Comparative analysis of linear programming relaxations for the robust knapsack problem

Seulgi Joung[1], Seyoung Oh[2*] and Kyungsik Lee[2*]

[1]Department of Industrial Engineering, Chonnam National University, 77, Yongbong-ro, Buk-gu, Gwangju, 61186, Republic of Korea.

[2*]Department of Industrial Engineering, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul, 08826, Republic of Korea.

*Corresponding author(s). E-mail(s): sayzeros@snu.ac.kr; optima@snu.ac.kr;
Contributing authors: sgjoung@jnu.ac.kr;

Abstract

In this study, we consider the robust knapsack problem defined by the model of Bertsimas and Sim [*Operations Research* 52(1) pp.35-53, 2004] where each item weight is uncertain and is defined with an interval. The problem is to choose a subset of items that is feasible for all of the cases in which up to a pre-specified number of items are allowed to take maximum weights simultaneously while maximizing the sum of profits of chosen items. Several integer optimization formulations for the problem have been proposed, however the strength of the upper bounds obtained from their LP-relaxations have not been theoretically analyzed and compared. In this paper, we establish a theoretical relationship among those formulations in terms of their LP-relaxations. Especially, we theoretically prove that previously proposed strong formulations (two extended formulations and a formulation using submodularity) yield the same LP-relaxation bound. In addition, through computational tests with benchmark instances, we analyze the trade-off between the strength of the lower bounds and the required computation time to solve the LP-relaxations. The results show that the formulation using submodularity shows competitive theoretical and computational performance.

Keywords: Robust knapsack problem, Integer optimization models, Strong formulations, Linear programming relaxations, Comparative analysis

1 Introduction

Robust optimization is a representative approach for solving optimization problems under data uncertainty (Ben-Tal et al, 2009; Bertsimas et al, 2011). This approach defines an uncertainty set for uncertain data and finds a *robust solution* that is feasible for all data in the uncertainty set. Several uncertainty sets have been proposed, such as the ellipsoidal set (Ben-Tal and Nemirovski, 1999), the polyhedral set (Bertsimas and Sim, 2004), and the permutohull uncertainty set (Bertsimas and Brown, 2009). Recently, data-driven robust optimization, which defines uncertainty sets using data, has been studied (Bertsimas et al, 2018; Chassein et al, 2019).

Bertsimas and Sim (2003, 2004) proposed a polyhedral uncertainty set, which has been used in numerous studies (Atamtürk, 2006; Fischetti and Monaci, 2012; Joung and Park, 2018, 2021; Klopfenstein and Nace, 2012; Lee et al, 2012; Solyalı et al, 2012). In addition, Bertsimas and Sim (2004) proposed a compact linear formulation using the dual of the inner maximization problem. The robust counterpart of their model maintains the linearity of the deterministic problem. Moreover, their model has been used to obtain an approximate solution to solve nonlinear stochastic programming problems (Han et al, 2016; Joung and Lee, 2020; Klopfenstein and Nace, 2008).

The knapsack problem (KP) is one of the most studied combinatorial optimization problems (Kellerer et al, 2004). The objective is to select items to maximize the profit sum while satisfying the knapsack capacity. In this paper, we focus on the robust knapsack problem (RKP) of the Bertsimas and Sim (2004) model where item weights have uncertainty. There are n items $N = \{1, 2, \dots, n\}$ with profits $p_i \in \mathbb{R}_+$ and uncertain weights $\tilde{a}_i$ for $i \in N$. The problem decides which items to put in a knapsack, which has a capacity, $b \in \mathbb{R}_+$, while maximizing the profit sum. Each uncertain weight of an item is defined using the robust model of Bertsimas and Sim (2004). For $i \in N$, $\bar{a}_i \in \mathbb{R}_+$ is the nominal value, and $d_i \in \mathbb{R}_+$ is the deviation value of $\tilde{a}_i$. The non-negative integer parameter, Γ, between 0 and n, controls the robustness of the model. Each uncertain weight $\tilde{a}_i$ is defined by an interval, $[\bar{a}_i, \bar{a}_i + d_i]$, and Γ restricts the maximum number of items that can take maximum weights simultaneously. The decision variable, $x_i \in \mathbb{B}$, is 1 if item i is chosen and 0 otherwise for all $i \in N$. Then, RKP is formulated as

$$\begin{aligned} \text{(RKP)} \quad \max \quad & \sum_{i \in N} p_i x_i \\ \text{s.t.} \quad & \boldsymbol{x} \in \mathcal{X}, \end{aligned} \tag{1}$$

where

$$\mathcal{X} = \left\{ \boldsymbol{x} \in \mathbb{B}^n : \sum_{i \in N} \bar{a}_i x_i + \max_{R \subseteq N, |R| \le \Gamma} \sum_{i \in R} d_i x_i \le b \right\}.$$

When item weights are integral, Klopfenstein and Nace (2008) and Monaci et al (2013) proposed dynamic programming algorithms to solve RKP. In this case, RKP can be solved in pseudo-polynomial time, $O(n\Gamma b)$. Also, RKP can be solved by solving multiple deterministic KPs. For instance, Bertsimas and Sim (2003) solved RKP by solving $n+1$ KPs. Lee et al (2012) proved that the number of $n+1$ can be reduced to $n-\Gamma+1$. Then, Lee and Kwon (2014) reduced the number of KPs to $\lceil (n-\Gamma)/2 \rceil + 1$.

Some previous formulations for robust optimization problems can be applied to formulate the solution set $\mathcal{X}$ of RKP. Fischetti and Monaci (2012) compared the compact formulation of Bertsimas and Sim (2004) and the cutting-plane approach where each cut represents robustness. The number of cuts is exponential, and each cut can be separated in polynomial time. Furthermore, we can apply strong formulations of Atamtürk (2006) for robust mixed binary programming with uncertain objective coefficients. Atamtürk (2006) proposed a strong formulation using disjunctive programming. This formulation has an exponential number of constraints, and the separation of each constraint can be accomplished by solving the shortest-path problem. Subsequently, they proposed an extended formulation with a polynomial number of variables and constraints. In addition, we can use the formulation of Joung and Park (2021), who studied the solution set of RKP with a single unrestricted continuous variable. They defined submodular inequalities that can be applied to RKP using the submodularity of the robust knapsack set function. They showed that submodular inequalities are effective when solving robust 0–1 programming problems with multiple robust knapsack constraints. Strong formulations applicable to RKP have been proposed as above, but no studies have yet theoretically analyzed and compared the strength of the upper bounds obtained from their LP-relaxations.

In this paper, we first show that extended formulations of Atamtürk (2006) and the formulation of Joung and Park (2021) using submodularity have the same strength in terms of the upper bounds provided by the linear programming (LP) relaxations. This result means that the strong formulation for RKP can be defined only with the original variables $\boldsymbol{x}$. Next, we compare the formulations computationally by solving LP-relaxations of the problem.

The rest of the paper is organized as follows. In Section 2, we introduce different formulations of RKP. In Section 3, we theoretically compare the different formulations with the objective values of their LP-relaxation. Finally, in Section 4, we implement the formulations and computationally compare them by solving their LP-relaxation.

2 Formulations of RKP

The problem (1) can be reformulated using strong duality of the inner maximization problem of (1) as follows:

$$\begin{aligned}
(\text{RKP-DUAL}) \quad \max \quad & \sum_{i \in N} p_i x_i \\
\text{s.t.} \quad & \sum_{i \in N} \bar{a}_i x_i + \Gamma u + \sum_{i \in N} v_i \le b, \\
& u + v_i \ge d_i x_i, \quad \forall i \in N, \\
& \boldsymbol{v} \ge \mathbf{0}, u \ge 0, \\
& \boldsymbol{x} \in \mathbb{B}^n.
\end{aligned} \tag{2}$$

Here $\boldsymbol{v} \in \mathbb{R}^n_+$ and $u \in \mathbb{R}_+$ are dual variables of the inner maximization problem. Naturally, (1) is equivalent to the following model with exponential linear constraints (Fischetti and Monaci, 2012):

$$\begin{aligned}
(\text{RKP-CUT}) \quad \max \quad & \sum_{i \in N} p_i x_i \\
\text{s.t.} \quad & \sum_{i \in N} \bar{a}_i x_i + \sum_{i \in R} d_i x_i \le b, \quad \forall R \subseteq N : |R| \le \Gamma, \\
& \boldsymbol{x} \in \mathbb{B}^n.
\end{aligned} \tag{3}$$

Atamtürk (2006) proposed strong formulations of robust mixed 0-1 programming with uncertain objective coefficients. They studied the set

$$\mathcal{Y} = \left\{ (\boldsymbol{x}, \boldsymbol{v}, u) \in \mathbb{B}^n \times \mathbb{R}^n \times \mathbb{R} : u + v_i \ge d_i x_i, \forall i \in N, \boldsymbol{v} \ge \mathbf{0}, u \ge 0 \right\},$$

hence their formulation can be used for a strong formulation of RKP. Note that $\mathcal{X} = proj_{\boldsymbol{x}}\{(\boldsymbol{x}, \boldsymbol{v}, u) \in \mathcal{Y} : \sum_{i \in N} \bar{a}_i x_i + \Gamma u + \sum_{i \in N} v_i \le b\}$. The solution set of RKP applying the approach of Atamtürk (2006) is defined with original variables $\boldsymbol{x}$ and dual variables, $\boldsymbol{v}$ and u. A subset S of N can be expressed as a tuple $\boldsymbol{\tau} = (\tau_{(1)}, \ldots, \tau_{(s)})$ according to the non-decreasing order of d_i values, where $s = |S|$. We denote $\mathcal{S}$ be a set of all such tuples of N.

Let $d_{\tau_{(0)}} = 0$. Then, $0 = d_{\tau_{(0)}} \le d_{\tau_{(1)}} \le d_{\tau_{(2)}} \le \cdots \le d_{\tau_{(s)}}$ by definition. Here, they showed that $(\boldsymbol{x}, \boldsymbol{v}, u) \in \mathbb{R}^{2n+1}_+$ feasible to (2), satisfy

$$\sum_{i \in N} \bar{a}_i x_i + \Gamma u + \sum_{i \in N} v_i \le b, \tag{4}$$

and

$$\sum_{j=1}^{s} \left(d_{\tau_{(j)}} - d_{\tau_{(j-1)}} \right) x_{\tau_{(j)}} \le u + \sum_{j=1}^{s} v_{\tau_{(j)}}, \quad \forall \boldsymbol{\tau} \in \mathcal{S}. \tag{5}$$

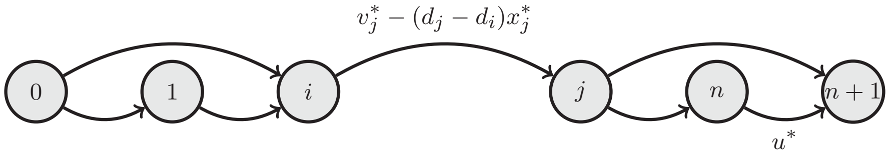

Fig. 1 Separation for (5) (Atamtürk, 2006)

The first formulation of Atamtürk (2006) is

$$\begin{aligned} \text{(RKP-Atam1)} \quad \max \quad & \sum_{i\in N} p_i x_i \\ \text{s.t.} \quad & (\boldsymbol{x}, \boldsymbol{v}, u) \in \mathcal{P}, \\ & \boldsymbol{x} \in \mathbb{B}^n, \end{aligned} \tag{6}$$

where

$$\mathcal{P} = \left\{ (\boldsymbol{x}, \boldsymbol{v}, u) \in \mathbb{R}^{2n+1} : (4), (5), \text{ and } \boldsymbol{0} \le \boldsymbol{x} \le \boldsymbol{1}, \boldsymbol{v} \ge \boldsymbol{0}, u \ge 0 \right\}.$$

Proposition 1 *(Atamtürk, 2006) Let*

$$\mathcal{P}' = \left\{ (\boldsymbol{x}, \boldsymbol{v}, u) \in \mathbb{R}^{2n+1} : (5), \textit{ and } \boldsymbol{0} \le \boldsymbol{x} \le \boldsymbol{1}, \boldsymbol{v} \ge \boldsymbol{0}, u \ge 0 \right\}.$$

Then,

$$\mathcal{P}' = conv(\mathcal{Y}).$$

Corollary 2

$$\mathcal{X} = proj_{\boldsymbol{x}}(\mathcal{P}) \cap \mathbb{B}^n.$$

Proposition 3 *(Atamtürk, 2006) The constraint, (5), can be separated by solving the shortest-path problem defined on a graph, as shown in Figure 2. The graph contains nodes from 0 to $n+1$ and arcs (i, j) for $0 \le i < j \le n+1$. For a given fractional solution, $(\boldsymbol{x}^*, \boldsymbol{v}^*, u^*)$, the length of each arc, (i, j), is $v_j^* - (d_j - d_i)x_j^*$ if $j \in [1, n]$ and u^* if $j = n+1$. If the length of the shortest path from 0 to $n+1$ is less than 0, it gives a violated inequality (5).*

Furthermore, they proposed an extended formulation of (6) with additional variables $\boldsymbol{w} \in \mathbb{R}^{n+2}$. Assume that items are sorted in non-decreasing order of d_i values. They defined the following extended set:

$$\mathcal{Q}' = \left\{ (\boldsymbol{x}, \boldsymbol{v}, u, \boldsymbol{w}) \in \mathbb{R}^{3n+3} : \begin{array}{l} (d_j - d_i)x_j + w_j - w_i \leq v_j, \quad 0 \leq i < j \leq n, \\ w_{n+1} - w_i \leq u, \quad 0 \leq i \leq n, \\ w_{n+1} - w_0 \geq 0, \\ \boldsymbol{v} \geq \boldsymbol{0}, \\ \boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{1}, \end{array} \right\}.$$

Proposition 4 *(Atamtürk, 2006)*

$$\mathcal{P}' = proj_{\boldsymbol{x},\boldsymbol{v},u}(\mathcal{Q}').$$

Let $\mathcal{Q} = \{(\boldsymbol{x}, \boldsymbol{v}, u, \boldsymbol{w}) \in \mathbb{R}^{3n+3} : (4), \text{ and } (\boldsymbol{x}, \boldsymbol{v}, u, \boldsymbol{w}) \in \mathcal{Q}'\}$.

Corollary 5

$$\mathcal{P} = proj_{\boldsymbol{x},\boldsymbol{v},u}(\mathcal{Q})$$

and

$$\mathcal{X} = proj_{\boldsymbol{x}}(\mathcal{Q}) \cap \mathbb{B}^n.$$

The second formulation of Atamtürk (2006) is

$$\begin{aligned} \text{(RKP-ATAM2)} \quad \max \quad & \sum_{i \in N} p_i x_i \\ \text{s.t.} \quad & (\boldsymbol{x}, \boldsymbol{v}, u, \boldsymbol{w}) \in \mathcal{Q}, \\ & \boldsymbol{x} \in \mathbb{B}^n. \end{aligned} \tag{7}$$

Recently, Joung and Park (2021) proposed a model for RKP with a single unrestricted continuous variable using submodularity. By setting the continuous variable to be 0, we can apply their approach to RKP. Because we consider binary variables $\boldsymbol{x}$, we can interpret $\boldsymbol{x} \in \mathbb{B}^n$ as a subset of N. For a set function $f : 2^N \to \mathbb{R}$, we use $f(X) = f(\boldsymbol{x})$ for $X \subseteq N$ and an indicator vector $\boldsymbol{x} \in \mathbb{B}^n$ of X with a slight abuse of notation. We define the robust knapsack set function for $X \subseteq N$ as

$$f(X) = \sum_{i \in X} \bar{a}_i + g(X),$$

where

$$g(X) = \max_{R \subseteq X, |R| \leq \Gamma} \sum_{i \in R} d_i.$$

The function, $g(X)$, is a submodular set function (Joung and Park, 2021; Kutschka, 2013). Let

$$\mathcal{R} = \left\{ \boldsymbol{x} \in \mathbb{R}^n : \sum_{i \in N} (\bar{a}_i + \pi_i) x_i \leq b, \forall \boldsymbol{\pi} \in \Pi_g, \boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{1} \right\},$$

and

$$\Pi_g = \left\{ \boldsymbol{\pi} \in \mathbb{R}^n : \sum_{i \in X} \pi_i \leq g(X), \forall X \subseteq N \right\}.$$

The set Π_g is called a *submodular polyhedron* related to the submodular function, g. Let $d_0 = 0$. Then, for each permutation, $\boldsymbol{\sigma} = (\sigma_{(1)}, \ldots, \sigma_{(s)})$, of $S \subseteq N$, a vector $\boldsymbol{\pi} \in \Pi_g$ can be obtained by

$$\pi_{\sigma_{(i)}} = d_{\sigma_{(i)}} - d_{\sigma_{(i)_{\min}}} \text{ for } i = 1, \ldots, s, \text{ and } \pi_i = 0, \text{ if } i \notin S, \tag{8}$$

where

$$\sigma_{(i)_{\min}} = \begin{cases} 0, & \text{if } i \leq \Gamma, \\ \arg\min_{j \in D_{i-1} \cup \{\sigma_{(i)}\}} d_j, & \text{if } i \geq \Gamma + 1 \end{cases}$$

and

$$D_i = \underset{R \subseteq \{\sigma_{(1)}, \ldots, \sigma_{(i)}\}, |R| \leq \Gamma}{\arg\max} \sum_{j \in R} d_j$$

for $i = 1, \ldots, s$. Let $D_0 = \emptyset$. The set, D_i, for $i = 1, \ldots, s$ can be updated as follows:

$$D_i = \begin{cases} D_{i-1} \cup \{\sigma_{(i)}\}, & \text{if } i \leq \Gamma, \\ D_{i-1} \cup \{\sigma_{(i)}\} \setminus \{\sigma_{(i)_{\min}}\}, & \text{if } i \geq \Gamma + 1. \end{cases} \tag{9}$$

Proposition 6 *(Edmonds, 1970; Joung and Park, 2021) The set of extreme points of Π_g, denoted as $ext(\Pi_g)$, is obtained by (8) with all permutations when $S = N$.*

Proposition 7 *For a permutation $\boldsymbol{\sigma} = (\sigma_{(1)}, \ldots, \sigma_{(s)})$ of $S \subseteq N$,*

$$\sum_{i \in N} (\bar{a}_i + \pi_i) x_i \leq b \tag{10}$$

is valid for RKP, where $\boldsymbol{\pi}$ is obtained by (8) for $\boldsymbol{\sigma}$.

Proof Let a permutation $\boldsymbol{\sigma}'$ of N be the same order as $\boldsymbol{\sigma}$ in the first s entries, and the entries after s is arbitrarily determined. Let $\boldsymbol{\pi}'$ is obtained by (8) for $\boldsymbol{\sigma}'$. By Proposition 6, $\boldsymbol{\pi}'$ is an extreme point of Π_g. Also, by (8), $\pi_i \leq \pi'_i$ for all $i \in N$. Hence, $\boldsymbol{\pi} \in \Pi_g$ by the definition of Π_g. Then, $\sum_{i \in X} (\bar{a}_i + \pi_i) \leq \sum_{i \in X} \bar{a}_i + g(X) \leq b$ for all $X \subseteq N$ which corresponds to an element of $\mathcal{X}$. Therefore, $\sum_{i \in N} (\bar{a}_i + \pi_i) x_i \leq b$ is valid for all $\boldsymbol{x} \in \mathcal{X}$. $\square$

Proposition 8 *Let*

$$\mathcal{R}_g = \left\{ \boldsymbol{x} \in \mathbb{R}^n : \sum_{i \in N} (\bar{a}_i + \pi_i) x_i \le b, \forall \boldsymbol{\pi} \in ext(\Pi_g), \boldsymbol{0} \le \boldsymbol{x} \le \boldsymbol{1} \right\},$$

then,

$$\mathcal{R} = \mathcal{R}_g.$$

Proof Trivially, $\mathcal{R} \subseteq \mathcal{R}_g$ because $ext(\Pi_g) \subseteq \Pi_g$. Then, we prove $\mathcal{R}_g \subseteq \mathcal{R}$. Assume that there exists $\boldsymbol{x}' \in \mathcal{R}_g \setminus \mathcal{R}$. Then, $\sum_{i \in N} (\bar{a}_i + \pi_i) x'_i \le b$ for all $\boldsymbol{\pi} \in ext(\Pi_g)$, but there exists $\boldsymbol{\pi}' \in \Pi_g \setminus ext(\Pi_g)$ such that $\sum_{i \in N} (\bar{a}_i + \pi'_i) x'_i > b$. It means that

$$\max \left\{ \sum_{i \in N} \pi_i x'_i : \pi \in \Pi_g \right\} \tag{11}$$

is greater than $b - \sum_{i \in N} \bar{a}_i x'_i$. It contradicts the fact that an optimal solution of (11) is obtained at an extreme point of Π_g (Edmonds, 1970). $\square$

Proposition 9

$$\mathcal{X} = \mathcal{R}_g \cap \mathbb{B}^n.$$

Proof This can be easily proved using the same argument with Joung and Park (2021). By Propositions 6 and 7, $\sum_{i \in N} (\bar{a}_i + \pi_i) x_i \le b, \forall \boldsymbol{\pi} \in ext(\Pi_g)$ are valid inequalities for $\mathcal{X}$. Therefore, $\mathcal{X} \subseteq \mathcal{R}_g \cap \mathbb{B}^n$. Also, for a permutation $\boldsymbol{\sigma} = (\sigma_{(1)}, \ldots, \sigma_{(n)})$ of N, $\pi_{\sigma_{(i)}} = d_{\sigma_{(i)}}$ for $i = 1, \ldots, \Gamma$ and $\pi_{\sigma_{(i)}} = d_{\sigma_{(i)}} - d_{\sigma_{(i)_{\min}}} \ge 0$ for $i = \Gamma + 1, \ldots, n$ holds for $\boldsymbol{\pi}$ obtained by (8). In other words, for each permutation $\boldsymbol{\sigma}$, $\sum_{i \in N} (\bar{a}_i + \pi_i) x_i \le b$ is stronger than or equal to $\sum_{i \in N} \bar{a}_i x_i + \sum_{i=1}^{\Gamma} d_{\sigma_{(i)}} x_{\sigma_{(i)}} \le b$, which is a constraint of (3). Thus every constraint of (3) is redundant for $\mathcal{R}_g \cap \mathbb{B}^n$ and, therefore, $\mathcal{X} \supseteq \mathcal{R}_g \cap \mathbb{B}^n$. $\square$

The formulation of Joung and Park (2021) is

$$\begin{aligned} (\text{RKP-S}{\small\text{UB}}) \quad \max \quad & \sum_{i \in N} p_i x_i \\ \text{s.t.} \quad & \boldsymbol{x} \in \mathcal{R}_g \cap \mathbb{B}^n. \end{aligned}$$

3 Comparison of formulations for RKP

In this section, we compare different linear formulations for RKP. The number of variables and functional constraints (without bound constraints) are summarized in Table 1. Our main result is a comparison between RKP-Atam1 and RKP-Sub, that is, a comparison between $\mathcal{R}$ and $\mathcal{P}$. Clearly, $conv(\mathcal{X}) \subseteq \mathcal{R}$ by Propositions 8 and 9, and $conv(\mathcal{X}) \subseteq proj_{\boldsymbol{x}}(\mathcal{P})$ by Corollary 2.

To show $\mathcal{R} \supseteq proj_{\boldsymbol{x}}(\mathcal{P})$, we propose an algorithm to find the inequalities, (4) and (5), of $\mathcal{P}$ corresponding to the given submodular inequality (10) of

Algorithm 1 Obtaining $\boldsymbol{\tau}^1, \ldots, \boldsymbol{\tau}^\Gamma$

Input: permutation $(\sigma_{(1)}, \ldots, \sigma_{(n)})$ of N
Output: Corresponding $\boldsymbol{\tau}^1, \ldots, \boldsymbol{\tau}^\Gamma$ in $\mathcal{S}$
1: Initialize: $l_i \leftarrow 0$ for $i \in N$, $D_i \leftarrow \emptyset$ for $i \in N \cup \{0\}$, $\boldsymbol{\tau}^k \leftarrow ()$ for $k = 1, \ldots, \Gamma$
2: **for** $i = 1, \ldots, \Gamma$ **do**
3:   $l_{\sigma_{(i)}} \leftarrow i$
4:   $D_i \leftarrow D_{i-1} \cup \{\sigma_{(i)}\}$
5:   $\sigma_{(i)_{\min}} \leftarrow 0$
6: **for** $i = \Gamma + 1, \ldots, n$ **do**
7:   $\sigma_{(i)_{\min}} \leftarrow \arg\min_{j \in D_i \cup \{\sigma_{(i)}\}} d_j$
8:   **if** $\sigma_{(i)_{\min}} = \sigma_{(i)}$ **then**
9:    $l_{\sigma_{(i)}} \leftarrow 0$
10:   **else**
11:    $l_{\sigma_{(i)}} \leftarrow l_{\sigma_{(i)_{\min}}}$
12:    $D_i \leftarrow D_{i-1} \cup \{\sigma_{(i)}\} \setminus \{\sigma_{(i)_{\min}}\}$
13: **for** $i \in N$ **do**
14:   **if** $l_i > 0$ **then**
15:    add i at the end of $\boldsymbol{\tau}^{l_i}$

$\mathcal{R}_g$, as the following Algorithm 1. For a given permutation $\boldsymbol{\sigma}$ of N to define (10), Algorithm 1 gives Γ tuples $\boldsymbol{\tau}^1, \ldots, \boldsymbol{\tau}^\Gamma \in \mathcal{S}$ to define inequalities (5). After the first loop (from lines 2 to 5) of Algorithm 1, we have $l_{\sigma_{(i)}} = i$ for $i = 1, \ldots, \Gamma$ and $D_\Gamma = \{\sigma_{(1)}, \ldots, \sigma_{(\Gamma)}\}$. During the second loop (from lines 6 to 12), i.e. for each $i \in [\Gamma + 1, n]$, we can see that the size of D_i remains at Γ and $\{l_j : j \in D_i\} = \{1, \ldots, \Gamma\}$ by lines 11 and 12. Notice that $l_{\sigma_{(i)}}$ is equal to l value of $\sigma_{(i)_{\min}}$ which is removed from D_i when $\sigma_{(i)}$ is added. Therefore, we obtain Γ mutually disjoint tuples, $\boldsymbol{\tau}^1, \ldots, \boldsymbol{\tau}^\Gamma$ by the last loop (from lines 13 to 15). Also, D_i and $\sigma_{(i)_{\min}}$ obtained through lines 1-12 are identical with the definitions provided in (8) and (9). Furthermore, among the items with the same label, l, the value of d_i is non-decreasing according to the order in which the items are added. In other words, entries in each $\boldsymbol{\tau}^k, k = 1, \ldots, \Gamma$ are ordered in non-decreasing order of d_i values.

Table 1 Summary of different formulations for RKP

Models	#variables	#constraints	variables
RKP-Dual (Bertsimas and Sim, 2004)	$2n+1$	$n+1$	$\boldsymbol{x}, \boldsymbol{v}, u$
RKP-Cut (Fischetti and Monaci, 2012)	n	$\sum_{k=0}^{\Gamma} \binom{n}{k}$	$\boldsymbol{x}$
RKP-Atam1 (Atamtürk, 2006)	$2n+1$	$2^n + 1$	$\boldsymbol{x}, \boldsymbol{v}, u$
RKP-Atam2 (Atamtürk, 2006)	$3n+3$	$\frac{n^2}{2} + \frac{3n}{2} + 2$	$\boldsymbol{x}, \boldsymbol{v}, u, \boldsymbol{w}$
RKP-Sub (Joung and Park, 2021)	n	$n!$	$\boldsymbol{x}$

Example 1 Let $n = 6, \Gamma = 3$ and $\boldsymbol{d} = (2, 5, 4, 3, 7, 3)$. For a permutation, $\boldsymbol{\sigma} = (1, 2, 3, 4, 5, 6)$, the submodular inequality is

$$\sum_{i=1}^{6} \bar{a}_i x_i + (2x_1 + 5x_2 + 4x_3 + x_4 + 4x_5 + 0x_6) \leq b. \tag{12}$$

By Algorithm 1, we obtain $\boldsymbol{\tau}^1 = (1, 4, 5), \boldsymbol{\tau}^2 = (2), \boldsymbol{\tau}^3 = (3)$. In Algorithm 1, $(l_1, l_2, l_3, l_4, l_5, l_6) = (1, 2, 3, 1, 1, 0)$. Then, corresponding inequalities, (4) and (5), are

$$\begin{aligned}
&\sum_{i=1}^{6} \bar{a}_i x_i + 3u + \sum_{i=1}^{6} v_i \leq b, \\
&2x_1 + x_4 + 4x_5 \leq u + v_1 + v_4 + v_5, \\
&5x_2 \leq u + v_2, \\
&4x_3 \leq u + v_3.
\end{aligned}$$

We can see that the original submodular inequality (12) can be obtained by combining resulting inequalities.

Theorem 10

$$\mathcal{R} \supseteq proj_{\boldsymbol{x}}(\mathcal{P}).$$

Proof Recall we have $\mathcal{R} = \mathcal{R}_g$ by Proposition 8. To prove $\mathcal{R}_g \supseteq proj_{\boldsymbol{x}}(\mathcal{P})$, we show that any inequality (10) of $\mathcal{R}_g$ defined with $\boldsymbol{\pi}$ obtained by (8) when $S = N$ is satisfied for all $\boldsymbol{x} \in proj_{\boldsymbol{x}}(\mathcal{P})$.
Take arbitrary permutation $\boldsymbol{\sigma}$ of N, and let $\boldsymbol{\tau}^1, \ldots, \boldsymbol{\tau}^\Gamma \in \mathcal{S}$ be outputs of Algorithm 1 with $\boldsymbol{\sigma}$ as an input. Let t^k be the size of $\boldsymbol{\tau}^k$ for $k = 1, \ldots, \Gamma$. Then, from the construction of $\boldsymbol{\tau}^1, \ldots, \boldsymbol{\tau}^\Gamma$, for $j = 1, \ldots, t^k$ and $k = 1, \ldots, \Gamma$,

$$\left(d_{\tau^k_{(j)}} - d_{\tau^k_{(j-1)}}\right) x_{\tau^k_{(j)}} = \left(d_{\sigma(l)} - d_{\sigma(l)_{\min}}\right) x_{\sigma(l)} \tag{13}$$

with $l \in N$ such that $\sigma(l) = \tau^k_{(j)}$ for $k = 1, \ldots, \Gamma$. Also, if $i \in N$ is not in any of $\boldsymbol{\tau}^k$, for $k = 1, \ldots, \Gamma$, then $\sigma(i) = \sigma(i)_{\min}$ by the construction (see lines 8-9 and 14-15 of Algorithm 1). Therefore, for $\boldsymbol{x} \in proj_{\boldsymbol{x}}(\mathcal{P})$,

$$\begin{aligned}
\sum_{i \in N} \pi_i x_i &= \sum_{i=1}^{n} \left(d_{\sigma(i)} - d_{\sigma(i)_{\min}}\right) x_{\sigma(i)} && \text{(by the definition of } \boldsymbol{\pi}) \\
&= \sum_{k=1}^{\Gamma} \sum_{j=1}^{t^k} \left(d_{\tau^k_{(j)}} - d_{\tau^k_{(j-1)}}\right) x_{\tau^k_{(j)}} && \text{(by (13))} \\
&\leq \Gamma u + \sum_{k=1}^{\Gamma} \sum_{j=1}^{t^k} v_{\tau^k_{(j)}} && \text{(by (5))} \\
&\leq \Gamma u + \sum_{i \in N} v_i && (\boldsymbol{\tau}^1, \ldots, \boldsymbol{\tau}^\Gamma \text{ are mutually disjoint and } \boldsymbol{v} \geq \mathbf{0}) \\
&\leq b - \sum_{i \in N} \bar{a}_i x_i && \text{(by (4))}
\end{aligned}$$

is satisfied. Since the choice of the permutation $\boldsymbol{\sigma}$ of N is arbitrary, for each $\boldsymbol{\pi} \in ext(\Pi_g)$, the corresponding submodular inequality, $(\bar{\boldsymbol{a}} + \boldsymbol{\pi})^T \boldsymbol{x} \leq b$, can be obtained by combining inequalities (4) and (5), as described above. Hence any $\boldsymbol{x} \in proj_{\boldsymbol{x}}(\mathcal{P})$ satisfies all submodular inequalities with $\boldsymbol{\pi} \in ext(\Pi_g)$; therefore, $\boldsymbol{x} \in \mathcal{R}_g$. Now we can conclude that $\mathcal{R} = \mathcal{R}_g \supseteq proj_{\boldsymbol{x}}(\mathcal{P})$. □

Next, we show $R \subseteq proj_{\boldsymbol{x}}(\mathcal{P})$ to conclude $R = proj_{\boldsymbol{x}}(\mathcal{P})$.

Theorem 11

$$\mathcal{R} \subseteq proj_{\boldsymbol{x}}(\mathcal{P}).$$

Proof We prove that if $\boldsymbol{x}' \notin proj_{\boldsymbol{x}}(\mathcal{P})$, then $\boldsymbol{x}' \notin \mathcal{R}$. If $\sum_{i\in N} \bar{a}_i x'_i > b$, then it is trivially satisfied. Assume that $\sum_{i\in N} \bar{a}_i x'_i \leq b$. We can then rewrite the set $proj_{\boldsymbol{x}}(\mathcal{P})$ as follows;

$$proj_{\boldsymbol{x}}(\mathcal{P}) = \left\{ \boldsymbol{x} \in \mathbb{R}^n : \sum_{i\in N} \bar{a}_i x_i + DEV(\boldsymbol{x}) \leq b, \boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{1} \right\},$$

where

$$\begin{aligned}
&DEV(\boldsymbol{x}) \\
&= \min_{\boldsymbol{v},u} \left\{ \Gamma u + \sum_{i\in N} v_i : \sum_{j=1}^{s} (d_{\tau_{(j)}} - d_{\tau_{(j-1)}}) x_{\tau_{(j)}} \leq u + \sum_{j=1}^{s} v_{\tau_{(j)}}, \forall \boldsymbol{\tau} \in \mathcal{S}, \boldsymbol{v} \geq \boldsymbol{0}, u \geq 0 \right\} \\
&= \max_{\boldsymbol{\alpha}} \left\{ \sum_{\boldsymbol{\tau}\in\mathcal{S}} \sum_{j=1}^{s} (d_{\tau_{(j)}} - d_{\tau_{(j-1)}}) x_{\tau_{(j)}} \alpha_{\boldsymbol{\tau}} : \sum_{\boldsymbol{\tau}\in\mathcal{S}} \alpha_{\boldsymbol{\tau}} \leq \Gamma, \sum_{\boldsymbol{\tau}\in\mathcal{S}:\boldsymbol{\tau}\ni i} \alpha_{\boldsymbol{\tau}} \leq 1, \forall i \in N, \boldsymbol{\alpha} \geq 0 \right\}.
\end{aligned}$$

The last equality holds by the strong duality of LP with dual variables, $\boldsymbol{\alpha}$. Note that the second term above, the minimization problem, is feasible and bounded. If $\boldsymbol{x}' \in [0,1]^n$ and $\boldsymbol{x}' \notin proj_{\boldsymbol{x}}(\mathcal{P})$, then the maximization version of $DEV(\boldsymbol{x}')$ has an optimal solution, $\boldsymbol{\alpha}'$, such that the corresponding objective function value is bigger than $b - \sum_{i\in N} \bar{a}_i x'_i$. For each tuple $\boldsymbol{\tau} = (\tau_{(1)}, \ldots, \tau_{(s)}) \in \mathcal{S}$, since $\boldsymbol{\tau}$ can be interpreted as a permutation of a subset of N, we can obtain $\boldsymbol{\pi}^{(\boldsymbol{\tau},1)} \in \Pi_g$ using (8) by setting $\Gamma = 1$. Then,

$$\pi^{(\boldsymbol{\tau},1)}_{\tau_{(j)}} = d_{\tau_{(j)}} - d_{\tau_{(j-1)}}, \quad \forall j = 1, \ldots, s$$

is the increase of the submodular function g with $\Gamma = 1$, when the item, $\tau_{(j)}$, is added to the set, $\{\tau_{(1)}, \tau_{(2)}, \ldots, \tau_{(j-1)}\}$. We denote $\boldsymbol{\tau} \cap T$ as the tuple in $\mathcal{S}$ consisting of common elements of the set T and the tuple $\boldsymbol{\tau}$. For any $T \subseteq N$ and any feasible $\boldsymbol{\alpha}'$,

$$\begin{aligned}
\sum_{i\in T} \sum_{\boldsymbol{\tau}\in\mathcal{S}:\boldsymbol{\tau}\ni i} \alpha'_{\boldsymbol{\tau}} \pi_i^{(\boldsymbol{\tau},1)} &= \sum_{\boldsymbol{\tau}\in\mathcal{S}} \alpha'_{\boldsymbol{\tau}} \sum_{i\in\boldsymbol{\tau}\cap T} \pi_i^{(\boldsymbol{\tau},1)} \\
&\leq \sum_{\boldsymbol{\tau}\in\mathcal{S}} \alpha'_{\boldsymbol{\tau}} \sum_{i\in\boldsymbol{\tau}\cap T} \pi_i^{(\boldsymbol{\tau}\cap T,1)} && (14) \\
&= \sum_{\boldsymbol{\tau}\in\mathcal{S}} \alpha'_{\boldsymbol{\tau}} \cdot d_{\max}^{\boldsymbol{\tau}\cap T} && (15) \\
&\leq \max_{R\subseteq T, |R|\leq\Gamma} \sum_{i\in T} d_i = g(T), && (16)
\end{aligned}$$

where

$$d_{\max}^{\boldsymbol{\tau}\cap T} = \begin{cases} \max_{i\in\boldsymbol{\tau}\cap T} d_i, & \text{if } \boldsymbol{\tau}\cap T \neq \emptyset, \\ 0, & \text{otherwise.} \end{cases}$$

(14) and (15) hold by the definition of $\pi_i^{(\boldsymbol{\tau},1)}$ and the diminishing returns property (Wolsey, 1998) of the submodular function, g with $\Gamma = 1$. Also, (16) is satisfied as $\sum_{\boldsymbol{\tau}\in\mathcal{S}} \alpha'_{\boldsymbol{\tau}} \leq \Gamma$ and $\sum_{\boldsymbol{\tau}\in\mathcal{S}:\boldsymbol{\tau}\ni i} \alpha'_{\boldsymbol{\tau}} \leq 1$ for all $i \in T$. In other words,

$$\boldsymbol{\pi}^* = \left(\sum_{\boldsymbol{\tau}\in\mathcal{S}:\boldsymbol{\tau}\ni 1} \alpha'_{\boldsymbol{\tau}} \pi_1^{(\boldsymbol{\tau},1)}, \sum_{\boldsymbol{\tau}\in\mathcal{S}:\boldsymbol{\tau}\ni 2} \alpha'_{\boldsymbol{\tau}} \pi_2^{(\boldsymbol{\tau},1)}, \ldots, \sum_{\boldsymbol{\tau}\in\mathcal{S}:\boldsymbol{\tau}\ni n} \alpha'_{\boldsymbol{\tau}} \pi_n^{(\boldsymbol{\tau},1)} \right) \in \Pi_g$$

by the definition of Π_g. Here,

$$\sum_{i\in N} \pi_i^* x'_i = \sum_{\boldsymbol{\tau}\in\mathcal{S}} \sum_{j=1}^{s} (d_{\tau(j)} - d_{\tau(j-1)}) x'_{\tau(j)} \alpha'_{\boldsymbol{\tau}} = DEV(\boldsymbol{x}') > b - \sum_{i\in N} \bar{a}_i x'_i.$$

Therefore, if $\boldsymbol{x}' \notin proj_{\boldsymbol{x}}(\mathcal{P})$, then $\boldsymbol{x}' \notin \mathcal{R}$. □

Corollary 12

$$\mathcal{R} = proj_{\boldsymbol{x}}(\mathcal{P}).$$

Let z_{LP}^{model} be the optimal value of the LP-relaxation of the *model*. Then, the following corollary holds by above results:

Corollary 13 $z_{LP}^{\text{RKP-Dual}} = z_{LP}^{\text{RKP-Cut}} \geq z_{LP}^{\text{RKP-Atam1}} = z_{LP}^{\text{RKP-Atam2}} = z_{LP}^{\text{RKP-Sub}}$.

4 Computational results

In this section, we computationally compare different formulations (RKP-Dual, RKP-Atam1, RKP-Atam2, and RKP-Sub) by solving their LP-relaxation. The tests were conducted using a 3.40-GHz Intel Xeon E3-1240 CPU with 8-GB RAM. All models were implemented in Java and CPLEX 20.1. The time limit was 1800 seconds. We tested with the following five RKP types (Joung and Park, 2021). Here, $\bar{a}_i, p_i$, and d_i are non-negative integers for each $i \in N$.

- UN (Uncorrelated): $\bar{a}_i$ and p_i both are randomly generated in $[1, 100]$.
- WC (Weakly correlated): $\bar{a}_i$ and p_i are randomly generated in $[1, 100]$ and $[\max\{1, \bar{a}_i - 10\}, \bar{a}_i + 10]$, respectively.
- SC (Strongly correlated): $\bar{a}_i$ is randomly generated in $[1, 100]$ and $p_i = \bar{a}_i + 10$.
- IC (Inverse correlated): p_i is randomly selected in $[1, 100]$ and $\bar{a}_i = \min\{100, p_i + 10\}$.
- SS (Subset sum): $\bar{a}_i$ is randomly generated in $[1, 100]$ and $p_i = \bar{a}_i$.

The deviation value d_i is randomly generated in $[0, 100 - \bar{a}_i]$, and $b = \lfloor \sum_{i\in N} \bar{a}_i/2 \rfloor$. We tested when $n = 200, 500, \Gamma = 1, 5, 10$. For each combination, 10 random instances were generated, and the average results are reported. We compared the LP-relaxations of four different formulations of Table 1 except RKP-Cut, and the implementation details are as follows:

- RKP-Dual (Bertsimas and Sim, 2004): The LP-relaxation of (2) is solved by CPLEX with its default settings.
- RKP-Atam1 (Atamtürk, 2006): Atamtürk (2006) compared two separation approaches for the separation of inequalities, (5). The first approach is adding multiple violated inequalities, (5), by computing all shortest paths from node 0 to every other node, i. The second approach is finding a path of negative cost from 0 to i with the smallest number of arcs using the Bellman–Ford algorithm. Then, Atamtürk (2006) mentioned that faster implementation was obtained by the second approach. We solved the LP-relaxation of (2) and found a violated inequality, (5), using the second separation approach. We repeatedly solved the LP-relaxed problem and added a violated constraint until there was no violated inequality.
- RKP-Atam2 (Atamtürk, 2006): Sort items in non-decreasing order of d_i value. Then, solve the LP-relaxation of (7) using CPLEX with its default settings.
- RKP-Sub (Joung and Park, 2021): Solve the LP-relaxation of (2). Then, sort items in non-increasing order of x_i^* and break ties by increasing order of d_i. With this permutation, obtain the submodular inequality, (10), by (8) with $S = N$ (details are given in Joung and Park, 2021). Repeat solving the problem and adding a violated submodular inequality, (10), until there is no violated inequality.

Note that we solved the LP-relaxations of RKP. For the results of RKP with binary constraints of RKP-Dual, RKP-Atam2 and RKP-Sub, we refer the readers to Joung and Park (2021). Tables 2-3 report the average results when $n = 200$ and $n = 500$, respectively. Here, "time" is the average computational time. The number of unfinished instances within the time limit is given as "#U". The average number of added violated inequalities are given as "#cuts", and the average closed gap "%gap" is calculated as

$$\%\text{gap} = \frac{z_{LP} - z_{LP}^{model}}{z_{LP} - z_{OPT}},$$

where z_{LP} is the objective value of the LP-relaxation of the original model, RKP-Dual (2).

Here, z_{OPT} was the best objective value of RKP obtained by solving RKP-Dual (2) within the time limit (1800 seconds). By Corollary 13, %gap of RKP-Dual is 0, and %gap of other models are same. Therefore, we only report %gap of RKP-Atam1, RKP-Atam2, RKP-Sub in one column. If some instances were not finished, we then reported the closed gap within the time limit in a separate column "%gap" of the corresponding model.

As can be seen, RKP-Dual is solved significantly faster than using other methods for all instances. Other models substantially improved %gap values compared to RKP-Dual. When $\Gamma = 1$, %gap values are relatively small, but

Table 2 Comparison of different models for RKP ($n = 200$)

type	Γ	RKP-Dual	%gap	RKP-Atam1				RKP-Atam2	RKP-Sub	
		time		#U	time	#cuts	%gap	time	time	#cuts
UN	1	0.0	33.7	0	0.0	4.5	-	0.2	0.0	2.1
	5	0.0	76.9	0	0.0	32.8	-	0.3	0.0	8.1
	10	0.0	93.2	0	0.1	110.3	-	0.3	0.0	20.5
WC	1	0.0	96.9	0	0.3	464.1	-	0.2	0.0	14.6
	5	0.0	99.3	0	86.4	6074.4	-	0.2	0.1	66.9
	10	0.0	99.0	0	238.5	8622.8	-	0.3	0.1	87.1
SC	1	0.0	69.8	0	0.2	272.6	-	0.3	0.0	14.2
	5	0.0	98.7	3	654.3	15227.7	95.9	0.3	0.2	143.6
	10	0.0	99.3	10	-	18815.6	82.9	0.4	2.1	379.6
IC	1	0.0	51.0	0	0.1	190.7	-	0.3	0.0	7.1
	5	0.0	90.2	0	0.9	912.1	-	0.2	0.0	15.0
	10	0.0	91.3	0	5.4	1601.4	-	0.2	0.0	50.8
SS	1	0.0	86.2	3	1223.9	15484.7	86.0	0.4	0.1	58.9
	5	0.0	92.0	9	1621.0	15811.0	86.9	0.5	0.1	101.9
	10	0.0	94.6	10	-	14183.4	77.6	0.5	0.2	143.7
Average		0.0	84.8	2.3	157.3	6520.5	82.0	0.3	0.2	74.3

Table 3 Comparison of different models for RKP ($n = 500$)

type	Γ	RKP-Dual	%gap	RKP-Atam1				RKP-Atam2	RKP-Sub	
		time		#U	time	#cuts	%gap	time	time	#cuts
UN	1	0.0	19.4	0	0.1	3.5	-	1.0	0.0	0.7
	5	0.0	80.3	0	0.2	36.7	-	1.1	0.0	7.7
	10	0.0	95.3	0	0.2	84.3	-	1.0	0.0	15.0
WC	1	0.0	96.4	0	1.1	572.0	-	0.9	0.0	18.6
	5	0.0	99.6	5	694.1	19481.0	98.2	1.1	0.7	209.5
	10	0.0	99.8	10	-	21703.3	86.6	1.1	1.9	299.4
SC	1	0.0	65.6	0	0.4	219.6	-	1.0	0.0	11.0
	5	0.0	96.3	6	980.4	18208.4	92.5	1.1	0.2	117.9
	10	0.0	99.2	10	-	19117.2	66.4	1.2	4.4	435.7
IC	1	0.0	60.1	0	1.0	509.9	-	0.9	0.0	11.4
	5	0.0	91.6	0	34.3	2657.5	-	1.0	0.0	26.3
	10	0.0	94.6	0	149.0	4894.1	-	1.0	0.1	58.3
SS	1	0.0	88.2	10	-	17700.3	25.3	2.2	0.6	111.2
	5	0.0	92.2	10	-	16294.4	25.4	2.6	0.8	148.0
	10	0.0	95.5	10	-	15301.2	25.6	2.7	1.7	247.5
Average		0.0	84.9	4.1	104.0	9118.9	68.2	1.3	0.7	114.5

they are close to 100% when $\Gamma = 5$ or 10. RKP-Atam1 took longer to solve the LP-relaxation problem; it could not even solve some instances within the time limit. Moreover, it added a large number of cuts for some hard instances such as SC and SS. RKP-Atam2 and RKP-Sub solved the instances in a significantly shorter duration. RKP-Sub added a smaller number of cuts compared with RKP-Atam1, and it took slightly less time than RKP-Atam2 on average.

5 Conclusion

In this paper, we compared different formulations of RKP theoretically and computationally. We proved that previously proposed formulations have the same strength in terms of the objective value of the LP-relaxation. Thus, the strong formulation of RKP can be defined using only the original variables, $\boldsymbol{x}$. In computational tests, we showed that strong formulations could improve the closed gap substantially. In addition, the strong formulation on the original space was solved in a relatively short time, compared with other strong formulations.

Acknowledgments. This study was financially supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2021R1G1A1003653).

Conflict of interest. The authors declare that they have no conflict of interest.

References

Atamtürk A (2006) Strong formulations of robust mixed 0–1 programming. Mathematical Programming 108(2):235–250

Ben-Tal A, Nemirovski A (1999) Robust solutions of uncertain linear programs. Operations Research Letters 25(1):1–13

Ben-Tal A, El Ghaoui L, Nemirovski A (2009) Robust Optimization. Princeton University Press

Bertsimas D, Brown DB (2009) Constructing uncertainty sets for robust linear optimization. Operations Research 57(6):1483–1495

Bertsimas D, Sim M (2003) Robust discrete optimization and network flows. Mathematical Programming 98(1):49–71

Bertsimas D, Sim M (2004) The price of robustness. Operations Research 52(1):35–53

Bertsimas D, Brown DB, Caramanis C (2011) Theory and applications of robust optimization. SIAM Review 53(3):464–501

Bertsimas D, Gupta V, Kallus N (2018) Data-driven robust optimization. Mathematical Programming 167(2):235–292

Chassein A, Dokka T, Goerigk M (2019) Algorithms and uncertainty sets for data-driven robust shortest path problems. European Journal of Operational Research 274(2):671–686

Edmonds J (1970) Submodular functions, matroids, and certain polyhedra. in Combinatorial Structures and Their Applications, Gordon and Breach, New York pp 11–26

Fischetti M, Monaci M (2012) Cutting plane versus compact formulations for uncertain (integer) linear programs. Mathematical Programming Computation pp 1–35

Han J, Lee K, Lee C, et al (2016) Robust optimization approach for a chance-constrained binary knapsack problem. Mathematical Programming 157(1):277–296

Joung S, Lee K (2020) Robust optimization-based heuristic algorithm for the chance-constrained knapsack problem using submodularity. Optimization Letters 14(1):101–113

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

Joung S, Park S (2021) Robust mixed 0-1 programming and submodularity. INFORMS Journal on Optimization 3(2):183–199

Kellerer H, Pferschy U, Pisinger D (2004) Knapsack Problems. Springer

Klopfenstein O, Nace D (2008) A robust approach to the chance-constrained knapsack problem. Operations Research Letters 36(5):628–632

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. PhD thesis, RWTH Aachen University

Lee C, Lee K, Park K, et al (2012) Technical note–branch-and-price-and-cut approach to the robust network design problem without flow bifurcations. Operations Research 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

Monaci M, Pferschy U, Serafini P (2013) Exact solution of the robust knapsack problem. Computers & Operations Research 40(11):2625–2631

Solyalı O, Cordeau JF, Laporte G (2012) Robust inventory routing under demand uncertainty. Transportation Science 46(3):327–340

Wolsey LA (1998) Integer Programming, vol 42. Wiley New York