A Benders decomposition approach for the robust spanning tree problem with interval data

Roberto Montemanni

Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA)

Galleria 2, CH-6928 Manno-Lugano, Switzerland

Tel: +41 91 610 8671   Fax: +41 91 610 8661

E-mail: roberto@idsia.ch

Abstract

The robust spanning tree problem is a variation, motivated by telecommunications applications, of the classic minimum spanning tree problem. In the robust spanning tree problem edge costs lie in an interval instead of having a fixed value.

Interval numbers model uncertainty about the exact cost values. A robust spanning tree is a spanning tree whose total cost minimizes the maximum deviation from the optimal spanning tree over all realizations of the edge costs. This robustness concept is formalized in mathematical terms and is used to drive optimization.

This paper describes a new exact method, based on Benders decomposition, for the robust spanning tree problem with interval data. Computational results highlight the efficiency of the new method, which is shown to be very fast on all the benchmarks considered, and in particular on those that were harder to solve for the methods previously known.

Key words: Combinatorial optimization, robustness, interval data, minimum spanning tree, Benders decomposition.

1 Introduction

This paper presents an exact algorithm, based on Benders decomposition, for a robust version of the minimum spanning tree problem where edge costs lie in intervals instead of having fixed values. Each interval is used to model uncertainty about the real value of the respective cost, which can take any value in the interval, independently from the costs associated with the other edges of the graph.

Adopting the model described above, the classic optimality criterion of the *minimum spanning tree problem* (MST - where a fixed cost is associated with each edge of the graph) does not apply anymore, and the classic polynomial-time algorithms (e.g. Kruskal [16] and Prim [26]) cannot be used. A more complex optimization criterion has then to be adopted. We consider here the *relative robustness criterion*, described in Kouvelis and Yu [14] and applied to many combinatorial optimization problems with interval data in [2–4,9,13,21–25,28–31], although the list is by no means exhaustive.

The study has practical motivations, and in particular there are some applications in the field of telecommunications. Consider a supervisor node in a data network where transmission lines are subject to uncertain delays, that wants to send a control message to all other nodes in the network. The supervisor node generally wants to broadcast the message over a robust spanning tree, in order to have a relatively quick broadcast whatever the situation in the network is (see Bertsekas and Gallagher [6] for a more detailed description of the problem). A second application concerns the design of communication networks where routing delays on edges are uncertain, since they depend on the network traffic. The ideal network guarantees good performance whatever is the real traffic, i.e. a robust spanning tree is desirable (see Kouvelis and Yu [14] for more details).

In the literature there are some other studies related to robust versions of the minimum spanning tree problem. Kozina and Perepelista [15] defined an order relation on the set of feasible solutions and generated a Pareto set. Aron and Van Hentenryck [3] proved that the problem is $\mathcal{NP}$-hard. In Yaman et al. [28] a mixed integer programming formulation and a preprocessing technique are presented. A branch and bound algorithm is presented in Montemanni et al. [23]. Two other branch and bound approaches, similar to that described in [23], have been independently developed by Aron and Van Hentenryck (see [2]).

In Section 2 the robust spanning tree problem with interval data is formally described. In Section 3 a new mixed integer programming formulation for the problem is presented. Section 4 discusses how Benders decomposition can be applied to the robust spanning tree problem with interval data. Section 5 is dedicated to computational results, while conclusions are presented in Section 6.

2 Problem description

The robust spanning tree problem with interval data is defined on a graph $G = \{V, E\}$, where V is a set of vertices and E is a set of edges. An interval $[l_{ij}, u_{ij}]$, with $0 \leq l_{ij} < u_{ij}$, is associated with each edge $\{i, j\} \in E$. Intervals represent ranges of possible costs. An example of interval graph is given in

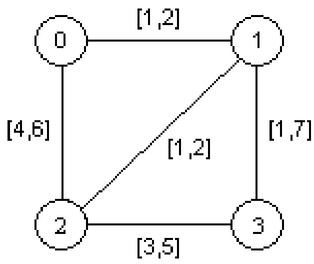

Fig. 1. Example of interval graph.

Figure 1.

The problem is formally described through the following definitions:

Definition 1 *A* scenario *s is a realization of edge costs, i.e. a cost* $c^s_{ij} \in [l_{ij}, u_{ij}]$ *is fixed* $\forall \{i, j\} \in E$.

Definition 2 *The* robust deviation *for a spanning tree t in a scenario s is the difference between the cost of t in s and the cost of the minimum spanning tree in s.*

Definition 3 *A spanning tree t is said to be a* relative robust spanning tree *if it has the smallest (among all spanning trees) maximum (among all possible scenarios) robust deviation.*

A scenario can be seen as a snapshot of the network situation, while a relative robust spanning tree (*robust spanning tree* - *RST* - for short) is a tree which minimizes the maximum deviation from the optimal spanning tree over all realizations of the edge costs.

The following result is at the basis of the method we propose:

Theorem 4 (Yaman et al. [28]) *Given a spanning tree t, a scenario $s(t)$ which makes the robust deviation maximum for t is the one where* $c^{s(t)}_{ij} =$

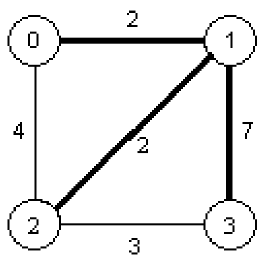

Fig. 2. Scenario induced by spanning tree $t = \{\{0,1\},\{1,2\},\{1,3\}\}$ on the interval graph of Figure 1.

$u_{ij}\ \forall\{i,j\} \in t$ *and* $c_{kh}^{s(t)} = l_{kh}\ \forall\{k,h\} \in E \backslash t.$

In the remainder of this paper we will refer to the scenario $s(t)$ as the scenario *induced* by tree t. We will also refer to the cost of t in $s(t)$ minus the cost in $s(t)$ of a minimum spanning tree of $s(t)$ as the *robustness cost* of t.

A polynomial-time procedure for the evaluation of the robustness cost of a given spanning tree t arises. It simply works by subtracting the cost of the minimum spanning tree in scenario $s(t)$ (see, for example, Prim [26]) from the cost, in the same scenario, of t.

Figure 2 depicts the scenario induced by the spanning tree $t = \{\{0,1\},\{1,2\},\{1,3\}\}$ on the graph of Figure 1. Since the minimum spanning tree in this scenario is $t' = \{\{0,1\},\{1,2\},\{2,3\}\}$, the robustness cost of t is $\underbrace{(2+2+7)}_{\text{cost of } t} - \underbrace{(2+2+3)}_{\text{cost of } t'} = 4.$

3 Mixed integer programming formulation

A first mathematical formulation for the *RST* problem has been presented in Yaman et al. [28]. This formulation exploits Theorem 4 to join together a classic formulation for the MST problem and the dual of another classic MST formulation. Here we use the same mechanism, but we substitute the primal MST formulation with another one with fewer variables (but many more constraints), more suitable - according to some preliminary tests - to be used in the algorithms we will describe in Section 4.

In order to describe the new mathematical formulation, we need some definitions. First we define, starting from the edge set E, the arc set A as follows:

$$A := \{(i,j),(j,i)|\{i,j\} \in E\} \tag{1}$$

The arc set A contains, for each undirected edge of E, the two corresponding directed arcs.

For each $C \subset V$, we also define:

$$\Gamma(C) := \{\{i,j\}|\{i,j\} \in E \wedge ((i \in C \wedge j \in V \backslash C) \vee (j \in C \wedge i \in V \backslash C))\} \tag{2}$$

In practice, $\Gamma(C)$ contains all the edges of E with a vertex in C and the other in $V \backslash C$.

In the new formulation we present, x variables have the following meaning: $x_{ij} = 1$ if edge $\{i,j\}$ is on the robust spanning tree and 0 otherwise. The remaining variables σ, α and μ define the dual of a multi-commodity flow formulation for the MST problem, where flows are originated at node $0 \in V$ (see Yaman et al. [28]). For this dual formulation, the cost of each arc (i,j) is defined by $l_{ij} + (u_{ij} - l_{ij})x_{ij}$, for a given vector x. In this way, when $x_{ij} = 1$ the length of arc (i,j) is set to its upper bound. Elsewhere the length of arc (i,j) is at its lower bound.

$$
\begin{aligned}
(RST)\ \ \min & \sum_{\{i,j\}\in E} u_{ij}x_{ij} - \sum_{k\in V\setminus\{0\}} \left(\alpha_k^k - \alpha_0^k\right) - (|V|-1)\mu && && (3)\\
\text{s.t. } & \sigma_{ij}^k \geq \alpha_j^k - \alpha_i^k && \forall (i,j)\in A, \forall k \in V\setminus\{0\} && (4)\\
& \sum_{k\in V\setminus\{0\}} \sigma_{ij}^k + \mu \leq l_{ij} + (u_{ij}-l_{ij})x_{ij} && \forall \{i,j\}\in E && (5)\\
& \sum_{k\in V\setminus\{0\}} \sigma_{ji}^k + \mu \leq l_{ij} + (u_{ij}-l_{ij})x_{ij} && \forall \{i,j\}\in E && (6)\\
& \sum_{\{i,j\}\in E} x_{ij} = |V|-1 && && (7)\\
& \sum_{\{i,j\}\in \Gamma(C)} x_{ij} \geq 1 && \forall C\subset V && (8)\\
& \sigma_{ij}^k \geq 0 && \forall (i,j)\in A, k\in V\setminus\{0\} && (9)\\
& \alpha_i^k \geq 0 && \forall i\in V, \forall k\in V\setminus\{0\} && (10)\\
& \mu \text{ unrestricted} && && (11)\\
& x_{ij}\in\{0,1\} && \forall \{i,j\}\in E && (12)
\end{aligned}
$$

Constraints (7) and (8) define, together with the first summation of the objective function (3), a classic minimum spanning tree problem on the scenario where every cost is at its upper bound. The second and third terms of (3) define, together with constraints (4)-(6), the dual of a classic minimum spanning tree formulation in the scenario induced by x variables (see Yaman et al [28]). Constraints (9)-(12) finally define variables domains.

The bottleneck of formulation RST is represented by constraints (8), which are simply too many already for medium size problems. In Section 4.3 we will describe a method which permits us to efficiently handle them within the Benders decomposition algorithm.

4 A Benders decomposition approach

Benders partitioning method was originally proposed in 1962 in Benders [5] (see also Geoffrion [11]). It was initially developed to solve mixed integer programming problems. Geoffrion and Graves [12] confirmed that the method is suitable to solve large scale multicommodity distribution system design models. Many other applications of Benders decomposition have been proposed since then (see, for example, Richardson [27] and Magnanti et al. [17], Cordeau et al. [7,8]). Methodologies for improving the performance of the method have been proposed in McDaniel and Devine [20] and Magnanti and Wong [19].

Kouvelis and Yu [14] derived an algorithm for the scenario version of robust network design problems by adapting Benders decomposition. Montemanni and Gambardella [22] applied Benders decomposition to the robust shortest path problem with interval data.

In this section we describe the application of Benders decomposition to the robust spanning tree problem with interval data.

4.1 Reformulation of RST

Let X be the set of binary vectors for the x variables that satisfy constraints (7) and (8) (i.e. vectors of x variables that describe spanning trees in G). For any given vector $\widetilde{x} \in X$, it is possible to define a problem in variables σ, α and μ only, starting from the mixed integer program RST. We will refer to this problem as the *primal subproblem*. It is defined as follows:

$$
\begin{aligned}
(P(\tilde{x})) \quad & \sum_{\{i,j\}\in E} u_{ij}\tilde{x}_{ij} + \overbrace{\min\left(-\sum_{k\in V\setminus\{0\}}\left(\alpha_k^k-\alpha_0^k\right)-(|V|-1)\mu\right)}^{-\max\left(\sum_{k\in V\setminus\{0\}}\left(\alpha_k^k-\alpha_0^k\right)+(|V|-1)\mu\right)} && (13)\\
\text{s.t. } & \sigma_{ij}^k \geq \alpha_j^k - \alpha_i^k && \forall(i,j)\in A, \forall k\in V\setminus\{0\} \quad (14)\\
& \sum_{k\in V\setminus\{0\}} \sigma_{ij}^k + \mu \leq l_{ij} + (u_{ij}-l_{ij})\tilde{x}_{ij} && \forall\{i,j\}\in E \quad (15)\\
& \sum_{k\in V\setminus\{0\}} \sigma_{ji}^k + \mu \leq l_{ij} + (u_{ij}-l_{ij})\tilde{x}_{ij} && \forall\{i,j\}\in E \quad (16)\\
& \sigma_{ij}^k \geq 0 && \forall(i,j)\in A, k\in V\setminus\{0\} \quad (17)\\
& \alpha_i^k \geq 0 && \forall i\in V, \forall k\in V\setminus\{0\} \quad (18)\\
& \mu \text{ unrestricted} && (19)
\end{aligned}
$$

We now consider the dual of problem $P(\tilde{x})$, which we will refer to as the *dual subproblem*. As observed near the end of Section 3, variables σ, α and μ describe the dual of a multi-commodity flow formulation for the minimum spanning tree problem. Consequently, by dualizing again, we end up with a classic formulation for the minimum spanning tree problem in variables f and y (see Magnanti and Wolsey [18]):

$$
\begin{aligned}
(D(\widetilde{x}))\quad & z_D^*(\widetilde{x}) = \sum_{\{i,j\}\in E} u_{ij}\widetilde{x}_{ij} - \min \sum_{\{i,j\}\in E} \left(l_{ij} + (u_{ij}-l_{ij})\widetilde{x}_{ij}\right)(y_{ij}+y_{ji}) && (20)\\
\text{s.t.}\quad & \sum_{(j,0)\in A} f_{j0}^k - \sum_{(0,j)\in A} f_{0j}^k = -1 && \forall k \in V\backslash\{0\} \quad (21)\\
& \sum_{(j,i)\in A} f_{ji}^k - \sum_{(i,j)\in A} f_{ij}^k = 0 && \forall i \in V, \forall k \in V\backslash\{0\}, i \neq k \quad (22)\\
& \sum_{(j,k)\in A} f_{jk}^k - \sum_{(k,j)\in A} f_{kj}^k = 1 && \forall k \in V\backslash\{0\} \quad (23)\\
& f_{ij}^k \leq y_{ij} && \forall (i,j) \in A, k \in V\backslash\{0\} \quad (24)\\
& \sum_{(i,j)\in A} y_{ij} = |V| - 1 && (25)\\
& f_{ij}^k \geq 0 && \forall (i,j) \in A, \forall k \in V\backslash\{0\} \quad (26)\\
& y_{ij} \geq 0 && \forall (i,j) \in A \quad (27)
\end{aligned}
$$

Notice that notwithstanding constraints (27) are linear, there will be at least one optimal solution of $D(\widetilde{x})$ where y (and f) variables assume binary values only. This happens since the constraints matrix of $D(\widetilde{x})$ is unimodular (see Magnanti and Wolsey [18]).

Notice that problem $D(\widetilde{x})$ is feasible $\forall \widetilde{x} \in X$, since it is a minimum spanning tree problem on a version of graph G with modified costs, and in Section 2 we made the assumption that graph G is connected.

Let R be the feasible region of the dual subproblem and let R_P be the set of extreme points of R (notice that we have no extreme rays of R since it is bounded).

Note that R does not depend on $\widetilde{x}$ (it appears only in the objective function (20)) and $R \neq \emptyset$ by definition, since we suppose that graph G is connected (see Section 2). Hence, by strong duality and by using the fact that R is a polytope, the primal subproblem is feasible and bounded.

We can observe that $D(\widetilde{x})$ is a linear program, i.e. its optimal solutions are in the extreme points. We can then rewrite the original problem RST in a more compact form as follows:

$$\min_{x \in X} \{z_D^*(x)\} \tag{28}$$

By expanding the definition of $z_D^*(x)$ within (28), problem RST can be further rewritten as follows:

$$\min_{x \in X} \left\{ \sum_{\{i,j\} \in E} u_{ij} x_{ij} - \min_{y \in R_P} \sum_{\{i,j\} \in E} \left(l_{ij} + (u_{ij} - l_{ij}) x_{ij}\right) (y_{ij} + y_{ji}) \right\} \tag{29}$$

We now introduce the additional free variable z and we expand the definitions of X and R_P. We obtain the Benders reformulation of RST, which we will refer to as the *master problem* M:

$$(M) \qquad \min z \tag{30}$$

$$\text{s.t. } z \geq \sum_{\{i,j\} \in E} u_{ij} x_{ij} - \sum_{\{i,j\} \in E} \left(l_{ij} + (u_{ij} - l_{ij}) x_{ij}\right) (y_{ij} + y_{ji}) \ \ \forall y \in R_P \tag{31}$$

$$\sum_{\{i,j\} \in E} x_{ij} = |V| - 1 \tag{32}$$

$$\sum_{\{i,j\} \in \Gamma(C)} x_{ij} \geq 1 \qquad \forall C \subset V \tag{33}$$

$$x_{ij} \in \{0, 1\} \qquad \forall \{i,j\} \in E \tag{34}$$

The master problem is in a suitable form for applying the Benders decomposition algorithm, which is described in the following section.

4.2 The algorithm

The algorithm is based on an iterative mechanism. Let τ represent the iteration number and let R_P^τ represent the restricted set of extreme points of R_P available at iteration τ. Each of these extreme points produces a so-called Benders cut. These cuts will be iteratively added during the execution of the Benders decomposition algorithm, which can be summarized as follows:

- **Initialization step:**

  Set $\tau = 1$ and $R_P^1 = \emptyset$.

- **Main step:**

(1) Solve the following mixed integer problem, M^τ, which is the relaxed version of the master problem obtained by replacing R_P with R_P^τ, i.e. by considering the extreme points available at iteration τ only.

$$(M^\tau) \quad \min z \tag{35}$$

$$\text{s.t. } z \geq \sum_{\{i,j\}\in E} u_{ij}x_{ij} - \sum_{\{i,j\}\in E} (l_{ij} + (u_{ij} - l_{ij})x_{ij})\,(y_{ij} + y_{ji}) \quad \forall y \in R_P^\tau \tag{36}$$

$$\sum_{\{i,j\}\in E} x_{ij} = |V| - 1 \tag{37}$$

$$\sum_{\{i,j\}\in \Gamma(C)} x_{ij} \geq 1 \qquad \forall C \subset V \tag{38}$$

$$x_{ij} \in \{0,1\} \qquad \forall \{i,j\} \in E \tag{39}$$

A procedure to efficiently solve M^τ to optimality will be described in Section 4.3.

(2) Let (z^τ, x^τ) be an optimal solution of M^τ.

Solve the mixed integer problem $D(x^\tau)$. Notice that it being a minimum spanning tree problem, a polynomial time algorithm (see, for example, Ahuja et al [1]) can be used to solve it. Using an ad-hoc minimum spanning tree algorithm instead of directly solving $D(x^\tau)$, also excludes optimal fractional solutions, which would produce useless extreme points. They might exist when more than one optimal integer solutions exist.

(3) It is now possible to observe (see Benders [5]) that $z_D^*(x^\tau)$ and z^τ are respectively an upper bound and a lower bound of the optimal cost of the original problem RST.

If $z_D^*(x^\tau) \leq z^\tau$ then the optimal solution of RST has been found, **stop**.
Otherwise, let (f^τ, y^τ) be an optimal solution of $D(x^\tau)$.
Set $R_P^{\tau+1} = R_P^\tau \cup \{y^\tau\}$, $\tau = \tau + 1$ and **repeat the Main step**.

Notice that problem M^1 is a minimum spanning tree problem, and its constraint matrix is consequently unimodular (i.e. M^1 is an "easy" integer program). As τ increases, M^τ progressively looses unimodular characteristics, and it becomes more and more difficult to solve in terms of integer programming.

In Yaman et al. [28] a preprocessing technique, which identifies edges that will never be on a robust spanning tree, is presented. This technique has been shown to be very effective in reducing computation times and is easily incorporated into our approach, by running it before the Benders decomposition algorithm is launched. In particular the edges identified during the preprocessing phase are deleted from graph G, according to Montemanni and Gambardella [23]. Note that instead of the original implementation of the procedure (see [28]), we adopted the (faster) implementation suggested in Aron and Van Hentenryck [2].

4.3 A procedure to efficiently solve the mixed integer program M^τ

As already observed in Section 3, it is not very convenient to consider all the cut constraints (38). We will then add them as the computation proceeds in case they are violated. For this purpose we consider a set β^τ, which will contain the subsets of V (each one corresponds to a constraint of type (38)) considered at each iteration τ of the algorithm described in Section 4.2. β^0 has to be initialized to $\emptyset$ during the **Initialization step** and will be expanded during the executions of the **Main step**.

At each iteration of the **Main step**, a nested iterative procedure is used to solve M^τ. We define problem $\widehat{M}^\tau$ as M^τ with constraints (38) substituted by the following subset of them:

$$\sum_{\{i,j\}\in\Gamma(C)} x_{ij} \geq 1 \quad \forall C \in \beta^\tau \tag{40}$$

The mixed integer program $\widehat{M}^\tau$ is solved. Let $(\widehat{z}^\tau, \widehat{x}^\tau)$ be its optimal solution. If the elements of $\widehat{x}^\tau$ with value 1 form a spanning tree then an optimal solution of M^τ has been retrieved and the procedure can be stopped, otherwise set β^τ is expanded to include the new subsets corresponding to the connected components defined by variables with value 1 in $\widehat{x}^\tau$. In particular, if there are exactly two connected components, one of them is added to β^τ. If there are more than two, each connected component is inserted into β^τ. After the enlargement of β^τ, the augmented problem $\widehat{M}^\tau$ is solved, and the nested iterative step is repeated again.

At the end of the **Main step**, if the optimal solution of problem RST has not been found, the assignment $\beta^{\tau+1} = \beta^\tau$ has to be done.

With the approach described above, the set β^τ is incrementally augmented during the execution of the Benders decomposition algorithm, since at each iteration of the **Main step** new subsets of V are potentially inserted into it by the nested iterative procedure.

The computational experiments presented in Section 5.2 will show that the solving strategy described above is a crucial factor in the good performance of the method described in this paper, since only a small fraction of all of the subsets of V will be sufficient to define an optimal solution of RST.

5 Computational experiments

In Section 5.1 we describe the benchmarks we will adopt for our experiments, while in Section 5.2 we analyze the behavior of the Benders decomposition approach described in Section 4. Section 5.3 is finally devoted to the comparison of the performance of the new method with those of other exact methods recently appeared in the literature.

5.1 Benchmarks description

The benchmarks we will use are those usually adopted in the literature. The first set of problems has been originally introduced in Yaman et al. [28]. It is composed of complete graphs with $|V| = 10$, 15, 20 and 25. This set is here extended (as already done in Aron and Van Hentenryck [2]) with graphs with $|V| = 30$, 35 and 40. For problems with $|V| = 10$, 15 and 20, six groups of five problems each are generated at random, according to the following definition. Each group k is defined by two intervals L^k and U^k, summarized in Table 1.

Table 1

Definition of benchmark set 1.

Group	1	2	3	4	5	6
L^k	$[0, 10)$	$[0, 15)$	$[0, 20)$	$[0, 10)$	$[0, 15)$	$[0, 20)$
U^k	$(l_{ij}, 10]$	$(l_{ij}, 15]$	$(l_{ij}, 20]$	$(l_{ij}, 20]$	$(l_{ij}, 30]$	$(l_{ij}, 40]$

In each problem of group k, $\forall\{i, j\} \in E$ we define the interval associated with the edge by first selecting l_{ij} at random from L^k and then u_{ij} at random from U^k. For graphs with $|V| = 25$, 30, 35 and 40, only five test problems from group 1 of Table 1 are considered, because of the long computation times.

The second set of benchmarks considered has been already adopted in Montemanni and Gambardella [23]. All these problems are complete graphs with $|V| = 20$. Vertices are randomly placed on a 50×50 grid. $\forall\{i, j\} \in E$, l_{ij} is randomly selected in $[ed_{ij}(1-p), ed_{ij})$ and u_{ij} in $(l_{ij}, ed_{ij}(1+p)]$, where ed_{ij} is the euclidian distance between i and j, and p is a distortion parameter, which regulates interval width. Experiments on these benchmarks will help to understand how the performance of the algorithms are affected by changes in parameter p. In particular, we will consider three different values for p, 0.15, 0.50 and 0.85, and for each of them we report the average results over five instances.

5.2 *Analysis of the Benders decomposition algorithm*

This section is dedicated to the experimental analysis of the Benders decomposition algorithm described in Section 4. In particular we aim to study the behavior of the algorithm on problems with different characteristics and to

Table 2

Analysis of the Benders decomposition algorithm. Benchmark set 1.

$\vert V\vert$	$\overline{\tau}$	$\overline{\vert\beta^\tau\vert/\tau}$
10	5.83	2.25
15	11.73	1.78
20	21.43	1.42
25	25.60	1.98
30	31.00	3.19
35	44.67	3.92
40	63.33	4.72

understand the characteristics of the problems on which the new approach is likely to perform well.

In Tables 2 and 3 we report, for each family of problems, the average number of iterations of the **Main step** required by the algorithm described in Section 4.2 to certify an optimal solution ($\overline{\tau}$) and the average number of iterations of the nested iterative procedure (see Section 4.3) necessary to solve each master problem M^τ ($\overline{|\beta^\tau|/\tau}$).

The number of iterations required by the Benders decomposition approach to converge increases, as expected, with $|V|$. The positive note is that the number of iterations does not explode for large values of $|V|$.

The changes of $\overline{|\beta^\tau|/\tau}$ as $|V|$ increases are also very interesting. When $|V|$ increases from 10 to 20 (and $\overline{\tau}$ increases consequently) $\overline{|\beta^\tau|/\tau}$ decreases. This

Table 3

Analysis of the Benders decomposition algorithm. Benchmark set 2.

p	$\overline{\tau}$	$\overline{\vert\beta^\tau\vert/\tau}$
0.15	2.40	2.25
0.50	9.20	1.76
0.85	23.20	1.72

indicates that the augmenting strategy we implemented for β^τ is good, since when $\overline{\tau}$ increases fewer and fewer new constraints are necessary for each new master problem M^τ faced, because the constraints contained in $\beta^{\tau-1}$ already define quite well the polytope of M^τ. This nice phenomenon is, in our opinion, hidden by the intrinsic complexity explosion of the problems, when V is greater than 20.

Table 3 shows that $\overline{\tau}$ increases at the increasing of p, i.e. at the increasing of the average interval width. There is an intuitive explanation for this. When interval widths are small, the scenarios induced by the different spanning trees tend to be similar to each other. For this reason also the minimum spanning trees on these scenarios will be potentially very similar, and consequently already after a few iterations the set R_P^τ will give a good approximation of the polytope defined by the whole R_P.

Our conjecture about the decreasing of $\overline{|\beta^\tau|/\tau}$ at the increasing of $\overline{\tau}$ is supported by Table 3.

5.3 *Exact algorithms comparison*

In this section we compare the computation times required by the new algorithm described in Section 4.2 (*Benders decomposition*) with those of other exact methods. In particular we consider the approach described in Yaman et al [28] (*YKP* - preprocessing technique and mixed integer programming model), Aron and Van Hentenryck [2] (*AVHa* and *AHVb* - branch and bound algorithms) and Montemanni and Gambardella [23] (*MG* - branch and bound algorithm).

In Tables 4 and 5 we report, for each value of $|V|$, the average computation time required by the methods considered. Entries marked with "-" correspond to combinations for which no result is available. Entries containing ">" correspond to combinations for which the algorithm was not able to find the optimal solution for some of the instances within the time limit of 3600 seconds, after which the computation was stopped. In Table 5 only the methods for which some results are available, are reported.

The reference machine for Table 4 is an Intel Pentium II 400 MHz computer. The computational times reported for the algorithm described in Yaman et al. [28] were obtained on a slightly faster Intel Pentium II 450 MHz machine. In brackets we report also the results of new experiments run on the reference Intel Pentium II 400 MHz machine. The results reported in the columns relative to the algorithms presented in Aron and Van Hentenryck [2] were obtained on a substantially faster Sun Sparc 440 MHz computer. In brackets we report the same results multiplied by a factor of 2.5 (as suggested in Dongarra [10]) in order to have computation times comparable with those of the reference Intel

Pentium II 400 MHz machine. ILOG CPLEX [1] 6.0 has been finally adopted to solve the mixed integer programs faced during the execution of the algorithm described in Section 4.2.

[1] http://www.cplex.com.

Table 4

Exact algorithms comparison (computation times). Benchmark set 1.

$\|V\|$	YKP [28]		AVHa [2]		AVHb [2]		MG [23]	Benders decomposition
	P2 450	(P2 400)	SS 440	(P2 400)	SS 440	(P2 400)	P2 400	P2 400
10	1.96	(0.85)	0.11	(0.28)	0.08	(0.20)	**0.04**	0.15
15	33.09	(286.47)	1.90	(4.74)	1.07	(2.68)	1.84	**1.23**
20	693.88	(>2016.28)	27.98	(69.95)	15.18	(37.94)	56.46	**16.43**
25	2027.60	(>3600.00)	244.18	(610.45)	121.85	(304.63)	484.24	**58.19**
30	-	(-)	2100.43	(5251.08)	926.65	(2316.63)	-	**531.30**
35	-	(-)	10771.37	(26928.43)	4639.55	(11598.88)	-	**3995.00**
40	-	(-)	29421.70	(73554.25)	27206.38	(68015.95)	-	**16479.85**

Table 5

Exact algorithms comparison (computation times). Benchmark set 2.

p	YKP [28]	MG [23]	Benders decomposition
0.15	320.91	1.53	**0.35**
0.50	61.41	0.73	**0.66**
0.85	33.74	**0.46**	16.70

The computational results presented in Table 4 show that the Benders decomposition approach presented in this paper is substantially faster than the other exact algorithms. In fact it is not the faster one only for the smallest problems considered, for which computational times of all the algorithms are almost negligible. We believe that for these problems, the time requires to load the data structures used by the mixed integer program solver, dominates the solving time itself.

In Table 5 the average computations times obtained by some of the algorithm on the second benchmark set are reported. The experiments were carried out on a SUNW Ultra-30 machine (with ILOG CPLEX 6.0).

The results of Table 5 are very interesting. They show that the new approach described in Section 4.2 has an opposite behavior, with respect to the other methods considered, when parameter p is varied. In fact, it is considerably better than the other approaches when average cost intervals are small ($p = 0.15$), it is still better (but comparable with the branch and bound algorithm presented in Montemanni and Gambardella [23]) for medium size cost intervals ($p = 0.50$), and (significantly) slower than the method presented in [23] for large average cost intervals ($p = 0.85$).

This result was expected since, as explained in Section 5.2, the Benders decomposition approach profits from small average interval widths. On the other hand, in Montemanni and Gambardella [23] it is explained why the branch and bound algorithms considered there are likely to perform better on problems with large average interval widths. Our conclusion is then that the characteristics of the problem under consideration (in particular interval widths) should drive the choice of the most promising algorithm to tackle the problem.

6 Conclusion

A new algorithm, based on Benders decomposition has been proposed for the interval data version of the robust spanning tree problem.

As shown by computational experiments, the new method is very efficient. It appears to be the fastest approach for most of the benchmarks considered. In particular the new Benders decomposition method is very efficient on problems with small average interval width. It is important to note that these problems are among the most difficult ones for the exact algorithms previously appeared in the literature.

Acknowledgements

The work was partially funded by the European Commission IST project *BISON* (IST-2001-38923).

References

[1] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms and Applications.* Prentice-Hall, 1993.

[2] I. Aron and P. Van Hentenryck. A constraints satisfaction approach to the robust spanning tree problem with interval data. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence (UAI)*, August 2002.

[3] I. Aron and P. Van Hentenryck. On the complexity of the robust spanning tree with interval data. *Operations Research Letters*, 32(1):136–140, 2003.

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

[5] J.F. Benders. Partitioning procedures for solving mixed integer variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

[6] D.P. Bertsekas and R. Gallagher. *Data Networks.* Prentice-Hall, Englewood Cliffs, NJ, 1987.

[7] J.-F. Cordeau, G. Laporte, and A. Mercier. A Benders decomposition approach for the locomotive and car assignment problem. *Transportation Science*, 34:133–149, 2000.

[8] J.-F. Cordeau, G. Stojkovic, F. Soumis, and J. Desrosiers. Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation Science*, 35:375–388, 2001.

[9] A.V. Donati, R. Montemanni, L.M. Gambardella, and A.E. Rizzoli. Integration of a robust shortest path algorithm with a time dependent vehicle routing model and applications. In *Proceedings of CIMSA 2003*, pages 26–31, July 2003.

[10] J.J. Dongarra. Performance of various computers using standard linear algebra software in a fortran environment. Technical Report CS-89-85, University of Tennessee, July 2003.

[11] A.M. Geoffrion. Generalized Benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, 1972.

[12] A.M. Geoffrion and G.W. Graves. Multicommodity distribution system design by Benders decomposition. *Management Science*, 20(5):822–844, 1974.

[13] O.E. Karaşan, M.Ç. Pinar, and H. Yaman. The robust shortest path problem with interval data. Bilkent University, 2001.

[14] P. Kouvelis and G. Yu. *Robust Discrete Optimization and its applications*. Kluwer Academic Publishers, 1997.

[15] G.L. Kozina and V.A. Perepelista. Interval spanning trees problem: solvability and computational complexity. *Interval Computations*, 1:42–50, 1994.

[16] J.B. Kruskal. On the shortest spanning subtree of a graph and the travelling salesman problem. *Preceedings of the American Mathematical Society*, 7:48–50, 1956.

[17] T.L. Magnanti, R. Mirchandani, and R.T. Wong. Tailoring Benders decomposition for uncapacitated network design. *Mathematical Programming Study*, 26:112–154, 1986.

[18] T.L. Magnanti and L. Wolsey. Optimal trees. In *Handbook in Operations Research and Management Science* (M.O. Ball et al. eds.), volume 7, pages 503–615. North Holland, Amsterdam, 1995.

[19] T.L. Magnanti and R.T. Wong. Accelerating Benders decomposition: algorithmic enhancement and model selection criteria. *Operations Research*, 29(3):464–483, 1981.

[20] D. McDaniel and M. Devine. A modified Benders' partitioning algorithm for mixed integer programming. *Management Science*, 24:312–379, 1977.

[21] R. Montemanni and L.M. Gambardella. An exact algorithm for the robust shortest path problem with interval data. *Computers and Operations Research*, 31(10):1667–1680, 2004.

[22] R. Montemanni and L.M. Gambardella. The robust shortest path problem with interval data via Benders decomposition. Submitted for publication, 2004.

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

[24] R. Montemanni, L.M. Gambardella, and A.V. Donati. A branch and bound algorithm for the robust shortest path problem with interval data. *Operations Research Letters*, 32(3):225–232, 2004.

[25] R. Montemanni, L.M. Gambardella, and A.V. Donati. A comparison of two new exact algorithms for the robust shortest path problem. In *Proceedings of TRISTAN V – The Fifth TRIennial Symposium on Transportation ANalysis*, June 2004.

[26] R.C. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36:1389–1401, 1957.

[27] R. Richardson. An optimization approach to routing aircraft. *Transportation Science*, 10:52–71, 1976.

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

[29] H. Yaman, O.E. Karaşan, and M.Ç. Pinar. Restricted robust optimization for maximization over uniform matroid with interval data uncertainty. Bilkent University, 2004.

[30] G. Yu and J. Yang. On the robust shortest path problem. *Computers and Operations Research*, 25(6):457–468, 1998.

[31] P. Zieliński. The computational complexity of the relative robust shortest path problem with interval data. *European Journal of Operational Research*, 158(3):570–576, 2004.