Math. Program., Ser. B 108, 235–250 (2006)

Digital Object Identifier (DOI) 10.1007/s10107-006-0709-5

Alper Atamtürk

Strong Formulations of Robust Mixed 0–1 Programming

Received: January 27, 2004 / Accepted: March 17, 2005
Published online: April 25, 2006 – © Springer-Verlag 2006

Abstract. We introduce strong formulations for robust mixed 0–1 programming with uncertain objective coefficients. We focus on a polytopic uncertainty set described by a "budget constraint" for allowed uncertainty in the objective coefficients. We show that for a robust 0–1 problem, there is an α–tight linear programming formulation with size polynomial in the size of an α–tight linear programming formulation for the nominal 0–1 problem. We give extensions to robust mixed 0–1 programming and present computational experiments with the proposed formulations.

Key words. Robust optimization – Polyhedra – Modeling – Computation

1. Introduction

Robust optimization is emerging as a practical way of handling uncertainty in model parameters. Recently there has been a considerable interest in addressing issues such as controlling conservatism of the robust solutions and computational complexity of the robust models. We refer the readers to [4–7, 9, 13–15] for some of the recent developments. The efficient approaches proposed in these works spurred interesting applications in finance [12, 16] and inventory management [11]. Most of the work on robust optimization is concentrated on convex optimization. Literature on robust discrete optimization, which is the topic of this paper, is so far limited; see [1, 2, 8, 10, 18] for a few available examples. For earlier robust optimization approaches on discrete optimization see [19].

1.1. Problem description

Consider a 0–1 programming problem

$$\min_x \left\{ c'x : x \in \mathcal{F} \right\}, \tag{1}$$

where $c \in \mathbb{R}^n$ is the objective vector and $\mathcal{F} \subseteq \{0, 1\}^n$ is the set of feasible solutions. Now suppose that coefficients of the objective are not fixed, but are uncertain values that lie in the interval $a \leq c \leq a + d$, where $d \geq 0$. Under such objective uncertainty, Bertsimas and Sim [8] define an interesting *robust counterpart* for (1) as

$$\min_x \left\{ a'x + \max_{S \subseteq [1,n], |S| \leq r} \sum_{i \in S} d_i x_i \; : \; x \in \mathcal{F} \right\}, \tag{2}$$

A. Atamtürk: Industrial Engineering and Operations Research, University of California, 4141 Etcheverry Hall, CA 94720-1777, Berkeley, USA. e-mail: atamturk@ieor.berkeley.edu

Mathematics Subject Classification (2000): 93B35, 90C11, 90C47

where r is the maximum number of variables for which the objective coefficient is allowed to differ from a_i. In other words, r is a parameter used for controlling the degree of conservatism of the solution for (2). The bigger the r is, the more conservative is the solution. One obtains a *nominal* problem of the form (1) if $r = 0$ or $r = n$, by setting $c = a$ or $c = a + d$, respectively. Bertsimas and Sim show that (2) can be formulated as a linear mixed 0–1 program and if the nominal problem (1) is polynomially solvable, then so is its robust counterpart (2) for any value of r.

We state the robust counterpart of (1) in a more general form than (2) by using a *budget uncertainty set*. Let the robust counterpart of (1) be defined as

$$\min_{\xi,x} \left\{ \xi : \xi \ge c'x,\ x \in \mathcal{F} \text{ for all } c \in \mathcal{B} \right\}, \tag{3}$$

where

$$\mathcal{B} = \left\{ c \in \mathbb{R}^n : \pi'c \le \pi_o,\ a \le c \le a + g \right\}.$$

Here $\mathcal{B}$ is a rational polytope defined by bounds a, $a + g$, and a "budget constraint" $\pi'c \le \pi_o$ representing the allowed uncertainty in the objective coefficients. Letting $u = c - a$, we rewrite (3) as

$$\min_x \left\{ a'x + \max_u u'x : x \in \mathcal{F},\ \pi'u \le h,\ u \le g,\ u \in \mathbb{R}^n_+ \right\}, \tag{4}$$

where $h = \pi_o - \pi'a$. Without loss of generality, we may assume that $g, \pi, h > 0$, since $g_i = 0$ implies $u_i = 0$ and if $\pi_i \le 0$, then u_i can be eliminated from the problem by setting $u_i = g_i$. By defining $w_i = \pi_i u_i$, $b_i = \pi_i g_i$, and $d_i = 1/\pi_i$ for $i \in [1, n]$ and rewriting (4) as

$$\min_x \left\{ a'x + \max_w \sum_{i=1}^n d_i w_i x_i : x \in \mathcal{F},\ \mathbf{1}'w \le h,\ w \le b,\ w \in \mathbb{R}^n_+ \right\}, \tag{5}$$

we see that (5) reduces to (2) by taking $h = r \in \mathbb{Z}_+$ and $b = \mathbf{1}$ as $w \in \{0, 1\}^n$ for the extreme points of the uncertainty set in this case. After linearizing the objective of (5) using the dual of the inner maximization problem, (5) is rewritten as the following mixed 0–1 program

$$\text{(RP)} \qquad \min \left\{ a'x + b'y + hz : (x, y, z) \in \mathcal{S} \right\},$$

where

$$\mathcal{S} = \{ (x, y, z) : x \in \mathcal{F},\ (x, y, z) \in \mathcal{R} \}$$

and

$$\mathcal{R} = \left\{ (x, y, z) \in \{0, 1\}^n \times \mathbb{R}^{n+1}_+ : y_i + z \ge d_i x_i,\ \ i \in [1, n] \right\}.$$

Let $d_0 = 0$. As z equals d_i, $i \in [0, n]$ in extreme points of the convex hull of $\mathcal{R}$, RP can be solved by at most $n + 1$ calls to the nominal problem (1) by setting $z = d_i$ and $y_j = (d_j - d_i)^+ x_j$ for all $j \in [1, n]$, $i \in [0, n]$ since $b > 0$. Consequently, as in the case of (2), if the nominal problem (1) is polynomially solvable, then so is the robust counterpart (3) for any budget uncertainty set $\mathcal{B}$ for the objective.

1.2. Motivation and outline

The mixed–integer programming (MIP) formulation RP is in general difficult to solve with a linear programming (LP) based MIP solver. In their computational experiments, Bertsimas and Sim observe that even if the nominal problem is a trivial one over a cardinality set $\mathcal{F} = \left\{x \in \{0, 1\}^n : \mathbf{1}'x = k\right\}$, RP takes an unusually long time to solve. The difficulty is due to the fact that even if the LP relaxation of $\mathcal{F}$ is integral, the LP relaxation of RP is typically highly fractional and provides a weak bound for the optimal value.

The purpose of this paper is to introduce alternative formulations for robust (mixed) 0–1 programming with strong LP relaxation bounds. For a minimization problem we say that formulation A is *stronger* than formulation B, if the optimal value of the LP relaxation of formulation A is bigger than the optimal value of the LP relaxation of formulation B for any objective. Thus, with the strength of a formulation we refer to the strength of its LP relaxation bound. As most problems with binary variables are $\mathcal{NP}$–hard, strong formulations of their robust counterpart are essential for solving them by enumerative methods such as branch–and–bound or branch–and–cut, as well as by LP–based approximation methods. To this end, we investigate the set $\mathcal{R}$, which is the part of the formulation RP used for modeling robustness, independent from the nominal problem.

In Section 3 we introduce reformulations of RP based on three different linear descriptions of the convex hull of $\mathcal{R}$. We also show that for a robust 0–1 problem, there is an α–tight linear programming relaxation with size polynomial in the size of an α–tight linear programming relaxation for the nominal 0–1 problem. In Section 4 we extend some of the results to mixed 0–1 programming. Strong formulations are of significant interest for robust mixed 0–1 programming since practical problems typically include 0–1 as well as continuous variables. Furthermore, unlike the 0–1 case, polynomial algorithms are unknown for the robust counterpart of a mixed 0–1 problem, even if the nominal problem is polynomially solvable. We present a summary of computational experiments with the proposed formulations in Section 5 and conclude with Section 6.

1.3. Notation

For a set $S \subseteq \mathbb{R}^n$, we use conv(S) to denote its convex hull. For a set $S \subseteq \{(x, y) : x \in \mathbb{R}^n, y \in \mathbb{R}^m\}$, we use proj$_x(S)$ to denote its projection onto the n–dimensional subspace $H = \{(x, y) : y = 0\}$. We use e_i to denote the n–dimensional ith unit vector.

2. Optimization over conv$(\mathcal{R})$

First we observe that optimization of a linear function over conv$(\mathcal{R})$ is easy. Let the problem be defined as

$$\varphi = \max \left\{ a'x + b'y + hz : (x, y, z) \in \text{conv}(\mathcal{R}) \right\}. \tag{6}$$

Without loss of generality $b, h \leq 0$; otherwise, problem (6) is unbounded. Then there is an optimal solution (x, y, z) such that $x_i = 1$ and $y_i = (d_i - z)^+$ if $a_i + b_i(d_i - z)^+ \geq 0$, and $x_i = y_i = 0$ otherwise. Therefore,

$$\varphi(z) = hz + \sum_{i=1}^{n} \left(a_i + b_i(d_i - z)^+\right)^+ . \tag{7}$$

Since z takes at most $n+1$ discrete values $d_0, d_1, \ldots, d_n$ in extreme points of $\text{conv}(\mathcal{R})$, it follows that $\varphi = \max_{k \in [0,n]} \varphi(d_k)$. Observe that $\varphi(d_k)$ for all $k \in [0, n]$ can be computed in linear time after sorting d_k. Hence optimization problem (6) can be solved in $O(n \log n)$. Polynomial equivalence of optimization and separation for polyhedra [17] suggests efficient separation of $\text{conv}(\mathcal{R})$.

3. Strong formulations

3.1. A disjunctive formulation

The first strong formulation is based on the observation that variable z takes at most $n+1$ distinct values in extreme points of $\text{conv}(\mathcal{R})$. We form the convex hull of the restrictions of $\mathcal{R}$ for these values and then write the convex hull of their disjunction. To this end, let $\mathcal{R}(\delta) = \{(x, y, z) \in \mathcal{R} : z = \delta\}$ for a fixed $\delta \geq 0$, that is,

$$\mathcal{R}(\delta) = \left\{ (x, y, z) \in \{0, 1\}^n \times \mathbb{R}^n_+ \times \delta : d_i x_i \leq y_i + \delta,\ i \in [1, n] \right\}.$$

Since the constraints of $\mathcal{R}(\delta)$ are decoupled, strengthening each constraint independently, one sees that the convex hull of $\mathcal{R}(\delta)$ can be written as

$$\text{conv}(\mathcal{R}(\delta)) = \left\{ (x, y, z) \in \mathbb{R}^{2n} \times \delta : (d_i - \delta)^+ x_i \leq y_i,\ i \in [1, n],\ \mathbf{0} \leq x \leq \mathbf{1} \right\}.$$

On the other hand, since $z \in \{d_0, d_1, \ldots, d_n\}$ in extreme points of $\text{conv}(\mathcal{R})$,

$$\text{conv}(\mathcal{R}) = \text{conv}\left(\cup_{k=0}^{n} \text{ conv}(\mathcal{R}(d_k))\right) + \mathcal{K},$$

where

$$\mathcal{K} = \{ (x, y, z) \in \mathbb{R}^{2n+1} : x = \mathbf{0},\ y \geq \mathbf{0},\ z \geq 0 \}$$

is the recession cone of the LP relaxation of $\mathcal{R}$. Then, from disjunctive programming [3], it follows that

$$\mathcal{D} = \left\{ \begin{array}{l} (x, y, z, \omega, \lambda) \in \mathbb{R}^{n^2+4n+2} : \\ \mathbf{1}'\lambda = 1 \\ 0 \leq \omega_i^k \leq \lambda_k, \quad i \in [1, n], \quad k \in [0, n] \\ z \geq \sum_{k=0}^{n} d_k \lambda_k \\ y_i \geq \sum_{k=0}^{n} (d_i - d_k)^+ \omega_i^k, \quad i \in [1, n] \\ x = \sum_{k=0}^{n} \omega^k \end{array} \right\}$$

has integral λ for all extreme points of $\mathcal{D}$ and we have $\text{conv}(\mathcal{R}) = \text{proj}_{x,y,z}(\mathcal{D})$. A similar disjunctive formulation is given in [20] for a set with general integer variables. Due to the implicit convexification of $\mathcal{R}$ given by $\mathcal{D}$, the formulation

$$\text{(RP1)} \qquad \min\left\{ a'x + b'y + hz : \ x \in \mathcal{F}, \ (x, y, z, \omega, \lambda) \in \mathcal{D} \right\}$$

is stronger than RP.

3.2. Formulation in the original space

Even though polynomial in size, RP1 has a quadratic number of additional variables and constraints for modeling robustness. Therefore, it may be preferable to give an explicit description of $\text{conv}(\mathcal{R})$ using only the original variables x, y, and z. The next theorem describes valid inequalities for $\text{conv}(\mathcal{R})$ in the original space of variables.

Theorem 1. *For any* $T = \{i_1, i_2, \dots, i_t\} \subseteq [1, n]$ *with* $0 = d_{i_0} \leq \dots \leq d_{i_t}$,

$$\sum_{i_j \in T} (d_{i_j} - d_{i_{j-1}}) x_{i_j} \leq z + \sum_{i \in T} y_i \tag{8}$$

is a valid inequality for $\mathcal{R}$. *Furthermore,* (8) *defines a facet of* $\text{conv}(\mathcal{R})$ *if and only if* $0 < d_{i_1} < \dots < d_{i_t}$.

Proof. Let $(x, y, z) \in \mathcal{R}$ and $k = \max\{j \in [0, t] : d_{i_j} \leq z\}$. Then

$$\begin{aligned}\sum_{j \in [1,t]} (d_{i_j} - d_{i_{j-1}}) x_{i_j} &\leq d_{i_k} + \sum_{j \in [k+1,t]} (d_{i_j} - d_{i_{j-1}}) x_{i_j} \\ &\leq z + \sum_{j \in [k+1,t]} (d_{i_j} - z) x_{i_j} \leq z + \sum_{i \in T} y_i,\end{aligned}$$

where the last inequality follows from $y \geq 0$ and $y_{i_j} \geq (d_{i_j} - z) x_{i_j}$ for $j \in [k+1, t]$ since $x_{i_j} \in \{0, 1\}$.

For the second part of the theorem, observe that $\text{conv}(\mathcal{R})$ is full–dimensional. Since, if $d_i > 0$ and distinct for $i \in T$, the points $(\mathbf{0}, \mathbf{0}, 0)$; $(e_i, d_i e_i, 0)$, $(e_i, (d_i + \epsilon) e_i, 0)$ for $i \in [1, n] \setminus T$; and $(\sum_{j=1}^k e_{i_j}, \mathbf{0}, d_{i_k})$, $(\sum_{j=1}^k e_{i_j}, \epsilon e_{i_k}, d_{i_k} - \epsilon)$ for $k \in [1, t]$, where $\epsilon > 0$ small, are affinely independent points of $\text{conv}(\mathcal{R})$ on the face defined by (8), inequality is facet–defining. Conversely, if $d_{i_j} = d_{i_{j-1}}$ for some $j \in [1, t]$, then the inequality defined by T is implied by the one defined by $T \setminus \{i_j\}$ and $y_{i_j} \geq 0$. □

The next theorems shows that the bounds on the variables and inequalities (8) are sufficient to describe $\text{conv}(\mathcal{R})$ explicitly.

Theorem 2. *The convex hull of* $\mathcal{R}$ *can be stated as*

$$\text{conv}(\mathcal{R}) = \{ (x, y, z) \in \mathbb{R}^{2n+1} : \mathbf{0} \leq x \leq \mathbf{1}, \ y \geq \mathbf{0}, \ \textit{and inequalities } (8) \}.$$

Proof. We will show that any proper face of conv($\mathcal{R}$) is defined by an inequality among $x \geq \mathbf{0}$, $x \leq \mathbf{1}$, $y \geq \mathbf{0}$, and (8). This implies that all facet–defining inequalities are included among this list of inequalities. Consider an arbitrary objective (a, b, h) for the optimization problem $\max\{a'x + b'y + hz : (x, y, z) \in \text{conv}(\mathcal{R})\}$ such that the set of optimal solutions is a proper face of conv($\mathcal{R}$), i.e., $b, h \leq 0$ and $(a, b, h) \neq (\mathbf{0}, \mathbf{0}, 0)$.

If $a_i < 0$, for some $i \in [1, n]$, then $x_i = 0$ in all optimal solutions, i.e., inequality $x_i \geq 0$ defines the optimal face; and we are done. So for the rest of the proof we may assume that $a \geq 0$. Suppose that $h = 0$; if $a_i > 0$ for some $i \in [1, n]$, then $x_i = 1$ in all optimal solutions; otherwise, $a_i = 0$ for all $i \in [1, n]$ and there exists some $b_k < 0$, which implies that $y_k = 0$ in all optimal solutions. So for the rest of the proof we may assume that $h < 0$ and $a \geq 0$.

If $z = 0$ for all optimal solutions, then the optimal face is defined by inequality (8) with $T = \emptyset$. Otherwise, since $h < 0$ and $z \in \{d_0, d_1, \dots, d_n\}$ for extreme points of conv($\mathcal{R}$), the maximum value of z among all optimum solutions equals $d_k > 0$ for some $k \in [1, n]$. Then $a_k > 0$ and $b_k < 0$, since otherwise, as $h < 0$, the objective is improved by reducing z. Define $N_1 = \{i \in [0, k] : a_i > 0 \text{ and } b_i < 0\}$, $p(i) = \min\{j \in N_1 : a_i + b_i(d_i - d_j) \geq 0\}$, and $r(i) = \max\{j \in N_1 : j < i\}$ for $i \in N_1 \setminus \{0\}$, and consider the set T defined recursively as

$$T := \emptyset; \text{ while } (k > 0) \{ T := T \cup \{k\}; \text{ if } k = p(k), k := r(k); \text{ else } k := p(k); \}.$$

Let the elements of $T = \{i_1, i_2, \dots, i_t\}$ be indexed such that $i_{j-1} = p(i_j)$ if $p(i_j) < i_j$ and $i_{j-1} = r(i_j)$ if $p(i_j) = i_j$ for $j \in [2, t]$.

Claim. $\sum_{j=1}^{t}(d_{i_j} - d_{i_{j-1}})x_{i_j} = z + \sum_{i \in T} y_i$ holds for all optimal solutions.

Consider an optimal solution (x, y, z). Suppose that $d_{i_{j-1}} \leq z < d_{i_j}$ for $j \in [1, t]$. Then $x_{i_\ell} = 1$ for $\ell \in [1, j-1]$ since $a_{i_\ell} > 0$ and $x_{i_\ell} = 0$ for $\ell \in [j+1, t]$ since $p(i_\ell) \geq i_j$ and $a_{i_\ell} + b_{i_\ell}(d_{i_\ell} - z) < 0$.

$$lhs = \sum_{j=1}^{t}(d_{i_j} - d_{i_{j-1}})x_{i_j} = d_{i_{j-1}} + (d_{i_j} - d_{i_{j-1}})x_{i_j}$$

$$rhs = z + \sum_{i \in T} y_i = z + (d_{i_j}x_{i_j} - z)^+.$$

If $z = d_{i_{j-1}}$, $lhs = rhs$ for $x_{i_j} \in \{0, 1\}$. Else ($d_{i_{j-1}} < z < d_{i_j}$), we need to consider two cases. If $p(i_j) = i_{j-1}$, then since $b_{i_j} < 0$, we have $a_{i_j} + b_{i_j}(d_{i_j} - z) > 0$ and consequently, $x_{i_j} = 1$ must hold, which implies $lhs = rhs$. On the other hand, if $p(i_j) = i_j$, then since $i_{j-1} = r(i_j)$, there exists no $i \in N_1$ such that $d_{i_{j-1}} < d_i < d_{i_j}$, hence there is no optimal solution with $d_{i_{j-1}} < z < d_{i_j}$.

Finally if $z = d_{i_t}$, equality holds as $lhs = d_{i_t}$ and $\sum_{i \in T} y_i = 0$ since $a_i > 0$ and $b_i < 0$ for all $i \in T$. □

Then by Theorem 2, using inequalities (8) for describing conv($\mathcal{R}$) explicitly, we can write an alternative strong formulation for RP as

$$\text{(RP2)} \qquad \min\left\{ a'x + b'y + hz : x \in \mathcal{F}, (x, y, z) \in \text{conv}(\mathcal{R}) \right\}.$$

Separation

Even though $\text{conv}(\mathcal{R})$ has up to 2^n facets defined by inequalities (8), they can be checked for violation with a polynomial algorithm: Let G be a directed graph with $n+2$ vertices labelled from 0 to $n+1$; and let (i, j) be an arc in G if and only if $0 \leq i < j \leq n+1$. There is a one–to–one correspondence between inequalities (8) and the paths from vertex 0 to vertex $n+1$ in G; that is, $j \in T$ if and only if j is contained in a 0–$(n+1)$ path (Figure 1). Given a point $(x, y, z) \in \mathbb{R}_+^{2n+1}$, let the length of arc (i, j) be $y_j - (d_j - d_i)x_j$ if $j \in [1, n]$ and z if $j = n+1$, and ζ be the length of a shortest 0–$(n+1)$ path. Then there exists an inequality (8) violated by (x, y, z) if and only if $\zeta < 0$, which can be checked in $O(n^2)$ by finding a shortest path on this acyclic network.

Theorem 3. *The separation problem for inequalities* (8) *is solved in* $O(n^2)$.

3.3. An extended formulation

Given a point $(x, y, z) \in \mathbb{R}_+^{2n+1}$, the separation problem for inequalities (8) can be formulated as the linear program

$$
\begin{aligned}
\zeta = \min \sum_{1 \leq i < j \leq n} (y_j - (d_j - d_i)x_j) f_{ij} + \sum_{i=1}^{n} z f_{in+1} \\
\text{s.t.} \sum_{0 \leq i < j} f_{ij} - \sum_{n+1 \geq i > j} f_{ji} = \begin{cases} 0 & \text{if } j \in [1, n] \\ -1 & \text{if } j = 0 \\ +1 & \text{if } j = n+1 \end{cases} \\
f \geq \mathbf{0}
\end{aligned}
\tag{9}
$$

using flow variables f. Introducing dual variables $\nu_j,\ j \in [0, n+1]$ for the flow balance constraints (9), we can write the constraints of the dual problem as

$$
\begin{aligned}
(d_j - d_i)x_j + \nu_j - \nu_i &\leq y_j, \quad 0 \leq i < j \leq n \\
\nu_{n+1} - \nu_i &\leq z, \quad 0 \leq i \leq n.
\end{aligned}
$$

By strong duality, the objective of the dual problem $\nu_{n+1} - \nu_0 \geq 0$ if and only if $\zeta \geq 0$, i.e., $(x, y, z) \in \mathbb{R}_+^{2n+1}$ is satisfied by all inequalities (8). Hence, by construction,

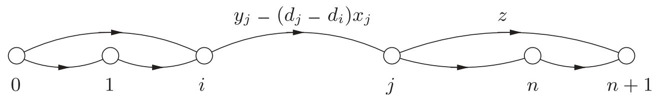

Fig. 1. Separation for inequalities (8)

$$\text{conv}(\mathcal{R}) = \text{proj}_{x,y,z}(\mathcal{Q}), \text{ where}$$

$$\mathcal{Q} = \left\{ (x, y, z, \nu) \in \mathbb{R}^{3n+3} : \begin{array}{rl} (d_j - d_i)x_j + \nu_j - \nu_i \le y_j, & 0 \le i < j \le n \\ \nu_{n+1} - \nu_i \le z, & 0 \le i \le n \\ \nu_{n+1} - \nu_0 \ge 0 & \\ y \ge \mathbf{0} & \\ \mathbf{0} \le x \le \mathbf{1} & \end{array} \right\}.$$

Consequently we have the third alternative formulation for RP:

$$\text{(RP3)} \qquad \min \left\{ a'x + b'y + hz : \ x \in \mathcal{F}, \ (x, y, z, \nu) \in \mathcal{Q} \right\}.$$

The advantage of RP3 is that it has the smallest size among the strong formulations RP1–RP3.

3.4. *Incorporating the nominal problem*

The three formulations RP1–RP3 are the strongest formulations for (3) that are independent from the nominal problem (1). Now we show how to incorporate the constraints of the nominal problem to obtain an even stronger formulation for (3). Let

$$\mathcal{F}_{LP} = \{ x \in \mathbb{R}^n : Ax \le t, \ \mathbf{0} \le x \le \mathbf{1} \}$$

denote an LP relaxation of $\mathcal{F}$, where A is a rational matrix and t is a rational column vector, *possibly with number of rows exponential in n*.

For a fixed $\delta \ge 0$, let $\mathcal{P}(\delta) = \mathcal{F}_{LP} \cap \text{conv}(\mathcal{R}(\delta))$; that is,

$$\mathcal{P}(\delta) = \left\{ (x, y, z) \in \mathbb{R}^{2n} \times \delta : Ax \le t, \mathbf{0} \le x \le \mathbf{1}, (d_i - \delta)^+ x_i \le y_i, i \in [1, n] \right\}.$$

Observe that if $\mathcal{F}_{LP}$ is integral, then so is $\mathcal{P}(\delta)$. Therefore, since the convex hull of union of integral polyhedra is integral, if $\mathcal{F}_{LP}$ is integral, then so is the polyhedron

$$\mathcal{P} = \text{conv}(\cup_{k=0}^{n} \mathcal{P}(d_k)) + \mathcal{K}.$$

Since $x \in \{0, 1\}^n$, we have $z \in \{d_0, \ldots, d_n\}$ for an extreme point of $\text{conv}(\mathcal{S})$. Then, if $\mathcal{F}_{LP}$ is integral, we have $\mathcal{P} = \text{conv}(\mathcal{S})$. Writing the robust optimization problem using $\mathcal{P}$, we obtain the fourth alternative formulation

$$\text{(RP4)} \quad \begin{aligned} \min \ & \sum_{k=0}^{n} a'\omega^k + b'y + hz && (10) \\ & \mathbf{1}'\lambda = 1 && (11) \\ & A\omega^k \le \lambda_k t, \quad k \in [0, n] && (12) \\ & \mathbf{0} \le \omega^k \le \lambda_k \mathbf{1}, \quad k \in [0, n] && (13) \\ & z \ge d'\lambda && (14) \\ & y_i \ge \sum_{k=0}^{n} (d_i - d_k)^+ \omega_i^k, \quad i \in [1, n] && (15) \\ & \lambda \in \mathbb{R}^{n+1}, \ \omega^k \in \{0, 1\}^n, \ k \in [0, n]. && (16) \end{aligned}$$

The preceding discussion leads to the following theorem.

Theorem 4. *If the LP relaxation $\mathcal{F}_{LP}$ of the nominal problem* (1) *is integral, then the linear program* (10)–(15) *solves the robust 0–1 problem* (3).

Theorem 4 establishes that there is a tight LP formulation for a robust 0–1 problem, with size polynomial in the size of a tight LP formulation of the nominal 0–1 problem. In particular, it describes tight LP formulations for robust counterparts of combinatorial optimization problems such as the shortest path problem, the spanning tree problem, the matching problem, for which explicit tight LP formulations are known. The following theorem implies polynomial solvability of RP4 for these problems.

Theorem 5. *If there is a polynomial separation oracle for $\mathcal{F}_{LP}$, then there is a polynomial algorithm for solving the linear program* (10)–(15).

Proof. Consider the separation problem for (10)–(15). Given a point $(x, y, z, \omega, \lambda)$ with $\mathbf{0} \leq \lambda \leq \mathbf{1}$, if $\lambda_k = 0$, then constraints (12) and (13) are trivially satisfied; otherwise, the separation oracle for $\mathcal{F}_{LP}$ can be used for constraints (12) and (13) with input $\tilde{\omega}^k = \omega^k/\lambda_k$ since $A\omega^k \leq \lambda_k t$ and $\mathbf{0} \leq \omega^k \leq \lambda_k \mathbf{1}$ if and only if $\tilde{\omega}^k \in \mathcal{F}_{LP}$. The remaining $n+2$ constraints (11), (14), and (15) can be checked by substitution. Then the result follows from polynomial equivalence of separation and optimization for polyhedra [17]. □

If $\mathcal{F}_{LP}$ is an integral polytope, the formulation RP4 may be viewed as combining the $n+1$ subproblems that need to be solved in the Bertsimas and Sim approach into a single optimization problem.

The proof of Theorem 4 is polyhedral and is based on the fact that convex hull of integral polyhedra is integral. We can generalize Theorem 4 to consider nonintegral $\mathcal{F}_{LP}$ (or $\mathcal{P}(\delta)$). The next theorem establishes that there is an α–tight linear programming formulation with size polynomial in the size of an α–tight linear programming formulation for the nominal 0–1 problem.

Theorem 6. *Let ξ_{IP} and ξ_{LP} be the optimal objective values for the nominal problem* (1) *and its LP relaxation. Similarly, let $\xi_{IP}(RP4)$ and $\xi_{LP}(RP4)$ denote the optimal objective values for formulation RP4 and its LP relaxation. If $1 \leq \xi_{IP}/\xi_{LP} \leq \alpha$ hold for any objective, then $\xi_{IP}(RP4)/\xi_{LP}(RP4) \leq \alpha$.*

Proof. Suppose $\xi_{LP}(RP4)$ is attained over $\mathcal{P}(d_k)$. Define $\xi_k = hd_k + \min\left\{\sum_{i=1}^n (a_i + b_i(d_i - d_k)^+)x_i : x \in \mathcal{F}_{LP}\right\}$ and $\xi_k' = hd_k + \min\left\{\sum_{i=1}^n (a_i + b_i(d_i - d_k)^+)x_i : x \in \mathcal{F}\right\}$. Then we have

$$\xi_{LP}(RP4) = \xi_k \leq \xi_{IP}(RP4) \leq \xi_k' \leq \alpha \xi_k,$$

where the last inequality follows from $\xi_k' - hd_k \leq \alpha(\xi_k - hd_k)$ (the assumption on the integrality gap of the LP relaxation of the nominal problem) and $\alpha \geq 1$. □

We conclude this section by summarizing the results on the strength of the LP relaxation of the formulations in the next theorem.

Theorem 7. *Let $\xi_{LP}(A)$ denote the optimal objective value of the LP relaxation of formulation A. Then $\xi_{LP}(RP) \leq \xi_{LP}(RP1) = \xi_{LP}(RP2) = \xi_{LP}(RP3) \leq \xi_{LP}(RP4)$.*

4. Robust mixed 0–1 programming

In this section we will give strong formulations for robust mixed 0–1 programming. Let the nominal problem be a mixed 0–1 programming problem with $x_i \in \{0, 1\}$ for $i \in B$ and $x_i \in \mathbb{R}_+$ for $i \in C$, where (B, C) is a partitioning of $[1, n]$. Let $C_1 \subseteq C$ be the index set of continuous variables with finite upper bound, which can be assumed to be one by scaling such variables if necessary. So the mixed 0–1 feasible set is

$$\mathcal{F}_M = \{ \, x \in \mathbb{R}^n_+ : Ax \le t, \ x_i \in \{0, 1\}, \ i \in B, \ x_i \le 1, \ i \in C_1 \, \}.$$

Consequently, the robust counterpart is written as

$$\text{(RPM)} \qquad \min \left\{ \, a'x + b'y + hz \ : \ (x, y, z) \in \mathcal{S}_M \, \right\},$$

where

$$\mathcal{S}_M = \{ \, (x, y, z) \ : \ x \in \mathcal{F}_M, \ (x, y, z) \in \mathcal{R}_M \, \}$$

and

$$\mathcal{R}_M = \left\{ (x, y, z) \in \mathbb{R}^{2n+1}_+ : \begin{array}{ll} y_i + z \ge d_i x_i, & i \in [1, n] \\ x_i \in \{0, 1\}, & i \in B \\ x_i \le 1, & i \in C_1 \end{array} \right\}.$$

First, observe that z equals d_0 or d_i, $i \in B \cup C_1$ in extreme points of $\text{conv}(\mathcal{R}_M)$. Then, since for $\mathcal{R}_M(\delta) = \{(x, y, z) \in \mathcal{R}_M : z = \delta\}$ and $\delta \ge 0$ we have

$$\text{conv}(\mathcal{R}_M(\delta)) = \left\{ (x, y, z) \in \mathbb{R}^{2n}_+ \times \delta : \begin{array}{rl} (d_i - \delta)^+ x_i \le y_i, & i \in B, \\ d_i x_i - \delta \le y_i, & i \in C, \\ x_i \le 1, & i \in B \cup C_1 \end{array} \right\},$$

generalizing RP1, we give a disjunctive formulation for robust mixed 0–1 programming as follows:

$$\text{(RPM1)} \quad \min \left\{ \, a'x + b'y + hz : x \in \mathcal{F}_M, \ (x, y, z, w, \lambda) \in \mathcal{D}_M \, \right\},$$

where

$$\mathcal{D}_M = \left\{ \begin{array}{l} (x, y, z, \omega, \lambda) : \sum_{k \in \{0\} \cup B \cup C_1} \lambda_k = 1 \\ 0 \le \omega_i^k \le \lambda_k, \ \ i \in B \cup C_1 \\ z \ge \sum_{k \in \{0\} \cup B \cup C_1} d_k \lambda_k \\ y_i \ge \sum_{k \in \{0\} \cup B \cup C_1} (d_i - d_k)^+ \omega_i^k, \ \ i \in B \\ y_i \ge \sum_{k \in \{0\} \cup B \cup C_1} (d_i \omega_i^k - d_k \lambda_k)^+, \ \ i \in C \\ x = \sum_{k \in \{0\} \cup B \cup C_1} \omega^k \end{array} \right\}.$$

The nonlinear terms $(d_i \omega_i^k - d_k \lambda_k)^+$ are due to the fact that $y_i \ge 0$, $i \in C$ are necessary to describe $\text{conv}(\mathcal{R}_M(d_k))$. These terms can be linearized by introducing auxiliary variables γ_i^k such that $\gamma_i^k \ge d_i \omega_i^k - d_k \lambda_k$ and $\gamma_i^k \ge 0$.

$\mathcal{R}_M$ is a relaxation of $\mathcal{R}$, in which binary and upper bound constraints on some of the variables are dropped. Consequently, not all valid inequalities for $\mathcal{R}$ are valid for its relaxation $\mathcal{R}_M$. The following theorem characterizes the subset of the facet–defining inequalities (8) that are valid for $\mathcal{R}_M$.

Theorem 8. *An inequality* (8) *with* $T = \{i_1, i_2, \ldots, i_t\} \subseteq [1, n]$ *and* $0 < d_{i_1} < d_{i_2} < \cdots < d_{i_t}$ *is valid for* $\mathcal{R}_M$ *if and only if the following conditions hold:*

1. *if* $i \in C_1 \cap T$*, then* $i = i_1$*;*
2. *if* $i \in (C \setminus C_1) \cap T$*, then* $T = \{i\}$.

Proof. If $C \cap T = \emptyset$, validity holds without any conditions by Theorem 1.
Necessity. 1. Suppose $i_k \in C_1$ and $k \neq 1$. Then, inequality (8) is violated by the solution with $x_{i_1} = 1, x_{i_k} = d_{i_1}/d_{i_k}, z = d_{i_1}$ and all other variables zero. 2. Suppose $i_k \in C \setminus C_1$. If $k \neq 1$, previous solution violates (8). If $k = 1$ and $\{i_1\} \subsetneq T$. Then inequality (8) is violated by the solution with $x_{i_1} = d_{i_2}/d_{i_1}, x_{i_2} = 1, z = d_{i_2}$ and all other variables zero.
Sufficiency. 1. Suppose $i_1 \in C_1$. Since $d_{i_1}x_{i_1} \leq d_{i_1}$, validity argument of Theorem 1 remains correct if $k = \max\{j \in [0, t] : d_{i_j} \leq z\} > 0$. Otherwise, we need a slight change in the argument:

$$\sum_{j\in[1,t]} (d_{i_j} - d_{i_{j-1}})x_{i_j} \leq d_{i_1}x_{i_1} + \sum_{j\in[2,t]} (d_{i_j} - d_{i_{j-1}})x_{i_j}$$

$$\leq z + (d_{i_1}x_{i_1} - z) + \sum_{j\in[2,t]} (d_{i_j} - z)x_{i_j} \leq z + \sum_{i\in T} y_i,$$

where the last inequality follows from $y_{i_1} \geq d_{i_1}x_{i_1} - z$ and $y_i \geq (d_i - z)x_i$ for all $i \in T \setminus \{i_1\}$ as by the necessity of condition 2, $T \setminus \{i_1\} \subseteq B$. 2. Validity with $T = \{i\}$ simply follows from $y_i + z \geq d_i x_i$. □

Sufficiency of the inequalities in Theorem 8 and the bounds on the variables for describing $\text{conv}(\mathcal{R}_M)$ is shown using the same steps as in the proof of Theorem 2. This leads to a formulation by an explicit listing of the constraints of $\text{conv}(\mathcal{R}_M)$:

$$\text{(RPM2)} \quad \min\left\{ a'x + b'y + hz : x \in \mathcal{F}_M,\ (x, y, z) \in \text{conv}(\mathcal{R}_M) \right\}.$$

Separation for the inequalities satisfying the conditions in Theorem 8 can be done by solving the shortest path problem defined in Section 3.2 after dropping from G the arcs

$$E_1 = \{\ (h, i) \text{ for } h \in [1, i-1] \text{ and } i \in C\} \text{ and}$$
$$E_2 = \{\ (i, j) \text{ for } j \in [i+1, n] \text{ and } i \in C \setminus C_1\}$$

in order to avoid inequalities (8) that are invalid for the description of $\text{conv}(\mathcal{R}_M)$. Letting $E = \{(i, j) : 0 \leq i < j \leq n\} \setminus (E_1 \cup E_2)$, we can then write an extended formulation for the robust mixed 0–1 programming as

$$\text{(RPM3)} \qquad \min\left\{ a'x + b'y + hz :\ x \in \mathcal{F}_M,\ (x, y, z, \nu) \in \mathcal{Q}_M \right\},$$

where

$$\mathcal{Q}_M = \left\{ (x, y, z, \nu) \in \mathbb{R}^{3n+3} : \begin{array}{r} (d_j - d_i)x_j + \nu_j - \nu_i \leq y_j, \quad (i, j) \in E \\ \nu_{n+1} - \nu_i \leq z, \quad 0 \leq i \leq n \\ \nu_{n+1} - \nu_0 \geq 0 \\ x_i \leq 1, \quad i \in B \cup C_1 \\ x, y \geq \mathbf{0} \end{array} \right\}.$$

Finally, we remark that z may take values different from 0 and d_i, $i \in B \cup C_1$ in extreme points of conv($\mathcal{S}_M$) if the objective coefficients of the continuous variables are uncertain, i.e., $d_i > 0$ for $i \in C$. We do not know extensions of Theorems 4–6 to mixed 0–1 programming in this case. Furthermore, if $d_i > 0$ for $i \in C$, it is also an open question whether the robust counterpart RPM is polynomially solvable if the nominal mixed 0–1 problem is polynomially solvable. On the other hand, for the case $d_i = 0$ for all $i \in C$, since continuous variables do not appear in the inner maximization problem (see (5)), Theorems 4–6 extend to the mixed 0–1 case trivially.

5. Computational experience

In this section we compare the computational difficulty of solving the formulations introduced in the paper. The experiments are performed using the MIP solver of CPLEX[1] Version 9.0 on a 3 MHz Intel Pentium4/RedHat Linux workstation.

The experiments are restricted to the case (2) introduced in [8] by letting $b = \mathbf{1}$ and $h = r$. The first experiment is on solving the optimization problem (6) over $\mathcal{R}$ using formulations RP, RP1, RP2, and RP3. The purpose of this experiment is to observe the behavior of the formulations without any side constraints. To this end, random instances with 200 variables with a_i drawn from integer uniform[1,1000] and d_i from integer uniform[1,2000] are generated. In order to see the effect of the parameter r controlling the conservatism of the solution, the instances are solved for varying values of r as shown in Table 1. Each entry in this table is an average for four instances. The columns under heading RP show the integrality gap of the LP relaxation of the original formulation RP, and the number of nodes and CPU time (seconds) taken by default CPLEX for solving the instances. Interestingly, the instances have small integrality gap $((\xi_{LP} - \xi_{IP})/\xi_{LP}$, where ξ_{LP} is the optimal value of the LP relaxation and ξ_{IP} is the optimal objective value) for small and large values of r, and are solved easily. This observation is intuitive since for extremely small or extremely large values for r, z takes value either d_0 or d_n in optimal solutions, which prevents x from being fractional in the extreme points of the LP relaxation. However, for values of r between 40 and 60, the LP relaxation is highly fractional and integrality gap is large. Consequently, considerable branching is needed for these instances. Since in this experiment $\mathcal{F} = \{0, 1\}^n$, the LP relaxations of RP1–RP3 are integral and no branching is required for these formulations. The solution time for the disjunctive formulation RP1 is unaffected by the choice of r. On the other hand, there is a high correlation between the integrality gap of RP and the solution time of the cut formulation RP2. The higher the gap, the larger is the number of cuts added to close it. In Table 1 we present results for two implementations of the separation algorithm in Section 3.2, denoted as RP2 and RP2′. In RP2, in each cut generation phase, a shortest path from vertex 0 to every other vertex i is computed and the corresponding inequality is added if violated by the LP solution. This allows us to add multiple cuts in each cut generation phase, and performs much better than adding a single most violated cut given by a shortest 0–$(n + 1)$ path in every phase. On the other hand, in RP2′, we look for a violated cut that corresponds to a 0–i path with the smallest number of arcs using the Bellman-Ford algorithm. Since the arcs correspond to variables in inequality (8), this

[1] CPLEX is a trademark of ILOG, Inc.

choice keeps the cuts sparse. As seen in Table 1, RP2′ outperforms RP2 by a significant margin. Even though separation in RP2′ takes longer (still polynomial), LPs with sparse cuts are solved much faster, which leads to significant saving in the overall computation time. Finally, the extended formulation RP3 is solved much more easily—only in a few seconds—for all instances. Recall that RP3 has the smallest size among RP1–RP3. Since RP3 has many more constraints than variables, a faster implementation is obtained by solving its dual, shown under column RP3′.

In the second experiment, we solve instances of the robust counterpart of the $\mathcal{NP}$–hard 0–1 knapsack problem. Thus in this case $\mathcal{F} = \{x \in \{0, 1\}^n : \mu' x \leq \mu_o\}$. To this end, we add to the instances used in the first experiment knapsack constraints $\mu' x \leq \mu_o$, with μ_i drawn from integer uniform[1,100] and $\mu_o = \frac{1}{2}\sum_{i=1}^{n} \mu_i$. In Table 2 we summarize the results for RP, RP1, RP2′, RP3′, as well as RP4. The gap columns show the integrality gap of the LP relaxation of the respective formulations. RP1, RP2′, RP3′ have almost the same integrality gap. Slight differences (not shown in table) are due to different cuts added by CPLEX. With the original formulation RP, the instances with high integrality gap require an excessive number of branching and could not be solved within the memory limit of 100 MB. Under the column egap[uslv] we show the average optimality gap, percentage gap between the best known upper bound and lower bound at termination

Table 1. Optimization over $\mathcal{R}$.

r	RP			RP1	RP2		RP2′		RP3	RP3′
	gap	node	time	time	cuts	time	cuts	time	time	time
10	2.9	2	0	27	1178	4	945	0	1	1
20	6.8	778	1	24	3472	26	2439	1	2	0
30	16.0	19244	26	28	7075	105	5192	4	2	0
40	30.9	244594	420	33	11806	346	9348	36	3	1
50	50.1	386932	576	37	18623	1462	19038	545	7	1
60	48.5	82082	70	35	14284	624	14864	50	4	1
70	36.8	764	1	36	9642	202	8548	3	3	1
80	24.6	0	0	36	6545	64	5657	2	2	1
90	13.1	0	0	36	4059	19	3560	0	2	1
100	4.3	0	0	36	1730	4	1252	0	2	1
vars		401		40802		401			603	20703
cons		400		40802		400+cuts			20703	603

Table 2. Experiments with robust 0–1 knapsack.

r	RP			RP1			RP2′			RP3′		RP4		
	gap	node	egap[uslv]	gap	node	time	cuts	node	time	node	time	gap	node	time
10	4.4	152	0.0[0]	0.0	33	52	1228	28	1	15	8	0.0	62	696
20	15.5	211172	1.0[2]	0.1	144	237	5733	93	63	67	54	0.1	150	2906
30	28.7	313276	9.7[4]	0.4	52	204	12469	235	623	73	102	0.1	147	1867
40	43.8	311286	27.0[4]	4.4	398	1189	179588	1396	6686	733	617	0.1	318	2327
50	53.0	312750	33.1[4]	3.2	161	690	21482	528	4982	206	278	0.1	36	201
60	48.5	336007	27.8[4]	0.0	0	33	8198	0	17	0	1	0.0	0	27
70	36.6	179474	8.1[2]	0.0	0	33	6558	0	3	0	1	0.0	0	27
80	24.6	707	0[0]	0.0	0	33	5047	0	1	0	1	0.0	0	27
90	13.1	0	0[0]	0.0	0	33	3560	0	0	0	1	0.0	0	27
100	4.3	0	0[0]	0.0	0	33	1215	0	0	0	1	0.0	0	27
vars		401			40802			401		20703			40602	
cons		401			40803			401+cuts		603			40603	

Table 3. Experiments with robust mixed 0–1 knapsack.

r	RPM			RPM1			RPM2′			RPM3′	
	gap	node	ogap[uslv]	gap	node	time	cuts	node	time	node	time
10	12.5	13138	0.0[0]	0.0	0	63	2972	1	1	0	2
20	4.0	500	0.0[0]	0.0	4	186	1157	2	2	3	5
30	10.3	147936	0.4[2]	0.0	4	145	3502	0	9	2	8
40	18.0	279087	4.1[4]	0.0	5	256	6762	0	44	0	10
50	26.0	291496	10.0[4]	0.3	2	758	10979	9	199	10	78
60	33.3	280605	16.4[4]	1.6	62	4689	11741	221	901	78	158
70	40.2	294063	22.7[4]	3.4	188*	13891*	12238	1917	9121	791	1141
80	41.5	277968	20.1[4]	1.0	3	1102	10554	22	215	13	66
90	33.9	320824	8.0[2]	0.0	0	51	6966	0	7	0	1
100	24.7	26273	0.0[0]	0.0	0	50	5820	0	1	0	1
vars		401			503352			401		20703	
cons		401			503353			401+cuts		603	

when the memory limit is reached, and the number of unsolved instances (out of four) in brackets. For r between 30 and 60, none of the instances is solved to optimality and a large optimality gap is left at termination. In contrast, all of the instances are solved to optimality with much less or no branching with the new formulations. This is predominantly due to the significantly smaller integrality gap obtained with their LP relaxations. In particular, for RP4 the integrality gap is almost zero in all instances. However, the additional reduction in the integrality gap achieved by RP4 over RP1–RP3′, does not appear to translate to faster solution times. As in the case without side constraints, solving the dual of RP3 at the nodes of the branch–and–bound tree outperforms the other choices.

The final experiment is on solving the robust mixed 0–1 knapsack problem. The instances for the experiment are generated by adding 50 bounded continuous variables to the 0–1 knapsack instances. The effect of including continuous variables is reduced integrality gap for the LP relaxations. The relative difficulty of solving the formulations is similar to the 0–1 knapsack case observed in the previous experiment. Most of the instances could not be solved with formulation RPM due to the large integrality gap and excessive branching. On the other hand, all but one of the unsolved instances could be solved to optimality using formulation RPM1 with much less branching. Computation for one instance with formulation RPM1 is terminated after 10 hours with %0.05 optimality gap (only 382 nodes are evaluated before termination). The averages 188 nodes and 13891 seconds reported for $r = 70$ includes this instance. All instances are solved to optimality much faster with the equally strong, but smaller formulations RPM2′ and RPM3′.

As in the previous experiments, solving the dual of the LP relaxation of RPM3 at the nodes of the search tree leads to the smallest overall solution time. In the current implementation, the dual of the LP relaxation of RPM3 is constructed and solved from scratch at each node. A close integration of branching on primal variables and successively solving the dual of the LP relaxation at the nodes of the search tree should speed up the computations significantly. As the need for solving MIPs with a very large number of constraints increases, we expect that such an integrated implementation that is available for branching and solving primal LPs in commercial MIP solvers, will extend to branching on primal variables and solving dual LPs. In addition, for solving large–scale robust mixed 0–1 programs with RPM3′ a column generation approach may be more practical than keeping all of the $O(n^2)$ variables in the dual formulation explicitly.

6. Conclusion

We introduce formulations for robust 0–1 programming and for robust mixed 0–1 programming with strong LP relaxations. We show that for a robust 0–1 problem, there is an α–tight linear programming formulation with size polynomial in the size of an α–tight linear programming formulation for the nominal 0–1 problem. Strong formulations are of significant interest for robust mixed 0–1 programming, for which polynomial algorithms are unknown even if the nominal mixed 0–1 problem is polynomially solvable.

Our computational experiments show that the difficulty of solving the alternative formulations differ by a large margin. Furthermore, algorithmic choices in generating cutting planes as well as in solving the linear programs affect the computational results significantly. Overall, the proposed strong formulations improve the solvability of the robust (mixed) 0–1 programs substantially.

Acknowledgements. I am grateful to Ed Rothberg of ILOG, Inc. for making the necessary changes in the CPLEX code that allowed solving the dual formulations at the nodes of the branch–and–cut tree.

References

1. Atamtürk, A., Zhang, M.: Two-stage robust network flow and design under demand uncertainty. Research Report BCOL.04.03, University of California–Berkeley, Dec 2004. Available at `http://ieor.berkeley.edu/~atamturk`
2. Averbakh, I.: On the complexity of a class for combinatorial optimization problems with uncertainty. Mathematical Programming **90**, 263–272 (2001)
3. Balas, E.: Disjunctive programming. Annals of Discrete Mathematics **5**, 3–51 (1979)
4. Ben-Tal, A., Goryashko, A., Guslitzer, E., Nemirovski, A.: Adjustable robust solutions of uncertain linear programs. Mathematical Programming **99**, 351–376 (2004)
5. Ben-Tal, A., Nemirovski, A.: Robust convex optimization. Mathematics of Operations Research **23**, 769–805 (1998)
6. Ben-Tal, A., Nemirovski, A., Roos, C.: Robust solutions to uncertain quadratic and conic-quadratic problems. SIAM Journal on Optimization **13**, 535–560 (2002)
7. Bertsimas, D., Pachamanova, D., Sim, M.: Robust linear optimization under general norms. Operations Research Letters **32**, 510–526 (2004)
8. Bertsimas, D., Sim, M.: Robust discrete optimization and network flows. Mathematical Programming **98**, 49–71 (2003)
9. Bertsimas D., Sim, M.: Robust conic optimization. Manuscript, 2004
10. Bertsimas, D., Sim, M.: Robust discrete optimization under ellipsoid uncertainty sets. Manuscript, 2004
11. Bertsimas, D., Thiele, A.: A robust optimization approach to supply chain management. In Proceedings of IPCO X Conference, volume 3064 of *Lecture Notes in Computer Science*, pages 86–100, 2004
12. Erdogan, E., Goldfarb, D., Iyengar, G.: Robust portfolio management. CORC Technical Report TR-2004-11, Columbia University, 2004
13. Erdogan, E., Iyengar, G.: Ambiguous chance constrained problems and robust optimization. CORC Technical Report TR-2004-10, Columbia University, 2004
14. Ghaoui, L.El., Lebret, H.: Robust solutions to least-squares problems with uncertain data. **18**, 1035–1064 (1997)
15. Ghaoui, L.El., Oustry, F., Lebret, H.: Robust solutions to uncertain semidefinite programs. SIAM Journal on Optimization **9**, 33–52 (1998)
16. Goldfarb, D., Iyengar, G.: Robust portfolio selection problems. Mathematics of Operations Research **28**, 1–38 (2003)

17. Grötschel, M., Lovász, L., Schrijver, A.: The ellipsoid method and its consequences in combinatorial optimization. Combinatorica **1**, 169–197 (1981)
18. Pinar, M.C., Yaman, H., Karasan, O.E.: The robust minimum spanning tree problem with interval data. Operations Research Letters **29**, 31–40 (2001)
19. Kouvelis, P., Yu, G.: Robust discrete optimization and its applications. Kluwer Academic Publishers, Norwell, MA, 1997
20. Miller A., Wolsey, L.A.: Tight formulations for some simple mixed integer programs and convex objective integer programs. Mathematical Programming **98**, 73–88 (2003)