRSA (F, B) let us define the set of scenarios induced by the binding and forbidden arcs, $S(F, B)$, that is the Cartesian product of the intervals $[\omega^-_{ij}(F, B), \omega^+_{ij}(F, B)]$ given in (1) and let us consider the regret function

$$R^{D_a}_{FB}(X) = \max_{s\in S(F,B)} \left\{ F^s(X) - \min_{Y\in\mathcal{A}(D_a)} F^s(Y) \right\} \tag{4}$$

where D_a and $\mathcal{A}(D_a)$ were given in Definition 2.2. Finally, let us take the coefficients $\overline{\omega}_{ij}^{D_a}(F,B)$ given by Definition 2.2 associated, in this case, to the scenario set $S(F,B)$.

We will assume that MRSA (F,B) has feasible solutions which can be tested by checking if β_{FB}^0 is finite or not, as it was established in Property 3.2. Then, one has the following

Theorem 3.2 *Given $D_a = (N, E_a)$, an acyclic spanning subgraph contained in D, the optimal objective value $\alpha_{FB}(D_a)$ of the MSA problem on D with arc costs given by $\overline{\omega}_{ij}^{D_a}(F,B)$ is a lower bound on the optimal objective of the* MRSA *(F,B) problem.*

Proof According with the property 3.1 and its corollary,

$$R_{FB}(X) = R(X) \quad \forall X \in \mathcal{A} : B \subseteq X \subseteq E \backslash F$$

whereas $R_{FB}(X) = +\infty$ on any $X \in \mathcal{A}$ which is infeasible for MRSA (F,B).

On the other hand, by definition of $R_{FB}^{D_a}(X)$, (4), one has that

$$R_{FB}^{D_a}(X) \leq R_{FB}(X) \quad \forall X \in \mathcal{A},$$

since $R_{FB}^{D_a}(X)$ represents the maximum regret for X respect to a subset of spanning arborescences of D, those contained in $\mathcal{A}(D_a)$. Hence, it is verified

$$\begin{array}{llll} \min & R_{FB}^{D_a}(X) & \leq \min & R(X) \\ \text{st}: & X \in \mathcal{A} & \text{st}: & X \in \mathcal{A} \\ & B \subseteq X \subseteq E \setminus F & & B \subseteq X \subseteq E \setminus F. \end{array} \tag{5}$$

Taking into account that, by definition,

$$\begin{array}{rl} \alpha_{FB}(D_a) = \min & \sum_{(k,p)\in X} \overline{\omega}_{kp}^{D_a}(F,B) \\ \text{st}: & X \in \mathcal{A}, \end{array}$$

which implies by [4]

$$\begin{array}{rl} \alpha_{FB}(D_a) = \min & R_{FB}^{D_a}(X) \\ \text{st}: & X \in \mathcal{A}, \end{array}$$

from (5) it follows that $\alpha_{FB}(D_a)$ is a lower bound for the optimal value of MRSA (F,B). □

4 The algorithm

As it is well-known the strategy of a Branch and Bound algorithm consists in solving a sequence of subproblems associated to a partition of the feasible set of the original optimization problem. In the process, the bounds on the optimal objective are adjusted. Using these bounds, some subproblems can be eventually pruned in the progress of the algorithm due to their best feasible solutions are worse than a given solution which has been obtained in previous iterations.

We will identify the subproblems solved in the iterations of the algorithm by a pair of disjoint subset of arcs, F and B and some other information about the known bounds on their optimal objectives. In order to describe the algorithm, a list

$\mathcal{L}$, whose elements are the subproblems that must be solved in the next iterations, will be used. Each element of this list is identified by $(X_{FB}, F, B, \alpha_{FB}, \beta_{FB}) \in \mathcal{L}$, where α_{FB} is a lower bound on the minimum of MRSA (F, B), β_{FB} is an upper bound and X_{FB} is the best feasible solution generated in previous iterations for the subproblem MRSA (F, B).

In order to initiate the process, one can take $F = B = \emptyset$, β_{FB}, the upper bound given by the corollary 3.2. Let X_{FB} be the optimal solution of the MSA problem defining β^0_{FB} (Property 3.2) and Y_{FB} so that

$$F^{s^0_{FB}(X_{FB})} = F^{s^0_{FB}(X_{FB})}(Y_{FB}).$$

An acyclic spanning subgraph D_a can be obtained from the spanning arborescence Y_{FB} by adding to it arcs from E, in such a way that its acyclicity is preserved. The lower bound $\alpha_{FB} = \alpha_{FB}(D_a)$ obtained from D_a can be directly used to initialize the list $\mathcal{L}$ or can be improved by applying sequentially this bounding process. That is, taking now $\overline{X} \in \mathcal{A}$, the optimal arborescence that defines $\alpha_{FB}(D_a)$ according to the theorem 3.2, one can repeat the above bounding process replacing the initial arborescence X_{FB} by $\overline{X}$, calculating Y_{FB} an optimal solution of the MSA $(s^0_{FB}(\overline{X}))$, adding to this arborescence arcs from E to obtain a new subgraph D_a and so on. In the scheme of the algorithm stated later, this process will be repeated for M times in the step 1:2, where M is a parameter that must be established by the user.

Once the list $\mathcal{L}$ has been initialized, each iteration of the algorithm will select an element of the list, that with smallest upper bound β_{FB}, and an arbitrary arc $(i, j) \in X_{FB} \backslash B$. Obviously, if the choice of such an arc is not possible, the corresponding subproblem can be deleted from $\mathcal{L}$ because the only feasible solution for this subproblem is given by X_{FB}, that is, this subproblem has been exactly solved and it must not be considered in the next iterations of the algorithm.

The branching process will divide the feasible set of the selected subproblem into two parts by including the arc (i, j) in the set of the binding arcs or in the set of the forbidden ones. Hence, the selected element of $\mathcal{L}$ will be replaced by two elements, $(X_{F_1B_1}, F_1, B_1, \alpha_{F_1B_1}, \beta_{F_1B_1})$, $(X_{F_2B_2}, F_2, B_2, \alpha_{F_2B_2}, \beta_{F_2B_2})$ where $F_1 = F$, $B_1 = B \cup \{(i, j)\}$, $F_2 = F \cup \{(i, j)\}$ and $B_2 = B$.

The upper bound for the first of these two new subproblems must not be recalculated since the spanning arborescence X_{FB} is feasible and $\beta_{FB} = R_{FB}(X_{FB}) = R(X_{FB})$ according with the corollary 3.1. Therefore, one can take $X_{F_1B_1} = X_{FB}$ and also maintain both, the upper and lower bounds $\beta_{F_1B_1} = \beta_{FB}$ and $\alpha_{F_1B_1} = \alpha_{FB}$.

For the second subproblem introduced into $\mathcal{L}$ one should actualize the upper bound since the bound β_{FB} does not longer apply. Once the new bound $\beta_{F_2B_2}$ has been calculated, we will actualize the lower bound for this subproblem from the spanning arborescence $Y_{F_2B_2}$ as it was already explained above.

The resulting algorithm will be stated below.

Algorithm 4.1

0: Input: $D = (N, E)$, $\omega^{\pm}_{ij}$ for each $(i, j) \in E$.

1: Initialization: Take $F = B = \emptyset$, $\alpha_{FB} = 0$ and $\mathcal{L} = \emptyset$.

1:1: Initialization of the lower and upper bounds: Let X_{FB} and β_{FB} be the spanning arborescence and the upper bound given by Corollary 3.2. Take $\overline{X} = X_{FB}$.

1:2: Improving the lower and upper bounds: Repeat this step for M times. Let $Y_{FB} \in \mathcal{A}$ be a spanning arborescence so that,

$$F^{s^0_{FB}(\overline{X})} = F^{s^0_{FB}(\overline{X})}(Y_{FB}).$$

Following the corollary 3.2 $R(\overline{X}) = F^{s^0_{FB}(\overline{X})}(\overline{X}) - F^{s^0_{FB}(\overline{X})}$ then, if $R(\overline{X}) < \beta_{FB}$ actualize $\beta_{FB} = R(\overline{X})$ and $X_{FB} = \overline{X}$. Add arcs to Y_{FB} maintaining its acyclicity. Let D_a be the resulting acyclic network.
Take

$$\begin{array}{rl} \alpha_{FB}(D_a) = \min & \sum_{(k,p)\in X} \overline{\omega}^{D_a}_{kp}(F,B) \\ \text{st}: & X \in \mathcal{A}, \end{array}$$

and let $\overline{X}$ be the arborescence where the optimal value $\alpha_{FB}(D_a)$ is reached. If $\alpha_{FB} < \alpha_{FB}(D_a)$ actualize $\alpha_{FB} = \alpha_{FB}(D_a)$.

1:3 If $\alpha_{FB} = \beta_{FB}$, STOP: X_{FB} is an optimal solution. In other case, actualize $\mathcal{L} := \mathcal{L} \cup \{(X_{FB}, F, B, \alpha_{FB}, \beta_{FB})\}$, $\beta = \beta_{FB}$ and $X^* = X_{FB}$.

2: Iteration: While $\mathcal{L} \neq \emptyset$:

2:1: Take $(X_{FB}, F, B, \alpha_{FB}, \beta_{FB}) \in \mathcal{L}$ so that β_{FB} is a minimum.

2:2: Select an arc $(i,j) \in X_{FB} \setminus B$, so that, there is at least an arc in $E \setminus (F \cup \{(i,j)\})$ entering in j.

- If no choice is possible, the subproblem MRSA (F, B) has as an optimal solution the spanning arborescence X_{FB}. Delete this subproblem from $\mathcal{L}$ and go to Iteration.
- Otherwise, eliminate this subproblem from $\mathcal{L}$ and continue.

2:3: Branching: Add to the list $\mathcal{L}$ the elements $(X_{F_1B_1}, F_1, B_1, \alpha_{F_1B_1}, \beta_{F_1B_1})$ and $(X_{F_2B_2}, F_2, B_2, \alpha_{F_2B_2}, \beta_{F_2B_2})$ where $F_1 = F$, $B_1 = B \cup \{(i,j)\}$, $F_2 = F \cup \{(i,j)\}$ and $B_2 = B$. Here $X_{F_1B_1} = X_{FB}$, $\alpha_{F_1B_1} = \alpha_{FB}$, $\beta_{F_1B_1} = \beta_{FB}$ and $X_{F_2B_2}$, $\alpha_{F_2B_2}$, $\beta_{F_2B_2}$ are the spanning arborescence and the bounds obtained by applying 1:1 and 1:2 with $M = 1$ (by initializing $\alpha_{F_2B_2} = \alpha_{FB}$).

2:4: If $\beta_{F_2B_2} < \beta$ actualize $\beta = \beta_{F_2B_2}$ and $X^* = X_{F_2B_2}$. Delete every element of the list $\mathcal{L}$ with a lower bound greater than β.

3: Output: The spanning arborescence X^* is an optimal solution for the problem MRSA .

Theorem 4.1 *Algorithm 4.1 exactly solves the problem* MRSA *after a finite number of iterations.*

Proof The proof follows from the finiteness of the branching process given that, in each iteration, the binding arc set or the forbidden arc set, increases in size for the new generated subproblems. □

5 Numerical experiments

In the following numerical experiment, random networks of number of nodes $n = 10, 20, 30, 40$ and 50 were solved using Algorithm 4.1. Given an ordered pair of nodes of one of these networks, i, j the corresponding arc (i, j) was generated with probability $p = 0'5$.

The necessary uncertain interval, $[\omega^-_{ij}, \omega^+_{ij}]$, for each generated arc, was randomly obtained according with a beforehand specified level of uncertainty δ. Specifically, for each arc (i, j) the lower bound of the uncertain interval, ω^-_{ij}, was randomly generated from a uniform distribution on the interval $[0, 100]$. Afterwards, the upper bound for this interval was generated according with the equation

$$\omega^+_{ij} = \omega^-_{ij}(1 + U\delta),$$

where U follows an independent uniform distribution on $[0, 1]$. Hence, the factor δ represents, in some way, the admissible level of uncertainty in the cost coefficients of the problem.

The results of Table 1 were obtained for four degrees of uncertainty $\delta = 1, 5, 10$ and 25% on a personal computer with Intel ® Pentium ® M processor, 1.60GHz with $0'99$ GB RAM and the code was written as a Matlab ® v.6.5 program. For each combination of δ and n, fifty instances of the problem were generated according to the process described before. In the step 1:2 of the algorithm 4.1 it was taken $M = 40$.

Analysing Table 1, one can see how the CPU time and number of branching operations, or NBO (step 2:3 of the algorithm 4.1), increase with the number of nodes. This is the type of behavior one may expect for the Branch and Bound process but, perhaps, what is more surprising is the fast increasing in computational times when the uncertain degree, δ, grows. This increase in the execution times turned out to be dramatic for the largest values of n for which the computational capabilities of the personal computer used in the experiment were widely exceeded. Specifically, for $n = 50$ and $\delta = 25\%$, the entry of Table 1 is empty, indicating that the first ten instances that were introduced as inputs of the algorithm 4.1, gave execution times greater than 2500 seconds of CPU, these processes were aborted before that their optimal values were reached.

In order to compare the results obtained with the algorithm 4.1, we will give, in what follows, a *Mixed Integer Programming* (MIP) formulation that will be solved later by means of a standard MIP solver. This MIP formulation has been obtained using the same ideas applied by Yaman et al. in [9]. These authors have, in turn, used a previous work of Magnanti and Wolsey, [6], where the minimum spanning tree problem was modeled as a version of a network flow problem. In this new problem, a flow of $n - 1$ units is sent from the root to the other nodes using only $n - 1$ edges. The minimum cost flow under such constraints, determines a minimum spanning tree

Table 1 Averages ($\bar{t}_{CPU}$), Standard Deviations (σ_{CPU}) of the CPU times, in seconds, and Averages of the Number of Branching Operations (NBO) spent by Algorithm 4.1 in the numerical experiment

		n				
$p = 0'5$		10	20	30	40	50
$\delta =$ 1%	$\bar{t}_{CPU}$	$0'08$	$0'88$	$4'11$	$15'29$	$40'60$
	σ_{CPU}	$0'03$	$0'09$	$0'22$	$5'11$	$7'91$
	NBO	$9'95$	$19'50$	$30'00$	$40'05$	$51'3$
$\delta =$ 5%	$\bar{t}_{CPU}$	$0'13$	$1'03$	$5'78$	$23'90$	$142'62$
	σ_{CPU}	$0'11$	$0'51$	$4'01$	$16'49$	$279'81$
	NBO	$9'45$	$20'50$	$36'76$	$60'90$	$108'60$
$\delta =$ 10%	$\bar{t}_{CPU}$	$0'13$	$3'07$	$8'37$	$53'12$	$250'32$
	σ_{CPU}	$0'09$	$5'80$	$4'52$	$62'32$	$276'96$
	NBO	$9'85$	$53'71$	$46'70$	$170'9$	$309'16$
$\delta =$ 25%	$\bar{t}_{CPU}$	$0'30$	$6'78$	$20'05$	$213'21$	–
	σ_{CPU}	$0'36$	$18'81$	$23'68$	$299'37$	–
	NBO	$11'15$	$77'10$	$140'40$	$683'20$	–

for the network. Using this idea, for a given scenario $s \in S$, the MSA problem can be formulated as the mixed integer linear program

$$
\begin{array}{lll}
\min & \sum_{(i,j)\in E} \omega_{ij}^s x_{ij} & \\
\text{st}: & & \\
& \sum_{(j,k)\in E} f_{jk} - \sum_{(i,j)\in E} f_{ij} = \begin{cases} n-1, & \text{if } j=1, \\ -1, & \text{if } j\neq 1, \end{cases} & \\
& f_{ij} \leq (n-1)x_{ij} & \forall (i,j)\in E, \\
& \sum_{(i,j)\in E} x_{ij} = n-1, & \\
& f_{ij} \geq 0 & \forall (i,j)\in E, \\
& x_{ij} \in \{0,1\} & \forall (i,j)\in E.
\end{array}
\tag{6}
$$

Unfortunately, in the formulation (6) the vertices of the feasible polyhedron are not integer in general, and in consequence, the binary variables x_{ij} can not be relaxed. However, following [6] one can have an equivalent linear program by splitting the flow emanating from the root into $n-1$ individual flows that serve exactly one unit of product, to each node. Using this idea, Magnanti and Wolsey proposed in ([6]) a linear program for the minimum spanning tree problem. Rewriting this formulation for our MSA problem one obtains

$$
\begin{array}{lll}
\min & \sum_{(i,j)\in E} \omega_{ij}^s y_{ij} & \\
\text{st}: & & \\
& \sum_{(1,k)\in E} f_{1k}^p - \sum_{(i,1)\in E} f_{i1}^p = 1 & \forall p\neq 1, \\
& \sum_{(j,k)\in E} f_{jk}^p - \sum_{(i,j)\in E} f_{ij}^p = 0 & \forall j\neq 1, p\neq 1, j\neq p, \\
& \sum_{(j,k)\in E} f_{jk}^j - \sum_{(i,j)\in E} f_{ij}^j = -1 & \forall j\neq 1, \\
& f_{ij}^p \leq y_{ij}, & \forall (i,j)\in E, p\neq 1, \\
& \sum_{(i,j)\in E} y_{ij} = n-1, & \\
& f_{ij}^p \geq 0 & \forall (i,j)\in E, p\neq 1, \\
& y_{ij} \geq 0 & \forall (i,j)\in E.
\end{array}
\tag{7}
$$

Finally, we will use the same technique as Yaman et al. in [9] in order to have a MIP formulation for our MRSA problem. Taking the dual problem of (7) one has that

$$
\begin{array}{lll}
F^{s^0(X)} = & \max \sum_{p\neq 1} (\gamma_p^p - \gamma_1^p) + (n-1)\mu & \\
& \delta_{ij}^p \geq \gamma_j^p - \gamma_i^p & \forall (i,j)\in E, \quad \forall p\neq 1, \\
& \sum_{p\neq 1} \delta_{ij}^p + \mu \leq \omega_{ij}^- + (\omega_{ij}^+ - \omega_{ij}^-)x_{ij}, & \forall (i,j)\in E, \\
& \gamma_i^p \geq 0 & \forall i, p\neq 1, \\
& \delta_{ij}^p \geq 0 & \forall (i,j)\in E, \quad \forall p\neq 1,
\end{array}
\tag{8}
$$

for all arborescence $X \in \mathcal{A}$ which is identified by its binary variables $x_{ij} \in \{0,1\}$, $(i,j) \in E$. Note that, in the dual formulation (8), we have written

$$
\omega_{ij}^{s^0(X)} = \omega_{ij}^- + (\omega_{ij}^+ - \omega_{ij}^-)x_{ij}
$$

according with Definition 2.1.

Hence, using Property 2.3 and the expression of $F^{s^0(X)}$ given by (8) we can modify the formulation (6), in order to give a mixed integer linear program for MRSA, as follows

$$
\begin{array}{llll}
\min & \sum_{(i,j)\in E}\omega_{ij}^{+}x_{ij} - \sum_{p\neq 1}(\gamma_p^p - \gamma_1^p) - (n-1)\mu & & \\
\text{st}: & & & \\
& \sum_{(j,k)\in E} f_{jk} - \sum_{(i,j)\in E} f_{ij} = \begin{cases} n-1, & \text{if } j=1,\\ -1, & \text{if } j\neq 1,\end{cases} & & \\
& \delta_{ij}^{p} \geq \gamma_j^p - \gamma_i^p, & \forall (i,j)\in E, & \forall p\neq 1,\\
& \sum_{p\neq 1}\delta_{ij}^{p} + \mu \leq \omega_{ij}^{-} + (\omega_{ij}^{+} - \omega_{ij}^{-})x_{ij} & \forall (i,j)\in E, & \\
& f_{ij} \leq (n-1)x_{ij}, & \forall (i,j)\in E, & \\
& \sum_{(i,j)\in E} x_{ij} = n-1, & & \\
& f_{ij}\geq 0, & \forall (i,j)\in E, & \\
& x_{ij}\in\{0,1\}, & \forall (i,j)\in E. & \\
& \gamma_i^p \geq 0, & \forall i,p\neq 1, & \\
& \delta_{ij}^{p}\geq 0 & \forall (i,j)\in E, & \forall p\neq 1
\end{array}
\qquad \text{(MIP)}
$$

Formulation (*MIP*) was solved for the same instances of the computational experiment developed to obtain Table 1 and in the same computer where such an experiment was carried out. The ILOG CPLEX ® 8.1 MIP solver was used to solve our (*MIP*) formulation. The results are shown in Table 2 where, in addition to the averages and deviations of the CPU times, it has been included the average of the number of iterations and nodes given by the MIP solver in the final report of the optimal solution for each instance.

The empty entries of Table 2 have the same meaning as in Table 1. Taking into account only the nonempty entries of both tables, one can see how the CPU times

Table 2 Averages ($\bar{t}_{\text{CPU}}$), Standard Deviations (σ_{CPU}) of the CPU times, in seconds, and Averages on the Number of Iterations (Iterations) and Nodes (Nodes) spent by the ILOG CPLEX ® 8.1 MIP solver in the numerical experiment

n						
$p = 0'5$		10	20	30	40	50
$\delta =$ 1%	$\bar{t}_{\text{CPU}}$	$0'05$	$1'03$	$30'31$	$587'32$	–
	σ_{CPU}	$0'03$	$0'59$	$22'15$	$223'45$	–
	Iterations	$247'95$	$2703'05$	$19081'65$	$88987'60$	–
	Nodes	$0'00$	$10'65$	$89'45$	$64'70$	–
$\delta =$ 5%	$\bar{t}_{\text{CPU}}$	$0'06$	$1'29$	$28'79$	$500'24$	–
	σ_{CPU}	$0'02$	$0'47$	$18'26$	$320'69$	–
	Iterations	$277'55$	$3258'95$	$19245'20$	$76890'30$	–
	Nodes	$0'20$	$3'40$	$65'35$	$84'50$	–
$\delta =$ 10%	$\bar{t}_{\text{CPU}}$	$0'07$	$1'58$	$45'69$	$631'70$	–
	σ_{CPU}	$0'04$	$0'47$	$24'96$	$406'40$	–
	Iterations	$312'60$	$4060'65$	$25994'40$	$89040'42$	–
	Nodes	$0'00$	$3'80$	$64'95$	$24'36$	–
$\delta =$ 25%	$\bar{t}_{\text{CPU}}$	$0'07$	$2'07$	$63'49$	$977'49$	–
	σ_{CPU}	$0'03$	$0'83$	$32'83$	$514'49$	–
	Iterations	$333'90$	$5158'65$	$32620'10$	$147694'05$	–
	Nodes	$0'15$	$5'60$	$62'25$	$333'55$	–

shown in Table 1 have been exceeded in a $536'88\%$, in average, by the corresponding CPU times of Table 2. It is also interesting to observe how the dependence between the CPU times spent in solving the MIP formulation and the admisible level of uncertainty, δ, is much less strong than in the case of the Algorithm 4.1.

6 Conclusions

In this paper, it has been proposed a Branch and Bound algorithm for solving the robust (minimax) version of a classical combinatorial problem: The Minimum Spanning Arborescence Problem. The bounds on the optimal value of this minimax problem, obtained in a previous work, have been adapted in this paper in order to develop the bounding process on the subproblems generated by our branching technique.

Recently, some other Branch and Bound algorithms have appeared in the literature [2,8] to solve the optimization problem considered in this paper on undirected networks. This is a specific subproblem of the optimization model presented here; hence, it seems to be unsuitable to compare these algorithms from a computational view point. However, considering the computational results presented in the papers [2,8] using similar numerical experiments to that displayed in the above section, one can conclude that the computational behavior of our algorithm moves in the same range of values. This can be considered a significant advance, taking into account that our problem is more complex than its undirected version.

Acknowledgement The author would like to thank the anonymous referees for their constructive comments.

References

1. Aron, I., Van Hentenryck, P.: On the complexity of the robust spanning tree with interval data. Oper. Res. Lett. **32**(1), 136–140 (2003)
2. Aron, I., Van Hentenryck, P.: A constraint satisfaction approach to the robust spanning tree with interval data. Proceedings of the 18th conference on uncertainty in artificial intelligence (UAI), (August 2002)
3. Averbakh, I., Lebedev, V.: Interval data minmax regret network optimization problems. Discrete Applied Mathematics, **138**, 289–301 (2004)
4. Conde, E., Candia, A.: Minimax regret spanning arborescences under uncertain costs. Eur. J. Operational Res. (in Press) (2006)
5. Edmonds, J.: Optimum branching. J. Res. Nat. Bureau of Standards, **71B** 233–240 (1967)
6. Magnanti, T.L., Wolsey, L.: Optimal trees, in: Network models, handbook in operations research and management science. In: Ball, M.O. et al. (eds.), vol. 7, pp. 503–615. North-Holland Amsterdam (1995)
7. Montemanni, R.: A Benders decomposition approach for the robust spanning tree problem with interval data. Eur. J. Operational Res. **174**(3), 1479–1490 (2006)
8. Montemanni, R., Gambardella, L.M.: A branch and bound algorithm for the robust spanning tree problem with interval data. Eur. J. Operational Res. **161**(3), 771–779 (2005)
9. Yaman, H., Karasan, O.E., Pinar, M.Ç.: The robust spanning tree problem with interval data. Operations Res. Lett. **29**(1), 31–40 (2001)