Available online at www.sciencedirect.com
SCIENCE DIRECT®
Discrete Optimization 2 (2005) 273–287
DISCRETE OPTIMIZATION
www.elsevier.com/locate/disopt

Computing and minimizing the relative regret in combinatorial optimization with interval data

Igor Averbakh

Division of Management, University of Toronto at Scarborough, 1265 Military Trail, Scarborough, Ont., Canada M1C 1A4

Received 14 December 2004; received in revised form 2 July 2005; accepted 21 July 2005
Available online 7 October 2005

Abstract

We consider combinatorial optimization problems with uncertain parameters of the objective function, where for each uncertain parameter an interval estimate is known. It is required to find a solution that minimizes the worst-case relative regret. For minmax relative regret versions of some subset-type problems, where feasible solutions are subsets of a finite ground set and the objective function represents the total weight of elements of a feasible solution, and for the minmax relative regret version of the problem of scheduling n jobs on a single machine to minimize the total completion time, we present a number of structural, algorithmic, and complexity results. Many of the results are based on generalizing and extending ideas and approaches from absolute regret minimization to the relative regret case.

Keywords: Minmax regret optimization; Robust optimization; Uncertainty; Interval data; Polynomial algorithm

1. Introduction

In *combinatorial optimization with interval data* it is assumed that values of some parameters (coefficients) of the problem are not known; instead, interval estimates (uncertainty intervals) are specified for unknown parameters, and it is assumed that each parameter can take on any value from the corresponding uncertainty interval, regardless of the values taken by other parameters. *Minmax regret* combinatorial optimization deals with problems where only parameters defining the objective function may be uncertain, but the set of feasible solutions is known precisely; it is required to find a feasible solution which is reasonably close to the optimal one (in terms of the objective function value) for all possible realizations of data.

Minmax regret optimization is one of several robust optimization approaches in the literature for optimization problems with uncertainty in data. For other approaches, see [7,8,23] and the references therein. In particular, for combinatorial optimization problems with uncertain coefficients, Bertsimas and Sim [8] presented a methodology that allows to obtain robust formulations that inherit good properties (e.g., polynomial solvability, approximability) of the underlying combinatorial optimization problems. They introduce a parameter that bounds the number of coefficients that can vary (and thus controls the “degree of conservatism” of the obtained solution), and seek a feasible solution that optimizes the worst-case value of the objective function.

E-mail address: averbakh@utsc.utoronto.ca.

1572-5286/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.disopt.2005.07.001

Diversity and complexity of real life implies that no single modeling approach can address all possible situations, and therefore different modeling approaches should be thoroughly and rigorously investigated to understand their capabilities and limitations. In some complex cases, different modeling approaches can be applied in parallel to get better understanding of the situation. The minmax regret approach, in particular, focuses on the degree of suboptimality of the proposed solution for each scenario, rather than on the value of the objective function. A comprehensive discussion of practical situations where this may be relevant is available in the book [15]. This book also provides the state-of-art in minmax regret discrete optimization up to 1997. Recent results on minmax regret optimization can be found in [2–6,11,13,14,16,18,21,22,28,29], and the references therein.

In minmax regret optimization, there are two possible criteria that can be optimized: the worst-case *absolute regret* and the worst-case *relative regret*. The absolute regret is the "opportunity loss", or the difference between the achieved objective function value and the optimal objective function value under the realized scenario. The relative regret is the ratio of the "opportunity loss" to the optimal objective function value under the realized scenario. The relative regret objective is more appropriate than the absolute regret objective in situations where the percentage deviation from the optimum is a more appropriate measure of quality of a solution than the absolute deviation; that is, when the statement "option A is 10% more expensive than option B" is more meaningful for the decision maker than the statement "option A is $30,000 more expensive than option B". Kouvelis and Yu [15, p. 349] remark that the relative regret criterion is "less conservative" than the absolute regret criterion.

Interval data minmax relative regret versions of some optimization problems have already been analyzed in the literature (see, e.g., [5,18]); however, most of the literature in minmax regret combinatorial optimization with interval data is devoted to absolute regret problems, perhaps because these problems usually have simpler structure than relative regret problems. In this paper, we study the relative regret criterion in the context of a rather broad class of combinatorial optimization problems—so-called *subset-type* problems, where feasible solutions are some subsets of a finite ground set of cardinality n, and the objective function represents the total weight of elements of a feasible solution. Also, we consider the minmax relative regret version of one of the fundamental scheduling problems—the problem of minimizing the total completion time of n jobs that should be scheduled on a single machine (the total flow-time scheduling problem). One of the main goals of the paper is to explore how the ideas and approaches developed for the absolute regret problems can be extended to the relative regret case. Our focus in this paper is on computational complexity issues and connections between the relative regret and the absolute regret cases; developing practical algorithms for NP-hard cases or computational studies are outside of the scope of this paper. The general structure of the paper is outlined below.

In Section 2, we provide the main definitions and notation from minmax regret combinatorial optimization.

In Section 3, we study minmax relative regret subset-type problems. In Section 3.1, we discuss some useful properties of worst-case scenarios. In Section 3.2, we show that for a fixed feasible solution, computing the corresponding value of the worst-case relative regret can be done in polynomial time whenever the original optimization problem without uncertainty can be solved in polynomial time. Also, under mild conditions, computing the value of the worst-case relative regret for a fixed feasible solution can be done in strongly polynomial time if the original optimization problem without uncertainty can be solved in strongly polynomial time. The latter result is illustrated in the context of uniform matroid optimization and the minimum spanning tree problem. In Section 3.3, we present a general methodology of developing mixed-integer linear programming formulations for minmax relative regret versions of subset-type combinatorial optimization problems, and illustrate this methodology in the context of uniform matroid optimization, the minimum spanning tree problem, and the assignment problem. In Section 3.4, we investigate the relationship between the complexities of absolute regret and relative regret problems, and show that under some very mild conditions the relative regret version of a problem is at least as hard as the absolute regret version; using this result, we prove that the interval data minmax relative regret spanning tree and shortest path problems are strongly NP-hard, and the interval data minmax relative regret assignment problem is NP-hard. In Section 3.5, we show that if the number of nondegenerate uncertainty intervals is fixed or is bounded by the logarithm of a polynomial function of n, then the minmax relative regret version of a subset-type problem can be solved in polynomial time if the original problem without uncertainty can be solved in polynomial time. In Section 3.6, as an illustration of the general approach considered in this paper, we develop a strongly polynomial algorithm for the minmax relative regret version of the problem of finding a minimum weight base of a uniform matroid (i.e. the problem of selecting p elements of minimum total weight out of a set of n elements); the algorithm has complexity $O(pn \log n)$.

In Section 4, we study the interval data minmax relative regret total flow-time scheduling problem. In Section 4.1, we show that computing the value of the worst-case relative regret for a fixed sequence of jobs can be done in polynomial time. In Section 4.2, we prove that the problem is NP-hard. In Section 4.3, we show that the problem can be solved in polynomial time if the number of nondegenerate uncertainty intervals is fixed.

In Section 5, we provide some concluding remarks.

2. Preliminaries

Let us introduce the main definitions from minmax regret combinatorial optimization. Consider a combinatorial optimization problem in the following generic form:

$$\text{Minimize } \{F(X) \mid X \in A\},$$

where A is a set of feasible solutions, which is assumed to be finite in this paper, and $F(\cdot)$ is a function defined on A. Suppose that there is uncertainty in the objective function, that is, it is known only that $F(\cdot)$ is a member of a family of functions $\{F(s, \cdot), s \in S\}$ for some *set of scenarios* S. The set S is assumed to be a compact set in some metric space, and $F(s, \cdot)$ is assumed to be continuous in s. We assume that $F(s, X) > 0$ for any $s \in S$ and $X \in A$. Let $F^*(s)$ denote the optimum objective value for the following problem:

Problem OPT(s): Minimize $\{F(s, X) \mid X \in A\}$.

For any $X \in A$ and $s \in S$, the value $F(s, X) - F^*(s)$ (the value $(F(s, X) - F^*(s))/F^*(s)$) is called the *absolute regret* (the *relative regret*) for X under scenario s. Let

$$Z'(X) = \max_{s \in S} \{F(s, X) - F^*(s)\},$$

$$Z''(X) = \max_{s \in S} \left\{ \frac{F(s, X) - F^*(s)}{F^*(s)} \right\}.$$

Value $Z'(X)$ is called the *worst-case absolute regret* for X, and the problem of computing $Z'(X)$ will be referred to as Problem ABSREGR(X). Value $Z''(X)$ is called the *worst-case relative regret* for X, and the problem of computing $Z(X) = Z''(X) + 1$ will be referred to as Problem RELREGR(X). Observe that

$$Z(X) = Z''(X) + 1 = \max_{s \in S} \left(\frac{F(s, X)}{\min_{Y \in A} F(s, Y)} \right) = \max_{s \in S} \max_{Y \in A} \frac{F(s, X)}{F(s, Y)}. \tag{1}$$

In minmax regret optimization, either the worst-case absolute regret or the worst-case relative regret is minimized. Instead of minimizing the worst-case relative regret $Z''(X)$, it is convenient to (equivalently) minimize $Z(X)$. The following problem is the minmax absolute regret version of Problem OPT(s).

Problem ABSROB: $\min\{Z'(X) \mid X \in A\}$.

The following problem is the minmax relative regret version of Problem OPT(s).

Problem RELROB: $\min\{Z(X) \mid X \in A\}$.

(Acronim “ROB” refers to robustness, since minmax regret optimization is sometimes called robust optimization [15], although there are other concepts of robustness in the literature, see, e.g., [7,23].) Let z^* denote the optimum objective function value for Problem RELROB. Problems RELREGR(X) and RELROB are the focus of this paper.

For a $\mu \geqslant 1$ and $X \in A$, consider the following problem.

Problem RELAX(X, μ): Find $\hat{z}^*(X, \mu)$, where

$$\hat{z}^*(X, \mu) = \max_{s \in S} \max_{Y \in A} (F(s, X) - \mu F(s, Y)). \tag{2}$$

Consider also

Problem ROBRELAX(μ): $\hat{z}^*(\mu) = \min_{X \in A} \hat{z}^*(X, \mu)$.

(Note that Problem ABSROB is a special case of Problem ROBRELAX(μ) corresponding to $\mu = 1$.) The following two lemmas describe important relationship between Problems RELAX(X, μ), ROBRELAX(μ) and Problems RELREGR(X), RELROB, respectively.

Lemma 1. (a) *If* $\mu = Z(X)$, *then* $\hat{z}^*(X, \mu) = 0$, *and the sets of optimal solutions in* (1) *and* (2) *are equal*.
(b) *If* $\mu < Z(X)$, *then* $\hat{z}^*(X, \mu) > 0$.
(c) *If* $\mu > Z(X)$, *then* $\hat{z}^*(X, \mu) < 0$.

Proof. Variants of Lemma 1 are well known in optimization with ratio objective functions; see, e.g., [19,25]. □

Lemma 2. (a) *If* $\mu = z^*$, *then* $\hat{z}^*(\mu) = 0$.
(b) *If* $\mu < z^*$, *then* $\hat{z}^*(\mu) > 0$.
(c) *If* $\mu > z^*$, *then* $\hat{z}^*(\mu) < 0$.

Proof. If $\mu < z^*$, then Lemma 1(b) implies $\hat{z}^*(X, \mu) > 0$ for all $X \in A$, which implies statement (b). If $\mu > z^*$, then Lemma 1(c) implies that for some $X \in A$, $\hat{z}^*(X, \mu) < 0$, which implies $\hat{z}^*(\mu) < 0$. If $\mu = z^*$, then Lemma 1(a), (b) implies that $\hat{z}^*(X, \mu) \geqslant 0$ for all $X \in A$ and this inequality holds as an equality for at least one $X \in A$, which implies statement (a). □

For any integers k, t such that $k \leqslant t$, $[k : t]$ will denote the set of all integers between k and t (including k, t).

3. Subset-type combinatorial optimization problems with interval data

In this section, we assume that all elements of A are some subsets of a finite *ground set* E, $|E| = n$, and that for any $s = \{w_e^s, e \in E\}$ and $X \in A$, $F(s, X) = \sum_{e \in X} w_e^s$. Problems OPT($s$) of this type are sometimes called *subset-type* combinatorial optimization problems, and we will use this terminology. For example, in network optimization problems (shortest path, minimum spanning tree, minimum weight perfect matching, etc.) the ground set is the set of edges of a graph. For any $e \in E$, two positive integer numbers w_e^-, w_e^+ are given, $w_e^- \leqslant w_e^+$; the interval $[w_e^-, w_e^+]$ is called *the uncertainty interval* for the weight of e. We assume that S is the Cartesian product of all intervals of uncertainty. Let $\mathrm{CH}(A) \subset \{0, 1\}^n$ be the set of characteristic vectors of elements of A. (Vector $x = \{x_e, e \in E\}$ is a characteristic vector for an $X \in A$, if $x_e = 1$ for any $e \in X$ and $x_e = 0$ for any $e \in E \backslash X$.)

3.1. Structure of worst-case scenarios

For any $X, Y \in A$, define scenario $s(X, Y)$ as

$$w_e^{s(X,Y)} = \begin{cases} w_e^+ & \text{when } e \in X \backslash Y, \\ w_e^- & \text{otherwise.} \end{cases}$$

Lemma 3. (a) *For any* $X \in A$ *and* $\mu \geqslant 1$, *if* (s, Y) *is an optimal solution to Problem* RELAX(X, μ), *then* $(s(X, Y), Y)$ *is also an optimal solution to Problem* RELAX(X, μ).
(b) *For any* $X, Y \in A$ *such that* $\max_{s \in S} F(s, X)/F(s, Y) \geqslant 1$, $F(s(X, Y), X)/F(s(X, Y), Y) = \max_{s \in S} F(s, X)/F(s, Y)$.

Proof. Straightforward. □

A scenario $s \in S$ is called a *worst-case scenario* for an $X \in A$, if $Z(X) = F(s, X)/F^*(s)$.
Let S' denote the set of all *extreme scenarios*, that is, scenarios where the weight of each $e \in E$ is equal to either the lower or the upper bound of the corresponding uncertainty interval. Lemma 3 implies the following:

Corollary 1. *For any* $X \in A$, *value* $Z(X)$ *will not change if we replace* S *with* S' *in* (1). *In other words, for any* $X \in A$, *there is a worst-case scenario for* X *which is an extreme scenario*.

Thus, in the context of subset-type problems, for Problem RELROB it is sufficient to consider only extreme scenarios. It is well known (and obvious) that the same property holds also for Problem ABSROB. The set of extreme scenarios is finite, but its cardinality is exponential in n.

3.2. Computing the worst-case relative regret

For $X \in A$ and $\mu \geqslant 1$, consider scenario $s'(X,\mu)$ defined by

$$w_e^{s'(X,\mu)} = \begin{cases} \mu w_e^- & \text{when } e \in E\setminus X, \\ \mu w_e^- + w_e^+ - w_e^- & \text{when } e \in X. \end{cases}$$

Lemma 4. *If Y is an optimal solution to Problem* OPT$(s'(X,\mu))$, *then* $(s(X,Y),Y)$ *is an optimal solution to Problem* RELAX(X,μ), *and* $\hat{z}^*(X,\mu) = \sum_{e\in X} w_e^+ - F^*(s'(X,\mu))$.

Proof. According to Lemma 3(a), Problem RELAX(X,μ) can be written as

$$\begin{aligned} &\max_{Y\in A}\left[\sum_{e\in X\setminus Y} w_e^+ + \sum_{e\in X\cap Y} w_e^-(1-\mu) - \sum_{e\in Y\setminus X} \mu w_e^-\right] \\ &= \max_{y=\{y_e,e\in E\}\in \mathrm{CH}(A)}\left[\sum_{e\in X}[w_e^+(1-y_e)+w_e^-(1-\mu)y_e] - \sum_{e\in E\setminus X}\mu w_e^- y_e\right] \\ &= \sum_{e\in X} w_e^+ - \min_{y=\{y_e,e\in E\}\in \mathrm{CH}(A)} \sum_{e\in E} w_e^{s'(X,\mu)} y_e. \end{aligned}$$

Observe that $w_e^{s'(X,\mu)} \geqslant 0$ for any $e \in E$. The statement of the lemma follows immediately. □

Corollary 2. *For subset-type combinatorial optimization problems, Problem* RELAX(X,μ) *can be solved with the same order of complexity as Problem* OPT(s).

Let $W = \max_{e\in E} w_e^+$. Lemma 3(b) implies that the optimal objective value of Problem RELREGR(X) is a rational number whose numerator and denominator are not greater than nW. According to Lemma 1, solving Problem RELAX(X,μ) for some $\mu \geqslant 1$ allows us to know whether $\mu < Z(X)$ or $\mu = Z(X)$ or $\mu > Z(X)$. Taking into account Lemma 4 and its Corollary, we have that Problem RELREGR(X) can be solved using binary search with $\mathrm{O}(\log(n+W))$ iterations (because $\mathrm{O}(\log(nW)) = \mathrm{O}(\log n + \log W) = \mathrm{O}(\log(n+W))$), where at each iteration an instance of Problem OPT(s) is solved. We obtain

Lemma 5. *Problem* RELREGR(X) *can be solved in time* $\mathrm{O}(\log(n+W))\cdot$(*complexity of Problem* OPT(s)).

If Problem OPT(s) is polynomially solvable, then the order of complexity in Lemma 5 is polynomial as well; however, it is not strongly polynomial even if there is a strongly polynomial algorithm for Problem OPT(s), because W is in the complexity bound. To obtain a strongly polynomial algorithm if it is needed (assuming that a strongly polynomial algorithm exists for Problem OPT(s)), one can use Megiddo's parametric approach [20] (instead of binary search) which has become a classical tool in theoretical computer science and computational geometry. Descriptions of the parametric approach can be found in [20,25]. For our purposes, it is sufficient to know the following. Suppose that there is a function $Q(\theta)$ of a variable θ, such that for some unknown value θ^*, $Q(\theta) \geqslant 0$ for all $\theta \geqslant \theta^*$ and $Q(\theta) < 0$ for all $\theta < \theta^*$. Suppose that there is a sequential algorithm A1 with time complexity T_{A1}, which, given a value of θ, obtains the sign of $Q(\theta)$. Suppose that there is also a parallel algorithm A2 which, given a value of θ, obtains the sign of $Q(\theta)$ using P_{A2} parallel processors and time T_{A2}. Suppose that algorithm A2 uses only additions, subtractions, comparisons, and multiplications by constants that do not depend on θ. Then, Megiddo's technique allows us to obtain a sequential algorithm A3 that finds value θ^* in time $T_{\mathrm{A3}} = \mathrm{O}(T_{\mathrm{A2}} \cdot P_{\mathrm{A2}} + T_{\mathrm{A1}} \cdot T_{\mathrm{A2}} \cdot \log P_{\mathrm{A2}})$. Cole [10] suggested a modification of the parametric search technique that reduces the order of complexity to $T_{\mathrm{A3}} = \mathrm{O}(T_{\mathrm{A2}} \cdot P_{\mathrm{A2}} + T_{\mathrm{A1}} \cdot T_{\mathrm{A2}})$. Cole's improvement is somewhat less general than the original Megiddo's technique, but is still applicable to a wide range of problems, for example, to problems whose decision procedure is based on sorting.

We can define function

$$Q(\mu)=\begin{cases}1 & \text{if } \mu\geqslant Z(X);\\ 0 & \text{if } \mu< Z(X).\end{cases}$$

Then according to Lemmas 1 and 4, obtaining the sign of $Q(\mu)$ at some $\mu\geqslant 1$ can be done by solving an instance of Problem OPT(s), and $Z(X)$ can be found using the parametric approach if there are appropriate algorithms for Problem OPT(s). (We note that the parametric approach was originally developed for problems with ratio objective functions [19].) Let us consider two examples.

Example 1. Suppose that A is the set of all p-element subsets of E for some $p<n$. Then, Problem OPT(s) reduces to selecting p smallest numbers in $\{w_e^s, e\in E\}$ which can be done in $T_{\mathrm{A1}}=\mathrm{O}(n)$ sequential time, or in $T_{\mathrm{A2}}=\mathrm{O}(\log n)$ parallel time using $P_{\mathrm{A2}}=\mathrm{O}(n/\log n)$ processors [26]. Requirements of Megiddo's technique and Cole's improvement are met; therefore, using the (improved) Megiddo's parametric search, Problem RELREGR(X) can be solved in $\mathrm{O}(T_{\mathrm{A2}}\cdot P_{\mathrm{A2}}+T_{\mathrm{A1}}\cdot T_{\mathrm{A2}})=\mathrm{O}(n\log n)$ time.

Example 2. *Minimum spanning tree*: Suppose that A is the set of all spanning trees for a connected network with n edges and m nodes. Then, Problem OPT(s) is the minimum spanning tree problem. Prim's or Kruskal's algorithm can solve this problem sequentially in $T_{\mathrm{A1}}=\mathrm{O}(n+m\log m)$ time [1]. The recent parallel algorithm of Chong et al. [9] can solve this problem in $T_{\mathrm{A2}}=\mathrm{O}(\log n)$ time using $P_{\mathrm{A2}}=n+m$ processors [9], and the algorithm satisfies the conditions of the Megiddo's technique and Cole's improvement; therefore, Problem RELREGR(X) can be solved in $\mathrm{O}(T_{\mathrm{A2}}\cdot P_{\mathrm{A2}}+T_{\mathrm{A1}}\cdot T_{\mathrm{A2}})=\mathrm{O}[(n+m\log m)\log n]$ time.

3.3. Mixed-integer linear programming formulation

Lemma 6. *Problem* ROBRELAX(μ) *can be formulated as follows*:

$$\min_{x\in\mathrm{CH}(A)}\left[\sum_{e\in E} w_e^+x_e+\max_{y\in\mathrm{CH}(A)}\sum_{e\in E}(-\mu w_e^- -(w_e^+-w_e^-)x_e)y_e\right]. \tag{3}$$

Proof. The statement of the lemma follows from Lemma 4. □

Suppose that Problem OPT(s) can be formulated as a linear programming problem of the type

$$\min\left\{\sum_{e\in E} w_e^s y_e \mid B_1y+B_1'y'\leqslant b_1\right\}, \tag{P1}$$

where B_1, B_1', b_1 are matrices and vector of appropriate dimensions, $y=\{y_e, e\in E\}$ and y' are vectors of main and auxiliary variables, respectively. (That is, the optimum objective values of (P1) and Problem OPT(s) are equal.) We will refer to this assumption as to *Assumption LP*. Many subset-type combinatorial optimization problems that can be solved in polynomial time (e.g. the shortest path problem, the minimum spanning tree problem, the assignment problem) can be formulated as linear programs of the type (P1). Let (P1′) be the equivalent (up to the change of sign of the optimal objective function value) maximization version of Problem (P1):

$$\max\left\{\sum_{e\in E} -w_e^s y_e \mid B_1y+B_1'y'\leqslant b_1\right\}. \tag{P1$'$}$$

Then the internal maximization problem in (3) (considering x as fixed) can be viewed as problem (P1') with $w_e^s=\mu w_e^- + (w_e^+-w_e^-)x_e$, $e\in E$ (we will call this problem (P1′(x,μ))), and can be equivalently replaced with its dual which is a minimization linear programming problem; thus, we obtain an equivalent formulation for Problem ROBRELAX(μ)

$$\min_{x\in\mathrm{CH}(A)}\ \min_{\lambda\in\Lambda(x,\mu)}\left[\lambda b_1+\sum_{e\in E} w_e^+x_e\right], \tag{4}$$

where λ is the vector of dual variables for $(\mathrm{P1}'(x,\mu))$ and $\Lambda(x,\mu)$ is the polyhedron defined by the constraints of the dual problem to $(\mathrm{P1}'(x,\mu))$; we will call the set of these constraints $(\mathrm{D1}(x,\mu))$. Notice that the constraints $(\mathrm{D1}(x,\mu))$ are linear in x and μ; the only terms in these constraints that contain μ and x are the right-hand sides $(-\mu w_e^- - (w_e^+ - w_e^-)x_e)$. Also, the objective of (4) is linear (in contrast with the objective of (3)).

Suppose also that $\mathrm{CH}(A)$ can be described by means of mixed-integer linear constraints

$$B_2 x + B_2' x' \leqslant b_2, \quad x \in \{0,1\}^n \tag{P2}$$

(where x' is a vector of continuous auxiliary variables, B_2, B_2', b_2 are matrices and vector of appropriate dimensions), in the sense that the set of x-parts of all feasible solutions to (P2) is exactly the set $\mathrm{CH}(A)$. We will refer to this assumption as to *Assumption MILP*. Usually, feasible sets of subset-type combinatorial optimization problems have formulations of the type (P2). Then according to (4), Problem ROBRELAX(μ) can be formulated as a minimization mixed-integer linear programming problem with the objective function $\lambda b_1 + \sum_{e\in E} w_e^+ x_e$ and constraints $(\mathrm{D1}(x,\mu))$ and (P2). Moreover, using Lemmas 1, 2, we obtain the following result.

Theorem 1. *If Assumptions LP and MILP hold, Problem RELROB can be formulated as the following mixed-integer linear programming problem*:

$$\text{minimize } \mu,$$

subject to constraints $(\mathrm{D1}(x,\mu))$, (P2), *and* $\lambda b_1 + \sum_{e\in E} w_e^+ x_e \leqslant 0$. *This problem has* $|E|$ *boolean variables* x_e, $e \in E$; *other variables* (μ, λ, x') *are continuous*.

The development of the mixed-integer linear programming formulation was inspired by the mixed-integer linear programming formulation of Problem ABSROB for the minimum spanning tree problem in [28].

Example 3. *Uniform matroid*: Suppose that A is the set of all p-element subsets of E for some p, $1 \leqslant p < n$. From the point of view of the underlying combinatorial structure, A is the set of bases of a uniform matroid of rank p on the ground set E [27]. Problem $(\mathrm{P1}'(x,\mu))$ in this context is

$$\max \sum_{e\in E} (-\mu w_e^- - (w_e^+ - w_e^-)x_e) y_e;$$

$$\sum_{e\in E} y_e = p, \quad 0 \leqslant y_e \leqslant 1, \quad e \in E.$$

The problem dual to $(\mathrm{P1}'(x,\mu))$ is

$$\min \left(\sum_{e\in E} \lambda_e + p\lambda' \right);$$

$$\lambda_e + \lambda' \geqslant -\mu w_e^- - (w_e^+ - w_e^-)x_e, \quad \lambda_e \geqslant 0, \quad e \in E. \tag{5}$$

Constraints (5) correspond to $(\mathrm{D1}(x,\mu))$; notice that variable λ' is not restricted to be nonnegative. As constraints (P2), we can use

$$\sum_{e\in E} x_e = p, \quad x_e \in \{0,1\}, \quad e \in E. \tag{6}$$

Then, previous discussion implies that Problem RELROB can be formulated as the following mixed-integer linear programming problem:

$$\left\{ \text{minimize } \mu \mid \text{subject to constraints (5), (6), and } \sum_{e\in E} \lambda_e + p\lambda' + \sum_{e\in E} w_e^+ x_e \leqslant 0 \right\}.$$

Using this mixed-integer linear programming formulation, in Section 3.6 we develop a fast polynomial algorithm for Problem RELROB on a uniform matroid.

Example 4. *Minimum spanning tree*: Suppose that A is the set of all spanning trees of a connected network G with the set of edges E, $|E| = n$, and the set of nodes V, $|V| = m$. Then Problem OPT(s) is the minimum spanning tree problem (MST). In [17], some network design models of the MST are presented, where flow needs to be sent between the nodes of the network, and some edges need to be installed to carry the flow. The directed multicommodity flow model of the MST from [17] can straightforwardly be represented in the form (P1), and the single commodity flow model of the MST from [17] is of the form (P2); using these models as (P1) and (P2), we obtain a mixed-integer linear programming formulation of Problem RELROB with n boolean variables and O(nm) continuous variables and O(nm) constraints. To save space, we omit the straightforward but lengthy details. The directed multicommodity flow model and the single commodity flow model were used in a similar way in [28] to obtain a mixed-integer linear programming formulation for Problem ABSROB for MST.

Example 5. *The assignment problem*: Suppose that E is the set of edges of a complete bipartite graph G with parts V_1, V_2, $|V_1| = |V_2| = m$, and suppose that A is the set of perfect matchings between elements of V_1 and V_2. Then Problem OPT(s) is the assignment problem. Let us introduce boolean variable x_{ij} for each arc $(i, j) \in E$; the arc connects node i of V_1 to node j of V_2. The set of characteristic vectors of possible perfect matchings is the set of feasible vectors for the constraints

$$\sum_{1 \leqslant i \leqslant m} x_{ij} = 1, \quad j \in [1 : m], \tag{7}$$

$$\sum_{1 \leqslant j \leqslant m} x_{ij} = 1, \quad i \in [1 : m], \tag{8}$$

$$x_{ij} \in \{0, 1\}, \quad i, j \in [1 : m]. \tag{9}$$

These constraints can be used as (P2). Constraints (7)–(8) together with the requirement that the variables be nonnegative can be used as constraints of (P1), because the extreme points of the polyhedron defined by these constraints are integral. Introducing dual variable λ_j for the constraint (7) corresponding to node j of V_2, and dual variable λ'_i for the constraint (8) corresponding to node i of V_1, and following the discussion that led to Theorem 1, we obtain the following formulation of Problem RELROB for the assignment problem:

minimize μ,

subject to constraints

$$\sum_{i \in [1:m]} \lambda'_i + \sum_{j \in [1:m]} \lambda_j + \sum_{i,j \in [1:m]} w^+_{ij} x_{ij} \leqslant 0,$$

$$\lambda'_i + \lambda_j \geqslant -\mu w^-_{ij} - (w^+_{ij} - w^-_{ij}) x_{ij}, \quad i, j \in [1 : m]$$

and constraints (7)–(9).

3.4. Relationship between the complexities of Problem RELROB and Problem ABSROB

In this subsection, we show that for a broad class of subset-type problems, Problem RELROB is at least as hard as Problem ABSROB. Specifically, we show that under some mild conditions, Problem ABSROB can be reduced to solving Problem RELROB on a slightly modified instance. As an illustration of this general result, we prove that the interval data minmax relative regret spanning tree and shortest path problems are strongly NP-hard.

An instance of a subset-type problem is defined by specifying the ground set E, the set of feasible solutions A, and the uncertainty intervals $[w^-_e, w^+_e]$, $e \in E$. Let $W = \max_{e \in E} w^+_e$.

Theorem 2. *Suppose that for the subset-type problem under consideration, there is a transformation T that, given an instance I, transforms it into an instance I' such that there is one-to-one correspondence between feasible solutions X of the instance I and feasible solutions X' of the instance I', and between extreme scenarios s of the instance I and extreme scenarios s' of the instance I' (the correspondence will also be denoted as T, i.e. $X' = T(X)$, $s' = T(s)$),* *and such that for any feasible solution X and an extreme scenario s of the instance I, for the corresponding feasible solution* $X' = T(X)$ *and the corresponding extreme scenario* $s' = T(s)$ *of the instance* I', $F(s', X') = F(s, X) + M$, *where M is an integer constant such that* $M > (Wn)^2$ *and W is defined by the data of the instance I. In other words, the transformation* $T : I \to I'$ *increases value* $F(s, X)$ *for all pairs* (s, X) *by the same constant M. Then, if* X' *is an optimal solution to Problem RELROB for the instance* I', *then* $X = T^{-1}(X')$ *is an optimal solution to Problem ABSROB for the instance I.*

Proof. It is sufficient to prove that for any feasible solutions X_1, Y_1, X_2, Y_2, and extreme scenarios s_1, s_2 for the instance *I*, the inequality

$$F(s_1, X_1) - F(s_1, Y_1) > F(s_2, X_2) - F(s_2, Y_2) \tag{10}$$

implies the inequality

$$\frac{F(s_1, X_1) + M}{F(s_1, Y_1) + M} > \frac{F(s_2, X_2) + M}{F(s_2, Y_2) + M}. \tag{11}$$

For convenience, let us denote $Q_1 = F(s_1, X_1)$, $R_1 = F(s_1, Y_1)$, $Q_2 = F(s_2, X_2)$, $R_2 = F(s_2, Y_2)$. Observe that values Q_1, R_1, Q_2, R_2 are integer, since s_1, s_2 are extreme scenarios. Then

$$\frac{Q_1 + M}{R_1 + M} - \frac{Q_2 + M}{R_2 + M} = \frac{M(Q_1 + R_2 - R_1 - Q_2) + Q_1 R_2 - Q_2 R_1}{(R_1 + M)(R_2 + M)}. \tag{12}$$

If (10) holds, then the numerator of the right side of (12) is positive, because $Q_1 + R_2 - R_1 - Q_2 \geqslant 1$ according to (10), and $M > Q_2 R_1$ according to the definition of *M*; this implies (11). □

Theorem 2 shows that, speaking informally, existence of an efficient algorithm for Problem RELROB implies existence of an efficient algorithm for Problem ABSROB, if the transformation specified in the theorem exists and is computable efficiently. The theorem allows to prove NP-hardness results for Problem RELROB, if such results are known for Problem ABSROB. To illustrate this, we prove that the interval data minmax relative regret versions of the minimum spanning tree and the shortest path problems are strongly NP-hard.

Consider the minimum spanning tree problem on a connected undirected network *N* with the set of edges *E*; that is, *A* is the set of all spanning trees of *N*. Consider the following transformation of the network *N*: add an additional node a' and connect it to some node *a* of *N* with an additional edge $e' = (a, a')$ with a "certain" weight $M = (Wn)^2 + 1$ (that is, $w^-_{e'} = w^+_{e'} = M$). This transformation clearly satisfies the conditions of Theorem 2; therefore, an optimal solution to Problem RELROB on the modified network immediately gives us an optimal solution to Problem ABSROB on the original network. In [2,6], it is proven (independently) that Problem ABSROB is strongly NP-hard for the minimum spanning tree problem; therefore, Problem RELROB is strongly NP-hard too.

Consider now the shortest path problem on a network $\tilde{N}$ (directed or undirected); that is, *A* is the set of simple paths from a specified source node *a* to a specified sink node *b*. In [6], it is proven that Problem ABSROB is strongly NP-hard for the shortest path problem (weak NP-hardness is independently proven in [29]). Using Theorem 2, we have that Problem RELROB is also strongly NP-hard (one can use the transformation that adds an additional node a' and connects it to the node *a* with an arc (or edge) (a', a) of "certain" length $M = (Wn)^2 + 1$, and makes the node a' a new source node).

Consider the assignment problem on a complete bipartite network with parts V_1, V_2 of equal cardinalities, $|V_1| = |V_2| = n$. In [14], it is proven that Problem ABSROB is NP-hard for the assignment problem. Using Theorem 2, we have that Problem RELROB is also NP-hard (since all feasible solutions have the same cardinality *n*, one can use the transformation that adds the same large constant $M' = W^2 n + 1$ to all bounds of uncertainty intervals). We obtain

Theorem 3. *The interval data minmax relative regret shortest path and spanning tree problems are strongly NP-hard. The interval data minmax relative regret assignment problem is NP-hard.*

Remark 1. Besides the interval data problems, in minmax regret optimization, some authors consider also *discrete-scenario* problems, where the set *S* of possible scenarios is finite and is described by listing explicitly all possible scenarios as a part of the input. Discrete-scenario minmax regret versions of the shortest path, minimum spanning tree and assignment problems are also known to be NP-hard [15].

3.5. The case of a small number of nondegenerate uncertainty intervals

The uncertainty interval $[w_e^-, w_e^+]$ for an $e \in E$ is called *nondegenerate* if $w_e^- < w_e^+$, and is called *degenerate* if $w_e^- = w_e^+$. Let d be the number of nondegenerate uncertainty intervals. In [6], it was shown that for subset-type problems, Problem ABSROB can be solved in polynomial time whenever Problem OPT(s) is solvable in polynomial time and d is sufficiently small (asymptotically) with respect to n; namely, when d is bounded by the logarithm of a polynomial function of n. In this subsection, we show that the same result holds for Problem RELROB.

For any $X \in A$, let $s(X) = \{w_e^{s(X)}, e \in E\}$ denote the scenario defined as follows:

$$w_e^{s(X)} = \begin{cases} w_e^+ & \text{if } e \notin X, \\ w_e^- & \text{if } e \in X. \end{cases}$$

Lemma 7. *Consider an arbitrary $X \in A$. Let $\tilde{X}$ be an optimal solution to Problem* OPT($s(X)$). *Then $Z(\tilde{X}) \leqslant Z(X)$.*

Proof. The proof is identical to the proof of Lemma 2 of [6]; we state it here for completeness. It is straightforward to observe that for any scenario $s \in S$, $F(s, X) - F(s, \tilde{X}) \geqslant F(s(X), X) - F(s(X), \tilde{X})$. Since $F(s(X), X) \geqslant F(s(X), \tilde{X})$ we have that for any $s \in S$, $F(s, X) \geqslant F(s, \tilde{X})$, which implies the statement of the lemma. □

Corollary 3. *If $X^* \in A$ is an optimal solution to Problem RELROB, then any optimal solution to Problem* OPT($s(X^*)$) *is also an optimal solution to Problem RELROB.*

Corollary 4. *There exists an extreme scenario $s' \in S'$ such that any optimal solution to Problem* OPT(s') *is also an optimal solution to Problem RELROB.*

Problem RELROB can be solved using the following scheme. For each extreme scenario $s' \in S'$, find an optimal solution $\tilde{X}(s')$ to Problem OPT(s') and compute value $Z(\tilde{X}(s'))$. The best of the obtained candidate solutions will be an optimal solution to Problem RELROB. According to the discussion in Section 3.2, computing value $Z(X)$ (solving Problem RELREGR(X)) for any $X \in A$ can be done in polynomial time if Problem OPT(s) can be solved in polynomial time. Since the number of extreme scenarios is 2^d, we have the following:

Theorem 4. *If the number d of nondegenerate uncertainty intervals is fixed or is bounded by the logarithm of a polynomial function of n, and if Problem* OPT(s) *is polynomially solvable, then Problem RELROB is also polynomially solvable, with the order of complexity* O(2^d (*complexity of Problem* OPT(s)+*complexity of Problem* RELREGR(X))).

3.6. A strongly polynomial algorithm for Problem RELROB on a uniform matroid

In this subsection, we make no assumptions about the number d of nondegenerate uncertainty intervals. According to Lemma 2, solving Problem ROBRELAX(μ) for some $\mu \geqslant 1$ allows us to know whether $\mu < z^*$ or $\mu = z^*$ or $\mu > z^*$. Since the optimal objective value of Problem RELROB is a rational number whose numerator and denominator are not greater than nW, Problem RELROB can be solved using binary search in time O(log $(n + W)$)·(complexity of Problem ROBRELAX(μ)), or using Megiddo's parametric approach. As an illustration of this approach, in this subsection we develop a strongly polynomial algorithm for Problem RELROB on a uniform matroid, which is the first known polynomial algorithm for this problem.

In this subsection, we assume that A is the set of all p-element subsets of E, $1 \leqslant p < n$, that is, A is the set of bases of a uniform matroid of rank p on the ground set E [27]. In [4], a strongly polynomial algorithm for Problem ABSROB on a uniform matroid with complexity O($p^2 n$) was presented; subsequently an improved algorithm with complexity O(pn) was proposed in [11]. Below, we extend the approach of [11] to obtain efficient parallel and sequential algorithms for Problem ROBRELAX(μ); then, using the Megiddo's parametric approach, we obtain a O($np \log n$) algorithm for Problem RELROB.

For simplicity of presentation we assume $p \leqslant \frac{1}{2}n$ (an extension for the general case is straightforward).

Consider the development of the mixed-integer linear programming formulation in Section 3.3. According to the discussion in Section 3.3 and Example 3, Problem ROBRELAX(μ) can be formulated as the following mixed-integer linear programming problem:

$$\min_{\lambda,\lambda',x} \left\{ \sum_{e\in E} \lambda_e + p\lambda' + \sum_{e\in E} w_e^+ x_e \mid \text{subject to the constraints (5), (6)} \right\}. \tag{13}$$

It is easy to observe that in an optimal solution to (13)

$$\lambda_e = \max\{0, -\mu w_e^- - (w_e^+ - w_e^-)x_e - \lambda'\}, \quad e \in E. \tag{14}$$

Taking into account (6), we observe that

$$p\lambda' = \sum_{e\in E} x_e \lambda' \tag{15}$$

for any solution that satisfies (6). After substituting (14) and (15) in (13) and straightforward algebraic simplifications, we obtain a new formulation of Problem ROBRELAX(μ):

$$\min_{\lambda',x} \left\{ \sum_{e\in E} \max\{(\lambda' + w_e^+)x_e, (-\mu w_e^- - \lambda') + (w_e^- + \lambda')x_e\} \mid \text{subject to constraints (6)} \right\}. \tag{16}$$

Finally, (16) can be rewritten as

$$\hat{z}^*(\mu) = \min_{\lambda',x} \left\{ \sum_{e\in E} (a_e(\lambda')x_e + b_e(\lambda')) \mid \text{subject to constraints (6)} \right\}, \tag{17}$$

where $a_e(\lambda') = \max\{\lambda' + w_e^+, w_e^-(1-\mu)\} - \max\{0, -\mu w_e^- - \lambda'\}$, $b_e(\lambda') = \max\{0, -\mu w_e^- - \lambda'\}$ (this can be easily verified by considering the expression inside the maximization operation in the objective of (16) separately for $x_e = 0$ and $x_e = 1$).

Let $e_1, \ldots, e_n$ be the elements of E indexed according to nondecreasing values of w_e^-, that is, $w_{e_1}^- \leqslant w_{e_2}^- \leqslant \cdots \leqslant w_{e_n}^-$; and let $e^1, \ldots, e^n$ be the elements of E indexed according to nondecreasing values of $\mu w_e^- + w_e^+ - w_e^-$. Let

$$T(\mu) = \{-\mu w_e^-, e \in \{e_p, e_{p+1}, \ldots, e_{2p}\}\} \cup \{-\mu w_e^- - w_e^+ + w_e^-, e \in \{e^1, e^2, \ldots, e^p\}\}$$

(notice that $|T(\mu)| \leqslant 2p + 1$).

Lemma 8. *There is an optimal solution* (λ'^*, x^*) *to* (17) *such that* $\lambda'^* \in T(\mu)$.

Proof. By considering the formulation (13), constraints (5) and expression (14), we can observe that there is an optimal solution $(\lambda^*, \lambda'^*, x^*)$ to (13) where $(-\lambda'^*)$ is the pth smallest of the numbers $\mu w_e^- + (w_e^+ - w_e^-)x_e^*$, $e \in E$. This implies that

$$(-\lambda'^*) \in \{\mu w_e^-, e \in \{e_1, \ldots, e_{2p}\}\} \cup \{\mu w_e^- + w_e^+ - w_e^-, e \in \{e^1, e^2, \ldots, e^{2p}\}\}$$

and also that $(-\lambda'^*) \geqslant \mu w_{e_p}^-$ and $(-\lambda'^*) \leqslant \mu w_{e^p}^- + w_{e^p}^+ - w_{e^p}^-$; therefore $\lambda'^* \in T(\mu)$. □

Observe that if λ' is fixed, then (17) reduces to selecting p smallest numbers in the set $\{a_e(\lambda'), e \in E\}$, which can be done in O(n) serial time (or in O($\log n$) parallel time using O($n/\log n$) processors [26]). Lemma 8 implies that it is sufficient to consider only O(p) possible values for λ'. We obtain

Theorem 5. *Problem* ROBRELAX(μ) *can be solved in* O(np) *serial time, or in* O($\log n$) *sequential time using* O($np/\log n$) *processors.*

Theorem 5 generalizes the corresponding result of Conde [11] for Problem ABSROB (that corresponds to $\mu = 1$); the development of the formulation (17) and Lemma 8 are inspired by [11].

Using Megiddo's parametric search with Cole's improvement and Lemma 2, we obtain

Theorem 6. *Problem* RELROB *can be solved in* $O(np \log n)$ *time*.

4. Total flow-time scheduling problem

Suppose there is a set J of jobs that have to be processed on a single machine, $|J| = n$, $n \geqslant 2$. The machine cannot process more than one job at any time. Let the set of feasible solutions A be the set of all possible orderings of the jobs from J, $|A| = n!$; elements of A will be called *permutations*. For any job $j \in J$, its processing time is uncertain and belongs to the corresponding uncertainty interval $[w_j^-, w_j^+]$, $0 < w_j^- \leqslant w_j^+$, w_j^-, w_j^+ are integer. A *scenario* is an assignment of values to processing times of jobs. For any scenario $s = \{w_j^s, j \in J\}$ and any permutation $X = (j_1, j_2, \ldots, j_n) \in A$, the corresponding *total flow time* is

$$F(s, X) = w_{j_1}^s \cdot n + w_{j_2}^s \cdot (n-1) + \cdots + w_{j_{n-1}}^s \cdot 2 + w_{j_n}^s.$$

The set S of possible scenarios is the Cartesian product of the uncertainty intervals. Problem OPT(s) in this context is the problem of minimizing the total flow time and can be solved in $O(n \log n)$ time by ordering the jobs according to nondecreasing values of processing times w_j^s (the SPT rule).

4.1. Computing the worst-case relative regret

Let us fix a permutation $X \in A$; suppose for simplicity of presentation that the jobs are identified with their positions in X, that is, for a job j exactly $j-1$ jobs precede j in X. Below, we show that value $Z(X)$ can be computed in polynomial time.

Consider a complete bipartite graph $G = (V, U, L)$, with parts V, U and the set of edges L, where $V = \{v_1, \ldots, v_n\}$, $U = \{u_1, \ldots, u_n\}$. Let $\mu \geqslant 1$ be fixed; assign weights to the edges of L as follows. An edge $\{v_j, u_i\}$ receives weight $w_j^+(\mu i - j - (\mu - 1)(n+1))$ if $(\mu i - j - (\mu - 1)(n+1)) > 0$, and weight $w_j^-(\mu i - j - (\mu - 1)(n+1))$ if $\mu i - j - (\mu - 1)(n+1) \leqslant 0$. Consider the maximum weight perfect matching problem on the defined bipartite graph (also known as the assignment problem); denote it Problem MATCH.

Theorem 7. *The optimum objective values for Problem* MATCH *and Problem* RELAX(X, μ) *are equal*.

Proof. First, observe that there is a one-to-one correspondence between the permutations from A and perfect matchings in G. For a perfect matching $M = \{(v_{j_i}, u_i), i \in [1:n]\}$, the corresponding permutation is $Y = \{j_1, j_2, \ldots, j_n\}$. Next, observe that $\max_{s \in S} (F(s, X) - \mu F(s, Y))$ is exactly the weight of the perfect matching M. Indeed, for $s = (w_1^s, \ldots, w_n^s)$, $F(s, X) - \mu F(s, Y) = \sum_{i=1}^n (n - j_i + 1) w_{j_i}^s - \sum_{i=1}^n \mu(n - i + 1) w_{j_i}^s = \sum_{i=1}^n w_{j_i}^s [\mu i - j_i - (\mu - 1)(n+1)]$. The statement of the theorem follows immediately. □

It is straightforward to observe that for Problem RELAX(X, μ), and, therefore, for Problem RELREGR(X), it is sufficient to consider only extreme scenarios. Since for any extreme scenario $s \in S'$ and any $X \in A$, $F(s, X) \leqslant n^2 W$, where $W = \max_{i \in [1:n]} w_i^+$, we have that the optimal objective value of Problem RELREGR(X) is a rational number whose numerator and denominator are not greater than $n^2 W$. According to Lemma 1, solving Problem RELAX(X, μ) for some $\mu \geqslant 1$ allows us to know whether $\mu < Z(X)$ or $\mu = Z(X)$ or $\mu > Z(X)$. Thus, Problem RELREGR(X) can be solved using binary search with $O(\log(n + W))$ iterations, where at each iteration an assignment problem is solved. If, for example, the standard Hungarian $O(n^3)$ algorithm [24] is used for the assignment problem, then we obtain an $O(n^3 \log(n + W))$ algorithm for Problem RELREGR(X). We have

Theorem 8. *For any permutation X, value* $Z(X)$ *can be computed in* $O(n^3 \log(n + W))$ *time*.

4.2. Complexity of the problem

In [16], it is proven that Problem ABSROB is NP-hard. Using this result, we prove that Problem RELROB is NP-hard as well.

Theorem 9. *Problem* RELROB *is NP-hard.*

Proof. The idea of the proof is similar to the idea of the proof of Theorem 2, with some additional technical twists. We use a reduction from Problem ABSROB. Suppose we have an instance I of Problem ABSROB that is described by numbers $n, w_i^+, w_i^-, i \in [1:n]$. Then, for any $X \in A$ and $s \in S$, $F(s, X) \leqslant Wn^2$. Let us transform the instance I to an instance I' of Problem RELROB by adding a "long" job j' with $w_{j'}^- = w_{j'}^+ = W^2n^4 + W^2n^3 + 1$.

Main claim. The permutation that is obtained from the optimal solution to Problem RELROB for the instance I' by deleting the "long" job j' will be optimal for Problem ABSROB for the instance I.

Since Problem ABSROB is NP-hard [16], to prove the theorem it is sufficient to prove the main claim. Since the set of jobs with nondegenerate uncertainty intervals is the same for both instances, we can consider both instances as having the same set S of scenarios. Clearly, for any $s \in S$, in an optimal permutation for Problem OPT(s) for the instance I' the job j' is scheduled last, and in any optimal permutation for Problem RELROB for the instance I' the job j' is also scheduled last. Therefore, we can consider only permutations where the job j' is scheduled last as feasible solutions for the instance I'; denote the set of all such permutations as A'. Then we have a natural one-to-one correspondence between feasible solutions $X \in A$ for the instance I and feasible solutions $X' \in A'$ for the instance I'. It is straightforward to see that for any $X \in A$ and the corresponding $X' \in A'$, and for any $s \in S$, $F(s, X') = F(s, X) + M(s)$ for some $M(s)$ such that $W^2n^4 + W^2n^3 + 1 \leqslant M(s) \leqslant W^2n^4 + W^2n^3 + Wn + 1$.

To prove the main claim, it is sufficient to prove that for any feasible solutions X_1, Y_1, X_2, Y_2 and extreme scenarios s_1, s_2 for the instance I, and any numbers M_1, M_2 such that

$$W^2n^4 + W^2n^3 + 1 \leqslant M_i \leqslant W^2n^4 + W^2n^3 + Wn + 1, \quad i = 1, 2, \tag{18}$$

the inequality

$$F(s_1, X_1) - F(s_1, Y_1) > F(s_2, X_2) - F(s_2, Y_2) \tag{19}$$

implies the inequality

$$\frac{F(s_1, X_1) + M_1}{F(s_1, Y_1) + M_1} > \frac{F(s_2, X_2) + M_2}{F(s_2, Y_2) + M_2}. \tag{20}$$

For convenience, let us denote $Q_1 = F(s_1, X_1)$, $R_1 = F(s_1, Y_1)$, $Q_2 = F(s_2, X_2)$, $R_2 = F(s_2, Y_2)$. Observe that values Q_1, R_1, Q_2, R_2 are integer, since s_1, s_2 are extreme scenarios. Then

$$\frac{Q_1 + M_1}{R_1 + M_1} - \frac{Q_2 + M_2}{R_2 + M_2} = \frac{(Q_1 - R_1 - Q_2 + R_2)M_2 - (Q_2 - R_2)(M_1 - M_2) + Q_1R_2 - Q_2R_1}{(R_1 + M_1)(R_2 + M_2)}. \tag{21}$$

If (19) holds, then the numerator of the right side of (21) is positive, because $|M_1 - M_2| \leqslant Wn$, $|Q_2 - R_2| \leqslant Wn^2$, $Q_2R_1 \leqslant W^2n^4$, $Q_1 - R_1 - Q_2 + R_2 \geqslant 1$ according to (19) and therefore $(Q_1 - R_1 - Q_2 + R_2)M_2 \geqslant W^2n^4 + W^2n^3 + 1$. This implies (20). □

Remark 2. The discrete-scenario minmax regret version of the minimum total flow time scheduling problem, where the set S of possible scenarios is finite and is described by listing explicitly all possible scenarios as a part of the input, is also known to be NP-hard [12].

4.3. Dominance property and the case of fixed number of nondegenerate uncertainty intervals

If for some jobs j', j'', $w_{j'}^+ \leqslant w_{j''}^+$ and $w_{j'}^- \leqslant w_{j''}^-$, we say that the job j' *dominates* the job j''.

Lemma 9. *Suppose that job j' dominates job j'', $X \in A$ is an optimal permutation for Problem RELROB, and j'' precedes j' in X. Then switching the positions of the jobs j' and j'' will result in another optimal permutation for Problem* RELROB.

Proof. The proof can be conducted using an argument identical to that used in [12] to prove a similar statement for Problem ABSROB. □

Lemma 9 in combination with Theorem 8 can be used to show that Problem RELROB can be solved in polynomial time if the number d of nondegenerate uncertainty intervals is fixed. Indeed, Lemma 9 implies that we can order the jobs with degenerate uncertainty intervals according to nondecreasing values $w_j^- = w_j^+$, and consider only permutations where the positions of such jobs are consistent with the obtained order. Since there are $O(n^d)$ such permutations, and taking into account Theorem 8, we obtain

Observation. If the number d of nondegenerate uncertainty intervals is fixed, Problem RELROB can be solved in polynomial time.

5. Some concluding remarks

5.1. The case of a more general set of scenarios

Suppose that the set S is a general polyhedron P in R_+^n. Then, Lemma 1(a) implies that for the problems considered in the paper, for any $X \in A$ there is a worst-case scenario that is an extreme point of P (observe that for subset-type problems and the scheduling problem considered in Section 4 the objective of Problem RELAX(X, μ) is linear in w-variables if X, μ, Y are fixed). This implies that Problem RELREGR(X) is polynomially solvable if Problem OPT(s) is polynomially solvable and P has a polynomial number of extreme points. However, for a general polyhedron P there may be an exponential number of extreme points; we conjecture that in the case of a general polyhedron P, computing the value $Z(X)$ is NP-hard for the minimum spanning tree problem, the shortest path problem, the assignment problem, and the problem of minimizing the total flow time of n jobs. Thus, the results of Theorem 8 and Lemma 5 are due to the special structure created by the interval data representation of uncertainty (namely, S is a rectangular box in R_+^n with edges parallel to coordinate axes).

5.2. Branch-and-bound algorithms

We showed that interval data minmax relative regret versions of the minimum spanning tree problem, the shortest path problem, and the minimum total flow time scheduling problem are NP-hard, using the corresponding results for the absolute regret problems. We showed also that the value of the worst-case relative regret for a fixed feasible solution can be obtained in polynomial time for each of these three problems. The latter result can be used for developing branch-and-bound algorithms for these problems. Such algorithms can be developed similarly to branch-and-bound algorithms for the absolute regret versions of these problems in [21,22,12], with straightforward modification of lower bounds.

Acknowledgements

This research was supported by a grant from the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] R. Ahuja, T. Magnanti, J. Orlin, Network Flows, Prentice Hall, Upper Saddle River, NJ, 1993.
[2] I. Aron, P. Van Hentenryck, On the complexity of the robust spanning tree problem with interval data, Oper. Res. Lett. 32 (2004) 36–40.
[3] I. Averbakh, Minmax regret solutions for minimax optimization problems with uncertainty, Oper. Res. Lett. 27 (2) (2000) 57–65.

[4] I. Averbakh, On the complexity of a class of combinatorial optimization problems with uncertainty, Math. Programming 90 (2001) 263–272.
[5] I. Averbakh, The minmax relative regret median problem on networks, INFORMS J. Comput., to appear.
[6] I. Averbakh, V. Lebedev, Interval data minmax regret network optimization problems, Discrete Appl. Math. 138 (2004) 289–301.
[7] A. Ben-Tal, A. Nemirovski, Robust convex optimization, Math. Oper. Res. 23 (4) (1998) 769–805.
[8] D. Bertsimas, M. Sim, Robust discrete optimization and network flows, Math. Programming B 98 (2003) 43–71.
[9] K.W. Chong, Y. Han, T.W. Lam, Concurrent threads and optimal parallel minimum spanning trees algorithm, J. ACM 48 (2) (2001) 297–323.
[10] R. Cole, Slowing down sorting networks to obtain faster sorting algorithms, J. Assoc. Comp. Mach. 34 (1987) 200–208.
[11] E. Conde, An improved algorithm for selecting p items with uncertain returns according to the minmax-regret criterion, Math. Programming 100 (2004) 345–353.
[12] R. Daniels, P. Kouvelis, Robust scheduling to hedge against processing time uncertainty in single-stage production, Management Sci. 41 (2) (1995) 363–376.
[13] O.E. Karasan, M. Pinar, H. Yaman, The robust shortest path problem with interval data, Working Paper, Department of Industrial Engineering, Bilkent University, 06533 Bilkent, Ankara, Turkey, 2001.
[14] A. Kasperski, P. Zielinski, Minimizing maximal regret in the linear assignment problems with interval costs, Preprint 007, Instytut Matematyki Politechnika Wroslawska, Wroslaw, Poland, 2004. Available at <http://www.im.pwr.wroc.pl/∼pziel/publications/reports/dam.pdf>.
[15] P. Kouvelis, G. Yu, Robust Discrete Optimization and its Applications, Kluwer Academic Publishers, Boston, 1997.
[16] V. Lebedev, I. Averbakh, Complexity of minimizing the total flow time with interval data and minmax regret criterion, Discrete Appl. Math., to appear.
[17] T.L. Magnanti, L. Wolsey, Optimal trees, in: M.O. Ball et al. (Ed.), Network Models, Handbook in Operations Research and Management Science, vol. 7, North-Holland, Amsterdam, 1995, pp. 503–615.
[18] H. Mausser, M. Laguna, Minimizing the maximum relative regret for linear programmes with interval objective function coefficients, J. Oper. Res. Soc. 50 (1999) 1063–1070.
[19] N. Megiddo, Combinatorial optimization with rational objective functions, Math. Oper. Res. 4 (1979) 414–424.
[20] N. Megiddo, Applying parallel algorithms in the design of serial algorithms, J. Assoc. Comp. Mach. 30 (1983) 852–865.
[21] R. Montemanni, L.M. Gambardella, A branch-and-bound algorithm for the robust spanning tree problem with interval data, European J. Oper. Res. 161 (2005) 771–779.
[22] R. Montemanni, L.M. Gambardella, A.E. Rizzoli, A.V. Donati, A branch-and-bound algorithm for the robust shortest path problem with interval data, Oper. Res. Lett. 32 (2004) 225–232.
[23] J.M. Mulvey, R.J. Vanderbei, S.A. Zenious, Robust optimization of large scale systems, Oper. Res. 43 (2) (1995) 264–281.
[24] Ch.H. Papadimitriou, K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Prentice Hall, Englewood Cliffs, NJ, 1982.
[25] T. Radzik, Fractional combinatorial optimization, in: D.-Z. Du, P. Pardalos (Eds.), Handbook of Combinatorial Optimization, vol. 1, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998, pp. 429–478.
[26] in: J. Reif (Ed.), Synthesis of Parallel Algorithms, Kaufmann Publishers, California, 1993.
[27] D.J. Welsh, Matroid Theory, Academic Press, London, 1976.
[28] H. Yaman, O.E. Karasan, M.C. Pinar, The robust spanning tree problem with interval data, Oper. Res. Lett. 29 (2001) 31–40.
[29] P. Zielinski, The computational complexity of the relative robust shortest path problem with interval data, European J. Oper. Res. 158 (2004) 570–576.