4OR 3: 315–328 (2005)

DOI: 10.1007/s10288-005-0066-x

# The robust shortest path problem with interval data via Benders decomposition

**Roberto Montemanni and Luca Maria Gambardella**

Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Galleria 2, 6928 Lugano-Manno, Switzerland (e-mail: {roberto;luca}@idsia.ch)

Received: September 2004 / Accepted: March 2005

**Abstract.** Many real problems can be modelled as robust shortest path problems on digraphs with interval costs, where intervals represent uncertainty about real costs and a robust path is not too far from the shortest path for each possible configuration of the arc costs.

In this paper we discuss the application of a Benders decomposition approach to this problem.

Computational results confirm the efficiency of the new algorithm. It is able to clearly outperform state-of-the-art algorithms on many classes of networks. For the remaining classes we identify the most promising algorithm among the others, depending of the characteristics of the networks.

**Key words:** Shortest path problem, robust optimization, interval data, Benders decomposition

**MSC classification:** 90C47, 52B05, 90C57

## 1 Introduction

When transportation problems are modelled in mathematical terms, a road network is usually represented as a weighted digraph, where each arc is associated with a road and costs represent travel times. In this context, a shortest path problem has to be solved every time the quickest way to go from one place to another has to be calculated.

The work was partially supported by the European Commission IST projects *MOSCA* (IST-2000-29557) and *BISON* (IST-2001-38923).

*All correspondence to:* Roberto Montemanni

4OR A Quarterly Journal of Operations Research

A similar problem arises in telecommunication when a packet has to be sent from a source node to a destination node on a network. Also in this case, where the network is usually modelled as a weighted digraph and costs are associated with transmission delays, a shortest path problem is faced.

Unfortunately, in reality it is not easy to estimate arc costs exactly, since they depend on many factors which are difficult to predict, such as traffic conditions, accidents, traffic jams or weather conditions for the transportation case, network congestions or hardware failures for the telecommunication case. For this reason the fixed cost model previously introduced may be inadequate. To overcome this problem, more complex models have been presented in the literature. In particular a model where a set of alternative graphs are considered at the same time (*scenario model* – see Yu and Jang 1998; Dias and Clímaco 2000) and a model where an interval of possible values is associated with each arc (*interval data model* – see Dias and Clímaco 2000; Karaşan et al. 2001) have been studied. In this work the interval data model, which will be described in detail in Sect. 2, is considered.

With the interval data model, uncertainty is modelled by associating an interval of costs with each arc. Each interval represents a range of possible values for the real cost.

The *relative robustness criterion*, which will be formally defined in Sect. 2, has been chosen to drive optimization. This criterion is discussed in Kouvelis and Yu (1997), a book entirely devoted to robust discrete optimization.

A *relative robust shortest path* (sometimes referred to as the *robust deviation shortest path*) from $s$ to $t$ is a path from $s$ to $t$ which minimizes the maximum deviation from the optimal shortest path from $s$ to $t$ over all realizations of arc costs.

In Averbak and Lebedev (2004) it is proven that the relative robust shortest path problem with interval data is strongly $\mathcal{NP}$-hard.

In the remainder of this paper we will refer to the *relative robust shortest path problem* simply as the *robust shortest path problem* and to a *relative robust shortest path* simply as a *robust shortest path*.

Karaşan et al. (2001) proposed a mixed integer programming formulation for the problem based on an important theoretical result they gave (see Theorem 1). An exact algorithm, based on path-ranking, is presented in Montemanni and Gambardella (2004), while a branch and bound approach is described in Montemanni et al. (2004a) (see also Montemanni et al. 2004b).

In this paper two versions of a novel exact algorithm for the robust shortest path problem with interval data, based on Benders decomposition, are presented.

In Sect. 2 the robust shortest path problem with interval data is formally described. A mixed integer programming formulation for the problem is presented in Sect. 3. Section 4 is devoted to the discussion of the approaches we propose. Computational results are presented in Sect. 5, while conclusions are summarized in Sect. 6.

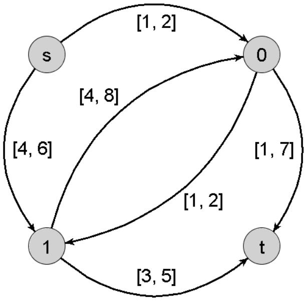

**Fig. 1.** Example of directed graph with interval costs

## 2 Problem description

A directed graph $G = (V, A)$, where $V$ is a set of vertices and $A$ is a set of arcs is given together with a starting vertex $s \in V$, and a destination vertex $t \in V$. We suppose that at least a path from $s$ to $t$ exists in $G$. An interval $[l_{ij}, u_{ij}]$, with $0 \leq l_{ij} \leq u_{ij}$, is associated with each arc $(i, j) \in A$. Intervals represent ranges of possible costs. An example of interval graph is given in Fig. 1.

The robust shortest path problem with interval data can be formally described by the following definitions:

**Definition 1.** *A* scenario *$r$ is a realization of arc costs, i.e. a cost $c^r_{ij} \in [l_{ij}, u_{ij}]$ is fixed $\forall (i, j) \in A$.*

**Definition 2.** *The* robust deviation *for a path $p$ from $s$ to $t$ in a scenario $r$ is the difference between the cost of $p$ in $r$ and the cost of the shortest path from $s$ to $t$ in scenario $r$.*

**Definition 3.** *A path $p$ from $s$ to $t$ is said to be a* robust shortest path *if it has the smallest (among all paths from $s$ to $t$) maximum (among all possible scenarios) robust deviation.*

A scenario can be seen as a snapshot of the network situation, and a robust shortest path is a path which guarantees reasonably good performance (compared to optimal solutions) under any possible configuration of travel times over the network.

Given a directed graph and an origin/destination pair $(s, t)$, the robust shortest path problem is the problem of retrieving a robust shortest path.

The following important result is at the basis of the mathematical formulations described in Sect. 3.

**Theorem 1 (Karaşan et al. 2001).** *The robust deviation for path $p$ is maximized at the scenario in which the lengths of all arcs on $p$ are at upper bounds and the lengths of all other arcs are at lower bounds.*

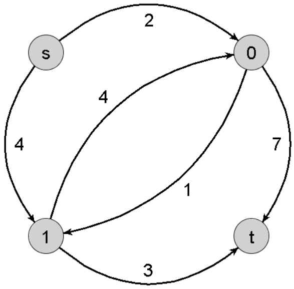

**Fig. 2.** Scenario induced by $p = \{s, 0, t\}$ on the directed graph with interval costs of Fig. 1

Theorem 1 implies that we need to consider only a finite number of scenarios, namely as many as the number of paths in the graph.

Figure 2 depicts the scenario induced by path $p = \{s, 0, t\}$ on the graph of Fig. 1. The robustness cost of $p$ is in this case $(2 + 7) - (2 + 1 + 3) = 3$.

## 3 Mixed integer programming formulation

Karaşan et al. (2001) derived a mixed integer programming formulation for the problem, based on Theorem 1. In this formulation, the $y$ variables have the following meaning: $y_{ij} = 1$ if arc $(i, j)$ is on the robust shortest path and 0 otherwise. The length of arc $(i, j)$ is defined by $l_{ij} + (u_{ij} - l_{ij})y_{ij}$ for a given vector $y$. This is because when $y_{ij} = 1$ the length of arc $(i, j)$ is at its upper bound on path $p$ defined by $y$. All the lengths of other arcs with $y_{ij} = 0$ are at their lower bounds. Variable $x_j$ contains the shortest distance from node $s$ to node $j$. Variable $x_t$ contains then the length of the shortest path in the graph under the scenario defined by $y$. The objective is to find a path $p$ for which the difference between the length of path $p$ and the length of the shortest path in the graph is the smallest when the lengths of all arcs on path $p$ are at their upper bounds and the lengths of all other arcs are at their lower bounds.

$$(RSP) \quad min \sum_{(i,j)\in A} u_{ij}y_{ij} - x_t \tag{1}$$

$$\text{s.t.} \quad x_j \leq x_i + l_{ij} + \left(u_{ij} - l_{ij}\right) y_{ij} \qquad \forall (i, j) \in A \tag{2}$$

$$\sum_{(s,k)\in A} y_{sk} - \sum_{(i,s)\in A} y_{is} = 1 \tag{3}$$

$$\sum_{(t,k)\in A} y_{tk} - \sum_{(i,t)\in A} y_{it} = -1 \tag{4}$$

$$\sum_{(j,k)\in A} y_{jk} - \sum_{(i,j)\in A} y_{ij} = 0 \qquad \forall j \in V \setminus \{s, t\} \tag{5}$$

$$x_s = 0 \tag{6}$$

$$y_{ij} \in \{0, 1\} \qquad \forall (i, j) \in A \tag{7}$$

$$x_j \geq 0 \qquad \forall j \in V \tag{8}$$

Constraints (2) specify shortest distances between nodes based on whether arcs ($y$ variables) are on the path or not. Inequalities (3)-(5) ensure that the resulting $y$ vector defines a path in the graph, while constraint (6) sets to 0 the distance to the source node $s$ and prevents an unbounded solution. Constraints (7) and (8) define the domains of the variables.

Notice that the problem defined by the $x$ variables is the dual of a classic shortest path problem formulation, where distances on arcs are defined by the $y$ variables as explained before.

## 4 A Benders decomposition approach

Benders partitioning method was originally proposed in 1962 in Benders (1962) (see also Geoffrion 1972). It was initially developed to solve mixed integer programming problems. Geoffrion and Graves (1974) 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 1976; Magnanti et al. 1986; Cordeau et al. 2000; Cordeau et al. 2001). Methodologies for improving the performance of the method have been proposed in McDaniel and Devine (1977) and Magnanti and Wong (1981).

Kouvelis and Yu (1997) derived an algorithm for the scenario version of the robust shortest path problem (see Yu and Jang 1998) by adapting Benders decomposition.

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

### 4.1 Reformulation of $RSP$

Let $Y$ be the set of binary vectors for the $y$ variables that satisfy constraints (3)-(5) and (7) (i.e., vectors of the $y$ variables that describe paths from $s$ to $t$ in $G$). For any given vector $\overline{y} \in Y$, it is possible to define a problem in the $x$ variables only, starting from the mixed integer program $RSP$. We will refer to this problem as the *primal subproblem*. It is defined as follows:

$$(P(\overline{y})) \qquad \sum_{(i,j)\in A} u_{ij}\overline{y}_{ij} \overbrace{+\min -x_t}^{-\max x_t} \tag{9}$$

$$\text{s.t.} \quad x_j \leq x_i + l_{ij} + \left(u_{ij} - l_{ij}\right)\overline{y}_{ij} \quad \forall (i, j) \in A \quad (w) \tag{10}$$

$$x_s = 0 \tag{11}$$

$$x_j \geq 0 \forall j \in V \tag{12}$$

We now consider the dual of problem $P(\overline{y})$, which we will refer to as the *dual subproblem.* As observed near the end of Sect. 3 (Karaşan et al. 2001), the $x$ variables describe the dual of a classic shortest path problem formulation. Consequently, by dualizing again, we end up with the following classic shortest path problem formulation on the variables $w$:

$$(D(\overline{y})) \quad z_D^*(\overline{y}) = \sum_{(i,j)\in A} u_{ij}\overline{y}_{ij} - \min \sum_{(i,j)\in A} (l_{ij} + \left(u_{ij} - l_{ij}\right)\overline{y}_{ij})w_{ij} \tag{13}$$

$$\text{s.t.} \quad \sum_{(s,k)\in A} w_{sk} - \sum_{(i,s)\in A} w_{is} = 1 \tag{14}$$

$$\sum_{(t,k)\in A} w_{tk} - \sum_{(i,t)\in A} w_{it} = -1 \tag{15}$$

$$\sum_{(j,k)\in A} w_{jk} - \sum_{(i,j)\in A} w_{ij} = 0 \qquad \forall j \in V\backslash\{s,t\} \tag{16}$$

$$0 \leq w_{ij} \leq 1 \qquad \forall (i,j) \in A \tag{17}$$

Notice that notwithstanding constraints (17), there will be at least an optimal solution of $D(\overline{y})$ where the $w$ variables assume binary values only. This happens since the constraints matrix of $D(\overline{y})$ (i.e., a shortest path problem) is unimodular (see, for example, Ford and Fulkerson 1962).

Notice that problem $D(\overline{y})$ is feasible $\forall \overline{y} \in Y$, since it is a shortest path problem on a version of graph $G$ with modified costs, and in Sect. 2 we made the assumption that at least a path from $s$ to $t$ in $G$ exists.

Let $R$ be the feasible region of the dual subproblem and let $P_R$ be the set of extreme points of $R$ (notice that we have no extreme rays of $R$ because of constraints (17)).

Note that $R$ does not depend on $\overline{y}$ (it appears only in the objective function (13), and $R \neq \emptyset$ by definition, since we suppose that at least a path from $s$ to $t$ in $G$ exists (see Sect. 2). Hence, by strong duality, the primal subproblem is feasible and bounded.

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

$$(RSP) \quad \min_{y\in Y} \left\{z_D^*(y)\right\} \tag{18}$$

By expanding the definition of $z_D^*(y)$ within (18), problem $RSP$ can be further rewritten as follows:

$$(RSP) \min_{y\in Y} \left\{ \sum_{(i,j)\in A} u_{ij}y_{ij} - \min_{w\in P_R} \sum_{(i,j)\in A} (l_{ij} + \left(u_{ij} - l_{ij}\right) y_{ij})w_{ij} \right\} \tag{19}$$

We now introduce the additional free variable $z$ and we expand the definition of $Y$. We obtain the Benders reformulation of $RSP$, which we will refer to as the *master problem M*:

$$(M) \qquad \min z \tag{20}$$

$$\text{s.t.} \quad z \geq \sum_{(i,j)\in A} u_{ij} y_{ij} - \sum_{(i,j)\in A} (l_{ij} + \left(u_{ij} - l_{ij}\right) y_{ij}) w_{ij} \quad \forall w \in P_R \tag{21}$$

$$\sum_{(s,k)\in A} y_{sk} - \sum_{(i,s)\in A} y_{is} = 1 \tag{22}$$

$$\sum_{(t,k)\in A} y_{tk} - \sum_{(i,t)\in A} y_{it} = -1 \tag{23}$$

$$\sum_{(j,k)\in A} y_{jk} - \sum_{(i,j)\in A} y_{ij} = 0 \qquad \forall\, j \in V \backslash \{s,t\} \tag{24}$$

$$y_{ij} \in \{0,1\} \quad \forall (i,j) \in A \tag{25}$$

$$z \in \mathcal{R}^+ \cup \{0\} \tag{26}$$

The master problem is in a suitable form for applying the Benders decomposition algorithm, which is described in the following section. Some improvements to the basic idea will be presented in Sect. 4.3.

### *4.2 Basic algorithm*

The algorithm is based on an iterative mechanism. Let $\tau$ represent the iteration number and let $P_R^\tau$ represent the restricted set of extreme points of $P_R$ available at iteration $\tau$. The basic Benders decomposition algorithm can be summarized as follows:

- **Initialization step:**
  Set $\tau = 1$ and $P_R^1 := \emptyset$.
- **Main step:**
  Solve the following mixed integer problem, $M^\tau$, which is the relaxed version of the master problem obtained by replacing $P_R$ with $P_R^\tau$, i.e. by considering the extreme points available at iteration $\tau$ only.
  Formally the mixed integer program $M^\tau$ is obtained by changing constraints (21) with the following ones:

$$z \geq \sum_{(i,j)\in A} u_{ij} y_{ij} - \sum_{(i,j)\in A} (l_{ij} + \left(u_{ij} - l_{ij}\right) y_{ij}) w_{ij} \quad \forall w \in P_R^\tau \tag{27}$$

  Let $(z^\tau, y^\tau)$ be an optimal solution of $M^\tau$.
  Solve the mixed integer problem $D(y^\tau)$. Notice that since this is a classic shortest path problem, a polynomial time algorithm (see, for example, Ahuja ez al. 1993) can be used to solve it. The use of an ad-hoc shortest path algorithm instead of directly solving $D(y^\tau)$ also prevents optimal fractional solutions,

which would produce less interesting extreme points. They can exist when several optimal integer solutions exist.
It is possible now to observe (see Benders 1962) that $z_D^*(y^\tau)$ and $z^\tau$ are respectively an upper bound and a lower bound of the optimal cost of the original problem $RSP$.
If $z_D^*(y^\tau) \leq z^\tau$ then the optimal solution of $RSP$ has been found, **stop**.
Otherwise, let $w^\tau$ be an optimal solution of $D(y^\tau)$.
Set $P_R^{\tau+1} := P_R^\tau \cup \{w^\tau\}$, $\tau := \tau + 1$ and **repeat the Main step**.

Notice that problem $M^1$ is a shortest path problem, and its constraints matrix is consequently unimodular (i.e., $M^1$ is an "easy" integer program). As $\tau$ increases, $M^\tau$ progressively looses unimodular characteristics, and it becomes more and more difficult (and time consuming) to solve in terms of integer programming.

We will then expect the algorithm to perform well on problems for which just a few iterations are necessary. Computational times (see Sect. 5.2) suggest that this is the case.

### 4.3 Improved algorithm

A major difficulty with the algorithm presented in Sect. 4.2 lies in the repeated solution of master problem $M^\tau$, which, as previously observed, becomes more and more difficult at the increasing of $\tau$. It is faced at each iteration of the algorithm.

To accelerate the solution process of the master problem, McDaniel and Devine (1977) suggested relaxing the integrality constraints on the variables of the master problem and generating cuts from the fractional solutions. This approach can be adapted to our problem, and the improved Benders decomposition we obtain can be summarized as follows:

- **Preamble:**
Run $n_{pp}$ iterations of the basic algorithm (see Sect. 4.2) on the relaxed problem $M_{LR}$, which is obtained from the mixed integer program $M$ by substituting constraints (25) with their linear relaxation

$$0 \leq y_{ij} \leq 1 \qquad \forall (i,j) \in A \tag{28}$$

Notice that since $y^\tau$ is not necessarily a binary vector now, $D(y^\tau)$ can have arc costs (i.e., coefficients of objective function (13)) that are not at their lower or upper bounds.
- **Initialization step:**
Set $P_R^1 := P_R^\tau$ and $\tau = 1$.
The difference from the basic algorithm is that, instead of starting with an empty set $P_R^1$, in this case we initialize $P_R^\tau$ in such a way that it contains the $n_{pp}$ cuts generated for the relaxed problem $M_{LR}$ during the **preamble**. This should help to skip some time-consuming problems $M^\tau$.

- **Main step:**
  Execute the main step of the basic algorithm (see Sect. 4.2).

Notice that the cuts contained in $P_R^\tau$ at the end of the $n_{pp}$ iterations carried out on $M_{LR}$ during the preamble phase, are feasible also for $M$, since the relaxation of constraint set (25) does not affect the dual subproblem polyhedron. For this reason these cuts can be inserted into $P_R^1$ during the initialization step.

The initial cuts obtained from the linear relaxation $M_{LR}$ are inserted in order to improve the convergency speed of the algorithm.

It is worth to observe that parameter $n_{pp}$ is very important, since too many initial cuts could end up to slow down the algorithm, since some of them may be useless. On the other hand, the contribution of the initial cuts should save some iterations of the main step of the algorithm. Some computational tests (not reported) clearly suggested that $n_{pp} \in \{2, 3\}$ is the best setting for all the problems we will consider. We will therefore set $n_{pp} = 2$ for the experiments presented in Sect. 5.

## 5 Computational experiments

In the next sections we summarize some experiments we have carried out on different families of networks. The aim is to measure the performances of the two versions of the Benders decomposition approach we propose, which are also compared with state-of-the-art algorithms. We also present a comparative study where we identify relations between the characteristics of the networks and the performances of the exact algorithms presented so far in the literature.

For all the tests reported, ILOG CPLEX 6.0 (http://www.cplex.com) has been used to solve linear and mixed integer programs. The algorithm described in Dijkstra (1959) has been adopted to solve classic shortest path problems. All the experiments have been carried out on the same Intel Pentium 4 1.5GHz / 256 MB computer.

### *5.1 Networks*

The different families of networks on which the experiments have been carried out are described in the following paragraphs.

#### 5.1.1 Random networks

This family of networks has been originally proposed in Montemanni and Gambardella (2004) and is composed of random graphs.

A graph of type *R-n-c-*$\delta$ has $n$ vertices and an approximate arc density of $\delta$ (i.e., $|A| \sim \delta n(n-1)$). Arcs are set up between random pairs of vertices and interval costs are generated randomly in such a way that $u_{ij} \leq c \;\; \forall (i,j) \in A$ and $0 \leq l_{ij} \leq u_{ij} \;\; \forall (i,j) \in A$.

### 5.1.2 Telecommunication networks

The randomly generated networks of this family have appeared for the first time in Karaşan et al. (2001). They simulate telecommunication networks.

These graphs are *acyclic*, *layered*, and have a small *width*. We remind the reader that an *acyclic* graph is a graph whose arcs do not form any cycle and a *layered* graph is a graph whose vertices can be partitioned into a chain of disjoint subsets, in such a way that the cardinality of each subset is limited by a given constant, called the *width*, and arcs exist only from each subset to the following one in the chain. These graphs are also *complete*, i.e. each node of a layer of the graph is directly connected to every node of the following layer.

A graph of type *K-n-c-d-w* (where $0 < d < 1$) has $n$ vertices; each interval cost $[l_{ij}, u_{ij}]$ is obtained by generating a random number $c_{ij} \in [1, c]$ and by randomly selecting $l_{ij}$ in $[(1-d)c_{ij}, (1+d)c_{ij}]$ and $u_{ij}$ in $[l_{ij}, (1+d)c_{ij}]$; $w$ is finally the *width* of the graph.

For these graphs the origin $s$ is always node 1, while the destination $t$ is always node $n$.

### 5.1.3 Real road networks

The networks belonging to this family represent real road networks, and the interval costs associated with arcs are realistic. The following graphs have been analyzed:

- Sottoceneri: this graph models the main roads of the Sottoceneri region, which is the southern part of Canton Ticino (Switzerland). It has 387 vertices and 1038 arcs and has been provided by *Pina Petroli SA* (http://www.pina.ch);
- Lugano: this graph models the road network of the city of Lugano (Switzerland). It has 576 vertices and 1327 arcs and has been provided by *CRTL (Commissione Regionale dei Trasporti del Luganese)*;
- Stuttgart: this graph models the (aggregated) road network of the Stuttgart area (Germany). It has 2490 vertices and 16153 arcs and has been provided by *PTV (Planung Transport Verkehr) AG* (http://www.ptv.de).
- Padua: this graph models the road network of the city of Padua (Italy). It has 1522 vertices and 2579 arcs and has been provided by *Comune di Padova* (http://www.comune.padova.it).

## 5.2 Results

For each combination network/algorithm considered we report the average computation time (in seconds) over the same 20 instances (with random origins and destinations for random networks and real road networks). In each table, the first column contains the names of the graphs. In the other columns the results achieved by the different algorithms considered are reported. Column $KPY$ contains the results obtained by the algorithm described in Karaşan et al. (2001) (i.e., directly

**Table 1.** Random networks 1. Computation times (in seconds)

| Networks | $KPY$ | $MG$ | $MGD$ | Basic Benders decomp. | Improved Benders decomp. |
|---|---|---|---|---|---|
| R-500-100-0.001 | 0.249 | 0.466 | 0.258 | **0.033** | **0.033** |
| R-500-100-0.010 | 1.668 | 0.789 | 0.465 | 0.444 | **0.310** |
| R-500-100-0.100 | 7.215 | 2.052 | **0.204** | 1.614 | 1.495 |
| R-900-1000-0.010 | 8.051 | 33.089 | 25.099 | 4.830 | **2.678** |
| R-900-1000-0.200 | 63.69 | – | 33.911 | 272.148 | **30.091** |
| R-900-1000-0.500 | 857.155 | – | **48.316** | 862.168 | 140.141 |
| R-900-1000-0.900 | – | – | **185.648** | – | 266.799 |

**Table 2.** Random networks 2. Computation times (in seconds)

| Networks | $KPY$ | $MG$ | $MGD$ | Basic Benders decomp. | Improved Benders decomp. |
|---|---|---|---|---|---|
| R-500-10-0.010 | 2.164 | **0.169** | 0.229 | 0.304 | 0.289 |
| R-500-100-0.010 | 1.668 | 0.789 | 0.465 | 0.444 | **0.310** |
| R-500-1000-0.010 | 2.521 | 1.528 | 1.320 | 0.396 | **0.337** |
| R-900-100-0.010 | 10.819 | 45.091 | 2.255 | 2.120 | **2.107** |
| R-900-1000-0.010 | 8.051 | 33.089 | 25.099 | 4.830 | **2.678** |
| R-900-30000-0.010 | 4.646 | 52.816 | 33.456 | 5.681 | **2.910** |

solving formulation $RSP$), while column $MG$ and $MGD$ summarize the performances of the methods presented in Montemanni and Gambardella (2004) and Montemanni et al. (2004a), respectively. The last two columns are devoted to the new algorithms discussed in this paper. Entries marked with "–" correspond to combinations for which the algorithm failed to find the optimal solution for at least one instance within the time limit of 3600 seconds.

The results of the first set of experiments are summarized in Table 1. The study is about the impact of changes in arc density over random networks. From the experiments it clearly emerges that the Benders decomposition approach is the best one for networks with low arc density, while the branch and bound method $MGD$ is the most promising while the arc density increases. It is however interesting to observe that the Benders decomposition algorithm maintains reasonably good performances over all the problems considered. The situation is different for methods $KPY$ and $MG$, that present really inadequate performances when the arc density becomes high.

The aim of the study presented in Table 2 is to understand how the changes in the maximum interval width affect the performances of the algorithms. The results indicate that algorithms $MG$ and $MGD$ present a substantial performance degradation when the maximum interval width is increased. On the other hand, method $KPY$ seems to benefit from this change (second set of experiments). The

**Table 3.** Random networks 3. Computation times (in seconds)

| Networks | $KPY$ | $MG$ | $MGD$ | Basic Benders decomp. | Improved Benders decomp. |
|---|---|---|---|---|---|
| R-500-100-0.010 | 1.668 | 0.789 | 0.465 | 0.444 | **0.310** |
| R-100-100-0.010 | 0.034 | **0.004** | 0.020 | **0.004** | 0.005 |
| R-900-100-0.010 | 10.819 | 45.091 | 2.255 | 2.120 | **2.107** |
| R-900-1000-0.010 | 8.051 | 33.089 | 25.099 | 4.830 | **2.678** |
| R-1600-1000-0.010 | 73.292 | 428.654 | 87.751 | 41.428 | **9.553** |
| R-2500-1000-0.010 | 159.111 | 844.545 | 245.385 | 286.552 | **56.114** |
| R-4000-1000-0.010 | 697.271 | – | – | – | **211.181** |

**Table 4.** Telecommunication networks. Computation times (in seconds)

| Networks | $KPY$ | $MG$ | $MGD$ | Basic Benders decomp. | Improved Benders decomp. |
|---|---|---|---|---|---|
| K-30-20-0.9-2 | **0.007** | 0.015 | 0.020 | 0.019 | 0.016 |
| K-60-20-0.9-2 | **0.038** | 5.008 | 3.047 | 0.632 | 0.714 |
| K-122-20-0.9-5 | 0.774 | 32.3547 | – | 1.124 | **0.586** |
| K-152-20-0.9-5 | **1.799** | – | – | 11.219 | 10.823 |

Benders decomposition approach is almost not affected by changes in the parameter under investigation. It proves also to be the best method over the problems reported in the table (that however have low arc density).

The results of our last set of experiments on random networks are reported in Table 3. The aim here is to understand how the methods scale up when the number of nodes is increased. It turns out that the new approach based on Benders decomposition scales up better. Among the other methods, $KPY$ is the only one able to solve all the problems considered, notwithstanding its performance is clearly worse than those of the novel algorithm presented in this paper.

The analysis of Table 4 suggests that, for the particular telecommunication networks considered, approach $KPY$ clearly dominates the others. It is however interesting to observe that Benders decomposition is the only algorithm, among the ones here examined, to maintain acceptable performances over all the problems reported in the table (that have low arc density by definition), being also the fastest approach on one problem (*K-122-20-0.9-5*). On the other hand, algorithms $MG$ and $MGD$ do not seem to scale up efficiently when the dimension of the networks increases.

Experiments on real road networks (see Table 5) indicate that the Benders decomposition approach is the most appropriate for this family of graphs. It is the fastest one on two of the four problems. Good performances on three of the

**Table 5.** Road networks. Computation times (in seconds)

| Networks | $KPY$ | $MG$ | $MGD$ | Basic Benders decomp. | Improved Benders decomp. |
|---|---|---|---|---|---|
| Sottoceneri | 0.096 | 0.078 | 0.073 | **0.009** | 0.073 |
| Lugano | 1.299 | 0.191 | **0.117** | 0.141 | 0.137 |
| Stuttgart | 9.648 | 3.129 | **1.752** | 5.952 | 4.906 |
| Padua | 21.138 | 73.785 | – | 19.499 | **17.648** |

problems are achieved also by method $MGD$, but it fails (within 3600 seconds) to solve some of the instances over the Padua network.

From the experiments we can conclude that an algorithm able to clearly dominate the others does not exist. Moreover, the choice of the most appropriate approach is strictly connected with the characteristics of the problem to be solved. In fact, we observed that the ranking based method $MG$ is the less competitive of the pool, and should be consequently avoided in general. On the other hand, the mixed integer programming based algorithm $KPY$ should be used for networks of type $K$, i.e. for a particular family of telecommunication networks. For general problems the choice should be between the remaining two algorithms, and should be driven by the arc density of the network under investigation. $MGD$ is the most promising method for graphs with a high arc density, while Benders decomposition is clearly the fastest approach when arc density is low.

It is however important to observe that the new algorithm based on Benders decomposition algorithm is, in its improved version, the only one able to obtain reasonable performances over all the problems considered, proving to be the most robust of the pool.

The results also suggest that the improved Benders decomposition algorithm described in Sect. 4.3 guarantees a substantial improvement over the basic Benders decomposition method (see Sect. 4.2) in terms of average execution time. It is however interesting to observe that on some problems (namely *R-100-100-0.010*, *K-60-20-0.9-2* and *Sottoceneri*) the improved algorithm is slower than the basic approach. This happens because the initial cuts obtained during the preamble are dominated by other cuts, and consequently the algorithm does not take advantage of them, while, on the contrary, it pays the extra complexity introduced by handling them within integer problems $M^\tau$.

## 6 Conclusion

Benders decomposition has been applied to the robust shortest path problem with interval data. In particular, two versions of the resulting algorithm have been presented.

Computational results on a wide range of benchmarks have also been presented. The experimental study suggests that the new Benders decomposition algorithm is able to outperform state-of-the-art methods on many classes of networks. It is also able to obtain reasonably good results on all the problems considered, proving to be particularly robust.

Finally, a guideline for the selection of the most promising algorithm, given a network to be investigated, has been presented.

## References

Ahuja RK, Magnanti TL, Orlin JB (1993) *Network flows: Theory, algorithms and applications*. Prentice Hall, Englewood Cliffs, NJ, USA

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

Benders JF (1962) Partitioning procedures for solving mixed integer variables programming problems. *Numerische Mathematik* 4: 238–252

Cordeau J-F, Soumis F, Desrosiers J (2000) A Benders decomposition approach for the locomotive and car assignment problem. *Transportation Science* 34: 133–149

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

Dias LC, Clímaco JN (2000) Shortest path problems with partial information: models and algorithms for detecting dominance. *European Journal of Operational Research* 121: 16–31

Dijkstra EW (1959) A note on two problems in connection with graphs. *Numerische Mathematik* 1: 269–271

Ford LR, Fulkerson DR (1962) *Flows in networks*. Princeton University Press, Princeton, NJ, USA

Geoffrion AM (1972) Generalized Benders decomposition. *Journal of Optimization Theory and Applications* 10(4): 237–260

Geoffrion AM, Graves GW (1974) Multicommodity distribution system design by Benders decomposition. *Management Science* 20(5): 822–844

Karaşan OE, Pinar MÇ, Yaman H (2001) The robust shortest path problem with interval data. Bilkent University, Ankara, Turkey

Kouvelis P, Yu G (1997) *Robust Discrete Optimization and its applications*. Kluwer Academic Publishers, Norwell, MA, USA

Magnanti TL, Wong RT (1981) Accelerating Benders decomposition: algorithmic enhancement and model selection criteria. *Operations Research* 29(3): 464–483

Magnanti TL, Mirchandani R, Wong RT (1986) Tailoring Benders decomposition for uncapacitated network design. *Mathematical Programming Study* 26: 112–154

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

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

Montemanni R, Gambardella LM, Donati AV (2004a) A branch and bound algorithm for the robust shortest path problem with interval data. *Operations Research Letters* 32(3): 225–232

Montemanni R, Gambardella LM, Donati AV (2004b) A comparison of two new exact algorithms for the robust shortest path problem. Proceedings of TRISTAN V – The Fifth Triennial Symposium on Transportation Analysis

Richardson R (1976) An optimization approach to routing aircraft. *Transportation Science* 10: 52–71

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