On Recoverable and Two-Stage Robust Selection Problems with Budgeted Uncertainty

André Chassein[1], Marc Goerigk[2], Adam Kasperski[3], and Paweł Zieliński[4]

[1]Fachbereich Mathematik, Technische Universität Kaiserslautern, Germany
[2]Department of Management Science, Lancaster University, United Kingdom
[3]Faculty of Computer Science and Management, Wrocław University of Science and Technology, Poland
[4]Faculty of Fundamental Problems of Technology, Wrocław University of Science and Technology, Poland

July 31, 2017

Abstract

In this paper the problem of selecting p out of n available items is discussed, such that their total cost is minimized. We assume that the item costs are not known exactly, but stem from a set of possible outcomes modeled through budgeted uncertainty sets, i.e., the interval uncertainty sets with an additional linear (budget) constraint, in their discrete and continuous variants. Robust recoverable and two-stage models of this selection problem are analyzed through an in-depth discussion of variables at their optimal values. Polynomial algorithms for both models under continuous budgeted uncertainty are proposed. In the case of discrete budgeted uncertainty, compact mixed integer formulations are constructed and some approximation algorithms are proposed. Polynomial combinatorial algorithms for the adversarial and incremental problems (the special cases of the considered robust models) under both discrete and continuous budgeted uncertainty are constructed.

Keywords: combinatorial optimization; robust optimization; selection problem; budgeted uncertainty; two-stage robustness; recoverable robustness

1 Introduction

In this paper we consider the following Selection problem. We are given a set of n items with cost c_i for each $i \in [n] := \{1, \dots, n\}$ and an integer $p \in [n]$. We seek a subset

$X \subseteq [n]$ of p items, $|X| = p$, whose total cost $\sum_{i \in X} c_i$ is minimum. It is easy to see that an optimal solution is composed of p items of the smallest cost. It can be found in $O(n)$ time by using the well-known fact, that the pth smallest item can be found in $O(n)$ time (see, e.g., [11]). SELECTION is a basic resource allocation problem [18]. It is also a special case of 0-1 knapsack, 0-1 assignment, single machine scheduling, and minimum matroid base problems (see [20] for an overview). It can be formulated as the following integer linear program:

$$\begin{array}{lll} \min & \sum_{i\in[n]} c_i x_i & \\ \text{s.t.} & \sum_{i\in[n]} x_i = p & \\ & x_i \in \{0,1\} & \forall i \in [n]. \end{array} \tag{1}$$

We will use $\Phi \subseteq \{0,1\}^n$ to denote the set of all feasible solutions to (1). Given $\boldsymbol{x} \in \{0,1\}^n$, we also define $X_{\boldsymbol{x}} = \{i \in [n] : x_i = 1\}$, and $\overline{X}_{\boldsymbol{x}} = [n] \setminus X_{\boldsymbol{x}}$, i.e. $X_{\boldsymbol{x}}$ is the item set induced by vector $\boldsymbol{x}$ and $\overline{X}_{\boldsymbol{x}}$ denotes its complement.

Consider the case when the item costs are uncertain. As a part of the input, we are given a scenario set $\mathcal{U}$, containing all possible vectors of the item costs, called *scenarios*. Several methods of defining $\mathcal{U}$ have been proposed in the existing literature (see, e.g., [3, 4, 19, 24, 27]). Under *discrete uncertainty* (see, e.g., [24]), the scenario set contains K distinct scenarios i.e. $\mathcal{U}^D = \{\boldsymbol{c}_1, \dots, \boldsymbol{c}_K\}$, $\boldsymbol{c}_i \in \mathbb{R}^n_+$. Under *interval uncertainty*, the cost of each item $i \in [n]$ belongs to a closed interval $[\underline{c}_i, \overline{c}_i]$, where $d_i := \overline{c}_i - \underline{c}_i \geq 0$ is the maximal deviation of the cost of i from its *nominal value* $\underline{c}_i$. In the traditional interval uncertainty representation, $\mathcal{U}^I$ is the Cartesian product of all the intervals (see, e.g., [24]). In this paper we will focus on the following two generalizations of scenario set $\mathcal{U}^I$, which have been examined in [3, 4, 27]:

- *Continuous budgeted uncertainty*:

$$\mathcal{U}^c = \{(\underline{c}_i + \delta_i)_{i\in[n]} : \delta_i \in [0, d_i], \sum_{i\in[n]} \delta_i \leq \Gamma\} \subseteq \mathbb{R}^n_+.$$

- *Discrete budgeted uncertainty*:

$$\mathcal{U}^d = \{(\underline{c}_i + \delta_i)_{i\in[n]} : \delta_i \in \{0, d_i\}, |\{i \in [n] : \delta_i = d_i\}| \leq \Gamma\} \subseteq \mathbb{R}^n_+.$$

The fixed parameter $\Gamma \geq 0$ is called a *budget* and it controls the amount of uncertainty which an adversary can allocate to the item costs. For a sufficiently large Γ, $\mathcal{U}^c$ reduces to $\mathcal{U}^I$, and $\mathcal{U}^d$ reduces to the extreme points of $\mathcal{U}^I$.

In order to compute a solution, under a specified scenario set $\mathcal{U}$, we can follow a robust optimization approach. For general overviews on robust optimization, see, e.g., [1, 15, 22, 24, 28]. In a typical, single-stage robust model we seek a solution minimizing the total cost in a worst case. This leads to the following *minmax* and *minmax regret* problems:

$$\text{MINMAX}: \min_{\boldsymbol{x}\in\Phi} \max_{\boldsymbol{c}\in\mathcal{U}} \boldsymbol{c}\boldsymbol{x},$$

$$\text{MINMAX-REGRET}: \min_{\boldsymbol{x}\in\Phi} \max_{\boldsymbol{c}\in\mathcal{U}} \max_{\boldsymbol{y}\in\Phi} (\boldsymbol{c}\boldsymbol{x} - \boldsymbol{c}\boldsymbol{y}).$$

The minmax (regret) versions of the SELECTION problem have been discussed in the existing literature. For scenario set $\mathcal{U}^D$ both problems are NP-hard even for $K = 2$ (the number of scenarios equals 2) [2]. If K is part of the input, then MINMAX and MINMAX-REGRET are strongly NP-hard and not approximable within any constant factor [20]. On the other hand, MINMAX is approximable within $O(\log K/ \log\log K)$ [13] but MINMAX-REGRET is only known to be approximable within K, which is due to the results given in [1]. The MINMAX problem under scenario sets $\mathcal{U}^c$ and $\mathcal{U}^d$ is polynomially solvable, according to the results obtained in [3]. Also, MINMAX-REGRET, under scenario set $\mathcal{U}^I$, is polynomially solvable by the algorithms designed in [2, 10].

The problems which arises in practice often have a two-stage nature. Namely, a partial solution is computed in the first stage and completed in the second stage, or a complete solution is formed in the first stage and modified to some extent in the second stage. Typically, the costs in the first stage are precisely known, while the costs in the second stage are uncertain. Before we formally define the two-stage models, let us introduce some additional notation:

- $\Phi_1 = \{\boldsymbol{x} \in \{0,1\}^n : \sum_{i\in[n]} x_i \leq p\}$,
- $\Phi_{\boldsymbol{x}} = \{\boldsymbol{y} \in \{0,1\}^n : \sum_{i\in[n]}(x_i + y_i) = p, x_i + y_i \leq 1, i \in [n]\},\ \ \boldsymbol{x} \in \Phi_1$,
- $\Phi^k_{\boldsymbol{x}} = \{\boldsymbol{y} \in \{0,1\}^n : \sum_{i\in[n]} y_i = p,\ \sum_{i\in[n]} x_i y_i \geq p - k\},\ \ \boldsymbol{x} \in \Phi, k \in [p] \cup \{0\}$.

If $\boldsymbol{y} \in \Phi_{\boldsymbol{x}}$, then $X_{\boldsymbol{x}} \cap X_{\boldsymbol{y}} = \emptyset$ and $|X_{\boldsymbol{x}} \cup X_{\boldsymbol{y}}| = p$. Hence $\Phi_{\boldsymbol{x}}$ encodes all subsets of the item set $[n]$, which added to $X_{\boldsymbol{x}}$ form a complete solution of cardinality p. Set $\Phi^k_{\boldsymbol{x}}$ is called a *recovery set*, k is a given *recovery parameter*. If $\boldsymbol{y} \in \Phi^k_{\boldsymbol{x}}$, then $|X_{\boldsymbol{x}} \setminus X_{\boldsymbol{y}}| = |X_{\boldsymbol{y}} \setminus X_{\boldsymbol{x}}| \leq k$, so $\Phi^k_{\boldsymbol{x}}$ encodes all solutions which can be obtained from $X_{\boldsymbol{x}}$ by exchanging up to k items. Let $\boldsymbol{C} = (C_1, \dots, C_n)$ be a vector of the first stage item costs, which are assumed to be precisely known. Let scenario set $\mathcal{U}$ contain all possible vectors of the uncertain second stage costs. Given $k \in [p] \cup \{0\}$, we study the following *robust recoverable selection* problem:

$$\text{RREC} : \min_{\boldsymbol{x}\in\Phi} \left(\boldsymbol{C}\boldsymbol{x} + \max_{\boldsymbol{c}\in\mathcal{U}} \min_{\boldsymbol{y}\in\Phi^k_{\boldsymbol{x}}} \boldsymbol{c}\boldsymbol{y} \right). \tag{2}$$

In RREC a complete solution (exactly p items) is chosen in the first stage. Then, after a scenario from $\mathcal{U}$ reveals, one can exchange optimally up to k items in the second stage. Notice that if $k = 0$ and $C_i = 0$ for each $i \in [n]$, then RREC becomes the MINMAX problem. The robust recoverable model for linear programming, together with some applications, was discussed in [25]. It has been also recently applied to the shortest path [5], spanning tree [16, 17], knapsack [6] and traveling salesman problems [8]. The RREC problem under scenario sets $\mathcal{U}^D$ and $\mathcal{U}^I$ has been recently discussed in [21]. Under $\mathcal{U}^D$ it turned out to be NP-hard for constant K, strongly NP-hard and not at all approximable when K is part of the input (this is true even if $k = 1$). On the other hand, under scenario set $\mathcal{U}^I$, a polynomial $O((p - k)n^2)$ time algorithm for RREC has been proposed in [21]. No results for scenario rests $\mathcal{U}^c$ and $\mathcal{U}^d$ have been known to date.

We also analyze the following *robust two-stage selection* problem:

$$\text{R2ST}: \min_{\boldsymbol{x}\in\Phi_1}\left(\boldsymbol{Cx}+\max_{\boldsymbol{c}\in\mathcal{U}}\min_{\boldsymbol{y}\in\Phi_{\boldsymbol{x}}}\boldsymbol{cy}\right), \tag{3}$$

In R2ST we seek a first stage solution, which may contain less than p items. Then, after a scenario from $\mathcal{U}$ reveals, this solution is completed optimally to p items. The robust two-stage model was introduced in [23] for the bipartite matching problem. The R2ST problem has been recently discussed in [21]. It is polynomially solvable under scenario set $\mathcal{U}^I$. For scenario set $\mathcal{U}^D$, the problem becomes strongly NP-hard and it has an approximability lower bound of $\Omega(\log n)$, but it has an $O(\log K + \log n)$ randomized approximation algorithm. No results for scenario sets $\mathcal{U}^c$ and $\mathcal{U}^d$ have been known to date.

Given a first stage solution $\boldsymbol{x}\in\Phi$ (resp. $\boldsymbol{x}\in\Phi_1$), we will also study the following *adversarial problems*:

$$\text{AREC}: \max_{\boldsymbol{c}\in\mathcal{U}}\min_{\boldsymbol{y}\in\Phi^k_{\boldsymbol{x}}}\boldsymbol{cy}, \tag{4}$$

$$\text{A2ST}: \max_{\boldsymbol{c}\in\mathcal{U}}\min_{\boldsymbol{y}\in\Phi_{\boldsymbol{x}}}\boldsymbol{cy}. \tag{5}$$

If, additionally, scenario $\boldsymbol{c}\in\mathcal{U}$ is fixed, then we get the following *incremental problems*:

$$\text{IREC}: \min_{\boldsymbol{y}\in\Phi^k_{\boldsymbol{x}}}\boldsymbol{cy}, \tag{6}$$

$$\text{I2ST}: \min_{\boldsymbol{y}\in\Phi_{\boldsymbol{x}}}\boldsymbol{cy}. \tag{7}$$

The adversarial and incremental versions of some network problems were discussed in [12, 27]. The incremental versions of the shortest path and the spanning tree problems are polynomially solvable [12], whereas the adversarial versions of these problems under scenario set $\mathcal{U}^d$ are strongly NP-hard [14, 26, 27].

A summary of the results for $\mathcal{U}^D$ and $\mathcal{U}^I$ obtained in [21] is presented in Table 1.

Table 1: The known results for $\mathcal{U}^D$ and $\mathcal{U}^I$ obtained in [21].

$\mathcal{U}$	IREC	AREC	RREC	I2ST	A2ST	R2ST
$\mathcal{U}^D$	$O(n)$	$O(Kn)$	NP-hard for const. K; str. NP-hard not at all appr. for unbounded K	$O(n)$	$O(Kn)$	NP-hard for const. K; str. NP-hard appr. in $O(\log K + \log n)$ $\Omega(\log n)$ approximability lower bound for unbounded K
$\mathcal{U}^I$	$O(n)$	$O(n)$	$O((p-k)n^2)$	$O(n)$	$O(n)$	$O(n)$

New results. All new results for scenario sets $\mathcal{U}^c$ and $\mathcal{U}^d$, shown in this paper, are summarized in Table 2. In particular, we show that all the considered problems are polynomially solvable under scenario set $\mathcal{U}^c$. The polynomial algorithms for RREC and R2ST under $\mathcal{U}^c$ are based on solving a polynomial number of linear programming subproblems. We also provide polynomial time combinatorial algorithms for AREC and A2ST under both $\mathcal{U}^c$ and $\mathcal{U}^d$. The complexity of RREC and R2ST under $\mathcal{U}^d$ remains open. For these problems we construct compact MIP formulations and propose approximation algorithms. To achieve these results, we provide an in-depth analysis of problem variables at optimality. In itself, this is a key contribution facilitating the understanding of our models, providing insight to a decision maker, and giving an intuitive interpretation of the problem complexity.

Table 2: New results for $\mathcal{U}^c$ and $\mathcal{U}^d$ shown in this paper.

$\mathcal{U}$	IREC	AREC	RREC	I2ST	A2ST	R2ST
$\mathcal{U}^c$	$O(n)$	$O(n^2)$ $O(n \log n)$	poly. sol. compact MIP	$O(n)$	$O(n^2)$ $O(n \log n)$	poly. sol. compact MIP
$\mathcal{U}^d$	$O(n)$	$O(n^3)$	compact MIP	$O(n)$	$O(n^2)$	compact MIP

2 Continuous Budgeted Uncertainty

In this section we address the recoverable and two-stage robust selection problems, RREC (2) and R2ST (3), under the continuous budgeted uncertainty $\mathcal{U}^c$. We first consider the incremental and adversarial problems, IREC (6) and AREC (4), which are inner ones of RREC. We will show that IREC can be solved in $O(n)$ time. The AREC is more involved problem for which we will build a nontrivial $O(n^2)$ algorithm (its complexity can be reduced to $O(n \log n)$, if we apply more clever data structure). Finally, we will show that RREC can be solved in polynomial time. We then investigate the two-stage setting. In order to solve the incremental and adversarial problems, I2ST (7) and A2ST (5), we will apply the results obtained for IREC and AREC and provide for both problems $O(n)$ and $O(n^2)$ algorithms, respectively. We will finish by showing that R2ST can be solved in polynomial time.

2.1 Recoverable Robust Selection

2.1.1 The incremental problem

Given $\boldsymbol{x} \in \Phi$ and $\boldsymbol{c} \in \mathcal{U}$, the incremental problem, IREC (6), can be formulated as the following linear program (notice that the constraints $y_i \in \{0, 1\}$ can be relaxed):

$$\begin{aligned} opt_1 = \min \quad & \sum_{i\in[n]} c_i y_i \\ \text{s.t.} \quad & \sum_{i\in[n]} y_i = p \\ & \sum_{i\in[n]} x_i y_i \geq p - k \\ & y_i \in [0,1] \qquad i \in [n] \end{aligned} \tag{8}$$

It is easy to see that the IREC problem can be solved in $O(n)$ time. Indeed, we first choose $p-k$ items of the smallest cost from $X_{\boldsymbol{x}}$ and then k items of the smallest cost from the remaining items. We will now show some additional properties of (8), which will be used extensively later. The dual to (8) is

$$\begin{array}{lll} \max & p\alpha + (p-k)\beta - \sum_{i\in[n]} \gamma_i & \\ \text{s.t.} & \alpha + x_i\beta \leq \gamma_i + c_i & i \in [n] \\ & \beta \geq 0 & \\ & \gamma_i \geq 0 & i\in[n] \end{array} \tag{9}$$

From now on, we will assume that $k>0$ (the case $k=0$ is trivial, since $\boldsymbol{y}=\boldsymbol{x}$ holds). Let $b(\boldsymbol{c})$ be the pth smallest item cost for the items in $[n]$ under $\boldsymbol{c}$ (i.e. if $c_{\sigma(1)} \leq \cdots \leq c_{\sigma(n)}$ is the ordered sequence of the item costs under $\boldsymbol{c}$, then $b(\boldsymbol{c}) = c_{\sigma(p)}$). Similarly, let $b_1(\boldsymbol{c})$ be the $(p-k)$th smallest item cost for the items in $X_{\boldsymbol{x}}$ and $b_2(\boldsymbol{c})$ be the kth smallest item cost for the items in $\overline{X}_{\boldsymbol{x}}$ under $\boldsymbol{c}$. The following proposition characterizes the optimal values of α and β in (9), and is fundamental in the following analysis:

Proposition 1. *Given scenario $\boldsymbol{c}\in\mathcal{U}$, the following conditions hold:*

1. *if $b_1(\boldsymbol{c}) \leq b(\boldsymbol{c})$, then $\alpha=b(\boldsymbol{c})$ and $\beta=0$ are optimal in (9),*
2. *if $b_1(\boldsymbol{c}) > b(\boldsymbol{c})$, then $\alpha=b_2(\boldsymbol{c})$ and $\beta=b_1(\boldsymbol{c})-b_2(\boldsymbol{c})$ are optimal in (9).*

Proof. By replacing γ_i by $[\alpha+\beta x_i - c_i]_+$, the dual problem (9) can be represented as follows:

$$\max_{\alpha,\beta\geq 0} f(\alpha,\beta) = \max_{\alpha,\beta\geq 0}\left\{ p\alpha + (p-k)\beta - \sum_{i\in[n]}[\alpha+\beta x_i - c_i]_+ \right\}, \tag{10}$$

where $[a]_+ = \max\{0,a\}$. Let us sort the items in $[n]$ so that that $c_{\sigma(1)} \leq \cdots \leq c_{\sigma(n)}$. Let us sort the items in $X_{\boldsymbol{x}}$ so that $c_{\nu(1)} \leq \cdots \leq c_{\nu(p)}$ and the items in $\overline{X}_{\boldsymbol{x}}$ so that $c_{\varsigma(1)} \leq \cdots \leq c_{\varsigma(n-p)}$. We distinguish two cases. The first one: $c_{\nu(p-k)} \leq c_{\sigma(p)}$ ($b_1(\boldsymbol{c}) \leq b(\boldsymbol{c})$). Then it is possible to construct an optimal solution to (8) with the cost equal to $\sum_{i\in[p]} c_{\sigma(i)}$. Namely, we choose $p-k$ items of the smallest costs from $X_{\boldsymbol{x}}$ and k items of the smallest cost from the remaining items. Fix $\alpha=c_{\sigma(p)}$ and $\beta=0$, which gives the case 1. By using (10), we obtain $f(\alpha,\beta)=\sum_{i\in[p]} c_{\sigma(i)} = opt_1$ and the proposition follows from the weak duality theorem. The second case: $c_{\nu(p-k)} > c_{\sigma(p)}$ ($b_1(\boldsymbol{c}) > b(\boldsymbol{c})$). The optimal solution to (8) is then formed by the items $\nu(1),\dots,\nu(p-k)$ and $\varsigma(1),\dots,\varsigma(k)$. Fix $\alpha = c_{\varsigma(k)}$ and $\beta = c_{\nu(p-k)} - c_{\varsigma(k)}$, which gives the case 2. By (10), we have

$$\begin{aligned} f(\alpha,\beta) &= p\alpha + (p-k)\beta - \sum_{i\in X_{\boldsymbol{x}}}[\alpha+\beta-c_i]_+ - \sum_{i\in\overline{X}_{\boldsymbol{x}}}[\alpha-c_i]_+ \\ &= pc_{\varsigma(k)} + (p-k)(c_{\nu(p-k)}-c_{\varsigma(k)}) - \sum_{i\in X_{\boldsymbol{x}}}[c_{\nu(p-k)}-c_i]_+ - \sum_{i\in\overline{X}_{\boldsymbol{x}}}[c_{\varsigma(k)}-c_i]_+ \\ &= pc_{\varsigma(k)} + (p-k)(c_{\nu(p-k)}-c_{\varsigma(k)}) - (p-k)c_{\nu(p-k)} + \sum_{i\in[p-k]} c_{\nu(i)} - kc_{\varsigma(k)} + \sum_{i\in[k]} c_{\varsigma(i)} \end{aligned}$$

$$= \sum_{i\in[p-k]} c_{\nu(i)} + \sum_{i\in[k]} c_{\varsigma(i)} = opt_1$$

and the proposition follows from the weak duality theorem. □

2.1.2 The adversarial problem

Consider the adversarial problem AREC (4) for a given solution $\boldsymbol{x} \in \Phi$. We will again assume that $k > 0$. If $k = 0$, then all the budget Γ is allocated to the items in $X_{\boldsymbol{x}}$. Scenario $\boldsymbol{c} \in \mathcal{U}^c$ which maximizes the objective value in this problem is called a *worst scenario* for $\boldsymbol{x}$ (worst scenario for short). We now give a characterization of a worst scenario.

Proposition 2. *There is a worst scenario $\boldsymbol{c} = (\underline{c}_i + \delta_i)_{i\in[n]} \in \mathcal{U}^c$ such that*

1. *$b_1(\boldsymbol{c}) \le b(\boldsymbol{c})$ or*
2. *$b_1(\boldsymbol{c})$ or $b_2(\boldsymbol{c})$ belongs to $\mathcal{D} = \{\underline{c}_1, \dots, \underline{c}_n, \overline{c}_1, \dots, \overline{c}_n\}$.*

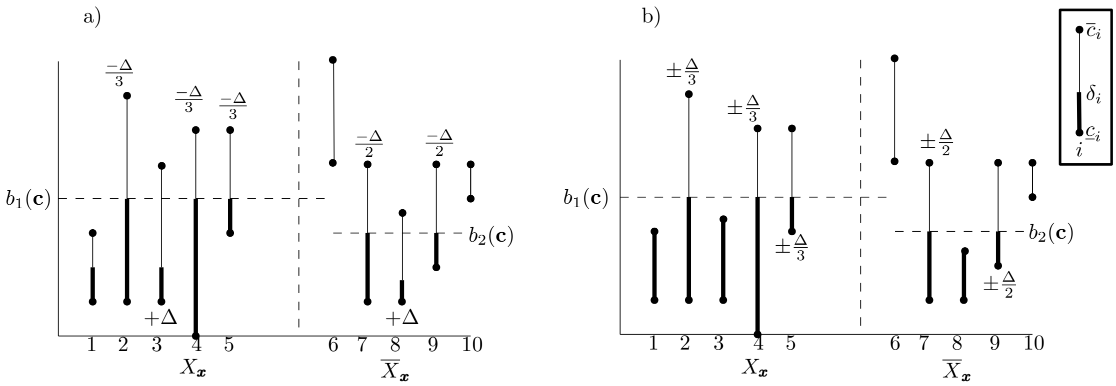

Figure 1: Illustration of the proof for $n = 10$, $p = 5$, $k = 2$, and $X_{\boldsymbol{x}} = \{1, \dots, 5\}$.

Proof. Let $\boldsymbol{c}$ be a worst scenario and assume that $b_1(\boldsymbol{c}) > b(\boldsymbol{c})$ and both $b_1(\boldsymbol{c})$ and $b_2(\boldsymbol{c})$ do not belong to $\mathcal{D}$. The idea of the proof is to show that there is also a worst scenario satisfying condition 1 or 2. Note that $b_1(\boldsymbol{c}) > b(\boldsymbol{c})$ implies $b_1(\boldsymbol{c}) > b_2(\boldsymbol{c})$. Let $A = \{i \in X_{\boldsymbol{x}} : \underline{c}_i + \delta_i = b_1(\boldsymbol{c})\}$ and $B = \{i \in \overline{X}_{\boldsymbol{x}} : \underline{c}_i + \delta_i = b_2(\boldsymbol{c})\}$. Observe that $A, B \neq \emptyset$ by the definition of $b_1(\boldsymbol{c})$ and $b_2(\boldsymbol{c})$. Also, a positive budget must be allocated to each item in A and B. In Figures 1a and 1b we have $A = \{2, 4, 5\}$ and $B = \{7, 9\}$. Let k_1 be the number of items in $X_{\boldsymbol{x}}$ such that $\underline{c}_i + \delta_i < b_1(\boldsymbol{c})$ and k_2 be the number of items in $\overline{X}_{\boldsymbol{x}}$ such that $\underline{c}_i + \delta_i < b_2(\boldsymbol{c})$. In the sample problem (see Figures 1a and 1b) we have $k_1 = 2$ and $k_2 = 1$. Suppose that there is an item $j \in X_{\boldsymbol{x}}$ such that $\underline{c}_j + \delta_j < b_1(\boldsymbol{c})$ and $\delta_j < d_j$ (see the items 1 and 3 in Figure 1a). Let us transform scenario $\boldsymbol{c} \in \mathcal{U}^c$ into scenario $\boldsymbol{c}_1 \in \mathcal{U}^c$ as follows: $\delta_j := \delta_j + \Delta$ and $\delta_i := \delta_i - \Delta/|A|$ for each $i \in A$, where $\Delta > 0$ is a sufficiently small number, i.e. such that $\underline{c}_j + \delta_j + \Delta \le \min\{b_1(\boldsymbol{c}), \underline{c}_j + d_j\}$ and

$\delta_i - \Delta/|A| \geq 0$ for each $i \in A$. Let $\boldsymbol{y}$ be an optimal solution under $\boldsymbol{c}$ and let $\boldsymbol{y}_1$ be an optimal solution under $\boldsymbol{c}_1$. The following equality holds

$$\boldsymbol{c}_1\boldsymbol{y}_1 = \boldsymbol{c}\boldsymbol{y} + \Delta - (p - k - k_1)\frac{\Delta}{|A|}.$$

Since $|A| + k_1 \geq p - k$, $\boldsymbol{c}_1\boldsymbol{y}_1 \geq \boldsymbol{c}\boldsymbol{y}$ and $\boldsymbol{c}_1$ is also a worst scenario. We can increase Δ so that the condition 1 or 2 of the proposition is satisfied, or $b_1(\boldsymbol{c}_1) = \underline{c}_j + \delta_j$, or $\underline{c}_j + \delta_j = \underline{c}_j + d_j$ (see Figure 1a). The same reasoning can be applied to every item $j \in \overline{X}_{\boldsymbol{x}}$ such that $\underline{c}_j + \delta_j < b_2(\boldsymbol{c})$ and $\delta_j < d_j$ (see the item 8 in Figure 1a). So, it remains to analyze the case shown in Figure 1b. Let us again choose some sufficiently small $\Delta > 0$. Define scenario $\boldsymbol{c}_1$ by modifying $\boldsymbol{c}$ in the following way $\delta_i := \delta_i + \Delta/|A|$ for each $i \in A$ and $\delta_i := \delta_i - \Delta/|B|$ for each $i \in B$. Similarly, define scenario $\boldsymbol{c}_2$ by modifying $\boldsymbol{c}$ as follows $\delta_i := \delta_i - \Delta/|A|$ for each $i \in A$ and $\delta_i := \delta_i + \Delta/|B|$ for each $i \in B$. Let $\boldsymbol{y}_1$ be an optimal solution under $\boldsymbol{c}_1$ and $\boldsymbol{y}_2$ be an optimal solution under $\boldsymbol{c}_2$. The following equalities hold

$$\boldsymbol{c}_1\boldsymbol{y}_1 = \boldsymbol{c}\boldsymbol{y} + (p - k - k_1)\frac{\Delta}{|A|} - (k - k_2)\frac{\Delta}{|B|},$$

$$\boldsymbol{c}_2\boldsymbol{y}_2 = \boldsymbol{c}\boldsymbol{y} - (p - k - k_1)\frac{\Delta}{|A|} + (k - k_2)\frac{\Delta}{|B|}.$$

Hence, either $\boldsymbol{c}_1\boldsymbol{y}_1 \geq \boldsymbol{c}\boldsymbol{y}$ or $\boldsymbol{c}_2\boldsymbol{y}_2 \geq \boldsymbol{c}\boldsymbol{y}$, so $\boldsymbol{c}_1$ or $\boldsymbol{c}_2$ is also a worst scenario. We can now increase Δ until $\boldsymbol{c}_1$ ($\boldsymbol{c}_2$) satisfies condition 1 or 2. □

Using (9) and the definition of scenario set $\mathcal{U}^c$, we can represent AREC as the following linear programming problem:

$$\begin{array}{lll} \max & p\alpha + (p-k)\beta - \sum_{i\in[n]} \gamma_i & \\ \text{s.t.} & \alpha + x_i\beta \leq \gamma_i + \underline{c}_i + \delta_i & \forall i \in [n] \\ & \sum_{i\in[n]} \delta_i \leq \Gamma & \\ & \delta_i \leq d_i & \forall i \in [n] \\ & \beta \geq 0 & \\ & \gamma_i, \delta_i \geq 0 & i \in [n] \end{array} \tag{11}$$

Thus AREC can be solved in polynomial time. In the following we will construct a strongly polynomial combinatorial algorithm for solving AREC. The following corollary is a consequence of Proposition 1 and Proposition 2:

Corollary 3. *There is an optimal solution to (11) in which*

1. $\beta = 0$ *or*
2. α *or* $\alpha + \beta$ *belongs to* $\mathcal{D} = \{\underline{c}_1, \ldots, \underline{c}_n, \overline{c}_1, \ldots, \overline{c}_n\}$.

Proof. According to Proposition 1, there is an optimal solution to (11) which induces a worst scenario $\boldsymbol{c} = (\underline{c}_i + \delta_i)_{i\in[n]} \in \mathcal{U}^c$, which satisfies conditions 1 and 2 of Proposition 1. If the condition 1 is fulfilled, i.e. $b_1(\boldsymbol{c}) \leq b(\boldsymbol{c})$, then according to Proposition 2 we get $\beta = 0$. If $b_1(\boldsymbol{c}) > b(\boldsymbol{c})$, then condition 2 from Proposition 1 and condition 2 from Proposition 2 hold. Both these conditions imply the condition 2 of the corollary. □

Proposition 4. *The optimal values of α and β in (11) can be found by solving the following problem:*

$$\max_{\alpha,\beta\geq 0} \left\{ \alpha p + \beta(p-k) - \max\left\{ \sum_{i\in[n]} [\alpha + \beta x_i - \underline{c}_i]_+ - \Gamma, \sum_{i\in[n]} [\alpha + \beta x_i - \overline{c}_i]_+ \right\} \right\} \tag{12}$$

Proof. Let us first rewrite (11) in the following way:

$$\begin{aligned}
\max \quad & p\alpha + (p-k)\beta - \sum_{i\in[n]} [\alpha + \beta x_i - \underline{c}_i - \delta_i]_+ \\
\text{s.t.} \quad & \sum_{i\in[n]} \delta_i \leq \Gamma \\
& 0 \leq \delta_i \leq d_i \qquad\qquad i \in [n] \\
& \beta \geq 0
\end{aligned} \tag{13}$$

Let us fix α and $\beta \geq 0$ in (13). Then the optimal values of δ_i can be then found by solving the following subproblem:

$$\begin{aligned}
z = \min \quad & \sum_{i\in[n]} [\alpha + \beta x_i - \underline{c}_i - \delta_i]_+ \\
\text{s.t.} \quad & \sum_{i\in[n]} \delta_i \leq \Gamma \\
& 0 \leq \delta_i \leq d_i \qquad\qquad i \in [n]
\end{aligned} \tag{14}$$

Let $U = \sum_{i\in[n]} [\alpha + \beta x_i - \underline{c}_i]_+$. Observe that $[U - \Gamma]_+$ is a lower bound on z as $z \geq 0$ and it is not possible to decrease U by more than Γ. The subproblem (14) can be solved by applying the following greedy method. For $i = 1, \dots, n$, if $\alpha + \beta x_i - \underline{c}_i > 0$, we fix $\delta_i = \min\{\alpha + \beta x_i - \underline{c}_i, d_i, \Gamma\}$ and modify $\Gamma := \Gamma - \delta_i$. If, at some step, $\Gamma = 0$ we have reached the lower bound. Hence $z = [U - \Gamma]_+$. On the other hand if, after the algorithm terminates, we still have $\Gamma > 0$, then $z = \sum_{i\in[n]} [\alpha + \beta x_i - \underline{c}_i - d_i]_+$. In consequence

$$\begin{aligned}
z &= \max\left\{ [U - \Gamma]_+, \sum_{i\in[n]} [\alpha + \beta x_i - \underline{c}_i - d_i]_+ \right\} \\
&= \max\left\{ \sum_{i\in[n]} [\alpha + \beta x_i - \underline{c}_i]_+ - \Gamma, \sum_{i\in[n]} [\alpha + \beta x_i - \overline{c}_i]_+ \right\},
\end{aligned}$$

which together with (13) completes the proof. □

Having the optimal values of α and β, the worst scenario $\boldsymbol{c} = (\underline{c}_i + \delta_i)_{i\in[n]}$, can be found in $O(n)$ time by applying the greedy method to (14), described in the proof of Proposition 4. We now construct an efficient algorithm for solving (12), which will give us the optimal values of α and β. We will illustrate this algorithm by using the sample problem shown in Figure 2.

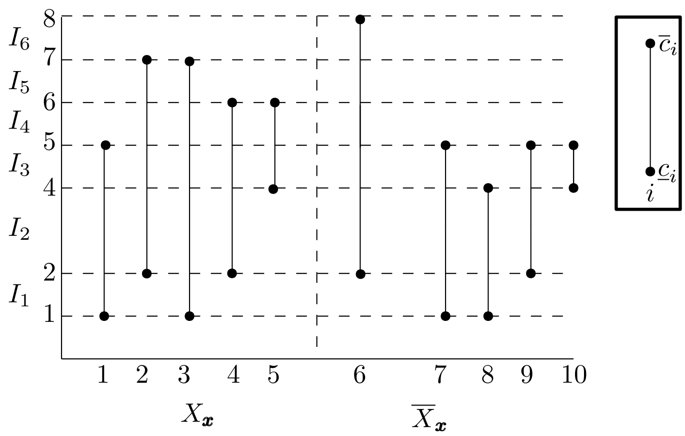

Figure 2: A sample problem with $n = 10$, $p = 5$, $k = 2$, $\Gamma = 24$, and $X_{\boldsymbol{x}} = \{1, \ldots, 5\}$.

Let $h_{(1)} \leq h_{(2)} \leq \cdots \leq h_{(l)}$ be the ordered sequence of the distinct values from $\mathcal{D}$. This sequence defines a family of closed intervals $I_j = [h_{(j)}, h_{(j+1)}]$, $j \in [l-1]$, which partitions the interval $[\min_{i\in[n]} \underline{c}_i, \max_{i\in[n]} \overline{c}_i]$. Notice that $l \leq 2n$. In the example shown in Figure 2 we have six intervals $I_1, \ldots I_6$ which split the interval $[1, 8]$.

By Corollary 3, we need to investigate two cases. In the first case, we have $\beta = 0$. Then (12) reduces to the following problem:

$$\max_{\alpha} f(\alpha) = \max_{\alpha} \left\{ \alpha p - \max\left\{ \sum_{i\in[n]} [\alpha - \underline{c}_i]_+ - \Gamma, \sum_{i\in[n]} [\alpha - \underline{c}_i - d_i]_+ \right\} \right\}. \tag{15}$$

Consider the problem of maximizing $f(\alpha)$ over a fixed interval I_j. It is easy to verify that (15) reduces then to finding the maximum of a minimum of two linear functions of α over I_j. For example, when $\alpha \in I_3 = [4, 5]$, then after an easy computation, the problem (15) becomes

$$\max_{\alpha\in[4,5]} \min\{-5\alpha + 44, 4\alpha + 4\}.$$

It is well known that the maximum value of α is attained at one of the bounds of the interval I_j or at the intersection point of the two linear functions of α. In this case we compute α by solving $-5\alpha + 44 = \alpha + 4$ which yields $\alpha = 4.44$. We can now solve (15) by solving at most $2n$ subproblems consisting in maximizing $f(\alpha)$ over $I_1, \ldots I_l$. Notice, however, that in some cases we do not have to examine all the intervals $I_1, \ldots, I_l$. We can use the fact that α is the pth smallest item cost in the computed scenario. In the example, the optimal value of α belongs to $I_1 \cup I_2 \cup I_3$. The function $f(\alpha)$ for the sample problem is shown in Figure 2. The optimal value of α is 4.44. The scenario corresponding to $\alpha = 4.44$ can be obtained by applying a greedy method and it is also shown in Figure 3.

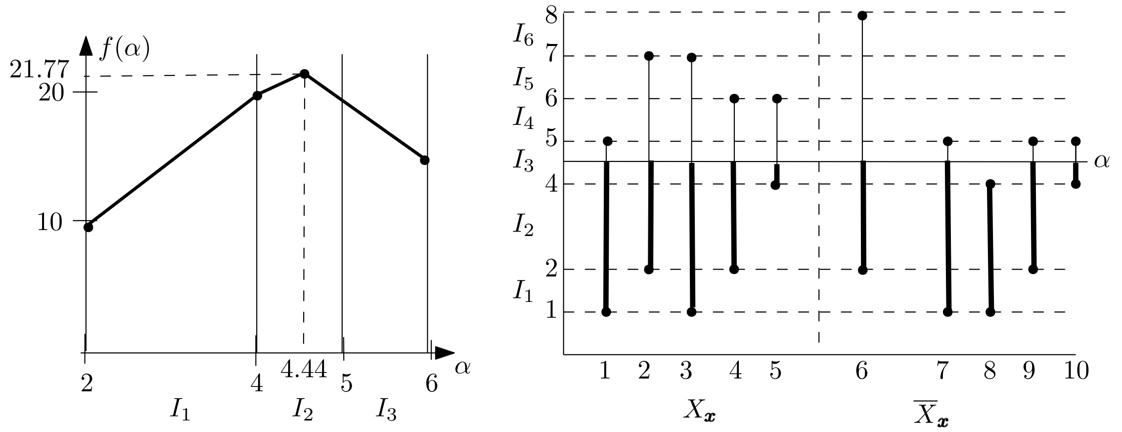

Figure 3: The function $f(\alpha)$ for the sample problem and the worst scenario for the optimal value of $\alpha = 4.44$.

We now discuss the second case in Corollary 3. Let us fix $\gamma = \alpha + \beta$ and rewrite (12) as follows:

$$\max_{\alpha,\gamma\geq\alpha} g(\alpha,\gamma) = \max_{\alpha,\gamma\geq\alpha}\{\alpha k + \gamma(p-k) -$$
$$\max\left\{\sum_{i\in X_{\boldsymbol{x}}}[\gamma - \underline{c}_i]_+ + \sum_{i\in\overline{X}_{\boldsymbol{x}}}[\alpha - \underline{c}_i]_+ - \Gamma, \sum_{i\in X_{\boldsymbol{x}}}[\gamma - \underline{c}_i - d_i]_+ + \sum_{i\in\overline{X}_{\boldsymbol{x}}}[\alpha - \underline{c}_i - d_i]_+\right\}\right\}. \quad (16)$$

According to Corollary 3, the optimal value of α or γ belongs to $\mathcal{D}$. So, let us first fix $\gamma \in \mathcal{D}$ and consider the problem $\max_\alpha g(\alpha,\gamma)$. The optimal value of α can be found by optimizing α over each interval I_j, whose upper bound is not greater then γ (it follows from the constraint $\alpha \leq \gamma$). Again, the problem $\max_{\alpha\in I_j} g(\alpha,\gamma)$ can be reduced to maximizing a minimum of two linear functions of α over a closed interval. To see this consider the sample problem shown in Figure 2. Fix $\gamma = 6$ and assume that $\alpha \in I_2$. Then, a trivial verification shows that

$$\max_{\alpha\in[2,4]} g(\alpha,6) = \max_{\alpha\in[2,4]} \min\{28 - 2\alpha, 2\alpha + 17\}.$$

The maximum is attained when $28 - 2\alpha = 2\alpha + 17$, so for $\alpha = 2.75$. The function $g(\alpha,6)$ is shown in Figure 4. It attains the maximum in the interval I_2 at $\alpha = 2.75$. The scenario which corresponds to $\alpha = 2.75$ and $\gamma = 6$ is also shown in Figure 4. In the same way we can find the optimal value of α for each fixed $\gamma \in \mathcal{D}$. Since γ is the $(p-k)$th smallest item cost in $X_{\boldsymbol{x}}$ under the computed scenario, not all values of γ in $\mathcal{D}$ need to be examined. In the example we have to only try $\gamma \in \{2,4,5,6\}$.

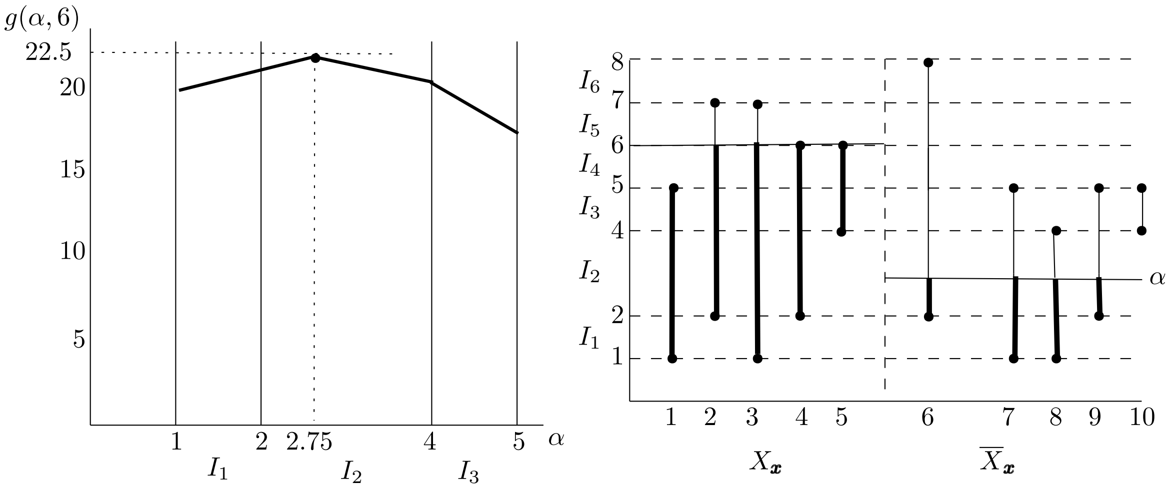

Figure 4: The function $g(\alpha, 6)$ for the sample problem and the worst scenario for the optimal value of $\alpha = 2.75$.

We can then repeat the reasoning for every fixed $\alpha \in \mathcal{D}$. Namely, we solve the problem $\max_{\gamma \geq \alpha} g(\alpha, \gamma)$ by solving the problem for each interval whose lower bound is not less than α. Again, not all values of $\alpha \in \mathcal{D}$ need to be examined. Since α is the kth smallest item cost in $\overline{X}_{\boldsymbol{x}}$, we should check only the values of $\alpha \in \{1, 2, 4, 5\}$.

Theorem 5. *The problem* AREC *under scenario set* $\mathcal{U}^c$ *can be solved in* $O(n^2)$ *time.*

Proof. We will present a sketch of the $O(n^2)$ algorithm. We first determine the family of intervals $I_1, \ldots, I_l$, which requires $O(n \log n)$ time. Now, the key observation is that we can evaluate first all the sums that appear in (15) and (16) for each interval I_j. We can compute and store these sums for every I_j in $O(n^2)$ time. Now each problem $\max_{\alpha \in I_j} g(\alpha, \gamma)$ for $\gamma \in \mathcal{D}$, $\max_{\gamma \in I_j} g(\alpha, \gamma)$ for $\alpha \in \mathcal{D}$, and $\max_{\alpha \in I_j} f(\alpha)$ can be solved in constant time by inserting the computed earlier sums into (16) and (15). The number of problems that must be solved is $O(n^2)$, so the overall running time of the algorithm is $O(n^2)$. □

Using a more refined analysis and data structures such as min-heaps (see, e.g., [11]), this $O(n^2)$ result can be further improved to $O(n \log n)$. The detailed proof can be found in the technical report of this paper (see [9]).

We now show the following proposition, which will be used later:

Proposition 6 (**Dominance rule**). *Let* k, l *be two items such that* $\underline{c}_k \leq \underline{c}_l$ *and* $\overline{c}_k \leq \overline{c}_l$. *Let* $x_l = 1$ *and* $x_k = 0$ *in (11). Then the maximum objective value in (11) will not increase when we change* $x_l = 0$ *and* $x_k = 1$.

Proof. Let $X_{\boldsymbol{x}'} = X_{\boldsymbol{x}} \cup \{k\} \setminus \{l\}$. Notice that (15) does not depend on the first stage solution $\boldsymbol{x}$, so it remains to investigate the effect of replacing $X_{\boldsymbol{x}}$ with $X_{\boldsymbol{x}'}$ in (16). It is enough to show that for each α and $\gamma \geq \alpha$ the following inequalities hold:

$$U_1 = [\gamma - \underline{c}_k]_+ - [\gamma - \underline{c}_l]_+ + [\alpha - \underline{c}_l]_+ - [\alpha - \underline{c}_k]_+ \geq 0 \tag{17}$$

and

$$U_2 = [\gamma - \overline{c}_k]_+ - [\gamma - \overline{c}_l]_+ + [\alpha - \overline{c}_l]_+ - [\alpha - \overline{c}_k]_+ \geq 0. \tag{18}$$

Inequality (17) can be proven by distinguishing the following cases: if $\alpha \leq \gamma \leq \underline{c}_k \leq \underline{c}_l$, then $U_1 = 0$; if $\alpha \leq \underline{c}_k \leq \gamma \leq \underline{c}_l$, then $U_1 = \gamma - \underline{c}_k \geq 0$; if $\alpha \leq \underline{c}_k \leq \underline{c}_l \leq \gamma$, then $U_1 = \underline{c}_l - \underline{c}_k \geq 0$; if $\underline{c}_k \leq \alpha \leq \gamma \leq \underline{c}_l$, then $U_1 = \gamma - \alpha \geq 0$; if $\underline{c}_k \leq \alpha \leq \underline{c}_l \leq \gamma$, then $U_1 = \gamma - \underline{c}_k - \gamma + \underline{c}_l - \alpha + \underline{c}_k = \underline{c}_l - \alpha \geq 0$; if $\underline{c}_k \leq \underline{c}_l \leq \alpha \leq \gamma$, then $U_1 = 0$. The proof of the fact that $U_2 \geq 0$ is just the same. □

2.1.3 The recoverable robust problem

In this section we study RREC (2) under scenario set $\mathcal{U}^c$. We first identify some special cases of this problem, which are known to be polynomially solvable. If Γ is sufficiently large, say $\Gamma \geq \sum_{i\in[n]} d_i$, then scenario set $\mathcal{U}^c$ reduces to $\mathcal{U}^I$ and the problem can be solved in $O((p-k)n^2)$ time [21]. Also the boundary cases $k = 0$ and $k = p$ are polynomially solvable. When $k = p$, then we choose in the first stage p items of the smallest costs under $\boldsymbol{C}$. The total cost of this solution can be then computed in $O(n^2)$ time by solving the corresponding adversarial problem. If $k = 0$, then RREC is equivalent to the MinMax problem with cost intervals $[C_i + \underline{c}_i, C_i + \overline{c}_i]$, $i \in [n]$. Hence it is polynomially solvable due to the results obtained in [3].

Consider now the general case with any $k \in [p]$. We first show a method of preprocessing a given instance of the problem. Given two items $i, j \in [n]$, we write $i \preceq j$ if $C_i \leq C_j$, $\underline{c}_i \leq \underline{c}_j$ and $\overline{c}_i \leq \overline{c}_j$. For any fixed item $l \in [n]$, suppose that $|\{k : k \preceq l\}| \geq p$. Let an optimal solution $\boldsymbol{x} \in \Phi$ to RREC be given, in which $x_l = 1$. There is an item x_k such that $k \preceq l$ and $x_k = 0$ in $\boldsymbol{x}$. We form solution $\boldsymbol{x}'$ by setting $x_k = 1$ and $x_l = 0$. From Proposition 6 and inequality $C_k \leq C_l$, we get

$$\boldsymbol{C}\boldsymbol{x}' + \max_{\boldsymbol{c}\in\mathcal{U}^c} \min_{\boldsymbol{y}\in\Phi^k_{\boldsymbol{x}'}} \boldsymbol{c}\boldsymbol{y} \leq \boldsymbol{C}\boldsymbol{x} + \max_{\boldsymbol{c}\in\mathcal{U}^c} \min_{\boldsymbol{y}\in\Phi^k_{\boldsymbol{x}}} \boldsymbol{c}\boldsymbol{y},$$

and $\boldsymbol{x}'$ is also an optimal solution to RREC. In what follows, we can remove l from $[n]$ without violating the optimum obtaining (after renumbering the items) a smaller item set $[n-1]$. We can now repeat iteratively this reasoning, which allows us to reduce the size of the input instance. Also, for each $l \in [n]$, if $|\{k : l \preceq k\}| \geq n - p$, then we do not violate the optimum after setting $x_l = 1$.

We now reconsider the adversarial problem (11). Its dual is the following:

$$\begin{aligned} \min \; & \sum_{i\in[n]} \underline{c}_i y_i + \Gamma\pi + \sum_{i\in[n]} d_i\rho_i \\ \text{s.t.} \; & \sum_{i\in[n]} y_i = p \end{aligned}$$

$$
\begin{aligned}
&\sum_{i\in[n]} x_i y_i \geq p-k \\
&\pi + \rho_i \geq y_i && i\in[n] \\
&y_i \in [0,1] && i\in[n] \\
&\pi \geq 0 \\
&\rho_i \geq 0 && i\in[n]
\end{aligned}
$$

Using this formulation, we can represent the RREC problem under scenario set $\mathcal{U}^c$ as the following compact mixed-integer program:

$$
\begin{aligned}
\min\quad & \sum_{i\in[n]} C_i x_i + \sum_{i\in[n]} \underline{c}_i y_i + \Gamma\pi + \sum_{i\in[n]} d_i\rho_i \\
\text{s.t.}\quad & \sum_{i\in[n]} y_i = p \\
& \sum_{i\in[n]} x_i = p \\
& \sum_{i\in[n]} x_i y_i \geq p-k \\
& \pi + \rho_i \geq y_i && i\in[n] \\
& x_i \in \{0,1\} && i\in[n] \\
& y_i \in [0,1] && i\in[n] \\
& \pi \geq 0 \\
& \rho_i \geq 0 && i\in[n]
\end{aligned}
\tag{19}
$$

The products $x_i y_i$, $i \in [n]$, can be linearized by using standard methods, which leads to a linear MIP formulation for RREC. Before solving this model the preprocessing described earlier can be applied. We now show that (19) can be solved in polynomial time. Notice that we can assume $\pi \in [0,1]$ and $\rho_i = [y_i - \pi]_+$. Let us split each variable $y_i = \underline{y}_i + \overline{y}_i$, where $\underline{y}_i \in [0,\pi]$ is the cheaper, and $\overline{y}_i \in [0, 1-\pi]$ is the more expensive part of y_i (through the additional costs of ρ_i). The resulting formulation is then

$$
\begin{aligned}
\min\quad & \sum_{i\in[n]} C_i x_i + \sum_{i\in[n]} \underline{c}_i \underline{y}_i + \sum_{i\in[n]} \overline{c}_i \overline{y}_i + \Gamma\pi \\
\text{s.t.}\quad & \sum_{i\in[n]} (\underline{y}_i + \overline{y}_i) = p \\
& \sum_{i\in[n]} x_i = p \\
& \sum_{i\in[n]} x_i(\underline{y}_i + \overline{y}_i) \geq p-k \\
& x_i \in \{0,1\} && i\in[n] \\
& \underline{y}_i \in [0,\pi] && i\in[n] \\
& \overline{y}_i \in [0,1-\pi] && i\in[n] \\
& \pi \in [0,1]
\end{aligned}
\tag{20}
$$

Observe that if $\overline{y}_i > 0$, then $\underline{y}_i = \pi$ for each $i \in [n]$ in some optimal solution, as the whole cheaper part of each item is taken first. Substituting $z_i\pi$ into $\underline{y}_i$ and $\overline{z}_i(1-\pi)$ into $\overline{y}_i$, we can write equivalently

$$
\begin{aligned}
\min \quad & \sum_{i\in[n]} C_i x_i + \sum_{i\in[n]} \pi \underline{c}_i z_i + \sum_{i\in[n]} (1-\pi)\overline{c}_i \overline{z}_i + \Gamma\pi \\
\text{s.t.} \quad & \sum_{i\in[n]} (\pi z_i + (1-\pi)\overline{z}_i) = p \\
& \sum_{i\in[n]} x_i = p \\
& \sum_{i\in[n]} x_i(\pi z_i + (1-\pi)\overline{z}_i) \geq p - k \\
& x_i \in \{0,1\} && i \in [n] \\
& z_i, \overline{z}_i \in [0,1] && i \in [n] \\
& \pi \in [0,1]
\end{aligned}
\tag{21}
$$

Again, in some optimal solution, if $\overline{z}_i > 0$, then $z_i = 1$ (and if $z_i < 1$, then $\overline{z}_i = 0$) for each $i \in [n]$. The following lemma characterizes the optimal solution to (21):

Lemma 7. *There is an optimal solution to* (21) *which satisfies the following properties:*

1. *at most one variable in* $\{z_1, \ldots, z_n, \overline{z}_1, \ldots, \overline{z}_n\}$ *is fractional,*
2. $\pi = \frac{q}{r}$ *for* $r \in [n+1]$, $q \in [n] \cup \{0\}$.

Proof. Let $\boldsymbol{x}^* \in \Phi$ be optimal in (21). Fix $\boldsymbol{x} = \boldsymbol{x}^*$ in (20) and consider the problem with additional slack variables:

$$
\begin{aligned}
\min \quad & \sum_{i\in[n]} \underline{c}_i \underline{y}_i + \sum_{i\in[n]} \overline{c}_i \overline{y}_i + \Gamma\pi \\
\text{s.t.} \quad & \sum_{i\in[n]} (\underline{y}_i + \overline{y}_i) = p \\
& \sum_{i\in X_{\boldsymbol{x}^*}} (\underline{y}_i + \overline{y}_i) - \delta = p - k \\
& \underline{y}_i + \alpha_i = \pi && i \in [n] \\
& \overline{y}_i + \beta_i = 1 - \pi && i \in [n] \\
& \pi + \gamma = 1 \\
& \underline{y}_i, \overline{y}_i, \alpha_i, \beta_i \geq 0 && i \in [n] \\
& \delta, \pi, \gamma \geq 0
\end{aligned}
$$

This problem contains $2n+3$ constraints and $4n+3$ variables. Thus, there is an optimal solution with $2n+3$ basis variables and $2n$ non-basis variables. The following cases are possible.

- First let us assume $\pi = 0$. Then, $y_i = 0$ for all $i \in [n]$ and for the resulting problem

$$\begin{aligned}
\min\ & \sum_{i\in[n]} \overline{c}_i \overline{y}_i \\
\text{s.t.}\ & \sum_{i\in[n]} \overline{y}_i = p \\
& \sum_{i\in X_{\boldsymbol{x}^*}} \overline{y}_i \geq p-k \\
& \overline{y}_i \in [0,1] \qquad\qquad\qquad\qquad i \in [n]
\end{aligned}$$

there exists an optimal integer solution $\overline{\boldsymbol{y}}$ (by first taking the $p-k$ cheapest items from $X_{\boldsymbol{x}^*}$, and completing the solution with the k cheapest items from $[n]$). Since $\pi = 0$, we get $\boldsymbol{z} = \mathbf{1}$ and $\overline{\boldsymbol{z}} = \overline{\boldsymbol{y}}$ and the claim is shown. The proof for $\pi = 1$ is analogous.

- Assume that $0 < \pi < 1$, so both π and γ are basis variables. If y_i is a non-basis variable, then α_i is a basis-variable (and vice versa). The same holds for $\overline{y}_i$ and β_i. The following cases result:

  1. If δ is a basis variable, then it follows that the $2n$ non-basis variables are found in $\boldsymbol{y}$, $\overline{\boldsymbol{y}}$, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Hence, $y_i \in \{0,\pi\}$ and $\overline{y}_i \in \{0, 1-\pi\}$, so $z_i, \overline{z}_i \in \{0,1\}$ for each $i \in [n]$. Let $\mathscr{X} = \sum_{i\in[n]} z_i$ and $\overline{\mathscr{X}} = \sum_{i\in[n]} \overline{z}_i$. We then get $\pi = (p - \overline{\mathscr{X}})/(\mathscr{X} - \overline{\mathscr{X}})$ (see model (21)) and point 2 of the lemma is proven (note that if $\mathscr{X} = \overline{\mathscr{X}}$, we can assume $\pi = 1$).
  2. Let us assume that δ is a non-basis variable. Then one of two cases must hold:

     a) There is j such that both y_j and α_j are basis variables. Then, all other y_i are either 0 or π ($z_i \in \{0,1\}$), and all other $\overline{y}_i$ are either 0 or $1-\pi$ ($\overline{z}_i \in \{0,1\}$). In terms of formulation (21), we have $z_j \in [0,1]$ and only z_j can be fractional (i.e. other than 0 or 1). In order to show the second point of the lemma, let us define $\mathscr{X} = \sum_{i\in[n]\setminus\{j\}} z_i$, $\overline{\mathscr{X}} = \sum_{i\in[n]} \overline{z}_i$, $\mathscr{X}' = \sum_{i\in X_{\boldsymbol{x}^*}\setminus\{j\}} z_i$, and $\overline{\mathscr{X}}' = \sum_{i\in X_{\boldsymbol{x}^*}} \overline{z}_i$. Since $z_i \geq \overline{z}_i$ for each $i \in [n]$ and $\overline{z}_j = 0$, the inequalities $\mathscr{X} \geq \overline{\mathscr{X}}$ and $\mathscr{X}' \geq \overline{\mathscr{X}}'$ hold. We consider now the subproblem of (21) that reoptimizes the solution only in π and z_j:

$$\begin{aligned}
\min\ & \underline{c}_j \pi z_j + t\pi \\
\text{s.t.}\ & \pi z_j = p - \overline{\mathscr{X}} - (\mathscr{X} - \overline{\mathscr{X}})\pi \\
& \pi\mathscr{X}' + (1-\pi)\overline{\mathscr{X}}' + \pi z_j \geq p-k \\
& z_j \in [0,1] \\
& \pi \in [0,1],
\end{aligned}$$

     where $t = \left(\Gamma + \sum_{i\in[n]\setminus\{j\}} \underline{c}_i z_i - \sum_{i\in[n]} \overline{c}_i \overline{z}_i\right)$ is a constant. We remove variable z_j using the equality $z_j = (p - \overline{\mathscr{X}} - (\mathscr{X} - \overline{\mathscr{X}})\pi)/\pi$ from the problem.

The constraint $z_j \geq 0$ becomes $\pi \leq (p-\overline{\mathfrak{X}})/(\mathfrak{X}-\overline{\mathfrak{X}})$, while the constraint $z_j \leq 1$ is $\pi \geq (p-\overline{\mathfrak{X}})/(\mathfrak{X}-\overline{\mathfrak{X}}+1)$. Hence, the reoptimization problem becomes

$$\begin{aligned} \min\ & t\pi + \underline{c}_j(p - \overline{\mathfrak{X}} - (\mathfrak{X} - \overline{\mathfrak{X}})\pi) \\ \text{s.t. }\ & \pi((\mathfrak{X}' - \overline{\mathfrak{X}}') - (\mathfrak{X} - \overline{\mathfrak{X}})) \geq \overline{\mathfrak{X}} - \overline{\mathfrak{X}}' - k \\ & \frac{p - \overline{\mathfrak{X}}}{1 + \mathfrak{X} - \overline{\mathfrak{X}}} \leq \pi \leq \frac{p - \overline{\mathfrak{X}}}{\mathfrak{X} - \overline{\mathfrak{X}}} \\ & \pi \in [0,1] \end{aligned}$$

We can conclude that the optimal value of π is one of $\frac{p-\overline{\mathfrak{X}}}{\mathfrak{X}-\overline{\mathfrak{X}}}$, $\frac{p-\overline{\mathfrak{X}}}{\mathfrak{X}-\overline{\mathfrak{X}}+1}$, or $\frac{\overline{\mathfrak{X}}-\overline{\mathfrak{X}}'-k}{(\mathfrak{X}'-\overline{\mathfrak{X}}')-(\mathfrak{X}-\overline{\mathfrak{X}})}$. Since $\overline{\mathfrak{X}}, \overline{\mathfrak{X}}', \mathfrak{X}, \mathfrak{X}'$ are all integers from 0 to n, the second point of the lemma is true.

b) There is j such that both $\overline{y}_j$ and β_j are basis variables. This case is analogue to the previous case.

□

Lemma 8. *Problem* RREC *under* $\mathcal{U}^c$ *can be solved by solving the problem:*

$$\begin{aligned} \min\quad & \sum_{i\in\mathcal{J}} C_i x_i + \sum_{i\in\mathcal{J}} \pi \underline{c}_i z_i + \sum_{i\in\mathcal{J}} (1-\pi)\overline{c}_i \overline{z}_i + \Gamma\pi \\ s.t.\quad & \sum_{i\in\mathcal{J}} x_i = p \\ & \sum_{i\in\mathcal{J}} z_i = \mathfrak{X} \\ & \sum_{i\in\mathcal{J}} \overline{z}_i = \overline{\mathfrak{X}} \\ & \sum_{i\in\mathcal{J}} z'_i \geq \mathfrak{X}' \\ & \sum_{i\in\mathcal{J}} \overline{z}'_i \geq \overline{\mathfrak{X}}' \\ & z'_i \leq x_i & i \in \mathcal{J} \\ & z'_i \leq z_i & i \in \mathcal{J} \\ & \overline{z}'_i \leq x_i & i \in \mathcal{J} \\ & \overline{z}'_i \leq \overline{z}_i & i \in \mathcal{J} \\ & x_i \in \{0,1\} & i \in \mathcal{J} \\ & z_i, \overline{z}_i, z'_i, \overline{z}'_i \in \{0,1\} & i \in \mathcal{J} \end{aligned} \tag{22}$$

for polynomially many sets $\mathcal{J}$ *and values of* $\mathfrak{X}$, $\overline{\mathfrak{X}}$, $\mathfrak{X}'$, $\overline{\mathfrak{X}}'$ *and* π.

Proof. Using Lemma 7, we first consider the case when $z_i, \overline{z}_i \in \{0,1\}$ for all $i \in [n]$. Then we set $\mathcal{J} = [n]$ and guess the values $\mathfrak{X} = \sum_{i\in\mathcal{J}} z_i$ and $\overline{\mathfrak{X}} = \sum_{i\in\mathcal{J}} \overline{z}_i$. We set $\pi = (p-\overline{\mathfrak{X}})/(\mathfrak{X}-\overline{\mathfrak{X}})$, and further guess all possible values of $\mathfrak{X} = \sum_{i\in\mathcal{J}} x_i z_i$ and

$\overline{\mathcal{X}} = \sum_{i\in\mathcal{J}} x_i\overline{z}_i$ for which the constraint $\pi\mathcal{X} + (1-\pi)\overline{\mathcal{X}} \geq p-k$ is fulfilled. In total, we have to try polynomially many values. For each resulting problem we linearize $z_i x_i$ and $\overline{z}_i x_i$ and we get (22).

Assume now that some $z_j \in [0,1]$ can be fractional (notice that in this case we can fix $\overline{z}_j = 0$). We guess the index j, the value of π, and the value of x_j. We fix then $\mathcal{J} = [n]\setminus\{j\}$ and continue as in the first part of the proof. Namely we guess $\mathfrak{X}$, $\overline{\mathfrak{X}}$, and $\mathcal{X}$, $\overline{\mathcal{X}}$ for the fixed π, and construct the problem (22). Notice that the value of z_j can be retrieved from $\pi z_j = (p - \overline{\mathfrak{X}} - (\mathfrak{X} - \overline{\mathfrak{X}})\pi)$. The case where $\overline{z}_j \in [0,1]$ can be fractional is analogue. Again, we have to try polynomially many values. To solve problem (21), we then take the best of all solutions. □

Lemma 9. *For fixed $\mathcal{J}$, $\mathfrak{X}$, $\overline{\mathfrak{X}}$, $\mathfrak{X}'$, $\overline{\mathfrak{X}}'$ and π, the problem* (22) *can be solved in polynomial time.*

Proof. We will show that the coefficient matrix of the relaxation of (22) is totally unimodular. We will use the following Ghouila-Houri criterion [7]: an $m\times n$ integral matrix is totally unimodular, if and only if for each set of rows $R = \{r_1,\ldots,r_K\} \subseteq [m]$ there exists a coloring (called a valid coloring) $l(r_i) \in \{-1,1\}$ such that the weighted sum of every column restricted to R is -1, 0, or 1. For simplicity, we assume w.l.o.g. that $\mathcal{J} = [n]$. Note that it is enough to show that the coefficient matrix of (22) without the relaxed constraints $x_i, z_i, \overline{z}_i, z_i', \overline{z}_i' \leq 1$ is totally unimodular. The matrix, together with a labeling of its rows, is shown in Table 3.

Table 3: The coefficient matrix of (22)

	x_1	x_2	$\cdots$	x_n	z_1	z_2	$\cdots$	z_n	$\overline{z}_1$	$\overline{z}_2$	$\cdots$	$\overline{z}_n$	z_1'	z_2'	$\cdots$	z_n'	$\overline{z}_1'$	$\overline{z}_2'$	$\cdots$	$\overline{z}_n'$
a_1	1	1	$\cdots$	1	0	0	$\cdots$	0	0	0	$\cdots$	0	0	0	$\cdots$	0	0	0	$\cdots$	0
a_2	0	0	$\cdots$	0	1	1	$\cdots$	1	0	0	$\cdots$	0	0	0	$\cdots$	0	0	0	$\cdots$	0
a_3	0	0	$\cdots$	0	0	0	$\cdots$	0	1	1	$\cdots$	1	0	0	$\cdots$	0	0	0	$\cdots$	0
a_4	0	0	$\cdots$	0	0	0	$\cdots$	0	0	0	$\cdots$	0	1	1	$\cdots$	1	0	0	$\cdots$	0
a_5	0	0	$\cdots$	0	0	0	$\cdots$	0	0	0	$\cdots$	0	0	0	$\cdots$	0	1	1	$\cdots$	1
b_1	1	0	$\cdots$	0	0	0	$\cdots$	0	0	0	$\cdots$	0	-1	0	$\cdots$	0	0	0	$\cdots$	0
b_2	0	1	$\cdots$	0	0	0	$\cdots$	0	0	0	$\cdots$	0	0	-1	$\cdots$	0	0	0	$\cdots$	0
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$
b_n	0	0	$\cdots$	1	0	0	$\cdots$	0	0	0	$\cdots$	0	0	0	$\cdots$	-1	0	0	$\cdots$	0
c_1	0	0	$\cdots$	0	1	0	$\cdots$	0	0	0	$\cdots$	0	-1	0	$\cdots$	0	0	0	$\cdots$	0
c_2	0	0	$\cdots$	0	0	1	$\cdots$	0	0	0	$\cdots$	0	0	-1	$\cdots$	0	0	0	$\cdots$	0
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$
c_n	0	0	$\cdots$	0	0	0	$\cdots$	1	0	0	$\cdots$	0	0	0	$\cdots$	-1	0	0	$\cdots$	0
d_1	1	0	$\cdots$	0	0	0	$\cdots$	0	0	0	$\cdots$	0	0	0	$\cdots$	0	-1	0	$\cdots$	0
d_2	0	1	$\cdots$	0	0	0	$\cdots$	0	0	0	$\cdots$	0	0	0	$\cdots$	0	0	-1	$\cdots$	0
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$
d_n	0	0	$\cdots$	1	0	0	$\cdots$	0	0	0	$\cdots$	0	0	0	$\cdots$	0	0	0	$\cdots$	-1
e_1	0	0	$\cdots$	0	0	0	$\cdots$	0	1	0	$\cdots$	0	0	0	$\cdots$	0	-1	0	$\cdots$	0
e_2	0	0	$\cdots$	0	0	0	$\cdots$	0	0	1	$\cdots$	0	0	0	$\cdots$	0	0	-1	$\cdots$	0
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$
e_n	0	0	$\cdots$	0	0	0	$\cdots$	0	0	0	$\cdots$	1	0	0	$\cdots$	0	0	0	$\cdots$	-1

Given a set of rows R we use the following algorithm to color the rows in R:

1. $l(d_i) = 1$ for each $d_i \in R$
2. If $a_5 \in R$, set $l(a_5) = 1$, $l(e_i) = 1$ for each $e_i \in R$ and $l(a_3) = -1$.
3. If $a_5 \notin R$, set $l(e_i) = -1$ for each $e_i \in R$ and $l(a_3) = 1$.
4. If $a_1 \in R$, set $l(a_1) = -1$ and $l(b_i) = 1$ for each $b_i \in R$.
   a) If $a_4 \in R$, set $l(a_4) = 1$, $l(c_i) = 1$ for each $c_i \in R$ and $l(a_2) = -1$.
   b) If $a_4 \notin R$, set $l(c_i) = -1$ for each $c_i \in R$ and $l(a_2) = 1$.
5. If $a_1 \notin R$, set $l(b_i) = -1$ for each $b_i \in R$.
   a) If $a_4 \in R$, set $l(a_4) = -1$, $l(c_i) = -1$ for each $c_i \in R$ and $l(a_2) = 1$.
   b) If $a_4 \notin R$, set $l(c_i) = 1$ for each $c_i \in R$ and $l(a_2) = -1$.

If $a_1 \in R$, then $l(a_1) = -1$ and the rows $b_i, d_i, \in R$ have always color 1; if $a_1 \notin R$, then the rows $b_i \in R$ have color -1 and the rows $d_i \in R$ have color 1. So the coloring is valid for all columns corresponding to x_i. In order to prove that the coloring is valid for the columns corresponding to $z_1, \ldots, z_n$, $\overline{z}_1, \ldots, \overline{z}_n$ it is enough to observe that the algorithm always assigns different colors to a_2 and the rows $c_i \in R$, and a_3 and the rows in $e_i \in R$. If $a_4 \in R$, then a_4 has the same color as all $b_i \in R$ or all $c_i \in R$; if $a_4 \notin R$, then $b_i \in R$ and $c_i \in R$ have different color. In consequence, the coloring is valid for the columns corresponding to $z'_1, \ldots, z'_n$. It is also easy to see that the coloring is valid for the columns corresponding to variables $\overline{z}'_1, \ldots, \overline{z}'_n$ (see steps 1-3).

□

Theorem 10. *The RREC problem under scenario set $\mathcal{U}^c$ is solvable in polynomial time.*

Proof. This result is a direct consequence of Lemma 8 and Lemma 9. □

2.2 Two-Stage Robust Selection

In this section we investigate the two-stage model, namely the I2ST (7), A2ST (5) and R2ST (3) problems under scenario set $\mathcal{U}^c$. In order to solve A2ST we will use the results obtained for AREC. We will also show that R2ST is polynomially solvable as it can be reduced to solving a polynomial number of linear programming problems.

2.2.1 The incremental and adversarial problems

We are given a first stage solution $\boldsymbol{x} \in \Phi_1$ with $|X_{\boldsymbol{x}}| = p_1$, where $p_1 \in [p] \cup \{0\}$. Define $\tilde{p} = p - p_1$. The incremental problem, I2ST, can be solved in $O(n)$ time. It is enough to choose $\tilde{p}$ items of the smallest costs out of $\overline{X_{\boldsymbol{x}}}$ under the given scenario $\boldsymbol{c}$. On the other hand, the adversarial problem, A2ST, can be reduced to AREC. We first remove from $[n]$ all the items belonging $X_{\boldsymbol{x}}$, obtaining (after an appropriate renumbering) the item set $[n-p_1]$. We then fix $k=\tilde{p}$. As we can exchange all the items, the choice of the first stage solution in AREC is irrelevant. Consider the formula (16) for the constructed instance of AREC. The optimal value of γ satisfies $\gamma=\alpha$ and (16) becomes:

$$\max_{\alpha}\left\{\alpha\tilde{p}-\max\left\{\sum_{i\in[\tilde{p}]}[\alpha-\underline{c}_i]_+-\Gamma,\sum_{i\in[\tilde{p}]}[\alpha-\overline{c}_i]_+\right\}\right\}, \tag{23}$$

which is in turn the same as (15). Problem (23) can be solved in $O(n^2)$ time, by the method described in Section 2.1.2. This means that A2ST is solvable in $O(n^2)$ time.

2.2.2 The two-stage robust problem

Given $\boldsymbol{x}\in\Phi_1$ and $\boldsymbol{c}\in\mathcal{U}$, the incremental problem, I2ST, can be formulated as the following linear program (the constraints $y_i\in\{0,1\}$ can be relaxed):

$$\begin{aligned} \min\ & \sum_{i\in[n]} c_i y_i \\ \text{s.t. } & \sum_{i\in[n]}(y_i+x_i)=p \\ & y_i\le 1-x_i && i\in[n] \\ & 0\le y_i\le 1 && i\in[n] \end{aligned} \tag{24}$$

Using the dual to (24), we can find a compact formulation for the adversarial problem, A2ST, under scenario set $\mathcal{U}^c$:

$$\begin{aligned} \max\ & (p-\sum_{i\in[n]}x_i)\alpha-\sum_{i\in[n]}(1-x_i)\gamma_i \\ \text{s.t. } & \alpha\le\delta_i+\gamma_i+\underline{c}_i && i\in[n] \\ & \delta_i\le d_i && i\in[n] \\ & \sum_{i\in[n]}\delta_i\le\Gamma \\ & \gamma_i,\delta_i\ge 0 && i\in[n] \end{aligned} \tag{25}$$

Dualizing (25), we get the following problem:

$$\begin{aligned} \min\ & \sum_{i\in[n]}\underline{c}_i y_i+\Gamma\pi+\sum_{i\in[n]}d_i\rho_i \\ \text{s.t. } & \sum_{i\in[n]}(y_i+x_i)=p \\ & \pi+\rho_i\ge y_i && i\in[n] \\ & y_i\le 1-x_i && i\in[n] \\ & \pi\ge 0 \\ & \rho_i\ge 0 && i\in[n] \end{aligned}$$

$$y_i \in [0,1] \qquad i \in [n]$$

which can be used to construct the following mixed-integer program for R2ST:

$$\begin{aligned}
\min \quad & \sum_{i\in[n]} C_i x_i + \sum_{i\in[n]} \underline{c}_i y_i + \Gamma\pi + \sum_{i\in[n]} d_i \rho_i \\
\text{s.t.} \quad & \sum_{i\in[n]} (y_i + x_i) = p \\
& x_i + y_i \leq 1 && i \in [n] \\
& \pi + \rho_i \geq y_i && i \in [n] \\
& \pi \geq 0 \\
& \rho_i \geq 0 && i \in [n] \\
& x_i \in \{0,1\} && i \in [n] \\
& y_i \in [0,1] && i \in [n]
\end{aligned} \tag{26}$$

We now show that (26) can be solved in polynomial time. We first apply to (26) similar transformation as for the RREC problem (see Section 2.1.3), which results in the following equivalent formulation:

$$\begin{aligned}
\min \quad & \sum_{i\in[n]} C_i x_i + \sum_{i\in[n]} \underline{c}_i \underline{y}_i + \sum_{i\in[n]} \overline{c}_i \overline{y}_i + \Gamma\pi \\
\text{s.t.} \quad & \sum_{i\in[n]} (x_i + \underline{y}_i + \overline{y}_i) = p \\
& x_i + \underline{y}_i \leq 1 && i \in [n] \\
& x_i + \overline{y}_i \leq 1 && i \in [n] \\
& x_i \in \{0,1\} && i \in [n] \\
& \underline{y}_i \in [0,\pi] && i \in [n] \\
& \overline{y}_i \in [0,1-\pi] && i \in [n] \\
& \pi \in [0,1]
\end{aligned} \tag{27}$$

Again, by setting $z_i\pi = \underline{y}_i$ and $\overline{z}_i(1-\pi) = \overline{y}_i$, we rescale the variables and find the following equivalent, nonlinear problem:

$$\begin{aligned}
\min \quad & \sum_{i\in[n]} C_i x_i + \sum_{i\in[n]} \pi \underline{c}_i z_i + \sum_{i\in[n]} (1-\pi)\overline{c}_i \overline{z}_i + \Gamma\pi \\
\text{s.t.} \quad & \sum_{i\in[n]} (x_i + \pi z_i + (1-\pi)\overline{z}_i) = p \\
& x_i + z_i \leq 1 && i \in [n] \\
& x_i + \overline{z}_i \leq 1 && i \in [n] \\
& x_i \in \{0,1\} && i \in [n] \\
& z_i, \overline{z}_i \in [0,1] && i \in [n] \\
& \pi \in [0,1]
\end{aligned} \tag{28}$$

Note that we can write $x_i + z_i \leq 1$ instead of $x_i + \pi z_i \leq 1$ and $x_i + \overline{z}_i \leq 1$ instead of $x_i + (1-\pi)\overline{z}_i \leq 1$ for each $i \in [n]$.

Lemma 11. *There exists an optimal solution to* (28) *in which*

1. $z_i, \overline{z}_i \in \{0,1\}$ *for all* $i \in [n]$,
2. $\pi = \frac{q}{r}$, *where* $q \in [p] \cup \{0\}$, $r \in [n]$.

Proof. We first prove point 1. Let $\boldsymbol{x}^* \in \Phi_1$ be optimal in (28). Using formulation (27), we consider the following linear program for fixed $\boldsymbol{x} = \boldsymbol{x}^*$ and additional slack variables:

$$
\begin{aligned}
\min \ & \sum_{i \in \overline{X}_{\boldsymbol{x}^*}} \underline{c}_i \underline{y}_i + \sum_{i \in \overline{X}_{\boldsymbol{x}^*}} \overline{c}_i \overline{y}_i + \Gamma\pi \\
\text{s.t. } & \sum_{i \in \overline{X}^*} (\underline{y}_i + \overline{y}_i) = p - \sum_{i \in [n]} x_i^* \\
& \underline{y}_i + \alpha_i = \pi && i \in \overline{X}_{\boldsymbol{x}^*} \\
& \overline{y}_i + \beta_i = 1 - \pi && i \in \overline{X}_{\boldsymbol{x}^*} \\
& \pi + \gamma = 1 \\
& \underline{y}_i, \overline{y}_i, \alpha_i, \beta_i \geq 0 && i \in \overline{X}_{\boldsymbol{x}^*} \\
& \pi, \gamma \geq 0
\end{aligned}
$$

This problem has $4|\overline{X}_{\boldsymbol{x}*}| + 2$ variables and $2|\overline{X}_{\boldsymbol{x}^*}| + 2$ constraints. Thus, there is an optimal solution with $2|\overline{X}_{\boldsymbol{x}^*}| + 2$ basis variables and $2|\overline{X}_{\boldsymbol{x}^*}|$ non-basis variables. If $\pi = 0$, then $\underline{y}_i = 0$ for all $i \in \overline{X}_{\boldsymbol{x}^*}$ and the problem becomes a selection problem in $\overline{\boldsymbol{y}}$, for which there is an optimal integer solution. If $\pi = 1$, then $\overline{y}_i = 0$ for each $i \in [n]$ and the problem becomes a selection problem in $\underline{\boldsymbol{y}}$. Hence, in both these cases, there exists an optimal solution to (28) that is integer in $\boldsymbol{z}$ and $\overline{\boldsymbol{z}}$. So let us assume that $\pi > 0$ and $\pi < 1$, i.e., both π and γ are basis variables. Note that whenever $\underline{y}_i$ (resp. $\overline{y}_i$) is a non-basis variable, then α_i (resp. β_i) is a basis variable, and vice versa. Hence, all variables $\underline{y}_i$ are either 0 or π, and all variables $\overline{y}_i$ are either 0 or $1-\pi$. This corresponds to a solution where all z_i and $\overline{z}_i$ are either 0 or 1 in formulation (28).

We now prove point 2. Let $x_i^*, z_i^*, \overline{z}_i^* \in \{0,1\}$, $i \in [n]$, be optimal in (28). If $\sum_{i \in [n]} z_i^* = \sum_{i \in [n]} \overline{z}_i^*$, then there exists an optimal solution with $\pi^* \in \{0,1\}$. So let us assume $\sum_{i \in [n]} z_i^* > \sum_{i \in [n]} \overline{z}_i^*$ (recall that $z_i^* \geq \overline{z}_i^*$ for each $i \in [n]$). By rearranging terms, we obtain

$$
\pi = \frac{p - \sum_{i \in [n]} (x_i^* + \overline{z}_i^*)}{\sum_{i \in [n]} (z_i^* - \overline{z}_i^*)}.
$$

Write $\mathcal{X} = \sum_{i \in [n]} x_i^*$, $\mathcal{Z} = \sum_{i \in [n]} z_i^*$ and $\overline{\mathcal{Z}} = \sum_{i \in [n]} \overline{z}_i^*$. We have $\mathcal{X} \in \{0, \ldots, p\}$, $\mathcal{Z}, \overline{\mathcal{Z}} \in \{0, \ldots, n\}$. Note that if for an item i we have $\overline{z}_i^* = 1$, then also $z_i^* = 1$. Consequently, π is of the form described in point 2 of the lemma. □

Lemma 12. *The* R2ST *problem under* $\mathcal{U}^c$ *can be solved by solving problem*

$$\begin{aligned}
\min \quad & \sum_{i\in[n]} C_i x_i + \sum_{i\in[n]} \pi \underline{c}_i z_i + \sum_{i\in[n]} (1-\pi)\overline{c}_i \overline{z}_i + \Gamma\pi \\
s.t. \quad & \sum_{i\in[n]} x_i = \mathfrak{X} \\
& \sum_{i\in[n]} z_i = \mathfrak{Z} \\
& \sum_{i\in[n]} \overline{z}_i = \overline{\mathfrak{Z}} \\
& x_i + z_i \le 1 && i \in [n] \\
& x_i + \overline{z}_i \le 1 && i \in [n] \\
& x_i, z_i, \overline{z}_i \in \{0,1\} && i \in [n]
\end{aligned} \tag{29}$$

for polynomially many values of $\mathfrak{X}, \mathfrak{Z}, \overline{\mathfrak{Z}}$ *and* π.

Proof. Using Lemma 11, we will try all possible values of π and for each fixed π we will find an optimal solution to (28) where all $\boldsymbol{x}$, $\boldsymbol{z}$ and $\overline{\boldsymbol{z}}$ are integer. Let the value $\pi = \pi^*$ be fixed. The resulting problem is then

$$\begin{aligned}
\min \quad & \sum_{i\in[n]} C_i x_i + \sum_{i\in[n]} \pi^* \underline{c}_i z_i + \sum_{i\in[n]} (1-\pi^*)\overline{c}_i \overline{z}_i + \Gamma\pi^* \\
\text{s.t.} \quad & \sum_{i\in[n]} x_i + \pi^* \sum_{i\in[n]} z_i + (1-\pi^*) \sum_{i\in[n]} \overline{z}_i = p \\
& x_i + z_i \le 1 && i \in [n] \\
& x_i + \overline{z}_i \le 1 && i \in [n] \\
& x_i, z_i, \overline{z}_i \in \{0,1\} && i \in [n]
\end{aligned}$$

As π^* is fixed, we can enumerate all possible values of $\mathfrak{X} = \sum_{i\in[n]} x_i$, $\mathfrak{Z} = \sum_{i\in[n]} z_i$ and $\overline{\mathfrak{Z}} = \sum_{i\in[n]} \overline{z}_i$ that generate this value π^*, i.e., we enumerate all possible solutions to $\mathfrak{X} + \pi^*\mathfrak{Z} + (1-\pi^*)\overline{\mathfrak{Z}} = p$. There can be at most p choices of $\mathfrak{X}$ and at most n choices of $\mathfrak{Z}$ and $\overline{\mathfrak{Z}}$. This leads to the problem (29). By choosing the best of the computed solutions, we then find an optimal solution to R2ST. □

Lemma 13. *For fixed* $\mathfrak{X}, \mathfrak{Z}, \overline{\mathfrak{Z}}$ *and* π*, the problem* (29) *can be solved in polynomial time.*

Proof. We prove that the coefficient matrix of the relaxation of (29) is totally unimodular. We will use the Ghouila-Houri criterion [7] (see the proof of Lemma 9). The coefficient matrix of the constraints of (29) is shown in Table 4 (we can skip the relaxed constraints $x_i, z_i, \overline{z}_i \le 1$).

Let us choose a subset of the rows $R = A \cup B \cup C$ with $A \subseteq \{a_1, a_2, a_3\}$, $B \subseteq \{b_1, \ldots, b_n\}$ and $C \subseteq \{c_1, \ldots, c_n\}$. We now determine the coloring for R in the following way:

- If $A = \emptyset$, then $l(b_i) = -1$, $l(c_i) = 1$.

Table 4: coefficient matrix of problem (29).

	x_1	x_2	x_3	$\cdots$	x_n	z_1	z_2	z_3	$\cdots$	z_n	$\overline{z}_1$	$\overline{z}_2$	$\overline{z}_3$	$\cdots$	$\overline{z}_n$
a_1	1	1	1	$\cdots$	1	0	0	0	$\cdots$	0	0	0	0	$\cdots$	0
a_2	0	0	0	$\cdots$	0	1	1	1	$\cdots$	1	0	0	0	$\cdots$	0
a_3	0	0	0	$\cdots$	0	0	0	0	$\cdots$	0	1	1	1	$\cdots$	1
b_1	1	0	0	$\cdots$	0	1	0	0	$\cdots$	0	0	0	0	$\cdots$	0
b_2	0	1	0	$\cdots$	0	0	1	0	$\cdots$	0	0	0	0	$\cdots$	0
b_3	0	0	1	$\cdots$	0	0	0	1	$\cdots$	0	0	0	0	$\cdots$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
b_n	0	0	0	$\cdots$	1	0	0	0	$\cdots$	1	0	0	0	$\cdots$	0
c_1	1	0	0	$\cdots$	0	0	0	0	$\cdots$	0	1	0	0	$\cdots$	0
c_2	0	1	0	$\cdots$	0	0	0	0	$\cdots$	0	0	1	0	$\cdots$	0
c_3	0	0	1	$\cdots$	0	0	0	0	$\cdots$	0	0	0	1	$\cdots$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
c_n	0	0	0	$\cdots$	1	0	0	0	$\cdots$	0	0	0	0	$\cdots$	1

- If $A = \{a_1\}$, then $l(a_1) = -1$, $l(b_i) = l(c_i) = 1$.
- If $A = \{a_2\}$, then $l(a_2) = -1$, $l(b_i) = 1$, $l(c_i) = -1$.
- If $A = \{a_1, a_2\}$, then $l(a_1) = l(a_2) = -1$, $l(b_i) = l(c_i) = 1$.
- If $A = \{a_3\}$, then $l(a_3) = -1$, $l(b_i) = -1$, $l(c_i) = 1$.
- If $A = \{a_1, a_3\}$, then $l(a_1) = l(a_3) = -1$, $l(b_i) = l(c_i) = 1$.
- If $A = \{a_2, a_3\}$, then $l(a_2) = -1$, $l(a_3) = 1$, $l(b_i) = 1$, $l(c_i) = -1$.
- If $A = \{a_1, a_2, a_3\}$, then $l(a_i) = -1$, $l(b_i) = l(c_i) = 1$.

It is easy to verify that the coloring is valid for each of these cases. □

Theorem 14. *The* R2ST *problem under scenario set* $\mathcal{U}^c$ *is solvable in polynomial time.*

Proof. A direct consequence of Lemma 12 and Lemma 13. □

3 Discrete Budgeted Uncertainty

In this section we consider the RREC (2) and R2ST (3) problems under scenario set $\mathcal{U}^d$. We will use some results obtained for the continuous budgeted uncertainty (in particular Proposition 1). Notice also that the incremental problems IREC and I2ST are the same as for the continuous case.

3.1 Recoverable Robust Selection

3.1.1 The adversarial problem

Let us fix solution $\boldsymbol{x} \in \Phi$. The adversarial problem, AREC (4), under scenario set $\mathcal{U}^d$, can be represented as the following mathematical programming problem:

$$
\begin{aligned}
\max_{\substack{\boldsymbol{\delta}\in\{0,1\}^n \\ \sum_{i\in[n]}\delta_i\le\Gamma}} \quad \min_{\boldsymbol{y}} \quad & \sum_{i\in[n]} (\underline{c}_i + d_i\delta_i) y_i \\
\text{s.t.} \quad & \sum_{i\in[n]} y_i = p \\
& \sum_{i\in[n]} x_i y_i \ge p - k \\
& y_i \in \{0,1\} \qquad i \in [n]
\end{aligned} \tag{30}
$$

Relaxing the integrality constraints $y_i \in \{0,1\}$ in (30) for the inner incremental problem, and taking the dual of it, we obtain the following integer linear program for AREC:

$$
\begin{aligned}
\max \quad & p\alpha + (p-k)\beta - \sum_{i\in[n]} \gamma_i \\
\text{s.t.} \quad & \alpha + x_i\beta - \gamma_i \le \underline{c}_i + d_i\delta_i \quad i \in [n] \\
& \sum_{i\in[n]} \delta_i \le \Gamma \\
& \beta \ge 0 \\
& \gamma_i \ge 0 \qquad i \in [n] \\
& \delta_i \in \{0,1\} \qquad i \in [n]
\end{aligned} \tag{31}
$$

Let $\boldsymbol{\delta}^* \in \{0,1\}^n$ be optimal in (31). The vector $\boldsymbol{\delta}^*$ describes the worst scenario $\hat{\boldsymbol{c}} = (\hat{c}_i)_{i\in[n]} = (\underline{c}_i + d_i\delta_i^*)_{i\in[n]} \in \mathcal{U}^d$. When we fix this scenario in (31), then we get the problem (9), discussed in Section 2.1, to which Proposition 1 can be applied. Since $\hat{c}_i$ is either $\underline{c}_i$ or $\overline{c}_i$ for each $i \in [n]$, only a finite number of values of α and β need to be considered as optimal to (31) (see Proposition 1). In the following, we will show how to find these values efficiently. We can fix $\gamma_i = [\alpha + x_i\beta - d_i\delta_i - \underline{c}_i]_+$ for each $i \in [n]$ in (31). Hence, $\gamma_i = [\alpha + x_i\beta - d_i - \underline{c}_i]_+$ if $\delta_i = 1$, and $\gamma_i = [\alpha + x_i\beta - \underline{c}_i]_+$ if $\delta_i = 0$. In consequence, (31) can be rewritten as follows:

$$
\begin{aligned}
\max \quad & p\alpha + (p-k)\beta - \sum_{i\in[n]} [\alpha + x_i\beta - \underline{c}_i]_+ \\
& + \sum_{i\in[n]} ([\alpha + x_i\beta - \underline{c}_i]_+ - [\alpha + x_i\beta - d_i - \underline{c}_i]_+)\delta_i \\
\text{s.t.} \quad & \sum_{i\in[n]} \delta_i \le \Gamma \\
& \delta_i \in \{0,1\} \qquad i \in [n] \\
& \beta \ge 0
\end{aligned} \tag{32}
$$

It is easily seen that for fixed α, β, and $\boldsymbol{x}$, (32) is the Selection problem, which can be solved in $O(n)$ time. We now find the sets of relevant values of α and β. Let us order the elements in $[n]$ according to their costs $\hat{c}_i$ and the cost bounds $\underline{c}_i$ and $\overline{c}_i$ for $i \in [n]$ in the following way:

$$\hat{c}_{\sigma(1)} \leq \cdots \leq \hat{c}_{\sigma(n)},\ \underline{c}_{\underline{\sigma}(1)} \leq \cdots \leq \underline{c}_{\underline{\sigma}(n)},\ \overline{c}_{\overline{\sigma}(1)} \leq \cdots \leq \overline{c}_{\overline{\sigma}(n)}.$$

Similarly, let us order the elements in $X_{\boldsymbol{x}}$ so that

$$\hat{c}_{\nu(1)} \leq \cdots \leq \hat{c}_{\nu(p)},\ \underline{c}_{\underline{\nu}(1)} \leq \cdots \leq \underline{c}_{\underline{\nu}(p)},\ \overline{c}_{\overline{\nu}(1)} \leq \cdots \leq \overline{c}_{\overline{\nu}(p)}$$

and in $\overline{X}_{\boldsymbol{x}}$, namely

$$\hat{c}_{\varsigma(1)} \leq \cdots \leq \hat{c}_{\varsigma(n-p)},\ \underline{c}_{\underline{\varsigma}(1)} \leq \cdots \leq \underline{c}_{\underline{\varsigma}(n-p)},\ \overline{c}_{\overline{\varsigma}(1)} \leq \cdots \leq \overline{c}_{\overline{\varsigma}(n-p)}.$$

According to Proposition 1, $\alpha = \hat{c}_{\sigma(p)}$ and $\beta = 0$, or $\alpha = \hat{c}_{\varsigma(k)}$, and $\beta = \hat{c}_{\nu(p-k)} - \hat{c}_{\varsigma(k)}$ are optimal in (32). Thus we have

$$\begin{aligned}
&\hat{c}_{\sigma(p)} \in \mathcal{C}_{\sigma(p)} = \{\underline{c}_{\underline{\sigma}(p)}, \ldots, \underline{c}_{\underline{\sigma}(p+\Gamma)}\} \cup \{\overline{c}_{\overline{\sigma}(1)}, \ldots, \overline{c}_{\overline{\sigma}(p)}\},\\
&\hat{c}_{\nu(p-k)} \in \mathcal{C}_{\nu(p-k)} = \{\underline{c}_{\underline{\nu}(p-k)}, \ldots, \underline{c}_{\underline{\nu}(p-k+\Gamma)}\} \cup \{\overline{c}_{\overline{\nu}(1)}, \ldots, \overline{c}_{\overline{\nu}(p-k)}\},\\
&\hat{c}_{\varsigma(k)} \in \mathcal{C}_{\varsigma(k)} = \{\underline{c}_{\underline{\varsigma}(k)}, \ldots, \underline{c}_{\underline{\varsigma}(k+\Gamma)}\} \cup \{\overline{c}_{\overline{\varsigma}(1)}, \ldots, \overline{c}_{\overline{\varsigma}(k)}\}.
\end{aligned}$$

For simplicity of notation, we write $\underline{c}_{\underline{\sigma}(n)}$ instead of $\underline{c}_{\underline{\sigma}(p+\Gamma)}$, when $p+\Gamma > n$. The same holds for $\underline{c}_{\underline{\nu}(p-k+\Gamma)}$ and $\underline{c}_{\underline{\varsigma}(k+\Gamma)}$. Observe that we can assume that $\Gamma \leq n/2$. Indeed, if $\Gamma > n/2$, then it suffices to substitute variables z_i by $1-w_i$, $w_i \in \{0,1\}$, $i \in [n]$. Now the constraint $\sum_{i\in[n]} \delta_i \leq \Gamma$ and the costs $\hat{c}_i$ become $\sum_{i\in[n]} w_i \geq n-\Gamma$ and $\hat{c}_i = \underline{c}_i + d_i(1-w_i)$, respectively. From Proposition 1 and the above, it follows that $(\alpha, \beta) \in \mathcal{S}_{\boldsymbol{x}}$, where $\mathcal{S}_{\boldsymbol{x}}$ is defined as follows:

$$\begin{aligned}
\mathcal{S}_{\boldsymbol{x}} =& \{(\alpha,\beta) : \alpha = \hat{c}_{\sigma(p)}, \beta = 0, \hat{c}_{\nu(p-k)} \leq \hat{c}_{\sigma(p)}, \hat{c}_{\sigma(p)} \in \mathcal{C}_{\sigma(p)}, \hat{c}_{\nu(p-k)} \in \mathcal{C}_{\nu(p-k)}\} \cup\\
&\{(\alpha,\beta) : \alpha = \hat{c}_{\varsigma(k)}, \beta = \hat{c}_{\nu(p-k)} - \hat{c}_{\varsigma(k)}, \hat{c}_{\nu(p-k)} > \hat{c}_{\varsigma(k)}, \hat{c}_{\nu(p-k)} \in \mathcal{C}_{\nu(p-k)}, \hat{c}_{\varsigma(k)} \in \mathcal{C}_{\varsigma(k)}\}.
\end{aligned}$$

Finally, (32) becomes

$$\begin{aligned}
\max \quad & p\alpha + (p-k)\beta - \sum_{i\in[n]} [\alpha + x_i\beta - \underline{c}_i]_+ \\
& - \sum_{i\in[n]} ([\alpha + x_i\beta - d_i - \underline{c}_i]_+ - [\alpha + x_i\beta - \underline{c}_i]_+)\delta_i \\
\text{s.t.} \quad & \sum_{i\in[n]} \delta_i \leq \Gamma \\
& \delta_i \in \{0,1\} \qquad\qquad i \in [n] \\
& (\alpha,\beta) \in \mathcal{S}_{\boldsymbol{x}}
\end{aligned} \tag{33}$$

Accordingly, it now suffices to solve (33) for each $(\alpha, \beta) \in \mathcal{S}_{\boldsymbol{x}}$ and choose the best of the computed solutions which encodes a worst scenario. Solving (33) for fixed (α, β) can be done in $O(n)$. Since the cardinality of the sets $\mathcal{C}_{\sigma(p)}$, $\mathcal{C}_{\nu(p-k)}$ and $\mathcal{C}_{\varsigma(k)}$ is $O(n)$, the cardinality of the set $\mathcal{S}_{\boldsymbol{x}}$ is $O(n^2)$. This leads to the following theorem:

Theorem 15. *The problem* AREC *under scenario set* $\mathcal{U}^d$ *can be solved in* $O(n^3)$ *time.*

3.1.2 The recoverable robust problem

We first identify some special cases of RREC (2) which are polynomially solvable.

Observation 16. *The following special cases of* RREC *under* $\mathcal{U}^d$ *are polynomially solvable:*

(i) if $k = 0$, *then* RREC *is solvable in* $O(n^2)$ *time,*

(ii) if $\Gamma = n$, *then* RREC *is solvable in* $O((p - k + 1)n^2)$ *time,*

(iii) if $k \geq \Gamma$ *and* $C_i = 0$, $i \in [n]$, *then* RREC *is solvable in* $O(n)$ *time.*

Proof. (i) If $k = 0$, then RREC is equivalent to the MinMax problem under scenario set $\mathcal{U}^d$ with the cost intervals $[C_i + \underline{c}_i, C_i + \overline{c}_i]$, $i \in [n]$. This problem is solvable in $O(n^2)$ according to the results obtained in [3].

(ii) If $\Gamma = n$, then RREC can be reduced to the recoverable robust problem under scenario set $\mathcal{U}^I$, which can be solved in $\mathcal{O}((p - k + 1)n^2)$ time [21].

(iii) Consider the case $k \geq \Gamma$. Let $\boldsymbol{x}^* \in \Phi$ be an optimal solution to the Selection problem for the costs $\underline{c}_i$, $i \in [n]$, and $\boldsymbol{\delta} \in \{0, 1\}^n$ stands for any vector that encodes scenario $\underline{c}_i + \delta_i d_i$, $i \in [n]$, $\sum_{i \in [n]} \delta_i \leq \Gamma$. Let $\boldsymbol{y}$ be an optimal solution to the Selection problem with respect to $\underline{c}_i + \delta_i d_i$, $i \in [n]$. Now $\boldsymbol{x}^*$ and $\boldsymbol{y}$ have at least $p - \Gamma$ elements in common, which is due to the fact that $\sum_{i \in [n]} \delta_i \leq \Gamma$. Since $k \geq \Gamma$, $\boldsymbol{x}^*$ can be recovered to $\boldsymbol{y}$. Hence, no better solution can exist.

□

We will now construct a compact MIP formulation for the general RREC problem under scenario set $\mathcal{U}^d$. In order to do this we will use the formulation (33). Observe that the sets $\mathcal{C}_{\nu(p-k)}$ and $\mathcal{C}_{\varsigma(k)}$, defined in Section 3.1.1, depend on a fixed solution $\boldsymbol{x}$ (the set $\mathcal{C}_{\sigma(p)}$ does not depend on $\boldsymbol{x}$). We now define $\boldsymbol{x}$-independent sets of possible values of the $(p-k)$th smallest element in $X_{\boldsymbol{x}}$ and the kth smallest element in $\overline{X_{\boldsymbol{x}}}$ under any scenario in $\mathcal{U}^d$ by

$$\begin{aligned}
&\hat{c}_{\sigma(p-k)} \in \mathcal{C}_{\sigma(p-k)} = \{\underline{c}_{\underline{\sigma}(p-k)}, \ldots, \underline{c}_{\underline{\sigma}(n-k+\Gamma)}\} \cup \{\overline{c}_{\overline{\sigma}(1)}, \ldots, \overline{c}_{\overline{\sigma}(n-k)}\},\\
&\hat{c}_{\sigma(k)} \in \mathcal{C}_{\sigma(k)} = \{\underline{c}_{\underline{\sigma}(k)}, \ldots, \underline{c}_{\underline{\sigma}(k+p+\Gamma)}\} \cup \{\overline{c}_{\overline{\sigma}(1)}, \ldots, \overline{c}_{\overline{\sigma}(k+p)}\}.
\end{aligned}$$

Again, from Proposition 1 and the above, we get a new set of relevant values of α and β

$$\begin{aligned}
\mathcal{S} =& \{(\alpha, \beta) : \alpha = \hat{c}_{\sigma(p)}, \beta = 0, \hat{c}_{\sigma(p-k)} \leq \hat{c}_{\sigma(p)}, \hat{c}_{\sigma(p)} \in \mathcal{C}_{\sigma(p)}, \hat{c}_{\sigma(p-k)} \in \mathcal{C}_{\sigma(p-k)}\} \cup\\
&\{(\alpha, \beta) : \alpha = \hat{c}_{\sigma(k)}, \beta = \hat{c}_{\sigma(p-k)} - \hat{c}_{\sigma(k)}, \hat{c}_{\sigma(p-k)} > \hat{c}_{\sigma(k)}, \hat{c}_{\sigma(p-k)} \in \mathcal{C}_{\sigma(p-k)}, \hat{c}_{\sigma(k)} \in \mathcal{C}_{\sigma(k)}\}.
\end{aligned}$$

Obviously $\mathcal{C}_{\nu(p-k)} \subseteq \mathcal{C}_{\sigma(p-k)}$, $\mathcal{C}_{\varsigma(k)} \subseteq \mathcal{C}_{\sigma(k)}$, $\mathcal{S}_{\boldsymbol{x}} \subseteq \mathcal{S}$ for any $\boldsymbol{x} \in \Phi$ and the cardinality of $\mathcal{S}$ remains $\mathcal{O}(n^2)$. Let us represent the adversarial problem (33) as follows:

$$
\begin{aligned}
\max_{(\alpha,\beta)\in\mathcal{S}} \max_{\boldsymbol{\delta}} \quad & p\alpha + (p-k)\beta - \sum_{i\in[n]} [\alpha + x_i\beta - \underline{c}_i]_+ + \\
& \sum_{i\in[n]} ([\alpha + x_i\beta - \underline{c}_i]_+ - [\alpha + x_i\beta - d_i - \underline{c}_i]_+)\delta_i \\
\text{s.t.} \quad & \sum_{i\in[n]} \delta_i \le \Gamma \\
& \delta_i \in \{0,1\} \qquad\qquad i \in [n]
\end{aligned}
\tag{34}
$$

Dualizing the inner SELECTION problem in (34), we get:

$$
\begin{aligned}
\max_{(\alpha,\beta)\in\mathcal{S}} \min_{\pi,\boldsymbol{\rho}} \ & \Gamma\pi + \sum_{i\in[n]} \rho_i + p\alpha + (p-k)\beta - \sum_{i\in[n]} [\alpha + x_i\beta - \underline{c}_i]_+ \\
\text{s.t.} \ & \pi + \rho_i \ge [\alpha + x_i\beta - \underline{c}_i]_+ - [\alpha + x_i\beta - \underline{c}_i - d_i]_+ && i \in [n] \\
& \rho_i \ge 0 && i \in [n] \\
& \pi \ge 0
\end{aligned}
$$

For every pair $(\alpha^\ell, \beta^\ell) \in \mathcal{S}$, we introduce a set of variables π^ℓ, $\boldsymbol{\rho}^\ell$, $\ell \in [S]$ with $S = |\mathcal{S}|$. The RREC problem under scenario set $\mathcal{U}^d$ is then equivalent to the following mathematical programming problem:

$$
\begin{aligned}
& \min \lambda \\
\text{s.t.} \ & \lambda \ge \sum_{i\in[n]} C_i x_i + \Gamma\pi^\ell + \sum_{i\in[n]} \rho_i^\ell + p\alpha^\ell + (p-k)\beta^\ell - \sum_{i\in[n]} [\alpha^\ell + x_i\beta^\ell - \underline{c}_i]_+ && \ell \in [S] \\
& \pi^\ell + \rho_i^\ell \ge [\alpha^\ell + x_i\beta^\ell - \underline{c}_i]_+ - [\alpha^\ell + x_i\beta^\ell - \underline{c}_i - d_i]_+ && i \in [n], \ell \in [S] \\
& \sum_{i\in[n]} x_i = p \\
& x_i \in \{0,1\} && i \in [n] \\
& \rho_i^\ell \ge 0 && \ell \in [S], i \in [n] \\
& \pi^\ell \ge 0 && \ell \in [S]
\end{aligned}
$$

Finally, we can linearize all the nonlinear terms $[a + bx_i]_+ = [a]_+ + ([a+b]_+ - [a]_+)x_i$, where a, b are constant. In consequence, we obtain a compact MIP formulation for RREC, with $O(n^3)$ variables and $O(n^3)$ constraints.

We now present an approximation algorithm, which can be applied for larger problem instances. Suppose that $\underline{c}_i \ge \alpha\overline{c}_i$ for each item $i \in [n]$, where $\alpha \in (0,1]$ is a constant. This inequality means that for each item i the nominal cost $\underline{c}_i$ is positive and $\overline{c}_i$ is at most $1/\alpha$ greater than $\underline{c}_i$. It is reasonable to assume that this condition will be true in many practical applications for not very large value of $1/\alpha$. Consider scenario set $\mathcal{U}' = \{(\underline{c}_i)_{i\in[n]}\}$, so $\mathcal{U}'$ contains only one scenario composed of the nominal item costs. Let $\hat{\boldsymbol{x}}$ be an optimal solution to the RREC problem under $\mathcal{U}'$. This solution can be computed in polynomial time [21]. Using the same reasoning as in [16], one can show that the cost of $\hat{\boldsymbol{x}}$ is at most $1/\alpha$ greater than the optimum. Hence there is an $1/\alpha$ approximation algorithm for RREC under scenario set $\mathcal{U}^d$.

3.2 Two-Stage Robust Selection

3.2.1 The adversarial problem

Let $\boldsymbol{x} \in \Phi_1$ be a fixed first stage solution, with $|X_{\boldsymbol{x}}| = p_1$. Using the same reasoning as in Section 2.2.1, we can represent the A2ST (5) problem as a special case of AREC with $\tilde{p} = p - p_1$ and $k = \tilde{p}$. In this case $\beta = 0$ in the formulation (31) and there are only $O(n)$ candidate values for α. Hence the problem can be solved in $O(n^2)$ time under scenario set $\mathcal{U}^d$.

We now show some additional properties of the adversarial problem, which will then be used to solve the more general R2ST problem. By dualizing the linear programming relaxation of the incremental problem (24), we find the following MIP formulation for the adversarial problem A2ST:

$$\begin{aligned} \max \quad & (p - \sum_{i\in[n]} x_i)\alpha - \sum_{i\in[n]}(1-x_i)\gamma_i \\ \text{s.t.} \quad & \alpha \le \gamma_i + \underline{c}_i + d_i\delta_i && i \in [n] \\ & \sum_{i\in[n]} \delta_i \le \Gamma \\ & \gamma_i \ge 0 && i \in [n] \\ & \delta_i \in \{0,1\} && i \in [n] \end{aligned} \tag{35}$$

The following lemma characterizes an optimal solution to (35):

Lemma 17. *There is an optimal solution to* (35) *in which $\alpha = 0$, $\alpha = \underline{c}_j$ or $\alpha = \overline{c}_j$ for some $j \in [n]$.*

Proof. Let us fix $\boldsymbol{\delta}$ in (35). Then the resulting linear program with additional slack variables is

$$\begin{aligned} \max\ & (p - \sum_{i\in[n]} x_i)\alpha - \sum_{i\in[n]}(1-x_i)\gamma_i \\ \text{s.t.}\ & \alpha + \epsilon_i - \gamma_i = \underline{c}_i + d_i\delta_i && i \in [n] \\ & \alpha \ge 0 \\ & \epsilon_i, \gamma_i \ge 0 && i \in [n] \end{aligned}$$

Note that we only consider nonnegative values of dual variable α associated with the cardinality constraint in (24), since replacing this constraint in (24) by $\sum_{i\in[n]}(y_i + x_i) \ge p$ does not change the set of optimal solutions. The problem has $2n+1$ variables and n constraints. If α is a non-basis variable in an optimal solution, then $\alpha = 0$. So, let us assume that α is a basis variable. As there are $n-1$ remaining basis variables, there is at least one $j \in [n]$ where both ϵ_j and γ_j are non-basis variables. Hence, $\alpha = \underline{c}_j + d_j\delta_j$ and the lemma follows since $\delta_j \in \{0,1\}$. □

Define $\mathcal{S} = \{0\} \cup \{\underline{c}_i : i \in [n]\} \cup \{\overline{c}_i : i \in [n]\}$ and write $\mathcal{S} = \{\alpha^1, \dots, \alpha^S\}$ with $S = |\mathcal{S}| \in O(n)$. Using Lemma 17, problem (35) is then equivalent to

$$
\begin{aligned}
\max_{\alpha \in \mathcal{S}} \max_{\boldsymbol{\delta},\boldsymbol{\gamma}} \ & (p - \sum_{i\in[n]} x_i)\alpha - \sum_{i\in[n]} (1-x_i)\gamma_i \\
\text{s.t. } & \gamma_i \geq \alpha - \underline{c}_i - d_i\delta_i \\
& \sum_{i\in[n]} \delta_i \leq \Gamma \\
& \gamma_i \geq 0 && i \in [n] \\
& \delta_i \in \{0,1\} && i \in [n]
\end{aligned}
$$

Let $(\boldsymbol{\delta}^*, \boldsymbol{\gamma}^*)$ be an optimal solution to the inner maximization problem. Note that we can assume that if $\delta_i^* = 0$, then $\gamma_i^* = [\alpha - \underline{c}_i]^+$ and if $\delta_i^* = 1$, then $\gamma_i^* = [\alpha - \underline{c}_i - d_i]^+$. Hence, the inner problem is equivalent to

$$
\begin{aligned}
\max \quad & (p - \textstyle\sum_{i\in[n]} x_i)\alpha - \sum_{i\in[n]} (1-x_i)[\alpha - \underline{c}_i]^+ \\
& + (1-x_i)([\alpha - \underline{c}_i - d_i]^+ - [\alpha - \underline{c}_i]^+)\delta_i \\
\text{s.t.} \quad & \textstyle\sum_{i\in[n]} \delta_i \leq \Gamma \\
& \delta_i \in \{0,1\} && i \in [n]
\end{aligned} \tag{36}
$$

As this is the Selection problem, we state the following result:

Theorem 18. *The problem A2ST under scenario set $\mathcal{U}^d$ can be solved in $O(n^2)$ time.*

3.2.2 The two-stage robust problem

The R2ST (3) problem is polynomially solvable when $\Gamma \geq n$. Scenario set $\mathcal{U}^d$ can be then replaced by $\mathcal{U}^I$ and the problem is solvable in $O(n)$ time [21].

We now present a compact MIP formulation for R2ST under scenario set $\mathcal{U}^d$. We can use the dual of the linear relaxation of (36) and the set $\mathcal{S}$ of candidate values for α to arrive at the following formulation:

$$
\begin{aligned}
\min \ & \lambda \\
\text{s.t. } & \lambda \geq \sum_{i\in[n]} C_i x_i + (p - \sum_{i\in[n]} x_i)\alpha^\ell - \sum_{i\in[n]} (1-x_i)[\alpha^\ell - \underline{c}_i]^+ + \Gamma\pi^\ell + \sum_{i\in[n]} \rho_i^\ell && \ell \in [S] \\
& \sum_{i\in[n]} x_i \leq p \\
& \pi^\ell + \rho_i^\ell \geq (1-x_i)([\alpha^\ell - \underline{c}_i]^+ - [\alpha^\ell - \underline{c}_i - d_i]^+) && i \in [n], \ell \in [S] \\
& x_i \in \{0,1\} && i \in [n] \\
& \pi^\ell \geq 0 && \ell \in [S] \\
& \rho_i^\ell \geq 0 && i \in [n], \ell \in [S]
\end{aligned}
$$

This formulation has $O(n^2)$ variables and $O(n^2)$ constraints.

We now propose a fast approximation algorithm for the problem. The idea is the same as for the robust recoverable problem (see Section 3.1.2 and also [16]). Let us fix scenario $\underline{\boldsymbol{c}} = (\underline{c}_i)_{i\in[n]} \in \mathcal{U}^d$, which is composed of the nominal item costs. Consider the following problem:

$$\min_{\boldsymbol{x}\in\Phi_1}(\boldsymbol{C}\boldsymbol{x} + \min_{\boldsymbol{y}\in\Phi_{\boldsymbol{x}}} \underline{\boldsymbol{c}}\boldsymbol{y}). \tag{37}$$

Problem (37) can be solved in $O(n)$ time in the following way. Let $c'_i = \min\{C_i, \underline{c}_i\}$ for each $i \in [n]$ and let $\boldsymbol{x} \in \Phi$ be an optimal solution to the Selection problem for the costs c'_i. The optimal solution $\hat{\boldsymbol{x}} \in \Phi_1$ to (37) is then formed by fixing $\hat{x}_i = 1$ if $x_i = 1$ and $c'_i = C_i$, and $\hat{x}_i = 0$ otherwise. We now prove the following result:

Proposition 19. *Let $\underline{c}_i \geq \alpha\overline{c}_i$ for each $i \in [n]$ and $\alpha \in (0, 1]$. Let $\hat{\boldsymbol{x}} \in \Phi_1$ be an optimal solution to (37). Then $\hat{\boldsymbol{x}}$ is an $\frac{1}{\alpha}$- approximate solution to* R2ST.

Proof. Let $\boldsymbol{x}^*$ be an optimal solution to R2ST with the objective value OPT. It holds

$$OPT = \boldsymbol{C}\boldsymbol{x}^* + \max_{\boldsymbol{c}\in\mathcal{U}^d} \min_{\boldsymbol{y}\in\Phi_{\boldsymbol{x}^*}} \boldsymbol{c}\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x}^* + \boldsymbol{c}^*\boldsymbol{y}^* \geq \boldsymbol{C}\boldsymbol{x}^* + \underline{\boldsymbol{c}}\boldsymbol{y}^*,$$

because $\underline{c}_i \leq c^*_i$ for each $i \in [n]$. Since $\hat{\boldsymbol{x}}$ is an optimal solution to (37) we get

$$\boldsymbol{C}\boldsymbol{x}^* + \underline{\boldsymbol{c}}\boldsymbol{y}^* \geq \boldsymbol{C}\hat{\boldsymbol{x}} + \underline{\boldsymbol{c}}\hat{\boldsymbol{y}},$$

where $\hat{\boldsymbol{y}} = \min_{\boldsymbol{y}\in\Phi_{\hat{\boldsymbol{x}}}} \underline{\boldsymbol{c}}\boldsymbol{y}$. By the assumption that $\underline{c}_i \geq \alpha\overline{c}_i$ for each $i \in [n]$, we obtain

$$\boldsymbol{C}\hat{\boldsymbol{x}} + \underline{\boldsymbol{c}}\hat{\boldsymbol{y}} \geq \boldsymbol{C}\hat{\boldsymbol{x}} + \alpha\overline{\boldsymbol{c}}\hat{\boldsymbol{y}}.$$

Finally, as $\alpha \in (0, 1]$

$$\boldsymbol{C}\hat{\boldsymbol{x}} + \alpha\overline{\boldsymbol{c}}\hat{\boldsymbol{y}} \geq \alpha(\boldsymbol{C}\hat{\boldsymbol{x}} + \overline{\boldsymbol{c}}\hat{\boldsymbol{y}}) \geq \alpha(\boldsymbol{C}\hat{\boldsymbol{x}} + \max_{\boldsymbol{c}\in\mathcal{U}^d} \boldsymbol{c}\hat{\boldsymbol{y}}) \geq \alpha(\boldsymbol{C}\hat{\boldsymbol{x}} + \max_{\boldsymbol{c}\in\mathcal{U}^d} \min_{\boldsymbol{y}\in\Phi_{\hat{\boldsymbol{x}}}} \boldsymbol{c}\boldsymbol{y})$$

and the proposition follows. □

4 Conclusion

The Selection problem is one of the main objects of study for the complexity of robust optimization problems. While the robust counterpart of most combinatorial optimization problems is NP-hard, its simple structure allows in many cases to construct efficient polynomial algorithms. As an example, the MinMax-Regret Selection problem was the first MinMax-Regret problem for which polynomial time solvability could be proved [2].

In this paper we continue this line of research by considering recoverable and two-stage robust problems combined with discrete and budgeted uncertainty sets. All four problem combinations have not been analyzed before, and little is known about other problems of this kind.

We showed that the continuous uncertainty problem variants allow polynomial-time solution algorithms, based on solving a set of linear programs. Additionally, we derived strongly polynomial combinatorial algorithms for the adversarial subproblems and discussed ways to preprocess instances.

For the discrete uncertainty case, we also presented strongly polynomial combinatorial algorithms for the adversarial problems, and constructed mixed-integer programming formulations of polynomial size. It remains an open problem to analyze in future research if the problems with discrete uncertainty are NP-hard or also allow for polynomial-time solution algorithms.

Further research includes the application of our setting to other combinatorial optimization problems, such as Spanning Tree or Shortest Path.

References

[1] H. Aissi, C. Bazgan, and D. Vanderpooten. Min-max and min-max regret versions of combinatorial optimization problems: a survey. *European Journal of Operational Research*, 197:427–438, 2009.

[2] I. Averbakh. On the complexity of a class of combinatorial optimization problems with uncertainty. *Mathematical Programming*, 90:263–272, 2001.

[3] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming*, 98:49–71, 2003.

[4] D. Bertsimas and M. Sim. The price of robustness. *Operations research*, 52:35–53, 2004.

[5] C. Büsing. Recoverable robust shortest path problems. *Networks*, 59:181–189, 2012.

[6] C. Büsing, A. M. C. A. Koster, and M. Kutschka. Recoverable robust knapsacks: the discrete scenario case. *Optimization Letters*, 5:379–392, 2011.

[7] P. Camion. Caractérisation des matrices unimodulaires. *Cahier du Centre d'Etudes de Recherche Opérationelle*, 5:181–190, 1963.

[8] A. Chassein and M. Goerigk. On the recoverable robust traveling salesman problem. *Optimization Letters*, 10:1479–1492, 2016.

[9] A. Chassein, M. Goerigk, A. Kasperski, and P. Zieliński. On recoverable and two-stage robust selection problems with budgeted uncertainty. *arXiv preprint arXiv:1701.06064*, 2017.

[10] E. Conde. An improved algorithm for selecting p items with uncertain returns according to the minmax regret criterion. *Mathematical Programming*, 100:345–353, 2004.

[11] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. MIT Press, 1990.

[12] O. Şeref, R. K. Ahuja, and J. B. Orlin. Incremental network optimization: theory and algorithms. *Operations Research*, 57:586–594, 2009.

[13] B. Doerr. Improved approximation algorithms for the Min-Max selecting Items problem. *Information Processing Letters*, 113:747–749, 2013.

[14] N. G. Frederickson and R. Solis-Oba. Increasing the weight of minimum spanning trees. *Journal of Algorithms*, 33:244–266, 1999.

[15] V. Gabrel, C. Murat, and A. Thiele. Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235:471–483, 2014.

[16] M. Hradovich, A. Kasperski, and P. Zieliński. Recoverable robust spanning tree problem under interval uncertainty representations. *Journal of Combinatorial Optimization*, 34(2):554–573, 2017.

[17] M. Hradovich, A. Kasperski, and P. Zieliński. The recoverable robust spanning tree problem with interval costs is polynomially solvable. *Optimization Letters*, 11:17–30, 2017.

[18] T. Ibaraki and N. Katoh. *Resource Allocation Problems*. The MIT Press, 1988.

[19] A. Kasperski. *Discrete Optimization with Interval Data - Minmax Regret and Fuzzy Approach*, volume 228 of *Studies in Fuzziness and Soft Computing*. Springer, 2008.

[20] A. Kasperski, A. Kurpisz, and P. Zieliński. Approximating the min-max (regret) selecting items problem. *Information Processing Letters*, 113:23–29, 2013.

[21] A. Kasperski and P. Zieliński. Robust recoverable and two-stage selection problems. *CoRR*, abs/1505.06893, 2015.

[22] A. Kasperski and P. Zieliński. Robust Discrete Optimization Under Discrete and Interval Uncertainty: A Survey. In *Robustness Analysis in Decision Aiding, Optimization, and Analytics*, pages 113–143. Springer, 2016.

[23] I. Katriel, C. Kenyon-Mathieu, and E. Upfal. Commitment under uncertainty: two-stage matching problems. *Theoretical Computer Science*, 408:213–223, 2008.

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

[25] C. Liebchen, M. E. Lübbecke, R. H. Möhring, and S. Stiller. The concept of recoverable robustness, linear programming recovery, and railway applications. In *Robust and Online Large-Scale Optimization*, volume 5868 of *Lecture Notes in Computer Science*, pages 1–27. Springer-Verlag, 2009.

[26] K. Lin and M. S. Chern. The most vital edges in the minimum spanning tree problem. *Information Processing Letters*, 45:25–31, 1993.

[27] E. Nasrabadi and J. B. Orlin. Robust optimization with incremental recourse. *CoRR*, abs/1312.4075, 2013.

[28] B. Roy. Robustness in operational research and decision aiding: A multi-faceted issue. *European Journal of Operational Research*, 200:629–638, 2010.