Covering Linear Programming with Violations

Feng Qiu, Shabbir Ahmed, Santanu S. Dey and Laurence A. Wolsey

May 8, 2012

Abstract

We consider a class of linear programs involving a set of covering constraints of which at most k are allowed to be violated. We show that this covering linear program with violation is strongly $\mathcal{NP}$-hard. In order to improve the performance of mixed-integer programming (MIP) based schemes for these problems, we introduce and analyze a coefficient strengthening scheme, adapt and analyze an existing cutting plane technique, and present a branching technique. Through computational experiments, we empirically verify that these techniques are significantly effective in improving solution times over the CPLEX MIP solver. In particular, we observe that the proposed schemes can cut down solution times from as much as six days to under four hours in some instances.

1 Introduction

A point belongs to the feasible region of a linear program (LP) if it satisfies all the linear constraints defining the LP. However, when certain problems are being modeled, the feasibility requirement is soft. That is, a point is considered feasible even if it violates no more than a specified number of the constraints defining the problem. Such a linear program is called a *k-violation linear program* (KVLP) [19]:

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & a_i^\top x \geq b_i \quad i = 1, ..., m, \\ & \text{at most } k \text{ of the } m \text{ constraints can be violated,} \\ & x \in \mathbb{R}^n_+. \end{aligned} \tag{1}$$

The feasible region of a KVLP is the union of $\binom{m}{k}$ polyhedral sets, each of which are defined by the intersection of some subset of $(m-k)$ inequalities from the m inequalities in (1). In general, such a feasible region is nonconvex and KVLP is a strongly $\mathcal{NP}$-hard optimization problem [1]. Much of the existing work on this class of problems focuses on polynomial time algorithms for low dimensional problems (i.e. n is fixed and small) (cf. [5] for a survey).

This paper addresses KVLPs where the linear system (1) consists of covering type linear inequalities, i.e., a_i and b_i are non-negative for all i. We call such a problem a *covering-type k-violation linear program* (CKVLP). CKVLPs, which are an important subclass of KVLPs, have many applications.

As a concrete example, consider a probabilistically-constrained portfolio optimization problem [16] to determine a minimum cost distribution of a unit investment among n assets with uncertain returns, requiring the overall return to be at least r with a probability of $1-\epsilon$, where $\epsilon \in (0,1)$ is a prespecified risk level. A formulation of this problem is

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & e^\top x = 1 \\ & \mathbb{P}\{\tilde{a}^\top x \geq r\} \geq 1 - \epsilon \\ & x \in \mathbb{R}^n_+, \end{aligned} \tag{2}$$

where $\tilde{a}$ is the random return vector for n assets following some known distribution, $\mathbb{P}\{A\}$ denotes the probability of the random event A, and c is the cost vector. A common approach to dealing with the probabilistic constraint in (2) is the sample average approximation method [12] where the distribution of $\tilde{a}$ is approximated by an empirical distribution corresponding to an i.i.d sample of return vectors $\{a_i\}_{i=1}^m$. The approximated problem then reads as follows:

$$\begin{array}{rl} \min & c^\top x \\ \text{s.t.} & e^\top x = 1 \\ & a_i^\top x \geq r \quad i = 1, ..., m, \\ & \text{at most } k \text{ of the } m \text{ constraints can be violated,} \\ & x \in \mathbb{R}_+^n, \end{array} \tag{3}$$

where $k = \lfloor m\epsilon \rfloor$. Since the return is non-negative and only nonnegative investments are allowed, (3) is an example of CKVLP with an additional equality constraint. In Section 6, we discuss a similar application of CKVLP in an optimal vaccine allocation under probabilistic constraints [18]. Additional applications of CKVLP arise in intensity modulated radiation therapy (IMRT) planning [20] and signal broadcasting coverage design [17].

A CKVLP can be modeled as a mixed integer program (MIP) in a straight-forward manner. First, note that if $b_i = 0$ for any $i \in \{1, \ldots, m\}$, then the corresponding inequality is redundant since then the inequality is implied by the non-negativity constraints on the x variables. Thus, we assume henceforth that $b_i > 0$ for all $i \in \{1, \ldots, m\}$ and so they can be scaled to 1. Then, an MIP formulation of CKVLP is

$$\begin{array}{rl} \min & c^\top x \\ \text{s.t.} & a_i^\top x + z_i \geq 1 \quad i = 1, ..., m, \\ & \displaystyle\sum_{i=1}^m z_i \leq k \\ & x \in \mathbb{R}_+^n, z_i \in \{0, 1\} \quad i = 1, ..., m, \end{array} \tag{4}$$

where we have introduced the binary variables z_i taking the value 1 if the i-th constraint is violated. For large scale CKVLPs, the above MIP formulation performs very poorly. The goal of this paper is to study a number of enhancement schemes to improve the computational performance of MIP-based approaches for solving CKVLPs.

We begin by studying the theoretical complexity of CKVLPs and illustrating the difficulty of solving realistic instances directly by the CPLEX MIP solver (Section 2) as well. Next, in order to improve the performance of standard solvers on the MIP model (4) of CKVLPs, we introduce and analyze a coefficient strengthening scheme (Section 3), adapt and analyze an existing cutting plane technique (Section 4), and present a branching technique (Section 5). Through computational experiments on the probabilistic portfolio optimization problem (3) and an optimal vaccination allocation problem, we empirically verify that these techniques are extremely effective in improving solution times (Section 6). In particular, we observe that the proposed schemes can cut down solution times from as much as six days to under four hours in some instances.

We close this section by pointing out that all three enhancement schemes studied here are applicable when there are additional side constraints in the MIP (4). This follows since these schemes attempt to tighten the LP relaxation of (4), which is a valid relaxation even when additional side constraints are present.

2 Difficulty of Solving CKVLP

2.1 Computational Complexity

General KVLP has been shown to be $\mathcal{NP}$-complete [1]. However, to the best of our knowledge, the complexity of CKVLP, a sub-class of KVLP, has not been addressed. In a recent paper [20], Tunçel et al. showed that a packing version KVLP is weakly $\mathcal{NP}$-hard (the linear inequalities in KVLP are packing inequalities) by reduction from the partition problem. This result can be modified to show the $\mathcal{NP}$-hardness of CKVLP. In this paper we provide a direct proof that CKVLP is strongly $\mathcal{NP}$-hard.

By complementing the binary variables z in (4), we have the following equivalent formulation of CKVLP:

$$\begin{array}{ll} \min & c^\top x \\ \text{s.t.} & Ax \geq z \\ & e^\top z \geq p \\ & x \in \mathbb{R}^n_+ \\ & z \in \{0,1\}^m, \end{array} \tag{5}$$

where $A = [a_1^\top, ..., a_m^\top] \in \mathbb{Q}^{m\times n}_+$, $c \in \mathbb{Q}^n_+$, $p = m - k$, e is the column vector with each entry equal to 1, and $\mathbb{Q}$ is the set of rationals.

To prove that CKVLP (5) is $\mathcal{NP}$-hard, we first verify that the following intermediate decision problem is $\mathcal{NP}$-complete.

> *Intermediate CKVLP Feasibility Problem*: Given $\eta \in \mathbb{Q}$, $A \in \mathbb{Q}^{m\times n}_+$ and $c \in \mathbb{Q}^n$, does there exist a solution $(x, z) \in \mathbb{R}^n_+ \times \{0, 1\}^m$ to the following system?
>
> $$\begin{array}{lcl} c^\top x - e^\top z & \leq & \eta \\ Ax & \geq & z. \end{array} \tag{6}$$

Lemma 1. *The Intermediate CKVLP Feasibility Problem (6) is strongly $\mathcal{NP}$-complete.*

Proof. Since (6) is a decision version of a mixed integer linear program, it is in $\mathcal{NP}$. In order to show that determining the feasibility of (6) is strongly $\mathcal{NP}$-complete, we polynomially reduce an arbitrary instance of the strongly $\mathcal{NP}$-complete vertex cover problem [8] to an instance of (6). An instance of the vertex cover problem is defined as follows:

> *Vertex Cover*: Given a graph $G = (V, E)$ and $q \in \mathbb{N}$, does there exist $S \subseteq V$ such that (i) $|S| \leq q$ and (ii) S is a vertex cover, that is for all $(i, j) \in E$, either $i \in S$ or $j \in S$?

Given an instance of the vertex cover problem, we construct an instance of (6) by setting $m := |V| + E|$, $n := |V|$, $\eta := q - |E|$, $c := 2e$, $A := \begin{bmatrix} H \\ I \end{bmatrix}$, where H is the node-arc incidence matrix of G and I is a $|V| \times |V|$ identity matrix. The resulting instance of (6) is then:

$$2\sum_{j\in V} x_j - \sum_{j\in V} z_j - \sum_{(i,j)\in E} y_{ij} \leq q - |E| \tag{7}$$

$$x_i + x_j \geq y_{ij} \qquad \forall\ (i, j) \in E \tag{8}$$

$$x_i \geq z_i \qquad \forall\ i \in V \tag{9}$$

$$x \in \mathbb{R}^{|V|}_+ \tag{10}$$

$$z \in \{0, 1\}^{|V|} \tag{11}$$

$$y \in \{0, 1\}^{|E|}. \tag{12}$$

Note that the size of (7)-(12) is polynomial in the encoding length of G and q. We complete the proof by showing that a vertex cover instance has an answer yes if and only if the associated system (7)-(12) has a solution.

($\Rightarrow$) Let S be a vertex cover for G such that $|S| \leq q$. Then, consider a solution $(\tilde{x}, \tilde{y}, \tilde{z}) \in \mathbb{R}^{|V|}_+ \times \{0, 1\}^{|E|} \times \{0, 1\}^{|V|}$ defined as:

$$\begin{array}{rcl} \tilde{x}_j = \tilde{z}_j & = & \begin{cases} 1 & \forall\ j \in S \\ 0 & \forall\ j \in V \setminus S, \end{cases} \\ \tilde{y}_{i,j} & = & 1\ \forall\ (i, j) \in E. \end{array}$$

The solution $(\tilde{x},\tilde{y},\tilde{z})$ satisfies (9)-(12) by construction, and since S is a vertex cover it also satisfies (8). Finally, $2\sum_{j\in V}\tilde{x}_j - \sum_{j\in V}\tilde{z}_j - \sum_{(i,j)\in E}\tilde{y}_{ij} = |S| - |E| \leq q - |E|$. Thus the system (7)-(12) has a solution.

($\Leftarrow$) Now assume that the system (7)-(12) has a solution. Note that an arbitrary feasible solution to (7)-(12) may have fractional x components that cannot be directly converted to a vertex cover for G. We show that there exists a feasible solution of (7)-(12) with integral values of x and $y = e$ whenever (7)-(12) is feasible. Towards this end, we first present some properties of feasible solutions to (7)-(12). Given $(x,y,z) \in \mathbb{R}_+^{|V|} \times \{0,1\}^{|E|} \times \{0,1\}^{|V|}$, which satisfies (8)-(12), let

$$f(x,y,z) := 2\sum_{j\in V} x_j - \sum_{j\in V} z_j - \sum_{(i,j)\in E} y_{ij},$$

i.e., if (x,y,z) is feasible for (7)-(12), then $f(x,y,z) \leq q - |E|$.

Claim a. Given (x^1,y^1,z^1) satisfying (8)-(12), there exists (x^2,y^2,z^2) satisfying (8)-(12) such that $y^2 = e$ i.e. a vector of ones, and $f(x^2,y^2,z^2) \leq f(x^1,y^1,z^1)$.

Proof of Claim a. Suppose there exists $(\tilde{i},\tilde{j}) \in E$ such that $y^1_{\tilde{i}\tilde{j}} = 0$. Construct (x^3,y^3,z^3) as follows:

$$x^3_j = \begin{cases} 1 & j = \tilde{i} \\ x^1_j & j \in V \setminus \{\tilde{i}\} \end{cases},$$

$$z^3_j = \begin{cases} 1 & j = \tilde{i} \\ z^1_j & j \in V \setminus \{\tilde{i}\} \end{cases},$$

$$y^3_{ij} = \begin{cases} 1 & (i,j) = (\tilde{i},\tilde{j}) \\ y^1_{ij} & (i,j) \in E \setminus \{(\tilde{i},\tilde{j})\}. \end{cases}$$

It is easy to see that (x^3,y^3,z^3) satisfies (8)-(12). We observe that $f(x^1,y^1,z^1) - f(x^3,y^3,z^3) = (2x^1_{\tilde{i}} - z^1_{\tilde{i}} - y^1_{\tilde{i},\tilde{j}}) - (2\times 1 - 1 - 1) = x^1_{\tilde{i}} + (x^1_{\tilde{i}} - z^1_{\tilde{i}}) \geq 0$, where the last inequality holds due to the fact that (x^1,y^1,z^1) satisfies (9). By repeating the above construction at most $|E|$ times we arrive at a solution (x^2,y^2,z^2) satisfying the claim. $\diamondsuit$

We now restrict our attention to feasible solutions of (7)-(12) with the vector y fixed to e. Next, we show that a feasible solution with integral x components exists.

Claim b. Given (x^1,e,z^1) satisfying (8)-(12), there exists a solution (x^2,e,z^2) satisfying (8)-(12) such that $x^2 \in \{0,1\}^{|V|}$ and $f(x^2,e,z^2) \leq f(x^1,e,z^1)$.

Proof of Claim b. If $x^1 \in \{0,1\}^{|V|}$, then there is nothing to verify. If there exists j such that $x^1_j > 1$, then we can set $x^1_j = 1$. The resulting solution still satisfies (8)-(12), and the value of the function f reduces. Therefore, the non-trivial case is when there exists $\tilde{j}$ such that $x^1_{\tilde{j}} \in (0,1)$. In this case, we construct a solution (x^3,e,z^3) as follows. Examine the set of neighboring vertices $N(\tilde{j})$ of the vertex $\tilde{j}$. If $x^1_{\tilde{i}} + x^1_{\tilde{j}} > y^1_{\tilde{i}\tilde{j}} = 1$ for all $\tilde{i} \in N(\tilde{j})$ then we may reduce the value of $x^1_{\tilde{j}}$ by a sufficiently small positive value so that (x^1,e,z^1) still satisfies (8)-(12) and the value of $f(x^1,e,z^1)$ reduces. Therefore, we may assume that there exists a vertex $\tilde{i} \in N(\tilde{j})$ such that $x^1_{\tilde{i}} + x^1_{\tilde{j}} = 1$. Without loss of generality, we may assume that $1 > x^1_{\tilde{j}} \geq \frac{1}{2}$ (otherwise we can swap $\tilde{i}$ and $\tilde{j}$), which also implies that $z^1_{\tilde{j}} = 0$. Next, we construct (x^3, e, z^3) as follows:

$$x^3_j = \begin{cases} 1 & j = \tilde{j} \\ x^1_j & j \in V \setminus \{\tilde{j}\} \end{cases},$$

$$z^3_j = \begin{cases} 1 & j = \tilde{j} \\ z^1_j & j \in V \setminus \{\tilde{j}\} \end{cases}.$$

It is easy to see that (x^3, e, z^3) with $x^3_{\tilde{j}} \in \{0, 1\}$ as constructed above satisfies (8)-(12). Furthermore $f(x^1, e, z^1) - f(x^3, e, z^3) = 2x^1_{\tilde{j}} - (2 - 1) = 2x^1_{\tilde{j}} - 1 \geq 0$. By repeating the above construction at most $|V|$ times, we obtain the required (x^2, e, z^2) satisfying the claim. ♢

From the claims a and b, it is clear that there exists a feasible solution of the form (x, y, z) with (i) $y = e$ and (ii) $x \in \{0, 1\}^{|V|}$. If $x_j = 1$ and $z_j = 0$ for some j, then we can set $z_j = 1$, and the resulting solution is still feasible for (7)-(12). Therefore, we may assume that the feasible solution also satisfies $x_j = z_j$ for all $j \in V$. We select any such feasible solution and let $S = \{j : \; x_j = 1\}$. Clearly, S is a vertex cover for G since $y = e$. Notice that $f(x, y, z) = 2|S| - |S| - |E| \leq q - |E|$ or equivalently $|S| \leq q$. □

Theorem 1. *CKVLP is strongly $\mathcal{NP}$-hard.*

Proof. To verify that (5) is $\mathcal{NP}$-hard, we show that if there exists a polynomial time algorithm for solving (5), then there is a polynomial time algorithm for deciding the feasibility of (6). This completes the proof, since by Lemma 1, we have that deciding the feasibility of (6) is $\mathcal{NP}$-complete.

Let $v(p)$ denote the optimal value of (5) as a function of $p \in \{0, \dots, m\}$. Consider the following algorithm for deciding the feasibility of (6):

1. Given $A \in \mathbb{Z}^{m \times n}_{+}$, $c \in \mathbb{Z}^n$, and $\eta \in \mathbb{Z}$, compute $v(p)$ for all $p \in \{0, \dots, m\}$, using the polynomial-time algorithm for solving (5).

2. Compute $\eta^* := \min_{0 \leq p \leq m} \{v(p) - p\}$. If $\eta^* \leq \eta$, return "yes," (i.e. (6) is feasible); otherwise return "no."

Notice that the above algorithm is a polynomial time algorithm in the size of the encoding of (6). It remains to verify the validity of the above algorithm.

Suppose $\eta^* \leq \eta$ and $p^* \in \operatorname{argmin}\{v(p) - p\}$. Consider an optimal solution (x^*, z^*) to (5) corresponding to $p = p^*$. Since $\eta \geq \eta^* = v(p^*) - p^* \geq v(p^*) - e^\top z^* = c^\top x^* - e^\top z^*$, the instance of (6) is feasible.

Suppose $\eta^* > \eta$. Assume by contradiction that the instance of (6) is feasible and let (x^*, z^*) be a feasible point. Let $p^* = \sum_{j=1}^m z^*_j$. Then, observe that (x^*, z^*) is feasible to (5) corresponding to $p = p^*$. Thus, $\eta^* \leq v(p^*) - p^* \leq c^\top x^* - p^* \leq \eta$, a contradiction. □

2.2 Performance of a standard MIP solver on CKVLP instances

Given the significant advancements made in MIP solution technology, many instances of $\mathcal{NP}$-hard problems are not necessarily difficult to solve in practice. To assess the practical computational difficulty of CKVLP, we next report on the performance of CPLEX, a state-of-the-art MIP solver, on randomly generated instances of the MIP (4).

We consider instances with $n = 20$, $m = 200$ and $k \in \{15, 20\}$. The data is generated as follows:

1. "Dense Data": Each left-hand-side coefficient a_{ij} is generated uniformly between 0.8 and 1.5, and then the coefficients are divided by 1.1. The cost vector is a vector of ones.

2. "Sparse Data": This uses the same input data as in "Dense Data", except that half of the left-hand-side coefficients are randomly set to zero.

3. “Random Objective”: These instances have the same constraint coefficients as in “Dense Data”, but with random integer cost coefficients between 1 and 100.

For each of the six combinations of two values of k and three data classes, we considered 10 instances with a total for 60 instances. The computations are run on Intel Xeon 2.27 GHz dual core Linux server installed with 4 Gb RAM. The model is implemented with the callable libraries and solved by the MIP solver in CPLEX 12.1 with default settings.

The average results over ten instances in each size-data combination are presented in Table 1. The ‘Gap’ column in the table reports the root node LP relaxation gap closed by CPLEX cuts. That is, the value $\left(\frac{z^{LP+Cuts}-z^{LP}}{z^{*}-z^{LP}}\right) \times 100$, where $z^{LP+Cuts}$, z^{LP}, and z^{*} are the objective function values of the LP relaxation with CPLEX cuts at the root node, of just the LP relaxation, and of the MIP, respectively. The ‘Nodes’ and the ‘Time’ columns report the number of branch-and-bound tree nodes generated and the time in seconds needed to solve the instances to optimality, respectively.

	Dense Data			Sparse Data			Random Objective		
k	Gap	Nodes	Time	Gap	Nodes	Time	Gap	Nodes	Time
15	2%	3,537,864	2,454	7%	158,039	83	17%	1,777	1
20	2%	43,296,679	25,948	6%	1,769,574	917	21%	6,227	2

Table 1: Performance of CPLEX on CKVLP

Following are some observations based on the above computations.

1. The effect of k: Setting k to a larger value results in a substantial increase in time and memory consumption (measured in the number of nodes in the branch-and-bound tree), as seen by a ten-fold increase for the first two types of instances. This phenomenon can perhaps be explained by the combinatorial nature of CKVLP, which is to choose the linear program with the best objective value among $\binom{m}{k}$ linear programs. When k increases to $\lfloor \frac{m}{2} \rfloor$, the number of possible linear programs increases rapidly.

2. The effect of sparsity: The coefficient matrix density, measured by the number of non-zeros, can make instances significantly harder to solve, as seen by a 20-time increase in nodes and 30-time increase in time when the density increases from 50% to 100%. The dense coefficients not only make the LP relaxation hard to solve, but also make it hard for CPLEX to find effective cuts, e.g., CPLEX default cuts close only 2% of the LP relaxation gap in the “Dense Data” instances, whereas 6-7% of the gap is closed in the “Sparse Data” instances.

3. The effect of objective function: The objective function coefficients play a crucial role in determining the computational difficulty, as demonstrated by the contrast between “Dense Data” and “Random Objective”. The instances with random objective coefficients can be solved in seconds; however, the instances with the same constraints but uniform objective coefficients in “Dense Data” take hours to solve. When the cost coefficients and the constraint coefficients are set up in a way so that the objective values of linear programs formed by different choices of linear constraints are close, the branch-and-bound procedure generates a great number of nodes, of which the LPs are similar in terms of bounds, and the MIP solver spends an enormous amount of time on proving optimality.

In the rest of the paper, we focus on variants of the most difficult class of the above instances, that is, instances that are very similar in type to “Dense Data,” and attempt to tighten the root node lower bound and reduce the size of the search tree.

3 Iterative Coefficient Strengthening

In this section, we propose and analyze a scheme for strengthening the coefficients of the binary variables in the MIP formulation (4) of CKVLP. Let X denote the set of feasible x solutions of (4), i.e.

$$X := \{x \in \mathbb{R}^n_+ : \ \exists z \in \{0,1\}^m \text{ s.t. } a_i^\top x + z_i \geq 1 \ \forall \ i = 1, ..., m \text{ and } \sum_{i=1}^m z_i \leq k\}. \tag{13}$$

Definition 1. *A vector $\ell \in \mathbb{R}^m$ is called a* valid bound vector *if $\ell_i \leq \min\{a_i^\top x : \ x \in X\}$ for all $i = 1, \ldots, m$.*

Given a valid bound vector ℓ, let

$$X(\ell) := \{x \in \mathbb{R}^n_+ : \ \exists z \in [0,1]^m \text{ s.t. } a_i^\top x + (1 - \ell_i)z_i \geq 1 \ \forall \ i = 1, ..., m \text{ and } \sum_{i=1}^m z_i \leq k\}.$$

Proposition 2. *(i) If ℓ is a valid bound vector then $X(\ell) \supseteq X$. (ii) The bound vector $\ell = 0$ is valid. (iii) For valid bounds ℓ^1 and ℓ^2, if $\ell^2 \geq \ell^1$ then $X(\ell^1) \supseteq X(\ell^2)$.*

Proof. (i) If $x \in X$ then there exists $z \in \{0,1\}^m$ such that $a_i^\top x \geq \max\{1 - z_i, \ell_i\}$ for all $i = 1, \ldots, m$ and $\sum_{i=1}^m z_i \leq k$. Since $\max\{1 - z_i, \ell_i\} = 1 - (1 - \ell_i)z_i$ when $z_i \in \{0,1\}$, it follows that $a_i^\top x + (1 - \ell_i)z_i \geq 1$ and $x \in X(\ell)$. (ii) Since $a^\top x \geq 0$ for all $x \in \mathbb{R}^n_+$, we obtain that $\ell = 0$ is a valid bound vector. (iii) If $x \in X(\ell^2)$ then there exists $z \in [0,1]^m$ such that $a_i^\top x \geq 1 - (1 - \ell_i^2)z_i$ for all $i = 1, \ldots, m$ and $\sum_{i=1}^m z_i \leq k$. Since $z_i \geq 0$ this implies that $a_i^\top x \geq 1 - (1 - \ell_i^1)z_i$, hence $x \in X(\ell^1)$. □

Note that $X(0)$ is the projection, on to the x variables, of the LP relaxation of the MIP formulation (4). Proposition 2 suggests that we can strengthen this LP relaxation by iteratively tightening the bound vector ℓ and hence the coefficients of the binary variables in (4), starting from $\ell = 0$. Algorithm 1 describes such a coefficient strengthening procedure. Note that procedure requires solving m feasible linear programs with bounded objectives in each iteration.

Algorithm 1 Iterative Coefficient Strengthening

```
Input : A threshold parameter $\varepsilon > 0$ and the data $(m, n, k, a_{ij})$ describing $X$
Output : A valid bound vector $\hat{\ell} \in \mathbb{R}^m_+$

  $\Delta \leftarrow 2\varepsilon$, $t \leftarrow 1$, $\ell^t \leftarrow 0$
  while $\Delta > \varepsilon$ do
    for $i = 1, ..., m$ do
      $\ell_i^{t+1} = \min\{a_i^\top x : \ x \in X(\ell^t)\}$
    end for
    $\Delta \leftarrow ||\ell^{t+1} - \ell^t||_\infty$
    $t \leftarrow t + 1$
  end while
  $\hat{\ell} \leftarrow \ell^t$
```

Proposition 3. *Let $\{\ell^t\}$ be the sequence of bound vectors produced in Algorithm 1. We have (i) $\ell^{t+1} \geq \ell^t$ and (ii) ℓ^t is a valid bound vector for all t. Accordingly, Algorithm 1 terminates finitely returning a valid bound vector $\hat{\ell}$.*

Proof. We proceed by induction on t. For the base case $t = 1$ we have $\ell^1 = 0$, then (ii) holds from part (ii) of Proposition 2. Moreover $\ell_i^2 = \min\{a_i^\top x : \ x \in X(0)\} \geq 0$ for all i, hence (i) holds. Suppose now that (i) and (ii) hold for some $t > 1$. By definition $\ell_i^{t+1} = \min\{a_i^\top x : \ x \in X(\ell^t)\}$ for all $i = 1, \ldots, m$. Thus, for each $i = 1, \dots, m$, $\ell_i^{t+1} \leq a_i^\top x$ for all $x \in X(\ell^t)$ and hence for all $x \in X$ since $X \subseteq X(\ell^t)$ from the validity of ℓ^t. Thus ℓ^{t+1} is a valid bound vector and (ii) holds for all t. By our induction hypothesis $\ell^{t+1} \geq \ell^t$ thus $X(\ell^{t+1}) \subseteq X(\ell^t)$ by part (iii) of Proposition 2. Thus $\ell_i^{t+2} = \min\{a_i^\top x : \ x \in X(\ell^{t+1})\} \geq \min\{a_i^\top x : \ x \in X(\ell^t)\} = \ell_i^{t+1}$ for all $i = 1, \dots, m$, and so (i) holds for all t. Finally note that, for any t, $X(\ell^t) \supseteq X$ from part (i) of Proposition 2, thus $\ell_i^t = \min\{a_i^\top x : \ x \in X(\ell^t)\} \leq \min\{a_i^\top x : \ x \in X\} =: \bar{\ell}_i^*$, where $\bar{\ell}_i^*$ is a well defined finite value, for all $i = 1, \dots, m$. Thus, for each $i = 1, \dots, m$, $\{\ell_i^t\}$ is a bounded nondecreasing sequence, and hence convergent. It follows that for any $\varepsilon > 0$ there exists a sufficiently large value of t such that $||\ell^{t+1} - \ell^t||_\infty \leq \varepsilon$ ensuring finite termination of the algorithm. □

Next we analyze the strength of the LP relaxation of (4) using tightened coefficients derived using Algorithm 1. Given a cost vector c, let

$$v^* = \min\{c^\top x : \ x \in X\} \ \text{ and } \ z^L(\ell) = \min\{c^\top x : \ x \in X(\ell)\}, \tag{14}$$

be the optimal value of the MIP (4) and the optimal value of the LP relaxation corresponding to bound vector ℓ, respectively. Note that these values are finite as long as $c \geq 0$. Recall that $v^L(0)$ is the natural LP relaxation bound for (4), and the coefficient tightening scheme in Algorithm 1 is aimed to improve this bound. In the following we analyze this improvement as a function of the instance data. For simplicity of the analysis we assume that $c_j > 0$ and $a_{ij} > 0$ for all i and j. Let

$$\rho = \min_{i=1,\dots,m} \min_{j=1,\dots,n} \left\{ \frac{a_{ij}}{(1/m)\sum_{i'=1}^m a_{i'j}} \right\}. \tag{15}$$

Note that ρ is a measure of the variability of the constraint coefficient data and $\rho \in (0, 1]$. Let $\{\ell^t\}$ be the sequence of bound vectors produced by the scheme in Algorithm 1 with a threshold of $\varepsilon = 0$. From Proposition 3 we know that this sequence is convergent. Let

$$\ell^* = \lim_{t \to \infty} \ell^t. \tag{16}$$

Recall that m is the number of constraints in (4) and k is maximum number of constraints allowed to be violated.

Lemma 2. *Assuming $a_{ij} > 0$ for all $i = 1, \dots, m$ and $j = 1, \dots, n$,*

$$\ell_i^* \geq \frac{m - k}{m - \rho k} \rho \ \ \forall \ i = 1, \dots, m,$$

where ρ and ℓ^* are as defined in (15) and (16), respectively.

Proof. Let $\{u^t\}$ be a sequence of m dimensional vectors defined by the following recursion:

$$u_i^1 = 0 \ \text{ and } \ u_i^{t+1} = \rho(1 - (1 - u_i^t)k/m) \ \forall \ i = 1, \dots, m, \ \forall \ t \geq 1. \tag{17}$$

First, we claim that

$$\ell^t \geq u^t \geq 0 \ \forall \ t \geq 1. \tag{18}$$

We prove this claim by induction on t. Note that (18) holds for $t = 1$ since $\ell_i^1 = u_i^1 = 0$ for all $i = 1, \dots, m$. Suppose now that (18) holds for some $t > 1$. Since $u_i^t \geq 0$ and $0 < k/m \leq 1$ we have that $(1 - (1 - u_i^t)k/m) = (1 - k/m) + u_i^t k/m \geq 0$, and hence $u_i^{t+1} \geq 0$. Let $\mu_j = \sum_{i=1}^m a_{ij}/m$ for $j = 1, \dots, n$ and μ be the corresponding

n-dimensional vector. For any $i = 1, \ldots, m$,

$$
\begin{aligned}
\ell_i^{t+1} &= \min\{a_i^\top x : \ x \in X(\ell^t)\} && (19)\\
&\geq \min\{a_i^\top x : \ x \in X(u^t)\} && (20)\\
&= \min\{a_i^\top x : \ a_{i'}^\top x + (1 - u_{i'}^t) z_{i'} \geq 1 \ \forall \ i' = 1, \ldots, m, \sum_{i'=1}^{m} z_{i'} \leq k, x \in \mathbb{R}_+^n, z \in [0,1]^m\} && (21)\\
&\geq \min\{a_i^\top x : \ \mu^\top x \geq 1 - (1 - u_i^t)k/m, x \in \mathbb{R}_+^n\} && (22)\\
&= (1 - (1 - u_i^t)k/m)(\min_{j=1,\ldots,n}\{a_{ij}/\mu_j\}) && (23)\\
&\geq \rho(1 - (1 - u_i^t)k/m) && (24)\\
&= u_i^{t+1}, && (25)
\end{aligned}
$$

where (20) follows from the induction hypothesis $\ell^t \geq u^t$ since $X(\ell^t) \subseteq X(u^t)$ by Proposition 2(iii); (21) follows from the definition of $X(u^t)$; (22) follows by aggregating the m rows of the linear program defined in (21), noting that $u_{i'}^t = u_i^t$ for all i and i', and eliminating the z variables; since $(1 - (1 - u_i^t)k/m) \geq 0$, (23) follows from the optimal solution of the single constrained linear program defined in (22); (24) follows from the definition of ρ; and (25) follows from the definition of u_i^{t+1}. Thus (18) holds.

Next we claim that, for all $i = 1, \ldots, m$, $\{u_i^t\}$ is convergent and

$$\lim_{t\to\infty} u_i^t = \frac{m-k}{m-\rho k}\rho. \tag{26}$$

Consider any $i \in \{1, \ldots, m\}$. We first verify that $u_i^t \leq \frac{m-k}{m-\rho k}\rho$ for all t. We proceed by induction on t. By definition $u_t^1 = 0 \leq \frac{m-k}{m-\rho k}\rho$. By induction hypothesis, we have that $u_i^t \leq \frac{m-k}{m-\rho k}\rho$. Now $u_i^{t+1} = \rho - \rho\frac{k}{m} + \rho\frac{k}{m}u_i^t \leq \rho - \rho\frac{k}{m} + \rho\frac{k}{m}\left(\frac{m-k}{m-\rho k}\rho\right) = \frac{m-k}{m-\rho k}\rho$. Now we verify that the sequence $\{u_i^t\}$ is non-decreasing. Observe that $u_i^t - u^{t+1} = u_i^t - \left(\rho - \rho\frac{k}{m} + \rho\frac{k}{m}u_i^t\right) = u_i^t\left(1 - \rho\frac{k}{m}\right) - \rho + \rho\frac{k}{m} \leq \left(\frac{m-k}{m-\rho k}\rho\right)\left(1 - \rho\frac{k}{m}\right) - \rho + \rho\frac{k}{m} = 0$. Finally suppose by contradiction that the sequence $\{u_i^t\}$ converges to a value $\frac{m-k}{m-\rho k}\rho - \delta$, where $\delta > 0$. Therefore, there exists a t such that $\frac{m-k}{m-\rho k}\rho - \delta > u_i^t > \frac{m-k}{m-\rho k}\rho - \delta - \epsilon$, in which $\epsilon = \delta\left(1 - \rho\frac{k}{m}\right)$. Since $\rho\frac{k}{m} < 1$, we have $\left(1 - \rho\frac{k}{m}\right) < 1$. Hence, we obtain $u_i^t - u_i^{t+1} < \left(\frac{m-k}{m-\rho k}\rho - \delta\right)\left(1 - \rho\frac{k}{m}\right) - \rho + \rho\frac{k}{m} = -(\delta)\left(1 - \rho\frac{k}{m}\right) = -\epsilon$. Thus, $u_i^{t+1} > u_i^t + \epsilon > \frac{m-k}{m-\rho k}\rho - \delta$ which is a contradiction. Thus (26) holds.

It then follows from (18) and (26) that

$$\ell_i^* \geq \frac{m-k}{m-\rho k}\rho \ \ \forall \ i = 1, \ldots, m.$$

□

Theorem 4. *Assuming $c_j > 0$ and $a_{ij} > 0$ for all $i = 1, \ldots, m$ and $j = 1, \ldots, m$,*

$$\frac{v^* - v^L(\ell^*)}{v^*} \leq \frac{m(1-\rho)}{m-\rho k}. \tag{27}$$

Proof. Note that

$$
\begin{aligned}
v^L(\ell^*) &= \min\left\{c^\top x: \ a_i^\top x + (1-\ell_i^*)z_i \geq 1 \ \forall\ i=1,\dots,m, \ \sum_{i=1}^m z_i \leq k, \ x \in \mathbb{R}^n_+, \ z \in [0,1]^m\right\} && (28)\\
&\geq \min\left\{c^\top x: \ a_i^\top x + (1-\frac{m-k}{m-\rho k}\rho)z_i \geq 1 \ \forall\ i=1,\dots,m, \ \sum_{i=1}^m z_i \leq k, \ x \in \mathbb{R}^n_+, \ z \in [0,1]^m\right\} && (29)\\
&\geq \min\left\{c^\top x: \ \mu^\top x + (1-\frac{m-k}{m-\rho k}\rho)\frac{k}{m} \geq 1 \ \forall\ i=1,\dots,m, \ x \in \mathbb{R}^n_+\right\} && (30)\\
&= \frac{c_{\hat{j}}}{\mu_{\hat{j}}}\frac{m-k}{m-\rho k} && (31)
\end{aligned}
$$

where

$$\hat{j} \in \operatorname{argmin}\left\{\frac{c_j}{\mu_j}: \ j=1,\dots,n\right\}. \tag{32}$$

In the above, (29) follows from Lemma 2; (30) follows from aggregating the rows of the LP defined in (29) and eliminating the z variables; and (31) follows from solving the single constrained LP defined in (30).

Note that

$$v^* = \min\left\{c^\top x: \ a_i^\top x + z_i \geq 1 \ \forall i \in \{1,\dots,m\}, \ \sum_{i=1}^m z_i \leq k, \ x \in \mathbb{R}^n_+, \ z \in \{0,1\}^m\right\}.$$

Next we obtain an upper bound on v^*. For $\hat{j}$ defined in (32):

1. Sort $a_{i\hat{j}}$'s from smallest to largest.
2. Let $a_{\hat{i}\hat{j}}$ be the $(k+1)^{\text{st}}$ smallest number.
3. Let $v^H = \frac{c_{\hat{j}}}{a_{\hat{i}\hat{j}}}$. This corresponds to the objective function value of the feasible solution $x_j = 0$ for $j \neq \hat{j}$ and $x_{\hat{j}} = \frac{1}{a_{\hat{i}\hat{j}}}$. Thus $v^* \leq v^H$.

Now observe that

$$\frac{c_{\hat{j}}}{\mu_{\hat{j}}}\frac{m-k}{m-\rho k} \leq v^L \leq v^* \leq \frac{c_{\hat{j}}}{a_{\hat{i}\hat{j}}} = z^H. \tag{33}$$

Therefore using the definition of ρ we obtain that,

$$\frac{v^* - v^L}{v^*} \leq \frac{z^H - v^L}{z^H} \leq \frac{m(1-\rho)}{m-\rho k}. \tag{34}$$

□

4 Mixing Set Inequalities

In this section, we study valid inequalities derived from a mixing set relaxation of CKVLP. A *mixing set* is defined as follows:

$$P = \{(y,z) \in \mathbb{R}_+ \times \{0,1\}^n \ : \ y + h_i z_i \geq h_i, i = 1,...,n\}, \tag{35}$$

where $h_1 \geq h_2 \geq \cdots \geq h_n$. The mixing set was introduced by Günlük and Pochet [10], and its variants in different contexts have also been studied in [6, 15, 7, 21, 9, 11]. The following inequalities, known as *mixing (set) inequalities*, are valid for P:

$$y + \sum_{j=1}^{l} (h_{t_j} - h_{t_{j+1}}) z_{t_j} \geq h_{t_1} \quad \forall\, T = \{t_1, ..., t_l\} \subseteq N, \tag{36}$$

where $h_{t_1} > h_{t_2} > \cdots > h_{t_l}$ and $h_{t_{l+1}} := 0$. Furthermore, these inequalities can be separated in polynomial time, are facet-defining for P when $t_1 = 1$, and are sufficient to describe the convex hull of P [2, 10].

Recently, the mixing set inequalities have been applied to solve the MIP formulation of chance-constrained problems, which has a k-violation-type substructure, i.e., a feasible solution must satisfy the constraints corresponding to at least k out of m scenarios [13, 14]. CKVLPs can be viewed as a special case of this substructure in which each scenario consists of only one covering linear constraint. We next describe and analyze the mixing set inequalities for CKVLPs.

Let the set of (x, z)-solutions to the MIP (4) be denoted by X_{MIP}, and recall from (13) that the set of x-solutions to (4) is denoted by X. Note that X is the projection of X_{MIP} into x-space, i.e., $X = \text{Proj}_x(X_{\text{MIP}})$. Following [14], we can obtain a mixing set relaxation of X_{MIP} as follows. Given a vector $\alpha \in \mathbb{R}^n_+$, calculate $\beta_i^\alpha, i \in \{1, \ldots, m\}$ as below:

$$\begin{aligned} \beta_i^\alpha := \quad & \min \quad \alpha^\top x \\ & \text{s.t.} \quad a_i^\top x \geq 1,\ x \in \mathbb{R}^n_+, \end{aligned}$$

where a_i is the coefficient vector for the i-th constraint in the MIP (4). Assume without loss of generality that $\beta_1^\alpha \geq \beta_2^\alpha \geq ... \geq \beta_m^\alpha$, and consider the following set

$$Y(\alpha) := \{(x, z) \in \mathbb{R}^n_+ \times \{0, 1\}^m \,:\, \alpha^\top x + (\beta_i^\alpha - \beta_{k+1}^\alpha) z_i \geq \beta_i^\alpha, \quad i = 1, ..., k\}. \tag{37}$$

Proposition 5. *For any $\alpha \in \mathbb{R}^n_+$, $X_{\text{MIP}} \subseteq Y(\alpha)$ and $X \subseteq \text{Proj}_x(Y(\alpha))$*

Proof. Let $(\bar{x}, \bar{z}) \in X_{\text{MIP}}$. Then the non-negativity constraints and integrality constraints in $Y(\alpha)$ are satisfied by $(\bar{x}, \bar{z})$. Without loss of generality, we may assume that the indexes $1, ..., k$ in $Y(\alpha)$ are the first k indexes in X_{MIP}. It remains to verify that $(\bar{x}, \bar{z})$ satisfies the constraints $\alpha^\top \bar{x} + (\beta_i^\alpha - \beta_{k+1}^\alpha)\bar{z}_i \geq \beta_i^\alpha$ for all $i = 1, ..., k$.

(i) For i such that $\bar{z}_i$=1: We require to verify that $\alpha^\top \bar{x} \geq \beta_{k+1}^\alpha$. Since $(\bar{x}, \bar{z}) \in X_{\text{MIP}}$, there exists some $u \in \{1, \ldots, k+1\}$ such that $a_u^\top \bar{x} \geq 1$. Moreover as $\beta_u^\alpha = \min\{\alpha^\top x : x \in \mathbb{R}^n_+,\ a_u^\top x \geq 1\}$, we obtain that $\alpha^\top \bar{x} \geq \beta_u^\alpha \geq \beta_{k+1}^\alpha$, where the last inequality is due to the fact that $u \leq k+1$.

(ii) For i such that $\bar{z}_i$=0: We require to verify that $\alpha^\top \bar{x} \geq \beta_i^\alpha$. Since $(\bar{x}, \bar{z}) \in X_{\text{MIP}}$, we obtain that $a_i^\top \bar{x} \geq 1$. Moreover as $\beta_i^\alpha = \min\{\alpha^\top x : x \in \mathbb{R}^n_+,\ a_i^\top x \geq 1\}$, we have that $\alpha^\top \bar{x} \geq \beta_i^\alpha$.

Therefore, $(\bar{x}, \bar{z}) \in Y(\alpha)$ and $X_{\text{MIP}} \subseteq Y(\alpha)$. The result $X \subseteq \text{Proj}_x(Y(\alpha))$ follows from the fact that $X = \text{Proj}_x(X_{\text{MIP}})$. □

The set $Y(\alpha)$ is a valid relaxation of X_{MIP} and it is in the form of a mixing set. This can be noted by considering $y := (\alpha^\top x - \beta_{k+1}^\alpha)$ as a nonnegative continuous variable to obtain the mixing system

$$y + (\beta_i^\alpha - \beta_{k+1}^\alpha) z_i \geq \beta_i^\alpha - \beta_{k+1}^\alpha \ \ \forall\, i = 1, \ldots, k.$$

Thus, we have the complete description of $\text{conv}(\text{Y}(\alpha))$ using the inequalities (36), which are also valid for X_{MIP}, i.e., $\text{conv}(\text{X}_{\text{MIP}}) \subseteq \text{conv}(\text{Y}(\alpha))$. Let us call $\bigcap_{\alpha \in \mathbb{R}^n_+} \text{conv}(\text{Y}(\alpha))$ the *mixing closure*. Clearly, the mixing closure is a valid relaxation of $\text{conv}(\text{X}_{\text{MIP}})$. Let v^{MIX} be the optimal objective value of optimizing over the mixing closure, and v^* be the optimal objective value of the MIP (4). Then, the best root node gap that can be potentially achieved by the mixing inequality procedure is bounded by $(v^* - v^{\text{MIX}})/v^*$. To study this gap quantitatively, e.g., deriving a bound for $(v^* - v^{\text{MIX}})/v^*$, we analyze the projection of the mixing closure on the x-space, i.e., $\text{Proj}_x(\bigcap_{\alpha \in \mathbb{R}^n_+} \text{conv}(\text{Y}(\alpha)))$ in the following subsections.

4.1 The mixing closure

Note that

$$\begin{aligned} \mathrm{conv}(X) = \mathrm{Proj}_x(\mathrm{conv}(X_{\mathrm{MIP}})) &\subseteq \mathrm{Proj}_x\big(\bigcap_{\alpha\in\mathbb{R}^n_+} \mathrm{conv}(Y(\alpha))\big) \\ &\subseteq \bigcap_{\alpha\in\mathbb{R}^n_+} \mathrm{Proj}_x(\mathrm{conv}(Y(\alpha))) = \bigcap_{\alpha\in\mathbb{R}^n_+} \mathrm{conv}(\mathrm{Proj}_x(Y(\alpha))). \end{aligned} \tag{38}$$

Thus, minimizing over $\bigcap_{\alpha\in\mathbb{R}^n_+} \mathrm{conv}(\mathrm{Proj}_x(Y(\alpha)))$ yields a lower bound for v^{MIX}.

Proposition 6. $\mathrm{Proj}_x(Y(\alpha)) = \{x \in \mathbb{R}^n_+ : \alpha^\top x \geq \beta^\alpha_{k+1}\}$.

Proof. $\subseteq$: Let $\bar{x} \in \mathrm{Proj}_x(Y(\alpha))$, then there exists $\bar{z} \in \{0,1\}^k$ such that $\alpha^\top \bar{x} + (\beta^\alpha_i - \beta^\alpha_{k+1})\bar{z}_i \geq \beta^\alpha_i, i = 1, ..., k$. Thus $\alpha^\top \bar{x} \geq \beta^\alpha_i(1-\bar{z}_i) + \beta^\alpha_{k+1}\bar{z}_i \geq \beta^\alpha_{k+1}$ since $\beta^\alpha_i \geq \beta^\alpha_{k+1}$ and $\bar{z}_i \in [0,1]$.

$\supseteq$: Let $\bar{x} \in \{x \in \mathbb{R}^n_+ : \alpha^\top x \geq \beta^\alpha_{k+1}\}$, set $\bar{z}_i = 1, i = 1, ..., k$, then $(\bar{x}, \bar{z}) \in Y(\alpha)$ and $\bar{x} \in \mathrm{Proj}_x(Y(\alpha))$. □

Since $\mathrm{Proj}_x(Y(\alpha))$ is a half space in the non-negative orthant and hence convex, the convex hull operator in $\bigcap_{\alpha\in\mathbb{R}^n_+} \mathrm{conv}(\mathrm{Proj}_x(Y(\alpha)))$ is unnecessary.

Proposition 7.

$$\bigcap_{\alpha\in\mathbb{R}^n_+} \mathrm{conv}(\mathrm{Proj}_x(Y(\alpha))) = \bigcap_{\alpha\in\mathbb{R}^n_+} \mathrm{Proj}_x(Y(\alpha)) = \bigcap_{\alpha\in\mathbb{R}^n_+} \{x \in \mathbb{R}^n_+ : \alpha^\top x \geq \beta^\alpha_{k+1}\}.$$

Proposition 7 and (38) indicate that the projection of the mixing closure onto the x-space is contained in the closure constituted by infinitely many half spaces. To study this closure, we give a formal definition as below:

Definition 2 (Basic Mixing Closure)**.** The *Basic Mixing Closure* is defined as

$$\mathcal{M} := \bigcap_{\alpha\in\mathbb{R}^n} \{x \in \mathbb{R}^n_+ \,:\, \alpha^\top x \geq \beta^\alpha\}, \tag{39}$$

where $\beta_\alpha := \beta^\alpha_{k+1}$.

We call $\alpha^\top x \geq \beta^\alpha$ a *basic mixing inequality* corresponding to α. In order to understand the basic mixing closure, we describe another class of inequalities.

Definition 3 (Simple Disjunctive Cuts and Closure)**.**

1. Select a subset S of $k+1$ constraints. Since at least one of these constraints must be satisfied, we obtain the simple disjunction:

$$(a_{i_1}^\top x \geq 1, x \in \mathbb{R}^n_+) \vee (a_{i_2}^\top x \geq 1, x \in \mathbb{R}^n_+) \vee \cdots \vee (a_{i_{k+1}}^\top x \geq 1, x \in \mathbb{R}^n_+), \tag{40}$$

   where $S = \{i_1, \ldots i_{k+1}\}$.

2. Define $a_S \in \mathbb{R}^n$ as

$$(a_S)_j = \max_{i\in S}\{a_{ij}\} \quad \forall j = 1, ..., n.$$

   The convex hull of (40) is

$$(a_S)^\top x \geq 1, x \in \mathbb{R}^n_+,$$

   and we call $(a_S)^\top x \geq 1$ a *simple disjunctive cut.*

We define the *simple disjunctive closure* as

$$\mathcal{D} := \bigcap_{S\subseteq\{1,\ldots,m\},|S|=k+1} \{x \in \mathbb{R}^n_+ \,:\, (a_S)^\top x \geq 1\}. \tag{41}$$

Proposition 8. $\mathcal{D} = \mathcal{M}$

Proof. $\mathcal{D} \subseteq \mathcal{M}$: For any given α, without loss of generality, let $\beta_1 \geq \ldots \beta_k \geq \beta_{k+1} \geq \cdots \geq \beta_m$. Then $\beta_\alpha = \beta_{k+1}$. Since $\alpha^\top x \geq \beta_i$ is a valid inequality for the set $\{a_i^\top x \geq 1, x \in \mathbb{R}^n_+\} \quad \forall i = 1, ..., k+1$, $\alpha^\top x \geq \beta_\alpha$ is a valid inequality for the convex hull of the set

$$(a_1^\top x \geq 1, x \in \mathbb{R}^n_+) \vee (a_2^\top x \geq 1, x \in \mathbb{R}^n_+) \vee \cdots \vee (a_{k+1}^\top x \geq 1, x \in \mathbb{R}^n_+),$$

or equivalently $\alpha^\top x \geq \beta_\alpha$ is dominated by the inequality $(a_S)^\top x \geq 1$.

$\mathcal{M} \subseteq \mathcal{D}$: Let $S \subseteq \{1, \ldots, m\}$ such that $|S| = k+1$. We set $\alpha = \alpha_S$. Then for any $i \in \{1, \ldots, m\}$,

$$\begin{aligned} \beta_i = \quad & \min \quad (a_S)^\top x \\ & \text{s.t.} \quad (a_i)^\top x \geq 1, x \in \mathbb{R}^n_+. \end{aligned}$$

Since $a_{ij} \leq (a_S)_j$, we obtain that $\beta_i = \min_{1\leq j\leq n} \frac{(a_S)_j}{a_{ij}} \geq 1$. Therefore, $\beta_{a_S} \geq 1$. Hence, the basic mixing inequality is

$$(a_S)^\top x \geq \beta_{(a_S)}$$

which dominates the inequality $(a_S)^\top x \geq 1$. □

Because m and k are finite numbers, the number of simple disjunctive cuts is also finite, the following result is immediate:

Corollary 9. *$\mathcal{M}$ is polyhedral.*

4.2 Bound Quality

Using the equivalence of $\mathcal{D}$ and $\mathcal{M}$, and the fact that $\mathcal{D}$ has an explicit form and simple structure, we derive a lower bound for v^{MIX} by studying $\mathcal{D}$. We then provide a bound on the best possible gap achievable by the addition of all possible mixing inequalities, i.e., $(v^* - v^{\text{MIX}})/v^*$.

Proposition 10. *Suppose $c > 0$ and $a_{ij} > 0$ for all i, j. Let $\underline{a} = \min_{ij}\{a_{ij}\}$ and $\overline{a} = \max_{ij}\{a_{ij}\}$. Let v^* be the optimal objective value over X and v^M be the optimal value over the basic mixing closure, then*

$$0 \leq \frac{v^* - v^{\text{MIX}}}{v^*} \leq \frac{v^* - v^M}{v^*} \leq \frac{\overline{a} - \underline{a}}{\overline{a}}.$$

Proof. Let $\underline{c} = \min_j\{c_j\}$. Note that $v^* \leq \min\{c^\top x : \; a_i^\top x \geq 1 \forall \; i = 1, \ldots, m, \; x \in \mathbb{R}^n_+\} \leq \min\{c^\top x : \; (e^\top x) \geq 1/\underline{a}, \; x \in \mathbb{R}^n_+\} = \underline{c}/\underline{a}$. By the equivalence of $\mathcal{D}$ and $\mathcal{M}$, we obtain that $v^M = \min\{c^\top x : (a_S)^\top x \geq 1 \; \forall S \subseteq \{1, ..., m\}, |S| = k+1, x \in \mathbb{R}^n_+\} \geq \min\{c^\top x : \; \overline{a}(e^\top x) \geq 1, x \in \mathbb{R}^n_+\} = \underline{c}/\overline{a}$. Thus, $(v^* - v^M)/v^* = 1 - v^M/v^* \leq 1 - (\underline{c}/\overline{a})/(\underline{c}/\underline{a}) = (\overline{a} - \underline{a})/\overline{a}$. □

The above result implies that the relaxations $\mathcal{D}$ and equivalently $\mathcal{M}$ can be tight when the variation of the constraint coefficients is small. However, the separation of the most violated simple disjunctive cut from $\mathcal{D}$ is $\mathcal{NP}$-complete. Consider an arbitrary $x^* \in \mathbb{R}^n_+$ that we want to separate. Let $M := \{i \in \{1, \ldots, m\} :$

$a_i^\top x^* < 1\}$. Clearly, $|M| > k$, because otherwise, x^* belongs to the feasible region of the k-violation problem X and therefore belongs to $\mathcal{D}$. When $|M| \geq k+1$, we solve the following separation problem:

$$\begin{aligned}
\eta = \quad & \min \quad \sum_{j=1}^{n} \pi_j x_j^* - 1 \\
& \text{s.t.} \quad \pi_j + a_{ij} w_i \geq a_{ij} \;\; j = 1, ..., n; \;\; \forall i \in M \\
& \qquad\;\; \sum_{i \in \mathcal{M}} w_i = |M| - (k+1) \\
& \qquad\;\; \pi_j \geq 0 \;\; \forall j \in \{1, \ldots, n\} \\
& \qquad\;\; w_i \in \{0,1\} \;\; \forall i \in M,
\end{aligned}$$

where π_j is the cut coefficient for variable x_j and w_i is a binary variable taking value 0 whenever the i-th row is considered in the disjunction (40). The inequality $\sum_{j=1}^{n} \pi_j x_j \geq 1$ separates x^* from $\mathcal{D}$ if and only if $\eta < 0$. This separation problem is NP-hard [13]. Notice that although the mixing closure is contained in $\mathcal{D}$ and separation over $\mathcal{D}$ is $\mathcal{NP}$-complete, we do not know the complexity of the separation over $\text{Proj}_x(\bigcap_\alpha \text{conv}(Y(\alpha)))$.

5 Branching Scheme

As demonstrated in Table 1, the branch and bound search tree could be enormously large even for a small-sized instance of the MIP (4). Part of the reason for the excessive number of nodes is the overlap in the search tree. Without loss of generality, we assume that z_j is the binary variable to branch on at the root node. The left branch with z_j fixed at zero consists of the following set

$$\mathcal{B}^L := \{(x,z) : \sum_{i \neq j} z_i \leq k, a_j^\top x \geq 1, (x,z) \in X_{\text{MIP}}^j\},$$

where X_{MIP}^j represents the set X_{MIP} with the constraint $a_j^\top x + z_j \geq 1$ dropped and the variable z_j removed from the formulation. The right branch with z_j fixed at one consists of the following set

$$\mathcal{B}^R := \{(x,z) : \sum_{i \neq j} z_i \leq k-1, a_j^\top x \geq 0, (x,z) \in X_{\text{MIP}}^j\},$$

which is the union of the following two sets:

$$\mathcal{B}_{\geq}^R := \{(x,z) : \sum_{i \neq j} z_i \leq k-1, a_j^\top x \geq 1, (x,z) \in X_{\text{MIP}}^j\}$$

and

$$\mathcal{B}_{\leq}^R := \{(x,z) : \sum_{i \neq j} z_i \leq k-1, a_j^\top x \leq 1, (x,z) \in X_{\text{MIP}}^j\}.$$

Note that $\mathcal{B}_{\geq}^R$ is in fact a restriction of $\mathcal{B}^L$ and hence a overlap between the left and right branches. Re-exploring $\mathcal{B}_{\geq}^R$ in the right branch is a redundancy which could also hinder the infeasibility-based pruning: When $\mathcal{B}_{\leq}^R$ is infeasible but $\mathcal{B}_{\geq}^R$ is feasible, the overall right branch will be treated as a feasible node that, otherwise, would have been pruned. We can safely take $\mathcal{B}_{\geq}^R$ out of the right branch and the remaining search tree will still cover the whole solution space. This logic applies to any node with a z_i fixed at one.

One way to remove the overlap from the search tree is to introduce extra constraints and use a big-M

formulation to model the dichotomy of z_is:

$$\begin{aligned} \min \quad & c^\top x \\ & a_i^\top x + z_i \geq 1 \quad i = 1, ..., m \\ & a_i^\top x + M z_i \leq 1 + M \quad i = 1, ..., m \\ & \sum_{i=1}^{m} z_i \leq k \\ & x \in \mathbb{R}_+^n, z_i \in \{0, 1\} \quad \forall i = 1, ..., m. \end{aligned}$$

With this big-M formulation, however, the number of constraints doubles, and an appropriate large number M is not obvious. Instead, we remove the overlap during the branch-and-bound process as follows: whenever a branching variable z_i is fixed at one, we reverse the sign of $a_i^\top x \geq 1$ and add it as a local cut to this node. The addition of these local cuts keeps the search tree compact and could improve infeasibility-based node pruning. Another example exploring similar infeasiblity-based pruning can be found in [4].

6 Computational Experiments

In this section, we examine the potential impact of the proposed MIP approaches in solving two classes of problems with the CKVLP structure, i.e. MIPs of the form of (4). We implement the algorithms using CPLEX callable libraries (version 12.1), run the programs on Intel Xeon 2.27 GHz dual core Linux servers installed with 4 Gb RAM, and compare the performance against the CPLEX MIP solver with default settings.

6.1 Implementation Details

The implementation of the coefficient strengthening technique (described in Section 3) straightforwardly follows Algorithm 1. Notice that, we could obtain a tighter ℓ^t by enforcing integrality constraints on some binary variables in $X(\ell^t)$, but the series of minimization problems in Algorithm 1 would become more time-consuming. We keep $X(\ell^t)$ in Algorithm 1 as the set in Definition 1. The threshold parameter Δ is chosen to be 0.001.

In the implementation of the mixing set inequality procedure (described in Section 4), we add cuts only at the root nodes of search trees. We first solve the root node LP relaxation and obtain an optimal solution $(\bar{x}, \bar{z})$. Next we select the vector α from the following two sets:

- those constraint vectors a_i's for which $a_i^\top \bar{x} < 1$; and
- the cost vector c, if all a_is have been used as α.

Then we build a mixing set $Y(\alpha)$ as described in Section 4. Other than the most violated mixing inequality from (36), we also add violated inequalities (36) with $|T| = 2$ and $t_1 = 1$ to the root-node LP relaxation and solve it. The choice of these inequalities is based on recommendations in [14]. We iterate this process until one of the following stopping criteria is reached: (1) no cut with a violation of more than 0.00001 is identified, (2) the solution time exceeds 10,000 seconds, or (3) the cut generation procedure has run for 1000 iterations. To obtain the most violated mixing inequality, we implemented the separation algorithm in [2]. At the end of the cut generation phase, we keep only the tight cuts in the final model that is passed on to the branch-and-bound phase.

In the implementation of the branching rule, we add $a_i^\top x \leq 1$ as a local cut to the nodes in which z_i is fixed at one.

6.2 Probabilistic Portfolio Optimization

The first class of instances we test are from the probabilistically-constrained portfolio optimization model (2) introduced in Section 1. This problem can be approximated by the sample approximation approach as in (3) and reformulated as the following MIP [16]:

$$\begin{array}{rl} \min & c^\top x \\ \text{s.t.} & e^\top x = 1 \\ & a_i^\top x + r z_i \geq r, \quad \forall i = 1, ..., m \\ & \displaystyle\sum_{i=1}^{m} z_i \leq k, \\ & x \in \mathbb{R}^n_+, z_i \in \{0,1\} \quad \forall i = 1, ..., n. \end{array}$$

where a_i is the i-th sample drawn from the distribution of $\tilde{a}_i$ and $k = \lfloor m \times \epsilon \rfloor$. The k-violation substructure in this formulation enforces that the number of sampled scenarios in which the overall return is not achieved must not exceed $\lfloor m \times \epsilon \rfloor$. Hence, $\frac{k}{m}$, the frequency, approximates the risk level ϵ. The constraint $e^\top x = 1$ is the budget constraint obtained by scaling the investment levels to a unit budget. We also considered instances where there is no budget constraint.

Each component of a_i is drawn from an independent uniform distribution between 0.8 and 1.5, which, in this context, represents the range between a 20% loss on one's investment and a 50% profit. The required return r is chosen to be 1.1, and ϵ is set at 0.05, indicating a ten percent average return with a probability of 95%. We set $n = 20$, $m = 200$, and $k = 15$, allowing, at most, 15 of 200 linear inequalities to be violated. The cost coefficients in the model with a budget constraint take on integer values uniformly distributed between 1 and 100. For the model without the budget constraint, we use the vector with all components equal to one as the cost vector, since the instances with this particular cost vector are especially difficult to solve. We select ten randomly generated instances for each model that can be solved by CPLEX within ten hours, and compare the proposed methods against CPLEX with default settings.

Tables 2 and 3 present the computational results for the model with a budget constraint and the model without a budget constraint, respectively. The first column gives the instance number. The second and third columns give the branch-and-bound (B&B) time (in seconds) and nodes of the CPLEX MIP solver (CPX). Columns 4-6 give the root node gap closed by the cuts generated by CPX, the coefficient strengthening (CS), and the mixing set inequalities (MIX), respectively. Finally, columns 7-9 and 10-12 compare the percentage improvements of the three schemes: the branching rule (BR), CS, and MIX, over the CPLEX MIP solver on branch-and-bound time and nodes, respectively. The percentage improvement in time for BR is computed as 100 × (Time(CPX) - Time(BR))/Time(CPX), where Time(CPX) is the branch-and-bound time for default CPLEX and Time(BR) is the branch-and-bound time using the proposed branching rule. The percentage improvements for the other two schemes, and the nodes saved are computed analogously.

The reported solution times are only for the branch-and-bound phase of the overall procedure. The mixing set cutting plane algorithm spends 20 to 30 seconds on root node until no more cuts can be separated. The time spent on coefficient strengthening, which amounts to solving a series of linear programming problems, is under 20 seconds. The local cuts added in the branching scheme can be obtained instantly by simply reversing the sign of the corresponding constraint. Since the preprocessing times in these instances are negligible in comparison with the branch-and-bound times, we do not include them in the solution time.

Table 2: Percentage Improvements Over CPLEX (Portfolio Optimization with Budget Constraint)

Instance	CPX Default		Root Gap Closed			B&B Time Saved			B&B Nodes Saved		
Number	B&B time	B&B nodes	CPX	CS	MIX	BR	CS	MIX	BR	CS	MIX
1	11,094	13,640,260	19%	24%	25%	91%	89%	60%	89%	85%	62%
2	968	1,657,606	19%	23%	24%	75%	82%	51%	70%	79%	54%
3	223	505,037	36%	36%	41%	77%	86%	25%	77%	86%	36%
4	19,830	25,651,409	7%	10%	13%	90%	91%	79%	88%	89%	80%
5	400	786,701	29%	31%	33%	70%	69%	-4%	66%	62%	-3%
6	5,044	9,786,835	14%	19%	22%	68%	82%	27%	66%	80%	37%
7	10,923	14,365,495	29%	33%	35%	82%	88%	26%	80%	85%	35%
8	1,822	3,177,889	12%	18%	21%	87%	90%	20%	86%	88%	35%
9	2,115	3,454,682	17%	24%	28%	71%	69%	-17%	66%	65%	-2%
10	13,017	10,526,548	15%	18%	21%	75%	88%	11%	79%	76%	-15%
Average	6,544	8,355,246	20%	24%	26%	79%	83%	28%	77%	80%	32%

Table 3: Percentage Improvements Over CPLEX (Portfolio Optimization Without Budget Constraint)

Instance	CPX Default		Root Gap Closed			B&B Time Saved			B&B Node Saved		
Number	B&B Time	B&B Nodes	CPX	CS	MIX	BR	CS	MIX	BR	CS	MIX
1	11,903	23,786,322	2%	64%	57%	-15%	85%	-176%	22%	91%	85%
2	14,584	24,366,521	4%	66%	61%	29%	95%	65%	38%	95%	89%
3	8,730	17,586,672	2%	64%	58%	14%	92%	-181%	19%	93%	88%
4	5,516	10,898,121	5%	64%	59%	7%	90%	51%	19%	91%	83%
5	12,462	18,021,273	4%	66%	62%	19%	95%	65%	19%	94%	89%
6	21,475	30,948,921	2%	64%	58%	58%	92%	65%	46%	93%	84%
7	6,928	14,634,688	2%	64%	60%	-27%	86%	44%	17%	88%	80%
8	15,547	20,957,656	2%	65%	61%	42%	93%	68%	33%	94%	89%
9	34,512	68,752,624	2%	64%	55%	41%	89%	63%	50%	94%	84%
10	5,314	9,376,843	2%	65%	60%	-2%	94%	69%	14%	94%	88%
Average	13,697	23,932,964	3%	65%	59%	17%	91%	13%	28%	93%	86%

From Tables 2 and 3 we observe that the mixing set inequalities and coefficient strengthening have comparable performance in terms of closing root node gaps. They both close more gap than the CPLEX default cuts, especially in the model without a budget constraint. However, in the branch-and-bound process afterwards, the mixing set inequalities cannot take full advantage of the tighter lower bounds to reduce overall time and nodes. In fact, in four of the 20 instances, the mixing set inequalities even worsen the performance. The reason lies in the difficulty of selecting effective cuts to keep in the model throughout the branch-and-bound process. In our experiment, we also try to employ the CPLEX cut pool to dynamically manage all the cuts generated at root nodes, but we have not been successful in identifying the most useful cuts.

The coefficient strengthening technique closes gap amounts similar to those closed by the mixing inequalities, but the improvement in the overall branch-and-bound process is significantly larger than for the mixing inequalities. The coefficient strengthening is able to cut down the time and nodes by an average of over 80%. This achievement can be attributed to the fact that the coefficient strengthening tightens the lower bound without introducing any extra variables or constraints at the root node.

The branching rule performs remarkably better in the model with the budget constraint, over 70% savings on nodes and time versus less than 30% savings in the model without the budget constraint. This sizable difference can be explained by the presence of the budget constraint. The budget constraint, as one type of side constraint, greatly reduces the feasible region of the node problems. Consequently, the feasibility of the node problems that have budget constraints is more sensitive to the addition of local cuts obtained by reversing the signs of the corresponding covering inequalities. Therefore, adding the local cuts to the models with budget constraints is more likely to lead to infeasible node problems, triggering the infeasibility-based node pruning more frequently.

6.3 Optimal Vaccination Allocation

The second class of test instances is the optimal vaccination allocation problem under uncertainty addressed in [18]. In this application, a scarce vaccine is allocated to households in a community to prevent an epidemic from breaking out by restricting the post-vaccination reproductive number to be strictly less than one. The sample average approximation approach to this problem yields a MIP formulation which has a CKVLP structure, plus some side constraints of form $\sum_{i \in S} x_i = 1$, where S is some subset of the index set of the decision variables. A full description of the model is provided in the Appendix. We use the same test instances of this problem as in [18]. These instances have 302 continuous variables and m binary variables (see Column 1 in Table 4). The risk level ϵ is set to 0.05, and the value of k can be determined accordingly by $k = \lfloor m \times \epsilon \rfloor$.

Table 4 compares the performance of three schemes against the performance of the CPLEX MIP solver. The first two columns describe the sizes of the instances. The next three columns provide the root node gaps closed by the cuts generated by the CPLEX MIP solver, the coefficient strengthening procedure, and the mixing inequalities, respectively. Columns 6-7 present the time (in seconds) spent on coefficient strengthening and generating mixing inequalities at the root node, respectively. Columns 8-11 and columns 12-15 compare the time (in seconds) and the number of nodes in the branch-and-bound phase by the CPLEX MIP solver and the three proposed schemes, respectively. Table 5 summarizes the percentage improvements of the three schemes over the CPLEX MIP solver with default settings. The percentage improvements in total time (root node time + branch-and-bound time) for CS is computed as 100 × (Time(CPX) - Time(CS))/Time(CPX), where Time(CPX) is the total time for default CPLEX and Time(CS) is the total time using coefficient strengthening. The percentage improvements in the branch-and-bound time (excluding the coefficient strengthening time) and the nodes saved are computed analogously. The percentage improvements for MIX and BR are computed similarly.

Table 5: Percentage Improvements Over CPLEX (Optimal Vaccination Allocation Problem)

Size		B&B Node Saved			B&B Time Saved			Total Time Saved		
m	k	CS	MIX	BR	CS	MIX	BR	CS	MIX	BR
250	12	43%	17%	40%	100%	100%	50%	-5033%	-7669%	50%
		88%	81%	80%	100%	100%	33%	-3220%	-5237%	33%
		92%	95%	92%	100%	100%	60%	-2186%	-3257%	60%
		99%	96%	95%	100%	100%	78%	-1037%	-1836%	78%
		78%	84%	78%	100%	100%	50%	-5191%	-7379%	50%
500	25	96%	88%	86%	90%	86%	69%	-1083%	-4142%	69%
		95%	87%	92%	91%	78%	78%	-701%	-2637%	78%
		96%	93%	84%	93%	82%	61%	-1808%	-6459%	61%
		91%	91%	82%	91%	82%	50%	-2251%	-7785%	50%
		100%	100%	98%	98%	96%	82%	-913%	-3833%	82%
750	37	100%	99%	97%	99%	99%	96%	-24%	-590%	96%
		78%	72%	-82%	75%	74%	-159%	-2506%	-14788%	-159%
		95%	91%	84%	89%	83%	57%	-1418%	-8252%	57%
		97%	96%	83%	97%	93%	63%	-380%	-3399%	63%
		95%	94%	89%	94%	86%	73%	-764%	-6194%	73%
1000	50	100%	99%	99%	100%	99%	97%	54%	-58%	97%
		97%	96%	96%	98%	96%	87%	12%	-180%	87%
		99%	96%	96%	99%	96%	92%	30%	-127%	92%
		91%	46%	67%	93%	61%	47%	-827%	-2800%	47%
		88%	59%	54%	88%	55%	7%	-1498%	-5689%	7%
2000	100	99%	78%	97%	99%	78%	92%	93%	72%	92%
		99%	59%	99%	99%	52%	96%	97%	49%	96%
		100%	87%	100%	99%	85%	98%	96%	81%	98%
		98%	85%	99%	99%	82%	96%	91%	74%	96%
		100%	30%	100%	100%	14%	98%	98%	12%	98%

Table 4: Computational Results (Optimal Vaccination Allocation Problem)

Size		Root Gap Closed			Root Node Time		B&B Time				B&B Nodes			
m	k	CPX	CS	MIX	CS	MIX	CPX	CS	MIX	BR	CPX	CS	MIX	BR
250	12	62%	95%	95%	103	155	2	0	0	1	81	46	67	49
		61%	93%	92%	100	160	3	0	0	2	673	81	129	134
		64%	93%	93%	114	168	5	0	0	2	1,096	83	52	91
		58%	92%	90%	102	174	9	0	0	2	3,054	39	112	149
		55%	94%	94%	106	150	2	0	0	1	153	33	24	33
500	25	52%	91%	89%	493	1,776	42	4	6	13	7,082	286	820	995
		50%	93%	90%	507	1,738	64	6	14	14	20,166	1,004	2,600	1,551
		55%	91%	90%	532	1,832	28	2	5	11	5,329	218	390	839
		54%	90%	90%	515	1,731	22	2	4	11	3,907	347	359	694
		48%	94%	92%	506	1,964	50	1	2	9	18,822	38	26	458
750	37	30%	91%	90%	1,656	9,230	1,340	7	16	59	206,179	463	1,648	5,276
		44%	91%	91%	1,574	9,066	61	15	16	158	7,124	1,554	1,989	12,944
		49%	93%	92%	1,522	8,418	101	11	17	43	16,857	804	1,436	2,729
		45%	91%	90%	1,106	8,101	232	8	16	87	40,393	1,057	1,497	6,692
		42%	91%	90%	1,116	8,164	130	8	18	35	23,886	1,198	1,332	2,658
1000	50	26%	90%	89%	2,928	10,162	6,449	30	35	182	829,738	3,142	4,773	6,351
		32%	91%	89%	3,107	10,056	3,636	86	143	482	433,591	11,998	16,333	15,915
		29%	92%	90%	3,157	10,144	4,546	28	166	376	348,976	2,552	13,239	15,253
		38%	91%	88%	3,218	10,012	350	25	137	185	28,423	2,424	15,406	9,278
		33%	91%	88%	2,791	10,109	176	21	79	163	15,241	1,821	6,310	7,065
2000	100	16%	88%	84%	10,684	10,804	166,074	843	36,096	13,992	9,978,113	57,601	2,237,052	294,053
		15%	89%	85%	10,259	11,084	386,246	2,740	185,168	15,960	24,523,780	191,138	10,109,370	316,629
		14%	89%	86%	10,570	11,276	324,023	1,779	49,923	6,305	24,463,163	116,273	3,297,139	114,177
		15%	89%	85%	11,082	11,091	141,172	2,023	24,990	5,129	9,032,616	138,575	1,350,094	89,084
		15%	89%	84%	11,257	11,448	574,819	1,889	493,400	10,795	39,500,399	128,181	27,842,365	184,543

The results in Table 4 and 5 show the effectiveness of the coefficient strengthening technique in both closing root node gaps and reducing nodes and time of the branch-and-bound phase. We observe that the performance of the coefficient strengthening algorithm is significantly more consistent than the other two methods and exhibits a certain stability. For example when $m = 1000$, the branch-and-bound time saved by the branching scheme ranges from 7.4% to 97.2%; the branch-and-bound time saved by the mixing set inequalities ranges from 55.1% to 99.5%; in contrast, the coefficient strengthening algorithm varies only from 88.1% to 99.5%. This consistent behavior is also observed for the probabilistic portfolio optimization instances in Tables 2 and 3. The branching scheme has a comparable impact on reducing the search tree size to the coefficient strengthening in the vaccination instances, especially for the difficult ones with $m = 2000$. Since this model consists of equalities as side constraints, the local cuts added by the branching rule cause infeasibility in the node problems frequently, therefore, effectively reducing the search tree size.

The performance improvement in the branch-and-bound phase comes at the expense of computational effort in coefficient strengthening and separation of mixing inequalities at the root node. Unlike the portfolio optimization instances, this effort is quite significant for the vaccination instances (see columns 6-7 in Table 4). Each iteration of the coefficient strengthening requires solving m linear programs – for the instances with $m = 1000$ and $m = 2000$, several thousand linear programs need to be solved. Similarly, in generating the mixing set inequalities, m linear programs need to be solved in order to form one mixing set for a given α, and there are m possible choices for α. Accordingly, the cut generation time increases in the order of m^2. Comparing column 8 in Table 4 and column 9 in Table 5, we observe that significant effort on coefficient strengthening is not justifiable for instances that CPLEX can solve in under 1500 seconds. For example, for the instances with $m = 1000$, the coefficient strengthening technique takes around 3000 seconds. Recall that we impose a time limit of 10000 seconds, so for these instances coefficient strengthening is run till no coefficients can be further tightened. Considering the fact that CPLEX takes only one to two hours to solve these instances, running the strengthening procedure to termination is not economical. Similarly, we observe (by comparing column 8 in Table 4 and column 10 in Table 5) that the effort on mixing inequalities is not justified for instances with $m < 2000$ that CPLEX can solve within 6500 seconds. On the overall solution time, the branching rule has a more consistent performance since it requires no additional effort at the root node. For the larger size instances with $m = 2000$, it is worth spending about three hours on strengthening to reduce the branch-and-bound time from days to minutes. The CPLEX MIP solver takes one to six days to solve these instances to optimality, whereas the coefficient strengthening reduces the overall effort to under four hours.

7 Concluding Remarks

In this paper, we study covering-type k-violation linear programs. We show that such problems are strongly NP-hard, and study empirically the computational difficulty of MIP-based approaches for these problems. We introduce and analyze a coefficient strengthening scheme, adapt and analyze an existing cutting plane technique, and present a branching technique to improve the performance of MIP approaches. Computational experiments on two classes of problems show that the proposed methods are effective in significantly reducing running times. The coefficient strengthening is most effective for large instances and reduces the solution time and the number of search tree nodes by 80% to 98% in these instances. The branching scheme reduces the size of search trees by removing overlaps between branches and incurring infeasiblity-based node pruning. It takes no effort to implement and works most effectively on the CKVLP models with side constraints. The mixing set cuts are capable of closing a large percentage of root node gaps. However, the impact of these cuts on the branch-and-bound process are mixed. Perhaps better performance might be achieved by a more effective separation procedure for mixing inequalities. We have also investigated the performance of various combinations of the three schemes, but the gains are not significant.

References

[1] E. Amaldi and V. Kann. The complexity and approximability of finding maximum feasible subsystems of linear relations. *Theoretical Computer Science*, 147(1-2):181–210, 1995.

[2] A. Atamtürk, G. L. Nemhauser, and M. W. P. Savelsbergh. The mixed vertex packing problem. *Mathematical Programming*, 89(1, Ser. A):35–53, 2000.

[3] N.G. Becker and D.N. Starczak. Optimal vaccination strategies for a community of households. *Mathematical Biosciences*, 139(2):117–132, 1997.

[4] D. Bienstock. Computational study of a family of mixed-integer quadratic programming problems. *Mathematical Programming*, 74(2):121–140, 1996.

[5] T.M. Chan. Low-dimensional linear programming with violations. *SIAM Journal on Computing*, 34(4):879–893, 2005.

[6] M. Conforti and L.A. Wolsey. Compact formulations as a union of polyhedra. *Mathematical Programming*, 114(2):277–289, 2008.

[7] M. Constantino, A.J. Miller, and M. Van Vyve. Mixing MIR inequalities with two divisible coefficients. *Mathematical Programming*, 123(2):451–483, 2010.

[8] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness.* WH Freeman & Co. New York, NY, USA, 1979.

[9] Y. Guan, S. Ahmed, and G.L. Nemhauser. Sequential pairing of mixed integer inequalities. *Discrete Optimization*, 4(1):21–39, 2007.

[10] O. Günlük and Y. Pochet. Mixing mixed integer inequalities. *Mathematical Programming*, 90(3):429–457, 2001.

[11] S. Küçükyavuz. On mixing sets arising in chance-constrained programming. *Mathematical Programming*, 132(1-2):31–56, 2012.

[12] J. Luedtke and S. Ahmed. A sample approximation approach for optimization with probabilistic constraints. *SIAM Journal on Optimization*, 19(2):674–699, 2008.

[13] J. Luedtke, S. Ahmed, and G.L. Nemhauser. An integer programming approach for linear programs with probabilistic constraints. *Mathematical Programming*, 122(2):247–272, 2010.

[14] J. Luedtke. An integer programming and decomposition approach to general chance-constrained mathematical programs. In F. Eisenbrand and F. Shepherd, editors, *Integer Programming and Combinatorial Optimization*, volume 6080 of *Lecture Notes in Computer Science*, pages 271–284. Springer Berlin / Heidelberg, 2010.

[15] A.J. Miller and L.A. Wolsey. Tight formulations for some simple mixed integer programs and convex objective integer programs. *Mathematical Programming*, 98(1):73–88, 2003.

[16] B. Pagnoncelli, S. Ahmed, and A. Shapiro. Computational study of a chance constrained portfolio selection problem. *Journal of Optimization Theory and Applications*, 142:399–416, 2009.

[17] F. Rossi, S. Smriglio, and A. Sassano. Models and algorithms for terrestrial digital broadcasting. *Annals of Operations Research*, 107(1):267–283, 2001.

[18] M.W. Tanner and L. Ntaimo. IIS branch-and-cut for joint chance-constrained stochastic programs and application to optimal vaccine allocation. *European Journal of Operational Research*, 207(1):290–296, 2010.

[19] R. Thomas and P. Widmayer. k-violation linear programming. *Information Processing Letters*, 52:109–114, 1994.

[20] A. Tunçel, F. Preciado, R. Rardin, M. Langer, and J.-P.P. Richard. Strong valid inequalities for fluence map optimization problem under dose-volume restrictions. *Annals of Operations Research*, pages 1–22, 2010.

[21] M. Zhao and I.R. de Farias Jr. The mixing-MIR set with divisible capacities. *Mathematical Programming*, 115(1):73–103, 2008.

Appendix

Optimal Vaccination Allocation Model of [18]

The vaccination allocation problem allocate a scarce vaccine to households in a community to prevent an epidemic from breaking out. The epidemic will die out if the post-vaccination reproductive number is strictly less than one. Assume a community has a set F of types of households and each type of household $f \in F$ consists of a combination of person types $t \in T$, e.g., child, adult, or elderly. A vaccination policy $v \in V$ is a delivery of vaccine to certain types of persons in a household $f \in F$. For example, a vaccination policy could be a delivery of vaccine only to the two children in a household type that consists of two adults and two children. The decision problem is to determine an implementation of vaccination policies for each type of household in this community with a minimal cost which guarantees that the post-vaccination reproductive number is strictly below one with a high probability $1 - \epsilon$. We state below the probabilistically-constrained model in [18]:

$$\begin{aligned}
\min: \quad & \sum_{f\in F}\sum_{v\in V}\sum_{t\in T} v_t h_f x_{fv} \\
\text{s.t.} \quad & \sum_{v\in V} x_{fv} = 1 \quad \forall f \in F \\
& \mathbb{P}\{\sum_{f\in F}\sum_{v\in V} a_{fv}(\omega) x_{fv} \leq 1\} \geq 1-\epsilon \\
& 0 \leq x_{fv} \leq 1 \quad \forall f \in F,\, v \in V,
\end{aligned}$$

where x_{fv} is the decision variable representing the percentage of policy v to be implemented for household type f, v_t is the number of people of type t vaccinated in policy v, h_f is the proportion of households in the community that are of type f, and $a_{fv}(\omega)$ is the computed random parameter for impact of the vaccination policy v for household type f, which is a function of different random numbers following some known distributions. For more details, see [3, 18].

After m i.i.d. samples are taken from $a_{fv}(\omega)$s, the above probabilistically-constrained problem can be approximated by the following MIP, which has a CKVLP structure:

$$\begin{aligned}
\max: \quad & \sum_{f\in F}\sum_{v\in V}\sum_{t\in T} v_t h_f x'_{fv} - \sum_{f\in F}\sum_{v\in V}\sum_{t\in T} v_t h_f \\
\text{s.t.} \quad & \sum_{v\in V} x'_{fv} = 1 \quad \forall f \in F \\
& \sum_{f\in F}\sum_{v\in V} a^i_{fv} x'_{fv} + b_i z_i \geq b_i \quad i = 1, ..., m \\
& \sum_{i=1}^{m} z_i \leq k \\
& 0 \leq x'_{fv} \leq 1 \quad \forall f \in F,\, v \in V,\ z_i \in \{0,1\}\ i = 1, .., m,
\end{aligned}$$

where a^i_{fv} is the i-th sample of $a_{fv}(\omega)$, $x'_{fv} = 1 - x_{fv}$, $b_i = \sum_{f\in F}\sum_{v\in V} a^i_{fv} - 1$, and $k = \lfloor \epsilon \times m \rfloor$.