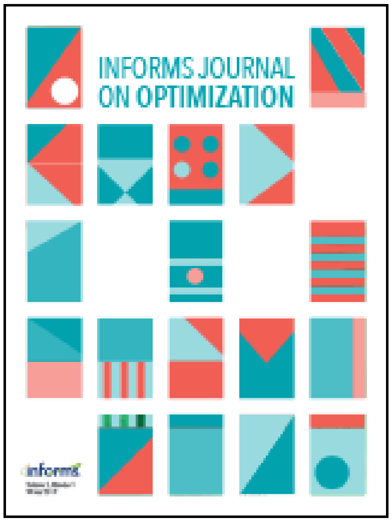


INFORMS Journal on Optimization

Robust Mixed 0-1 Programming and Submodularity

Seulgi Joung, Sungsoo Park

INFORMS JOURNAL ON OPTIMIZATION
Vol. 3, No. 2, Spring 2021, pp. 183–199
ISSN 2572-1484 (print), ISSN 2575-1492 (online)
http://pubsonline.informs.org/journal/ijoo

Robust Mixed 0-1 Programming and Submodularity

Seulgi Joung,[a] Sungsoo Park[b]

[a] Department of Industrial Engineering, Chonnam National University, Gwangju 61186, Republic of Korea; [b] Department of Industrial and Systems Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, Republic of Korea

Contact: sgjoung@jnu.ac.kr, https://orcid.org/0000-0001-7608-2266 (SJ); sspark@kaist.ac.kr, https://orcid.org/0000-0001-8226-4955 (SP)

Received: February 1, 2019
Revised: February 7, 2019
Accepted: July 16, 2020
Published Online in Articles in Advance: March 17, 2021

https://doi.org/10.1287/ijoo.2019.0042

Abstract. We study the solution set of the robust continuous 0-1 knapsack problem with a single unrestricted continuous variable. Using the submodularity of the cardinality-constrained robust 0-1 knapsack set function, we identify extended polymatroid inequalities that can be used as cutting planes in a branch-and-cut algorithm. We propose a polynomial-time separation algorithm for the most violated extended polymatroid inequality. We prove that the convex hull of the feasible solutions to the robust continuous 0-1 knapsack problem can be described completely by extended polymatroid inequalities and bound inequalities. Additionally, we propose a polynomial-time algorithm for the robust continuous 0–1 knapsack problem itself. We report computational results that show the effectiveness of the proposed extended polymatroid inequalities.

Funding: This work was supported by the National Research Foundation of Korea [Grant 2016R1A2B4013590].

Keywords: robust optimization • submodular function • valid inequalities

1. Introduction

For real-world decision-making problems, uncertainty of input data is an important issue that has been extensively studied. Stochastic optimization, which was initiated by Dantzig (1955), is a traditional approach that considers data uncertainty in mathematical programming. It assumes that random data follow a given or partially known probability distribution (see Kall et al. 1994, Shapiro et al. 2009, Birge and Louveaux 2011). However, in reality, it is not easy to obtain accurate information on the probability distribution of data. Even if we know the probability distribution of random data, it might not be easy to solve the problem considering probabilistic characteristics. Another representative approach is robust optimization (see Ben-Tal et al. 2009, Bertsimas et al. 2011, Gabrel et al. 2014). It defines an uncertainty set for uncertain data and finds a feasible solution for any realization of the data in that predefined uncertainty set. Naturally, the definition of the uncertainty set is important because robust models have different characteristics according to how the model is defined. We consider the optimization problem $\max_{x\in X}\{p^T x \mid Ax \le b\}$ with the uncertain coefficient matrix A. Soyster (1973) defined a column-wise uncertainty set, the solution for which is too conservative, which means that this model gives much worse solutions than the original deterministic model. Ben-Tal and Nemirovski (1999) proposed an uncertainty set of ellipsoidal form that gives less conservative solutions than the model of Soyster (1973). However, the robust counterpart problem of this model is a second-order cone problem, which is nonlinear. If integer restricted variables exist in the model, the robust counterpart becomes a nonlinear integer problem that can be difficult to solve computationally. Bertsimas and Sim (2004) proposed a polyhedral uncertainty set. Their model defines an interval for each uncertain coefficient and controls the maximum number of intervals that can deviate simultaneously for each constraint. This model maintains the linearity of the original deterministic problem. In this paper, we focus on the polyhedral study of robust mixed 0-1 programming problems with uncertain coefficients defined using the polyhedral uncertainty set of Bertsimas and Sim (2004).

There have been several studies on the polyhedral structure of the convex hull of feasible solutions to 0-1 knapsack problems with continuous variables that can be used for general mixed 0-1 programming problems. Marchand and Wolsey (1999) studied the following solution set of the *continuous* 0-1 *knapsack problem* with an index set of binary variables $N = \{1, \dots, n\}$ and a single nonnegative continuous variable:

$$Y = \left\{ (x, y) \in \mathbb{B}^n \times \mathbb{R}^1_+ \mid \sum_{i\in N} a_i x_i \le b + y \right\}, \tag{1}$$

where $a_i > 0$ for $i \in N$ and $b \ge 0$. There is a set of items $N = \{1, \dots, n\}$, and each item $i \in N$ has weight a_i. We have a knapsack with capacity b, and the continuous variable y can be interpreted as additional capacity of the knapsack. Marchand and Wolsey (1999) studied some classes of facet-defining valid inequalities for the convex hull of solutions based on restriction and lifting. Miller et al. (2000) defined new facet-deriving inequalities for Y and related the results to lot-sizing problems. Richard et al. (2003a, b) focused on the following solution set of the *mixed* 0-1 *knapsack problem* with an index set of binary variables N and an index set of bounded continuous variables $M = \{n+1, \dots, n+m\}$:

$$Z = \left\{ x \in \mathbb{B}^n \times [0,1]^m \mid \sum_{i \in N} a_i x_i + \sum_{i \in M} a_i x_i \le b \right\}, \tag{2}$$

where a_i for $i \in N \cup M$ and b are positive integers, and $a_i \le b$ for all $i \in N \cup M$. These authors developed lifting procedures for the bounded continuous variables. In this paper, we study the robust continuous 0-1 knapsack problem with a single continuous variable when the uncertainty set is defined as a polyhedral uncertainty set of Bertsimas and Sim (2004). Valid inequalities for the solution set of the robust continuous 0-1 knapsack problem can be used for general robust mixed 0-1 programming problems. Each constraint can be represented as a robust mixed 0-1 knapsack constraint, and continuous variables can be substituted by a single continuous variable. Then each constraint of a general robust mixed 0-1 programming problem is a robust continuous 0-1 knapsack constraint. For the robust 0-1 knapsack problem with only binary variables, Klopfenstein and Nace (2012) defined (extended) robust cover inequalities that are valid for the convex hull of feasible solutions. Kutschka (2013) defined strengthened extended robust cover inequalities and provided an integer linear program formulation for the separation problem. Joung and Park (2018) proposed a polynomial-time lifting algorithm for robust cover inequalities and several separation algorithms for (extended) robust cover inequalities. However, few studies have focused on polyhedral studies of robust mixed 0-1 programming problems. Atamtürk (2006) proposed strong formulations for robust mixed 0-1 programming problems with uncertain objective coefficients. We focus on the robust continuous 0-1 knapsack problem with binary variables and a single unrestricted continuous variable.

We define strong valid inequalities using the submodularity of the robust 0-1 knapsack set function. A submodular function is a set function that has a diminishing-returns property. Edmonds (1970) discovered the importance of submodularity in some combinatorial optimization problems. He found important results on the related polymatroids and their intersections. Little work has been done on the utilization of submodularity in the study of optimization problems considering data uncertainty. Atamtürk and Narayanan (2008) studied discrete optimization problems with a mean-risk objective function. The objective function, when restricted to problems with only binary variables, is submodular. Kutschka (2013) considered submodular robustness and proved that the robust 0-1 knapsack function is a submodular set function. Atamtürk and Bhardwaj (2018) considered a network design problem with random arc capacities. For the correlated capacities, they reformulated probabilistic capacity constraints using submodularity.

In this paper, we define strong valid inequalities for the solution set of the robust continuous 0-1 knapsack problem using submodularity, and we propose a polynomial-time separation algorithm. We prove that the convex hull of the robust continuous 0-1 knapsack solution set can be described completely by the proposed inequalities and bound inequalities. In addition, we propose a polynomial-time algorithm for the robust continuous 0-1 knapsack problem itself.

The rest of this paper is organized as follows. In Section 2, we introduce the robust continuous 0-1 knapsack problem and review the background on the submodular set function. Also, we propose a polynomial-time algorithm for the robust continuous 0-1 knapsack problem itself. In Section 3, we define extended polymatroid inequalities and propose a polynomial-time separation algorithm. Also, we study the solution set of the robust continuous 0-1 knapsack problem. We prove that the convex hull of feasible solutions to the robust continuous 0-1 knapsack problem can be described completely using extended polymatroid inequalities and bound inequalities. In Section 4, the branch-and-cut algorithm with the proposed extended polymatroid cuts is tested on the robust (mixed) 0-1 knapsack problem as well as the multidimensional robust 0-1 knapsack problem. Last, in Section 5, we offer concluding remarks.

2. Preliminaries

2.1. The Robust Continuous 0-1 Knapsack Problem

We focus on the set Y described in (1) when the coefficients are uncertain and y is unrestricted. Each uncertain coefficient, that is, $\tilde{a}_i$, is defined using the Bertsimas and Sim (2004) model. For $i \in N$, $\tilde{a}_i$ is assumed to deviate from its nominal value a_i and be in the interval $[a_i - d_i, a_i + d_i]$. The parameter Γ controls the maximum number of coefficients that can change simultaneously in the intervals. We assume that the parameter Γ takes a nonnegative integer value. Here uncertain coefficients $\tilde{a}$ are included in the polyhedral uncertainty set $\mathcal{U} = \{\tilde{a} \in \mathbb{R}^n \mid \tilde{a}_i = a_i + r_i d_i, \sum_{i\in N} |r_i| \le \Gamma, -1 \le r_i \le 1, \forall i \in N\}$. We can see that a solution (x, y) satisfies the constraints defined by all uncertain coefficients $\tilde{a} \in \mathcal{U}$ if $\max_{\tilde{a}\in\mathcal{U}} \sum_{i\in N} \tilde{a}_i x_i \le b + y$ is satisfied. Then the solution set of the robust continuous 0-1 knapsack problem studied in this paper can be represented as

$$Q = \left\{(x, y) \in \mathbb{B}^n \times \mathbb{R}^1 \mid \sum_{i\in N} a_i x_i + \max_{R\subseteq N, |R|\le\Gamma} \sum_{i\in R} d_i x_i \le b + y\right\} \tag{3}$$

$$= \left\{(x, y) \in \mathbb{B}^n \times \mathbb{R}^1 \mid \sum_{i\in N} a_i x_i + \sum_{i\in R} d_i x_i \le b + y, \forall R \subseteq N, |R| \le \Gamma\right\}. \tag{4}$$

Note that the continuous variable y is unrestricted. We assume that $a_i > 0$ and $d_i \ge 0$ for $i \in N$ and $b \ge 0$. Here (3) has a nonlinear constraint, and (4) has an exponential number of linear constraints. Fischetti and Monaci (2012) used the constraints of (4) as cuts to solve the robust linear programming (LP) and mixed-integer LP models. When $y = 0$, Q is the solution set of the robust 0-1 knapsack problem. The robust continuous 0-1 knapsack problem using the constraint of (3) can be formulated as

$$\begin{aligned} \max \quad & \sum_{i\in N} p_i x_i - qy \\ \text{subject to (s.t.)} \quad & \sum_{i\in N} a_i x_i + \max_{R\subseteq N, |R|\le\Gamma} \sum_{i\in R} d_i x_i \le b + y, \\ & x \in \mathbb{B}^n, y \in \mathbb{R}^1. \end{aligned} \tag{5}$$

Each item $i \in N$ has profit p_i, nominal weight a_i, and deviation value of weight d_i. We assume that $p_i \ge 0$ for all $i \in N$. The parameter q can be interpreted as the cost induced by the additional capacity y. Note that the variable y can be negative. Therefore, (5) is not equivalent to the robust 0-1 knapsack problem even if we set $q = \infty$. We can show that problem (5) itself can be solved in polynomial time.

Proposition 1. *The robust continuous* 0-1 *knapsack problem* (5) *can be solved in* $O(n^2)$.

Proof. We consider the case $q \ge 0$ only because otherwise (5) is unbounded. Let (x^*, y^*) be an optimal solution of (5). When $q = 0$, it is trivial because $x_i^* = 1$ for all $i \in N$, and $y^* \ge \sum_{i\in N} a_i x_i^* + \max_{R\subseteq N, |R|\le\Gamma} \sum_{i\in R} d_i x_i^* - b$ gives the maximum objective value. When $q > 0$, (x^*, y^*) satisfies the constraint $\sum_{i\in N} a_i x_i + \max_{R\subseteq N, |R|\le\Gamma} \sum_{i\in R} d_i x_i \le b + y$ at equality. Otherwise, $(x^*, y^* - \epsilon)$ is a feasible solution to (5) for small $\epsilon > 0$, and it contradicts the optimality of (x^*, y^*).

Let z^* be the optimal value of (5); then (5) can be represented as the minimization problem

$$\begin{aligned} z^* &= \sum_{i\in N} p_i x_i^* - qy^* \\ &= \max_{x\in\mathbb{B}^n} \left\{ \sum_{i\in N} p_i x_i - q\left(\sum_{i\in N} a_i x_i + \max_{R\subseteq N, |R|\le\Gamma} \sum_{i\in R} d_i x_i - b \right) \right\} \\ &= qb - \min_{x\in\mathbb{B}^n} \left\{ \sum_{i\in N} (qa_i - p_i) x_i + \max_{R\subseteq N, |R|\le\Gamma} \sum_{i\in R} q d_i x_i \right\}. \end{aligned} \tag{6}$$

Bertsimas and Sim (2003) showed that (6) can be solved by solving the $n + 1$ nominal problems (see also Lee et al. 2012, Lee and Kwon 2014). We assume that items are sorted in nonincreasing order of d_i, and define $d_{n+1} = 0$. Then

$$z^* = \min_{l=1,\ldots,n+1} G^l,$$

where for $l = 1, \ldots, n + 1$,

$$G^l = qb - \Gamma d_l - \min_{x\in\mathbb{B}^n} \left\{ \sum_{i\in N} (qa_i - p_i) x_i + \sum_{i=1}^{l} q(d_i - d_l) x_i \right\}.$$

Each nominal problem G^l can be solved in linear time; therefore, z^* can be obtained in $O(n^2)$. □

However, the main purpose of our study is not to solve (5) in a short time but to identify the valid inequalities for the solution set of (5) so that they can be used to describe the solution sets of general robust mixed 0-1 problems more tightly. Problem (5) can be reformulated using strong duality to the inner maximization problem as follows (see Bertsimas and Sim 2004):

$$\max \quad \sum_{i\in N} p_i x_i - qy \tag{7}$$

$$\text{s.t.} \quad \sum_{i\in N} a_i x_i + \sum_{i\in N} u_i + \Gamma v \le b + y, \tag{8}$$

$$d_i x_i - u_i - v \le 0, \quad \forall i \in N, \tag{9}$$

$$u_i \ge 0, \quad \forall i \in N, \tag{10}$$

$$v \ge 0, \tag{11}$$

$$x \in \mathbb{B}^n, \tag{12}$$

$$y \in \mathbb{R}^1. \tag{13}$$

Here u and v are dual variables of the inner maximization problem in (5). The preceding model has $O(n)$ variables and $O(n)$ linear constraints. The objective value of the LP relaxation problem of (7)–(13) is equal to the objective value of the LP relaxation problem of (5) (see Bertsimas and Sim 2003, 2004; Fischetti and Monaci 2012). The convex hull of solutions satisfying constraints (9)–(12) was studied by Atamtürk (2006). He focused on strong formulations for robust mixed 0-1 programming with uncertain objective coefficients, which can be applied when the coefficients of constraints have uncertainty as well. He proposed valid inequalities for the set defined by constraints (9)–(12) and showed that the convex hull of solutions satisfying constraints (9)–(12) can be formulated as follows with additional variables w:

$$\left\{ (x,u,v,w) \in \mathbb{R}^{3n+3} : \begin{array}{l} (d_j - d_i)x_j + w_j - w_i \le u_j, \quad 0 \le i < j \le n, \\ w_{n+1} - w_i \le v, \quad 0 \le i \le n, \\ w_{n+1} - w_0 \ge 0, \\ u_i \ge 0, \quad \forall i \in N, \\ v \ge 0, \\ \mathbf{0} \le x \le \mathbf{1} \end{array} \right\}. \tag{14}$$

Here $d_0 = 0$, and the x variables are sorted in nondecreasing order of d_i. The preceding formulation is of polynomial size, but $O(n)$ variables and $O(n^2)$ constraints should be considered for each uncertain constraint. In this paper, we investigate valid inequalities for the feasible solutions of (5). As a result, we prove that the convex hull of the solution set of (5) can be completely described using the linear inequalities we propose and bound inequalities in the space of the original x and y variables.

2.2. Submodularity

Given a ground set N with n items, 2^N stands for the family of all subsets of N. Then a scalar-valued function $f : 2^N \to \mathbb{R}$ is called a *set function*. A submodular set function that has a diminishing-returns property is defined as follows.

Definition 1. A set function f is *submodular* if it satisfies

$$f(A \cup \{j\}) - f(A) \ge f(B \cup \{j\}) - f(B), \quad \text{for } A, B \subseteq N, A \subseteq B, \text{ and } j \in N \setminus B. \tag{15}$$

Equivalently, f is submodular if

$$f(A) + f(B) \ge f(A \cup B) + f(A \cap B), \quad \text{for } A, B \subseteq N. \tag{16}$$

Examples of submodular functions include

• *Cut functions.* Given a directed graph $G = (V, A)$ and a capacity function $c : A \to \mathbb{R}_+$, the cut function $f(U)$ for $U \subseteq V$ is defined as the total capacity of the arcs leaving U.

• *Rank functions of matroids.* Let $M = (S, I)$ be a matroid. Then the rank function $r_M : 2^S \to \mathbb{R}_+$ is submodular.

• *Coverage functions.* Let $E_1, E_2, \ldots, E_n$ be subsets of the ground set E. The coverage function f is defined as $f(C) = |\cup_{i\in C} E_i|$.

If f is submodular, a polyhedron associated with f can be defined as follows (see Schrijver 2003).

Definition 2. The extended polymatroid (*submodular polyhedron*) associated with a submodular function f is

$$\Pi_f = \left\{ \pi \in \mathbb{R}^n \mid \sum_{i \in S} \pi_i \le f(S), \forall S \subseteq N \right\}.$$

Given a weight vector $w \ge 0$ and a submodular function f with $f(\emptyset) = 0$, Edmonds (1970) showed that

$$\begin{aligned} &\max \ w^T \pi \\ &\text{s.t.} \quad \pi \in \Pi_f \end{aligned}$$

can be solved using a greedy algorithm. We apply this greedy algorithm to propose a polynomial time-separation algorithm for valid inequalities for the set Q defined in (3).

3. Extended Polymatroid Inequalities

3.1. Extended Polymatroid Inequalities and a Polynomial-Time Separation Algorithm

In this section, we define valid inequalities for the solution set of the robust continuous 0-1 knapsack problem using the submodularity of the robust 0-1 knapsack set function. We then propose a polynomial-time algorithm for the separation of the most violated inequality. We consider only nonnegative integer Γ values. We define the robust 0-1 knapsack set function as follows.

Definition 3. The robust 0-1 knapsack set function $f(S)$ for $S \subseteq N$ is defined as

$$f(S) = \sum_{i \in S} a_i + \max_{R \subseteq S, |R| \le \Gamma} \sum_{i \in R} d_i. \tag{17}$$

Thus, the robust 0-1 knapsack set function is a submodular set function (see Kutschka 2013), and $f(\emptyset) = 0$. We can define valid inequalities for Q using the submodularity of the robust 0-1 knapsack set function.

Proposition 2. *If $\pi \in \Pi_f$, then*

$$\sum_{i \in N} \pi_i x_i \le b + y \tag{18}$$

is a valid inequality for Q.

Proof. Let $X = \{i \in N \mid x_i = 1\}$ for $x \in \mathbb{B}^n$. By the definition of Π_f, $\sum_{i \in N} \pi_i x_i \le f(X)$ is satisfied if $\pi \in \Pi_f$. In addition, we know that $f(X) \le b + y$ for all $(x, y) \in Q$. Therefore, $\sum_{i \in N} \pi_i x_i \le b + y$ is valid for Q if $\pi \in \Pi_f$. □

For the submodular function f, we call (18) an extended polymatroid inequality. If f is a submodular function with $f(\emptyset) = 0$, the separation problem for extended polymatroid inequalities $\max\{\sum_{i \in N} x_i^* \pi_i \mid \pi \in \Pi_f\}$ can be solved by the greedy algorithm (see Edmonds 1970). Let $S_i = \{1, 2, \dots, i\}$ and $S_0 = \emptyset$. The most violated extended polymatroid inequality $\bar{\pi} x \le b + y$ for the current fractional solution (x^*, y^*) can be obtained as $\bar{\pi}_i = f(S_i) - f(S_{i-1})$ for $i = 1, \dots, n$ when items of N are sorted in nonincreasing order of x_i^*.

Proposition 3. *For the robust* 0-1 *knapsack set function $f(S)$ for $S \subseteq N$,*

$$\bar{\pi}_i = f(S_i) - f(S_{i-1}) = \begin{cases} a_i + d_i, & \text{if } i \le \Gamma, \\ a_i + d_i - d_{i_{\min}}, & \text{if } i \ge \Gamma + 1, \end{cases} \quad \text{for } i = 1, \dots, n, \tag{19}$$

where $i_{\min} = \mathrm{argmin}_{j \in D_{i-1} \cup \{i\}} d_j$ and $D_i = \arg\max_{R \subseteq S_i, |R| \le \Gamma} \sum_{j \in R} d_j$.

Proof. If $i \le \Gamma$, then $f(S_i) - f(S_{i-1}) = (\sum_{j \in S_i} a_j + \sum_{j \in S_i} d_j) - (\sum_{j \in S_{i-1}} a_j + \sum_{j \in S_{i-1}} d_j) = a_i + d_i$. *If $i \ge \Gamma + 1$, then* $f(S_i) - (S_{i-1}) = (\sum_{j \in S_i} a_j + \sum_{j \in D_i} d_j) - (\sum_{j \in S_{i-1}} a_j + \sum_{j \in D_{i-1}} d_j)$. Here $D_i = D_{i-1} \cup \{i\} \setminus \{i_{\min}\}$ because $|D_{i-1}| = \Gamma$. Therefore, $f(S_i) - f(S_{i-1}) = a_i + d_i - d_{i_{\min}}$. □

In addition, D_i can be updated as follows:

$$D_i = \begin{cases} D_{i-1} \cup \{i\}, & \text{if } i \le \Gamma, \\ D_{i-1} \cup \{i\} \setminus \{i_{\min}\}, & \text{if } i \ge \Gamma + 1, \end{cases} \quad \text{for } i = 1, \dots, n.$$

Because the size of D_i is always smaller than or equal to Γ, we can find $i_{\min}$ in $O(\Gamma)$ for $i = 1, \dots, n$.

Example 1. Consider a robust continuous 0-1 knapsack set with $n = 4$, $\Gamma = 2$, $(a_1, a_2, a_3, a_4) = (8, 9, 11, 7)$, $(d_1, d_2, d_3, d_4) = (3, 1, 4, 2)$, and $b = 20$. Assume that the items are sorted in nonincreasing order of the fractional solution x_i^* (i.e., $x_1^* \geq x_2^* \geq x_3^* \geq x_4^*$). Then the most violated extended polymatroid inequality can be obtained as follows:

1. $\bar{\pi}_1 = f(S_1) - f(S_0) = 11, D_1 = \{1\}$.
2. $\bar{\pi}_2 = f(S_2) - f(S_1) = 10, D_2 = \{1, 2\}$.
3. $\bar{\pi}_3 = f(S_3) - f(S_2) = 15 - \min\{d_1, d_2, d_3\} = 15 - \min\{3, 1, 4\} = 14, D_3 = \{1, 2, 3\} \setminus \{2\} = \{1, 3\}$.
4. $\bar{\pi}_4 = f(S_4) - f(S_3) = 9 - \min\{d_1, d_3, d_4\} = 9 - \min\{3, 4, 2\} = 7, D_4 = \{1, 3, 4\} \setminus \{4\} = \{1, 3\}$.

If $11x_1^* + 10x_2^* + 14x_3^* + 7x_4^* > 20 + y^*$, then

$$11x_1 + 10x_2 + 14x_3 + 7x_4 \leq 20 + y \tag{20}$$

is the most violated extended polymatroid inequality.

Note that $d_i - d_{i_{\min}}$ of (19), which is always greater than or equal to zero, is the term that can make the extended polymatroid inequalities stronger than the constraints in the original robust model (4). If we use a_i instead of $a_i + d_i - d_{i_{\min}}$ when $i \geq \Gamma + 1$, the corresponding inequality is equivalent to the constraint of (4) when $R = S_\Gamma = \{1, \ldots, \Gamma\}$ of the sorted set. Therefore, proposed extended polymatroid inequalities are always equivalent to or stronger than the cuts used by Fischetti and Monaci (2012). When we separate the most violated extended polymatroid inequality, if some variables have the same x^* values, we break the tie by sorting the variables in nondecreasing order of d_i to make the value of $d_i - d_{i_{\min}}$ larger. This tie-breaking rule does not affect the extent of violation of the extended polymatroid inequality. However, it is expected that this rule can be effective when the extended polymatroid inequalities are used as cutting planes in a branch-and-cut algorithm. See the following example.

Example 2 (Continuation of Example 1)**.** If $x_1^* \geq x_2^* = x_3^* \geq x_4^*$, the extended polymatroid inequality separated in the order of x_1, x_3, x_2, and x_4 is

$$11x_1 + 9x_2 + 15x_3 + 7x_4 \leq 20 + y. \tag{21}$$

Then the extents of violations of (20) and (21) are equal because $x_2^* = x_3^*$. Here (20) is stronger than the following constraint of (4) when $R = \{1, 2\}$, that is,

$$11x_1 + 10x_2 + 11x_3 + 7x_4 \leq 20 + y,$$

because the coefficient of x_3 of (20) is 14, which is greater than 11. Note that the coefficient of x_3 in (20) differs by $d_3 - d_{3_{\min}} = 3$. By contrast, we can see that (21) is equivalent to the constraint of (4) when $R = \{1, 3\}$. Because $d_2 < d_3$, the tie-breaking rule chooses (20) as the most violated extended polymatroid inequality. Therefore, the most violated inequality $\sum_{i \in N} \bar{\pi}_i x_i \leq b + y$ for a given (x^*, y^*) can be separated in $O(n \log n + \Gamma n)$. The overall algorithm is summarized in Algorithm 1.

Algorithm 1 (Separation of the Most Violated Extended Polymatroid Inequality (18))

Sort variables in nonincreasing order of x_i^* ▷ Initialization: $O(n \log n)$.
Break ties by sorting variables in nondecreasing order of d_i.
$D_0 = \emptyset$.
for $i = 1$ to n, **do** ▷ Compute $\bar{\pi}_i$.
  if $i \leq \Gamma$, **then**
    $\bar{\pi}_i = a_i + d_i$ and $D_i = D_{i-1} \cup \{i\}$.
  else
    $i_{\min} = \operatorname{argmin}_{j \in D_{i-1} \cup \{i\}} d_j$ ▷ $O(\Gamma)$.
    $\bar{\pi}_i = a_i + d_i - d_{i_{\min}}$ and $D_i = D_{i-1} \cup \{i\} \setminus \{i_{\min}\}$.
  end if
end for
if $\sum_{i \in N} \bar{\pi}_i x_i^* > b + y^*$, **then**
  (x^*, y^*) is violated by $\sum_{i \in N} \bar{\pi}_i x_i \leq b + y$.
else
  There is no violated extended polymatroid inequality.
end if

We let $ext(\Pi_f)$ denote the set of extreme points of Π_f. Note that the extreme points of Π_f are precisely the vectors π obtained by (19) for all permutations of N (see Edmonds 1970). Therefore, as we noted earlier, the extended polymatroid inequalities $\sum_{i\in N}\pi_i x_i \le b+y$ for all $\pi\in ext(\Pi_f)$ are equivalent to or stronger than the constraints of (4); that is, $\sum_{i\in N} a_i x_i + \sum_{i\in R} d_i x_i \le b+y, \forall R\subseteq N, |R|\le\Gamma$. We can prove that the set Q defined in (4) can be described only by extended polymatroid inequalities with $\pi\in ext(\Pi_f)$.

Proposition 4. *Let*

$$Q_f = \left\{(x,y)\in\mathbb{B}^n\times\mathbb{R}^1 \mid \sum_{i\in N}\pi_i x_i \le b+y, \forall\pi\in ext(\Pi_f)\right\}. \tag{22}$$

Then $Q_f = Q$.

Proof. Here $\sum_{i\in N}\pi_i x_i\le b+y, \forall\pi\in ext(\Pi_f)$ can be obtained, instead of the nonincreasing order of x_i^*, using (19) for all permutations of N. In addition,

$$\begin{aligned}&\left\{(x,y)\in\mathbb{B}^n\times\mathbb{R}^1 \mid \sum_{i\in N} a_i x_i + \sum_{i\in R} d_i x_i \le b+y, \forall R\subseteq N, |R|\le\Gamma\right\}\\ =&\left\{(x,y)\in\mathbb{B}^n\times\mathbb{R}^1 \mid \sum_{i\in N}\pi'_i x_i \le b+y, \forall\pi'\in\Pi'_f\right\},\end{aligned}$$

where $\Pi'_f=\{\pi'\in\mathbb{R}^n \mid \pi'_{\sigma_{(i)}}=a_{\sigma_{(i)}}+d_{\sigma_{(i)}}$ if $i\le\Gamma$ and $\pi'_{\sigma_{(i)}}=a_{\sigma_{(i)}}$ *if* $i>\Gamma$ for some permutations $(\sigma_{(1)},\dots,\sigma_{(n)})$ of $N\}$. Hence, $\sum_{i\in N}\pi_{\sigma_{(i)}}x_{\sigma_{(i)}}\le b+y$ is equivalent to or stronger than $\sum_{i\in N}\pi'_{\sigma_{(i)}}x_{\sigma_{(i)}}\le b+y$ for some permutations $(\sigma_{(1)},\dots,\sigma_{(n)})$ of N because $d_{\sigma_{(i)}}-d_{\sigma_{(i)_{\min}}}\ge 0$ for $i=1,\dots,n$ in (19). This means that $Q_f\subseteq Q$. In addition, $Q_f\supseteq Q$ is satisfied because $\sum_{i\in N}\pi_i x_i\le b+y$ for all $\pi\in ext(\Pi_f)$ are valid inequalities for Q by Proposition 2. Therefore, $Q_f=Q$. □

Furthermore, the extended polymatroid inequality obtained by Algorithm 1 defines a facet of $conv(Q)$ where $conv(Q)$ denotes the convex hull of Q. Let $e^i\in\mathbb{B}^n$ be the n-dimensional ith unit vector. Note that $conv(Q)$ is full dimensional because $(\mathbf{0},0)$, $(\mathbf{1},\sum_{i\in N}(a_i+d_i)+1)$, and (e^i, a_i+d_i) for $i\in N$ are $n+2$ affinely independent feasible points.

Proposition 5. *If* $\pi\in ext(\Pi_f)$, *then* $\sum_{i\in N}\pi_i x_i\le b+y$ *defines a facet of* $conv(Q)$.

Proof. Edmonds (1970) showed that if f is a submodular function with $f(\emptyset)=0$ and $\pi\in ext(\Pi_f)$, then $\pi_{\sigma_{(i)}}=f(S_{\sigma_{(i)}})-f(S_{\sigma_{(i)-1}})$ is satisfied for some permutations $(\sigma_{(1)},\sigma_{(2)},\dots,\sigma_{(n)})$ of N. Note that $\sum_{j=1}^i\pi_{\sigma_{(j)}}=f(S_{\sigma_{(i)}})$. Then $(\mathbf{0},-b)$, $(\sum_{j=1}^i e^{\sigma(j)}, \sum_{j=1}^i\pi_{\sigma_{(j)}}-b)$ for $1\le i\le n$ are feasible for (3) and satisfy $\sum_{i\in N}\pi_i x_i\le b+y$ at equality. In addition, they are affinely independent $n+1$ points. □

Also, we prove that $conv(Q)$ can be described completely by the extended polymatroid inequalities (18).

Proposition 6. *The convex hull of Q can be described completely by extended polymatroid inequalities and bound inequalities; that is,*

$$conv(Q)=\left\{(x,y)\in\mathbb{R}^n\times\mathbb{R}^1 \mid \sum_{i\in N}\pi_i x_i\le b+y, \forall\pi\in ext(\Pi_f), \mathbf{0}\le x\le\mathbf{1}\right\}.$$

Proof. We define an unrestricted continuous variable $z=b+y$. Then the epigraph of the submodular function f,

$$E_f=\left\{(x,z)\in\mathbb{B}^n\times\mathbb{R}^1 \mid \sum_{i\in N}a_i x_i+\sum_{i\in R}d_i x_i\le z, \forall R\subseteq N, |R|\le\Gamma\right\},$$

is equivalent to Q. The convex hull of E_f with $f(\emptyset)=0$ can be described completely by bound inequalities and $\sum_{i\in N}\pi_i x_i\le z$, where π values are obtained by Algorithm 1 (see Edmonds 1970, Baumann et al. 2013). □

The following example represents Proposition 6 graphically.

Example 3. We consider the solution set of the robust continuous 0-1 knapsack problem (4) with $n = 2$, $(a_1, a_2) = (3, 4)$, $(d_1, d_2) = (2, 3)$, $\Gamma = 1$, and $b = 0$. The solution sets of the deterministic model with $\Gamma = 0$, the original robust model (4), and the proposed robust model with extended polymatroid inequalities (22) are as follows:

a. For the deterministic model, $\{(x, y) \in \mathbb{B}^2 \times \mathbb{R}^1 \mid 3x_1 + 4x_2 \le y\}$.
b. For the original robust model, $\{(x, y) \in \mathbb{B}^2 \times \mathbb{R}^1 \mid 5x_1 + 4x_2 \le y, 3x_1 + 7x_2 \le y\}$.
c. For the proposed robust model, $\{(x, y) \in \mathbb{B}^2 \times \mathbb{R}^1 \mid 5x_1 + \mathbf{5}x_2 \le y, 3x_1 + 7x_2 \le y\}$.

Figure 1 represents the solution set of each model. The shaded region is the solution set of the LP relaxation. The thick lines represent feasible solutions to the mixed-integer programming (MIP) problem. As can be seen, the solution set of the robust MIP problem (the thick lines in Figure 1, (b) and (c)) is included in the solution set of the deterministic model (the thick lines of Figure 1(a)). In Figure 1(b), the shaded region is not the convex hull of feasible solutions to the MIP problem. However, the proposed robust model (Figure 1(c)) shows that extended polymatroid inequalities represent the convex hull of feasible solutions to the MIP problem.

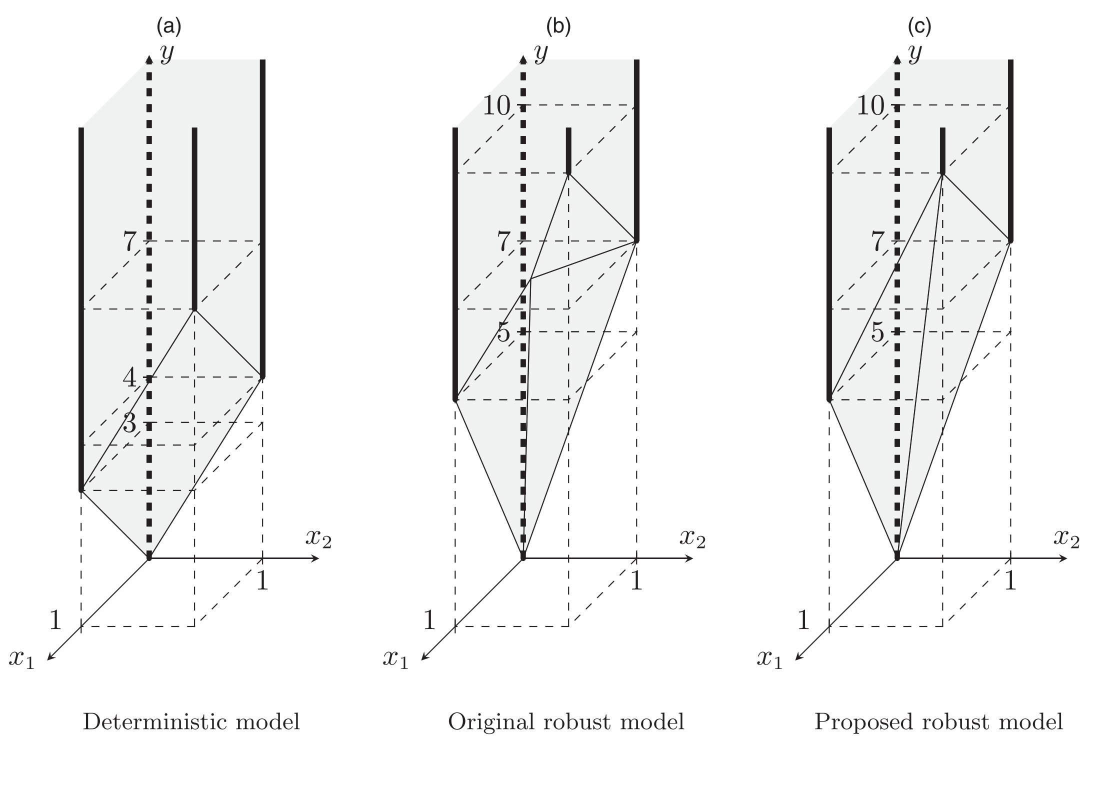
By the equivalence of optimization and separation (see Grötschel et al. 1981), the results of Proposition 6 together with the polynomial-time separation algorithm (Algorithm 1) imply the existence of a polynomial-time algorithm for the robust continuous 0-1 knapsack problem (5). An algorithm whose running time is $O(n^2)$ is presented in Proposition 1. Note that we define the continuous variable y as unrestricted. If y is restricted ($y \ge 0$), then Propositions 5 and 6 do not hold, but extended polymatroid inequalities are still valid for Q. Also, we assume that Γ is a nonnegative integer, and our results are not extended directly to the case with noninteger Γ.

3.2. Applying Extended Polymatroid Inequalities to Robust Mixed 0-1 Problems

Extended polymatroid inequalities also can be applied to the solution set Z described in (2) with uncertain coefficients $\tilde{a}$. Here the set Z has bounded continuous variables, and uncertain coefficients of binary variables and continuous variables are included in the same uncertainty set. The upper bounds on bounded continuous variables are defined as one for all continuous variables by scaling. Let $\tilde{Z}$ be the following solution set of the robust mixed 0-1 knapsack problem:

$$\tilde{Z} = \left\{ x \in \mathbb{B}^n \times [0,1]^m \mid \sum_{i \in N \cup M} a_i x_i + \max_{R \subseteq N \cup M, |R| \le \Gamma} \sum_{i \in R} d_i x_i \le b \right\}. \tag{23}$$

Figure 1. Solution Sets of LP Relaxed Formulations



Then the robust mixed 0-1 knapsack problem is as follows:

$$\begin{aligned} \max \quad & \sum_{i \in N \cup M} p_i x_i \\ \text{s.t.} \quad & \sum_{i \in N \cup M} a_i x_i + \max_{R \subseteq N \cup M, |R| \le \Gamma} \sum_{i \in R} d_i x_i \le b, \\ & x_i \in \{0,1\}, \quad \forall i \in N, \\ & 0 \le x_i \le 1, \quad \forall i \in M. \end{aligned} \tag{24}$$

In addition, (24) can be reformulated, using strong duality, to the inner maximization problem in the constraint as (see Bertsimas and Sim 2004)

$$\begin{aligned} \max \quad & \sum_{i \in N \cup M} p_i x_i \\ \text{s.t.} \quad & \sum_{i \in N \cup M} a_i x_i + \sum_{i \in N \cup M} u_i + \Gamma v \le b, \\ & d_i x_i - u_i - v \le 0, \quad \forall i \in N \cup M, \\ & u_i \ge 0, \quad \forall i \in N \cup M, \\ & v \ge 0, \\ & x_i \in \{0,1\}, \quad \forall i \in N, \\ & 0 \le x_i \le 1, \quad \forall i \in M. \end{aligned} \tag{25}$$

Because we assume that Γ has an integer value, $\Gamma = \gamma + (\Gamma - \gamma)$ for $\gamma = 0, \ldots, \Gamma$. Therefore, $\tilde{Z}$ can be represented by dividing the penalty terms in the constraint of (23) into the binary part and the continuous part as follows:

$$\tilde{Z} = \left\{ x \in \mathbb{B}^n \times [0,1]^m \mid \sum_{i \in N} a_i x_i + \max_{R \subseteq N, |R| \le \gamma} \sum_{i \in R} d_i x_i + \sum_{i \in M} a_i x_i + \max_{R \subseteq M, |R| \le \Gamma - \gamma} \sum_{i \in R} d_i x_i \le b, \forall \gamma = 0, \ldots, \Gamma \right\}.$$

Then the binary part $\sum_{i \in N} a_i x_i + \max_{R \subseteq N, |R| \le \gamma} \sum_{i \in R} d_i x_i$ is a robust 0-1 knapsack set function defined with the item set N and the parameter γ. Therefore, the coefficients of the extended polymatroid inequalities π^γ for the set N can be obtained using (19). When $\gamma = 0$, we can see that $\pi_i^0 = a_i$ for all $i \in N$ by (19). In this case, because any feasible solution to the LP relaxation of (25) satisfies $\sum_{i \in N} a_i x_i + \sum_{i \in M} a_i x_i + \max_{R \subseteq M, |R| \le \Gamma} \sum_{i \in R} d_i x_i \le b$, we consider only when $\gamma = 1, \ldots, \Gamma$. For each $\gamma = 1, \ldots, \Gamma$, if π^γ satisfies

$$\sum_{i \in N} \pi_i^\gamma x_i \le \sum_{i \in N} a_i x_i + \max_{R \subseteq N, |R| \le \gamma} \sum_{i \in R} d_i x_i,$$

then

$$\sum_{i \in N} \pi_i^\gamma x_i \le b - \sum_{i \in M} a_i x_i - \max_{R \subseteq M, |R| \le \Gamma - \gamma} \sum_{i \in R} d_i x_i \tag{26}$$

is a valid inequality for $\tilde{Z}$. Inequality (26) can be represented as the following inequality using dual variables u and v:

$$\sum_{i \in N} \pi_i^\gamma x_i \le b - \sum_{i \in M} a_i x_i - \sum_{i \in M} u_i - \left(\Gamma - \gamma\right) v. \tag{27}$$

The other necessary constraints for variables u and v are defined in (25). When the fractional solution values (x^*, u^*, v^*) to (25) are given, the most violated inequality can be separated in the polynomial time of $O(n \log n + \Gamma^2 n)$. Items of N can be sorted in $O(n \log n)$, and $\bar{\pi}^\gamma$ for each $\gamma = 1, \ldots, \Gamma$ can be obtained in $O(\Gamma n)$ as in Algorithm 1. The overall separation algorithm is given as Algorithm 2.

Algorithm 2 (Separation of the Most Violated Extended Polymatroid Inequality (27))

Sort variables of N in nonincreasing order of x_i^* ▷ Initialization: $O(n \log n)$.
Break ties by sorting variables in nondecreasing order of d_i.
$D_0 = \emptyset$.
for $\gamma = 1$ to Γ, **do** ▷ Compute $\bar{\pi}^\gamma$.
  for $i = 1$ to n, **do**
    if $i \le \gamma$, **then**
      $\bar{\pi}_i^\gamma = a_i + d_i$ and $D_i = D_{i-1} \cup \{i\}$.

INFORMS Journal on Optimization, 2021, vol. 3, no. 2, pp. 183–199, © 2021 INFORMS


```
        else
            i_min = argmin_{j∈D_{i−1}∪{i}} d_j                                   ▷ O(Γ).
            π̄_i^γ = a_i + d_i − d_{i_min} and D_i = D_{i−1} ∪ {i}\{i_min}.
        end if
    end for
    if Σ_{i∈N} π̄_i^γ x_i^* + Σ_{i∈M} a_i x_i^* + (Γ − γ)v^* + Σ_{i∈M} u_i^* > b, then
        (x^*, u^*, v^*) is violated by Σ_{i∈N} π̄_i^γ x_i + Σ_{i∈M} a_i x_i + (Γ − γ)v + Σ_{i∈M} u_i ≤ b.
    else
        There is no violated extended polymatroid inequality.
    end if
end for
```

4. Computational Results

In this section, we report the computational results on the robust (mixed) 0-1 knapsack problem and the multidimensional robust 0-1 knapsack problem. All the experiments were performed on an Intel Core i5-3570K processor (3.40 GHz with 8 GB of random-access memory). We implemented the algorithm using Java and IBM ILOG CPLEX 12.5. The time limit was 600 central processing unit seconds. Extended polymatroid cuts were generated only at the root node of the branch-and-cut tree.

4.1. Results on the Robust (Mixed) 0-1 Knapsack Problem

We report the computational results on the robust 0-1 knapsack problem

$$\begin{aligned} \max \quad & \sum_{i\in N} p_i x_i \\ \text{s.t.} \quad & \sum_{i\in N} a_i x_i + \max_{R\subseteq N, |R|\le\Gamma} \sum_{i\in R} d_i x_i \le b, \\ & x \in \mathbb{B}^n, \end{aligned} \tag{28}$$

and the robust mixed 0-1 knapsack problem (24). Note that the optimal solution of the robust 0-1 knapsack problem (28) itself can be obtained in pseudopolynomial time $O(\Gamma nb)$ using dynamic programming (see Monaci et al. 2013) or by solving deterministic 0-1 knapsack problems iteratively (see Bertsimas and Sim 2003, Lee et al. 2012). However, the purpose of the experiments was to verify the effectiveness of the extended polymatroid cuts when they are applied in a branch-and-cut algorithm. Therefore, we did not compare the results with these methods. We implemented the reformulated problems

$$\begin{aligned} \max \quad & \sum_{i\in N} p_i x_i \\ \text{s.t.} \quad & \sum_{i\in N} a_i x_i + \sum_{i\in N} u_i + \Gamma v \le b, \\ & d_i x_i - u_i - v \le 0, \qquad \forall i \in N, \\ & u_i \ge 0, \qquad \forall i \in N, \\ & v \ge 0, \\ & x \in \mathbb{B}^n, \end{aligned} \tag{29}$$

and (25).

The robust 0-1 knapsack problem (28) is the robust continuous 0-1 knapsack problem (5) with the constraint $y = 0$. Therefore, $\sum_{i\in N} \pi_i x_i \le b$ for $\pi \in ext(\Pi_f)$ are valid inequalities for the solution set of the robust 0-1 knapsack problem. In addition, for the robust mixed 0-1 knapsack problem (24), extended polymatroid inequalities can be separated using Algorithm 2. In this computational test, the most violated extended polymatroid inequality among the obtained extended polymatroid inequalities for $\gamma = 1, \ldots, \Gamma$ was used in the branch-and-cut method. For the computational tests, we used five robust 0-1 knapsack instance types considered by Monaci et al. (2013):

- *Uncorrelated (UN) instances:* Here p_i and a_i are randomly generated integers in $[1, 100]$.
- *Weakly correlated (WC) instances:* Here a_i is a random integer in $[1, 100]$, and p_i is an integer randomly generated in $[\max\{1, a_i - 10\}, a_i + 10]$.

- *Strongly correlated (SC) instances:* Here a_i is an integer randomly chosen in $[1, 100]$, and $p_i = a_i + 10$.
- *Inverse strongly correlated (IC) instances:* Here p_i is a random integer in $[1, 100]$, and $a_i = \min\{100, p_i + 10\}$.
- *Subset sum (SS) instances:* Here a_i is a random integer in $[1, 100]$, and $p_i = a_i$.

The deviation value d_i is an integer randomly generated in $[0, 100 - a_i]$, and we used $b = \lfloor \sum_{i\in N} a_i/2 \rfloor$. For the robust mixed 0-1 knapsack problems, $m = n/4$ bounded continuous variables were added to the robust 0-1 knapsack problem. Here p_i, a_i, and d_i for $i \in M$ were generated in the same way as binary variables. We performed tests on the problems with $n = 200, 500$ and $\Gamma = 1, 5, 10$. We report the averaged results of 10 random instances for each combination.

We compared the branch-and-cut method with extended polymatroid cuts with other branch-and-cut methods. "CPLEX" denotes the CPLEX default branch-and-cut method. Here the reformulated forms of the original robust problems (29) and (25) were used. In addition, the extended formulation of Atamtürk (2006) given in (14) is solved using the CPLEX default branch-and-cut method. When extended polymatroid cuts are used and the extended formulation of Atamtürk (2006) is solved, we reported both results of using CPLEX default cuts and not using them. Atamtürk (2006) constructed and solved the dual of the extended formulation at the nodes of the branch-and-cut tree to obtain faster implementation. However, such capability is not provided by a regular CPLEX package; thus, we implemented their extended formulation using CPLEX default settings for a fair comparison.

The extended formulation of Atamtürk (2006) also can be applied to robust mixed 0-1 knapsack problems. The constraints

$$\begin{aligned}
&d_i x_i - u_i - v \le 0, \quad \forall i \in N \cup M,\\
&u_i \ge 0, \quad \forall i \in N \cup M,\\
&v \ge 0,\\
&x_i \in \{0, 1\}, \quad \forall i \in N,\\
&0 \le x_i \le 1, \quad \forall i \in M,
\end{aligned}$$

of (25) can be extended to the following constraints with additional variables w:

$$\begin{aligned}
&(d_j - d_i)x_j + w_j - w_i \le u_j, \quad (i, j) \in E,\\
&w_{n+m+1} - w_i \le v, \quad 0 \le i \le n + m,\\
&w_{n+m+1} - w_0 \ge 0,\\
&u_i \ge 0, \quad \forall i \in N \cup M,\\
&v \ge 0,\\
&x_i \in \{0, 1\}, \quad \forall i \in N,\\
&0 \le x_i \le 1, \quad \forall i \in M,
\end{aligned}$$

where $E = \{(i, j) : 0 \le i < j \le n + m\} \setminus \{(h, j) \text{ for } h \in [1, j - 1] \text{ for } j \in M\}$. The extended formulation has $O(n + m)$ additional variables w and $O(n^2 + mn)$ constraints. Here the items of $N \cup M$ are sorted in nondecreasing order of d_i.

The average results of the different approaches are summarized in Tables 1–4. Tables 1 and 2 show the average results of the robust 0-1 knapsack problem (28) for different numbers of variables. Tables 3 and 4 report the average results of the robust mixed 0-1 knapsack problem (24). The average gap value ("Gap (%)"), the average number of explored nodes ("Nodes"), the average computation time in seconds ("Time"), and the average number of added extended polymatroid cuts ("Cuts") are given. Here, Gap (%) = $(z_{lp} - z_{cuts})/(z_{lp} - z_{opt}) \times 100$, where z_{lp} is the LP relaxation value of the original robust model of (29) and (25), z_{cuts} is the objective value of the LP relaxation problem after cuts are added, and z_{opt} is the optimal objective value. Therefore, the gap value is between 0% and 100%, a higher gap value indicating that the added cuts are more effective. When some instances are not solved within the time limit, we report the average computation time and the average number of nodes of instances solved to optimality within the time limit. In addition, the number of unsolved instances out of 10 instances is given in parentheses.

For the robust 0-1 knapsack problem, as can be seen in Tables 1 and 2, the gap values were significantly improved when extended polymatroid cuts were applied or the extended formulation of Atamtürk (2006) was used compared with the default branch-and-cut method of CPLEX. We also noted that when the CPLEX default cuts were used, improvements in gap values deteriorated as the value of Γ increased. For some instance types, especially the SS type with $\Gamma = 5$ and 10, the bound improvements with the CPLEX default cuts were almost negligible, which might indicate that SS-type problems are particularly hard to solve. It can be posited

Table 1. Robust 0-1 Knapsack Problem with n = 200

					Extended formulation of Atamtürk (2006)						Extended polymatroid cuts							
		CPLEX			Without CPLEX default cuts			With CPLEX default cuts			Without CPLEX default cuts				With CPLEX default cuts			
Γ	Type	Gap (%)	Time	Nodes	Gap (%)	Time	Nodes	Gap (%)	Time	Nodes	Gap (%)	Time	Nodes	Cuts	Gap (%)	Time	Nodes	Cuts
1	UN	70.8	0	46	26.2	1	99	79.5	2	33	30.5	0	63	1	83.5	0	18	1
	WC	65.7	0	83	98.9	2	88	99.3	2	52	99.1	0	67	13	99.2	0	49	12
	SC	100.0	0	0	84.1	10 (4)	4,876	86.9	9 (4)	3,213	84.6	5 (2)	57,250	4	99.7	0	40	5
	IC	44.9	6	74,305	58.9	54 (3)	16,135	69.2	16 (3)	3,389	59.6	7	85,596	18	71.9	9	102,945	25
	SS	29.5	0	191	89.8	26	377	93.9	10	91	94.7	1	30	77	100.0	0	0	38
5	UN	85.8	0	48	83.8	2	67	90.3	2	37	84.4	0	48	8	90.6	0	19	2
	WC	44.1	0	1,123	99.2	7	178	99.3	9	187	99.3	0	46	55	99.3	0	59	67
	SC	45.3	19	197,428	98.3	32 (4)	1,913	98.3	35 (4)	1,998	98.3	7	34,581	97	98.3	6	28,408	92
	IC	58.0	0	892	88.0	14	824	90.4	16	815	88.0	0	688	38	90.2	0	261	41
	SS	6.1	3	15,987	93.1	39	596	93.7	37	497	93.2	2	1,807	118	93.1	2	1,723	116
10	UN	92.6	0	45	92.6	3	75	95.2	4	63	92.5	0	53	11	99.9	0	28	6
	WC	33.0	9	32,312	97.9	17	284	98.1	16	229	97.8	2	2,186	124	98.0	2	949	123
	SC	28.6	122 (1)	317,002	99.3	45 (1)	1,418	99.3	42 (2)	957	99.3	11	3,705	302	99.3	16	4,842	320
	IC	61.5	0	1,354	90.7	19	968	91.3	26	1,123	90.7	0	1,081	41	91.7	0	755	43
	SS	9.5	118	576,049	96.5	47	527	97.5	32	402	96.6	3	2,829	116	96.4	2	1,482	116
Average		51.7	18 (1)	79,541	86.5	20 (12)	1,502	92.1	17 (13)	715	87.2	3 (2)	12,066	68	94.1	3	9,438	67

Note. UN, uncorrelated; WC, weakly correlated; SC, strongly correlated; IC, inverse strongly correlated; SS, subset sum.

that CPLEX default cuts might not be effective for robust mixed 0-1 problems in terms of bound improvements. Compared with the other methods, extended polymatroid cuts improved the gap values evenly without any significant differences for different instance types or different Γ values. Regardless of the CPLEX default cuts, the proposed cuts improved the gap values highly, especially when Γ = 5 and 10. On average, the branch-and-cut method with extended polymatroid cuts and CPLEX default cuts could solve more instances to optimality and reduce the average computation time relative to the other methods. The extended formulation of Atamtürk (2006) also could improve gap values significantly compared with CPLEX, but the computation time increased rapidly as the problem size increased. Because of the size of the reformulation, the

Table 2. Robust 0-1 Knapsack Problem with n = 500

					Extended formulation of Atamtürk (2006)						Extended polymatroid cuts							
		CPLEX			Without CPLEX default cuts			With CPLEX default cuts			Without CPLEX default cuts				With CPLEX default cuts			
Γ	Type	Gap (%)	Time	Nodes	Gap (%)	Time	Nodes	Gap (%)	Time	Nodes	Gap (%)	Time	Nodes	Cuts	Gap (%)	Time	Nodes	Cuts
1	UN	83.5	0	12	42.5	8	55	88.9	25	28	43.5	0	28	1	89.6	0	8	1
	WC	82.7	0	23	96.8	16	77	99.4	37	37	97.9	0	55	8	99.6	0	10	6
	SC	93.6	0 (1)	0	72.4	121 (7)	8,503	77.5	21 (7)	3	72.8	69 (6)	535,779	1	99.0	0 (1)	0	1
	IC	36.8	56 (2)	381,729	66.3	48 (7)	2435	78.1	65 (5)	1160	68.4	36 (7)	215,012	41	70.1	29 (3)	163,965	51
	SS	8.5	1	1,051	93.1	193 (3)	28	70.8	239 (3)	28	97.3	12	1	236	100.0	1	0	36
5	UN	90.8	0	30	88.6	37	81	91.9	51	29	88.6	0	50	8	94.2	0	16	4
	WC	32.9	1	2,132	99.6	73	85	99.7	106	144	99.8	1	61	63	99.9	1	24	56
	SC	33.1	64 (4)	432,667	97.6	43 (7)	111	97.6	57 (7)	21	97.7	51 (4)	304,359	31	98.0	38 (2)	172,730	97
	IC	32.5	2	6,111	90.9	270 (5)	1549	92.1	56 (8)	13	90.9	3	7,866	52	91.2	2	6,605	54
	SS	1.4	229 (1)	765,524	91.6	(10)	—	91.8	(10)	—	91.6	236 (4)	37,230	254	91.6	253 (4)	49,749	252
10	UN	95.8	0	54	94.6	46	91	96.2	62	57	95.1	0	40	20	97.4	0	19	9
	WC	22.4	152	465,944	99.9	137	137	99.9	172	118	99.9	4	57	176	100.0	4	10	167
	SC	15.0	190 (5)	835,491	99.2	37 (9)	0	99.2	42 (9)	0	99.2	244 (4)	585,375	215	99.2	42 (5)	99,150	213
	IC	27.5	24	51,435	97.0	201 (6)	318	97.0	399 (5)	1417	97.0	3	3,515	87	97.4	2	858	83
	SS	1.4	(10)	—	95.8	(10)	—	95.8	(10)	—	96.1	65 (4)	4,684	356	96.1	67 (4)	4,845	360
Average		43.9	44 (23)	173,110	88.4	82 (64)	518	93.2	104 (64)	201	89.0	35 (29)	72,124	103	94.9	21 (19)	26,171	93

Note. UN, uncorrelated; WC, weakly correlated; SC, strongly correlated; IC, inverse strongly correlated; SS, subset sum.

Table 3. Robust Mixed 0-1 Knapsack Problem with $n = 200$ and $m = 50$

					Extended formulation of Atamtürk (2006)						Extended polymatroid cuts							
		CPLEX			Without CPLEX default cuts			With CPLEX default cuts			Without CPLEX default cuts				With CPLEX default cuts			
Γ	Type	Gap (%)	Time	Nodes	Gap (%)	Time	Nodes	Gap (%)	Time	Nodes	Gap (%)	Time	Nodes	Cuts	Gap (%)	Time	Nodes	Cuts
1	UN	100.0	0	0	74.1	0	2	98.1	1	0	80.8	0	2	1	100.0	0	0	1
	WC	84.8	0	11	99.7	0	3	100.0	1	0	99.8	0	3	12	99.9	0	3	8
	SC	100.0	0	0	99.8	1	26	99.9	1	17	99.8	0	17	4	100.0	0	0	2
	IC	93.9	0	8	96.0	0	4	99.4	1	2	97.1	0	3	14	97.9	0	1	11
	SS	100.0	0	0	93.6	6	8	93.6	7	5	92.5	1	26	62	99.3	0	0	9
5	UN	98.8	0	1	95.5	1	2	99.5	1	1	98.9	0	2	6	100.0	0	0	2
	WC	52.2	0	529	99.9	1	4	99.9	1	2	99.6	1	14	70	99.7	1	10	80
	SC	40.9	0	833	100.0	1	10	100.0	2	10	99.9	1	17	122	99.9	1	15	111
	IC	79.7	0	33	97.7	1	9	98.8	2	4	95.9	0	38	22	97.9	0	18	20
	SS	16.1	1	3,630	90.9	10	98	93.8	12	88	90.7	2	919	87	93.4	2	1,255	90
10	UN	99.4	0	2	99.2	1	3	99.9	1	2	99.2	0	8	15	99.9	0	1	5
	WC	40.8	2	5,220	99.9	1	3	99.9	2	2	99.3	1	167	106	100.0	1	70	97
	SC	32.5	30	80,236	100.0	1	2	100.0	2	1	100.0	7	10	267	96.1	5	6	245
	IC	79.1	0	75	98.1	1	10	99.2	2	7	95.4	0	117	45	94.5	0	34	40
	SS	9.3	9	52,324	95.5	10	95	96.7	11	79	94.8	11	9,265	116	98.5	8	5,720	110
Average		68.5	3	9,527	96.0	2	18	98.6	3	15	96.2	2	707	63	98.5	1	476	55

Note. UN, uncorrelated; WC, weakly correlated; SC, strongly correlated; IC, inverse strongly correlated; SS, subset sum.

extended formulation of Atamtürk (2006) appeared to be unable to solve the instances that needed to explore a large number of nodes within the time limit. For instances solved within the time limit, it took a longer computation time even though the number of explored nodes was smaller than for other methods. Furthermore, because this method needs $O(n^2)$ additional constraints for each uncertain constraint, it can be burdensome to apply its reformulation to solve general robust problems with multiple uncertain constraints. Note that the results vary greatly depending on the instance type. In particular, it seems that the SC- and SS-type instances are hard instances (for hard deterministic 0-1 knapsack problems, see Pisinger 2005). These instances have longer computation times, and more extended polymatroid cuts are added than for other instance types.

The results for the robust mixed 0-1 knapsack problem (Tables 3 and 4) are similar to previous results (Tables 1 and 2). The robust mixed 0-1 knapsack problem is easier in general than the robust 0-1 knapsack problem because additional continuous variables have the effect of reducing the integrality gap values, as mentioned in Atamtürk (2006). When extended polymatroid cuts were applied or the extended formulation of Atamtürk (2006) was used, the gap values were greatly improved compared with the CPLEX default branch-and-cut method. In addition, the branch-and-cut method with extended polymatroid cuts reduced the average computation time relative to the other methods when they were used with CPLEX default cuts. Here the extended formulation of Atamtürk (2006) had a smaller average number of nodes than the branch-and-cut method with extended polymatroid cuts. However, as can be seen in Table 4, the extended formulation of Atamtürk (2006) took more computation time to solve the subproblem at each node of the branch-and-cut tree and could not solve more instances within the time limit than the other methods, especially for SS-type instances.

4.2. Results on the Multidimensional Robust 0-1 Knapsack Problem

We also report the computational results on the multidimensional robust 0-1 knapsack problem

$$\begin{aligned} \max \quad & \sum_{i \in N} p_i x_i \\ \text{s.t.} \quad & \sum_{i \in N} a_{ij} x_i + \max_{R \subseteq N, |R| \le \Gamma} \sum_{i \in R} d_{ij} x_i \le b_j, \quad \forall j \in K, \\ & x \in \mathbb{B}^n, \end{aligned} \tag{30}$$

where $K = \{1, \ldots, k\}$. We used five instance types (UN, WC, SC, IC, and SS), as used in Section 4.1. We set the density to 0.25 for each robust 0-1 knapsack constraint. In other words, for each $i \in N$ and $j \in K$, a_{ij} was

Table 4. Robust Mixed 0-1 Knapsack Problem with $n = 500$ and $m = 125$

					Extended formulation of Atamtürk (2006)						Extended polymatroid cuts							
		CPLEX			Without CPLEX default cuts			With CPLEX default cuts			Without CPLEX default cuts				With CPLEX default cuts			
Γ	Type	Gap (%)	Time	Nodes	Gap (%)	Time	Nodes	Gap (%)	Time	Nodes	Gap (%)	Time	Nodes	Cuts	Gap (%)	Time	Nodes	Cuts
1	UN	100.0	0	0	80.9	12	1	99.8	26	0	90.9	0	1	1	100.0	0	0	0
	WC	99.0	0	2	99.7	12	3	99.9	25	1	99.8	0	3	5	99.9	0	2	4
	SC	100.0	0	0	100.0	12	1	100.0	16	0	100.0	0	1	2	100.0	0	0	1
	IC	83.0	0	7	98.5	12	2	100.0	27	1	99.1	0	3	33	98.9	0	2	19
	SS	90.2	0	1	95.3	266 (3)	28	95.3	270 (4)	5	96.2	15	165	196	99.5	5	2	110
5	UN	98.9	0	2	98.3	13	5	99.2	29	2	98.9	0	5	7	99.5	0	2	1
	WC	28.5	1	1,501	100.0	13	3	100.0	29	3	100.0	2	4	89	100.0	2	5	90
	SC	33.6	0	367	100.0	19	65	100.0	26	25	100.0	0	69	39	100.0	0	44	33
	IC	37.1	0	401	99.3	15	5	99.5	35	4	98.9	1	78	59	98.4	1	51	55
	SS	3.0	19	88,918	92.7	345 (6)	60	93.2	361 (7)	41	93.3	39	12,855	176	91.9	38	10,152	173
10	UN	99.9	0	1	99.7	13	2	99.9	28	1	99.9	0	3	19	100.0	0	0	3
	WC	23.7	81	169,971	100.0	19	2	100.0	34	2	99.9	6	6	200	99.9	7	6	201
	SC	17.4	138	335,275	100.0	22	9	100.0	33	9	100.0	9	23	258	100.0	10	19	268
	IC	27.7	4	6,938	99.2	16	8	99.7	42	7	98.7	4	183	130	98.2	4	172	126
	SS	1.5	315 (6)	1583,780	94.8	283 (8)	120	94.9	366 (7)	147	95.1	134 (3)	14,094	464	93.9	146 (3)	13,537	457
Average		56.2	26 (6)	85,896	97.2	42 (17)	13	98.8	55 (18)	9	98.0	12 (3)	1,583	112	98.7	12 (3)	1,356	103

Note. UN, uncorrelated; WC, weakly correlated; SC, strongly correlated; IC, inverse strongly correlated; SS, subset sum.

generated to have a positive value with a probability of 0.25 according to its instance type. When a_{ij} had a positive value, d_{ij} was randomly generated in $[0, 100 - a_{ij}]$. Otherwise, $a_{ij} = 0$ and $d_{ij} = 0$. For each $j \in K$, we used $b_j = \lfloor \sum_{i \in N} a_{ij}/2 \rfloor$. We report average results of 10 random instances. We compared the CPLEX default branch-and-cut method, the extended formulation of Atamtürk (2006), and the branch-and-cut method with extended polymatroid cuts. The most violated extended polymatroid cut of each constraint was added. CPLEX default cuts were also added when the extended polymatroid cuts were added and the extended formulation was solved. We report the average gap value (the gap value when CPLEX default cuts were not added is given in parentheses), the average computation time (the number of unsolved instances within the time limit is given in parentheses), the average number of explored nodes, and the number of added extended

Table 5. Multidimensional Robust 0-1 Knapsack Problem with $n = 100$ and $k = 5$

		CPLEX			Extended formulation of Atamtürk (2006)			Extended polymatroid cuts			
Γ	Type	Gap (%)	Time	Nodes	Gap (%)	Time	Nodes	Gap (%)	Time	Nodes	Cuts
1	UN	94.5	0	29	50.0 (16.3)	9	1,098	96.1 (16.0)	0	10	7
	WC	69.5	0	1,153	52.9 (45.6)	31	5,375	81.1 (46.0)	0	542	33
	SC	69.4	0	445	60.2 (53.3)	20	3,174	84.5 (53.5)	0	233	35
	IC	75.6	0	443	48.0 (27.2)	24	3,769	80.1 (27.5)	0	541	25
	SS	68.6	0	817	58.1 (45.5)	9	1,160	80.7 (45.4)	0	422	33
5	UN	45.6	1	3,511	53.5 (46.5)	21	2,513	86.1 (46.8)	0	229	50
	WC	58.0	2	6,292	52.1 (40.5)	33	3,496	87.8 (40.6)	0	174	72
	SC	52.6	5	18,384	54.7 (50.1)	79	8,178	85.7 (50.2)	1	906	90
	IC	40.9	2	7,431	37.3 (26.6)	52	5,800	85.6 (26.6)	0	123	62
	SS	50.5	2	4,887	45.2 (37.0)	34	3,561	91.4 (37.2)	0	41	70
10	UN	43.4	1	3,845	36.3 (20.4)	28	2,936	89.1 (20.2)	0	52	37
	WC	56.8	1	1,373	36.1 (8.8)	20	1,701	92.4 (9.0)	0	49	37
	SC	48.3	1	2,484	25.4 (12.2)	63	4,840	92.0 (12.5)	0	23	53
	IC	69.9	0	340	26.4 (3.1)	27	3,171	96.9 (3.7)	0	14	26
	SS	50.3	0	815	26.3 (5.1)	23	1,931	97.1 (4.6)	0	8	30
Average		59.6	1	3,483	44.2 (29.2)	31	3,514	88.4 (29.3)	0	224	44

Note. UN, uncorrelated; WC, weakly correlated; SC, strongly correlated; IC, inverse strongly correlated; SS, subset sum.

Table 6. Multidimensional Robust 0-1 Knapsack Problem with n = 100 and k = 10

		CPLEX			Extended formulation of Atamtürk (2006)			Extended polymatroid cuts			
Γ	Type	Gap (%)	Time	Nodes	Gap (%)	Time	Nodes	Gap (%)	Time	Nodes	Cuts
1	UN	78.9	0	602	29.7 (18.4)	98	6,368	81.4 (18.0)	0	301	16
	WC	58.1	6	9,759	42.1 (35.8)	290 (1)	14,965	68.0 (35.6)	3	4,186	57
	SC	53.7	32	60,453	44.1 (42.4)	330 (5)	14,033	63.1 (42.3)	16	23,194	60
	IC	59.1	4	5,525	29.6 (21.0)	253 (3)	14,071	65.1 (21.1)	3	3,318	48
	SS	56.2	9	14,323	36.7 (32.6)	263 (3)	12,559	66.0 (32.3)	5	6,522	63
5	UN	36.8	24	45,473	37.5 (35.9)	261 (5)	10,297	72.4 (35.9)	4	3,123	114
	WC	36.3	77	93,396	33.4 (30.9)	342 (7)	9,095	72.9 (31.0)	9	4,001	170
	SC	36.9	123 (4)	122,547	39.0 (37.3)	(10)	—	64.4 (37.5)	78	42,041	191
	IC	27.9	51 (2)	65,176	18.4 (15.0)	470 (9)	11,411	67.9 (15.1)	10	5,628	139
	SS	33.5	87 (1)	83,154	30.8 (27.4)	403 (8)	7,442	65.4 (27.4)	18	8,280	186
10	UN	36.2	14	22,105	20.9 (12.9)	294 (2)	9,078	79.5 (12.7)	2	901	90
	WC	41.5	13	11,923	13.1 (4.3)	274 (6)	6,793	82.3 (4.4)	1	294	126
	SC	30.8	36	31,444	13.6 (6.9)	405 (9)	7,817	71.7 (7.0)	2	829	132
	IC	55.7	8	7,000	11.3 (1.5)	298 (8)	14,039	79.3 (1.8)	1	378	90
	SS	35.2	40	34,291	10.4 (3.7)	475 (7)	8,708	72.9 (3.8)	3	1,205	129
Average		45.1	32 (7)	37,539	27.4 (21.7)	274 (83)	10,773	71.5 (21.7)	10	6,947	107

Note. UN, uncorrelated; WC, weakly correlated; SC, strongly correlated; IC, inverse strongly correlated; SS, subset sum.

polymatroid cuts. When some instances out of 10 random instances were not solved within the time limit, we calculated the average computation time and the average number of nodes of solved instances.

We report the results when $n = 100, 200$ and $k = 5, 10$ in Tables 5–8. Naturally, the computation time and the number of explored nodes increase as the number of variables and constraints increase. In contrast with the results of Section 4.1, CPLEX default cuts also played an important role in the branch-and-cut methods for general problems with multiple robust constraints. When CPLEX defaults cuts were not added, the extended formulation and the branch-and-cut method with extended polymatroid cuts could not improve the gap values significantly compared with CPLEX. However, as can be seen, extended polymatroid cuts improved the gap values significantly when they were used with CPLEX default cuts. Extended polymatroid cuts could reduce the computation time and the number of explored nodes substantially. For larger-sized instances, CPLEX could not solve numerous instances within the time limit, and our method could solve more instances within the time limit than CPLEX. This shows the effectiveness of the proposed cuts when they are used for

Table 7. Multidimensional Robust 0-1 Knapsack Problem with n = 200 and k = 5

		CPLEX			Extended formulation of Atamtürk (2006)			Extended polymatroid cuts			
Γ	Type	Gap (%)	Time	Nodes	Gap (%)	Time	Nodes	Gap (%)	Time	Nodes	Cuts
1	UN	79.4	0	300	34.0 (19.8)	122	4,733	84.0 (20.5)	0	180	5
	WC	67.2	4	11,837	62.7 (59.7)	213 (3)	7,590	77.9 (59.9)	3	7,735	25
	SC	66.4	13	44,611	69.3 (63.8)	185 (9)	7,211	75.6 (63.6)	11	28,670	31
	IC	63.5	3	9,426	50.5 (40.1)	346 (5)	18,089	68.0 (40.0)	4	10,521	13
	SS	83.8	21	101,301	88.0 (87.6)	321 (5)	17,821	89.8 (87.6)	21	91,547	14
5	UN	64.4	3	9,257	69.1 (65.1)	124 (3)	2,876	89.9 (65.2)	1	479	41
	WC	72.1	79	190,710	76.2 (74.3)	307 (9)	8,829	84.0 (74.3)	24	27,436	91
	SC	80.3	75	166,862	85.0 (84.5)	(10)	—	88.7 (84.5)	39	46,082	114
	IC	51.1	42	124,669	52.8 (46.9)	326 (8)	10,875	75.3 (46.8)	8	10,976	75
	SS	75.0	50	156,756	77.3 (76.2)	312 (8)	9,014	82.5 (76.1)	37	79,455	74
10	UN	68.5	16	45,317	70.8 (68.5)	229 (8)	4,855	89.1 (68.6)	2	1,545	95
	WC	62.4	65	128,914	64.2 (62.4)	283 (7)	4,261	82.7 (62.6)	14	12,189	133
	SC	73.7	120	167,801	74.3 (73.7)	(10)	—	83.9 (73.7)	41	24,597	156
	IC	39.6	50	125,830	44.9 (39.6)	397 (7)	8,194	75.2 (39.6)	3	2,433	102
	SS	58.7	118	234,740	59.6 (58.7)	(10)	—	76.3 (58.6)	48	40,497	154
Average		67.1	44	101,222	65.2 (61.4)	233 (102)	8,397	81.5 (61.4)	17	25,623	75

Note. UN, uncorrelated; WC, weakly correlated; SC, strongly correlated; IC, inverse strongly correlated; SS, subset sum.

Table 8. Multidimensional Robust 0-1 Knapsack Problem with $n = 200$ and $k = 10$

		CPLEX			Extended formulation of Atamtürk (2006)			Extended polymatroid cuts			
Γ	Type	Gap (%)	Time	Nodes	Gap (%)	Time	Nodes	Gap (%)	Time	Nodes	Cuts
1	UN	57.9	29	40,679	21.6 (12.5)	(10)	—	56.9 (13.0)	30	40,096	10
	WC	55.1	190 (1)	229,640	53.0 (50.7)	(10)	—	65.8 (50.7)	139	124,188	60
	SC	52.7	271 (2)	320,643	51.0 (47.9)	(10)	—	60.8 (47.8)	168 (2)	133,704	56
	IC	49.9	163 (2)	191,634	31.5 (27.3)	(10)	—	55.7 (27.2)	152 (1)	150,633	35
	SS	51.0	157 (3)	185,770	50.3 (46.7)	(10)	—	57.9 (47.0)	145 (2)	139,197	54
5	UN	53.9	240 (8)	316,360	56.0 (55.1)	(10)	—	72.5 (55.0)	86	51,752	107
	WC	61.4	(10)	—	64.8 (64.0)	(10)	—	74.5 (64.0)	327 (8)	99,005	196
	SC	70.1	(10)	—	74.2 (73.8)	(10)	—	79.6 (73.8)	(10)	—	246
	IC	42.7	(10)	—	42.6 (41.5)	(10)	—	63.3 (41.5)	282 (7)	118,238	163
	SS	60.0	(10)	—	62.2 (61.8)	(10)	—	73.2 (61.8)	365 (7)	105,861	206
10	UN	57.1	(10)	—	60.2 (59.7)	(10)	—	68.9 (59.7)	301 (3)	129,599	204
	WC	51.4	356 (8)	183,988	52.5 (52.2)	(10)	—	77.1 (52.2)	187 (7)	40,536	266
	SC	61.1	(10)	—	63.7 (63.4)	(10)	—	73.3 (63.4)	(10)	—	337
	IC	31.0	(10)	—	31.4 (29.3)	(10)	—	64.1 (29.3)	172 (1)	43,720	213
	SS	48.3	(10)	—	50.8 (50.0)	(10)	—	69.8 (50.0)	(10)	—	260
Average		53.6	169 (104)	192,888	51.1 (49.1)	(150)	—	67.6 (49.1)	144 (68)	86,868	161

Note. UN, uncorrelated; WC, weakly correlated; SC, strongly correlated; IC, inverse strongly correlated; SS, subset sum.

more general robust discrete optimization problems. By contrast, the extended formulation of Atamtürk (2006) could not improve the gap values more than CPLEX even when CPLEX default cuts were also added. In addition, the extended formulation took much more time to solve the problems because of the size of the reformulation. For example, the extended formulation could not solve all instances when $n = 200$ and $k = 10$, whereas our branch-and-cut method could solve more than half.

5. Conclusion

In this paper, we define extended polymatroid inequalities and propose a polynomial-time separation algorithm. We prove that proposed extended polymatroid inequalities can describe completely the convex hull of the robust continuous 0-1 knapsack solution set in the original space. In addition, we propose a polynomial-time algorithm for the robust continuous 0-1 knapsack problem itself. In our computations, extended polymatroid inequalities were applied to robust (mixed) 0-1 knapsack problems and the multidimensional robust 0-1 knapsack problem. We showed that extended polymatroid cuts could improve the gap values remarkably compared with CPLEX default cuts. The proposed cuts were also effective when they were applied to general robust discrete optimization problems with multiple constraints. Overall, the branch-and-cut method with extended polymatroid inequalities could improve the gap values and reduce the average computation time relative to other branch-and-cut methods.

References

Atamtürk A (2006) Strong formulations of robust mixed 0–1 programming. *Math. Programming* 108(2):235–250.
Atamtürk A, Bhardwaj A (2018) Network design with probabilistic capacities. *Networks* 71(1):16–30.
Atamtürk A, Narayanan V (2008) Polymatroids and mean-risk minimization in discrete optimization. *Oper. Res. Lett.* 36(5):618–622.
Baumann F, Berckey S, Buchheim C (2013) Exact algorithms for combinatorial optimization problems with submodular objective functions. Jünger M, Reinelt G, eds. *Facets of Combinatorial Optimization* (Springer, Berlin), 271–294.
Ben-Tal A, Nemirovski A (1999) Robust solutions of uncertain linear programs. *Oper. Res. Lett.* 25(1):1–13.
Ben-Tal A, El Ghaoui L, Nemirovski A (2009) *Robust Optimization* (Princeton University Press, Princeton, NJ).
Bertsimas D, Sim M (2003) Robust discrete optimization and network flows. *Math. Programming* 98(1):49–71.
Bertsimas D, Sim M (2004) The price of robustness. *Oper. Res.* 52(1):35–53.
Bertsimas D, Brown DB, Caramanis C (2011) Theory and applications of robust optimization. *SIAM Rev.* 53(3):464–501.
Birge JR, Louveaux F (2011) *Introduction to Stochastic Programming* (Springer, New York).
Dantzig GB (1955) Linear programming under uncertainty. *Management Sci.* 1(3–4):197–206.
Edmonds J (1970) Submodular functions, matroids, and certain polyhedra. Guy R, Hanani H, Sauer N, Schönheim J, eds. *Combinatorial Structures and Their Applications* (Gordon and Breach, New York), 69–87.
Fischetti M, Monaci M (2012) Cutting plane vs. compact formulations for uncertain (integer) linear programs. *Math. Programming Comput.* 4(3):239–273.
Gabrel V, Murat C, Thiele A (2014) Recent advances in robust optimization: An overview. *Eur. J. Oper. Res.* 235(3):471–483.
Grötschel M, Lovász L, Schrijver A (1981) The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica* 1(2):169–197.

Joung S, Park S (2018) Lifting and separation of robust cover inequalities. *Networks* 72(2):272–305.

Kall P, Wallace SW (1994) *Stochastic Programming* (John Wiley & Sons, Chichester, UK).

Klopfenstein O, Nace D (2012) Cover inequalities for robust knapsack sets—Application to the robust bandwidth packing problem. *Networks* 59(1):59–72.

Kutschka M (2013) Robustness concepts for knapsack and network design problems under data uncertainty. Unpublished doctoral thesis, Rheinisch-Westfälische Technische Hochschule Aachen University, Aachen, Germany.

Lee C, Lee K, Park K, Park S (2012) Technical note: Branch-and-price-and-cut approach to the robust network design problem without flow bifurcations. *Oper. Res.* 60(3):604–610.

Lee T, Kwon C (2014) A short note on the robust combinatorial optimization problems with cardinality constrained uncertainty. *4OR* 12(4):373–378.

Marchand H, Wolsey LA (1999) The 0-1 knapsack problem with a single continuous variable. *Math. Programming* 85(1):15–33.

Miller AJ, Nemhauser GL, Savelsbergh MW (2000) On the capacitated lot-sizing and continuous 0–1 knapsack polyhedra. *Eur. J. Oper. Res.* 125(2):298–315.

Monaci M, Pferschy U, Serafini P (2013) Exact solution of the robust knapsack problem. *Comput. Oper. Res.* 40(11):2625–2631.

Pisinger D (2005) Where are the hard knapsack problems? *Comput. Oper. Res.* 32(9):2271–2284.

Richard JP, de Farias IR Jr, Nemhauser GL (2003a) Lifted inequalities for 0-1 mixed integer programming: Basic theory and algorithms. *Math. Programming* 98(1):89–113.

Richard JP, de Farias IR Jr, Nemhauser GL (2003b) Lifted inequalities for 0-1 mixed integer programming: Superlinear lifting. *Math. Programming* 98(1):115–143.

Schrijver A (2003) *Combinatorial Optimization: Polyhedra and Efficiency*, vol. 24 (Springer, Berlin).

Shapiro A, Dentcheva D, Ruszczyński A (2009) *Lectures on Stochastic Programming: Modeling and Theory* (Society for Industrial and Applied Mathematics, Philadelphia).

Soyster AL (1973) Technical note: Convex programming with set-inclusive constraints and applications to inexact linear programming. *Oper. Res.* 21(5):1154–1157.