Computing Min-Max Regret Solutions in Possibilistic Combinatorial Optimization Problems

Adam Kasperski and Paweł Zieliński

Abstract. In this chapter we discuss a wide class of combinatorial optimization problems with a linear sum and a bottleneck cost function. We first investigate the case when the weights in the problem are modeled as closed intervals. We show how the notion of optimality can be extended by using a concept of a deviation interval. In order to choose a solution we adopt a robust approach. We seek a solution that minimizes the maximal regret, that is the maximal deviation from optimum over all weight realizations, called scenarios, which may occur. We then explore the case in which the weights are specified as fuzzy intervals. We show that under fuzzy weights the problem has an interpretation consistent with possibility theory. Namely, fuzzy weights induce a possibility distribution over the scenario set and the possibility and necessity measures can be used to extend the optimality evaluation and the min-max regret approach.

1 Introduction

In many optimization problems we seek an object composed of elements of a given set to achieve some goal. For instance, in a wide class of network problems the element set consists of all edges of a given graph and we seek an optimal path, spanning tree, cut, matching etc. in this graph. A comprehensive review of various problems of this type can be found in [1, 30, 35]. While describing a particular system we often meet some parameters associated with the elements whose values are not

Adam Kasperski
Wrocław University of Technology, Institute of Industrial Engineering and Management, Wybrzeże Wyspiańskiego 27, 50-370, Wrocław, Poland
e-mail: adam.kasperski@pwr.wroc.pl

Paweł Zieliński
Wrocław University of Technology, Institute of Mathematics and Computer Science, Wybrzeże Wyspiańskiego 27, 50-370, Wrocław, Poland
e-mail: pawel.zielinski@pwr.wroc.pl

W.A. Lodwick & J. Kacprzyk (Eds.): Fuzzy Optimization, STUDFUZZ 254, pp. 287–312.
springerlink.com

precisely known. For instance, in a traffic network the traveling times between distinct points are rarely known in advance and this uncertainty must be taken into account while choosing a path in this network. In order to model the risk connected with imprecise parameters a stochastic approach can be adopted (see. e.g. [21]). For every parameter a probability distribution for its values is specified and, typically, the expected cost of a solution is minimized. The stochastic approach has several drawbacks. Namely, it may be hard or expensive to estimate the probability distribution for an unknown parameter. Also, the obtained solution may be not reasonable if it is used only once because it may be poor under the worst parameter realization which may occur.

An approach which has received an increasing attention in the recent years is the one of *robust optimization*. The idea of robust approach is to find a solution that hedges against the worst realizations of parameters which may occur. A good introduction to robust optimization can be found in a book [29]. For this class of problems a part of the input is a *scenario set*, which contains all realizations of the parameters, called *scenarios*, which may occur. No probability distribution over the scenario set is specified. Then a solution is computed, which minimizes a given criterion under the worst scenario. One of the most popular methods of defining the scenario set is to specify for every parameter a closed interval, which contains all its possible values. The scenario set is then the Cartesian product of all the uncertainty intervals. In order to choose a solution a *maximal regret* criterion can be used. The maximal regret is the maximal difference between the cost of a solution and the optimum over all scenarios. It was first suggested as a criterion for choosing a decision under uncertainty by Savage [39]. A deep discussion on the maximal regret can also be found in a book [31].

The min-max regret approach to combinatorial optimization problems with interval data has attracted a considerable attention recently. A recent survey of the known results in this area can be found in [2, 22]. It turns out that the complexity of the min-max regret problem strongly depends on the choice of the cost function in its deterministic version. Under a *bottleneck cost* the min-max regret problem is polynomially solvable if only the deterministic problem is polynomially solvable [7]. However, under a more popular *linear sum cost*, the min-max regret versions of the shortest path [7, 24, 42], the minimum spanning tree [6, 7], the minimum assignment [3] and the minimum $s-t$ cut [4] turned out to be NP-hard. A polynomial algorithm is known for the min-max regret selecting items problem [8, 12], which is a special case of the 0-1 Knapsack with unit capacities of all items. Some approximation algorithms for the class of min-max regret problems with the linear sum cost can be found in [23, 26].

In this chapter we show how the known min-max regret approach can be extended. The key idea is to model the imprecision using *fuzzy intervals*. A fuzzy interval can be seen as a monotone, under inclusion, family of closed intervals parametrized by the value of $\lambda \in [0,1]$. It is also a fuzzy set in the space of reals, whose membership function is a *possibility distribution* for the values of an unknown quantity. A description of possibility theory can be found in a book [14], where one can also find some methods of obtaining possibility distributions from the possessed knowledge. Fuzzy intervals allow us to define a possibility distribution over the scenario set. So now, for every scenario we can assign a real number from the interval [0,1], which says us what is the possibility that this scenario will occur. In order to choose a solution we can adopt an elegant concept proposed for fuzzy linear programming in [19, 20]. It turns out that this solution method can be viewed as a direct extension of the min-max regret approach to the fuzzy case, which additionally has a clear possibilistic interpretation. Furthermore, the fuzzy combinatorial optimization problems are easier to solve than fuzzy linear programming described in [20].

This chapter is organized as follows. First, in Section 2, we recall a formulation of a combinatorial optimization problem with deterministic weights. We describe the problems with two types of cost functions, namely the bottleneck and the linear sum ones. We also introduce the concept of a deviation, which is a distance of a solution (element) from optimality. The concept of deviation will play a central role in our analysis. In Section 3, we discuss the combinatorial optimization problems with interval weights. By extending the concept of deviation we show how the optimality of solutions and elements can be characterized and how to choose a solution. We seek a solution that minimizes the maximal regret, that is the largest deviation which may occur for this solution. We present all known complexity results for the interval problems. In Section 4, we investigate the combinatorial optimization problems with fuzzy weights. We first recall some basic notions of possibility theory. We then show how the concept of scenario set can be extended by defining a possibility distribution over all scenarios. We also introduce the concept of a fuzzy deviation and show how to characterize the optimality of solutions and elements, using possibility and necessity measures. Finally, we adopt a method of choosing a solution under fuzzy weights and we construct several methods of computing this solution.

2 Deterministic Combinatorial Optimization Problems

In this section we briefly recall a formulation of a general combinatorial optimization problem. Let $E = \{e_1, \ldots, e_n\}$ be a finite set of elements and let $\Phi \subseteq 2^E$ be a set of subsets of E called a set of *feasible solutions*. For every element $e \in E$ there is a nonnegative weight w_e, which expresses a single parameter associated with e such as cost, time, length etc. We will use $F(X)$ to denote a cost of solution $X \in \Phi$. Two types of the cost function are widely used, namely a *linear sum cost* $F(X) = \sum_{e \in X} w_e$ and a *bottleneck cost* $F(X) = \max_{e \in X} w_e$. The deterministic combinatorial optimization problem P is the following one:

$$\text{P:} \quad \min_{X \in \Phi} F(X), \tag{1}$$

where $F(X)$ is either the linear sum or the bottleneck cost. So, an instance of the problem is specified by a triple $(E, \Phi, \mathbf{w})$, where $\mathbf{w}$ is a vector of element weights.

The formulation (1) encompasses a large variety of problems. In the important class of *network problems*, E is a set of edges of a given directed or undirected graph $G = (V, E)$ and Φ consists of all subsets of the edges that form some objects in G such as paths, spanning trees, matchings, cuts etc. In general (1) includes the problems, which can be formulated as 0-1 programming ones. To see this, we need to associate a binary variable $x_i \in \{0,1\}$ with every element $e_i \in E$ and describe Φ using a system of constraints involving the binary variables. Notice that some of the problems are polynomially solvable while the other ones are NP-hard. In this chapter we will assume that P is polynomially solvable. A description of such problems with both linear sum and bottleneck cost can be found for instance in books [30, 35] and in papers [10, 17, 36, 37].

In theory and practice the class of *matroidal problems* is of great importance. Recall that a *matroid* is a pair $(E, \mathscr{I})$, where E is a nonempty element set and $\mathscr{I}$ is a set of subsets of E such that $\mathscr{I}$ is closed under inclusion (if $A \in \mathscr{I}$ and $B \subseteq A$ then $B \in \mathscr{I}$) and fulfills the so-called *growth property* (if $A, B \in \mathscr{I}$ and $|A| < |B|$ then there is $e \in B \setminus A$ such that $A \cup \{e\} \in \mathscr{I}$). The maximal under inclusion elements in $\mathscr{I}$ are called *bases*. In a matroidal problem the set of feasible solutions Φ consists of all bases of a given matroid. Perhaps, the best known example of a matroidal problem is the minimum spanning tree, where E is a set of edges of a given undirected graph and $\mathscr{I}$ consists of all subsets of the edges that form acyclic subgraphs of G. Then $(E, \mathscr{I})$ is called a *graphic matroid* and its base is a spanning tree of G, so Φ contains all spanning trees of G. Another important example is the minimum selecting items problem. In this problem, E is a set of items and $\mathscr{I}$ consists of all subsets of E, whose cardinalities are less than or equal to a given number p. The system $(E, \mathscr{I})$ is the so-called *uniform matroid* and X is a base of this matroid if and only if $|X| = p$. In this case Φ contains all subsets of E, whose cardinalities are precisely p. We will see in the next sections that the particular structure of matroidal problems sometimes allows us to design efficient algorithms under uncertainty.

In the approach presented in this chapter a central role will be played by the concept of a deviation. A *deviation of solution* $X \in \Phi$ is defined as follows:

$$\delta_X = F(X) - \min_{Y \in \Phi} F(Y).$$

Hence deviation δ_X expresses a "distance" of X from optimum. Obviously X is optimal if and only if $\delta_X = 0$. A similar concept can be introduced for elements. Let $\Phi_f \subseteq \Phi$ be the set of all feasible solutions that contain element f. Then a *deviation of element* $f \in E$ is defined as follows:

$$\delta_f = \min_{Y \in \Phi_f} F(Y) - \min_{Y \in \Phi} F(Y).$$

We call element f *optimal* if $\delta_f = 0$. In other words, f is optimal if and only if it is a part of an optimal solution. The solution (element) deviation gives us an information how far from optimality a solution (element) is.

3 Combinatorial Optimization Problems with Interval Weights

In practice, precise values of the element weights in a combinatorial optimization problem may be unknown. In this section we discuss perhaps the simplest uncertainty representation, where for every unknown weight a closed interval containing all its possible values is specified. We extend the concept of deviation and we show how the optimality of a given solution or element can be characterized and how to choose a solution under interval weights.

3.1 Scenario Set

Assume that we only know that the value of the weight w_e of element $e \in E$ will fall within a closed interval $W_e = [\underline{w}_e, \overline{w}_e]$. Notice that a precise weight w_e can be modeled as a *degenerate* interval such that $\underline{w}_e = \overline{w}_e$. We assume that there is no probability distribution in W_e, $e \in E$, and all weights are unrelated, that is the value of every weight does not depend on the values of the remaining weights. A vector $S = (s_e)_{e \in E}$ such that $s_e \in W_e$ for all $e \in E$ is called a *scenario* and it represents the state of the world in which $w_e = s_e$ for all $e \in E$. A *scenario set* Γ is formed by the Cartesian product of all the uncertainty intervals, namely $\Gamma = \times_{e \in E} W_e$. Notice that our assumptions imply that for any two scenarios S_1 and S_2 it is not possible to say which one is more likely to happen. In other words, there is no probability distribution in scenario set Γ.

Among the scenarios an important role is played by the *extreme* ones, where all weights take the lower or upper bounds in their uncertainty intervals, i.e. the scenarios from the set $\times_{e \in E}\{\underline{w}_e, \overline{w}_e\}$. Let $A \subseteq E$ be a subset of the elements. In scenario S_A^+ all elements $e \in A$ have weights $\overline{w}_e$ and all the remaining elements have weights $\underline{w}_e$. In the symmetric scenario S_A^- all elements $e \in A$ have weights $\underline{w}_e$ and all the remaining elements have weights $\overline{w}_e$.

Under the interval uncertainty representation the cost of solution X depends on scenario $S \in \Gamma$ an we will denote it as $F(X,S)$. Of course, $F(X,S)$ is either the linear sum cost $\sum_{e \in X} s_e$ or the bottleneck cost $\max_{e \in X} s_e$. We will use $F^*(S)$ to denote the cost of an optimal solution under scenario S. In order to obtain $F^*(S)$ we must solve the deterministic problem (1) under the weight realization specified by scenario S. Now the solution and element deviations also depend on scenario S and we will denote them as $\delta_X(S)$ and $\delta_f(S)$, respectively.

3.2 Deviation Interval and Optimality Evaluation

Recall that in the deterministic case a deviation gives a full characterization of optimality. In the interval case the optimality can be fully characterized by the so-called *deviation interval*. For a given solution $X \in \Phi$ we define $\Delta_X = [\underline{\delta}_X, \overline{\delta}_X]$, where $\underline{\delta}_X = \min_{S\in\Gamma} \delta_X(S)$ and $\overline{\delta}_X = \max_{S\in\Gamma} \delta_X(S)$. The quantity $\overline{\delta}_X$ is called in literature the *maximal regret* of X [29] and it expresses the largest distance of X from optimality. Similarly, for a given element $f \in E$ we have $\Delta_f = [\underline{\delta}_f, \overline{\delta}_f]$, where $\underline{\delta}_f = \min_{S\in\Gamma} \delta_f(S)$ and $\underline{\delta}_f = \max_{S\in\Gamma} \delta_f(S)$.

The intervals Δ_X and Δ_f contain all values of solution and element deviations which may occur and allow us to give the following optimality characterization. We say that a solution X is *possibly optimal* if $\underline{\delta}_X = 0$ and it is *necessarily optimal* if $\overline{\delta}_X = 0$. Obviously, solution X is possibly optimal if and only if it is optimal under some scenario $S \in \Gamma$ and it is necessarily optimal if and only if it is optimal under all scenarios $S \in \Gamma$. Exactly the same optimality characterization can be given for the elements. So, we can also introduce the possibly and necessarily optimal elements using deviation intervals of elements. It is easy to check that every possibly (necessarily) optimal solution is composed of possibly (necessarily) optimal elements. However, the converse statement is not true since it is not difficult to give an example of a solution composed of possibly (necessarily) optimal elements, which is not possibly (necessarily) optimal (see [25]).

3.3 Choosing a Solution under Interval Weights

Now an important question arises which solution should be chosen under interval weights. One can simply choose a possibly optimal one. This can be done by computing an optimal solution under any particular scenario $S \in \Gamma$. This choice is optimistic because we need to assume that a good scenario will occur. However, the quality of the solution may be very poor if a bad scenario will realize. One can also try to compute a necessarily optimal solution. Indeed, such a solution is an ideal choice but, contrary to the possibly optimal solutions, it rarely exists. In other words, the necessary optimality is too strong criterion. We thus can see that in order to choose a solution, a compromise between the possible and necessary optimality is required. This compromise is achieved by computing a solution that minimizes the maximal regret $\overline{\delta}_X$, that is the largest deviation (a distance to optimality) over all scenarios. So, under the interval uncertainty representation we focus on the following optimization problem:

$$\min_{X\in\Phi} \overline{\delta}_X. \tag{2}$$

An optimal solution to (2) is called an *optimal min-max regret solution*. We get immediately that every necessarily optimal solution is an optimal min-max regret one (but the converse statement is not true). In the next two sections we will show that every optimal min-max regret solution is possibly optimal. Hence it fulfills the minimum requirement of being optimal under some scenario. In consequence, the deviation interval of an optimal min-max regret solution is of the form $[0, \overline{\delta}_X]$, where $\overline{\delta}_X$ is the smallest among all $X \in \Phi$.

3.4 Computational Properties of the Interval-Valued Problem

In this section we focus on the computational properties of problem P with interval weights. We will show that the complexity of computing deviation intervals and min-max regret solutions strongly depends on the choice of the cost function.

3.4.1 Problems with Linear Sum Cost

In this section we discuss the case when $F(X,S) = \sum_{e \in X} s_e$, so we consider a problem with the linear sum cost function. The following proposition results directly from the definition of the cost function:

Proposition 1. *For any solution $X \in \Phi$ it holds $\underline{\delta}_X = \delta_X(S_X^-)$ and $\overline{\delta}_X = \delta_X(S_X^+)$.*

If the deterministic problem P is polynomially solvable, then the deviation interval Δ_X for a given solution X can be computed in polynomial time. Hence we can also characterize efficiently the optimality of X and compute its maximal regret. This is very important property of this class of problems. It is worth pointing out that for the linear programming problem with interval objective function coefficients, computing the maximal regret of a given solution is NP-hard [9]. Proposition 1 implies the following result:

Proposition 2. *Every optimal min-max regret solution X is possibly optimal and it is composed of possibly optimal elements.*

Proof. Suppose, by contradiction, that an optimal min-max regret solution X is not possibly optimal. Then, by Proposition 1, $\underline{\delta}_X = \delta_X(S_X^-) = F(X, S_X^-) - F^*(S_X^-) > 0$. Let $Y \in \Phi$ be an optimal solution under S_X^-. Hence $F(Y, S_X^-) < F(X, S_X^-)$. Using the definition of the linear sum cost function we can see that $F(Y,S) < F(X,S)$ for all scenarios $S \in \Gamma$, so $\delta_Y(S) < \delta_X(S)$ for all $S \in \Gamma$. Finally, using again Proposition 1, we get $\overline{\delta}_Y = \delta_Y(S_Y^+) < \delta_X(S_Y^+) \leq \overline{\delta}_X$, which contradicts the assumption that X is an optimal min-max regret solution. Since X is possibly optimal it must be composed of possibly optimal elements. □

We know that a necessarily optimal solution X, i.e. such that $\overline{\delta}_X = 0$, is an optimal min-max regret one. Sometimes such a solution may exist and it can be detected by using the following result:

Theorem 1 ([23]). *Let Y be an optimal solution under scenario S such that $s_e = \frac{1}{2}(\underline{w}_e + \overline{w}_e)$ for all $e \in E$. Then there is a necessarily optimal solution if and only if Y is necessarily optimal.*

So, if problem P is polynomially solvable, then we can detect in polynomial time a necessarily optimal solution if it exists. There is also a general link between necessarily optimal elements and optimal min-max regret solutions.

Theorem 2 ([28]). *If all weight intervals are nondegenerate, then there is an optimal min-max regret solution which contains all necessarily optimal elements.*

The assumption that all weight intervals are nondegenerate is crucial. To see this, consider the minimum spanning tree problem in a connected graph $G = (V,E)$. Assume that $W_e = [1,1]$ for all $e \in E$. Of course, every element (edge) $e \in E$ is necessarily optimal but all elements do not even form a feasible solution. If there are some degenerate weights, then it can only be shown that for every necessarily optimal element f there is an optimal min-max regret solution that contains f [28].

Let us now focus on computing the deviation interval Δ_f for a given element $f \in E$. Unfortunately, this problem is much harder than computing a solution deviation interval. It is not difficult to show that $\delta_f(S)$ attains minimum and maximum in some extreme scenarios [22]. However, computing these scenarios is not trivial and algorithms for performing this task are known only for some special cases of problem P. A general result can be proven for matroidal problems:

Theorem 3 ([25]). *If* P *is a matroidal problem, then for any element* $f \in E$ *it holds* $\underline{\delta}_f = \delta_f(S^-_{\{f\}})$ *and* $\overline{\delta}_f = \delta_f(S^+_{\{f\}})$.

If P is not a matroidal problem, then computing Δ_f may be NP-hard. Specifically, if P is the shortest path, the minimum assignment or the minimum s-t cut, then computing $\underline{\delta}_f$ for a given element f is NP-hard [28]. Furthermore, for these problems even deciding whether $\underline{\delta}_f \leq 0$ is NP-complete, so the problem of asserting the possible optimality of a given element is computationally intractable. This result also means that the lower bound of an element deviation interval is hard to approximate. Interestingly, no polynomially solvable deterministic problem is known for which computing the upper bound $\overline{\delta}_f$ under interval weights is NP-hard. Apart from matroidal problems, $\overline{\delta}_f$ can be efficiently computed in the shortest path problem provided that the input graph is directed and acyclic [16].

Finally, let us focus on solving the min-max regret problem (2). Unfortunately, it turns out to be NP-hard if P is shortest path [7, 24, 42], minimum spanning tree [7, 6], minimum assignment [3] and minimum $s-t$ cut [4]. It is polynomially solvable for the minimum selecting items problem, which has a very simple combinatorial structure [7, 12]. In literature there are two general methods of solving (2). One can design a mixed integer programming model and solve it by using one of many available packages [22, 32, 40]. Alternatively, a branch and bound algorithm can be used to solve the problem [5, 33, 34]. Both techniques have appeared to be quite efficient for some problems and for a description of the results of computational tests we refer the reader to [22, 33, 34, 40].

Notice that Proposition 2 and Theorem 2 suggest a method of preprocessing a problem before solving it. Suppose that we are able to partition the set of elements into three sets, namely $E = A \cup B \cup C$, where A contains nonpossibly optimal elements, B contains necessarily optimal elements and C contains all the remaining elements (the set C contains possibly optimal elements and elements whose status is unknown). According to Proposition 2, we can remove all elements in A from E without violating optimal min-max regret solutions. Similarly, according to Theorem 2, under nondegenerate weights we can automatically add all elements from B to the constructed solution (if there are some degenerate weights, then we can add a single element from B to the constructed solution). This may significantly reduce the problem size and speed up determining of an optimal min-max regret solution.

3.4.2 Problems with Bottleneck Cost

In this section we discuss a problem with the bottleneck cost $F(X,S) = \max_{e\in X} s_e$. The following theorem suggests a method of computing the deviation interval of a specified solution:

Theorem 4. *For any solution X it holds*

$$\underline{\delta}_X = \max\{0, \max_{e\in X} \underline{w}_e - F^*(S_E^+)\}, \tag{3}$$

$$\overline{\delta}_X = \max_{e\in X} \max\{0, \overline{w}_e - F^*(S_{\{e\}}^+)\}. \tag{4}$$

Proof. The proof of equality (4) can be found in [7]. We prove equality (3). Let $S \in \Gamma$ be a scenario that minimizes the deviation, that is $\underline{\delta}_X = \delta_X(S) = F(X,S) - F^*(S)$. Since $\max_{e\in X} \underline{w}_e \leq F(X,S)$, $F^*(S_E^+) \geq F^*(S)$ and $\underline{\delta}_X \geq 0$ it follows immediately that

$$\underline{\delta}_X \geq \max\{0, \max_{e\in X} \underline{w}_e - F^*(S_E^+)\}. \tag{5}$$

It remains to show that the inequality $\leq$ also holds in (5). Let Y be an optimal solution under S_E^+ and let $g = \arg\max_{e\in Y} \overline{w}_e$. We consider two cases. (i) $\max_{e\in X} \underline{w}_e > \overline{w}_g$. Denote $h = \arg\max_{e\in X} \underline{w}_e$. Consider scenario S such that $s_e = \min\{\underline{w}_h, \overline{w}_e\}$ for all $e \in X$ and $s_e = \overline{w}_e$ for all $e \in E\setminus X$. Since $\underline{w}_h \geq \underline{w}_e$ for all $e \in X$, $S \in \Gamma$. It is easy to check that $F(X,S) = \underline{w}_h$ and $F^*(S) = F(S_E^+)$. Hence $\underline{\delta}_X \leq \delta_X(S) = \max_{e\in X} \underline{w}_e - F^*(S_E^+) \leq \max\{0, \max_{e\in X} \underline{w}_e - F^*(S_E^+)\}$, which together with (5) yield (3). (ii) $\max_{e\in X} \underline{w}_e \leq \overline{w}_g$. Consider scenario S such that under this scenario all elements $e \in E\setminus X$ have weights $\overline{w}_e$ and all the elements $e \in X$ have weights $\min\{\overline{w}_e, \overline{w}_g\}$. Since $\underline{w}_e \leq \overline{w}_g$ for all $e \in X$, $S \in \Gamma$. One can easily verify that X is optimal under S, which means that $\underline{\delta}_X = 0 \leq \max\{0, \max_{e\in X} \underline{w}_e - F^*(S_E^+)\}$. This, together with (5), give (3). □

We thus can see that it is not difficult to compute the deviation interval for a given solution and to characterize its optimality, provided that the deterministic problem P is polynomially solvable. Similarly to the problems with linear sum cost function, the following proposition holds:

Proposition 3. *Every optimal min-max regret solution X is possibly optimal and it is composed of possibly optimal elements.*

Proof. If X is not possibly optimal, then by Theorem 4, we have $\max_{e\in X} \underline{w}_e > F^*(S_E^+)$. Let Y be an optimal solution under S_E^+. It is easy to verify that $F(X,S) > F(Y,S)$ under all scenarios $S \in \Gamma$. The same argument as in the proof of Proposition 1 yields $\overline{\delta}_X > \overline{\delta}_Y$, so X cannot be an optimal min-max regret solution. Of course, a possibly optimal solution is entirely composed of possibly optimal elements. □

Let us now focus on the elements. The following theorem allows us to compute a lower bound on the deviation interval of a given element. Its proof is very similar to the proof of Theorem 2.

Theorem 5. *For any element $f \in E$ it holds*

$$\underline{\delta}_f = \max\{0, \min_{X \in \Phi_f} F(X, S_E^-) - F^*(S_E^+)\}. \quad (6)$$

Using Theorem 5 we can design an efficient method of computing the quantity $\underline{\delta}_f$ for every particular problem, which is polynomially solvable. In order to compute $\min_{X\in\Phi_f} F(X,S_E^-)$ a slight modification of the algorithm for solving P is only required. Therefore, contrary to the problems with the linear sum cost, we can also characterize efficiently the possible optimality of a given element. However, a general characterization of the quantity $\overline{\delta}_f$ is unknown and it is an interesting subject of further research. Both bounds of Δ_f can be efficiently computed if P is matroidal problem. It is not difficult to prove the following result:

Proposition 4. *If* P *is a matroidal problem then* $\underline{\delta}_f = \max\{0, \underline{w}_f - F^*(S_E^+)\}$ *and* $\overline{\delta}_f = \max\{0, \overline{w}_f - F^*(S_{\{f\}}^+)\}$.

Equality (4) allows us to solve efficiently the min-max regret problem (2), provided that P is polynomially solvable. To see this let us define weights $\hat{w}_e = \max\{0, \overline{w}_e - F^*(S_{\{e\}}^+)\}$ for all $e \in E$. Then

$$\min_{X \in \Phi} \overline{\delta}_X = \min_{X \in \Phi} \max_{e \in X} \hat{w}_e$$

and the min-max regret problem reduces to solving the deterministic problem P with nonnegative real weights $\hat{w}_e$, $e \in E$. We thus get the following theorem:

Theorem 6 ([7]). *If the deterministic problem* P *can be solved in $f(n)$ time, then its min-max regret version can be solved in $O(nf(n))$ time.*

The running time $O(nf(n))$ follows from the fact that we need to solve n times the deterministic problem P to obtain weights $\hat{w}_e$ for all $e \in E$. The computations can be additionally refined and for details we refer the reader to [7]. The most important consequence of Theorem 6 is that the min-max regret version of problem P is polynomially solvable if only the deterministic problem P is polynomially solvable. So, the situation is quite different from the problems with the linear sum cost.

4 Combinatorial Optimization Problems with Fuzzy Weights

In the previous section we have described the class of problems with interval weights. It turns out that all the introduced concepts can be naturally extended without significant increasement of the problem complexity. The key idea is to use fuzzy intervals to model the uncertain element weights and apply possibility theory to extend the concept of deviation. This section is devoted to the class of combinatorial problems with fuzzy weighs.

4.1 Basic Notions of Possibility Theory

Possibility theory offers us a framework of dealing with imprecision. A detailed description of this theory can be found in a book [14]. We now recall some of its notions, which will be used later in this section. A *fuzzy interval* $\tilde{A}$ is a fuzzy set in the space of reals whose membership function $\mu_{\tilde{A}} : \mathbb{R} \to [0,1]$ is normal, quasi concave, upper semicontinuous and has a compact support. The main property of a fuzzy interval is that all its λ-*cuts*, that is the sets $\tilde{A}^{\lambda} = \{x : \mu_{\tilde{A}}(x) \geq \lambda\}$ for $\lambda \in (0,1]$, are closed intervals. We will also denote by $\tilde{A}^0$ the smallest closed set containing the support of $\tilde{A}$. So, we can represent a fuzzy interval $\tilde{A}$ as a family of closed intervals $\tilde{A}^{\lambda} = [\underline{a}^{\lambda}, \overline{a}^{\lambda}]$ parametrized by the value of $\lambda \in [0,1]$. It is easy to see that this family is monotone, that is $\tilde{A}^{\lambda_1} \subseteq \tilde{A}^{\lambda}_2$ if $\lambda_1 \geq \lambda_2$. Having the family of λ-cuts of $\tilde{A}$, the membership function $\mu_{\tilde{A}}$ can be computed as follows:

$$\mu_{\tilde{A}}(x) = \sup\{\lambda \in [0,1] : \ x \in \tilde{A}^{\lambda}\} \tag{7}$$

and $\mu_{\tilde{A}}(x) = 0$ if $x \notin \tilde{A}^0$.

In practice the class of *trapezoidal fuzzy intervals* is commonly used (see Figure 1). Every trapezoidal fuzzy interval can be described as a quadruple $(\underline{a}, \overline{a}, \alpha, \beta)$ and can be represented by the following family of λ-cuts:

$$\tilde{A}^{\lambda} = [\underline{a} - \alpha(1-\lambda), \overline{a} + \beta(1-\lambda)], \ \lambda \in [0,1]. \tag{8}$$

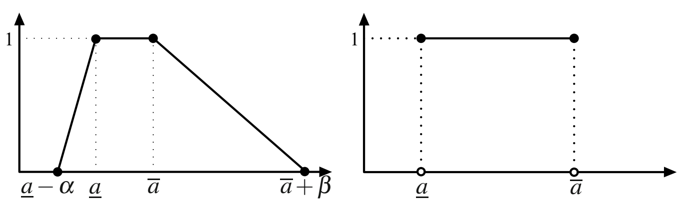

Fig. 1 Trapezoidal fuzzy interval $(\underline{a}, \overline{a}, \alpha, \beta)$ and closed interval $[\underline{a}, \overline{a}] = (\underline{a}, \overline{a}, 0, 0)$.

Notice that this representation also contains closed intervals (if $\alpha = \beta = 0$) and real numbers (if additionally $\underline{a} = \overline{a}$). We will use shorter notation (a, α, β) if $a = \underline{a} = \overline{a}$ and we will call (a, α, β) a *triangular fuzzy interval*. We also define $(\overline{a}, \beta) = (0, \overline{a}, 0, \beta)$. In order to simplify notations and discussion we will only use trapezoidal fuzzy intervals. However, it is not difficult to extend all the introduced notions to a more general class of fuzzy intervals of the L-R type with compact support (see [14] for a descriptions of this class of fuzzy intervals).

W now give an interpretation of a fuzzy interval. Let a be a real quantity whose value is not precisely known. We associate with a a fuzzy interval $\tilde{A}$, whose membership function $\mu_{\tilde{A}}$ is a *possibility distribution* for the values of a, that is

$$\Pi(a = x) = \mu_{\tilde{A}}(x),$$

where $\Pi(a = x)$ is the possibility of the event that a will take the value of x. There are several methods of obtaining possibility distribution for an unknown quantity and their description can be found in [14]. Observe that $\tilde{A}^{\lambda}$ contains all values of a whose possibility of occurrence is not less than λ. In consequence, $\tilde{A}^0$ should contain all possible values of a and $\tilde{A}^1$ should contain the most plausible ones.

Let $\tilde{G}$ be a fuzzy set in the space of reals. Then $a \in \tilde{G}$ is a *fuzzy event* and the possibility and necessity of $a \in \tilde{G}$ are defined as follows [13]:

$$\Pi(a \in \tilde{G}) = \sup_{x \in \mathbb{R}} \min\{\mu_{\tilde{A}}(x), \mu_{\tilde{G}}(x)\}. \tag{9}$$

$$\mathrm{N}(a \in \tilde{G}) = 1 - \Pi(a \notin \tilde{G}) = 1 - \sup_{x \in \mathbb{R}} \min\{\mu_{\tilde{A}}(x), 1 - \mu_{\tilde{G}}(x)\}. \tag{10}$$

where $1 - \mu_{\tilde{G}}(x)$ is the membership function of the complement of the fuzzy set $\tilde{G}$. It is not difficult to show that if $\tilde{G} = (0, \overline{g}, 0, \beta) = (\overline{g}, \beta)$, then the following equality is true:

$$\mathrm{N}(a \in \tilde{G}) = 1 - \inf\{\lambda \in [0,1] : \overline{a}^{\lambda} \leq \overline{g}^{1-\lambda}\} \tag{11}$$

and $\mathrm{N}(a \in \tilde{G}) = 0$ if $\overline{a}^1 > \overline{g}^0$. Equality (11) is illustrated in Figure 2.

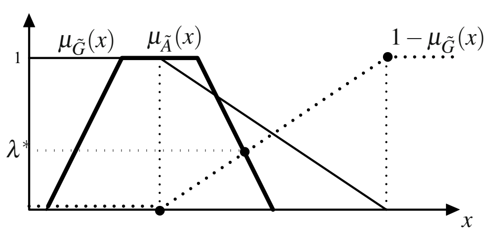

Fig. 2 $\mathrm{N}(a \in \tilde{G}) = 1 - \lambda^*$

In the next section we will show how possibility theory allows us to extend the concept of scenario set.

4.2 Fuzzy Scenario Set

Assume that the element weights in a combinatorial optimization problem are unknown real quantities w_e, $e \in E$. We associate with every weight w_e a fuzzy interval $\tilde{W}_e$. According to the interpretation given in the previous section, the membership function $\mu_{\tilde{W}_e}$ is a possibility distribution for the values of the weight w_e. Let $S = (s_e)_{e\in E} \in \mathbb{R}^n$ be a vector representing the state of the world in which $w_e = s_e$ for all $e \in E$. As in Section 3, we will call S a scenario. The possibility distributions associated with element weights induce the following possibility distribution over all scenarios $S \in \mathbb{R}^n$:

$$\pi(S) = \Pi\left(\bigwedge_{e\in E}[w_e = s_e]\right) = \min_{e\in E} \Pi(w_e = s_e) = \min_{e\in E} \mu_{\tilde{W}_e}(s_e). \tag{12}$$

Observe that $\pi(S)$ may be regarded as a membership function of a fuzzy set in $\mathbb{R}^n$. We will call this fuzzy set a *fuzzy scenario set* and $\pi(S)$ is the possibility of the event that scenario $S \in \mathbb{R}^n$ will occur. Notice that we generalize in this way scenario set Γ defined in Section 3. Indeed, under interval uncertainty representation $\pi(S) = 1$ if $S \in \Gamma$ and $\pi(S) = 0$ otherwise, so $\pi(S)$ is then a characteristic function of the set Γ. Under fuzzy weights, $\pi(S)$ may take any value in the interval $[0,1]$. Hence fuzzy weights provide us more information about the state of the world which may occur. In particular, scenario S is impossible if $\pi(S) = 0$ and we have $\pi(S) = 1$ for the most plausible scenarios. Notice that the definition of a fuzzy interval assures that $\pi(S) = 1$ for at least one scenario S.

Using (12) and the definition of λ-cut it is easily seen that for every $\lambda \in [0,1]$ the following equality holds:

$$\{S:\ \pi(S) \geq \lambda\} = \times_{e\in E}[\underline{w}_e^\lambda, \overline{w}_e^\lambda]. \tag{13}$$

So, the set of all scenarios whose possibility of occurrence is not less than λ is the Cartesian product of the interval weights being the λ-cuts of the fuzzy weights. Hence it forms a scenario set, which we will denote as Γ^λ. This property allows us to decompose the fuzzy problem into a family of interval problems. We will make use of this fact in the next sections.

4.3 Fuzzy Deviations

As for the problems with deterministic and interval weights, we can use the concept of deviation to characterize the optimality of solutions and elements. Recall that under interval weights deviations δ_X and δ_f fall within closed intervals Δ_X and Δ_f. Under fuzzy weights, the solution and element deviations are unknown quantities, which fall within fuzzy intervals $\tilde{\Delta}_X$ and $\tilde{\Delta}_f$. The membership functions $\mu_{\tilde{\Delta}_X}$ and $\mu_{\tilde{\Delta}_f}$

are possibility distributions for the values of δ_X and δ_f and, according to possibility theory, they are defined as follows:

$$\mu_{\tilde{\Delta}_X}(y) = \Pi(\delta_X = y) = \sup_{\{S:\, \delta_X(S)=y\}} \pi(S),$$
$$\mu_{\tilde{\Delta}_f}(y) = \Pi(\delta_f = y) = \sup_{\{S:\, \delta_f(S)=y\}} \pi(S).$$

Consider fuzzy deviation interval $\tilde{\Delta}_X$ of a given solution $X \in \Phi$. A λ-cut of $\tilde{\Delta}_X$ contains all values of the deviation of X whose possibility of occurrence is not less than λ. Hence

$$\tilde{\Delta}_X^{\lambda} = \{y : \mu_{\tilde{\Delta}_X}(y) \geq \lambda\} = \{\delta_X(S) :\ \pi(S) \geq \lambda, S \in \mathbb{R}^n\}.$$

But (13) implies $\tilde{\Delta}_X^{\lambda} = \{\delta_X(S) :\ S \in \Gamma^{\lambda}\} = [\underline{\delta}_X^{\lambda}, \overline{\delta}_X^{\lambda}]$, where $\underline{\delta}_X^{\lambda}$ minimizes and $\overline{\delta}_X^{\lambda}$ maximizes $\delta_X(S)$ over all $S \in \Gamma^{\lambda}$. We can now use the results from Sections 3.4.1 and 3.4.2 to compute the bounds of $\tilde{\Delta}_X^{\lambda}$. For a problem with the linear sum cost, Proposition 1 gives

$$\underline{\delta}_X^{\lambda} = \delta_X(S_X^{-\lambda}) = F(X, S_X^{-\lambda}) - F^*(S_X^{-\lambda}) = \sum_{e \in X} \underline{w}_e^{\lambda} - F^*(S_X^{-\lambda}), \tag{14}$$

$$\overline{\delta}_X^{\lambda} = \delta_X(S_X^{+\lambda}) = F(X, S_X^{+\lambda}) - F^*(S_X^{+\lambda}) = \sum_{e \in X} \overline{w}_e^{\lambda} - F^*(S_X^{+\lambda}), \tag{15}$$

where $S_X^{-\lambda}$ and $S_X^{+\lambda}$ are the corresponding extreme scenarios in Γ^{λ}.

Now our aim is to compute the family of cuts $\tilde{\Delta}_X^{\lambda}$ for $\lambda \in [0,1]$. The possibility distribution for the deviation of X can be then obtain by formula (7). Observe that it remains to compute functions $F^*(S_X^{-\lambda})$ and $F^*(S_X^{+\lambda})$ of $\lambda \in [0,1]$. This task can be performed by applying a parametric technique. Namely, we wish to compute sequences $0 = \lambda_0 < \lambda_1 < \ldots < \lambda_k = 1$ and $X_0, \ldots, X_{k-1}$ such that X_i is an optimal solution under $S_X^{+\lambda}$ or $S_X^{-\lambda}$ for $\lambda \in [\lambda_i, \lambda_{i+1}]$. Having these sequences it is easy to describe analytically functions $F^*(S_X^{-\lambda})$ or $F^*(S_X^{+\lambda})$ for $\lambda \in [0,1]$. It turns out that if $\underline{w}_e^{\lambda}$ and $\overline{w}_e^{\lambda}$ are linear functions of λ for each $e \in E$, then for some particular problems such as shortest path or minimum spanning tree their parametric counterparts can be efficiently solved (see e.g. [15, 38, 41]). In consequence, the family of intervals $\tilde{\Delta}_X^{\lambda}$, $\lambda \in [0,1]$, can be efficiently computed if the uncertain weights are modeled as trapezoidal fuzzy intervals. A similar reasoning applies to the problems with the bottleneck cost function. One should only use Theorem 4 to obtain the corresponding parametric problems. We now illustrate the computation of $\tilde{\Delta}_X$ by an example.

Example 1. Consider a shortest path problem shown in Figure 3a. We are given a directed graph composed of 5 arcs and we wish to find a shortest path between nodes s and t.

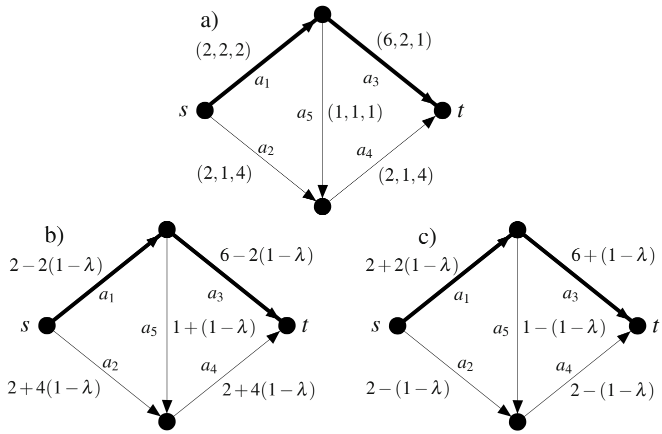

Fig. 3 a) A sample shortest path problem with fuzzy weights. b) The extreme scenario $S^{-\lambda}_{\{a_1,a_3\}}$. c) The extreme scenario $S^{+\lambda}_{\{a_1,a_3\}}$.

For every arc weight w_a, $a \in A$, a triangular fuzzy interval $\tilde{W}_a = (w_a, \alpha_a, \beta_a)$ is given. Let us examine path $X = \{a_1, a_3\}$. In Figures 3b and 3c the extreme scenarios $S^{-\lambda}_X$ and $S^{+\lambda}_X$ are shown. Notice that these scenarios are linear functions of $\lambda \in [0,1]$ obtained by formula (8). It holds $F(X, S^{-\lambda}_X) = 8 - 4(1-\lambda)$ and $F(X, S^{+\lambda}_X) = 8 + 3(1-\lambda)$ for $\lambda \in [0,1]$. Applying a parametric technique to the problem shown in Figure 3b, we get a sequence of λ's $0 < \frac{4}{7} < \frac{4}{5} < 1$ that corresponds to the sequence of optimal solutions $\{a_1,a_3\}, \{a_1,a_5,a_4\}, \{a_2,a_4\}$. That is, solution $\{a_1,a_3\}$ is optimal for $\lambda \in [0, \frac{4}{7}]$, solution $\{a_1,a_5,a_4\}$ is optimal for $\lambda \in [\frac{4}{7}, \frac{4}{5}]$ and solution $\{a_2,a_4\}$ is optimal for $\lambda \in [\frac{4}{5}, 1]$. Hence $F^*(S^{-\lambda}_X)$ is a piecewise linear function, whose value is $8-4(1-\lambda)$ for $\lambda \in [0, \frac{4}{7}]$, $5+3(1-\lambda)$ for $\lambda \in [\frac{4}{7}, \frac{4}{5}]$ and $4+8(1-\lambda)$ for $\lambda \in [\frac{4}{5}, 1]$. Subtracting $F^*(S^{-\lambda}_X)$ from $F(X, S^{-\lambda}_X)$ yields $\underline{\delta}^{\lambda}_X$. Similarly, the function $\overline{\delta}^{\lambda}_X$ is obtained by applying the parametric technique to the problem shown in Figure 3c. The resulting functions $\underline{\delta}^{\lambda}_X$ and $\overline{\delta}^{\lambda}_X$ are presented in Figure 4. Having the bounds $\underline{\delta}^{\lambda}_X$ and $\overline{\delta}^{\lambda}_X$ for $\lambda \in [0,1]$ we can construct the possibility distribution $\mu_{\tilde{\Delta}_X}$ for the deviations of X by applying formula (7). This possibility distribution is shown in Figure 4. □

Computing fuzzy deviation of a given element is more complex. It is a direct consequence of the fact that the corresponding interval problem may be hard to solve. In other words, it may be hard to identify extreme scenarios that minimize or maximize an element deviation. We can compute the fuzzy interval $\tilde{\Delta}_f$ only for some particular problems such as matroidal ones (see e.g. [27]).

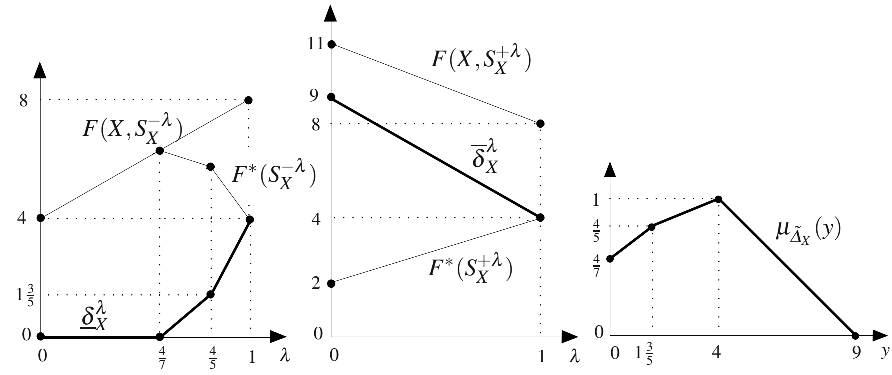

Fig. 4 Bounds $\underline{\delta}_X^{\lambda}$ and $\overline{\delta}_X^{\lambda}$ and the possibility distribution $\mu_{\tilde{\Delta}_X}(y) = \Pi(\delta_X = y)$

4.4 Degrees of Possible and Necessary Optimality

The fuzzy deviations allow us to characterize possible and necessary optimality of solutions and elements. Recall that the statement "X is optimal" is equivalent to the assertion $\delta_X = 0$. So, we can define the *degrees of possible and necessary optimality* of solution X in the following way:

$$\Pi(X \text{ is optimal}) = \Pi(\delta_X = 0) = \mu_{\tilde{\Delta}_X}(0), \tag{16}$$

$$\mathrm{N}(X \text{ is optimal}) = 1 - \Pi(\delta_X > 0) = 1 - \sup_{y>0} \mu_{\tilde{\Delta}_X}(y). \tag{17}$$

In the same way we can define the degrees of optimality of the elements. It is enough to replace X with f in (16) and (17). The following relations hold between the optimality degrees of solutions and elements:

$$\Pi(X \text{ is optimal}) \leq \max_{e \in X} \Pi(e \text{ is optimal}).$$

$$\mathrm{N}(X \text{ is optimal}) \leq \max_{e \in X} \mathrm{N}(e \text{ is optimal}).$$

Having possibility distributions $\mu_{\tilde{\Delta}_X}$ and $\mu_{\tilde{\Delta}_f}$ we can immediately compute the degrees of optimality of X and f. However, if one wishes to obtain only the optimality degrees, then the computations can be significantly simplified. Equalities (7) and (16) imply

$$\Pi(X \text{ is optimal}) = \sup\{\lambda \in [0,1] : 0 \in \tilde{\Delta}_X^{\lambda}\} = \sup\{\lambda \in [0,1] : \underline{\delta}_X^{\lambda} = 0\} \tag{18}$$

and $\Pi(X \text{ is optimal}) = 0$ if $\underline{\delta}_X^0 > 0$. So, in order to compute the degree of possible optimality of X, we need to find the largest value of λ such that X is possibly optimal under scenario set Γ^λ (which is equivalent to the condition $\underline{\delta}_X^\lambda = 0$). Since $\underline{\delta}_X^\lambda$ is nondecreasing function of λ, the standard binary search technique can be applied to perform this task. Also, the following equality is easy to establish:

$$\mathrm{N}(X \text{ is optimal}) = 1 - \inf\{\lambda \in [0,1] : \overline{\delta}_X^\lambda = 0\} \tag{19}$$

and $\mathrm{N}(X \text{ is optimal}) = 0$ if $\overline{\delta}_X^1 > 0$. So, we need to find the smallest value of λ such that X is necessarily optimal under scenario set Γ^λ (which is equivalent to the condition $\overline{\delta}_X^\lambda = 0$). Because $\overline{\delta}_X^\lambda$ is nonincreasing function of λ, the binary search technique also solves this problem. If $f(n)$ is the time required to assert whether a given solution is possibly (necessarily) optimal in the interval problem, then its degree of possible (necessary) optimality can be computed in $O(f(n)\log \varepsilon^{-1})$ time, where $\varepsilon \in (0,1)$ is an assumed precision of calculations.

Exactly the same reasoning can be applied to the elements (we only need to replace X with f in (18) and (19)). Note, however, that the complexity of computations for an element strongly depends on the combinatorial structure of problem P.

4.5 Choosing a Solution under Fuzzy Weights

We now address the problem of choosing a solution under fuzzy weights. The degrees of optimality, introduced in the previous section, suggest us a solution method. We can choose a solution, which maximizes the degree of possible or necessary optimality. Maximizing the degree of possible optimality is trivial. There is always at least one solution $X \in \Phi$ for which the degree of possible optimality attains its maximal value equal to 1. It can be obtained by computing an optimal solution under scenario S such that $\pi(S) = 1$. On the other hand, the degree of necessary optimality of every feasible solution may be very small or even equal to 0. We thus meet the same problem as in the interval uncertainty representation - the possible optimality is too weak criterion of choosing a solution and the necessary optimality is too strong.

Now the idea is to replace the strong optimality requirement with a weaker one. Suppose that a decision maker knows his/her preference about solution deviation and expresses it using a *fuzzy goal* $\tilde{G} = (\overline{g}, \beta_g)$. So, the values of deviation in $[0,\overline{g}]$ are completely accepted, the values in $[\overline{g}+\beta_g,\infty)$ are not at all accepted and the degree of acceptance decreases linearly in the interval $[\overline{g},\overline{g}+\beta_g]$ (the assumption that it decreases linearly is not restrictive and any decreasing function can be used to model the decision maker preferences). We can now replace the strong requirement $\delta_X = 0$ with a weaker one, namely $\delta_X \in \tilde{G}$. Recall that δ_X is an unknown quantity characterized by possibility distribution $\mu_{\tilde{\Delta}_X}$. So, $\delta_X \in \tilde{G}$ is a fuzzy event and we can compute the necessity that it holds, $\mathrm{N}(\delta_X \in \tilde{G})$, using (10):

$$\mathrm{N}(\delta_X \in \tilde{G}) = 1 - \sup_{y \in \mathbb{R}} \min\{\mu_{\tilde{\Delta}_X}(y), 1 - \mu_{\tilde{G}}(y)\}.$$

Consider again the shortest path problem from Example 1. The function $\mu_{\tilde{\Delta}_X}$ is a possibility distribution for deviation δ_X of path X. This possibility distribution is shown in Figure 5. The part of $\mu_{\tilde{\Delta}_X}$ representing the largest deviation of X is shown in bold. In Figure 5 a fuzzy goal $\tilde{G}$ and its complement $\tilde{G}^d$ are also shown. The complement $\tilde{G}^d$ expresses a degree of dissatisfaction of the values of solution deviation.

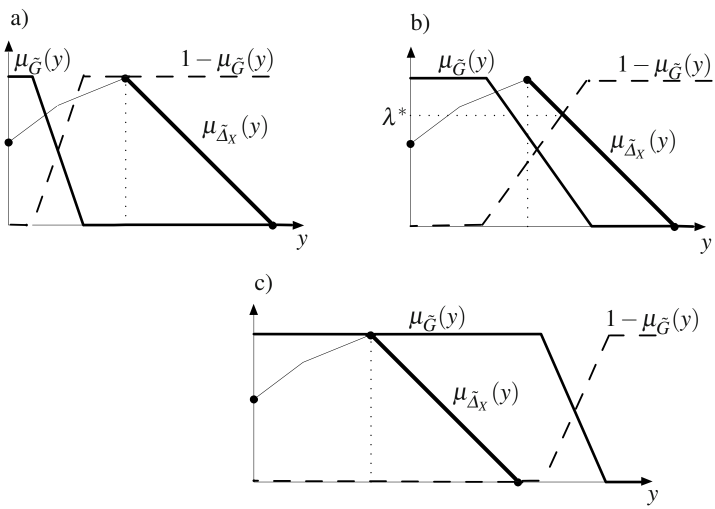

Fig. 5 Three different situations depending on the choice of fuzzy goal $\tilde{G}$: a) $\mathrm{N}(\delta_X \in \tilde{G}) = 0$, b) $\mathrm{N}(\delta_X \in \tilde{G}) = 1-\lambda^*$, c) $\mathrm{N}(\delta_X \in \tilde{G}) = 1$

Consider the case illustrated in Figure 5a. The goal $\tilde{G}$ is chosen so that the largest deviation of X is fully contained in its complement $\tilde{G}^d$. So, $\Pi(\delta_X \in \tilde{G}^d) = 1$ and $\mathrm{N}(\delta_X \in \tilde{G}^d) = 0$. In other words, with possibility equal to 1 a scenario may occur for which the deviation of X is not at all accepted. Figure 5c shows an opposite case. The goal $\tilde{G}$ is chosen so that the largest deviation of X is completely not in $\tilde{G}^d$. So, $\Pi(\delta_X \in \tilde{G}^d) = 0$ and $\mathrm{N}(\delta_X \in \tilde{G}^d) = 1$. In this case for every scenario S such that $\pi(S) > 0$ the deviation $\delta_X(S)$ is completely accepted. Clearly, this is an ideal situation. In Figure 5b a third case is shown, where the largest deviation of X is only partially contained in $\tilde{G}^d$. So, $\Pi(\delta_X \in \tilde{G}^d) = \lambda^*$ and $\mathrm{N}(\delta_X \in \tilde{G}^d) = 1-\lambda^*$. This means that for all scenarios S such that $\pi(S) \geq \lambda^*$ the degree of dissatisfaction is not greater than λ^* or, equivalently, the degree of satisfaction is not less than $1-\lambda^*$.

Now it is reasonable to choose a solution whose deviation belongs to $\tilde{G}$ with the highest confidence. This leads to the following optimization problem:

$$\max_{X \in \Phi} \mathrm{N}(\delta_X \in \tilde{G}). \tag{20}$$

An optimal solution to (20) is called a *most necessarily soft optimal solution* and it was first proposed as a solution under fuzzy weights in [20]. If we choose $\tilde{G} = (0,0)$, then we get the following special case of (20):

$$\max_{X \in \Phi} \mathrm{N}(\delta_X = 0) = \max_{X \in \Phi} \mathrm{N}(X \text{ is optimal}). \tag{21}$$

So, in (21) we seek a most necessarily optimal solution. As we will see in the next section, the problem (21) may be easier to solve than (20). Using (11) we can express the problem (20) as the following mathematical programming one:

$$\begin{array}{l} \min \lambda \\ \overline{\delta}_X^{\lambda} \leq \overline{g}^{1-\lambda} \\ X \in \Phi \\ \lambda \in [0,1] \end{array} \tag{22}$$

If λ^* is the optimal objective value of (22) and X^* is an optimal solution, then $\mathrm{N}(\delta_{X^*} \in \tilde{G}) = 1 - \lambda^*$. If (22) is infeasible, then $\mathrm{N}(\delta_X \in \tilde{G}) = 0$ for all feasible solutions X.

It is easy to check that problem (22) is a generalization of the min-max regret approach. If all $\tilde{W}_e$, $e \in E$, are closed intervals and $\tilde{G} = (0,M)$ for a sufficiently large number M, then (22) is equivalent to (2). In the next two sections we will focus on some methods of solving (22).

4.5.1 Binary Search Technique

Observe that $\overline{\delta}_X^{\lambda}$ is nonincreasing and $\overline{g}^{1-\lambda}$ is nondecreasing function of $\lambda \in [0,1]$. Therefore (22) can be solved by applying the standard binary search technique shown in Figure 6. The algorithm simply seeks a minimal value of λ in the interval [0,1], for which there is a solution $X \in \Phi$ that satisfies inequality $\overline{\delta}_X^{\lambda} \leq \overline{g}^{1-\lambda}$. The quantity $\overline{\delta}_X^{\lambda}$ is the maximal regret of solution X under scenario set Γ^{λ}. Therefore, the inequality $\overline{\delta}_X^{\lambda} \leq \overline{g}^{1-\lambda}$ is satisfied for some $X \in \Phi$ if and only if it is satisfied by an optimal min-max regret solution under Γ^{λ}. So, if we are able to solve the min-max regret problem with interval data in $f(n)$ time, then the binary search solves the fuzzy problem in $O(f(n)\log \varepsilon^{-1})$ time with a given precision $\varepsilon \in (0,1)$.

We can see now that if the min-max regret problem is polynomially solvable, then its fuzzy generalization is polynomially solvable up to a given precision ε. Notice that for the class of problems with the bottleneck cost function, it is enough that the deterministic problem is polynomially solvable (see Theorem 4). For the problems with the linear sum cost the situation is more complex since the min-max regret problem is mostly NP-hard. However, if the deterministic problem is polynomially solvable, then we can solve efficiently the special case (21), that is we can find efficiently a most necessarily optimal solution with a given precision ε. If $\tilde{G} = (0,0)$, then $\overline{g}^{\lambda} = 0$ for all $\lambda \in [0,1]$. The condition $\overline{\delta}_X^{\lambda} \leq 0$ can be efficiently

Find an optimal min-max regret solution X under Γ^1
if $\overline{\delta}_X^1 > \overline{g}^0$ **then return** $\emptyset$
$\lambda_1 \leftarrow 0.5$, $k \leftarrow 1$, $\lambda_2 \leftarrow 0$
while $|\lambda_1 - \lambda_2| < \varepsilon$ **do**
    $\lambda_2 \leftarrow \lambda_1$
    Find an optimal min-max regret solution Y under Γ^{λ_1}
    if $\overline{\delta}_Y^{\lambda_1} \leq \overline{g}^{1-\lambda_1}$ **then** $\lambda_1 \leftarrow \lambda_1 - 1/2^{k+1}$, $X \leftarrow Y$ **else** $\lambda_1 \leftarrow \lambda_1 + 1/2^{k+1}$
    $k \leftarrow k+1$
end while
return X

Fig. 6 Computing a most necessarily soft optimal solution with a given precision $\varepsilon \in (0,1)$. Algorithm returns $\emptyset$ if $\mathrm{N}(\delta_X \in \tilde{G}) = 0$ for all $X \in \Phi$.

verified for a fixed λ by using Theorem 1 because $\overline{\delta}_X^{\lambda} \leq 0$ if and only if there is a necessarily optimal solution under scenario set Γ^{λ}.

The binary search is the most general method of solving the fuzzy problem. However, it gives only an approximate solution. Furthermore, it may be not efficient for the problems with the linear sum cost function because solving $O(\log \varepsilon^{-1})$ times an NP-hard problem may be time consuming. In the next sections we show some alternative methods of finding a most necessarily soft optimal solution.

4.5.2 Parametric Technique of the Problems with Bottleneck Cost

Consider the class of problems with the bottleneck cost function. Using (4) and (11) we can express the fuzzy problem in the following way:

$$\inf\left\{\lambda \in [0,1] : \min_{X\in\Phi} \max_{e\in X} \hat{w}_e^{\lambda} \leq \overline{g}^{1-\lambda}\right\}. \tag{23}$$

where $\hat{w}_e^{\lambda} = \max\{0, \overline{w}_e^{\lambda} - F^*(S_{\{e\}}^{+\lambda})\}$. We can obtain weights $\hat{w}_e^{\lambda}$ for all $e \in E$ using a parametric technique (see e.g. [11]). As the result we obtain another parametric bottleneck problem with weights $\hat{w}_e^{\lambda}$, $e \in E$, that is

$$\overline{\delta}^{\lambda} = \min_{X\in\Phi} \max_{e\in X} \hat{w}_e^{\lambda}. \tag{24}$$

Solving (24) we obtain sequences $0 \leq \lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_k = 1$ and $X_0, \ldots, X_{k-1}$ such that X_i is an optimal solution for $\lambda \in [\lambda_i, \lambda_{i+1}]$. Having these sequences it is easy to describe analytically function $\overline{\delta}^{\lambda}$ for $\lambda \in [0,1]$. The function $\overline{\delta}^{\lambda}$ is nonincreasing, hence from (23) we can see that in order to obtain a most necessarily soft optimal solution we must find the intersection point λ^* of $\overline{\delta}^{\lambda}$ with $\overline{g}^{1-\lambda}$. Then, if $\lambda^* \in [\lambda_i, \lambda_{i+1}]$, then X_i is a necessarily soft optimal solution. If such an intersection point does not exist, then two cases are possible - either $\overline{\delta}^1 > \overline{g}^0$ or $\overline{\delta}^0 < \overline{g}^1$. In the former case $\mathrm{N}(\delta_X \in \tilde{G}) = 0$ for all feasible solutions X and in the latter one $\mathrm{N}(\delta_{X_0} \in \tilde{G}) = 1$ and X_0 is a necessarily soft optimal solution.

The solution procedure based on a parametric technique is more time consuming than the binary search shown in the previous section. It has, however, two important advantages. First of all, it gives an exact necessarily soft optimal solution. Furthermore, it provides a lot of additional information in the fuzzy problem. Observe that, regardless of fuzzy goal, a most necessarily soft optimal solution is always among $X_0, \ldots, X_{k-1}$. We can thus treat the set of solutions $\{X_0, \ldots, X_{k-1}\}$ as a solution of the fuzzy problem. Introducing fuzzy goal $\tilde{G}$ allows us to chose one of these solutions. One can also check easily how the solution changes when the fuzzy goal $\tilde{G}$ is changed. So, we can perform a sensitivity analysis of the obtained solution.

4.5.3 MIP Formulation for the Problems with Linear Sum Cost

In this section we show an exact method of solving (20) for the problems with the linear sum cost. Under some additional assumptions we design a mixed integer linear programming (MIP) model, which can be then solved by some available software. Let us assign a binary variable $x_i \in \{0,1\}$ to every element $e_i \in E$. This variable will indicate whether element e_i is contained in a constructed solution. Every feasible solution $X \in \boldsymbol{\Phi}$ can be represented as a vector of binary variables $\boldsymbol{x} = [x_1, \ldots, x_n]$, where $x_i = 1$ if and only if $e_i \in X$. We assume that the set of feasible solutions can be described by a system of linear constraints of the form $\{\boldsymbol{x} \in \{0,1\}^n : \mathscr{A}\boldsymbol{x}^T = \boldsymbol{b}\}$, where $\mathscr{A}$ is a matrix and $\boldsymbol{b}$ is a vector of fixed coefficients. We allow also inequalities $\leq$ and $\geq$ in the constraints since they can easily be converted to equalities by adding a number of additional slack variables. In order to simplify notations we will use $\tilde{W}_i$ to denote the fuzzy interval associated with the weight of element e_i.

We will assume that the matrix $\mathscr{A}$ is *totally unimodular*. Recall that in a totally unimodular matrix the determinants of all its nonsingular square submatrices are equal to -1 or 1 (see e.g. [18]). This assumption restricts the class of considered problems. However, if the deterministic problem P is polynomially solvable, then it can often be formulated as a 0-1 linear programming problem with a totally unimodular constraints matrix. This is, for instance, the case for a wide class of network flow problems such as shortest path, minimum spanning tree, minimum assignment or minimum cut [1, 18].

Recall (see (15)) that $\overline{\delta}_X^{\lambda} = \delta_X(S_X^{+\lambda}) = F(X, S_X^{+\lambda}) - F^*(S_X^{+\lambda})$. Using the vector of binary variables $\boldsymbol{x}$ representing X, we can see that $F(X, S_X^{+\lambda}) = \sum_{i=1}^n \overline{w}_i^{\lambda} x_i$. Under scenario $S_X^{+\lambda}$ the weight of element e_i is $\overline{w}_i^{\lambda} x_i + \underline{w}_i^{\lambda}(1 - x_i)$. So, $F^*(S_X^{+\lambda})$ can be expressed as follows:

$$\begin{aligned} &\min \sum_{i=1}^{n} [\overline{w}_i^{\lambda} x_i + \underline{w}_i^{\lambda}(1 - x_i)] y_i \\ &\mathscr{A}\boldsymbol{y}^T = \boldsymbol{b} \\ &y_i \in \{0,1\} \qquad\qquad i = 1, \ldots, n \end{aligned} \tag{25}$$

We now use the assumption that matrix $\mathscr{A}$ is totally unimodular. Under this assumption (see e.g. [18]) we can replace constraints $y_i \in \{0,1\}$ in (25) with $0 \le y_i \le 1$ without changing the cost of an optimal solution to (25). As the result we get the following problem:

$$\begin{aligned} &\min \sum_{i=1}^{n} [\overline{w}_i^{\lambda} x_i + \underline{w}_i^{\lambda}(1-x_i)] y_i \\ &\mathscr{A}\boldsymbol{y}^T = \boldsymbol{b} \\ &0 \le y_i \le 1 \qquad\qquad i=1,\ldots,n \end{aligned} \tag{26}$$

We can now construct a dual model to (26). This dual model has a vector of dual variables $\boldsymbol{u}$ associated with the constraints of (26). Denote by $\phi(\boldsymbol{u})$ the objective of the dual and by $D^{\lambda}(\boldsymbol{x})$ the set of feasible dual vectors. So, the dual model is $\max_{\boldsymbol{u}\in D^{\lambda}(\boldsymbol{x})} \phi(\boldsymbol{u})$ and it is linear with respect to both $\boldsymbol{u}$ and $\boldsymbol{x}$ if λ is fixed. Now the strong duality theorem implies:

$$F^*(S_X^{+\lambda}) = \max_{\boldsymbol{u}\in D^{\lambda}(\boldsymbol{x})} \phi(\boldsymbol{u}).$$

Hence

$$\overline{\delta}_X^{\lambda} = \sum_{i=1}^{n} \overline{w}_i^{\lambda} x_i - \max_{\boldsymbol{u}\in D^{\lambda}(\boldsymbol{x})} \phi(\boldsymbol{u}),$$

which together with (22) give

$$\begin{aligned} &\min \lambda \\ &\sum_{i=1}^{n} \overline{w}_i^{\lambda} x_i - \max_{\boldsymbol{u}\in D^{\lambda}(\boldsymbol{x})} \phi(\boldsymbol{u}) \le \overline{g}^{1-\lambda} \\ &\mathscr{A}\boldsymbol{x}^T = \boldsymbol{b} \\ &x_i \in \{0,1\} \qquad\qquad i=1,\ldots,n \\ &\lambda \in [0,1] \end{aligned} \tag{27}$$

We can omit the maximum operator in (27) obtaining the following equivalent model:

$$\begin{aligned} &\min \lambda \\ &\sum_{i=1}^{n} \overline{w}_i^{\lambda} x_i - \phi(\boldsymbol{u}) \le \overline{g}^{1-\lambda} \\ &\mathscr{A}\boldsymbol{x}^T = \boldsymbol{b} \\ &\boldsymbol{u} \in D^{\lambda}(\boldsymbol{x}) \\ &x_i \in \{0,1\} \qquad\qquad i=1,\ldots,n \\ &\lambda \in [0,1] \end{aligned} \tag{28}$$

Assuming that the element weights are trapezoidal fuzzy intervals $\tilde{W}_i = (\underline{w}_i, \overline{w}_i, \alpha_i, \beta_i)$ for all $e_i \in E$, we can substitute $\overline{w}_i^{\lambda} = \overline{w}_i + \beta_i(1-\lambda)$ and $\underline{w}_i^{\lambda} = \underline{w}_i - \alpha_i(1-\lambda)$ in (28). The resulting model will be still not linear because some expressions of the form λx_i may appear. However, we can make (28) linear by replacing all such expressions with additional variables and adding some additional linear constraints. After this modification, problem (28) will be a mixed integer linear programming one. We will illustrate this method by an example.

Example 2. Consider the following *minimum selecting items* problem. Let $E = \{e_1,\dots,e_n\}$ be a set of items. The solution set Φ consists of all subsets X of E such that $|X| = p$, where p is a given integer. So, we wish to choose exactly p items among E. Assume that fuzzy interval $\tilde{W}_i = (\underline{w}_i, \overline{w}_i, \alpha_i, \beta_i)$ is given for every $e_i \in E$. We also fix a fuzzy goal $\tilde{G} = (\overline{g}, \beta_g)$. The binary variable $x_i \in \{0,1\}$ indicates whether item e_i is chosen or not. The solution set Φ in this problem can be described by the single constraint $x_1 + x_2 + \cdots + x_n = p$. Obviously, matrix $\mathscr{A} = [1,1,\dots,1]$ is totally unimodular. The subproblem (26) takes the following form:

$$\begin{array}{ll} \min \sum_{i=1}^{n} [\overline{w}_i^{\lambda} x_i + \underline{w}_i^{\lambda}(1-x_i)] y_i & \\ y_1 + y_2 + \cdots + y_n = p & \\ 0 \le y_i \le 1 & i = 1,\dots,n \end{array}$$

Assigning dual variable u_0 to the equality constraint and dual variables $u_1,\dots,u_n$ to constraints $y_i \le 1$, $i = 1,\dots,n$, we get the following dual model:

$$\begin{array}{ll} \max pu_0 - u_1 - \cdots - u_n & \\ u_0 - u_i \le \overline{w}_i^{\lambda} x_i + \underline{w}_i^{\lambda}(1-x_i) & \\ u_i \ge 0 & i = 1,\dots,n \end{array}$$

Consequently, $\phi(\boldsymbol{u}) = pu_0 - u_1 - \cdots - u_n$ and set $D^{\lambda}(\boldsymbol{x})$ is described by the constraints of the dual model. We are now ready to design the model using formulation (28). We also substitute $\overline{w}_i^{\lambda} = \overline{w}_i + \beta_i(1-\lambda)$ and $\underline{w}_i^{\lambda} = \underline{w}_i - \alpha_i(1-\lambda)$. After easy computations we get

$$\begin{array}{ll} \min \lambda & \\ \sum_{i=1}^{n} (\overline{w}_i + \beta_i) x_i - \sum_{i=1}^{n} \beta_i \lambda x_i - pu_0 + \sum_{i=1}^{n} u_i \le \overline{g} + \beta_g \lambda & \\ \sum_{i=1}^{n} x_i = p & \\ u_0 - u_i \le (\overline{w}_i - \underline{w}_i + \alpha_i + \beta_i) x_i - (\alpha_i + \beta_i)\lambda x_i - \alpha_i(1-\lambda) + \underline{w}_i & i = 1,\dots,n \\ u_i \ge 0 & i = 1,\dots,n \\ \lambda \in [0,1] & \\ x_i \in \{0,1\} & i = 1,\dots,n \end{array}$$

The obtained model is still not linear. We can, however, substitute $t_i = \lambda x_i$ and add additional linear constraints $t_i - x_i \le 0$, $\lambda - t_i + x_i \le 1$, $-\lambda + t_i \le 0$, $t_i \ge 0$ for all $i = 1,\dots,n$. This assures that $t_i = \lambda$ if $x_i = 1$ and $t_i = 0$ if $x_i = 0$. The resulting final model will be a mixed integer linear programming one and can be solved by using a standard software. Of course, the same technique can be applied to other problems with totally unimodular constraints matrix. □

5 Conclusions

In this chapter we have discussed a general class of combinatorial optimization problems with fuzzy weights. We have provided an interpretation of such problems in the setting of possibility theory. The possibility and necessity measures allow us to characterize the optimality of solutions and elements and to define a solution concept. This solution concept is an adaptation of the necessary soft optimality first proposed for fuzzy linear programming. In general, every fuzzy problem boils down to solving a small number of interval problems. Every algorithm for computing a deviation interval and a min-max regret solution under interval weights can be easily adopted to solve a more general fuzzy problem. It is enough to apply a standard binary search technique. The complexity of an interval problem depends on the type of the cost function in its deterministic counterpart. In general, the problems with bottleneck cost function are easier to solve than the ones with linear sum cost.

There are some open questions concerning the approach described in this chapter. Most of them refer to the interval uncertainty representation. For instance, the problem of evaluating the necessary optimality of elements is open (its complexity is known only for some particular problems). Also, designing fast algorithms for computing optimal min-max regret solutions is an important subject of further research. For fuzzy problems, the efficiency of the MIP formulation should be investigated. Also, the parametric techniques, which allow us to compute fuzzy deviation intervals and solve the bottleneck problems should be explored more deeply. Finally, if the interval problem is NP-hard, then some heuristics and approximation algorithms for its fuzzy generalization should be designed.

Acknowledgements. The second author was partially supported by Polish Committee for Scientific Research, grant N N111 1464 33.

References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms, and Applications. Prentice-Hall, Englewood Cliffs (1993)
2. Aissi, H., Bazgan, C., Vanderpooten, D.: Min-max and min-max regret versions of combinatorial optimization problems: A survey. European Journal of Operational Research (2008) doi:10.1016/j.ejor.2008.09.012
3. Aissi, H., Bazgan, C., Vanderpooten, D.: Complexity of the min-max and min-max regret assignment problems. Operations Research Letters 33(6), 634–640 (2005)
4. Aissi, H., Bazgan, C., Vanderpooten, D.: Complexity of the min-max (regret) versions of cut problems. In: Deng, X., Du, D.-Z. (eds.) ISAAC 2005. LNCS, vol. 3827, pp. 789–798. Springer, Heidelberg (2005)
5. Aron, I.D., Hentenryck, P.V.: A constraint satisfaction approach to the robust spanning tree problem with interval data. In: Darwiche, A., Friedman, N. (eds.) UAI 2002, Proceedings of the 18th Conference in Uncertainty in Artificial Intelligence, University of Alberta, Edmonton, Alberta, Canada, August 1-4, pp. 18–25. Morgan Kaufmann, San Francisco (2002)

6. Aron, I.D., Hentenryck, P.V.: On the complexity of the robust spanning tree problem with interval data. Operations Research Letters 32(1), 36–40 (2004)
7. Averbakh, I.: Minmax regret solutions for minimax optimization problems with uncertainty. Operations Research Letters 27(2), 57–65 (2000)
8. Averbakh, I.: On the complexity of a class of combinatorial optimization problems with uncertainty. Mathematical Programming 90, 263–272 (2001)
9. Averbakh, I., Lebedev, V.: On the complexity of minmax regret linear programming. European Journal of Operational Research 160(1), 227–231 (2005)
10. Camerini, P.M.: The minimax spanning tree problem and some extensions. Information Processing Letters 7, 10–14 (1978)
11. Chan, T.: Finding the shortest bottleneck edge in a parametric minimum spanning tree. In: Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 232–240 (2005)
12. Conde, E.: An improved algorithm for selecting p items with uncertain returns according to the minmax-regret criterion. Mathematical Programming 100(2), 345–353 (2004)
13. Dubois, D., Fargier, H., Galvagnon, V.: On latest starting times and floats in activity networks with ill-known durations. European Journal of Operational Research 147, 266–280 (2003)
14. Dubois, D., Prade, H.: Possibility Theory: An Approach to Computerized Processing of Uncertainty. Plenum Press (1988)
15. Fernandez-Baca, D., Slutzki, G.: Solving parametric problems on trees. Journal of Algorithms 10 (1989)
16. Fortin, J., Zieliński, P., Dubois, D., Fargier, H.: Interval analysis in scheduling. In: van Beek, P. (ed.) CP 2005. LNCS, vol. 3709, pp. 226–240. Springer, Heidelberg (2005)
17. Gabow, H.N., Tarjan, R.E.: Algorithms for two bottleneck optimization problems. Journal of Algorithms 9, 411–417 (1988)
18. Garfinkel, R.S., Nemhauser, G.L.: Integer Programming. John Wiley & Sons, Chichester (1972)
19. Inuiguchi, M.: On possibilistic/fuzzy optimization. In: Melin, P., Castillo, O., Aguilar, L.T., Kacprzyk, J., Pedrycz, W. (eds.) IFSA 2007. LNCS (LNAI), vol. 4529, pp. 351–360. Springer, Heidelberg (2007)
20. Inuiguchi, M., Sakawa, M.: Robust optimization under softness in a fuzzy linear programming problem. International Journal of Approximate Reasonning 18(1-2), 21–34 (1998)
21. Kall, P., Wallace, S.W.: Stochastic Programming. Wiley, Chichester etc. (1994)
22. Kasperski, A.: Discrete Optimization with Interval Data. Minmax Regret and Fuzzy Approach. Studies in Fuzziness and Soft Computing, vol. 228. Springer, Heidelberg (2008)
23. Kasperski, A., Zieliński, P.: An approximation algorithm for interval data minmax regret combinatorial optimization problems. Information Processing Letters 97(5), 177–180 (2006)
24. Kasperski, A., Zieliński, P.: The robust shortest path problem in series-parallel multidigraphs with interval data. Operations Research Letters 34(1), 69–76 (2006)
25. Kasperski, A., Zieliński, P.: On combinatorial optimization problems on matroids with uncertain weights. European Journal of Operational Research 177(2), 851–864 (2007)
26. Kasperski, A., Zieliński, P.: On the existence of an FPTAS for minmax regret combinatorial optimization problems with interval data. Operations Research Letters 35(4), 525–532 (2007)
27. Kasperski, A., Zieliński, P.: Using gradual numbers for solving fuzzy-valued combinatorial optimization problems. In: Melin, P., Castillo, O., Aguilar, L.T., Kacprzyk, J., Pedrycz, W. (eds.) IFSA 2007. LNCS (LNAI), vol. 4529, pp. 656–665. Springer, Heidelberg (2007)

28. Kasperski, A., Zieliński, P.: Minmax regret approach and optimality evaluation in combinatorial optimization problems with interval and fuzzy weights. European Journal of Operational Research (2009) (to appear)
29. Kouvelis, P., Yu, G.: Robust Discrete Optimization and Its Applications. Kluwer Academic Publishers, Boston (1997)
30. Lawler, E.: Combinatorial Optimization: Networks and Matroids. Saunders College Publishing, Fort Worth (1976)
31. Luce, R.D., Raiffa, H.: Games and Decisions. Introduction and Critical Survey. Dover Publications, Inc., New York (1957)
32. Montemanni, R.: A benders decomposition approach for the robust spanning tree problem with interval data. European Journal of Operational Research 174(3), 1479–1490 (2006)
33. Montemanni, R., Gambardella, L.M.: A branch and bound algorithm for the robust spanning tree problem with interval data. European Journal of Operational Research 161(3), 771–779 (2005)
34. Montemanni, R., Gambardella, L.M., Donati, A.V.: A branch and bound algorithm for the robust shortest path problem with interval data. Operations Research Letters 32(3), 225–232 (2004)
35. Papadimitriou, C.H., Steiglitz, K.: Combinatorial Optimization: Algorithms and Complexity. Dover Pub. (1998)
36. Punnen, A.P.: A linear time algorithm for the maximum capacity path problem. European Journal of Operational Research 53, 402–404 (1991)
37. Punnen, A.P.: A fast algorithm for a class of bottleneck problems. Computing 56, 397–401 (1996)
38. Raghavan, S., Ball, M.O., Trichur, V.S.: Bicriteria product design optimization: An efficient solution procedure using and/or trees. Naval Research Logistics 49, 574–592 (2002)
39. Savage, L.J.: The Foundations of Statistics. John Wiley and Sons, New York (1951)
40. Yaman, H., Karasan, O.E., Pinar, M.Ç.: The robust spanning tree problem with interval data. Operations Research Letters 29(1), 31–40 (2001)
41. Young, N., Tarjan, R., Orlin, J.: Faster parametric shortest path minimum-balance algorithms. Networks 21, 205–221 (1991)
42. Zieliński, P.: The computational complexity of the relative robust shortest path problem with interval data. European Journal of Operational Research 158(3), 570–576 (2004)