RAIRO-Oper. Res. 45 (2011) 101–129
DOI: 10.1051/ro/2011111

RAIRO Operations Research
www.rairo-ro.org

MINMAX REGRET COMBINATORIAL OPTIMIZATION PROBLEMS: AN ALGORITHMIC PERSPECTIVE*

Alfredo Candia-Véjar[1], Eduardo Álvarez-Miranda[1, 2] and Nelson Maculan[3]

Abstract. Uncertainty in optimization is not a new ingredient. Diverse models considering uncertainty have been developed over the last 40 years. In our paper we essentially discuss a particular uncertainty model associated with combinatorial optimization problems, developed in the 90's and broadly studied in the past years. This approach named *minmax regret* (in particular our emphasis is on the robust deviation criteria) is different from the classical approach for handling uncertainty, *stochastic approach*, where uncertainty is modeled by assumed probability distributions over the space of all possible scenarios and the objective is to find a solution with good probabilistic performance. In the minmax regret (MMR) approach, the set of all possible scenarios is described deterministically, and the search is for a solution that performs reasonably well for all scenarios, *i.e.*, that has the best worst-case performance. In this paper we discuss the computational complexity of some classic combinatorial optimization problems using MMR approach, analyze the design of several algorithms for these problems, suggest the study of some specific research problems in this attractive area, and also discuss some applications using this model.

Keywords. Minmax regret model, combinatorial optimization, exact algorithms and heuristics, robust optimization.

Mathematics Subject Classification. 90C11, 90C27, 90C35.

Received March 30, 2009. Accepted May 19, 2011.

* *E. Álvarez-Miranda was partially supported by P^4 Project Grant Center for Research and Applications in Plasma Physics and Pulsed Power Technology PBCT-Chile-ACT 26, CONICYT; A. Candia-Véjar was partially supported Dirección de Programas de Investigación, Universidad de Talca, Chile.*

[1] Modeling and Industrial Management Department, Universidad de Talca, Curicó, Chile. acandia@utalca.cl

[2] Operations Management Master Program, Universidad de Talca, Curicó, Chile.

[3] COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil.

Article published by EDP Sciences

1. Introduction

Uncertainty affects many systems concerning decision making. Uncertainty is present in production costs, prices, labor and other parameters, therefore it is convenient to include it as a property of the technological and business environment. Historically, uncertainty has been considered in mathematical models for decision making. One example is the stochastic optimization approach which recognizes the presence of multiple scenarios that might occur in the future. As part of the decision model, explicit information on the probability value with which these instances might be realized is necessary. Therefore, the decision model will typically generate a solution that optimizes an expected performance measure and a stochastically optimal decision is achieved. The main drawback of this approach (and also the one of the field of deterministic optimization) is its inability to recognize that associated with every decision is a whole distribution of outcomes depending on what data scenario is actually carried out. Thus any approach evaluating decisions using only one data scenario, either the expected or the most likely, could fail; the performance of a decision across all possible scenarios is important to study. For more details about the progress of developed heuristics for stochastic combinatorial optimization, see Bianchi *et al.* [22].

In our paper we discuss decision environments where there is a lack of complete knowledge about the random state of nature. This condition is in the heart of the robustness approach. Here, the idea is to produce decisions that will have a reasonable objective value under any possible scenario for the decision model in a pre-specified planning horizon. In this context high probability events or low probability events are equally important because this approach is appropiate in situations where the consideration of the worst case is critical. For example, the design of a bridge must consider three phenomena which might behave in different magnitudes under certain circumstances; truck and car transit, water flow and earth tremors or even earthquakes; a critical element like this must be able to stand the worst case magnitude of these variables.

Different minmax criterion can be used to select among robust decisions. Two known criterion are *minmax*, also called *absolute robust criteria*, where the robust decision is the one that minimizes the maximum across all scenarios and which is an overly conservative criteria; and *minmax regret*, also called *robust deviation*, which is the difference (absolute or percentage) between the cost of a solution in a given scenario and the cost of the optimal solution for that scenario. Regret might be considered as an opportunity loss, because it represents the difference between the performance of a given solution, and the performance of the solution that would have been chosen if one had known the state-of-nature *a priori*.

An important effort has been focalized on the use of MMR for solving real world problems, where uncertainty represents a central feature. As example of this, interesting papers were studied. Chang and Davila [25] combined the technique of grey mathematical programming, in which all or part of the input parameters are represented by interval numbers, with a MMR formulation for tackling a concrete complex solid waste management problem. The same authors, in another paper [26], attacked a related problem with the use of a MIP formulation for a MMR model. They applied a minimax regret optimization methodology for improving solid waste management strategies in the Lower Rio Grande Valley (TX, USA) in accordance with a variety of cost and benefit dimensions associated with expansion or construction interests in the solid waste management system. Loulou and Kanudia, in [48], use a MMR formulation for a large scale problem related to the definition of strategies for greenhouse gas abatement in Canada, and interesting results were obtained when the MMR solutions were compared with a stochastic formulation for the same problem. Kazakci *et al.* [46] developed an hybrid model, containing interval linear programming (ILP) and MMR criteria for managing the surface allocation in farms destined for biofuel production in France, in this application diverse factors were considered affected by uncertainty, such as for example, crop yields, agricultural policy decisions, climatic risks and energy demand, among others.

It is important to note that for an environment of uncertainty like the one considered here, there are other models besides minmax and minmax regret. A first important example is the Bertsimas and Sim approach [20,21] developed for general combinatorial optimization (CO) problems, which represents an interesting perspective both in formulation and algorithmic development. A second example is a deterministic risk management model [27–29], in this formulation the behavior of the decision maker influences the robustness of the solution and it is essentially focused on shortest path problem and minimum spanning tree. A third example is presented by Hites *et al.* [40], where they discuss the applicability of classic multicriteria concepts to the robustness framework including the MMR model. Another example is a parameter space-based approach [39], that is an interesting alternative for handling uncertainty especially for facility location problems due the concepts behind the optimality criteria. Finally, in [9] a two-stage model with MILP formulation, including data uncertainty and MMR criteria is presented and an algorithm is proposed which efficiently identifies the robust first-stage decisions under robust deviation of the two-stage MILP formulated problem.

Data uncertainty can be structured using discrete scenarios or interval data on the problem parameters. The use of discrete scenarios allows better representation of environment-specific knowledge but interval data can simplify the data collection and even the analysis process (leading to models with stronger structural properties).

A review of some robust combinatorial problems is presented in this paper by using robust deviation criteria and interval data. Some general results are shown for a class of known problems including minimum spanning tree (MST) and shortest path (SP). Algorithnic development is discussed for MST and other important problems such as traveling salesman problem (TSP) and assignment problem (AP).

The paper is organized as follows. In Section 2, notation and problem statement are presented, then a very important structural property is discussed an finally a general mixed integer linear programming formulation is described. In Section 3, we discuss the computational complexity of some classic CO problems in their robust versions. In Section 4 we discuss exact algorithms and heuristics for solving some important interval data minmax regret problems. Finally, in Section 5, we propose some avenues for future work and conclusions.

2. Definitions and a fundamental property

2.1. Definitions

The following notation is from Averbakh and Lebedev [17]. First of all, we define a generic CO problem.

CO Problem: Let Φ be a finite set of feasible solutions and F a function defined on Φ with the property that the optimum value of the problem: $\min\{F(X) : X \in \Phi\}$, always exists.

Suppose that there is uncertainty in the objective function such that it is only known that F is a member of a family of functions $\{F_S(\cdot), S \in \Gamma\}$ for some set of scenarios. The set Φ is scenario-independent. Let F_S^* denote the optimum value for the following problem:

OPT(s) Problem: $\min\{F_S(X) : X \in \Phi\}$. We note that this problem, for any scenario $S \in \Gamma$ is reduced to solve a classic CO Problem.

For any $X \in \Phi$ and $S \in \Gamma$, the function $R(S, X) = F_S(X) - F_S^*$ is called the *regret* for X under scenario S. For any $X \in \Phi$, the function Z is defined by:

$$Z(X) = \max\{R(S, X) : s \in \Gamma\}, \tag{2.1}$$

and is the *worst-case regret* for X and an optimal scenario S^* producing such a worst-case regret is called *worst-case scenario for* X. It is easy to note that $Z(X) \geq 0$, $\forall X \in \Phi$.

The minmax regret version of the CO problem is:

$$\mathbf{MMR - COProblem} : Z^* = \min\{Z(X) : X \in \Phi\}. \tag{2.2}$$

Note: This MMR version of the robust CO problem is also known as a *robust deviation* problem. Another MMR version of robust CO problems is known as *relative regret* and some results are studied in the paper by Averbakh [13].

In MMR-CO problems, we assume that each scenario s can be considered as a vector in R^m, where m is the number of uncertain parameters in the problem. Traditionally, there are two ways to represent uncertainty: the *discrete scenario* case where Γ is finite and described by explicitly listing all vectors $s \in \Gamma$, and the *interval data* case where, for each numerical parameter, only lower and upper bounds for the value of this parameter are known, and the parameter can take on any value between these bounds, regardless of values taken by other parameters. Thus, in this case, Γ is the Cartesian product of the intervals of uncertainty for the parameters. Intuitively speaking, a scenario can be seen as a snapshot of the problem situation [54].

We give special emphasis to MMR-CO problems where the set of scenarios Γ is defined by real intervals (*MMR-CO problem with interval data*) but some discussion about complexity issues including finite scenarios is also carried out.

A class of problems discussed here come from network optimization and, for this reason, special notation is given to represent uncertainty. Let $G = (V, E)$ be an undirected (directed) connected graph with V being the set of nodes and E being the set of edges (arcs), $|V| = n$, $|E| = m$. Suppose that for every edge (arc) $e \in E$, its cost belongs to a real interval $[c_e^-, c_e^+]$, and there is complete independence between the cost (or length) values taken by any pair of different edges (arcs). The set of scenarios Γ is the Cartesian product of the intervals of uncertainty $[c_e^-, c_e^+]$, $e \in E$.

In the case of problems where other parameters might include uncertainty, such as facility location problems where vertex weight and edge (arc) length could be represented as intervals, we can consider only one set of scenarios containing all the possible ocurrence of weight and length values as in [16], or we can consider different sets of scenarios, one for each parameter.

2.2. A Fundamental property for maximal regret

For a *MMR-CO problem with interval data* we consider $E = \{e_1, e_2, \ldots, e_n\}$ be a finite ground set and Φ is the set of feasible solutions. For every element $e \in E$ there is a closed interval $[c_e^-, c_e^+]$, $c_e^- \geq 0$, $\forall e \in E$. A particular realization of the costs $S = (c_e^S)_{e \in E}$, $\forall e \in E$ is a *scenario*. Then, $\Gamma = \times_{e \in E} [c_e^-, c_e^+]$.

Some special scenarios are important. Two extremes scenarios in which all the costs take the extreme values c_e^- or c_e^+ are utilized oftenly. If $A \subseteq E$ we will denote S_A^+ the scenario in which the elements $e \in A$ have weights c_e^+ and all the other elements have weights c_e^-. The scenario S_A^- is defined analogously.

Proposition 2.1. [42] *The scenario S_X^+ is the worst case scenario for solution X.*

From the above proposition we note it is possible to express the maximal regret of a given solution X as follows:

$$Z(X) = F_{S_X^+}(X) - F^*_{S_X^+}. \tag{2.3}$$

From this formula for $Z(X)$ it is important to note that if the underlying deterministic CO problem is polynomially solvable, then the maximal regret of a given solution X can be computed in polynomial time since $F^*_{S_X^+}$ can be computed in polynomial time.

2.3. A general MIP for MMR-CO problems with interval data

We present now a general approach for solving MMR-CO problems based on a mixed integer programming formulation (MIP) [42]. Under some assumptions it is possible to formulate a MIP model with linear constraints and linear objective function for MIP. One remarkable characteristic of this model is that it involves a polynomial number of variables and constraints, and thus it would be possible to solve it with a commercial package.

A binary variable $x_i \in \{0,1\}$ $\forall e_i \in E$ is defined saying whether element e_i is a part of the constructed optimal solution. A *characteristic vector* of a given subset of elements $A \subseteq E$ is a binary vector $\mathbf{x} = (x_i)_{i=1}^n$ such that $x_i = 1$ if and only if $e_i \in A$. Let us associate with the set of feasible solutions Φ a set of binary vectors $ch(\Phi) \subseteq \{0,1\}^n$, which satisfies two conditions

(1) if $\mathbf{x}$ is a characteristic vector of a feasible solution $X \in \Phi$ then $\mathbf{x} \in ch(\Phi)$, and
(2) if $\mathbf{y}$ is a characteristic vector of a subset $Y \subseteq E$ and $\mathbf{y} \in ch(\Phi)$ then there exists $X \subseteq Y$ such that $X \in \Phi$.

Then $ch(\Phi)$ contains all characteristic vectors of the feasible solutions in Φ and it may also contain a characteristic vector of a subset $Y \subseteq E$ such that $Y \notin \Phi$. However, in this case Y must contain a feasible solution X such that $X \in \Phi$.

It is assumed that $ch(\Phi)$ can be described by a set of linear constraints of the form:

$$ch(\Phi) = \left\{ \mathbf{x} \in \{0,1\}^n : \mathbf{A} \left[\mathbf{x}, \mathbf{x}'\right]^{\mathrm{T}} = \mathbf{b} \right\}, \tag{2.4}$$

where $\mathbf{x}$' is a vector of auxiliary variables (if required), $\mathbf{A}$ is a matrix and $\mathbf{b}$ is a vector of fixed coefficients. At first, it is supossed that matrix $\mathbf{A}$ is *totally unimodular* (that is, every square submatrix of $\mathbf{A}$ has determinant 0, +1 or –1). This is usually the case when the deterministic problem is polynomially solvable. The following function is defined:

$$\phi(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} (c_i^+ x_i + c_i^- (1 - x_i)) y_i. \tag{2.5}$$

Proposition 2.2. *The MMR-CO problem can be expressed as the following mathematical programming problem:*

$$\min_{\mathbf{x} \in ch(\Phi)} \left\{ \textstyle\sum_{i=1}^n c_i^+ x_i - \min_{\mathbf{y} \in ch(\Phi)} \phi(\mathbf{x}, \mathbf{y}) \right\}. \tag{2.6}$$

Clearly this formulation is not a linear program. However, if matrix $\mathbf{A}$ is totally unimodular then it can be tranformed into a linear problem with binary variables. To prove this important result, we first fix $\mathbf{x}$, and then consider the subproblem:

$$\min_{y \in ch(\Phi)} \phi(x, y), \tag{2.7}$$

which can be rewritten as follows:

$$\begin{aligned}
&\min \sum_{i=1}^{n} \left(c_i^+ x_i + c_i^- \left(1 - x_i\right)\right) y_i \\
&\mathbf{A}\left[\mathbf{y}, \mathbf{y}'\right]^T = \mathbf{b} \\
&y_i \in \{0, 1\} \quad \forall i = 1, \ldots, n.
\end{aligned} \tag{2.8}$$

From the total unimodularity of $\mathbf{A}$ integrality constraints for $\mathbf{y}$ can be relaxed to linear constraints obtaining:

$$\begin{aligned}
&\min \sum_{i=1}^{n} \left(c_i^+ x_i + c_i^- \left(1 - x_i\right)\right) y_i \\
&\mathbf{A}\left[\mathbf{y}, \mathbf{y}'\right]^T = \mathbf{b} \\
&0 \le y_i \le 1 \quad \forall i = 1, \ldots, n
\end{aligned} \tag{2.9}$$

For a fixed $\mathbf{x}$ problem, (2.9) is a linear program. Then it is possible to construct a dual problem to (2.9) with the vector of dual variables λ associated with every constraint. Denote by $\phi^*(\mathbf{x}, \lambda)$ the objective of the dual and by $\Lambda(\Phi)$ the set of feasible dual vectors. Then the dual problem is linear with respect to both $\mathbf{x}$ and $\mathbf{y}$. Moreover, strong duality theorem implies:

$$\min_{\mathbf{y} \in ch(\Phi)} \phi(\mathbf{x}, \mathbf{y}) = \max_{\lambda \in \Lambda(\Phi)} \phi^*(\mathbf{x}, \lambda). \tag{2.10}$$

Using above results is obtained:

$$\min_{\mathbf{y} \in ch(\Phi)} \phi(\mathbf{x}, \mathbf{y}) = \min_{\mathbf{x} \in ch(\Phi)} \min_{\lambda \in \Lambda(\Phi)} \left\{ \sum_{i=1}^{n} c_i^+ x_i - \phi^*(\mathbf{x}, \lambda) \right\}. \tag{2.11}$$

Problem (2.11) is a mixed integer linear programming problem with binary variables (x_i) and real variables (λ_i) and then it can be solved by standard software. Moreover, relaxing constraints $x_i \in \{0, 1\}$ with $0 \le x_i \le 1$, one obtains a problem which is polynomially solvable. The solution of this problem gives a lower bound for the maximal regret of the original MMR-CO problem.

3. Some complexity results on MMR-CO problems

Results about the computational status of MMR-CO problems and complexity of their algorithms are discussed in the classic book by Kouvelis and Yu [47]. They postulate in their book that MMR versions of most classic CO problems are NP-Hard in the case of discrete scenarios. Furthermore, little is known about complexity of interval data MMR-CO problems.

Aissi *et al.* [4] in a extensive survey paper have described the main advances over the last years concerning the complexity of some MMR problems. In our paper we comment on some of these results, however, the emphasis of our work lies in the description of exact algorithms and heuristics for MMR problems. In this sense, our paper is complementary to the former paper.

The following MMR-CO problems (very studied in their deterministic version) are proved to be NP-Hard [47] when the case of discrete scenarios is considered.

- **Assignment problem:** NP-Hard even if $|\Gamma| = 2$.
- **Shortest s-t path problem:** NP-Hard even in layered networks of width 2 and with only two scenarios.
- **Minimum spanning tree problem**: NP-Hard even if $|\Gamma| = 2$.
- **Resource allocation problem:** NP-Hard even if $|\Gamma| = 2$.
- **Knapsack problem** (in absolute robust version): Strongly NP-Hard for unbounded scenario set Γ.

Several problems are also analyzed in the book [47], where only a few MMR-CO problems have polynomial time algorithms. In the last 10 years a number of papers discussing the computational status and algorithms for MMR-CO problems using interval data have been published. Some of them are discussed next.

3.1. MMR select (p/m) (MMR-SELECT)

Averbakh [10] presented a polynomial algorithm for the MMR problem of selecting p elements of minimum total weight out of a set of m elements with uncertainty in the weights of the elements. This is the first known example of a robust CO problem that is NP-Hard in the case of discrete scenario uncertainty, but which is polynomially solvable in the case of the interval representation of uncertainty. More precisely, Averbakh [10] showed that MMR-SELECT is NP-Hard even if $|\Gamma| = 2$; however, for the case of interval uncertainty he presented a polynomial algorithm $O((\min\{p, m-p\})^2 \cdot m)$. He also commented that the algorithm posseses practical applications, since the problem represents a family of resource allocation problems and a class of scheduling problems. Conde [30] improved the former result, presenting an algorithm with complexity $O\left((\min\{p, m-p\}) \cdot m\right)$.

3.2. MMR s-t path problems (MMR-s-tP)

Averbakh and Lebedev [17] proved that MMR-s-tP is NP-Hard on undirected networks even if the bounds of all intervals of uncertainty belong to $\{0, 1\}$, it is also NP-Hard on directed networks even if the networks are acyclic, have layered structure, and the bounds of intervals of uncertainty belong to $\{0, 1\}$. It is polynomial if the number of edges with uncertain lengths is fixed or is bounded by the logarithm of a polynomial function of the total number of edges. The NP-Hardness of MMR-s-tP was independently proved by Zieliński [67].

Kasperski and Zieliński [44] studied MMR-s-tP for a sparse class of directed graphs: edge series-parallel multidigraphs. They commented that this class of digraphs appear in several applications. Also, several problems which are NP-Hard for general graphs can be solved in polynomial-time in series-parallel graphs.

They proved that MMR-s-tP remains NP-Hard even if a graph is restricted to being an edge series-parallel digraph with at most vertex degree equal to three.

3.3. MMR spanning tree problem (MMR-ST)

Averbakh and Lebedev [17] proved that MMR-ST is NP-Hard even if the intervals of uncertainty are equal to $[0, 1]$ and polynomial if the number of edges with uncertain lengths is fixed or is bounded by the logarithm of a polynomial function of the total number of edges.

In the same paper, they showed an interesting general result: any interval data MMR minisum network optimization problem is polynomially solvable if the number of edges with uncertain lengths is fixed or is bounded by the logarithm of a polynomial function of the total number of edges. This result is due to the special structure created by interval data representation of uncertainty.

The NP-Hardness of MMR-ST was independently proved by Aron and Van Hentenryck [8]. They also proved that this problem remains NP-Hard even if all the cost intervals are $[0, 1]$ and also if the graph is complete.

Salazar-Neumann [64] studied MMR-ST where edge costs are on a compact and convex subset of $\mathbb{R}^n$. Here, the location of the worst and best deviation scenarios for a tree are found. The paper also reports characterizations of strictly strong and non-weak edges for the above-mentioned problem. Hopefully, these results could help to find new algorithms that reduce the time to compute a robust spanning tree in this broad class of problems.

In a related work to MMR-ST, Conde and Candia [36] have shown that the robust version of minimum spanning arborescence problem (MMR-SA) (a standard generalization of MST to directed graphs) is polynomially solvable when the input directed network is acyclic. They presented a greedy linear time algorithm (in the number of arcs) for MMR-SA.

3.4. MMR assignment problem (MMR-A)

Aissi *et al.* [1] showed that minmax and minmax regret assignment problems are strongly NP-Hard when the number of scenarios is not bounded by a constant. In the interval data case, the minmax assignment problem is shown to be polynomial, whereas the minmax regret assignment problem is proved to be, in general, strongly NP-Hard, with some specific polynomial cases. Deineko and Woeginger [38] showed that the discrete MMR-A problem with a fixed number of scenarios, $k \geq 2$, can be transformed into the (standard) discrete minmax assigment problem with a fixed number of scenarios, $k \geq 2$ in polynomial time; moreover, both problems can be transformed into the Exact Perfect Matching Problem in polynomial time.

3.5. MMR linear programming (MMR-LP)

Averbakh and Lebedev [18] closed the open complexity status of MMR-LP. They proved that it is strongly NP-Hard and also they commented that the discrete-scenario MMR-LP is polynomially solvable since it can be written as an LP problem of polynomial size in n, m and Γ. Thus, this is the first known example of an MMR-CO problem that is NP-Hard in the interval data case while being polynomially solvable in the discrete-scenario case. The inverse situation, that is, an example of a MMR-CO problem that is polynomially solvable in the interval data case but is NP-Hard in the discrete-scenario case was discussed above in Section 3.1.

Mausser and Laguna [50–52] studied LP in both, MMR and minimax absolute versions. In those papers general algorithms were proposed for MMR linear programs relying on the fact that for a given solution, in the maximizing regret scenario, each uncertain parameter is set either to its lower or upper interval bound. Snyder [65] discussed some ideas about algorithmic approaches from these papers.

3.6. MMR facility location problems (MMR-FL)

Several papers advocate to the study of models and algorithms for this class of problems; the developed models include center, median and centdian problems, among others, for single or several facilities to be located in different types of networks. Averbakh [11] proves that if edge lengths are represented by intervals, then both 1-median and weighted 1-center problems on a general network are NP-Hard; however, Averbakh and Berman in [15] gave an exact $O\left(mn^2 \log n\right)$ algorithm for the 1-median problem on a general network with interval-uncertain node weights, where m is the number of edges and n the number of nodes. In the case of tree networks, with node or edge location, the same authors presented a new algorithm with complexity $O(n^2)$.

4. Algorithms for some MMR-CO problems

We discuss here different approaches to MMR-CO problems, including exact algorithms, approximation algorithms, heuristics and metaheuristics. An interesting review of these topics until 2006 is presented in the book of Kasperski [42].

4.1. Exact algorithms

4.1.1. *MMR-A*

We discuss here the results presented in [42,63]. At first, we present some notation and the formulation of MMR-A as a network flow problem as is shown in [42]. We are given a bipartite graph $G = (V, E)$ with interval costs defined on the set of edges E, $[c_e^-, c_e^+]$, $e \in E$. For bipartite graphs the set of nodes V can be partitioned in two disjoint sets V_1 and V_2 such that $\mid V_1 \mid=\mid V_2 \mid$ and if $\{i, j\} \in E$

then $i \in V_1$ and $j \in V_2$. An *assignment* is a perfect matching in G that is a subset of edges $B \subseteq E$ in which no two edges share a common node and every node of V is incident to a exactly one edge of B. The solution of MMR-A is an assignment which minimizes the maximal regret.

Let $G = (V_1 \cup V_2, E)$ be a given bipartite graph. G can be transformed in a directed graph $G' = (V_1 \cup V_2, A)$ by replacing every edge $\{i, j\} \in E$, $i \in V_1, j \in V_2$ with arc $(i, j) \in A$. If V_1 represents a set of sources and V_2 represents a set of sinks then it is supossed every source $i \in V_1$ supplies one unit of flow and every sink $j \in V_2$ requires one unit of flow. The arcs of G' represent available shipping links and the problem is to send the flow from the sources to the sinks. The amount of flow that is sent on every arc is assumed to be an integer or equivalently 0 or 1.

Binary variables $x_{ij} \in \{0, 1\}$ for every arc $(i, j) \in A$ are defined. The following constraints define feasible solutions for the netwok flow problem.

$$\begin{aligned}
&\sum_{\{j:(i,j)\in A\}} x_{ij} = 1 \quad \forall i \in V_1 \\
&\sum_{\{i:(i,j)\in A\}} x_{ij} = 1 \quad \forall j \in V_2 \\
&x_{ij} \in \{0, 1\} \quad \forall\, (i, j) \in A.
\end{aligned} \tag{4.1}$$

Then these constraints can be used to describe set $ch(\Phi)$ in MMR-A. It is known that the constraint matrix of the polyhedra defined above is totally unimodular and then the relaxed subproblem $min_{y\in ch(\Phi)}\phi(\mathbf{x}, \mathbf{y})$ can be represented in the following way:

$$\begin{aligned}
&\min \sum_{(i,j)\in A} \left(c^+_{ij} x_{ij} + c^-_{ij}\,(1 - x_{ij})\right) y_{ij} \\
&\sum_{\{j:(i,j)\in A\}} y_{ij} = 1 \quad \forall i \in V_1 \\
&\sum_{\{i:(i,j)\in A\}} y_{ij} = 1 \quad \forall j \in V_2 \\
&y_{ij} \geq 0 \quad \forall\, (i, j) \in A.
\end{aligned} \tag{4.2}$$

The constraints $0 \leq y_{ij} \leq 1$ have been replaced with $y_{ij} \geq 0$ because $y_{ij} \leq 1$ in every optimal solution of (4.2). The dual of (4.2), that is the problem $\max_\lambda \phi^*(x, \lambda)$ is the following:

$$\begin{aligned}
&\max \sum_{i\in V_1} \alpha_i + \sum_{j\in V_2} \beta_j \\
&\alpha_i + \beta_j \leq c^+_{ij} x_{ij} + c^-_{ij}\,(1 - x_{ij}) \quad \forall\, (i, j) \in A.
\end{aligned} \tag{4.3}$$

Using (2.11), (4.1) and (4.3) the following MIP formulation is obtained for MMR-A.

$$\begin{aligned}
&\min \sum_{(i,j)\in A} c^+_{ij} x_{ij} - \sum_{i\in V_1} \alpha_i - \sum_{j\in V_2} \beta_j\\
&\sum_{\{j:(i,j)\in A\}} x_{ij} = 1 \quad \forall i \in V_1\\
&\sum_{\{i:(i,j)\in A\}} x_{ij} = 1 \quad \forall j \in V_2 \qquad\qquad (4.4)\\
&\alpha_i + \beta_j \leq c^+_{ij} x_{ij} + c^-_{ij}\left(1 - x_{ij}\right) \quad \forall\,(i,j) \in A\\
&x_{ij} \in \{0,1\} \quad \forall\,(i,j) \in A.
\end{aligned}$$

We note that Kasperski [42] only used small instances and in these cases CPLEX can solve it in at most 1 minute. From these results we also note CPU time increase very rapidly with the number of nodes. This analysis using the same formulation is also considered next in the other paper of MMR-A.

In a recent paper ([63]), two exact approaches are compared, one is the same shown here (Kasperski [42]), and the other is based on Benders decomposition. For the latter approach, they developed a new linear integer programming formulation which contains a set with an exponential number of constraints. At each iteration of this approach a relaxed formulation with only a small subset of constraints is solved. Then a constraint (Benders cut) that is most strongly violated by the solution is identified and included it in the relaxed formulation of the next iteration. It is well known that the performance of Benders decomposition can be strongly influenced by heuristic strategies of including additional Benders cut. They implemented three different strategies, the first includes some constraints before starting the procedure and it is referred to as the *basic Benders decomposition* (bBD). The second strategy is to introduce more than one new constraint at each iteration of the algorithm. This strategy is likely to reduce the number of iterations but will increase the time required for the master problem and it will be referred to as *Benders decomposition with two cuts* (cBD). The third strategy is based on using the complicated set of constraints within a Branch and cut approach and it will be referred to as *branch and cut benders decomposition* (bcBD).

Beasley instances. The instances are derived from the assignment problem instances available in Beasley's OR-Library ([19]), the numbers of tasks range from 100 to 500 and the cost coefficients (c^P_{uv}) are integers from $[1, 100]$. These instances from the assignment problem are converted in MMR-A instances transforming costs in cost intervals by using a parameter $\beta \in [0, 1]$. The definition of cost intervals allows to generate instances with different levels of uncertainty by modifying β. For each original instance three different values of β were used, $\beta \in \{0.1, 0.25, 0.5\}$. Ten different instances of MMR-A problem were generated for each original instance of the assignment problem and a specific value of β, so 150 instances of MMR-A were generated in total.

McGeoch instances. Eight instances of the classic Assignment problem were initially generated based on two parameters, the number of tasks n, $n_{\min} = 100$ and

$n_{\max} = 500$, and the upper bound $M = 1\,000\,000$ for basic costs. For each classic instance three groups of instances of MMR-A were generated using the parameter $\beta \in \{0.1, 0.25, 0.5\}$. Each group contained 10 instances where uncertainty intervals for costs were generated in the same way as for Beasley instances. 240 instances were generated in total and the initial eight instances of the classic Assignment problem were generated using the generator of McGeoch.

Montemanni instances. A similar procedure to proposed for TSP in [59] for generate interval costs is used. 80 instances were generated in total, ten for each problem size.

Kasperski instances. The instances for MMR-A are generated in the same way described when we discussed the approach by Kasperski [42] above. A total of 80 instances were generated, ten for each problem size.

Computational results. According with the results obtained in their implementation, Pereira and Averbakh [63] concludes that the basic MILP formulation with the use of CPLEX is a reasonably reliable method for solving optimality or almost optimality small instances (up to $n = 100$), and it is the best of the exact methods for moderate size problems (up to $n = 300$), while Benders decomposition approaches run better for larger instances ($n > 300$). We note that Benders decomposition is more efficient for larger instances because it is based on the knowledge of the problem structure. Beasley instances were harder to solve, and for them Benders decomposition approach become more effective than CPLEX starting with smaller problem sizes ($n = 200$). With respect to the different Benders decomposition variants, for small instances, $n \leq 150$, bcBD variant performs better than bBD and cBD, for medium size instances ($n = 300$) all three variants are comparable and for large instances ($n > 300$), all three variants are comparable in terms of the quality of the obtained solutions, but the variant bBD is more successful in obtaining good lower bounds which results in significantly lower reported optimality gaps for Beasley, Montemanni and Kasperski instances. The authors also discussed the relation between the difficulty of the problems and the degree of uncertainty.

In our opinion, one interesting conclusion from the computational experiments realized with these algorithms for MMR-A is the high difficulty to solving McGeoch and Beasley instances; for these groups all the algorithms found high gaps and then stronger approaches for solving these instances are needed.

4.1.2. *MMR-ST*

Yaman *et al.* [66] presented a mixed integer linear programming formulation of the problem, and defined some characterizations leading to efficient preprocessing techniques before applying a CPLEX solver to the MILP formulation; computational experience was also reported illustrating the efficiency of the preprocessing procedures. MMR-ST is defined on a connected graph $G = (V, E)$, with $\mid V \mid= n$, and interval costs $[c_e^-, c_e^+]$, $\forall e \in E$.

Mathematical Programming Formulation. Two mixed linear integer programming formulations for MST were used in his approach for solving MMR-ST.

These formulations for MST appear in the book of Magnanti and Wolsey [49]; in the first formulation are defined variables x_e to indicate if an edge e in E to be available to carry any flow. In this single commodity model, one node (node 1) is defined as a source node and one unit of flow must be sent from 1 to every other node, A is the set of arcs formed with the set of nodes V and f_{ij} is the flow on arc (i, j). The model (P1) is formulated as follows.

$$\begin{aligned}
\min \quad & \sum_{e\in E} c_e x_e \\
\text{s.t.} \quad & \sum_{(i,j)\in A} f_{ij} - \sum_{(j,i)\in A} f_{ji} = \left\{ \begin{array}{c} n-1, \text{ if } i=1 \\ -1, \forall i \in V\setminus\{1\} \end{array} \right. \\
& f_{ij} \le (n-1)\, x_{ij} \qquad \forall \{i,j\} \in E, \\
& f_{ji} \le (n-1)\, x_{ij} \qquad \forall \{i,j\} \in E \\
& \sum_{e\in E} x_e = n-1, \qquad f \ge 0 \\
& x_e \in \{0,1\} \quad \forall e \in E.
\end{aligned} \tag{4.5}$$

A second formulation uses the directed version of the MST problem. Any node named *root* node, let say 1 is selected as the root node. Then it is possible directs the edges of any spanning tree so that the path from the root node to any other node is directed from the root node to that node. The digraph so obtained is denoted by $D = (V, A)$. The resulting formulation is known as the *directed multicommodity flow* model. In the model, every node $k \neq 1$ defines a commodity: one unit of commodity k originates at the root node 1 and must be delivered to node k. Letting f_{ij}^k be the flow of commodity k on arc (i, j); the variables x_{ij} define a capacity for the flow of each commodity on arc (i, j) only if the arc is a member of the directed spanning tree defined by the vector x. They formulated the model (P2) as follows.

$$\begin{aligned}
\min \quad & \sum_{\{i,j\}\in E} c_{ij}\,(x_{ij} + x_{ji}) \\
\text{s.t.} \quad & \sum_{(j,1)\in A} f_{j,1}^k - \sum_{(1,j)\in A} f_{1,j}^k = -1 \quad \forall k \in V\setminus\{1\} \\
& \sum_{(j,i)\in A} f_{j,i}^k - \sum_{(i,j)\in A} f_{i,j}^k = 0 \quad \forall i, \forall k \in V\setminus\{1\} \text{ and } i \neq k \\
& \sum_{(j,k)\in A} f_{j,k}^k - \sum_{(k,j)\in A} f_{k,j}^k = 1 \quad \forall k \in V\setminus\{1\} \\
& f_{ij}^k \le x_{ij} \quad \forall (i,j) \in A \text{ and } \forall k \in V\setminus\{1\} \\
& \sum_{(i,j)\in A} x_{ij} = n-1, \\
& f \ge 0 \text{ and } x \ge 0.
\end{aligned} \tag{4.6}$$

Note that in the model the variables x need not be forced to integer values since the polyhedra of (P2) has integer extreme points. Yaman *et al.* [66] use both formulation (P1) to characterize the edges on the robust spanning tree, and the dual version of model (P2) to find the cost of the minimum spanning tree when the costs of all edges on the robust tree are at upper bounds and the costs of all remaining edges are at lower bounds. The flow balance constraints are replaced by the equivalent inequality constraints and then the dual LP of (P2) can be written as follows:

$$\begin{aligned}
\max \quad & \sum_{k\in V,\ k\neq 1} \left(\alpha_k^k - \alpha_1^k\right) + (n-1)\,\mu \\
\text{s.t.} \quad & \sigma_{ij}^k \geq \alpha_j^k - \alpha_i^k \quad \forall\,(i,j)\in A \text{ and } \forall k\in V\setminus\{1\} \\
& \sum_{k\neq 1} \sigma_{ij}^k + \mu \leq c_{ij} \quad \forall\,\{i,j\}\in E \\
& \sum_{k\neq 1} \sigma_{ji}^k + \mu \leq c_{ij} \quad \forall\,\{i,j\}\in E \\
& \sigma,\ \alpha \geq 0 \text{ and } \mu \text{ unrestricted.}
\end{aligned} \tag{4.7}$$

The robust tree formulation (R) follows, where variables x_e define the edges included in the tree,

$$\begin{aligned}
\min \quad & \sum_{e\in E} c_e^- x_e - \sum_{k\in V,\ k\neq 1} \left(\alpha_k^k - \alpha_1^k\right) - (n-1)\,\mu \\
\text{s.t.} \quad & \sigma_{ij}^k \geq \alpha_j^k - \alpha_i^k \ \forall\,(i,j)\in A,\ \forall k\in V\setminus\{1\} \\
& \sum \sigma_{ij}^k + \mu \leq c_{ij}^- + \left(c_{ij}^+ - c_{ij}^-\right) x_{ij},\ \forall\,\{i,j\}\in E \\
& \sum \sigma_{ji}^k + \mu \leq c_{ij}^- + \left(c_{ij}^+ - c_{ij}^-\right) x_{ij},\ \forall\,\{i,j\}\in E \\
& \sum_{(i,j)\in A} f_{ij} - \sum_{(j,i)\in A} f_{ji} = \begin{cases} n-1 \text{ if } i=1 \\ -1\ \forall i\in V\setminus\{1\} \end{cases} \\
& f_{ij} \leq (n-1)\,x_{ij}\ \forall\,\{i,j\}\in E \\
& f_{ji} \leq (n-1)\,x_{ji}\ \forall\,\{i,j\}\in E \\
& \sum_{e\in E} x_e = n-1 \\
& f,\sigma,\alpha \geq 0 \text{ and } \mu \text{ unrestricted} \\
& x_e \in \{0,1\}\ \ \forall e\in E.
\end{aligned} \tag{4.8}$$

The preprocessing procedures are based on the identification of weak and strong edges in the input graph. Yaman *et al.* [66] found simple and efficient algorithms to identify these edges.

Computational experience and analysis. Formulation (R) was used to compute optimal solutions in complete graphs with $n = 10, 15, 20, 25$. The preprocessing procedures were effective eliminating % of the edges of the graph. However we note that the times execution after preprocessing remain high, showing the problem hardness.

We now analyze the Benders decomposition approach by Montemanni [53].

Mathematical programming formulation. Montemanni used the same formulations as [66] but the primal MST formulation is changed with another one with fewer variables and more constraints. The resulting MIP formulation is denoted by RST. For any given vector $\tilde{x} \in X$ it is possible to define a problem in variables σ, α and μ only, starting from the mixed integer program RST. This problem is called the primal subproblem and it is denoted as $(P(\tilde{x}))$. Dualizing again, they obtain the next dual subproblem $D(\tilde{x})$ which is a classic formulation for the minimum spanning tre problem in variables f and y (see Magnanti and Wolsey [49]. Let R be the feasible region (not empty) of the dual subproblem and let R_P be the set of extreme points of R. By strong duality and by using the fact that R is a polytope, the primal subproblem is feasible and bounded. Since $D(\tilde{x})$ is a linear program, original RST can be written (see [53] for the details) in more compact way as follows:

$$\begin{aligned}
(M)\ \min \quad & z \\
\text{s.t.} \quad & z \geq \sum_{\{i,j\}\in E} u_{ij}x_{ij} - \sum_{\{i,j\}\in E} \left(c_{ij}^{-} + \left(c_{ij}^{+} - c_{ij}^{-}\right) x_{ij}\right)\left(y_{ij} + y_{ji}\right) \ \forall y \in R_P \\
& \sum_{\{i,j\}\in E} x_{ij} = |V| - 1 \qquad\qquad (4.9) \\
& \sum_{\{i,j\}\in \Gamma(C)} x_{ij} \geq 1 \quad \forall C \subset V \\
& x_{ij} \in \{0,1\} \quad \forall \{i,j\} \in E.
\end{aligned}$$

From the formulation (M) a Benders decomposition approach is developed, which is based on an iterative scheme. 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 approach. The relaxed master problem is described as follows:

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.

$$
\begin{aligned}
(M^\tau)\text{min} \quad & z \\
\text{s.t.} \quad & z \geq \sum_{\{i,j\}\in E} u_{ij}x_{ij} - \sum_{\{i,j\}\in E} \left(c_{ij}^- + \left(c_{ij}^+ - c_{ij}^-\right) x_{ij}\right)(y_{ij} + y_{ji}) \ \forall y \in R_P^\tau \\
& \sum_{\{i,j\}\in E} x_{ij} = |V| - 1 \qquad (4.10) \\
& \sum_{\{i,j\}\in \Gamma(C)} x_{ij} \geq 1 \ \ \forall C \subset V \\
& x_{ij} \in \{0,1\} \quad \forall \{i,j\} \in E.
\end{aligned}
$$

A procedure to efficiently solve $M\tau$ to optimality was used.

Notice that as τ increases, M^τ progressively looses unimodular property (true only for $\tau = 1$), and it becames more and more difficult to solve in terms of integer programming. Montemanni [53] first applied the preprocessing procedure already discussed and then applied Benders decomposition.

In his paper, Montemanni [53] compared the performance of the Benders decomposition approach with the algorithm (already discussed) by Yaman *et al.* [66], the Branch and bound algorithm by Aron and Von Hentenryck [7], and the branch and bound algorithm by Montenmanni and Gambardella [56]. For each family of problems, Montemanni [53] reported the average number of iterations of the **Main step** required by the algorithm solving the Benders decomposition approach to certify an optimal solution. It is showed that the number of iterations required by the Benders decomposition approach to converge increases with V, as expected. Also, it is noted that the number of iterations does not explode for large values of $|V|$. For benchmark 2 Montemanni [53] reported that τ increases at the increasing of the average interval width (depending on p).

Considering the computational results in [53,56] we conclude that optimal solutions for this problem are very difficult to obtain using these exact algorithms applied to complete graphs with more than 50 nodes. The run time increases dramatically with the number of nodes.

Finally, Conde [33] developed a branch and bound algorithm for the MMR Spanning Arborescence problem. The computational behavior of the proposed method is good because it is close to the behavior of previous branch and bound algorithms for his undirected problem version.

4.1.3. *MMR-TSP*

The most important effort devoted to exactly solve the robust counterpart of the TSP was done by Montemmani *et al.* in [59]. Based on structural properties, an appropiate mathematical programming formulation is presented which allows the development of three exact approaches: B&B, B&C and Benders decomposition. These algorithms are implemented and then compared; again, Benders decomposition had the best relative performance.

Mathematical programming formulation. Since duality equivalences are not valid in the case of the TSP independently of the formulation used, it is not possible to derive MILP formulations like the one proposed for the MMR-ST. However, since a structural property similar to the one valid for the MMR-ST also holds for the MMR-TSP, it is possible to give a mathematical programming formulation. This formulation has a huge number of constraints, it is useful in order to design part of the algorithmic framework developed by the authors. The mathematical programming formulation given for the MMR-TSP is presented in (4.11)–(4.14).

$$\min \quad \sum_{\{i,j\}\in E} c_{ij}^{+} x_{ij} - r \tag{4.11}$$

$$\text{s.t.} \quad r \leq \sum_{\{i,j\}\in E} y_{ij} c_{ij}^{-} + \sum_{\{i,j\}\in E} y_{ij}\left(c_{ij}^{+} - c_{ij}^{-}\right) x_{ij} \qquad \forall y \in \mathcal{TSP} \tag{4.12}$$

$$x \in \mathcal{TSP} \tag{4.13}$$

$$r \in \mathcal{R}. \tag{4.14}$$

In this formulation, constraint (4.13) says that vector x should represent a Hamiltonian circuit. Any feasible MILP formulation for the classic TSP can be used. This set of constraints substitutes the nested min operator and incorporates the structural property for the calculation of the maximum regret of the tour defined by x variables; note that in these inequalities, y plays the role of a constant vector representing a tour.

Independently from the formulation chosen to make constraint (4.13) explicit, the main bottleneck of this formulation is represented by constraints (4.12), that are exponential in number (in case graph G is complete, all possible permutations of the nodes in $V\setminus\{0\}$ are feasible tours). The formulation is consequently not suitable to be directly handled as it is, in case of realistic size problems, and specific exact algorithms must be designed.

Exact algorithms. As it has been already said, three exact approaches for the MMR-TSP are developed: B&B, B&C and Benders decomposition. Only B&C and Benders decomposition methods are based on the formulation (4.11)–(4.14).

- **Branch and bound algorithm B&B**: This algorithm is based on the B&B algorithms presented by Montemmani and Gambardella [56] for the MMR-MST and Montemmani *et al.* [54] for the MMR-SP.

  The algorithm builds and visits a search-tree and the main elements of the procedure are the branching strategy, the calculation of a support solution and the robustness measure (regret) associated with that solution. For the efficient calculation of the support solution and the regret associated with that solution, classic instances of the TSP must be solved, therefore the available state-of-the-art tools for solving the classic TSP are used.

- **Branch and cut algorithm B&C**: As it is said, this algorithm relies on the formulation given by (4.11)–(4.14).

  The first element of the algorithm to take into consideration is the use of a MILP formulation for constraints (4.13) in the formulation of the MMR-TSP. The authors used the formulation presented by Dantzig *et al.* [37] to describe $\mathcal{TSP}$ because is convenient from a computational point of view. The main drawback of this formulation is the presence of *subtour elimination constraints* which are exponential in number, but it is known that only a few of them will be active at optimality. Thus, the strategy adopted in the paper is to iteratively add those subtour elimination constraints that are violated in the last available solution.

  Method B&C shares many elements of the B&B algorithm; they both build and visit a search-tree and each node of this search-tree can be identified the same elements used for the B&B algorithm. The main difference between the two methods is the calculation of the lower bound given by the solution at each node. At each node of the search-tree defined in the B&C procedure a linear programming relaxation of the MMR-TSP formulation is solved and the obtained solution reinforces the formulation with some cuts originated by the violation of one or more subtour elimination constraint.
- **Benders decomposition algorithm BD:** As it is observed in the formulation of the MMR-TSP, there is an exponential number of constraints of type (4.12). Therefore, a Benders descomposition approach is proposed by the authors to handle these constraints.

  In the case of formulation (4.11)–(4.14), only a few constraints of type (4.12) will be active at optimality, and therefore these constraints are generated iteratively when violated by using a Benders decomposition strategy. A relaxed version of the MMR-TSP is first solved with no constraint of type (4.12) and the most violated constraint of type (4.12) is added reinforcing the relaxed formulation which will be solved again; the process repeats until optimality tests are fulfilled. The auxiliary classic TSP problems that must be solved in the process are solved by using standard exact approaches developed for the symmetric TSP.

Computational results. Algorithms B&B, B&C and BD used ILOG CPLEX 9.0 to handle and solve the auxiliary MILP problems. Each time a classic TSP had to be solved to optimality, the authors used callable libraries based on algorithm LKH 1.3 and Concorde 03.12.19, that represented the state-of-the-art heuristic and exact algorithm for the symmetric TSP, respectively.

Since no benchmark problem was available for MMR-TSP with interval data, the authors generated a new benchmark of two families of problems. The first family, *random instances* is a set of non-Euclidean random instances and the second family of problems, *TSPLIB instances*, was generated from classic Euclidean TSP instances available at the TSPLIB.

The computational results presented in the paper for both benchmark instance suggest that B&B is not suitable even for small size problems (30 nodes). B&C

algorithm presents a better performance, being able to solve to optimality instances up to 50 nodes in the case of *random instances* and up to 42 nodes in the case of *TSPLIB instances*. BD strategy presents the best performance among the three algorithms; in the case of *random instances* the algorithm solves to optimality instances up to 80 nodes, and up to 58 nodes in the case of *TSPLIB instances*; however, BD is able to handle larger instances, 360 nodes in case of *random* and 180 in case of *TSPLIB*, presenting average GAP equal to 13.21% and 27.40% respectively.

The results also suggested an influence of parameter M in case of *random instances* and parameter β in case of *TSPLIB instances*. These parameters manage the different levels of uncertainty associated with each instance. When increasing the value of M nd β for a given instance, and thus increasing the range of the corresponding cost intervals, the computational times of BD also increases.

From the experiments it is clear that the algorithms implemented are not able to solve (with $GAP = 0$) instances with over 100 nodes, especially for TSPLIB instances.

4.1.4. *Another MMR-CO problems*

s-t path (MMR-SP*)*. A Benders decomposition method was also applied in [57] obtaining similar conclusions to those appearing in [53] for MMR-ST. Kasperski [42] proposed a MIP formulation and solved by CPLEX a set of instances containing until 900 nodes in reasonable times and he also proposed a branch and bound approach solving the same instances in lower times.

Linear resource allocation problem (MMR-RA). The MMR-RA problem studied by Averbakh [12] is defined as follows: Find non-negative integer (or real) numbers $x_1, x_2, \ldots, x_n$ so as to minimize the linear function $w_1x_1 + w_2x_2 + \ldots + w_nx_n$, subject to the constraint $x_1 + x_2 + \ldots + x_n = p$. Interval data uncertainty in the coefficients $w_1, w_2, \ldots, w_n$ is considered, where it is assumed that an interval estimate is known for each coefficient, and each coefficient can take any value from the corresponding interval of uncertainty, regardless of the values taken by other coefficients. He presented an algorithm with complexity $O(p \log n + n \log(n+p))$ for the case of integer variables and an $O(n \log n)$ algorithm for the case of continuous variables. The latter result implies an $O(n \log n)$ algorithm for the interval data MMR continuous Knapsack problem.

After that, Conde [31] developed a new linear-time algorithm for the MMR continuous unbounded knapsack problem, improving the result in [12].

MMR facility location problems. Exact algorithms for MMR robust location have generally been developed for only certain types of problems due to the NP-Hard behavior given by minmax formulations; locating a single facility or locating facilities on certain types of networks are usually the considered cases. Besides the long list of problems and their corresponding exact algorithms for both special and general networks presented by Snyder [65], recent papers consider related problems; Aloulou *et al.* [5] developed an $O\left(mnq\left(\log n + q\right)\right)$ algorithm for the 1-center problem in a general network with a discrete set of scenarios, being m

the number of edges, n the number of nodes and q the number of scenarios. In an extensive paper, Averbakh and Bereg [14] considered the 1-median and weighted 1-center problems on a plane with uncertain weights and customer coordinates; they present an $O\left(n^2 \log^2 n\right)$ algorithm for the 1-median problem and an $O\left(n \log n\right)$ algorithm for the weighted 1-center problem. They also developed exact algorithms considering Euclidean distances. Conde [34] studied doubly weighted *centdian* (a combination of center and median) location on trees assuming that both types of weights, demands and relative importance of the costumers, are represented by closed intervals but edge lengths are completely known; an $O\left(n^3 \log n\right)$ exact algorithm is presented.

4.2. Approximation algorithms, heuristics and metaheuristics

Aissi *et al.* [2] studied the approximation of minmax (regret) versions of problems like MST, SP and Knapsack. They obtain some important results which are summarized next.

# scenarios	Constant	Constant	Non-constant	Non-constant
	Minmax	Minmax regret	Minmax	Minmax regret
SP	FPTAS	FPTAS	Not $(2-\epsilon)$ approx.	Not $(2-\epsilon)$ approx.
MST	FPTAS	FPTAS	Not $(3/2-\epsilon)$ approx.	Not $(3/2-\epsilon)$ approx.
Knapsack	FPTAS	Not at all aprox.	Not at all aprox.	Not at all aprox.

Kasperski and Zieliński [43] designed a polynomial time approximation algorithm (K&Z algorithm) with a performance ratio of 2 for MMR-CO for which their classic version is polynomially solvable. This class of problems includes MST, SP and AP. The algorithm first determines scenario S in which the costs of the elements are the midpoints of their corresponding cost intervals and then a polynomial algorithm $AOpt$ is applied for the classic CO problem using the determined scenario S. The complexity of the K&Z algorithm is then defined by the complexity of algorithm $AOpt$. In [24], Candia and Álvarez presented a generalization of the K&Z algorithm, and they proved that the property of the solution remains valid if we replace the number $\frac{1}{2}$ by any positive real number p when the scenario S is defined in the K&Z algorithm. This result is interesting, because it is possible to think that the scenario S given by the midpoints of each interval is the only scenario satisfying the property established; to the contrary, the result shows that this property essentially comes from the sum of the extreme values of each interval cost. In the same paper, the situation where several solutions might be found is also analyzed. Conde [35] extends the result of [43] to models with compact constraint sets. Kasperzki and Zieliński (see [44]) also proposed a FPTAS for MMR CO under the assumption that a pseudopolynomial algorithm is given. Furthermore, they show the uselfuness of the obtained results when studying MMR versions of the shortest (longest) path and the minimum (maximum) s-t cut in series-parallel graphs.

Conde and Candia [36] presented a set of structural properties for the MMR spanning arborescence problem, a generalization of MMR-ST on directed networks (thus NP-Hard problem). The properties were used to obtain upper and lower bounds for the optimal value and a heuristic based on these bounds was proposed. Computational experiments illustrating the quality of the results achieved by the heuristic were presented; very near optimal solutions were found in short times for randomly generated graphs with up to 125 nodes. The experiments also included comparisons with the K&Z algorithm and they showed how a feasible solution, giving a good upper bound for the optimum value, improves by 5.36% on average the objective value obtained from the solution given by the K&Z algorithm. Finally, the convergence of the heuristic to an optimal solution was studied using a class of zero sum games.

Montemanni *et al.* [58] studied heuristics and preprocessing techniques for interval data MMR-TSP. In particular, they presented efficient tools for estimating the robustness of a given tour, producing heuristic solutions to the problem and preprocessing the problem in order to identify edges that will never be on an optimal robust tour. The tools were evaluated from a computational point of view. Three simple heuristics were tested, where K&Z was one of them; another (HU), is defined by the scenario U, where all the costs are at the highest possible value; and the last (HMU), runs in sequence K&Z and HU and returns the solution with the lowest upper bound for the cost between the two. HMU had a very good performance for several classes of instances considering up to 58 nodes.

Nikulin [61] proposed the application of the metaheuristic simulated annealing (SA) to MMR-ST, in order to efficiently obtain an approximate solution for a family of benchmarks similar to the ones presented in [53,56]. Similar to the approach of Yaman *et al.* [66], non weak edges can be deleted and strong edges are detected (both in polynomial time) in a preprocessing stage. SA is one of the most popular metaheuristics and here we present the most important steps of the implemented SA.

- *Search Space.* Here is an important variant of the typical definition of points in the search space. Typically the point are defined by spanning trees but in this approach then points are defined only by connected graphs. So, at each iteration it is not necessary to check whether the search point represents an acyclic graph or not. Clearly, as far as any connected graph with cycles contains some spanning tree and the value of objective of such graph is always greater than the value of the objective for the spannning tree then this spanning tree will be most likely detected later during the execution of SA. This definition allows to save a lot of running time avoiding extra checking for non-cicling.
- *Initial Solution.* An input graph (which is assumed connected) is chosen as the starting point.
- *Neighbourhood search moves.* One edge from the input graph is randomly selected. One move is defined in the following way. If the edge belongs to the current solution (a connected graph) and deleting it disconnects the graph then the search fail and a new edge must be selected. Otherwise, a new connected graph is obtained; if the worst case scenarios of the two connected graphs are evaluated then the cost charge Δ of performing a neighbourhood search move is calculated by the difference of two minimum

spanning tree problems minus the cost of the selected edge at his maximum value. Both minimum spanning trees problems are defined by the worst-case scenarios of the corresponding connected graphs. The situation is similar for the other move, that is, if the selected edge does not belong to the current solution. It is also noted the move in this case means to add the edge to a connected graph and then the new graph is necessarily connected and consequently the checking of connectivity is saved.

- *Acceptance probability rule.* The very known Metropolis acceptance rule is used.
- *Cooling Schedule.* The initial cooling temperature T_0 is defined as $T_0 = 100c_{\max}$. where $c_{\max}$ is the maximum of the upper costs. Annealing schedule was defined as a geometric decreasing with parameter $\alpha = 0.95$
- *Termination criterion.* SA is repeated until the temperature decrease to nearly zero before termination, the limit 0.001 was used.

The efficiency of the algorithm was improved by using gradient descent method. Let L be the parameter that determines the number of succesful moves that are considered at each temperature level. A larger value increases the optimization time, but tends to yield solutions with a narrower spread around the global optimum. For instances with small number of nodes (up to 10) the parameter was defined as 10, for medium instances (with 15–20 nodes) the parameter value was $L = 30$ and for greatest instances the parameter was defined as $L = 50$. Among L possible moves it is chosen the one with minimal value of Δ.

Computational experiments. A family of benchmarks similar to the one presented in [59] was used in order to evaluate SA. Complete graphs of 10, 25 and 50 nodes was considered. For each edge (i, j) the lower bound of its cost, l_{ij} was randomly generated from (0,20) and the upper bound u_{ij} from $(l_{ij}, 40)$. The running times of SA are compared with the result of Montenmanni [53,56] where the fastest exact algorithms based on branch and Bound and Benders decomposition were presented. For comparison the differences between the machines used in the experiments where considered when the algorithms were compared. Nikulin [61] shows that SA takes 185 s in solve the greatest instances (50 nodes) while Benders decomposition takes 1153 s. Based on these results the author commented that "the SA algorithm can be effectively applied for large instances where all the known exact techniques are time consuming". For a more detailed analysis of the SA algorithm, Nikulin addressed four benchmark instances with 5, 10, 15 and 20 nodes. He used the mixed integer reformulation of the robust spanning tree problem in Yaman *et al.* [66], encoded the model with AMPL-lenguage and solved the instances by ILOG CPLEX 7.0.o in order to get an exact solution for the instances. He runs $s = 100$ times the SA metaheuristic calculating the number s_1 of successful runs of SA, that is when SA found the optimal solution, and also calculated the number s_2 of satisfactory runs of SA, that is, where the approximate solution is very close to optimal one. Furthermore, Nikulin compared the results of SA with the results produced by the heuristic KZ ([43]). For all benchmarks considered, the solution produced by SA outperformed the solution given by KZ heuristic. In our opinion the results obtained by SA are good but it is difficult to be sure about the quality of SA solutions for greater instances.

MMR-A. Pereira and Averbakh ([63]) propose three heuristics for the problem: a variable depth neighborhood local search and two hybrid population-based heuristics. They also mention that the only references about heuristics designed for MMR-A are from the book of Kasperski [42] where he compared an exact algorithm with heuristics HM and HU defined starting this section. Since the variable depth neighborhood local search was not competetive with population-based heuristics, as informed by the authors, we only describe the latter approaches. The heuristics based on populations use the same concepts of genetic algorithms (GA), a traditional approach for solving NP-Hard optimization problems. The main components of a GA algorithm are: chromosome representation, initialization, selection, crossover and mutation operators; some detail about every component of GA is presented.

Representation and initialization. In the algorithm, an individual is a feasible solution for the problem, that is, a perfect matching and it is represented by a vector of size n (chromosome). Elements of sets U and V are indexed, so elements $u \in U$ and $v \in V$ can be considered as numbers from set $\{1, 2, \ldots, n\}$and a position k in the vector contains the task assigned to the worker k . An assignment of a task to a specific worker (a component of the cromosome vector) is called a *gene*. Then, a chromosome consists of n genes.

All individuals in the initial population are different and they are generated solving classic Assignment problems for specific scenarios.

Parent selection and crossover. Parent selection is based on a binary tournament selection operator. Two parents are randomly chosen from the current population and the better fit of the two becomes the parent. Two parents produce one child. The crossover operator is a combination of the generalized fitness-based crossover operator with an optimization subproblem. The basic idea of the authors is to keep the genes that are shared by both parents, and to optimize the other genes by trying to solve approximately a reduced MMR-A problem where the variables that correspond to the genes shared by the parents are fixed and the other variables are free. Two versions of the population-based heuristic are defined from this basic idea. The versions differ in the way the residual MMR-A is solved. In the first version the apply the scenario-based heuristic to the residual instance and use the obtained solution to define the values of the remaining free genes; this version is called GA-SCEN. In the second version (GA-MILP) they apply CPLEX MILP solver to the MILP formulation of the residual instance.

Mutation. The purpose of the mutation operator is to diversify the search. They define the maximum number of genes to mutate, v, as follows:

$$v = \frac{m_f}{1 + \mathrm{e}^{\frac{(-4m_g(t-m_c))}{m_f}}},$$

where t is the number of child solutions that have been generated, m_f specifies the final stable mutation rate, m_c specifies the number of child solutions generated at which the mutation rate of $\frac{m_f}{2}$ is reached, and m_g specifies the gradient at $t = m_c$. The value v is then rounded up and all these constants are parameters of the algorithm. The objective of this rate is that the importance of mutations increases as the search approaches convergence.

Fitness evaluation. The fitness (maximal regret) of any individual X can be obtained by solving a classic Assignment problem. The child obtained after the application of the crossover operator is inserted in the population instead of the previously worst member.

Some remarks. The authors commented that their GA approach contains some non standard characteristics. First, not all genes are inherited from the parents, and second, using optimization within the crossover operator introduces Lamarckian features to the process. Population-based heuristics that combine Darwinian features and Lamarckian ones are known as *memetic algorithms.* This GA approach exploits the efficiency of the MILP solver for small instances when solving a large problem and also exploits the good performance of the fast scenario-based heuristic. With respect to *convergence criteria* one standard criteria is used, GA stops if the best known solution is not improved for a number of iterations equal to the population size.

Computational experience with GA. The authors considered the same four sets of instances they used for the study of the performance of an exact algorithm. They make some observations after the analysis of the computational experience.

- Population-based heuristics are superior to other tested heuristics.
- Starting with some problem size, population-based heuristics are superior to exact methods (remember that for many instances they ouput large gaps).
- Starting with some problem size, GA-SCEN is superior to GA-MILP, while GA-MILP may be better or comparable for smaller problem sizes.
- For large instances ($n > 300$), GA-SCEN is overall superior to all other tested approaches.

Finally they suggested some decisions when you must solve a MMR-A problem depending on the problem size.

Our opinion about the performance of both exact and heuristic algorithms proposed for MMR-A is that the performance strongly depends on the particular class studied. Clearly, McGeoch and Beasley instances are harder than Montemanni and Kasperski instances. Other important point is the small difference in the performance of the algorithms tested for each class of instances of both exact and heuristic approaches. Again, new approaches, principally heuristics could be designed for Beasley and McGeoch instances.

5. Future work

Optimization under uncertainty is nowadays a developing area . Theory, models, algorithms and applications are expanding the limits of optimization generating new opportunities of solving more real problems. In particular, an important part of this area related to Combinatorial Optimization problems using the MMR model has been presented here. However, we are sure important challenges remain today. Some of them are discussed next.

It would be very interesting to study relations among the MMR models and other robustness models like the Bertsimas and Sim model [20], and the Chen *et al.* model [29]; preliminary studies we conducted (see [6]) showing some particular connections between these two models when applied to robust shortest path. Furthermore, because the robust models are applicable, we feel there are opportunities to consider robust optimization models for solving more real world problems than those presented here.

Computational aspects are very important because the three models commented here, MMR, Bertsimas and Sim [20] and Chen *et al.* [29] are associated to NP-hard problems in most cases (MMR model). Otherwise there are polynomial algorithms for several class of problems but with high degree of polynomiality [20,29]. The design of new exact algorithms and heuristics exploting the particularities of the complex problems and the uncertainty structure is necessary to get an effective algorithmic design.

We suggest as an important challenge the study of robust problems that even in their classic versions are proved NP-Hard, and that the existing algorithms for such problems consider the SP or MST as underlying problems. In this context, Montemmani *et al.* [59] studied the TSP using a structural property proven valid for certain problems, and heuristics are developed considering the fact that SP must be solved. The Steiner problem (STN) might be an interesting candidate because it falls within the mentioned framework and it is known that the heuristic development for the STN uses SP and MST as underlying problems and several algorithms exist for their MMR problems.

Finally, it is necessary the design of efficient heuristics for solving MMR-CO. From the analysis done in this paper it is clear that exact algorithms only can solve, with gap 0, small instances for most problems studied. Only a few heuristics have been designed for these problems considering the great development of these algorithms.

References

[1] H. Aissi, C. Bazgan and D. Vanderpooten, Complexity of the minmax and minmax regret assignment problem. *Oper. Res. Lett.* **33** (2005) 634–640.

[2] H. Aissi, C. Bazgan and D. Vanderpooten, Approximation of min-max and min-max regret versions of some combinatorial optimization problems. *Eur. J. Oper. Res.* **179** (2007) 281–290.

[3] H. Aissi, C. Bazgan and D. Vanderpooten, Approximation complexity of min-max (regret) versions of shortest path, spanning tree, and knapsack. *Lect. Notes Comput. Sci.* **3669** (2005) 862–873.

[4] H. Aissi, C. Bazgan and D. Vanderpooten, Min-max and min-max regret versions of some combinatorial optimization problems: a survey. *Eur. J. Oper. Res.* **197** (2009) 427–438.

[5] M.A. Aloulou, R. Kalai and D. Vanderpooten, *Minmax regret 1-center problem on a network with a discrete set of scenarios.* Lamsade technical Report No. 132, LAMSADE, Université Paris-Dauphine, Cahier du LAMSADE (2006).

[6] E. Alvarez-Miranda and A. Candia-Vejar, Robust Shortest Path: Models, Algorithms and Comparisons, *Proceedings of the VI ALIO/EURO Workshop on Applied Combinatorial Optimization.* Buenos Aires, Argentina (2008).

[7] I. Aron and P. Van Hentenryck, A constraint satisfaction approach to the robust spanning tree with interval data, *Proceedings of the International Conference on Uncertainty in Artificial Intelligence UAI* (2002) 18–25.

[8] I. Aron and P. Van Hentenryck, On the complexity of the robust spanning tree problem with interval data, *Oper. Res. Lett.* **32** (2004) 36–40.

[9] T. Assavapokee, M.J. Realff and J.C. Ammons, Min-max Regret robust optimization approach on interval data uncertainty. *J. Optim. Theory Appl.* **137** (2008) 297–316.

[10] I. Averbakh, On the complexity of a class of combinatorial optimization problems with uncertainty. *Math. Program. Ser. A* **90** (2001) 263–272.

[11] I. Averbakh, Complexity of robust single facility location problems on networks with uncertain edge lengths. *Discr. App. Math.* **127** (2003) 505–522.

[12] I. Averbakh, Minmax regret linear resource allocation problems. *Oper. Res. Lett.* **32** (2004) 174–180.

[13] I. Averbakh, Computing and minimizing the relative regret in combinatorial optimization with interval data. *Discr. Optim.* **2** (2005) 273–287.

[14] I. Averbakh and S. Bereg, Facility location problems with uncertainty on the plane. *Discret. Optim.* **2** (2005) 3–34.

[15] I. Averbakh and O. Berman, Minmax regret median location on a network under uncertainty. *ORSA J. Comput.* **12** (2000) 104–110.

[16] I. Averbakh and O. Berman, Algorithms for the robust 1-center problem on a tree. *Eur. J. Oper. Res.* **123** (2000) 292–302.

[17] I. Averbakh and V. Lebedev, Interval data regret network optimization problems. *Discr. App. Math.* **138** (2004) 289–301.

[18] I. Averbakh and V. Lebedev, On the complexity of minmax regret linear programming. *Eur. J. Oper. Res.* **160** (2005) 227–231.

[19] J.E. Beasley, OR-Library: distributing test problems by electronic mail. *J. Oper. Res. Soc.* **41** (1990) 1069–1072.

[20] D. Bertsimas and M. Sim, Robust discrete optimization and network flows. *Math. Program. Ser. B* **98** (2003) 49–71.

[21] D. Bertsimas and M. Sim, The price of robustness. *Oper. Res.* **52** (2004) 35–53.

[22] L. Bianchi, M. Dorigo, L. Gambardella and W. Gutjahr, *Metaheuristics in Stochastic Combinatorial Optimization: a Survey.* IDSIA Technical Report, IDSIA-08-06 (2006), *Natural Computing* **8** (2009) 239–287.

[23] R.E. Burkard and H. Dollani, A note on the robust 1-center problem on trees. *Disc. Appl. Math.* **138** (2004) 289–301.

[24] A. Candia-Véjar and E. Álvarez-Miranda, On a class of interval data minmax regret CO problems. **2007** (2007) 123–128.

[25] N. Chang and E. Davila, Siting and routing assessment for solid waste management under uncertainty using the grey min-max regret criterion. *Environ. Manag.* **38** (2006) 654–672.

[26] N. Chang and E. Davila, Minimax regret optimization analysis for a regional solid waste management system. *Waste Manag.* **27** (2007) 820–832.

[27] X. Chen, J. Hu and X. Hu, On the minimum risk-sum path problem, *ESCAPE'07 Proceedings, Lect. Notes Comput. Sci.* **4614** (2007) 175–185.

[28] X. Chen, J. Hu and X. Hu, The minimum risk spanning tree problem, *COCOA'07 Proceedings, Lecture Notes in Computer Science* **4616** (2007) 81–90.

[29] X. Chen, J. Hu and X. Hu, A new model for path planning with interval data. *Comput. Oper. Res.* **39** (2009) 1893–1899.

[30] E. Conde, An improved algorithm for selecting p items with uncertain return according to the minmax-regret criterion. *Math. Program. Ser. A* **100** (2001) 345–353.

[31] E. Conde, On the complexity of the continuous unbounded knapsack problem with uncertain coefficients. *Oper. Res. Lett.* **33** (2005) 481–485.

[32] E. Conde, Minmax regret location-allocation problem on a network under uncertainty. *Eur. J. Oper. Res.* **179** (2007) 1025–1039.

[33] E. Conde, A branch and Bound algorithm for minimum spanning arborescences. *J. Glob. Optim.* **37** (2007) 467–480.

[34] E. Conde, A note on the minmax regret centdian location on trees. *Oper. Res. Lett.* **36** (2008) 271–275.

[35] E. Conde, A 2-approximation for minmax regret problems via a mid-point scenario optimal solution. *Oper. Res. Lett.* **38** (2010) 326–327.

[36] E. Conde and A. Candia, Minimax regret spanning arborescences under uncertain costs. *Eur. J. Oper. Res.* **182** (2007) 561–577.

[37] G.B. Dantzig, D.R. Fulkerson and S.M. Johnson, Solutions of a large scale traveling salesman problem. *Oper. Res.* **2** (1954) 393–410.

[38] V. Deineko and G. Woeginger, On the robust assigment problem under fixed number of cost scenarios. *Oper. Res. Lett.* **34** (2006) 175–179.

[39] M. Demir, B. Tansel and G. Scheuenstuhl, Tree Network 1-median location with interval data: a parameter space-based approach. *IIE Trans.* **37** (2005) 429–439.

[40] R. Hites, Y. De Smeta, N. Risse, N. Salazar-Neumann and P. Vincke, About the applicability of MCDA to some robustness problems. *Eur. J. Oper. Res.* **174** (2006) 322–332.

[41] O. Karasan, M. Pinar and H. Yaman, *The Robust Shortest Path Problem with Interval Data*. Technical Report Bilkent University (2001), revised (2004).

[42] A. Kasperski, *Discrete Optimization with Interval Data: Minmax Regret and Fuzzy Approach*. Studies in Fuzzines and Soft Computing, Springer (2008).

[43] A. Kasperski and P. Zieliński, An approximation algorithm for interval data minmax regret combinatorial optimization problems. *Inf. Process. Lett.* **97** (2006) 177–180.

[44] A. Kasperski and P. Zieliński, The robust shortest path problem in series-parallel multidigraphs with interval data. *Oper. Res. Lett.* **34** (2006) 69–76.

[45] A. Kasperski and P. Zieliński, On combinatorial optimization problems on matroids with uncertain weights. *Eur. J. Oper. Res.* **177** (2007) 851–864.

[46] A. Kazakci, S. Rozakis and D. Vanderpooten, Energy crop supply in France: a min-max regret approach. *J. Oper. Res. Soc.* **58** (2007) 1470–1479.

[47] P. Kouvelis and G. Yu, *Robust discrete optimization and its applications.* Kluwer Academic Publishers (1997).

[48] R. Loulou and A. Kanudia, Minimax regret strategies for greenhouse gas abatement: methodology and application. *Oper. Res. Lett.* **25** (1999) 219–230.

[49] T.L. Magnanti and L. Wolsey, optimal trees, network models, in *Handbook in Operations research and management science* **7**, edited by M.O. Ball *et al.*, North-Holland, Amsterdam (1997) 503–615.

[50] H.E. Mausser and M. Laguna, A new mixed integer formulation for the maximum regret problem. *Int. Trans. Oper. Res.* **5** (1998) 389–403.

[51] H.E. Mausser and M. Laguna, Minimising the maximum relative regret for linear programmes with interval objective function coefficents. *J. Oper. Res. Soc.* **50** (1999) 1063–1070.

[52] H.E. Mausser and M. Laguna, A heuristic to minmimax absolute regret for linear programs with interval objective function. *Eur. J. Oper. Res.* **117** (1999) 157–174.

[53] R. Montemanni, A Benders decomposition approach for the robust spanning tree problem with interval data. *Eur. J. Oper. Res.* **174** (2006) 1479–1490.

[54] R. Montemanni, L.M. Gambardella and A.V. Donati, A branch and bound algorithm for the robust shortest path with interval data. *Oper. Res. Lett.* **32** (2004) 225–232.

[55] R. Montemanni and L.M. Gambardella, An exact algorithm for the robust shortest path problem with interval data. *Comput. Oper. Res.* **31** (2004) 1667–1680.

[56] R. Montemanni and L.M. Gambardella, A branch and bound algorithm for the robust spanning tree with interval data. *Eur. J. Oper. Res.* **161** (2005) 771–779.

[57] R. Montemanni and L.M. Gambardella, The robust shortest path problem with interval data *via* Benders decomposition, *4OR* **3** (2005) 315–328.

[58] R. Montemanni, J. Barta and L.M. Gambardella, *Heuristic and preprocessing techniques for the robust traveling salesman problem with interval data*. Technical Report IDSIA-01-06 (2006).

[59] R. Montemanni, J. Barta and L.M. Gambardella, The robust traveling salesman problem with interval data. *Transp. Sci.* **41** (2007) 366–381.

[60] Y. Nikulin, *Robustness in combinatorial optimization and scheduling theory: An annotated bibliography.* Christian-Albrechts University in Kiel, Working paper (2005).

[61] Y. Nikulin, Simulated annealing algorithm for the robust spanning tree problem. *J. Heurist.* **14** (2008) 391–402.

[62] Y. Nikulin, *Solving the robust shortest path problem with interval data using probabilistic metaheuristic approach.* N597 CAU (2005) (with A. Drexl).

[63] J. Pereira and I. Averbakh, Exact and heuristic algorithms for the interval data robust assignment problem. *Comput. Oper. Res.* **38** (2011) 1153–1163

[64] M. Salazar-Neumann, The robust minimum spanning tree problem: Compact and convex uncertainty. *Oper. Res. Lett.* **35** (2007) 17–22.

[65] L.V. Snyder, Facility location under uncertainty: A review. *IIE Trans.* **38** (2006) 547–564.

[66] H. Yaman, O. Karasan and M. Pinar, The robust spanning tree with interval data. *Oper. Res. Lett.* **29** (2001) 31–40.

[67] P. Zieliński, The computational complexity of the relative robust shortest path problem with interval data. *Eur. J. Oper. Res.* **158** (2004) 570–576.