A Bilevel Cutting-Plane Algorithm for Cardinality-Constrained Mean-CVaR Portfolio Optimization

Ken Kobayashi\textsuperscript{a,c,*}, Yuichi Takano\textsuperscript{b}, Kazuhide Nakata\textsuperscript{c}

\textsuperscript{a}Artificial Intelligence Laboratory, Fujitsu Laboratories Ltd.,
4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8588, Japan
\textsuperscript{b}Faculty of Engineering, Information and Systems, University of Tsukuba,
1-1-1 Tennodai, Tsukuba-shi, Ibaraki 305-8577, Japan
\textsuperscript{c}Department of Industrial Engineering and Economics, School of Engineering, Tokyo Institute of Technology,
2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552 Japan

Abstract

This paper studies the mean-risk portfolio optimization problems with a constraint of the number of assets to be invested. We employ conditional value-at-risk (CVaR) as a risk measure. While several studies aim at efficiently solving mean-CVaR model, it has been computationally expensive to solve cardinality-constrained mean-CVaR model, and especially for exact algorithms, the size of a problem that can be handled has been limited. In this paper, we devised an exact and efficient algorithm to handle a large-sized mean-CVaR optimization problem with a cardinality constraint. For computational efficiency, our proposed method integrates two cutting-plane algorithms, upper-level cutting-plane algorithm and lower-level one, and each algorithm solves the master problem and the subproblem, respectively. Numerical experiments demonstrated that our algorithms can attain optimal solutions to large-scale problems in a reasonable amount of time.

Keywords: mixed-integer optimization, portfolio optimization, cutting-plane algorithm, conditional value-at-risk

*Corresponding author

Email address: ken-kobayashi@fujitsu.com (Ken Kobayashi)
1. Introduction

Since the introduction of the original Markowitz [26] mean-variance portfolio optimization model, portfolio optimization is widely used in the financial industry, and it has been actively studied by both academic researchers and institutional investors. The traditional framework created by Markowitz [26] determines the asset allocation intending to make low-risk and high return investments. This paper addresses the mean-risk portfolio optimization model using the conditional value-at-risk (CVaR) [30, 31], also called average value-at-risk, as a risk measure.

CVaR is a downside risk measure for evaluating a potential heavy loss. It is known to be a coherent risk measure that has desirable properties, i.e., translation invariance, subadditivity, positive homogeneity, and monotonicity [6, 28]. Besides, it is monotonic with respect to second-order stochastic dominance [28]. This means that CVaR minimization inconsistent with the preference of any rational and risk-averse decision-maker. These facts have highlighted the importance of CVaR for making decisions in uncertain situations. In fact, the mean-CVaR model has been widely studied to solve efficiently, e.g., linear optimization approaches [22, 27], nonsmooth optimization approaches [7, 18, 24, 30], scenario representation by factor model [21], cutting-plane algorithms [2, 17, 23, 33], level method [13], smoothing methods [3, 35] and successive regression approximations [1].

This paper considers solving mean-CVaR model that limits the number of assets to be held. This requirement comes from real-world practice where a portfolio made up of a large number of assets is not desirable. The reason is that, if the number of invested assets is large, it is difficult for investors or even institutional investors to monitor each asset, and required transaction costs get high [25, 36]. Therefore, the number of assets to be invested needs to be controlled easily in practice. Such a requirement can be realized by imposing a cardinality constraint on a portfolio optimization problem.

For the mean-variance model, several studies have pondered the inclusion of the cardinality constraint e.g., heuristic algorithms [4, 11, 12], and branch-and-bound based algorithms [9, 14, 32]. For solving the cardinality-constrained mean-CVaR model, on the other hand, a cutting-plane based algorithm has been proposed by Takano et al. [33]. Angelelli et al. have formulated the model as a mixed-integer linear optimization problem and proposed a heuristic algorithm [3]. Also, Cheng and Gao have proposed a heuristic algorithm based on approximating the cardinality constraint by ℓ1-norm [11].
Whether for the mean-variance model or the mean-CVaR model, however, it has been computationally expensive to solve cardinality-constrained portfolio optimization problems, and especially for exact algorithms, the size of a problem that can be handled has been limited. Moreover, the mean-CVaR model requires us to handle a large number of scenarios when we approximate a CVaR with high precision, which also would raise a computational burden.

Recently, Bertsimas and Cory-Wright have proposed an exact method for solving the cardinality-constrained mean-variance portfolio optimization problem [8]. In this method, they formulate the problem as a bilevel optimization problem that has a nonsmooth convex objective function by using the duality theory, and employ a cutting-plane based algorithm to solve the master problem. Their numerical experiments have demonstrated that their proposed method solves problem instances much faster than the state-of-art mixed-integer optimization solver does. More surprisingly, it succeeds in obtaining optimal solutions for large-sized problem instances where the number of candidate assets to be invested is more than 300.

Based on this previous research, we aim to devise an exact and efficient algorithm named *bilevel cutting-plane algorithm* for solving the cardinality-constrained mean-CVaR model. To handle a large number of scenarios efficiently, when we formulate the mean-CVaR model as a bilevel optimization problem as Bertsimas and Cory-Wright did, we employ the Künzi-Bay and Mayer’s cutting-plane representation [23]. With this formulation, the subproblems within the master problem can be solved efficiently by using the cutting-plane algorithm [2, 17, 23]. As a result, our proposed method integrates two cutting-plane algorithms, upper-level cutting-plane algorithm and lower-level one, and each algorithm solves the master problem and the subproblem, respectively. This is the reason why we name our method *bilevel cutting-plane algorithm*.

To evaluate the performances of our bilevel cutting-plane algorithm, we conducted numerical experiments using some benchmark data sets. The results demonstrated that while it took a long time to solve the conventional mixed-integer optimization (MIO) formulations, our bilevel cutting-plane algorithm succeeded in obtaining optimal solutions more quickly especially for both the numbers of candidate assets and scenarios are large. Remarkably, our method attained an optimal solution to a problem instance where the number of candidate assets is 225 with 100,000 scenarios within an hour, which is a reasonable time. To the best of our knowledge, our studies are
the first to solve such large-sized cardinality-constrained mean-CVaR model with a guarantee of global optimality. In addition, when we investigated the sensitivity of the performance to hyperparameters, the results showed that our algorithm stably outperformed the state-of-the-art MIO solvers.

The rest of this paper is organized as follows: In Section 2 we formulate the cardinality-constrained mean-CVaR portfolio optimization problem as a mixed-integer optimization problem. Section 3 is devoted to our bilevel cutting-plane algorithm for solving the problem. The computational results are reported in Section 4. Finally, conclusion are given in Section 5.

2. Problem Formulation

2.1. Preliminaries

Let \( \mathbf{x} = (x_1, x_2, \ldots, x_n)^\top \) be the portfolio, where \( x_i \) is the investment proportion in financial asset \( i = 1, 2, \ldots, n \). This paper tackles the problem of construct a portfolio \( \mathbf{x} \) for low-risk high-return investments under the constraint of \( \mathbf{x} \).

The net return of the portfolio is expressed as

\[
\tilde{\mathbf{\mu}}^\top \mathbf{x}
\]

where \( \tilde{\mathbf{\mu}} = (\tilde{\mu}_1, \tilde{\mu}_2, \ldots, \tilde{\mu}_n)^\top \) is a random vector that represents the rate of return of each asset.

In addition, we denote by \( \mathcal{X} \) the set of feasible portfolios. Throughout of this research, we define the feasible region \( \mathcal{X} \) as follows:

\[
\mathcal{X} := \{ \mathbf{x} \in \mathbb{R}^n \mid \ell \leq A\mathbf{x} \leq \mathbf{u}, \mathbf{e}^\top \mathbf{x} = 1, \mathbf{x} \geq 0 \},
\]

where \( A \in \mathbb{R}^{n \times m}, \ell, \mathbf{u} \in \mathbb{R}^m \) are the lower and upper limits of \( A\mathbf{x} \), respectively, and \( \mathbf{e} \) is the vector whose entries are all one. The linear constrains \( \ell \leq A\mathbf{x} \leq \mathbf{u} \) includes various kinds of constraints of \( \mathbf{x} \). For example, if we impose a minimum return constraint on \( \mathbf{x} \), we describe it as follows:

\[
\mathbf{\mu}^\top \mathbf{x} \geq \bar{\mu},
\]

where \( \mathbf{\mu} \in \mathbb{R}^n \) is a vector that represents the rate of expected return of each assets, and \( \bar{\mu} \) is the minimum return to be satisfied.
2.2. Cardinality constraints

we introduce a cardinality constraint to control the number of assets to be held. Let $k$ be a positive integer that represents the upper limit of the number of non-zero entry of $x$, then, we impose the following constraint on $x$:

$$
\|x\|_0 \leq k, \tag{3}
$$

where the norm $\| \cdot \|_0$ is $\ell_0$-norm (the number of nonzero elements). Practically, this constraint is required by individual investors and even institutional investors, because they want to construct sparse portfolios to reduce transaction costs and monitoring costs.

This constraint can be realized by introducing 0-1 variables. Let $z := (z_1, z_2, \ldots, z_n)^\top$ be a vector of 0-1 decision variables for selecting assets: that is, $z_i = 1$ if $i$-th assets is selected; otherwise $z_i = 0$. Then, the cardinality constraint \[3\] can be written as follows:

$$
\left\{
  \begin{array}{l}
    z_i = 0 \Rightarrow x_i = 0 \quad (i = 1, 2, \ldots, n), \\
    z \in Z^k_n,
  \end{array}
\right. \tag{4}
$$

where $Z^k_n = \{z \in \{0, 1\}^n | \sum_{i=1}^n z_i \leq k\}$.

2.3. Conditional value-at-risk

Let $\delta \in (0, 1)$ be a parameter representing the confidence level, which is frequently set close to one. Accordingly, $\delta$-CVaR can approximately be regarded as the conditional expectation of a random loss exceeding the $\beta$-value-at-risk. The loss function $L(x, \tilde{\mu})$ is defined as the negative of the portfolio net return:

$$
L(x, \tilde{\mu}) = -\tilde{\mu}^\top x \tag{6}
$$

Given a portfolio $x \in \mathcal{X}$, the CVaR of the loss function \[6\] is calculated as the following:

$$
\mathcal{F}_\delta(a, x) := a + \frac{1}{1-\delta} \int_{\mu \in \mathbb{R}^n} [L(x, \mu) - a]_+ P(\mu) d\mu \tag{7a}
$$

where $P : \mathbb{R}^n \rightarrow \mathbb{R}$ is a probability density function of the random vector $\tilde{\mu}$, and $[\xi]_+$ is a positive part of the number of $\xi$ i.e., $[\xi]_+ = \max\{0, \xi\}$. 

5
Since multiple integration appeared in $F(a, x)$ is a computationally expensive, we often use the following scenario-based approximation:

$$F_\delta(a, x) \sim a + \frac{1}{1 - \delta} \sum_{s=1}^{S} p^{(s)} \left[ L(x, \mu^{(s)}) - a \right]_+$$

$$= a + \frac{1}{1 - \delta} \sum_{s=1}^{S} p^{(s)} \left[ -\mu^{(s)\top} x - a \right]_+,$$

where $\mu^{(s)} = (\mu_1^{(s)}, \mu_2^{(s)}, \ldots, \mu_n^{(s)})^\top$, $s = 1, 2, \ldots, S$ are scenarios of the rate of return generated from the probability density function $P$, and $p^{(s)}, s = 1, 2, \ldots, S$ is a probability of the scenario $s$ occurring.

2.4. $\ell_2$-regularized CVaR minimization problem

On the line of the previous research proposed by Bertsimas and Wright [8], we shall consider the following $\ell_2$-regularized mean-CVaR model that minimizes the risk CVaR with a cardinality constraint:

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2\gamma} x^\top x + a + y \\
\text{subject to} & \quad y \geq \frac{1}{1 - \delta} \sum_{s=1}^{S} p^{(s)} \left[ -\mu^{(s)\top} x - a \right]_+ \\
& \quad z_i = 0 \Rightarrow x_i = 0 \quad (i = 1, 2, \ldots n), \\
& \quad x \in X, \\
& \quad z \in Z^k. 
\end{align*}
\]

where $y$ is an auxiliary variable, and $\gamma > 0$ is a parameter to tune the strength of the regularization term.

To solve Problem (8) by MIO solvers, we have to transform the non-linear and non differentiable CVaR term appeared in the constraint (8b) into tractable ones. Here, we describe two formulation, the lifting representation and the cutting-plane representation.
**Lifting representation.** The most straightforward method is the lifting representation \cite{13, 30}, which converts (8b) into the following:

\[
\begin{align*}
    y & \geq \sum_{s=1}^{S} q^{(s)}, \\
    q^{(s)} & \geq \frac{1}{1-\delta} p^{(s)}(-\mu^{(s)}^\top x - a), \quad (s = 1, 2, \ldots, S) \\
    q^{(s)} & \geq 0, \quad (s = 1, 2, \ldots, S).
\end{align*}
\]

where \(q^{(s)}, s = 1, 2, \ldots, S\) are auxiliary decision variables.

**Cutting-plane representation.** Künzi-Bay and Mayer \cite{23} have shown that the constraint (8b) can be rewritten equivalently by using cutting-plane representation \cite{13, 23}:

\[
\begin{align*}
    y & \geq \frac{1}{1-\delta} \sum_{s \in \mathcal{J}} p^{(s)}(-\mu^{(s)}^\top x - a), \quad (\mathcal{J} \subseteq \{1, 2, \ldots, S\}) \\
    y & \geq 0,
\end{align*}
\]

where \(y\) is an auxiliary variable. Note that, for the cutting-plane representation, while the number of Constraint (10a) is \(2^S\), the number of auxiliary decision variables to represent CVaR does not depend on the number of scenario \(S\). Therefore, to construct efficient algorithm for solving Problem (8) with a large number of scenarios, we employ the cutting-plane representation instead of the lifting-representation.

### 3. Bilevel Cutting-Plane Algorithm

We describe our proposed algorithm named bilevel cutting-plane algorithm for solving Problem (8).

In this section, we consider solving the following problem with the cutting-plane representation:

\[
\begin{align*}
\text{minimize} \quad & \frac{1}{2\gamma} x^\top x + a + y \\
\text{subject to} \quad & \text{Constraints (8c), (8d), (8e), (10).}
\end{align*}
\]

7
3.1. Bilevel optimization reformulation

First, we reformulate Problem (11) as a bilevel optimization problem. We describe a diagonal matrix whose diagonal entries are \( x \in \mathbb{R}^n \) as \( \text{Diag}(x) \).

For a given \( z \in \mathbb{Z}_n^k \), we consider replacing \( x \) appears in \( X \) with \( \text{Diag}(z)x \), and define a set as follows:

\[
X_z := \{ x \in \mathbb{R}^n | \ell \leq A^\top \text{Diag}(z)x \leq u, e^\top \text{Diag}(z)x = 1, \text{Diag}(z)x \geq 0 \}.
\]

Then, we can see that Problem (11) can be written as the following bilevel optimization problem:

\[
\text{minimize}_{z \in \mathbb{Z}_n^k} f(z),
\]

where

\[
f(z) = \text{minimize}_{x, v, a, y} \frac{1}{2\gamma} x^\top x + a + y
\]

subject to Constraints(10) \( (13b) \)

\( x \in X_z. \) \( (13c) \)

Next, we consider the expression of \( f(z) \) to construct an algorithm for solving (12). By using the duality theory, we can obtain that expression as the following theorem:

**Theorem 1.** For each fixed \( z \in \mathbb{Z}_n^k \) such that Problem (13) is feasible, the strong duality holds, and the dual problem of Problem (13) is formulated as follows:

\[
f(z) = \text{maximize} \ -\frac{\gamma}{2} \sum_{i=1}^{n} z_i w_i^2 + \beta_{\ell}^\top \ell - \beta_u^\top u + \lambda
\]

subject to

\[
w \geq \frac{1}{1 - \delta} \sum_{J \subseteq \{1, 2, \ldots, S\}} \alpha^J \left( \sum_{s \in J} p(s) \mu(s) \right) + A^\top (\beta_{\ell} - \beta_u) + \lambda e
\]

\[
\sum_{J \subseteq \{1, 2, \ldots, S\}} \alpha^J \leq 1,
\]

\[
\sum_{J \subseteq \{1, 2, \ldots, S\}} \alpha^J \left( \sum_{s \in J} p(s) \right) = 1 = \delta,
\]

\[
\alpha^J \geq 0, \quad (J \subseteq \{1, 2, \ldots, S\}),
\]

\[
\beta_{\ell} \geq 0, \beta_u \geq 0.
\]

8
Proof. For a fixed $z \in \mathbb{Z}_k^k$ such that Problem (13) is feasible, we derive a Lagrange dual problem of Problem (13). Recall Problem (13) is formulated as:

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2\gamma} x^\top x + a + y \\
\text{subject to} & \quad y \geq \frac{1}{1-\delta} \sum_{s \in J} \rho_s (-\mu_s^\top \text{Diag}(z)x - a), \quad (J \subseteq \{1, 2, \ldots, S\}) \\
& \quad \ell \leq A \text{Diag}(z)x \leq u, \\
& \quad e^\top \text{Diag}(z)x = 1, \\
& \quad \text{Diag}(z)x \geq 0, \\
& \quad y \geq 0.
\end{align*}
\]

Let $\alpha = (\alpha_J)_{J \subseteq \{1, 2, \ldots, n\}} \geq 0$, $\beta_\ell, \beta_u \geq 0$, $\lambda \in \mathbb{R}$, $\pi \geq 0$, and $\zeta \geq 0$ be Lagrange multipliers of (15b), (15c), (15d), (15e), and (15f), respectively. Then, the Lagrange function of Problem (15) is described as follows:

\[
L := \frac{1}{2\gamma} x^\top x + a + y - \sum_{J \subseteq \{1, 2, \ldots, S\}} \alpha_J \left( y - \frac{1}{1-\delta} \sum_{s \in J} \rho_s (-\mu_s^\top \text{Diag}(z)x - a) \right) - \beta_\ell^\top (A \text{Diag}(z)x - \ell) - \beta_u^\top (u - A \text{Diag}(z)x) - \lambda (e^\top \text{Diag}(z)x - 1) - \pi^\top \text{Diag}(z)x - \zeta y.
\]

With this Lagrange function $L$, the Lagrange relaxation problem of Problem (15) is formulated as follows:

\[
\begin{align*}
\text{minimize} & \quad \max\{L \mid \alpha \geq 0, \beta_\ell \geq 0, \beta_u \geq 0, \lambda \in \mathbb{R}, \pi \geq 0, \zeta \geq 0\} 
\end{align*}
\]

For Problem (13), the objective function is proper convex, and all constraints are linear. Moreover, now we assume Problem (13) is feasible, then, the strong duality holds from [10]. As a result, Problem (17) and the following problem are the same:

\[
\begin{align*}
\text{maximize} & \quad \min\{L \mid x \in \mathbb{R}^n, y \in \mathbb{R}, a \in \mathbb{R}\}. 
\end{align*}
\]
Next, given a fixed tuple of feasible Lagrange multipliers \((\alpha, \beta_\ell, \beta_u, \lambda, \pi)\), let us consider the following problem that minimizes the Lagrange function \(L\) with respect to \(x, a\) and \(y\):

\[
\min_{x \in \mathbb{R}^n, a \in \mathbb{R}, y \in \mathbb{R}} L.
\]  

(19)

Note that Problem (19) is an unconstrained convex quadratic optimization problem and its objective function is linear in \(a\) and \(y\). Since Problem (19) must be bounded, the multipliers \(\alpha_J\) and \(\zeta\) are required to satisfy the following conditions:

\[
\nabla_a L = 1 - \frac{1}{1 - \delta} \sum_{J \subseteq \{1, 2, ..., S\}} \alpha_J \left( \sum_{s \in J} p^{(s)} \right) = 0, \quad (20)
\]

\[
\nabla_y L = 1 - \sum_{J \subseteq \{1, 2, ..., S\}} \alpha_J - \zeta = 0. \quad (21)
\]

Also, for the optimal solution of \(x\) to Problem (19), the following condition holds:

\[
\nabla_x L = \frac{1}{\gamma} x - \text{Diag}(z) \left( \frac{1}{1 - \delta} \sum_{J \subseteq \{1, 2, ..., S\}} \alpha_J \left( \sum_{s \in J} p^{(s)} \mu^{(s)} \right) + A^T (\beta_\ell - \beta_u) + \lambda e + \pi \right) = 0. \quad (22)
\]

According to the conditions (20), (21), and (22), the optimal value of Problem (19) can be calculated as follows:

\[
- \frac{\gamma}{2} w^T \text{Diag}(z)^2 w + \beta_\ell^T \ell - \beta_u^T u + \lambda, \quad (23)
\]

where \(w\) is a new auxiliary variables that satisfies

\[
w = \frac{1}{1 - \delta} \sum_{J \subseteq \{1, 2, ..., S\}} \alpha_J \left( \sum_{s \in J} p^{(s)} \mu^{(s)} \right) + A^T (\beta_\ell - \beta_u) + \lambda e + \pi. \quad (24)
\]

Since \(z \in \mathcal{Z}_n^k\), it holds that \(w^T \text{Diag}(z)^2 w = \sum_{i=1}^n z_i w_i^2\). Therefore, the Lagrange dual problem of Problem (15) that maximizes \(\min \{L \mid x \in \mathbb{R}^n, y \in \mathbb{R}\}\)
\( \mathbb{R}, a \in \mathbb{R} \) is formulated as follows:

\[
\begin{align*}
\text{maximize} \quad & -\frac{\gamma}{2} \sum_{i=1}^{n} z_i w_i^2 + \beta_{\ell}^T \ell - \beta_{u}^T u + \lambda \\
\text{subject to} \quad & w = \frac{1}{1-\delta} \sum_{J \subseteq \{1,2,\ldots,S\}} \alpha^J \left( \sum_{s \in J} p^{(s)} \mu^{(s)} \right) + A^T (\beta_{\ell} - \beta_{u}) + \lambda e + \pi \\
& \sum_{J \subseteq \{1,2,\ldots,S\}} \alpha^J + \zeta = 1, \\
& \sum_{J \subseteq \{1,2,\ldots,S\}} \alpha^J \left( \sum_{s \in J} p^{(s)} \right) = 1 - \delta, \\
& \alpha^J \geq 0, \quad (J \subseteq \{1,2,\ldots,S\}), \\
& \beta_{\ell} \geq 0, \beta_{u} \geq 0, \quad \zeta \geq 0, \quad \pi \geq 0.
\end{align*}
\]

Finally, we obtain the desired results by removing non-negative dual variables \( \zeta \) and \( \pi \) from Problem (25).

\begin{remark}
Theorem 1 assumes that \( z \in \mathcal{Z}_n^k \) yields feasible Problem (13) and ignore the case that Problem (13) is infeasible. However, when we minimize \( f(z) \) over \( \mathcal{Z}_n^k \), we can assume \( z \in \mathcal{Z}_n^k \) is feasible, without loss of generality. For an infeasible \( z \in \mathcal{Z}_n^k \), the dual of Problem (13) is unbounded by weak duality. Therefore, for any infeasible \( z \in \mathcal{Z}_n^k \), it holds that \( f(z) = +\infty \), and such solutions are omitted when we minimize \( f(z) \).
\end{remark}

Now we obtain the expression of the function \( f(z) \) to be minimized, and according to Section 3.2.3 in [10], we can show two key properties of the function \( f(z) \), convexity and subgradient as the following two lemmas:

\begin{lemma}[Convexity]
The function \( f(z) \) is convex on \( z \in [0,1]^n \).
\end{lemma}

\begin{lemma}[Subgradient]
For a fixed feasible \( z \in \mathcal{Z}_n^k \) to Problem (12), let \( w^*(z) \) be an optimal solution of \( w \) to Problem (14). Then a subgradient \( g(z) \in \partial f(z) \) is given as follows:
\[
g(z) = -\frac{\gamma}{2} w_1^*(z) \in \partial f(z).
\]
\end{lemma}
Lemma 2 and Lemma 3 show that once we evaluate the $f(\hat{z})$, for every $z \in \mathbb{Z}_n^k$, the following holds and we can obtain a lower approximation of the function $f$

$$f(z) \geq f(\hat{z}) + g(\hat{z})^T(z - \hat{z}).$$ (27)

3.2. Efficient calculation of function value and its subgradient

In the previous subsection, we describe that we can calculate the function value $f(z)$ and its subgradient $g(z) \in \partial f(z)$ by obtaining an optimal solution $w^*(z)$ to Problem (14). As we can see from Problem (14), however, this problem includes $O(2^S)$ variables. Thus, solving Problem (14) directly is still inefficient when the number of scenarios $S$ is large.

How can we obtain the function value $f(z)$ and its subgradient $g(z) \in \partial f(z)$ efficiently? Our proposal is to calculate the function value $f(z)$ and its subgradient $g(z)$ separately by solving both a primal subproblem and its dual. According to the definition of $f(z)$, $f(z)$ can be obtained by solving the primal Problem (13). Note that, for a fixed $z \in \mathbb{Z}_n^k$, the portfolio $x_i$ where $z_i = 0$ must be 0. Therefore, Problem (13) can be solved by only optimizing $O(k)$ variables. Since this problem is continuous, it can be solved the cutting-plane method proposed by Künzi-Bay and Mayer [23]. We describe the cutting-plane algorithm for evaluating $f(z)$ in Algorithm 1.

For the convergence of Algorithm 1, the number of constraint generated by each iteration is at most $2^S$. Thus, this algorithm terminates in a finite number of iterations and outputs an optimal solution to the subproblem (13) when we set a tolerance $\varepsilon_1 \geq 0$. Moreover, even if we set a tolerance $\varepsilon_1 > 0$, we can bound the gap between $f(z)$ and the objective function of the solution that Algorithm 1 outputs by the following theorem:

**Theorem 4.** Let $h(x, y, a)$ be the objective function (13a). For a given feasible $z \in \mathbb{Z}_n^k$, suppose that Algorithm 1 outputs a solution $(x_T, y_T, a_T)$ with a tolerance $\varepsilon_1 \geq 0$, the following holds:

$$h(x_T, y_T, a_T) \geq f(z) - \varepsilon_1.$$ (28)

**Proof.** According to Algorithm 1, it follows that:

$$y_T - \hat{y}_T \geq -\varepsilon_1.$$ (29)

Therefore, we can see the following:

$$h(x_T, y_T, a_T) - h(x_T, \hat{y}_T, a_T) = y_T - \hat{y}_T \geq -\varepsilon_1.$$ (30)
Next, we check the feasibility of the solution \((x_T, \hat{y}_T, a_T)\) to the subproblem (13). By the definition of \(\hat{y}_T\) described in Algorithm 1, we can see \(\hat{y}_T \geq 0\) and
\[
\hat{y}_T = \min_{s \in J} p(s)(-\mu(s)^\top x_T - a_T),
\]
\[
= \max_{J \subseteq S} \left\{ \sum_{s \in J} p(s)(-\mu(s)^\top x_T - a_T) \right\}.
\]
Therefore, the solution \((x_T, \hat{y}_T, a_T)\) is feasible to Problem (13), and we have
\[
h(x_T, \hat{y}_T, a_T) \geq f(z).
\]
Combining (30) and (33), we can obtain the desired result.

For a feasible \(z \in \mathbb{Z}_n^k\), suppose that Algorithm 1 output a solution \((x_{\varepsilon_1}^*, z), y_{\varepsilon_1}^*(z), a_{\varepsilon_1}^*(z))\) and generated the family \(\mathcal{S}_{\varepsilon_1}(z)\) with a tolerance \(\varepsilon_1 \geq 0\). Let us define a function as follows:
\[
f_{\varepsilon_1}(z) := h(x_{\varepsilon_1}^*(z), y_{\varepsilon_1}^*(z), a_{\varepsilon_1}^*(z)).
\]
Next, we consider the calculation of \(g(z) \in \partial f(z)\). At the previous Step, we have already solved Problem (13) by Algorithm 1. Then we solve the following problem to calculate the subgradient:
\[
\begin{align*}
\text{maximize} & \quad (14a) \\
\text{subject to} & \quad w \geq \sum_{J \subseteq \mathcal{S}_{\varepsilon_1}(z)} \alpha_J \left( \sum_{s \in J} p(s) \mu(s) \right) + A^\top (\beta_\ell - \beta_u) + \lambda e \\
& \quad \text{Constraints (14c), (14d), (14e), (14f).}
\end{align*}
\]
Contrary to Problem (14), Problem (34) includes \(O(\mathcal{S}_{\varepsilon_1}(z))\) variables; therefore, it can be solved efficiently. Let \(w_{\varepsilon_1}^*(z)\) to be an optimal solution of Problem (34), we define a vector \(g_{\varepsilon_1}(z)\) as follows:
\[
g_{\varepsilon_1}(z) := -\frac{1}{2\gamma}w_{\varepsilon_1}^*(z).
\]
Note that if we set \(\varepsilon_1 = 0\) for a tolerance, \(f_{\varepsilon_1}(z)\) and \(g_{\varepsilon_1}(z)\) are exactly the same as \(f(z)\) and \(g(z)\), respectively. Moreover, even if in the case of \(\varepsilon_1 > 0\), we can construct a lower approximation of \(f(z)\) with \(f_{\varepsilon_1}(z)\) and \(g_{\varepsilon_1}(z)\) as the following theorem:
Theorem 5. Suppose a feasible $\hat{z} \in \mathcal{Z}_n^k$ and $\varepsilon_1 \geq 0$ are given. Then, for all feasible $z \in \mathcal{Z}_n^k$ it holds that

$$f(z) \geq f_{\varepsilon_1}(\hat{z}) + g_{\varepsilon_1}(\hat{z})^\top (z - \hat{z}).$$

Proof. For a given feasible $z \in \mathcal{Z}_n^k$ and $\mathcal{S} \subseteq \{1, 2, \ldots, S\}$, let us define a function $f^S(z)$ that represents an optimal value of a relaxed problem of Problem (13) as follows:

$$f^S(z) := \minimize_{x,a,y} \frac{1}{2 \gamma} x^\top x + a + y$$

subject to

$$y \geq \frac{1}{1 - \delta} \sum_{s \in \mathcal{J}} p^{(s)}(-\mu^{(s)} \top x - a), \quad (\mathcal{J} \subseteq \mathcal{S}) \quad (36b)$$

$$y \geq 0, \quad (36c)$$

$$x \in \mathcal{X}_z. \quad (36d)$$

For all $z \in \mathcal{Z}_n^k$ and $\mathcal{S} \subseteq \{1, 2, \ldots, S\}$, it is clear that we have

$$f(z) \geq f^S(z). \quad (37)$$

Next, let us consider that for a feasible $\hat{z} \in \mathcal{Z}_n^k$ and $\varepsilon_1 \geq 0$, Algorithm 1 generates $\hat{S}$. Then, we have

$$f^\hat{S}(\hat{z}) = f_{\varepsilon_1}(\hat{z}). \quad (38)$$

Moreover, in the same way as we proved in Theorem 1 Lemma 2 and Lemma 3 for $f(z)$, $g_{\varepsilon_1}(z)$ is a subgradient of $f^\hat{S}(z)$ at $z$. Therefore, for all $z \in \mathcal{Z}_n^k$, it holds that

$$f^\hat{S}(z) \geq f_{\varepsilon_1}(\hat{z}) + g_{\varepsilon_1}(\hat{z})^\top (z - \hat{z}). \quad (39)$$

Then, by combining Equation (37) and Equation (39), we obtain the desired result. \qed

3.3. Bilevel cutting-plane algorithm

In this section, we describe our proposal, the bilevel cutting-plane algorithm for solving Problem (12). The bilevel cutting-plane algorithm is based on the cutting-plane algorithm proposed by Bertsimas and Cory-Wright [8].
for solving the cardinality-constrained mean-variance model. The key characteristics of our algorithm is to integrate two cutting-plane algorithms, upper-level cutting-plane algorithm and lower-level one, and each algorithm solves the master problem and the subproblem, respectively.

Our algorithm starts with calculating a lower bound of the optimal value of Problem (12). This can be done by solving a continuously relaxed problem:

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2\gamma} x^\top x + a + y \\
\text{subject to} & \quad x \in X, \\
& \quad \text{Constraints(10)}.
\end{align*}
\] (40a, 40b, 40c)

Since Problem (10) is continuous and the number of variables does not depend on the number of scenarios \(S\), this problem can be directly solved by the cutting-plane algorithm described in [23].

Let \(\theta_{\text{lower}}\) be the lower bound we calculated, then we start the upper-level cutting-plane algorithm and solve the following problem:

\[
\begin{align*}
\text{minimize} & \quad \theta \\
\text{subject to} & \quad (z, \theta) \in F, \quad (41)
\end{align*}
\]

where \(F\) is a relaxed feasible region defined as

\[
F := \{ (\theta, z) \in \mathbb{R} \times Z^k_n \mid \theta \geq \theta_{\text{lower}} \}. \quad (42)
\]

This relaxed problem is feasible and the objective function is bounded, an optimal solution \((\hat{z}, \hat{\theta})\) exists.

For the relaxed solution \(\hat{z}\), next we check the feasibility of the primal subproblem (13). If Problem (13) is infeasible for a fixed \(\hat{z}\), the solution \(\hat{z}\) is infeasible to Problem (12). Therefore, we update the feasible region \(F\) by cutting off the solution \(\hat{z}\) as follows:

\[
F \leftarrow F \cap \{ (\theta, z) \in \mathbb{R} \times Z^k_n \mid \hat{z}^\top z \leq e^\top \hat{z} \}. \quad (43)
\]

If Problem (13) is feasible on the other hand, we can calculate the function value \(f(\hat{z})\) and its subgradient \(g(\hat{z})\). As we described in Section 3.2, we can evaluate the function value \(f_{x_i}(\hat{z})\) by solving the primal problem (13) over \(k\) indices \(i\) where \(x_i = 1\) by applying Algorithm 1 with a tolerance \(\varepsilon_1 \geq 0\). After that, we calculate a subgradient of \(f\) at \(\hat{z}\). Let \(\hat{S}\) be the family of
scenario set generated by Algorithm 1. As we discussed in Section 3.2, the desired vector \( g_{\varepsilon_1}(\hat{z}) \) can be derived by solving the reduced dual problem Problem (34). Now that we obtain the value of \( f_{\varepsilon_1}(\hat{z}) \) and its subgradient \( g_{\varepsilon_1}(\hat{z}) \), we update the feasible region as follows:

\[
F \leftarrow F \cap \{(\theta, z) \in \mathbb{R} \times \mathcal{Z}_n^k \mid \theta \geq f_{\varepsilon_1}(\hat{z}) + g_{\varepsilon_1}(\hat{z})^\top (z - \hat{z})\}, \tag{44}
\]
and solve the master problem (41) again.

We repeat this procedure until the gap between the lower bound and the upper bound of \( f^* \) is sufficiently close to zero. Let \( LB_t \) and \( UB_t \) are that lower and upper bound at \( t \)-th iteration, respectively. It is clear that \( \hat{\theta} \) gives a lower bound, then, we update as \( LB_t \leftarrow \hat{\theta} \) at each iteration. For calculation of \( UB_t \), it needs to construct a feasible solution of Problem (13) for a fixed \( \hat{z} \).

Let \((x^*(\hat{z}), y^*(\hat{z}), a^*(\hat{z}))\) be the solution that Algorithm 1 output with the tolerance \( \varepsilon_1 \), an upper bound can be calculated by a sorting-based procedure proposed by Rockerfeller and Uryasev [31]. This procedure is described in Algorithm 2.

Now that, we summarise the entire procedure of our proposed bilevel cutting-plane algorithm in Algorithm 3.

Next, we discuss the convergence properties of Algorithm 3. First, we show the following lemma:

**Lemma 6.** For a sequence \( \{(z_t, \theta_t)\}_{t=1,2,...,T} \) generated by Algorithm 3 if \( z_{t'} \) \((1 \leq t' \leq T - 1)\) exists such that \( z_{t'} = z_T, (z_T, \theta_T) \) is an \( \varepsilon_1 \)-optimal solution to Problem (12), that is, the following holds:

\[
f^* + \varepsilon_1 \geq f(z_T).
\]

**Proof.** Recall that for any feasible \( z \in \mathcal{Z}_n^k \), we have

\[
f(z) \geq f_{\varepsilon_1}(z_t) + g_{\varepsilon_1}(z_t)^\top (z - z_t) \quad (t = 1, 2, \ldots, T - 1). \tag{45}
\]

Therefore, let \( f^* \) be an optimal value to Problem (12), it holds that

\[
f^* \geq \theta_T. \tag{46}
\]

Moreover, if \( z_{t'} \) \((1 \leq t' \leq T - 1)\) exists such that \( z_{t'} = z_T \), the following holds:

\[
\theta_T \geq f_{\varepsilon_1}(z_{t'}) = f_{\varepsilon_1}(z_T). \tag{47}
\]

16
From Equation (46) and Equation (47), we can see
\[ f^* \geq f_{\varepsilon_1}(z_T). \]  
(48)

In addition, according to Theorem 4, the following holds:
\[ f_{\varepsilon_1}(z_T) \geq f(z_T) - \varepsilon_1. \]  
(49)

Therefore, we have
\[ f^* + \varepsilon_1 \geq f(z_T), \]  
(50)
and \((\theta_T, z_T)\) is an \(\varepsilon_1\)-optimal solution.

From Lemma 6, we can show the following theorem:

**Theorem 7.** Suppose that \(\varepsilon_1 \geq 0\) and \(\varepsilon_2 \geq 0\). Then, Algorithm 3 terminates in a finite number of iterations and outputs \(\max\{\varepsilon_1, \varepsilon_2\}\)-optimal solution.

**Proof.** According to Lemma 6, we see that Algorithm 3 does not generate the same solution \(z \in \mathbb{Z}_n^k\) twice until \(\varepsilon_1\)-optimal solution is obtained. Thus, since the number of feasible solution \(z \in \mathbb{Z}_n^k\) is at most a finite, Lemma 6 terminates in a finite number of iterations and output \(\max\{\varepsilon_1, \varepsilon_2\}\)-optimal solution.

### 3.4. Practical implementation

Note that Algorithm 3 solves a linear MIO problem at every iteration, and therefore we must execute the branch-and-cut algorithm many times from scratch. To resolve this issue, we employ lazy constraint callbacks, which is offered by modern optimization software (e.g., CPLEX or Gurobi) to generate constraints dynamically during the branch-and-bound procedure as proposed by Quesada and Grossman [29].

### 4. Numerical Experiments

To verify the efficiency of our proposed algorithm, we conducted numerical experiments and reported the results in this section. For computational experiments, we downloaded four data sets from Kenneth R. French’s data library [20], and two data sets from OR-Library [19]. The list of these six instances are shown in Table 1.

For each the data set from Kenneth R. French’s digital library [20], We use monthly data from January 2010 to December 2019 to compute the
Table 1: Data description

| abbreviation | n  | original dataset                                      |
|--------------|----|-------------------------------------------------------|
| industry38   | 38 | 18 Industry Portfolios [20]                          |
| industry49   | 49 | 49 Industry Portfolios [20]                          |
| portfolio25  | 25 | 25 Portfolios Formed on Size and Book-to-Market (5 x 5) [20] |
| portfolio100 | 100| 100 Portfolios Formed on Size and Book-to-Market (10 x 10) [20] |
| port2        | 89 | OR-Library [19]                                      |
| port5        | 225| OR-Library [19]                                      |

mean vector $\mu$ and covariance matrix $\Sigma$ of the monthly expected rate of return. For the data set from OR-Library on the other hand, the mean vector and covariance matrix are given, and we multiplied them by 100 to align the scale of their values with those of Kenneth R. French’s data. Based on the mean vector $\mu$ and covariance matrix $\Sigma$ calculated, we randomly generated $S$ scenarios $\{\mu^{(s)}\}_{s=1}^S$ as $\mu^{(s)} \sim \mathcal{N}(\mu, \Sigma)$, where $\mathcal{N}(\mu, \Sigma)$ is a normal distribution that has a mean vector $\mu$ and covariance matrix $\Sigma$. Regarding to the significance level of CVaR, we set $\delta = 0.9$. In addition, we imposed a minimum return constraint on $x$, i.e., $\mu x \geq \bar{\mu}$, and the minimum return $\bar{\mu}$ is set in the following manner: Let $\mu_{\text{max}}$ and $\mu_{\text{min}}$ be the average of top and bottom $k$ expected returns and set $\bar{\mu} = \mu_{\text{min}} + 0.7(\mu_{\text{max}} - \mu_{\text{min}})$.

We compare the computational performance of the following methods:

**BigM** The big-$M$ formulation of Problem (8) with the lifted representation (9),

**SOCP** The perspective reformulation [15, 16] of Problem (8) with the lifted representation (9),

**CPA** Bertsimas and Cory-Wright’s cutting-plane algorithm [8] for Problem (8) with the lifted representation (9),

**CP+BigM** Künzi-Bay and Mayer’s algorithm [23] for Problem (11) with the big-$M$ formulation,

**CP+SOCP** Künzi-Bay and Mayer’s algorithm [23] for Problem (11) with the perspective reformulation [15, 16],

**BCPA** Bilevel cutting-plane algorithm (Algorithm 3) for solving (12).
All methods were implemented in Python 3.7 and used Gurobi version 8.1.1 for solving formulated problem. All experiments were performed on a Windows 10 PC with an Intel Core i7-4790 CPU (3.6.0GHz) and 16 GB of memory. The computation of each method was terminated if it did not finish by itself within 3600 seconds. In these cases, the results obtained at 3600s were taken as the final outcome.

The row labels used in the tables of experimental results are defined as follows:

**Time** computational time in seconds,

**Obj** best upper bound of the optimal value,

**Gap** relative gap between the upper bound and lower bound of the optimal value,

**#Nodes** number of nodes explored during the branch-and-bound procedure in Gurobi,

**#Cuts** number of cuts generated by callback function invocation.

### 4.1. Scalability with respect to the number of scenarios

First, we look at the scalability of each method with respect to the number of scenarios $S$. In this experiment, we set the upper bound of cardinality of $bmz$ and the parameter of $\ell_2$ regularization as $k = 10$ and $\gamma = 10/\sqrt{n}$, respectively.

Table 2 and Table 3 give the results for each data set with $S = 1000, 10000, 100000$ and $(\gamma, k) = (10/\sqrt{n}, 10)$. We can see from Table 2 and Table 3 that for the results of $S = 1000, 10000$, there is no clear inferior to superior relationship between each method for medium-sized data set instances where $n < 100$. For instance, in the case of industry48 with $S = 1000$ shown in Table 2, BCPA was the fastest among six methods. On the other hand, for the port2 with $S = 10000$ shown in Table 3, BigM was much faster than BCPA.

However, for the results of $S = 100000$, we can see that BCPA stably outperformed the other methods. As shown in Table 2 for the data portfolio100 with $S = 100000$, BCPA was more than ten times faster than BigM, SOCP and CPA. Additionally, in the case of port5 with $S = 100000$ shown in Table 3, BCPA was the only method that solved within the time limit, 3600 seconds, which is a remarkable result.
Since BCPA uses cutting-plane representation, the number of variables of the inner problem to be solved at every iteration does not depend on $S$. Meanwhile, in the case of BigM, SOCP and CPA, the number of variables to be optimized depends on $S$ due to lift-representation. Therefore, for large $S$, BigM, SOCP and CPA is computationally inefficient.

As Takeda and Kanamori [34] demonstrated, if we calculate CVaR accurately, a large number of scenarios are required via the scenario-based approximation. With this perspective, Table 2 and Table 3 verified the practicality of our proposed method, BCPA.

4.2. Sensitivity to hyperparemeters

Next, we explore the sensitivity of the performance of our proposed algorithm to the hyperparameters. In this experiments, we used three large-sized data sets industry49, portfolio100port5

4.2.1. $\ell_2$ regularization

We first investigate how the hyperparameter $\gamma$ affects the computational time. In this experiment, we set the upper bound of cardinality $k$ and the number of scenarios $S$ as $k = 10$ and $S = 100000$ Table 5 gives the results for each data set with $\gamma = 1/\sqrt{n}, 10/\sqrt{n}, 100/\sqrt{n}$ and $(S, k) = (100, 10)$. From Table 5, BCPA stably outperformed the other methods except for the case of portfolio100 with $\gamma = 100$. Moreover, for the performance of BCPA by itself, we can see that BCPA tended to be faster with a smaller $\gamma$. For the results of port5 in Table 5, BCPA obtained an optimal solution in about 800 seconds with $\gamma = 1/\sqrt{n}$, while it reached the time limit, 3600 seconds with $\gamma = 100/\sqrt{n}$.

4.2.2. cardinality constraints

Finally, we explore how the upper bound of cardinality $k$ affects computational time. In this experiment, we set the number of scenarios $k$ and the hyperparameter of $\ell_2$ regularization $\gamma$ as $S = 100000$ and $\gamma = 10/\sqrt{n}$, respectively. Table 4 gives the results for each data set with $k = 5, 10, 15$ and $(S, \gamma) = (100000, 10/\sqrt{n})$. From Table 4, BCPA outperformed the other methods for most of the cases. As for BCPA, we can see that BCPA tended to be faster with a larger $k$. For instance, in the case of the data port5 instance shown Table 4, BCPA was much faster for $k = 15$ than for $k = 5$. Let us look at the results for port5 in detail, we can see that #Nodes and #Cuts are both smaller for $k = 15$ than those for $k = 5$; this is part of the reason why the performance of BCPA was increased for large $k$. 20
5. Conclusion

This paper dealt with scenario-based mean-CVaR portfolio optimization problem with $\ell_2$ regularization and a cardinality constraint. While this problem can be posed as an MIO problem by using big-\textit{M} formulation or perspective reformulation, this approach does not work on large-scale problems. To deal with this issue, we developed a specialized cutting plane algorithm by extending the method proposed by Bertsimas and Cory-Wright [8]. Moreover, we integrated the cutting-plane algorithm proposed by Künzi-Bay and Mayer [23] into it for even higher computational efficiency. Note that our algorithms have a guarantee of global optimality, unlike heuristic optimization algorithms.

The computational results indicated that our cutting-plane algorithm was very effective at solving the problems when a large number of scenarios. Remarkably, our algorithm succeeded in attaining an optimal solution to the 225 assets with 100,000 scenarios in a reasonable amount of time. In the computational experiments, we also investigated the sensitivity of our proposed method to hyperparameters. As a result, our algorithm stably outperformed other methods.

A future direction of this study is to extend our algorithm so that it can handle non-convex transaction costs. Many practical problems are framed as scenario-based MILO problems, and our cutting-plane algorithms would be useful for solving them.

References

[1] K. C. Ágoston, CVaR minimization by the SRA algorithm, Central European Journal of Operations Research, 20 (2012), pp. 623–632.

[2] S. Ahmed, Convexity and decomposition of mean-risk stochastic programs, Mathematical Programming, 106 (2006), pp. 433–446.

[3] S. Alexander, T. F. Coleman, and Y. Li, Minimizing CVaR and VaR for a portfolio of derivatives, Journal of Banking and Finance, 30 (2006), pp. 583–605.

[4] K. Anagnostopoulos and G. Mamanis, The mean-variance cardinality constrained portfolio optimization problem: An experimental evaluation of five multiobjective evolutionary algorithms, Expert Systems with Applications, 38 (2011), pp. 14208–14217.
[5] E. Angelelli, R. Mansini, and M. G. Speranza, *A comparison of MAD and CVaR models with real features*, Journal of Banking and Finance, 32 (2008), pp. 1188–1197.

[6] P. Artzner, F. Delbaen, J. M. Eber, and D. Heath, *Coherent measures of risk*, Mathematical Finance, 9 (1999), pp. 203–228.

[7] G. Beliakov and A. Bagirov, *Non-smooth optimization methods for computation of the conditional value-at-risk and portfolio optimization*, Optimization, 55 (2006), pp. 459–479.

[8] D. Bertsimas and R. Cory-Wright, *A Scalable Algorithm For Sparse Portfolio Selection*, ArXiv e-prints, [http://arxiv.org/abs/1811.00138](http://arxiv.org/abs/1811.00138) (2018).

[9] D. Bertsimas and R. Shioda, *Algorithm for cardinality-constrained quadratic optimization*, Computational Optimization and Applications, 43 (2009), pp. 1–22.

[10] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*, Cambridge university press, 2004.

[11] R. Cheng and J. Gao, *On cardinality constrained mean-CVaR portfolio optimization*, Proceedings of the 2015 27th Chinese Control and Decision Conference, CCDC 2015, (2015), pp. 1074–1079.

[12] Y. Crama and M. Schyns, *Simulated annealing for complex portfolio selection problems*, European Journal of Operational Research, 150 (2003), pp. 546–571.

[13] C. I. Fábián, *Handling CVaR objectives and constraints in two-stage stochastic models*, European Journal of Operational Research, 191 (2008), pp. 888–911.

[14] J. Gao and D. Li, *Optimal cardinality constrained portfolio selection*, Operations Research, 61 (2013), pp. 745–761.

[15] O. Günlük and J. Linderoth, *Perspective reformulations of mixed integer nonlinear programs with indicator variables*, Mathematical Programming, 124 (2010), pp. 183–205.
[16] O. Günlük and J. Linderoth, *Perspective reformulation and applications*, in Mixed Integer Nonlinear Programming, Springer, 2012, pp. 61–89.

[17] W. K. Haneveld and M. H. van der Vlerk, *Integrated chance constraints: Reduced forms and an algorithm*, Computational Management Science, 3 (2006), pp. 245–269.

[18] G. Iyengar and A. K. C. Ma, *Fast gradient descent method for Mean-CVaR optimization*, Annals of Operations Research, 205 (2013), pp. 203–212.

[19] E. B. John, *Or-library: Distributing test problems by electronic mail*, The Journal of the Operational Research Society, 41 (1990), pp. 1069–1072.

[20] Kenneth, R. French, *Kenneth R. French - Data Library*. [https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html](https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html)

[21] H. Konno, H. Waki, and A. Yuuki, *Portfolio optimization under lower partial risk measures*, Asia-Pacific Financial Markets, 9 (2002), pp. 127–140.

[22] A. Krzemienowski and W. Ogryczak, *On extending the LP computable risk measures to account downside risk*, Computational Optimization and Applications, 32 (2005), pp. 133–160.

[23] A. Künzi-Bay and J. Mayer, *Computational aspects of minimizing conditional value-at-risk*, Computational Management Science, 3 (2006), pp. 3–27.

[24] C. Lim, H. D. Sherali, and S. Uryasev, *Portfolio optimization by minimizing Conditional Value-at-Risk via nondifferentiable optimization*, Computational Optimization and Applications, 46 (2010), pp. 391–415.

[25] R. Mansini, W. Ogryczak, and M. G. Speranza, *Twenty years of linear programming based portfolio optimization*, European Journal of Operational Research, 234 (2014), pp. 518–535.
[26] H. Markowitz, *Portfolio Selection*, The Journal of Finance, 7 (1952), pp. 77–91.

[27] W. Ogryczak and T. Śliwiński, *On solving the dual for portfolio selection by optimizing Conditional Value at Risk*, Computational Optimization and Applications, 50 (2011), pp. 591–595.

[28] G. C. Pflug, *Some Remarks on the Value-at-Risk and the Conditional Value-at-Risk*, (2000), pp. 272–281.

[29] I. Quesada and I. Grossmann, *An LP/NLP based branch and bound algorithm for convex MINLP optimization problems*, Computers and Chemical Engineering, 16 (1992), pp. 937–947.

[30] R. T. Rockafellar and S. Uryasev, *Optimization of conditional value-at-risk*, The Journal of Risk, 2 (2000), pp. 21–41.

[31] R. T. Rockafellar and S. Uryasev, *Conditional value-at-risk for general loss distributions*, Journal of Banking and Finance, 26 (2002), pp. 1443–1471.

[32] D. X. Shaw, S. Liu, and L. Kopman, *Lagrangian relaxation procedure for cardinality-constrained portfolio optimization*, Optimization Methods and Software, 23 (2008), pp. 411–420.

[33] Y. Takano, K. Nanjo, N. Sukegawa, and S. Mizuno, *Cutting plane algorithms for mean-CVaR portfolio optimization with nonconvex transaction costs*, Computational Management Science, 12 (2014), pp. 319–340.

[34] A. Takeda and T. Kanamori, *A robust approach based on conditional value-at-risk measure to statistical learning problems*, European Journal of Operational Research, 198 (2009), pp. 287–296.

[35] X. Tong, L. Qi, F. Wu, and H. Zhou, *A smoothing method for solving portfolio optimization with CVaR and applications in allocation of generation asset*, Applied Mathematics and Computation, 216 (2010), pp. 1723–1740.

[36] T. Wilding, *Using genetic algorithms to construct portfolios*, in Advances in Portfolio Construction and Implementation, S. Satchell and
A. Scowcroft, eds., Quantitative Finance, Butterworth-Heinemann, Oxford, 2003, pp. 135 – 160.
**Algorithm 1** The cutting-plane algorithm [23] for calculating the function value $f(z)$

**Require** Define $\varepsilon_1 \geq 0$ be a tolerance for feasibility, input $z \in \mathcal{Z}_n^k$.

**(Step 0)** Set $t \leftarrow 1$, and define a set of family and relaxed feasible region:

$$
S_t \leftarrow \{\{1, 2, \ldots, S\}\},
$$

$$
F^1_t \leftarrow \left\{ (x, y, a) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R} \mid y \geq \frac{1}{1-\delta} \sum_{s \in \mathcal{J}} (-\mu^{(s)}_t^\top x - a), \ (\mathcal{J} \in S_t) \right\}.
$$

**(Step 1)** Solve the following relaxed problem and obtain an optimal solution $(x_t, y_t, a_t, v_t)$ for the fixed $z \in \mathcal{Z}_n^k$

$$(x_t, y_t, a_t) \in \arg \min \left\{ \frac{1}{2\gamma} x^\top x + a + y \mid (x, y, a) \in F^1_t \right\}.
$$

**(Step 2)** Define a subset $\mathcal{J}_t \subseteq \{1, 2, \ldots, S\}$ as follows:

$$
\mathcal{J}_t \leftarrow \{ s \mid s = 1, 2, \ldots, S, \ -\mu^{(s)}_t^\top x_t - a_t > 0 \}.
$$

**(Step 3)** Calculate $\hat{y}_t$ as follows:

$$
\hat{y}_t \leftarrow \frac{1}{1-\delta} \sum_{s \in \mathcal{J}_t} p^{(s)} (-\mu^{(s)}_t^\top x_t - a_t).
$$

**(Step 4)** If $y_t - \hat{y}_t \geq -\varepsilon_1$, then $(x_t, y_t, a_t)$ is an $\varepsilon_1$-optimal solution, and terminate the algorithm. Otherwise, go to Step 5.

**(Step 5)** Update the feasible region:

$$
S_{t+1} \leftarrow S_t \cup \{\mathcal{J}_t\}
$$

$$
F_{t+1}^1 \leftarrow F_t^1 \cap \left\{ (x, y, a) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R} \mid y \geq \frac{1}{1-\delta} \sum_{s \in \mathcal{J}_t} (-\mu^{(s)}_t^\top x - a) \right\}.
$$

**(Step 6)** $t \leftarrow t + 1$, and return to Step 1.
Algorithm 2 Upper bound of estimation for solving Problem (12)

Require The solution \((\mathbf{x}^*_1(z), y^*_1(z), a^*_1(z))\) that Algorithm 1 outputs for a fixed feasible \(z \in \mathbb{Z}^k_n\).

(Step 1) Define a permutation \(\sigma\) of \(\{1, 2, \ldots, S\}\) such that the following losses are sorted in ascending order:

\[
N_s := -\mathbf{\mu}^{(s)} \top \mathbf{x}^*_1(z), (s = 1, 2, \ldots S).
\]

That is, \(N_{\delta(1)} \leq N_{\delta(2)} \leq \cdots \leq N_{\delta(S)}\).

(Step 2) Set an integer \(\tau := \min \{i \in \mathbb{Z} \mid \delta S \leq i\}\). Then, set \(a' := N_{\delta(\tau)}\) and \(y'\) as

\[
y := a' + \frac{1}{(1 - \delta)} \sum_{s=1}^{T} p^{(s)}[N_s - a']_+.
\]

(Step 3) Calculating an upper bound by substituting a feasible solution into the objective function appeared in Problem (13):

\[
\text{UB} := \mathbf{x}^*_1(z) \top \mathbf{x}^*_1(z) + a' + y'.
\]
Algorithm 3 Bilevel Cutting-plane Algorithm

(Require) Define two tolerances \( \varepsilon_1 \geq 0 \) and \( \varepsilon_2 \geq 0 \).

(Step 0) Set \( t \leftarrow 1 \), \( UB_1 \leftarrow \infty \), and \( LB_1 \) as an optimal value of Problem (40), and define an initial feasible region
\[
F_1 \leftarrow \{ (z, \theta) \in \mathbb{R}^n \times \mathcal{Z}_n^{k} \mid \theta \geq LB_1 \},
\]

(Step 1) Solve the following relaxed problem and obtain an optimal solution \( z_t, \theta_t \):
\[
(z_t, \theta_t) \in \arg \min_{(z, \theta) \in \mathcal{Z}_n^{k} \times \mathbb{R}} \{ \theta \mid (z, \theta) \in F_t \}.
\]
Then, update the lower bound \( LB_{t+1} \leftarrow \theta_t \).

(Step 2) Execute Algorithm 1 with a tolerance \( \varepsilon_1 \geq 0 \) to obtain a \( \varepsilon_1 \)-solution \( (x_{\varepsilon_1}^*, y_{\varepsilon_1}^*(z_t), a_{\varepsilon_1}^*(z_t)) \) to Problem (13) and calculate the function value \( f_{\varepsilon_1}(z_t) \) and subgradient \( g_{\varepsilon_1}(z_t) \)

- If Problem (13) is infeasible for \( z_t \), cut off the infeasible solution:
\[
F_{t+1} \leftarrow F_t \cap \{ (z, \theta) \in \mathbb{R} \times \mathcal{Z}_n^{k} \mid z^T z \leq 1^T z_t \}.
\]
Then, update \( UB_{t+1} \leftarrow UB_t \), and go to Step 5.

- Otherwise, go to Step 3.

(Step 3) Calculate UB by executing Algorithm 2 and update \( UB_{t+1} \leftarrow \min\{UB_t, UB\} \).

(Step 4) If there is \( t' \) (1 \( \leq t' \leq t \)) such that \( z_{t'} = z_t \) or \( UB_{t+1} - LB_{t+1} \leq \varepsilon_2 \), \( z_t \) is an \( \max\{\varepsilon_1, \varepsilon_2\} \)-solution to Problem (12) and terminate the algorithm. Otherwise, update the feasible region:
\[
F_{t+1} \leftarrow F_t \cap \{ (\theta, z) \in \mathbb{R} \times \mathcal{Z}_n^{k} \mid \theta \geq f_{\varepsilon_1}(z_t) + g_{\varepsilon_1}(z_t)^T (z - z_t) \}.
\]

(Step 5) \( t \leftarrow t + 1 \) and return to Step 1.
Table 2: Results for the problem instances from [20] with \((\gamma, k) = (10/\sqrt{n}, 10)\)

| data  | n   | S   | BigM | SOCP | CPA | CP+BigM | CP+SOCP | BCPA |
|-------|-----|-----|------|------|-----|---------|---------|------|
| industry38 | 38  | 1000| 1.3  | 1.7  | 2.4 | 3.4     | 2.6     | 1.4  |
|         |     |     | Time |      |     |         |         |      |
| Obj    | 3.989 | 3.989 | 3.989 | 3.989 | 3.989 | 3.989 | 3.989 | 3.989 |
| Gap(%) | 0.00  | 0.00  | 0.00  | 0.00  | 0.00  | 0.00   | 0.00   | 0.00 |
| #Nodes | 1     | 1    | 19   | 2693 | 3059 | 0      | 0      | 0    |
| #Cuts  | —     | —    | —    | 505  | 274  | 7      | 7      |      |
| 100000 |     |     |      | 14.0 | 23.5 | 23.1   | 15.9    | 6.7  |
| Obj    | 3.820 | 3.820 | 3.820 | 3.820 | 3.820 | 3.820 | 3.820 | 3.820 |
| Gap(%) | 0.00  | 0.00  | 0.00  | 0.00  | 0.00  | 0.00   | 0.00   | 0.00 |
| #Nodes | 1     | 1    | 11   | 3362 | 1597 | 0      | 0      | 0    |
| #Cuts  | —     | —    | —    | 479  | 178  | 7      | 7      |      |
| 1000000 |    |     |      | 493.9 | 3611.5 | 284.3 | 98.4   | 57.8 |
| Obj    | 3.860 | 3.860 | 3.860 | 3.860 | 3.860 | 3.860 | 3.860 | 3.860 |
| Gap(%) | 0.00  | 0.00  | 0.00  | 0.00  | 0.00  | 0.00   | 0.00   | 0.00 |
| #Nodes | 1     | 1    | 21   | 2327 | 1139 | 0      | 0      | 0    |
| #Cuts  | —     | —    | —    | 338  | 193  | 7      | 7      |      |
| industry49 | 49  | 1000| 1.8  | 2.3  | 6.1 | 4.8     | 11.8    | 8.6  |
|         |     |     | Time |      |     |         |         |      |
| Obj    | 3.095 | 3.095 | 3.095 | 3.095 | 3.095 | 3.095 | 3.095 | 3.095 |
| Gap(%) | 0.00  | 0.00  | 0.00  | 0.00  | 0.00  | 0.00   | 0.00   | 0.00 |
| #Nodes | 16    | 23   | 211  | 3792 | 4506 | 0      | 0      | 0    |
| #Cuts  | —     | —    | —    | 573  | 1101 | 45     | 45     |      |
| 1000000 |    |     |      | 709.8 | 3605.2 | 662.2 | 125.0  | 220.0 |
| Obj    | 3.379 | 3.379 | 3.379 | 3.379 | 3.379 | 3.379 | 3.379 | 3.379 |
| Gap(%) | 0.00  | 0.00  | 0.00  | 0.00  | 0.00  | 0.00   | 0.00   | 0.00 |
| #Nodes | 31    | 1    | 89   | 4101 | 3489 | 0      | 0      | 0    |
| #Cuts  | —     | —    | —    | 380  | 661  | 12     | 12     |      |
| portfolio25 | 25  | 1000| 1.3  | 1.7  | 1.3 | 0.1     | 0.2     | 0.7  |
|         |     |     | Time |      |     |         |         |      |
| Obj    | 5.017 | 5.017 | 5.017 | 5.017 | 5.017 | 5.017 | 5.017 | 5.017 |
| Gap(%) | 0.00  | 0.00  | 0.00  | 0.00  | 0.00  | 0.00   | 0.00   | 0.00 |
| #Nodes | 1     | 1    | 0    | 129  | 0    | 0      | 0      | 0    |
| #Cuts  | —     | —    | —    | 17   | 14   | 8      | 8      |      |
| 1000000 |    |     |      | 11.8 | 21.5 | 11.3   | 0.5     | 0.5  |
| Obj    | 4.931 | 4.931 | 4.931 | 4.931 | 4.931 | 4.931 | 4.931 | 4.931 |
| Gap(%) | 0.00  | 0.00  | 0.00  | 0.00  | 0.00  | 0.00   | 0.00   | 0.00 |
| #Nodes | 1     | 1    | 0    | 1    | 0    | 1      | 0      | 0    |
| #Cuts  | —     | —    | —    | 17   | 14   | 8      | 8      |      |
| portfolio100 | 100 | 1000| 3.3  | 3.7  | 4.9 | 1.1     | 4.2     | 0.8  |
|         |     |     | Time |      |     |         |         |      |
| Obj    | 4.337 | 4.337 | 4.337 | 4.337 | 4.338 | 4.338 | 4.338 | 4.337 |
| Gap(%) | 0.00  | 0.00  | 0.00  | 0.00  | 0.00  | 0.00   | 0.00   | 0.00 |
| #Nodes | 1     | 1    | 14   | 396  | 4626 | 0      | 0      | 0    |
| #Cuts  | —     | —    | —    | 70   | 189  | 7      | 7      |      |
| 1000000 |    |     |      | 39.1 | 45.9 | 51.1   | 8.1     | 9.3  |
| Obj    | 4.397 | 4.397 | 4.397 | 4.397 | 4.397 | 4.397 | 4.397 | 4.397 |
| Gap(%) | 0.00  | 0.00  | 0.00  | 0.00  | 0.00  | 0.00   | 0.00   | 0.00 |
| #Nodes | 1     | 1    | 58   | 1820 | 3274 | 0      | 0      | 0    |
| #Cuts  | —     | —    | —    | 103  | 107  | 7      | 7      |      |
| 1000000 |    |     |      | 1107.1 | >3600 | 482.7 | 65.1    | 84.6 |
| Obj    | 4.364 | 4.364 | 4.364 | 4.364 | 4.364 | 4.364 | 4.364 | 4.364 |
| Gap(%) | 0.00  | 0.00  | 0.00  | 0.00  | 0.00  | 0.00   | 0.00   | 0.00 |
| #Nodes | 1     | >1   | 10   | 412  | 4676 | 0      | 0      | 0    |
| #Cuts  | —     | —    | —    | 120  | 152  | 7      | 7      |      |
Table 3: Results for the problem instances from [19] with \((\gamma, k) = (10/\sqrt{n}, 10)\)

| data | port1 | port2 | port5 |
|------|-------|-------|-------|
|  n   | 30    | 89    | 225   |
| S    | 1000  | 1000  | 1000  |
| Time |       |       |       |
|      | 1.1   | 7.4   | 8.3   |
|      | 1.5   | 7.5   | 7.9   |
| BigM| 4.404 | 4.374 | 2.933 |
| SOCP| 4.404 | 4.374 | 2.933 |
| CPA | 4.404 | 4.374 | 2.933 |
| CP-| 4.404 | 4.374 | 2.933 |
| BigM| 4.404 | 4.374 | 2.933 |
| CP+SOCP| 4.404 | 4.374 | 2.933 |
| BCPA| 4.404 | 4.374 | 2.933 |
| Obj | 4.404 | 4.374 | 2.933 |
| Gap(%) | 0.00 | 0.00 | 0.00 |
| #Nodes | 1 | 1 | 368 |
| #Cuts | — | — | — |
| 10000 |       |       |       |
| Time | 11.7  | 468.6 | 311.2 |
|      | 24.0  | 1933.2| 429.0 |
| BigM| 4.374 | 4.264 | 2.624 |
| SOCP| 4.374 | 4.264 | 2.624 |
| CPA | 4.374 | 4.264 | 2.624 |
| CP-| 4.374 | 4.264 | 2.624 |
| BigM| 4.374 | 4.264 | 2.624 |
| CP+SOCP| 4.374 | 4.264 | 2.624 |
| BCPA| 4.374 | 4.264 | 2.624 |
| Obj | 4.374 | 4.264 | 2.624 |
| Gap(%) | 0.00 | 0.00 | 0.00 |
| #Nodes | 1 | 1 | 1432 |
| #Cuts | — | — | — |
| 100000 |       |       |       |
| Time | 468.6 | 468.6 | 311.2 |
|      | 1933.2| 1933.2| 429.0 |
| BigM| 4.264 | 4.264 | 2.624 |
| SOCP| 4.264 | 4.264 | 2.624 |
| CPA | 4.264 | 4.264 | 2.624 |
| CP-| 4.264 | 4.264 | 2.624 |
| BigM| 4.264 | 4.264 | 2.624 |
| CP+SOCP| 4.264 | 4.264 | 2.624 |
| BCPA| 4.264 | 4.264 | 2.624 |
| Obj | 4.264 | 4.264 | 2.624 |
| Gap(%) | 0.00 | 0.00 | 0.00 |
| #Nodes | 1 | 1 | 1432 |
| #Cuts | — | — | — |
| Data          | $n$  | $\gamma$ | BigM | SOCP | CPA | CP+BigM | CP+SOCP | BCPA |
|--------------|------|----------|------|------|-----|---------|---------|------|
| **industry49** | 49   | $l/\sqrt{n}$ | Time | >10000 | 10000 | 10000 | 10000 | 10000 |
|              |      |          | Obj  | 3.823 | 3.824 | 3.823 | 3.823 | 3.823 |
|              |      |          | Gap(%) | 0.00 | 0.12 | 0.00 | 0.00 | 0.00 |
|              |      |          | #Nodes | 483 | >6 | 656 | 4560 | 696 |
| **10/\sqrt{n}** |      |          | Obj  | 3.379 | 3.380 | 3.379 | 3.380 | 3.380 |
|              |      |          | Gap(%) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
|              |      |          | #Nodes | 31 | 1 | 25 | 4101 | 380 |
| **100/\sqrt{n}** |      |          | Obj  | 3.322 | 3.322 | 3.322 | 3.322 | 3.322 |
|              |      |          | Gap(%) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
|              |      |          | #Nodes | 1 | 1 | 2391 | 5273 | 1172 |
| **portfolio100** | 100  | 1/\sqrt{n} | Time | >10000 | >10000 | 10000 | 10000 | 10000 |
|              |      |          | Obj  | 5.446 | 5.075 | 5.075 | 5.075 | 5.075 |
|              |      |          | Gap(%) | 7.19 | 0.01 | 0.00 | 0.00 | 0.00 |
|              |      |          | #Nodes | >1 | >5 | 21 | 175 | 175 |
| **10/\sqrt{n}** |      |          | Obj  | 4.304 | 4.304 | 4.304 | 4.304 | 4.304 |
|              |      |          | Gap(%) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
|              |      |          | #Nodes | 1 | >1 | 10 | 412 | 120 |
| **100/\sqrt{n}** |      |          | Obj  | 4.225 | 4.225 | 4.225 | 4.225 | 4.225 |
|              |      |          | Gap(%) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
|              |      |          | #Nodes | 1 | 1 | 19 | 634 | 252 |
| **port5** | 225  | 1/\sqrt{n} | Time | >10000 | >10000 | >10000 | >10000 | >10000 |
|              |      |          | Obj  | ∞ | 3.143 | 3.173 | 3.246 | 3.276 |
|              |      |          | Gap(%) | 100.00 | 4.97 | 2.69 | 9.96 | 0.00 |
|              |      |          | #Nodes | >0 | >1 | >211 | >18422 | >2182 |
| **10/\sqrt{n}** |      |          | Obj  | 100.00 | 0.44 | 2.05 | 3.98 | 4.29 |
|              |      |          | Gap(%) | 100.00 | 0.00 | 0.00 | 0.00 | 0.00 |
|              |      |          | #Nodes | >0 | >61 | >2140 | >20833 | >20833 |
| **100/\sqrt{n}** |      |          | Obj  | 100.00 | 0.30 | 21.44 | 100.00 | 1.79 |
|              |      |          | Gap(%) | 100.00 | 0.00 | 0.00 | 0.00 | 0.00 |
|              |      |          | #Nodes | >0 | >31 | >826 | >11144 | >14020 |

Table 4: Results for the three problem instances with $(S, k) = (10000, 10)$
Table 5: Results for the three problem instances with (S, γ) = (10000, 10/√n)

| data      | n  | k | Time   | BigM | SOCP | CPA | CP+BigM | CP+SOCP | BCPA |
|-----------|----|---|--------|------|------|-----|---------|---------|------|
| industry49| 49 | 5 |        | 733.7| >3600| 1455.9| **160.0**| 4343.3  | 907.5|
|           |    |   |        | 3.443| 3.443| 3.443| 3.443    | 3.443    |      |
|           |    |   | Gap(%) | 0.00 | 0.00 | 0.00 | 0.00     | 0.00     | 0.00 |
|           |    |   | #Nodes | 83   | >61  | 2614 | 2475     | 5172     | 970  |
|           |    |   | #Cuts  | —    | —    | —    | 428      | 1197     | 172  |
| 10        |    |   | Time   | 1097.8| 3605.2| 662.2 | 125.0     | 220.0    | **103.6**|
|           |    |   | Gap(%) | 0.00 | 0.00 | 0.00 | 0.00     | 0.00     | 0.00 |
|           |    |   | #Nodes | 31   | 1    | 250  | 4101     | 3489     | 0    |
|           |    |   | #Cuts  | —    | —    | —    | 380      | 661      | 12   |
| 15        |    |   | Time   | 585.2| >3600| 448.6 | 128.2     | **76.7** | 82.3 |
|           |    |   | Gap(%) | 0.00 | 0.00 | 0.00 | 0.00     | 0.00     | 0.00 |
|           |    |   | #Nodes | 1    | >1   | 67   | 3392     | 1043     | 0    |
|           |    |   | #Cuts  | —    | —    | —    | 363      | 211      | 7    |
| portfolio100| 100 | 5 | Time   | 1260.7| 2941.5| 690.8 | 179.8     | 147.8    | **48.7**|
|           |    |   | Obj    | 4.367| 4.367| 4.367| 4.367    | 4.367    | 4.367 |
|           |    |   | Gap(%) | 0.00 | 0.00 | 0.00 | 0.00     | 0.00     | 0.00 |
|           |    |   | #Nodes | 92   | 3    | 116  | 1092     | 1866     | 0    |
|           |    |   | #Cuts  | —    | —    | —    | 296      | 239      | 12   |
| 10        |    |   | Time   | 1107.1| >3600| 482.7 | 65.1      | 84.6     | **36.6**|
|           |    |   | Obj    | 4.364| 4.364| 4.364| 4.364    | 4.364    | 4.364 |
|           |    |   | Gap(%) | 0.00 | 0.00 | 0.00 | 0.00     | 0.00     | 0.00 |
|           |    |   | #Nodes | 1    | >1   | 10   | 412      | 4676     | 0    |
|           |    |   | #Cuts  | —    | —    | —    | 120      | 152      | 7    |
| 15        |    |   | Time   | 1051.6| 3438.1| 437.3 | **45.2** | 44.4     | 45.9 |
|           |    |   | Obj    | 4.364| 4.364| 4.364| 4.364    | 4.364    | 4.364 |
|           |    |   | Gap(%) | 0.00 | 0.00 | 0.00 | 0.00     | 0.00     | 0.00 |
|           |    |   | #Nodes | 1    | 1    | 9    | 90       | 5738     | 0    |
|           |    |   | #Cuts  | —    | —    | —    | 70       | 64       | 7    |
| port5     | 225 | 5 | Time   | >3600| >3600| >3600 | >3600    | **2263.9**| >3600 |
|           |    |   | Obj    | ∞    | 3.755| 3.397| 3.491    | 3.295    | 3.295 |
|           |    |   | Gap(%) | 100.00| 14.25| 8.52  | 8.72     | 0.00     | 3.37 |
|           |    |   | #Nodes | >0   | >42  | >1390| >41433   | 19851    | >10767|
|           |    |   | #Cuts  | —    | —    | —    | >2644    | 1649     | >861 |
| 10        |    |   | Time   | >3600| >3600| >3600 | >3600    | **3300.8**| >3600 |
|           |    |   | Obj    | ∞    | 3.143| 3.173| 3.246    | 3.276    | 3.358 |
|           |    |   | Gap(%) | 100.00| 0.44| 2.05  | 3.98     | 4.29     | 0.00 |
|           |    |   | #Nodes | >0   | >61  | >2140| >20833   | >16790   | 5697 |
|           |    |   | #Cuts  | —    | —    | —    | >2409    | >2933    | 519  |
| 15        |    |   | Time   | >3600| 3817.5| >3600 | 1154.4   | **1156.1**| **121.4**|
|           |    |   | Obj    | ∞    | 3.109| 3.170| 3.109    | 3.109    | 3.109 |
|           |    |   | Gap(%) | 100.00| 0.00| 1.97  | 0.00     | 0.00     | 0.00 |
|           |    |   | #Nodes | >0   | 1    | >2516| 8454     | 5696     | 0    |
|           |    |   | #Cuts  | —    | —    | —    | 757      | 838      | 11   |