Optimization over Integers with Robustness in Cost and Few Constraints

Kai-Simon Goetzmann[1]*, Sebastian Stiller[2]**, and Claudio Telha[3]

[1] goetzmann@math.tu-berlin.de, Institut für Mathematik, TU Berlin
[2] sebastia@mit.edu, Sloan School of Management, MIT
[3] ctelha@mit.edu, Operations Research Center, MIT

Abstract. We consider robust counterparts of integer programs and combinatorial optimization problems (summarized as *integer problems* in the following), i.e., seek solutions that stay feasible if at most Γ-many parameters change within a given range. While there is an elaborate machinery for continuous robust optimization problems, results on robust integer problems are still rare and hardly general.

We show several optimization and approximation results for the robust (with respect to cost, or few constraints) counterpart of an integer problem under the condition that one can optimize or approximate the original integer problem with respect to a piecewise linear objective (respectively piecewise linear constraints).

For example, if there is a ρ-approximation for a minimization problem with non-negative costs and non-negative and bounded variables for piecewise linear objectives, then the cost robust counterpart can be $\rho(1+\varepsilon)$-approximated.

We demonstrate the applicability of our approach on two classes of integer programs, namely, totally unimodular integer programs and integer programs with two variables per inequality. Further, for combinatorial optimization problems our method yields polynomial time approximations and pseudopolynomial, exact algorithms for Robust Unbounded Knapsack Problems.

Keywords: Robust Optimization, Integer Programming, Total Unimodularity, Unbounded Knapsack, Integer Programs with two variables per inequality

1 Introduction

A solution to an optimization problem often has to be good not just for one instance but for a set of scenarios. This can either be due to uncertainty as to which of the scenarios will eventually occur, or because the solution shall be used several times in different scenarios.

* Supported by the Deutsche Forschungsgemeinschaft within the research training group 'Methods for Discrete Structures' (GRK 1408).

** Marie-Curie Fellow of the European Commission under the ROSES-Project (FP7-PEOPLE-2009-IOF 254402).

One solution concept for optimization over scenarios is Robust Optimization. In the robust paradigm feasibility and cost of a solution are measured by those scenarios in which the solution performs worst. This worst case approach contrasts to stochastic programming, where the cost of a solution is typically a weighted average over all scenarios, good ones and bad ones.

As an illustration, suppose we choose a route for regularly driving to work. We want to be on time no matter what happens, thus we have to evaluate each route by the travel time in its worst case scenario.

Robust optimization has thriven in the past decade, partly because its applicability became apparent, and partly because the resulting mathematical models allow for strong solution methods. For continuous problems a cohesive body of quite general methods has been developed. For combinatorial problems and integer linear programs (IPs) the picture is a lot more scattered. Typically, the results cover a specific combinatorial problem. This is of course a consequence of the richness of combinatorial optimization and integer linear programming. General results for all of these problems as in the continuous case are unlikely. Therefore the following result by Bertsimas and Sim is even more remarkable:

In [7], they show that for uncertain cost coefficients, where at most Γ of them can deviate from the nominal setting at the same time, solvability or approximability of any problem with binary decision variables extends to the robust case, as it suffices to solve a linear number of instances of the deterministic problem.

Bertsimas and Sim explicitly note that this result is intrinsically limited to binary variables. With the help of a new technique we get a corresponding result for integer, not necessarily binary cost robust problems[4]. Further, we can extend our method to general robust integer problems with uncertainty in one (or few) linear constraint(s). Restricting the latter result again to binary problems gives the exact sibling of the cost robust result in [7] for robustness in constraints. However, new insights were needed to translate the methods from [7] to the constraint robust setting.

Our Contribution: The main results of this paper are the following:

- The cost robust counterpart (in the same sense as in [7]) of an integer problem can be solved or approximated if the original problem can be solved for piecewise linear convex cost functions with at most two bends.
- To solve integer problems with uncertainty in a constant number of linear constraints, one has to solve a modified problem where the left hand sides of the constraints are replaced by piecewise linear convex functions.
- For *binary* problems with uncertainty in a constant number of linear constraints, it suffices to solve a polynomial number of instances of the original problem with slightly modified coefficients in the constraints.

At first sight, the requirement of solvability for piecewise linear functions seems clumsy, and not likely to be useful. To the contrary, we exemplify its

[4] We speak of integer *problems* to integrate IPs and combinatorial problems, where the feasibility sets need not be given explicitly by linear constraints.

usefulness by a number of quite different and broad applications of our results. Those general results allow us to develop methods for cost robust counterparts of entire classes of integer linear programs, notably, totally unimodular programs (TUM) and integer programs with two variables per constraint (IP2). Both classes have been studied intensely in the deterministic case, but we are not aware of any general results on their robust counterparts. Our general result on cost robust TUM problems broadly extends results on Robust Min Cost Flows in [7]. Further, we apply our general results to a combinatorial problem, namely, Unbounded Knapsack. In this case we derive an algorithm that handles cost robustness and robustness in the constraints at the same time.

We believe that this paper will motivate the consideration of piecewise linear cost functions and constraints for further classes of integer problems.

Let us remark that although many optimization problems with a natural non-binary IP description can be reformulated as binary IPs, this does usually not yield a workaround to apply results for robust binary IPs to the naturally non-binary problem – even granted the incurred blow-up of the instance. The hindrance is usually that the scenario sets of the robust counterparts make no sense once the problem is transformed into an unnatural binary program.

Related Work: Modern continuous robust optimization started with [20] for convex uncertainty sets and [4,5,6] for ellipsoidal uncertainty sets. Still a good overview for the state of the art is [1]. The Γ-scenario setting from [7,8] has found frequent application, e.g. in [9,10,16].

Robust Knapsack has so far only been considered in the binary setting. While for Γ-scenarios for uncertain costs the result from [7] applies, for the case of general scenarios there is no approximation algorithm at all [2,21]. Klopfenstein and Nace [17,18] considered polyhedral aspects of the robust Knapsack Problem, and in the context of the chance-constraint version also a weight robust Knapsack Problem. For the latter problem they derive a pseudo-polynomial algorithm, a result that also follows from our general result on constraint robust problems.

For integer linear programs with two variables per inequality (IP2), Hochbaum et al. [13] and Bar-Yehuda and Rawitz [3] provide a pseudopolynomial time 2-approximation algorithm. In case the inequalities are restricted to be monotone, there are pseudopolynomial time exact algorithms [3,14]. All algorithms explicitly assume that the variables are bounded.

Our general result generalizes a result for cost robust binary IPs [7], and our results for totally unimodular integer programs generalize results on specific totally unimodular problems, e.g., Min Cost Flows [7].

Structure of the paper: In Section 2 and 3 we present the general results on cost robust and constraint robust integer problems, respectively. In Section 4 we apply them to problems with a totally unimodular description and integer programs with two variables per inequality, enabling us to solve the cost robust counterparts, as well as to the Unbounded Knapsack Problem, where we can solve a robust version that features both, uncertainty in cost and weights.

2 Uncertainty in the Objective

We start with the general result on cost robust, not necessarily binary problems. We will use $[n]$ for the set $\{1, \dots, n\}$ here and throughout. Further, let us note that in our notation the set $\mathbb{N}$ includes the zero. By $\mathrm{T}_{\mathcal{A}}(I)$ we denote the running time of an algorithm $\mathcal{A}$ on an instance I.

The formal definition for the considered class of problems reads as follows:

Definition 1 (Cost Robust Optimization Problem). *For the optimization problems* $\min_{x \in X}\{c^{\mathrm{T}}x\}$ *and* $\max_{x \in X}\{c^{\mathrm{T}}x\}$*, given by* $P = (c, X)$ *where* $X \subseteq \mathbb{Z}^n$ *and* $c \in \mathbb{R}^n$*, and a non-negative integer vector* $d \in \mathbb{N}^n$ *together with* $\Gamma \in [n]$*, the* minimization (maximization) (d, Γ)-Cost Robust Counterpart (CRC) of P *is defined by*

$$\min_{x \in X}\left\{c^{\mathrm{T}}x + \max_{\substack{S \subseteq [n] \\ |S| \leq \Gamma}} \sum_{j \in S} |d_j x_j|\right\} \quad \textit{and} \quad \max_{x \in X}\left\{c^{\mathrm{T}}x - \max_{\substack{S \subseteq [n] \\ |S| \leq \Gamma}} \sum_{j \in S} |d_j x_j|\right\}, \tag{1}$$

respectively.

Our main goal is to show that one can solve or approximate the CRC of P, if one can solve or approximate the following variant of P:

Definition 2 (Modified Optimization Problem). *For the minimization (maximization) problem given by* $P = (X, c)$*,* $c' \in \mathbb{R}^n_{\geq 0}$ *and* $\alpha \geq 0$*, the* (c', α)-Modified Minimization (Maximization) Problem (MMin, MMax) of P *is*

$$\min_{x \in X}\left\{\sum\nolimits_{j \in [n]} \widetilde{c}_j(x_j)\right\} \qquad \textit{and} \qquad \max_{x \in X}\left\{\sum\nolimits_{j \in [n]} \widetilde{c}_j(x_j)\right\}, \tag{2}$$

where $\widetilde{c}_j(x_j) := c_j x_j \pm \max\{c'_j x_j - \alpha, 0\} \pm \max\{-c'_j x_j - \alpha, 0\}$ *for minimization ("+") and maximization ("−"), respectively.*

At this point minimization and maximization are fully symmetric. At a later stage it will come in handy to have them defined separately.

Theorem 3. *Consider the optimization problem of* $P = (c, X)$ *with* $X \subseteq \mathbb{Z}^n$ *and* $c \in \mathbb{R}^n$*. Suppose for some* $\rho \geq 1$ *there is a* ρ*-approximation algorithm* $\mathcal{A}_1$ *for the* (c', α)*-MMin (MMax) of* P *and arbitrary* c' *and* α*. Further suppose there is an algorithm* $\mathcal{A}_2$ *that, for given* $d \in \mathbb{N}^n$ *and* $\Gamma \in [n]$*, computes upper bounds* u_j *on the absolute value of each variable* x_j *in the optimal solution of the* (d, Γ)*-CRC of* P*. Then there is a* ρ*-approximation algorithm* $\mathcal{A}$ *for the* (d, Γ)*-CRC of* P *with running time* $\mathrm{T}_{\mathcal{A}}(P, d, \Gamma) \in \mathcal{O}\big(\mathrm{T}_{\mathcal{A}_2}(P, d, \Gamma) + \overline{\vartheta} \cdot \mathrm{T}_{\mathcal{A}_1}(P, d, \overline{\vartheta})\big)$*, where* $\overline{\vartheta} := \max_j\{u_j d_j\}$*.*

We will use $\overline{\vartheta} := \max_j\{u_j d_j\}$ throughout the remainder of this paper without defining it again.

Note that for $\rho = 1$, i.e., if we have an exact algorithm for the modified problem, we can solve the CRC exactly.

Proof. We only consider minimization problems, since we can transform any maximization problem into a minimization problem by taking the negative of the costs. Along the lines of the binary result from [7], we formulate the inner maximization problem of (1) as an IP, dualize and eliminate all but one dual variable to get the following reformulation of (1):

$$\min_{x \in X, \vartheta \geq 0} \Big\{ c^{\mathrm{T}} x + \Gamma\vartheta + \sum_{j \in [n]} \big(\max\{d_j x_j - \vartheta, 0\} + \max\{-d_j x_j - \vartheta, 0\} \big) \Big\}. \tag{3}$$

From this point on, the methods from [7] no longer apply because the variables are non-binary. We thus use our new technique, which utilizes the notion of the Modified Optimization Problem: For a fixed ϑ, (3) is equivalent to the (d, ϑ)-MMin of P. By the conditions of the Theorem, we can compute a ρ-approximate solution to this problem.

Let (x^*, ϑ^*) be an optimal solution to (3). We know that $|x_j^*| \leq u_j$, so if $\vartheta \geq \max_j u_j d_j = \overline{\vartheta}$, for all j both maxima in (3) vanish. Hence, if we increase ϑ beyond this number, the objective value increases. It follows that $\vartheta^* \leq \overline{\vartheta}$.

Also, we can assume that ϑ^* is integral: Denote by

$$C^*(\vartheta) := \Gamma\vartheta + \min_{x \in X} \Big\{ \sum_{j \in [n]} \big(c_j x_j + \max\{d_j x_j - \vartheta, 0\} + \max\{-d_j x_j - \vartheta, 0\} \big) \Big\} \tag{4}$$

the optimal cost for a fixed ϑ. Since x and d are integral, this function is linear in ϑ within each interval $[k, k+1]$, $k \in \mathbb{N}$. In such an interval the local maximum is obtained for $\vartheta = k$ or for $\vartheta = k + 1$, and thus the global maximum is obtained for some integral ϑ.

We can thus compute all ρ-approximate solutions corresponding to integral values of ϑ in $[0, \overline{\vartheta}]$, and choose the best among them, resulting in the claimed running time. □

Remark. Our model of robustness limits to deviation in at most Γ cost coefficients. The resulting inner maximization problem, which we dualized in the previous proof, is totally unimodular. Therefore a standard argument originating from [7] gives that this model is equivalent to protecting against any cost function $c + \delta d$ with δ in the set $\{\delta \in \mathbb{R}^n : \sum_{j \in [n]} |\delta_j| \leq \Gamma\}$.

Unless $\max_j u_j d_j$ is polynomial in the input, in Theorem 3 one ends up with a pseudopolynomial algorithm for the CRC, even if a polynomial algorithm for the modified optimization problem is given. This can be overcome if $\rho = 1$ and C^* as defined in (4) is convex as a function of ϑ, in which case ϑ^* can be found via a carefully constructed binary search (similar to the one in proof of Theorem 7 in [7]):

Theorem 4. *Consider the minimization problem of $P = (c, X)$ with $X \subseteq \mathbb{Z}^n$ and $c \in \mathbb{R}^n$. If the conditions of Theorem 3 hold, and if $\rho = 1$ and C^* is a convex function, then there is an exact algorithm $\mathcal{A}$ for the (d, Γ)-CRC of P with running time $\mathrm{T}_{\mathcal{A}}(P, d, \Gamma) \in \mathcal{O}\big(\mathrm{T}_{\mathcal{A}_2}(P, d, \Gamma) + \log(\overline{\vartheta}) \cdot \mathrm{T}_{\mathcal{A}_1}(P, d, \overline{\vartheta})\big)$.*

For an application of this result we refer the reader to the part on problems with totally unimodular description in Section 4.

When $\rho > 1$ or C^* is not convex, we can still restrict the number of calls of the oracle $\mathcal{A}_1$ to $\mathcal{O}(\log(\overline{\vartheta}))$ in exchange for a slightly weaker approximation guarantee. But for this result we have to consider minimization and maximization separately and restrict to combinatorial problems with non-negative cost coefficients and variables. Note that in this case the second maximum in both the definition of MMin and MMax vanishes.

It requires some additional non-trivial insights to prove that if ϑ^* is approximated, also the value of the solution will not deviate too much from the optimal value. We present these ideas in the following two proofs.

Theorem 5 (Minimization Problem). *Consider the minimization problem of $P = (c, X)$ with $X \subseteq \mathbb{N}^n$ and $c \in \mathbb{R}^n_{\geq 0}$. Under the conditions of Theorem 3, for all $\varepsilon > 0$ there is a $\rho(1+\varepsilon)$-approximation algorithm $\mathcal{A}$ for the (d, Γ)-CRC of P with running time $\mathrm{T}_{\mathcal{A}}(P, d, \Gamma) \in \mathcal{O}\big(\mathrm{T}_{\mathcal{A}_2}(P, d, \Gamma) + \frac{1}{\varepsilon} \cdot \log(\overline{\vartheta}) \cdot \mathrm{T}_{\mathcal{A}_1}(P, d, \overline{\vartheta})\big)$.*

Proof. We start as in the proof of Theorem 3. To attain the claimed running time, however, for any given $\varepsilon > 0$, we now solve (4) approximately for all $\vartheta \in \{0\} \cup \{(1+\varepsilon)^k : k \in \mathbb{N}, (1+\varepsilon)^{k-1} \leq \overline{\vartheta}\}$, and return the best of all these solutions. This yields a $\rho(1+\varepsilon)$-approximation for the CRC:

Let (x^*, ϑ^*) be an optimal solution to (3), w.l.o.g. $\vartheta^* \leq \overline{\vartheta}$ and $\vartheta^* \in \mathbb{N}$. In case $\vartheta^* \in \{0, 1\}$, our solution is within a factor of ρ of the optimum, since these two values for ϑ are checked. Otherwise, let $k_0 \in \mathbb{N} \setminus \{0\}$ be such that $(1+\varepsilon)^{k_0-1} < \vartheta^* \leq (1+\varepsilon)^{k_0} =: \vartheta_0$. Since Γ, ϑ^*, c, and $x \geq 0$, we get

$$\frac{C^*(\vartheta_0)}{C^*(\vartheta^*)} \leq \max\left\{\frac{\Gamma\vartheta_0}{\Gamma\vartheta^*}, \frac{\min_{x \in X}\left\{\sum_j c_j x_j + \max\{d_j x_j - \vartheta_0, 0\}\right\}}{\min_{x \in X}\left\{\sum_j c_j x_j + \max\{d_j x_j - \vartheta^*, 0\}\right\}}\right\} \leq 1 + \varepsilon\ .$$

Since we can compute ρ-approximations to $C^*(\vartheta)$, the best solution we find is a $\rho(1+\varepsilon)$-approximation for the CRC. Further, the oracle $\mathcal{A}_1$ is called $\mathcal{O}(\log_{(1+\varepsilon)} \overline{\vartheta}) = \mathcal{O}\big(\frac{1}{\varepsilon} \cdot \log(\overline{\vartheta})\big)$ times , resulting in the claimed running time. □

For maximization, the perturbed cost in a worst scenario can be relatively close to zero, while all numbers involved are rather large. This, roughly speaking, spoils an approximation result for maximization similar to Theorem 5 – unless we impose a further condition:

Theorem 6 (Maximization Problems). *Consider the maximization problem of $P = (c, X)$ with $X \subseteq \mathbb{N}^n$ and $c \in \mathbb{R}^n_{\geq 0}$. Suppose the conditions of Theorem 3 hold, and suppose that the relative cost decrease in the (d, Γ)-CRC of P is bounded from above by a constant $\beta < 1$, i.e.:*

$$\exists\, \beta < 1 : \qquad \frac{d_j}{c_j} \leq \beta \quad \forall\, j \in [n]\ .$$

Then there is a 2ρ-approximation algorithm $\mathcal{A}$ for the (d, Γ)-CRC of P with running time $\mathrm{T}_{\mathcal{A}}(P, d, \Gamma) \in \mathcal{O}\big(\mathrm{T}_{\mathcal{A}_2}(P, d, \Gamma) + \log(\overline{\vartheta}) \cdot \mathrm{T}_{\mathcal{A}_1}(P, d, \overline{\vartheta})\big)$.

Proof. As in the proof of Theorem 5, we solve the MMax of P for $\vartheta = (1+\varepsilon)^k$ for some $k \in \mathbb{N}$ and a particular $\varepsilon > 0$. For the choice of ε, consider an optimal solution (x^*, ϑ^*) with value OPT. We get that

$$\Gamma\vartheta^* \leq \Gamma\vartheta^* + \sum_{j\in[n]} \max\{d_j x_j^* - \vartheta^*, 0\} = \underbrace{c^{\mathrm{T}} x^* - \mathrm{OPT}}_{(1)} \overset{(*)}{\leq} d^{\mathrm{T}} x^* \leq \beta c^{\mathrm{T}} x^* \ ,$$

where $(*)$ holds because (1) is the cost we lose due to the decrease of some of the coefficients, and this cost is bounded by $d^{\mathrm{T}} x^*$.

We now set $\varepsilon := {}^{(1-\beta)}/_{2\beta}$ (w.l.o.g. $\beta > 0$). Then

$$\mathrm{OPT} \geq (c-d)^{\mathrm{T}} x^* \geq (1-\beta) c^{\mathrm{T}} x^* = 2\varepsilon\beta c^{\mathrm{T}} x^* \geq 2\varepsilon\Gamma\vartheta^* \ .$$

With this, we can bound the error that arises from approximating ϑ^*:

Denote by $C^*(\vartheta) := -\Gamma\vartheta + \max_{x\in X}\left\{\sum_{j\in[n]} c_j x_j - \max\{d_j x_j - \vartheta, 0\}\right\}$ the optimal cost for a fixed ϑ. With ϑ_0 as in the proof of Theorem 5 we then get

$$\begin{aligned}\frac{\mathrm{OPT}}{C^*(\vartheta_0)} &\leq \frac{\mathrm{OPT}}{-\Gamma\vartheta_0 + \max_{x\in X}\left\{\sum_{j\in[n]}\left(c_j x_j - \max\{d_j x_j - \vartheta^*, 0\}\right)\right\}} \\ &= \frac{\mathrm{OPT}}{-\Gamma\vartheta_0 + \mathrm{OPT} + \Gamma\vartheta^*} \leq \frac{\mathrm{OPT} - \varepsilon\Gamma\vartheta^* + \varepsilon\Gamma\vartheta^*}{-\Gamma(1+\varepsilon)\vartheta^* + \mathrm{OPT} + \Gamma\vartheta^*} \\ &= 1 + \frac{\varepsilon\Gamma\vartheta^*}{\mathrm{OPT} - \varepsilon\Gamma\vartheta^*} \leq 2 \ .\end{aligned}$$

Since we are able to approximate the optimal solution to the MMax of P within a factor of ρ, the considerations above prove that our algorithm yields a 2ρ-approximation. The number of calls of $\mathcal{A}_1$ is the same as in the proof of Theorem 5. Since ε is constant, we get the claimed overall running time. □

3 Uncertainty in Constraints

We now turn to the case where the coefficients of a single linear constraint (or those of a constant number of them) are uncertain. In the setting considered here minimization and maximization are equivalent, so we restrict to one of the two. The formal definition of the considered class of problems is as follows:

Definition 7 (Constraint Robust Maximization Problem). *Consider the problem* $\max_{x\in X}\{c^{\mathrm{T}} x\}$, *given by* $P = (c, X)$ *where* $c \in \mathbb{R}^n$ *and* $X = \{x \in X' : a^{\mathrm{T}} x \leq r\}$ *for some* $X' \subseteq \mathbb{Z}^n, a \in \mathbb{R}^n, r \in \mathbb{R}$. *For a non-negative integer vector* $b \in \mathbb{N}^n$ *together with* $\Gamma \in [n]$, *the* (b, Γ)-Constraint Robust Counterpart (ConsRC) of P *is defined by*

$$\max c^{\mathrm{T}} x \qquad s.t. \quad x \in X', \quad a^{\mathrm{T}} x + \max_{\substack{S\subseteq[n]\\|S|\leq\Gamma}} \sum_{j\in S} |b_j x_j| \leq r. \tag{5}$$

As in the cost robust setting, the left hand side of the constraint with uncertain coefficients can be transformed into a sum of piecewise linear convex function with two bends:

Lemma 8. *The (b,Γ)-Constraint Robust Counterpart of the maximization problem $P=(c,X)$ as defined in Definition 7 is equivalent to*

$$\max_{\xi\geq 0}\max_{x\in X(\xi)} c^{\mathrm{T}}x, \qquad \textit{with} \tag{6}$$
$$X(\xi) := \Big\{x\in X' \colon \Gamma\xi + \sum_{j\in[n]}\big(a_jx_j + \max\{b_jx_j-\xi,0\} + \max\{-b_jx_j-\xi,0\}\big)\leq r\Big\}.$$

Proof. With the same transformations as in the cost robust setting, we get that (5) is equivalent to

$$\max c^{\mathrm{T}}x \qquad \text{s.t.} \quad x\in X' \quad \text{and}$$
$$\min_{\xi\geq 0}\Big\{\Gamma\xi + \sum\nolimits_j a_jx_j + \max\{b_jx_j-\xi,0\} + \max\{-b_jx_j-\xi,0\}\Big\}\leq r.$$

Thus, for all feasible solutions x of (5) there exists some $\xi(x)\geq 0$ such that $x\in X(\xi(x))$. Consequently, (5) is equivalent to $\max_{\xi\geq 0}\max_{x\in X(\xi)} c^{\mathrm{T}}x$. □

For the non-binary case, the optimal ξ^* can be found by enumeration, since it is integral and bounded by the maximum deviation in the constraint coefficients:

Corollary 9. *Consider the (b,Γ)-ConsRC of the maximization problem $P=(c,X)$ as defined in Definition 7. Suppose there is an algorithm $\mathcal{A}_1$ computing a ρ-approximation for $\max_{x\in X(\xi)} c^{\mathrm{T}}x$ for any $\xi\geq 0$, and an algorithm $\mathcal{A}_2$ that computes upper bounds u_j on the absolute value of each variable x_j in the optimal solution of (5). Then there is a ρ-approximation algorithm $\mathcal{A}$ for the (b,Γ)-ConsRC of P with running time $\mathrm{T}_{\mathcal{A}}(P,b,\Gamma)=\mathcal{O}\big(\mathrm{T}_{\mathcal{A}_2}(P,b,\Gamma)+\overline{\xi}\cdot\mathrm{T}_{\mathcal{A}_1}(P,b,\Gamma,\overline{\xi})\big)$, where $\overline{\xi}:=\max_j\{u_jb_j\}$.*

If all variables are binary, i.e. $X'\subseteq\{0,1\}^n$, there are only $n+1$ possibilities for ξ^*, and for a fixed ξ the constraint of problem (6) becomes linear again. Hence, to solve the (b,Γ)-ConsRC of $P=(c,X)$, it suffices to solve $n+1$ problems of the type of P for slightly different coefficients in the linear constraint.

This result is an exact sibling of the result on cost robust binary problems in [7], but it requires some new insights to translate the methods from [7] to the constraint robust setting.

Theorem 10. *If $X'\subseteq\{0,1\}^n$, the (b,Γ)-ConsRC of the maximization problem $P=(c,X)$ as defined in Definition 7 is equivalent to*

$$\max_{\ell=1,\dots,n+1}\left(\max c^{\mathrm{T}}x \qquad \textit{s.t.} \quad x\in X', \quad \Gamma b_\ell + a^{\mathrm{T}}x + \sum_{j=1}^{\ell-1}(b_j-b_\ell)x_j\leq r\right),$$

whereby w.l.o.g. we assume $b_n\leq b_{n-1}\leq\ldots\leq b_1$ and define $b_{n+1}:=0$.

Proof. We know that $\xi^*\in[0,b_1]$. We split up this interval at $b_\ell,\ell=n,\dots,2$, and maximize over each subinterval, i.e. we reformulate (6) to get

$$\max_{\ell=1,\dots,n}\Big(\max_{\xi\in[b_{\ell+1},b_\ell]}\Big(\max_{x\in X(\xi)} c^{\mathrm{T}}x\Big)\Big). \tag{7}$$

For $x \in \{0,1\}^n$ we have $\max\{b_j x_j - \xi, 0\} = \max\{b_j - \xi, 0\} x_j$, and thus for $\xi \in [b_{\ell+1}, b_\ell]$

$$\begin{aligned} X(\xi) &= \Big\{ x \in X' : \varGamma\xi + \sum_{j\in[n]} \big(a_j x_j + \max\{b_j - \xi, 0\} x_j \big) \le r \Big\} \\ &= \Big\{ x \in X' : \varGamma\xi + a^{\mathrm{T}} x + \sum_{j=1}^{\ell} (b_j - \xi) x_j \le r \Big\}. \end{aligned} \tag{8}$$

For any fixed x, the left hand side of the constraint in (8) is a linear function in ξ that has to be no greater than r somewhere in $[b_{\ell+1}, b_\ell]$ for x to be feasible. Thus, if the constraint is satisfied for any ξ in this interval, because of linearity it will be satisfied for at least one of the values $\xi = b_{\ell+1}$ or $\xi = b_\ell$. As a consequence,

$$\max_{\xi \in [b_{\ell+1}, b_\ell]} \Big(\max_{x \in X(\xi)} c^{\mathrm{T}} x \Big) = \max_{\xi = b_{\ell+1}, b_\ell} \Big(\max_{x \in X(\xi)} c^{\mathrm{T}} x \Big). \tag{9}$$

Combining (7)–(9) yields the claimed result. □

As a corollary from Theorem 10 we get the existence of a pseudopolynomial exact algorithm as well as an FPTAS for the weight robust counterpart of the binary Knapsack Problem, generalizing a result from [18].

All the results from this section hold as well if there is a constant number k of constraints with uncertain coefficients. The problem $\max_{x \in X(\xi)} c^{\mathrm{T}} x$ would then have to be solved $(\max_j \{u_j b_j\})^k$ times in the setting of Corollary 9 and $(n+1)^k$ times in the binary case.

4 Applications

The final section is devoted to applications of the general results presented above. We first consider the cost robust setting for problems with a totally unimodular description and IPs with two variables per inequality, and then study the Unbounded Knapsack Problem, both with uncertain weights and cost, integrating our general results.

Problems with Totally Unimodular Description. The concept of totally unimodular matrices is arguably the most successful concept for solving a large class of integer programs. In general, robust counterparts need not inherit total unimodularity. We show that in our setting, however, the CRC of P can be solved exactly for those problems where the solution space of P can be described by a totally unimodular matrix of size polynomial in the size of the input of P.

This generalizes results on specific totally unimodular problems. In particular, it broadly generalizes the results on Robust Network Flows in [7], since the Min Cost Flow Problem is totally unimodular.

In this section we do not require non-negativity of the cost vector, so the minimization results we show can be used for maximization problems as well. We do require non-negative variables. This condition can be lifted, but this yields much less readable results that rest on similar arguments.

Definition 11. *A minimization problem $P = (c, X)$ is said to have a* bounded TUM description (A, b, u) *if the set of feasible solutions $X \subseteq \mathbb{N}^n$ is described by a totally unimodular matrix $A \in \mathbb{R}^{m \times n}$, an integral right-hand-side b, and an integral vector of upper bounds u, i.e.*

$$\operatorname{conv}(X) = \{x \in \mathbb{R}^n : Ax \leq b, x \leq u\}, \qquad A \text{ TUM}, b \in \mathbb{Z}^m, u \in \mathbb{Z}^n.$$

To apply Theorem 4 to solve problems of this kind, we need to establish the following two lemmas:

Lemma 12. *If the minimization problem $P = (c, X)$ is given by a bounded TUM description (A, b, u), then the Modified Minimization Problem can be solved in polynomial time.*

Lemma 13. *Let $C^*(\vartheta)$ be defined as in (4). Then for a minimization problem $P = (c, X)$ with a bounded TUM description (A, b, u), C^* is convex for any $c \in \mathbb{R}^n, d \in \mathbb{N}^n, \mathit{\Gamma} \in [n]$.*

The key idea is to split up each variable into three to model the piecewise linear cost function, and to observe that the resulting LP is still totally unimodular. For details we refer the reader to the technical report [11].

With Theorem 4 and the two lemmas, we get that we can solve the CRC of any problem with a bounded totally unimodular description in polynomial time:

Theorem 14. *If the minimization problem $P = (c, X)$ is given by a bounded TUM description (A, b, u), then for any $c \in \mathbb{R}^n, d \in \mathbb{N}^n$ and $\mathit{\Gamma} \in [n]$, there is an exact algorithm for the $(d, \mathit{\Gamma})$-CRC of P that runs in polynomial time.*

Integer Programs with Two Variables per Inequality. We now apply our main results to a second, large, and intensely studied class of integer programs, namely integer programs with two variables per inequality (IP2).

Definition 15 (Integer Programs with Two Variables per Inequality). *A* bounded integer program with two variables per inequality *(bounded IP2) is a system of the form*

$$\min \{c^{\mathrm{T}} x \colon a_i^{\mathrm{T}} x \geq b_i \text{ for } i = 1, \ldots, m,\ \ell \leq x \leq u,\ x \text{ integer}\},$$

where $b \in \mathbb{Z}^m$, $\ell, u \in \mathbb{Z}^n$, $c \in \mathbb{Q}^n$ and each vector $a_i \in \mathbb{Z}^n$ has two non-zero components.

A bounded IP2 is called monotone *if the non-zero coefficients of a_i have opposite signs.*

The conditions required in Section 2 allow to intensely use the existing techniques for non-robust IP2, in particular [13], [14] and [3]. We obtain a pseudopolynomial time 2-approximation for the CRC of bounded IP2s and an exact pseudopolynomial time algorithm for the CRC of bounded, monotone IP2s.

Theorem 16. *The cost robust counterpart of a bounded monotone IP2 can be solved in pseudopolynomial time.*

Remark. Theorem 16 can be established by extending (to handle piecewise linear functions) the pseudopolynomial time algorithm of Hochbaum and Naor [14] for bounded monotone IP2, cf. [11]. In [3], Bar-Yehuda and Rawitz give an exact pseudopolynomial algorithm for monotone cost functions but non-negative lower bounds.

Theorem 17. *There is a pseudopolynomial time 2-approximation algorithm for the cost robust counterpart of a bounded IP2 with non-negative coefficients in the objective function and non-negative lower bounds.*

Remark. Theorem 17 is proven by extending (to handle piecewise linear functions) the pseudopolynomial 2-approximation algorithm of Hochbaum et al. [13], cf. [11]. This result is also shown in [3].

Robust Unbounded Knapsack Problems. To demonstrate how versatile our main results are for combinatorial problems, we apply them to the *Unbounded Knapsack Problem*, the non-binary extension of the classical Knapsack Problem (KP). For this problem we will be able to handle counterparts that feature both, cost robustness and robustness in the constraint.

Definition 18 (Unbounded Knapsack Problem). *An instance of the* Unbounded Knapsack Problem *(UKP) is given by a knapsack capacity $W \geq 0$ and n types of items with weights $w_j \in \mathbb{N}$ and costs $c_j \in \mathbb{R}_{\geq 0}$, $j \in [n]$. The task is to find a vector $x \in \mathbb{N}^n$ with $\sum_j w_j x_j \leq W$ maximizing the cost $\sum_j c_j x_j$.*

UKP and its extensions, in particular its robust counterparts, are *NP*-hard. Intuitively, UKP seems to be more complex than the binary KP, since the input is more compact. Still, as for KP, there is both a pseudopolynomial Dynamic Program (DP) and an FPTAS [19,15].

We now consider the robust versions of the Unbounded Knapsack Problem.

Cost Robust UKP (CRUKP). While the result for binary cost robust programs [7] can be applied to the standard Knapsack Problem, the CRC of UKP surpasses the reach of [7]. As argued earlier, a reformulation as a binary integer program does not only cause a blow-up in size, but it also renders the scenario set meaningless. Thus, to solve CRUKP, we need to be able to solve UKP for piecewise linear concave cost functions. In [12], Hochbaum presented an FPTAS for this problem. We give an alternative FPTAS based on a dynamic program (DP) in our technical report [11]. With these results, by Theorem 6 it follows that for all $\varepsilon > 0$, there is a $(2 + \varepsilon)$-approximation algorithm for CRUKP, if the relative cost decrease is bounded away from 1 by a constant. On the other hand, using Theorem 3 with the DP from [11], we get an exact algorithm with pseudopolynomial running time.

Weight Robust UKP (WRUKP). Next we turn to the Unbounded Knapsack Problem where weights instead of costs are uncertain. In terms of Section 3 we have uncertainty in the only constraint. We consider the $(\Delta w, \Gamma)$-ConsRC of UKP, where Δw_j denotes the possible increase in weight of items of type j. From Corollary 9 we learn that we have to solve $\max_{x \in X(\xi)} c^{\mathrm{T}} x$ in order to get a pseudopolynomial algorithm for WRUKP. The FPTAS from [12] could be used for this. Alternatively, we can compute an exact solution in pseudopolynomial time by the DP described in [11]. With $u_j = \frac{W}{w_j}$, this yields an exact algorithm for WRUKP with running time $\mathcal{O}(\max_j \frac{\Delta w_j}{w_j} \cdot n^2 W^2)$.

General Robust UKP (RUKP). Finally, we consider a version of UKP where both weights and costs are uncertain. At most Γ_w types of items can increase their weight, and at most Γ_c cost coefficients decrease. This is the $(\Delta w, \Gamma_w)$-ConsRC of CRUKP. Since the DP from [11] works for concave cost and convex weight functions, by Theorem 3 we get an exact algorithm $\mathcal{A}_1$ for CRUKP on the modified solution space $X(\xi)$ with a running time of $\mathcal{O}(\max_j \frac{d_j}{w_j} \cdot n^2 W^2)$, and can thus solve RUKP exactly in a running time of $\mathcal{O}(\max_j \frac{\Delta w_j}{w_j} \cdot \max_j \frac{d_j}{w_j} \cdot n^2 W^3)$.

Acknowledgement. We are grateful to Martin Skutella and Günter Rote for discussions that substantially enhanced this paper.

References

1. Special issue on robust optimization. *Math. Program.*, 107(1-2), 2006.
2. H. Aissi, C. Bazgan, and D. Vanderpooten. Approximation of min-max and min-max regret versions of some combinatorial optimization problems. *Europ. J. of Oper. Res.*, 179(2):281–290, 2007.
3. R. Bar-Yehuda and D. Rawitz. Efficient algorithms for integer programs with two variables per constraint 1. *Algorithmica*, 29(4):595–609, 2001.
4. A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Math. Oper. Res.*, 23(4):769–805, 1998.
5. A. Ben-Tal and A. Nemirovski. Robust solutions to uncertain linear programs. *Oper. Res. Letters*, 25(1):1–13, 1999.
6. A. Ben-Tal and A. Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Math. Program.*, 88(3):411–424, 2000.
7. D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Math. Program.*, 98(1-3):49–71, 2003.
8. D. Bertsimas and M. Sim. The price of robustness. *Oper. Res.*, 52(1):35–53, 2004.
9. U. Feige, K. Jain, M. Mahdian, and V. Mirrokni. Robust combinatorial optimization with exponential scenarios. In *Proc. of IPCO*, pages 439–453. Springer, 2007.
10. Matteo Fischetti and Michele Monaci. Light robustness. In *Robust and Online Large-Scale Optimization*, pages 61–84. Springer, 2009.
11. Kai-Simon Goetzmann, Sebastian Stiller, and Claudio Telha. Optimization over integers with robustness in cost and few constraints. Technical Report 009-2011, Technische Universität Berlin, 2011.
12. D. Hochbaum. A nonlinear knapsack problem. *Oper. Res. Lett.*, 17:103–110, 1995.
13. D. Hochbaum, N. Megiddo, J. Naor, and A. Tamir. Tight bounds and 2-approximation algorithms for integer programs with two variables per inequality. *Math. Program.*, 62(1):69–83, 1993.

14. D. Hochbaum and J. Naor. Simple and fast algorithms for linear and integer programs with two variables per inequality. *SIAM J. Comput.*, 23:1179–1192, 1994.
15. O. H. Ibarra and C. E. Kim. Fast approximation algorithms for the knapsack and sum of subset problems. *J. ACM*, 22:463–468, 1975.
16. R. Khandekar, G. Kortsarz, V. Mirrokni, and M. Salavatipour. Two-stage robust network design with exponential scenarios. In *Proc. of ESA*, pages 589–600. Springer, 2008.
17. O. Klopfenstein and D. Nace. A note on polyhedral aspects of a robust knapsack problem. Available online at www.optimization-online.org, 2007.
18. O. Klopfenstein and D. Nace. A robust approach to the chance-constrained knapsack problem. *Oper. Res. Letters*, 36(5):628–632, 2008.
19. S. Martello and P. Toth. *Knapsack Problems. Algorithms and Computer Implementations*. John Wiley and Sons, 1990.
20. A. L. Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Oper. Res.*, 21(5):1154–1157, 1973.
21. G. Yu. On the max-min 0-1 knapsack problem with robust optimization applications. *Oper. Res.*, 44(2):407–415, 1996.