# A Benders decomposition approach for the robust shortest path problem with interval data*

R. Montemanni, L.M. Gambardella

Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA)
Galleria 2, CH-6928 Manno-Lugano, Switzerland
Phone: +41 91 610 8671, fax: +41 91 610 8661
Email: {roberto, luca}@idsia.ch

**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.

A Benders decomposition approach for this problem is presented.

**Keywords:** Shortest path problem, robust optimization, interval data, Benders decomposition.

## 1 Introduction

When transportation and telecommunications problems are modelled in mathematical terms, a network is usually represented as a weighted digraph, where arcs are associated with connections and costs represent travel times or transmission delays.

Unfortunately in the reality it is not easy to estimate arc costs exactly, since they depend on many factors which are difficult to predict. For this reason the classic fixed cost model may be inadequate. To overcome this problem more complex frameworks have been studied. In particular a model where an interval of values, representing a range of possible real costs, is associated with each arc has been proposed. It is the *interval data model*, which is considered in this paper and will be described in details in Section 2. Having adopted this model to represent reality, a criterion to drive optimization has to be chosen. We use the *robust deviation criterion* (sometimes referred to as *relative robustness criterion*). This criterion was discussed in Kouvelis and Yu [4], a book entirely devoted to robust discrete optimization.

A *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. Zieliński [8] proved that the robust deviation shortest path problem is $\mathcal{NP}$-hard.

Karaşan et al. [3] proposed a mixed integer programming formulation for the problem based on an important theoretical result (see Theorem 1). Other exact algorithms have been discussed in Montemanni and Gambardella [6] and Montemanni et al. [7].

## 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 $t \in V$. An interval $[l_{ij}, u_{ij}]$, with $0 < l_{ij} \leq u_{ij}$,

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

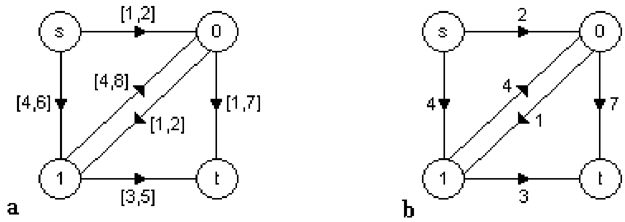


Figure 1: Digraph with interval costs (a) and scenario induced on it by $p = \{s, 0, t\}$ (b).

is associated with each arc $(i, j) \in A$. Intervals represent ranges of possible costs (travel times). An example of interval graph is given in Figure 1(a).

We can formally describe the robust deviation shortest path problem with interval data through 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 deviation 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 deviation shortest path (*robust shortest path* for short) is a path which guarantees reasonably good performance 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 methods developed so far.

**Theorem 1 (Karaşan et al. [3])** *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.*

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.

In the remainder of this paper we will refer to the scenario $r_p$, derived from path $p$ as described in Theorem 1, as the scenario *induced* by path $p$. We will also refer to the cost $\sum_{(i,j)\in p} u_{ij}$ minus the cost of a shortest path of the scenario $r_p$, as the *robustness cost* of $p$. Figure 1(b) depicts the scenario induced by path $p = \{s, 0, t\}$ on the graph of Figure 1(a). The robustness cost of $p$ is in this case $(2 + 7) - (2 + 1 + 3) = 3$.

# 3 Mixed integer programming formulation

Karaşan et al. [3] derived a mixed integer programming formulation for the problem, based on Theorem 1. In this formulation, $y$ 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. $x_j$ is the shortest distance from node $s$ to node $j$. $x_t$ is 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 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.

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

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

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

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

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

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

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

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

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) prevents an unbounded solution. Constraints (7) and (8) define variables domains.

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

# 4 Benders decomposition

Benders partitioning method was originally proposed in 1962 in Benders [1]. It was developed to solve mixed integer programming problems. In this section we apply Benders decomposition to the interval data version of the robust shortest path problem.

## 4.1 Reformulation of $RSP$

By adding a new free variable $z$, dualizing back the problem defined by $x$ variables and working a bit on the constraints, formulation $RSP$ can be reformulated as follows, obtaining the so-called *master problem* $M$:

$$(M) \qquad \min z \tag{9}$$

$$\text{s.t. } z \geq \sum_{(i,j)\in A} u_{ij} y_{ij} - \sum_{(i,j)\in A} (l_{ij} + (u_{ij} - l_{ij})\, y_{ij}) w_{ij} \qquad \forall w \in P_R \tag{10}$$

$$\sum_{(s,k)\in A} y_{sk} - \sum_{(i,s)\in A} y_{is} = 1 \tag{11}$$

$$\sum_{(t,k)\in A} y_{tk} - \sum_{(i,t)\in A} y_{it} = -1 \tag{12}$$

$$\sum_{(j,k)\in A} y_{jk} - \sum_{(i,j)\in A} y_{ij} = 0 \qquad \forall j \in V \backslash \{s,t\} \tag{13}$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{14}$$

$$z \in \mathcal{R} \tag{15}$$

Where set $P_R$ contains the extreme points of the polytope of the flow formulation of the classic shortest path problem (see Magnanti and Wolsey [5]). In particular the extreme points are defined as follows: $w_{ij} = 1$ means that arc $(i, j)$ is on the path under consideration; $w_{ij} = 0$ otherwise.

The master problem is in a suitable form for applying the Benders decomposition algorithm.

### 4.2 The algorithm

It is impossible to deal with all the constraints (10) for non toy-size problems, so a technique to deal with them is required. The algorithm is then based on an iterative mechanism, used to generate the useful elements of set $P_R$. 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 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 $M$ obtained by replacing $P_R$ with $P_R^\tau$, i.e. by substituting constraints (10) with the following ones:

$$z \geq \sum_{(i,j)\in A} u_{ij} y_{ij} - \sum_{(i,j)\in A} (l_{ij} + (u_{ij} - l_{ij})\, y_{ij}) w_{ij} \quad \forall w \in P_R^\tau \tag{16}$$

  Let $(z^\tau, y^\tau)$ be an optimal solution of $M^\tau$.

  Solve the shortest path problem in the scenario induced by variables $y^\tau$. The algorithm described in Dijkstra [2] has been adopted in our implementation.

  Let $z_{UB}(y^\tau)$ be the cost of the shortest path in the scenario induced by variables $y^\tau$ and set $w_{ij}^\tau = 1$ if arc $(i, j)$ is on this shortest path, $w_{ij}^\tau = 0$ otherwise.

  It is possible now to observe (see Benders [1]) that $z_{UB}(y^\tau)$ and $z^\tau$ are respectively an upper bound and a lower bound of the optimal cost of the original problem $RSP$.

  If $z_{UB}(y^\tau) \leq z^\tau$ then the optimal solution of $RSP$ has been found, **stop**.

  Set $P_R^{\tau+1} := P_R^\tau \cup \{w^\tau\}$, $\tau := \tau + 1$ and **repeat the Main step**.

## 5 Computational experiments

For all the test reported ILOG CPLEX 6.0 (http://www.cplex.com) has been used to solve linear and mixed integer programs. All the experiments have been carried out on an Intel Pentium 4 1.5GHz / 256 MB computer.

### 5.1 Networks

Experiments have been carried out on the following families of networks:

- **Random networks:** This family of networks has been originally proposed in Montemanni and Gambardella [6] 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 $0 \leq l_{ij} \leq u_{ij} \;\; \forall (i,j) \in A$.

Table 1: Computation times (in seconds).

| Networks | Karaşan et al. [3] | Montemanni and Gambardella [6] | Montemanni et al. [7] | Benders decomposition |
|---|---|---|---|---|
| R-500-100-0.01 | 1.668 | 0.789 | 0.465 | **0.310** |
| R-500-100-0.1 | 7.215 | 2.052 | **0.204** | 1.495 |
| R-100-100-0.01 | 0.034 | **0.004** | 0.020 | **0.004** |
| R-500-10-0.01 | 2.164 | **0.169** | 0.229 | 0.289 |
| R-500-100-0.001 | 0.249 | 0.466 | 0.258 | **0.033** |
| R-500-1000-0.01 | 2.521 | 1.528 | 1.320 | **0.337** |
| R-900-100-0.01 | 10.819 | 45.091 | 2.255 | **2.107** |
| K-30-20-0.9-2 | **0.007** | 0.015 | 0.020 | 0.016 |
| K-60-20-0.9-2 | **0.038** | 5.008 | 3.047 | 0.632 |
| K-122-20-0.9-5 | 0.774 | 32.3547 | n.a. | **0.586** |
| K-152-20-0.9-5 | **1.799** | n.a. | n.a. | 10.823 |
| Sottoceneri | 0.096 | 0.078 | 0.073 | **0.009** |
| Lugano | n.a. | 0.191 | **0.117** | 0.137 |
| Stuttgart | 9.648 | 3.129 | **1.752** | 4.906 |

- **Telecommunication networks:** The randomly generated networks of this family have appeared for the first time in Karaşan et al. [3]. They simulate telecommunication networks and are *acyclic*, *layered*, and have a small *width* graphs[1]. 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 $n$.

- **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[2]: main roads of the Sottoceneri region (southern part of Canton Ticino - Switzerland). The graph has 387 vertices and 1038 arcs;
  - Lugano[3]: road network of the city of Lugano (Switzerland). The graph has 576 vertices and 1327 arcs;
  - Stuttgart[4]: (aggregated) road network of the Stuttgart area (Germany). The graph has 2490 vertices and 16153 arcs.

## 5.2 Results

For each network considered we report, for each algorithm, the average computation time (in seconds) over 20 instances (with random origins and destinations for random networks and real road networks). The first column contains the names of the graphs. In the other columns the results achieved by the different algorithms considered - they represent the state of the art - are reported. Entries marked with "n.a." correspond to combinations for which no result is available.

[1]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.

[2]Network provided by *Pina Petroli SA* (http://www.pina.ch).

[3]Network provided by *CRTL (Commissione Regionale dei Trasporti del Luganese).*

[4]Network provided by *PTV (Planung Transport Verkehr) AG* (http://www.ptv.de).

Table 1 shows that the new Benders decomposition approach is comparable with the state-of-the-art algorithms recently appeared in the literature. In particular the new method is faster on many of the random networks considered. On telecommunications networks the best method remains the one proposed in Karaşan et al. [3], but it is interesting to observe that Benders decomposition is able to close the gap (and even being the fastest on *K-122-20-0.9-5*), outperforming the methods described in Montemanni and Gambardella [6] and Montemanni et al. [7]. The results of Benders decomposition on real road networks are finally in line with those of the other algorithms. Also for this family of problems the best known result for a problem (*Sottoceneri*) is reached.

## References


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

[2] E.W. Dijkstra. A note on two problems in connection with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[3] O.E. Karaşan, M.Ç. Pinar, and H. Yaman. The robust shortest path problem with interval data. *Computers and Operations Research*, 2002.

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

[5] 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.

[6] 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.

[7] 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.

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