Department of Industrial Engineering and Management

Technical Report

No. 2013-11

Cutting Plane Algorithms for Mean-CVaR Portfolio Optimization with Nonconvex Transaction Costs

Yuichi Takano, Keisuke Nanjo,
Noriyoshi Sukegawa, and Shinji Mizuno

October, 2013

Tokyo Institute of Technology

2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552, JAPAN
http://www.me.titech.ac.jp/index-e.html

Cutting Plane Algorithms for Mean-CVaR Portfolio Optimization with Nonconvex Transaction Costs

Yuichi Takano takano.y.ad@m.titech.ac.jp
Department of Industrial Engineering and Management
Graduate School of Decision Science and Technology, Tokyo Institute of Technology
2-12-1-W9-77 Ookayama, Meguro-ku, Tokyo 152-8552, JAPAN

Keisuke Nanjo nanjo.k.aa@m.titech.ac.jp
Department of Industrial Engineering and Management
Graduate School of Decision Science and Technology, Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552, JAPAN

Noriyoshi Sukegawa sukegawa.n.aa@m.titech.ac.jp
Department of Industrial Engineering and Management
Graduate School of Decision Science and Technology, Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552, JAPAN

Shinji Mizuno mizuno.s.ab@m.titech.ac.jp
Department of Industrial Engineering and Management
Graduate School of Decision Science and Technology, Tokyo Institute of Technology
2-12-1-W9-58 Ookayama, Meguro-ku, Tokyo 152-8552, JAPAN

Abstract

This paper studies a scenario-based mean-CVaR portfolio optimization problem with nonconvex transaction costs. This problem can be framed as a mixed integer linear programming (MILP) problem by making a piecewise linear approximation of the transaction cost function. Nevertheless, large-scale problems are computationally intractable even with state-of-the-art MILP solvers. To efficiently solve them, we devised a subgradient-based cutting plane algorithm. We also devised a two-phase cutting plane algorithm that is even more efficient. Numerical experiments demonstrated that our algorithms can attain near-optimal solutions to large-scale problems in a reasonable amount of time.

Keywords: Portfolio optimization, Conditional value-at-risk, Cutting plane algorithm, Transaction costs, Mixed integer linear programming

1 Introduction

The theory of portfolio selection is widely used in the financial industry, and it is actively studied by academic researchers and institutional investors. The traditional framework created by Markowitz [20] determines the asset allocation with the aim of making low-risk and high-return investments. This paper addresses the mean-risk portfolio optimization model using the conditional value-at-risk (CVaR) [23, 24], 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 (see [5, 22] for details). In addition, it is monotonic with respect to second-order stochastic dominance (see [22]). This means that CVaR minimization is consistent 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.

To avoid numerical difficulties resulting from multiple integration, CVaR is usually calculated by making a scenario-based approximation. The associated portfolio optimization problem of minimizing scenario-based CVaR can be formulated as a linear programming (LP) problem [23, 24]; however, this formulation, called the lifting representation [10], additionally requires as many decision variables and constraints as the number of scenarios. Takeda and Kanamori [26] analyzed the convergence properties of scenario-based CVaR. Their results imply that a huge number of scenarios must be generated in order to calculate CVaR accurately via the scenario-based approximation. More importantly, Kaut et al. [12] showed that at least 5,000 scenarios are necessary to ensure stability of the CVaR optimization model even if the number of investable assets is only 15. Accordingly, scenario-based CVaR requires a sufficiently large number of scenarios, which inevitably makes the lifting representation computationally intractable.

There are a number of studies that aim at efficiently solving large-scale CVaR minimization problems, e.g., nonsmooth optimization approaches [7, 11, 18, 23], scenario representation by factor model [14], cutting plane algorithms [2, 17], level method [9], smoothing methods [3, 27] and successive regression approximations [1]. We should however notice that none of these studies take into account nonconvex transaction costs, which are present in practical situations.

Transaction costs typically consist of brokerage commissions, taxes, and market impact costs (or illiquidity effects), and thus, they can be represented as a separable, nonlinear, nonconvex function of purchases or sales of assets (see e.g., Perold [21]). If one constructs a portfolio without considering such transaction costs, the arising profit might be wiped out by them. Consequently, a number of studies have pondered the inclusion of nonconvex transaction costs in portfolio optimization models (see, e.g., [8, 13, 15, 16, 19, 28]). Among them, Konno and Yamamoto [16] propose a mixed integer linear programming (MILP) formulation, where special ordered set type two (SOS2) constraints [6] are utilized to represent piecewise linear transaction cost functions. Since this formulation can deal with various nonlinear transaction cost functions, we shall focus on it here.

The purpose of the present paper is to devise an efficient algorithm with a guarantee of global optimality for the mean-CVaR portfolio optimization problem with nonconvex transaction costs. We shall develop a subgradient-based cutting plane algorithm, similarly to [2, 17], because it can be readily applied to MILP formulations. However, the cutting plane algorithm needs to solve an MILP problem in each iteration, and this requires a substantial computation time. Therefore, we will also devise a two-phase cutting plane algorithm that is specialized for MILP formulations and has higher computational efficiency. In the first phase, piecewise linear transaction cost functions are replaced with convex underestimators. As a result, in each iteration, the algorithm solves a relaxed problem without SOS2 constraints, which is an efficiently solvable LP problem. The second phase solves the original problem with SOS2 constraints. Although this phase needs to solve an MILP problem every iteration, the effective cuts that were added in the first phase considerably reduce the number of problems to be solved.

We conducted computational experiments assessing the performance of our cutting plane algorithms. Here, we solved two portfolio optimization problems, i.e., an initial investment problem and a rebalancing problem. The results show that standard MILP formulations based on the lifting representation easily lead to memory shortages when many scenarios are considered. By contrast, even for large-scale problems, our cutting plane algorithms reach a near-optimal solution with satisfactory accuracy in a reasonable amount of time. Moreover, when rebalancing a current portfolio that is close to an optimal one, our algorithms are clearly superior to other solution methods in terms of computation time.

The rest of the paper is organized as follows: In Section 2, we formulate the mean-CVaR portfolio optimization problem with nonconvex transaction costs. Section 3 is devoted to our cutting plane algorithms for solving the problem. The computational results are reported in Section 4. Finally, conclusions are given in Section 5.

2 Problem Formulation

2.1 Preliminaries

Let $\boldsymbol{x}^0 = (x_1^0, x_2^0, ..., x_I^0)^\top$ be the current portfolio, where x_i^0 is the investment proportion in financial asset $i = 1, 2, ..., I$. If one has no current portfolio, $\boldsymbol{x}^0$ is set to a zero vector $\mathbf{0}$. This paper tackles the problem of rebalancing the current portfolio $\boldsymbol{x}^0$ into a portfolio $\boldsymbol{x} = (x_1, x_2, ..., x_I)^\top$ for low-risk high-return investments in the presence of transaction costs.

The transaction cost is denoted by $\mathcal{C}_i(x_i - x_i^0)$, which is a function of the size of the transaction $x_i - x_i^0$ of each asset $i = 1, 2, ..., I$ (see Figure 1). When the transaction is small, brokerage commissions account for a large portion of the transaction cost, and the cost per unit accordingly decreases as the transaction increases. By contrast, when the transaction is large, the market impact (or illiquidity effects) drastically increases the transaction cost.

The net return of the portfolio is expressed as

$$\tilde{\boldsymbol{r}}^\top \boldsymbol{x} - \sum_{i=1}^{I} \mathcal{C}_i(x_i - x_i^0) = \sum_{i=1}^{I} (\tilde{r}_i x_i - \mathcal{C}_i(x_i - x_i^0)), \tag{1}$$

where $\tilde{\boldsymbol{r}} = (\tilde{r}_1, \tilde{r}_2, ..., \tilde{r}_I)^\top$ is a random vector representing the rate of return of each asset.

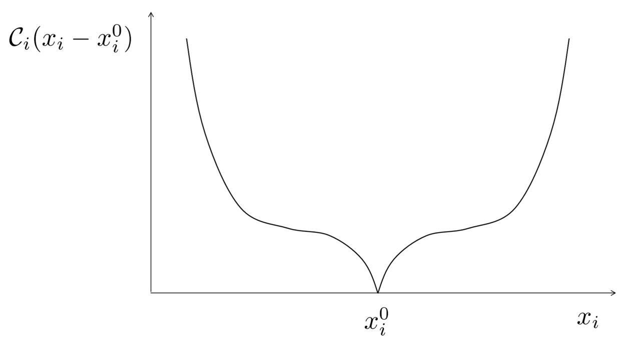


Figure 1: Illustration of transaction cost function

In addition, we denote by $\mathcal{X}$ the set of feasible portfolios. The following constraints are dealt with in what follows:

$$\sum_{i=1}^{I} x_i = 1;\ X_i^{\min} \le x_i \le X_i^{\max},\ \forall i = 1, 2, ..., I,$$

where $X_i^{\min}$ and $X_i^{\max}$ are the lower and upper limits of the investment proportion in asset i, respectively. In accordance with actual practice, it is assumed throughout this paper that short selling is prohibited, i.e., $X_i^{\min} \ge 0,\ \forall i = 1, 2, ..., I$.

2.2 Conditional value-at-risk

Let $\beta \in (0, 1)$ be a parameter representing the confidence level, which is frequently set close to one. Accordingly, β-CVaR can approximately be regarded as the conditional expectation of a random loss exceeding the β-value-at-risk (β-VaR), which is the β-quantile of the random loss (see Figure 2). The loss function $\mathcal{L}(\boldsymbol{x}, \tilde{\boldsymbol{r}})$ is defined as the negative of the portfolio net return (1):

$$\mathcal{L}(\boldsymbol{x}, \tilde{\boldsymbol{r}}) := -\sum_{i=1}^{I} (\tilde{r}_i x_i - \mathcal{C}_i(x_i - x_i^0)). \tag{2}$$

Since β-CVaR is a risk measure for evaluating heavy losses that occur with a low probability (i.e., $1 - \beta$), minimization of CVaR will mitigate the risk of suffering such a heavy loss.

The minimum CVaR of the loss function (2) is calculated as the optimal objective function value of the following problem (see [23, 24]):

$$\left|\begin{array}{ll} \underset{a,\,\boldsymbol{x}}{\text{minimize}} & \mathcal{F}_\beta(a, \boldsymbol{x}) := a + \dfrac{1}{(1-\beta)} \displaystyle\int_{\boldsymbol{r} \in \mathbb{R}^I} [\mathcal{L}(\boldsymbol{x}, \boldsymbol{r}) - a]_+ \, \mathcal{P}(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r} \\ \text{subject to} & \boldsymbol{x} \in \mathcal{X}, \end{array}\right.$$

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

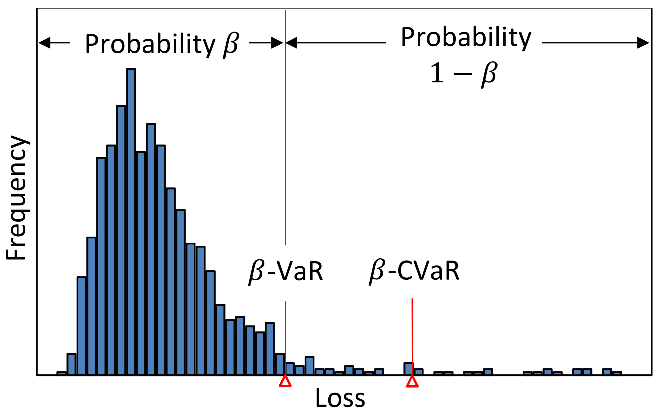


Figure 2: Conditional value-at-risk

Multiple integration in $\mathcal{F}_\beta(\boldsymbol{x}, a)$ is computationally burdensome; hence, we often use the following scenario-based approximation:

$$\begin{aligned}\mathcal{F}_\beta(a, \boldsymbol{x}) &\approx a + \frac{1}{(1-\beta)S} \sum_{s=1}^{S} \left[\mathcal{L}(\boldsymbol{x}, \boldsymbol{R}^{(s)}) - a\right]_+ \\ &= a + \frac{1}{(1-\beta)S} \sum_{s=1}^{S} \left[-\sum_{i=1}^{I} \left(R_i^{(s)} x_i - \mathcal{C}_i(x_i - x_i^0)\right) - a\right]_+,\end{aligned}$$

where $\boldsymbol{R}^{(s)} = (R_1^{(s)}, R_2^{(s)}, ..., R_I^{(s)})^\top$, $s = 1, 2, ..., S$ are scenarios of the rate of return generated from the probability density function $\mathcal{P}$.

2.3 Piecewise linear transaction cost

As mentioned in Section 2.1, transaction cost functions are generally nonconvex. In what follows, we present a mixed integer linear programming (MILP) formulation for approximating nonconvex transaction costs by piecewise linear functions. We first address the initial investment problem (i.e., $\boldsymbol{x}^0 = \boldsymbol{0}$) and then ponder the rebalancing problem (i.e., $\boldsymbol{x}^0 \neq \boldsymbol{0}$).

2.3.1 Initial investment problem

Following Konno and Yamamoto [16], we first assume that one has no current portfolio, i.e., $\boldsymbol{x}^0 = \boldsymbol{0}$. For all assets $i = 1, 2, ..., I$, let us introduce discrete points $0 = X_{i0} < X_{i1} < \cdots < X_{iL} = X_i^{\max}$ and decision variables $e_{i\ell}$, $\ell = 0, 1, ..., L$ corresponding to the internal division ratio. Then, as shown in Figure 3, the nonconvex transaction cost function $\mathcal{C}_i$ can be approximated with a piecewise linear function:

$$\mathcal{C}_i(x_i) \approx \sum_{\ell=0}^{L} e_{i\ell} \mathcal{C}_i(X_{i\ell}), \ \forall i = 1, 2, ..., I$$

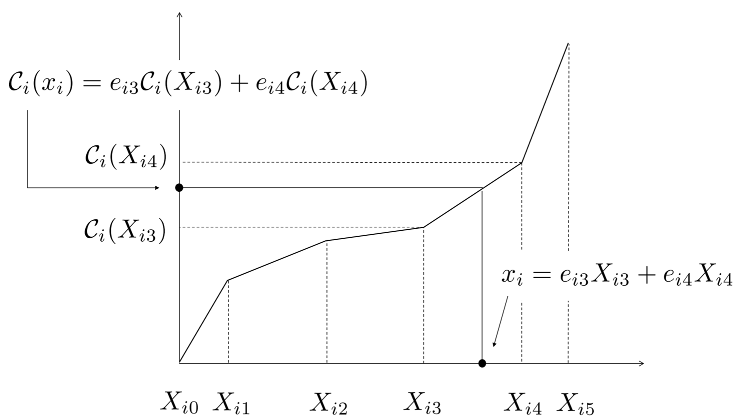


Figure 3: Piecewise linear transaction cost function

subject to the following constraints:

$$\left|\begin{array}{l} x_i = \sum_{\ell=0}^{L} e_{i\ell} X_{i\ell},\ \forall i = 1,2,...,I \\ \sum_{\ell=0}^{L} e_{i\ell} = 1,\ \forall i = 1,2,...,I \\ e_{i\ell} \geq 0,\ \forall i = 1,2,...,I,\ \forall \ell = 0,1,...,L \\ \{e_{i0} \preceq e_{i1} \preceq \cdots \preceq e_{iL}\}_2,\ \forall i = 1,2,...,I, \end{array}\right.$$

where $\{e_{i0} \preceq e_{i1} \preceq \cdots \preceq e_{iL}\}_2$ is a special ordered set type two (SOS2) constraint (see [6]). The SOS2 constraint implies that at most two consecutive elements of $e_{i\ell},\ \ell = 0,1,...,L$ can have nonzero values.

Considering that $e_{i\ell} \geq 0,\ \forall i = 1,2,...,I,\ \forall \ell = 0,1,...,L$, the above SOS2 constraints can be replaced with

$$\left|\begin{array}{l} \sum_{\ell=1}^{L} y_{i\ell} = 1,\ \forall i = 1,2,...,I \\ e_{i0} \leq y_{i1},\ \forall i = 1,2,...,I \\ e_{i\ell} \leq y_{i\ell} + y_{i\ell+1},\ \forall i = 1,2,...,I,\ \forall \ell = 1,2,...,L-1 \\ e_{iL} \leq y_{iL},\ \forall i = 1,2,...,I \\ y_{i\ell} \in \{0,1\},\ \forall i = 1,2,...,I,\ \forall \ell = 1,2,...,L. \end{array}\right. \tag{3}$$

Here, the decision variables are $y_{i\ell},\ i = 1,2,...,I,\ \ell = 1,2,...,L$. For instance, if $y_{i4} = 1$, only e_{i3} and e_{i4} can be nonzero, and $e_{i\ell}$ has to be zero for all $\ell \neq 3,4$. The SOS2 constraint is useful for making piecewise linear approximations of nonlinear functions, and hence, this constraint is supported by standard MIP solvers. We will utilize such an SOS2 implementation to speed up the search procedure in the branch and bound algorithm, whereas konno and Yamamoto [16] uses the constraints (3).

We shall consider a mean-CVaR model, i.e., one that minimizes the weighted sum of the measures of profitability (expected net return) and risk (CVaR). We are now in a position to formulate a mean-CVaR portfolio optimization problem with nonconvex transaction costs:

$$\left|\begin{array}{lll}\underset{a,\, e_{i\ell},\, u,\, x_i}{\text{minimize}} & (1-\lambda)u - \dfrac{\lambda}{S}\displaystyle\sum_{s=1}^{S}\sum_{i=1}^{I}\left(R_i^{(s)}x_i - \sum_{\ell=0}^{L} e_{i\ell}\mathcal{C}_i(X_{i\ell})\right) & \cdots(4.\,\mathrm{a}) \\ \text{subject to} & u \geq a + \dfrac{1}{(1-\beta)S}\displaystyle\sum_{s=1}^{S}\left[-\sum_{i=1}^{I}\left(R_i^{(s)}x_i - \sum_{\ell=0}^{L} e_{i\ell}\mathcal{C}_i(X_{i\ell})\right) - a\right]_+ & \cdots(4.\,\mathrm{b}) \\ & x_i = \displaystyle\sum_{\ell=0}^{L} e_{i\ell}X_{i\ell},\ \forall i = 1,2,...,I & \cdots(4.\,\mathrm{c}) \\ & \displaystyle\sum_{\ell=0}^{L} e_{i\ell} = 1,\ \forall i = 1,2,...,I & \cdots(4.\,\mathrm{d}) \\ & e_{i\ell} \geq 0,\ \forall i = 1,2,...,I,\ \forall \ell = 0,1,...,L & \cdots(4.\,\mathrm{e}) \\ & \{e_{i0} \preceq e_{i1} \preceq \cdots \preceq e_{iL}\}_2,\ \forall i = 1,2,...,I & \cdots(4.\,\mathrm{f}) \\ & \displaystyle\sum_{i=1}^{I} x_i = 1;\ X_i^{\min} \leq x_i \leq X_i^{\max},\ \forall i = 1,2,...,I, & \cdots(4.\,\mathrm{g}) \end{array}\right. \tag{4}$$

where $\lambda \in (0,1)$ is the trade-off parameter between profitability and risk. Here, for simplicity of presentation, an auxiliary decision variable u is introduced for CVaR. We refer to problem (4) as an initial investment problem.

2.3.2 Rebalancing problem

We can extend the MILP formulation [16] so that it can deal with the case where one has a current portfolio, i.e., $\boldsymbol{x}^0 \neq \mathbf{0}$. Here, it can be assumed without loss of generality that the transaction cost function, $\mathcal{C}_i$, can be decomposed into two nondecreasing functions, $\mathcal{C}_i^+$ and $\mathcal{C}_i^-$:

$$\mathcal{C}_i(x_i - x_i^0) = \mathcal{C}_i^+(x_i^+) + \mathcal{C}_i^-(x_i^-),\ \forall i = 1,2,...,I,$$

where $x_i^+ = |x_i - x_i^0|$ and $x_i^- = |x_i^0 - x_i|$ are the purchases and sales amounts of asset i, respectively. Since x_i^+ and x_i^- can also be represented as

$$x_i - x_i^0 = x_i^+ - x_i^-,\ x_i^+ \geq 0,\ x_i^- \geq 0,\ x_i^+x_i^- = 0,\ \forall i = 1,2,...,I, \tag{5}$$

the corresponding mean-CVaR portfolio optimization problem can be posed as

$$\left|\begin{array}{ll}\underset{a,\, u,\, x_i,\, x_i^+,\, x_i^-}{\text{minimize}} & (1-\lambda)u - \dfrac{\lambda}{S}\displaystyle\sum_{s=1}^{S}\sum_{i=1}^{I}\left(R_i^{(s)}x_i - \mathcal{C}_i^+(x_i^+) - \mathcal{C}_i^-(x_i^-)\right) \\ \text{subject to} & u \geq a + \dfrac{1}{(1-\beta)S}\displaystyle\sum_{s=1}^{S}\left[-\sum_{i=1}^{I}\left(R_i^{(s)}x_i - \mathcal{C}_i^+(x_i^+) - \mathcal{C}_i^-(x_i^-)\right) - a\right]_+ \\ & x_i - x_i^0 = x_i^+ - x_i^-,\ x_i^+ \geq 0,\ x_i^- \geq 0,\ x_i^+x_i^- = 0,\ \forall i = 1,2,...,I \\ & \displaystyle\sum_{i=1}^{I} x_i = 1;\ X_i^{\min} \leq x_i \leq X_i^{\max},\ \forall i = 1,2,...,I. \end{array}\right. \tag{6}$$

We show that the complementary conditions, $x_i^+x_i^- = 0,\ \forall i = 1,2,...,I$, can be eliminated from the above problem.

Proposition 2.1 *Suppose that* $(\hat{a}, \hat{u}, \hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}^+, \hat{\boldsymbol{x}}^-)$ *is an optimal solution to problem (6) without the complementary conditions. Define* $\dot{x}_i^+ := [\hat{x}_i - x_i^0]_+$ *and* $\dot{x}_i^- := [x_i^0 - \hat{x}_i]_+$ *for* $i = 1, 2, ..., I$*. Then* $(\hat{a}, \hat{u}, \hat{\boldsymbol{x}}, \dot{\boldsymbol{x}}^+, \dot{\boldsymbol{x}}^-)$ *is an optimal solution to problem (6).*

Proof. First, we show that $(\hat{a}, \hat{u}, \hat{\boldsymbol{x}}, \dot{\boldsymbol{x}}^+, \dot{\boldsymbol{x}}^-)$ is a feasible solution to problem (6). It follows from the definition that

$$\hat{x}_i - x_i^0 = \dot{x}_i^+ - \dot{x}_i^-,\ \dot{x}_i^+ \geq 0,\ \dot{x}_i^- \geq 0,\ \dot{x}_i^+ \dot{x}_i^- = 0,\ \forall i = 1, 2, ..., I.$$

In addition, since $0 \leq \hat{x}_i^+,\ \hat{x}_i - x_i^0 = \hat{x}_i^+ - \hat{x}_i^- \leq \hat{x}_i^+$ and $0 \leq \hat{x}_i^-,\ x_i^0 - \hat{x}_i = \hat{x}_i^- - \hat{x}_i^+ \leq \hat{x}_i^-$, we have

$$\dot{x}_i^+ = \max\{\hat{x}_i - x_i^0, 0\} \leq \hat{x}_i^+,\ \dot{x}_i^- = \max\{x_i^0 - \hat{x}_i, 0\} \leq \hat{x}_i^-,\ \forall i = 1, 2, ..., I.$$

Considering that $\mathcal{C}_i^+$ and $\mathcal{C}_i^-$ are nondecreasing functions, we can see that

$$\mathcal{C}_i^+(\dot{x}_i^+) + \mathcal{C}_i^-(\dot{x}_i^-) \leq \mathcal{C}_i^+(\hat{x}_i^+) + \mathcal{C}_i^-(\hat{x}_i^-),\ \forall i = 1, 2, ..., I. \tag{7}$$

Thus, $(\hat{a}, \hat{u}, \hat{\boldsymbol{x}}, \dot{\boldsymbol{x}}^+, \dot{\boldsymbol{x}}^-)$ satisfies all the constraints of problem (6).

Now we show that $(\hat{a}, \hat{u}, \hat{\boldsymbol{x}}, \dot{\boldsymbol{x}}^+, \dot{\boldsymbol{x}}^-)$ is an optimal solution to problem (6). It follows from (7) that the objective function value of $(\hat{a}, \hat{u}, \hat{\boldsymbol{x}}, \dot{\boldsymbol{x}}^+, \dot{\boldsymbol{x}}^-)$ is not greater than that of $(\hat{a}, \hat{u}, \hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}^+, \hat{\boldsymbol{x}}^-)$. Recall that $(\hat{a}, \hat{u}, \hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}^+, \hat{\boldsymbol{x}}^-)$ is an optimal solution to a relaxed problem, i.e., problem (6) without complementary conditions. This completes the proof. ■

Proposition 2.1 states that by solving the problem without the complementary conditions, $x_i^+ x_i^- = 0,\ \forall i = 1, 2, ..., I$, we can easily obtain an optimal solution to the original problem. For this reason, we focus on solving the problem without the complementary conditions.

Remark 2.1 When $\mathcal{C}_i^+$ and $\mathcal{C}_i^-$ are strictly increasing functions, $(\hat{a}, \hat{u}, \hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}^+, \hat{\boldsymbol{x}}^-)$ satisfies the complementary conditions by itself. This is because if i exists such that $\hat{x}_i^+ \hat{x}_i^- \neq 0$, the objective function value of $(\hat{a}, \hat{u}, \hat{\boldsymbol{x}}, \dot{\boldsymbol{x}}^+, \dot{\boldsymbol{x}}^-)$ is smaller than that of $(\hat{a}, \hat{u}, \hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}^+, \hat{\boldsymbol{x}}^-)$, which contradicts the optimality of $(\hat{a}, \hat{u}, \hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}^+, \hat{\boldsymbol{x}}^-)$.

We can rewrite problem (6) by making piecewise linear approximations of $\mathcal{C}_i^+$ and $\mathcal{C}_i^-$:

$$
\left|
\begin{array}{ll}
\underset{\substack{a, e_{i\ell}^+, e_{i\ell}^-, u,\\ x_i, x_i^+, x_i^-}}{\text{minimize}} & (1-\lambda)u - \dfrac{\lambda}{S}\displaystyle\sum_{s=1}^{S}\sum_{i=1}^{I}\left(R_i^{(s)}x_i - \sum_{\ell=0}^{L}\left(e_{i\ell}^+\mathcal{C}_i^+(X_{i\ell}^+) + e_{i\ell}^-\mathcal{C}_i^-(X_{i\ell}^-)\right)\right)\\
\text{subject to} & u \geq a + \dfrac{1}{(1-\beta)S}\displaystyle\sum_{s=1}^{S}\left[-\sum_{i=1}^{I}\left(R_i^{(s)}x_i - \sum_{\ell=0}^{L}\left(e_{i\ell}^+\mathcal{C}_i^+(X_{i\ell}^+) + e_{i\ell}^-\mathcal{C}_i^-(X_{i\ell}^-)\right)\right) - a\right]_+\\
& x_i^+ = \displaystyle\sum_{\ell=0}^{L} e_{i\ell}^+ X_{i\ell}^+,\ x_i^- = \sum_{\ell=0}^{L} e_{i\ell}^- X_{i\ell}^-,\ \forall i = 1,2,...,I\\
& \displaystyle\sum_{\ell=0}^{L} e_{i\ell}^+ = 1,\ \sum_{\ell=0}^{L} e_{i\ell}^- = 1,\ \forall i = 1,2,...,I\\
& e_{i\ell}^+ \geq 0,\ e_{i\ell}^- \geq 0,\ \forall i = 1,2,...,I,\ \forall \ell = 0,1,...,L\\
& \{e_{i0}^+ \preceq e_{i1}^+ \preceq \cdots \preceq e_{iL}^+\}_2,\ \{e_{i0}^- \preceq e_{i1}^- \preceq \cdots \preceq e_{iL}^-\}_2,\ \forall i = 1,2,...,I\\
& x_i - x_i^0 = x_i^+ - x_i^-,\ x_i^+ \geq 0,\ x_i^- \geq 0,\ \forall i = 1,2,...,I\\
& \displaystyle\sum_{i=1}^{I} x_i = 1;\ X_i^{\min} \leq x_i \leq X_i^{\max},\ \forall i = 1,2,...,I.
\end{array}
\right.
\tag{8}
$$

In what follows, we show that the problem size can be reduced when the purchases cost function and the sales cost function are the same, i.e.,

$$\mathcal{C}_i^+(x) = \mathcal{C}_i^-(x),\ \forall x \geq 0,\ \forall i = 1,2,...,I.$$

In fact, many securities companies set the purchases and sales costs of assets equal. For this reason, if the transaction costs are limited to brokerage commissions, the above assumption coincides with reality.

Furthermore, it follows from (5) that

$$\mathcal{C}_i(x_i - x_i^0) = \mathcal{C}_i(x_i^+ + x_i^-),\ \forall i = 1,2,...,I.$$

Thus, by utilizing a piecewise linear transaction cost function,

$$\mathcal{C}_i(x_i^+ + x_i^-) \approx \sum_{\ell=0}^{L} e_{i\ell}\mathcal{C}_i(X_{i\ell}),\ \forall i = 1,2,...,I,$$

problem (8) reduces to

$$
\left|\begin{array}{ll}
\underset{\substack{a,\, e_{i\ell},\, u,\\ x_i,\, x_i^+,\, x_i^-}}{\text{minimize}} & (1-\lambda)u - \dfrac{\lambda}{S}\displaystyle\sum_{s=1}^{S}\sum_{i=1}^{I}\left(R_i^{(s)}x_i - \sum_{\ell=0}^{L} e_{i\ell}\mathcal{C}_i(X_{i\ell})\right) \\
\text{subject to} & u \ge a + \dfrac{1}{(1-\beta)S}\displaystyle\sum_{s=1}^{S}\left[-\sum_{i=1}^{I}\left(R_i^{(s)}x_i - \sum_{\ell=0}^{L} e_{i\ell}\mathcal{C}_i(X_{i\ell})\right) - a\right]_+ \\
& x_i^+ + x_i^- = \displaystyle\sum_{\ell=0}^{L} e_{i\ell}X_{i\ell},\ \forall i = 1,2,...,I \\
& \displaystyle\sum_{\ell=0}^{L} e_{i\ell} = 1,\ \forall i = 1,2,...,I \\
& e_{i\ell} \ge 0,\ \forall i = 1,2,...,I,\ \forall \ell = 0,1,...,L \\
& \{e_{i0} \preceq e_{i1} \preceq \cdots \preceq e_{iL}\}_2,\ \forall i = 1,2,...,I \\
& x_i - x_i^0 = x_i^+ - x_i^-,\ x_i^+ \ge 0,\ x_i^- \ge 0,\ \forall i = 1,2,...,I \\
& \displaystyle\sum_{i=1}^{I} x_i = 1;\ X_i^{\min} \le x_i \le X_i^{\max},\ \forall i = 1,2,...,I.
\end{array}\right. \tag{9}
$$

Note that the complementary conditions have already been eliminated in the above problem. We can prove the validity of this elimination similarly to the proof of Proposition 2.1. We refer to problems (8) and (9) as rebalancing problems.

2.4 Lifting and cutting plane representation

To solve problems (4), (8) and (9) with MILP solvers, we need to transform the nonlinear and nondifferentiable CVaR constraint, e.g., (4. b), into a tractable one. The most common method is the lifting representation [10, 23], which converts the constraint (4. b) into

$$
\left|\begin{array}{l}
u \ge a + \dfrac{1}{(1-\beta)S}\displaystyle\sum_{s=1}^{S} w_s \\
w_s \ge 0,\ w_s \ge -\displaystyle\sum_{i=1}^{I}\left(R_i^{(s)}x_i - \sum_{\ell=0}^{L} e_{i\ell}\mathcal{C}_i(X_{i\ell})\right) - a,\ \forall s = 1,2,...,S
\end{array}\right. \tag{10}
$$

with auxiliary decision variables $w_s,\ s = 1,2,...,S$. Although these constraints are linear, a large number of scenarios are required for calculating CVaR accurately (see, e.g., [12, 26]). As a result, MILP problems with the constraints (10) are difficult to handle.

To overcome this drawback, subgradient-based cutting plane algorithms are employed in [2, 17]. This sort of algorithm gradually approximates the CVaR constraint through the use of linear constraints. Specifically, Künzi-Bay and Mayer [17] prove that the CVaR constraint can equivalently be rewritten in the following cutting plane representation [10, 17]:

$$
u \ge \mathcal{F}(a, \boldsymbol{e}, \boldsymbol{x}; \mathcal{J}),\ \forall \mathcal{J} \subseteq \{1,2,...,S\}, \tag{11}
$$

where $\boldsymbol{e} := (e_{i\ell};\ i = 1,2,...,I,\ \ell = 0,1,...,L)$, and

$$
\mathcal{F}(a, \boldsymbol{e}, \boldsymbol{x}; \mathcal{J}) := a + \frac{1}{(1-\beta)S}\sum_{s\in\mathcal{J}}\left(-\sum_{i=1}^{I}\left(R_i^{(s)}x_i - \sum_{\ell=0}^{L} e_{i\ell}\mathcal{C}_i(X_{i\ell})\right) - a\right).
$$

The cutting plane representation (11) is a set of linear constraints, and the number of constraints is equal to the number of subsets of the scenarios, i.e., 2^S. In the cutting plane algorithm, the necessary constraints are iteratively chosen from (11) and appended to the problem.

3 Cutting Plane Algorithm

Künzi-Bay and Mayer [17] developed a cutting plane algorithm for a portfolio optimization problem with no transaction costs. In this section, we describe a specialized cutting plane algorithm for efficiently solving problem (4) in the presence of nonconvex transaction costs. Note that our algorithm can be applied to problems (8) and (9).

3.1 Basic cutting plane algorithm

A cutting plane algorithm first solves problem (4) without the CVaR constraint (4. b). A feasible set for this relaxed problem is defined as follows:

$$\mathcal{Z}_1 := \{(a, \boldsymbol{e}, u, \boldsymbol{x}) \mid (4.\,\mathrm{c}), (4.\,\mathrm{d}), (4.\,\mathrm{e}), (4.\,\mathrm{f}), (4.\,\mathrm{g}),\ u \geq U^{\min}\}, \tag{12}$$

where $U^{\min}$ is a sufficiently small constant for preventing the objective function (4. a) from going to $-\infty$.

Our basic strategy involves repeatedly solving the relaxed problems and iteratively approximating the CVaR constraint (4. b) by using a portion of the cutting plane representation (11). Let UB_k and LB_k be respectively the best upper and lower bounds of the optimal objective function value of problem (4) at iteration k. Our algorithm terminates when the optimality gap $\mathrm{UB}_k - \mathrm{LB}_k$ is sufficiently small. LB_k is the optimal objective function value of the relaxed problem, whereas UB_k needs to be calculated by converting the solution $(\bar{a}, \bar{\boldsymbol{e}}, \bar{u}, \bar{\boldsymbol{x}})$ into a feasible one that satisfies the removed constraint (4. b).

An easy way to do this is to set

$$u' := \bar{a} + \frac{1}{(1-\beta)S} \sum_{s=1}^{S} \left[-\sum_{i=1}^{I} \left(R_i^{(s)} \bar{x}_i - \sum_{\ell=0}^{L} \bar{e}_{i\ell} \mathcal{C}_i(X_{i\ell}) \right) - \bar{a} \right]_+ . \tag{13}$$

Then $(\bar{a}, \bar{\boldsymbol{e}}, u', \bar{\boldsymbol{x}})$ satisfies the CVaR constraint (4. b), and hence, it is a feasible solution to problem (4). Meanwhile, in order to estimate a better upper bound, we implement a sorting-based procedure[1] by following Proposition 8 in [24]. This procedure is described in Algorithm 1 (UBE).

If the optimality gap is not sufficiently small, we add a constraint selected from the cutting plane representation (11) to separate the solution $(\bar{a}, \bar{\boldsymbol{e}}, \bar{u}, \bar{\boldsymbol{x}})$ from the feasible set. If $\mathrm{UB}_k \neq \mathrm{LB}_k$,

[1] This procedure aims at minimizing the right side of (13) with respect to $\bar{a} \in \mathbb{R}$, and its effectiveness was confirmed through preliminary computational experiments.

Algorithm 1 (UBE): Upper Bound Estimation for Solving Problem (4)

Step 1. (Feasibility Check) If a solution $(\bar{a}, \bar{\boldsymbol{e}}, \bar{u}, \bar{\boldsymbol{x}})$ satisfies the CVaR constraint (4. b), then $\mathrm{UB} := \mathrm{LB}_k$, $a' := \bar{a}$, $u' := \bar{u}$ and terminates the algorithm.

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

$$N_s := -\sum_{i=1}^{I} \left(R_i^{(s)} \bar{x}_i - \sum_{\ell=0}^{L} \bar{e}_{i\ell} \mathcal{C}_i(X_{i\ell}) \right), \ \forall s = 1, 2, ..., S.$$

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

Step 3. (VaR/CVaR Calculation) Set an integer $\tau := \min\{n \in \mathbb{Z} \mid \beta S \leq n\}$. Then, set VaR as $a' := N_{\sigma(\tau)}$ and CVaR as

$$u' := \frac{1}{(1-\beta)S} \left(\left(\frac{\tau}{S} - \beta \right) N_{\sigma(\tau)} + \sum_{s=\tau+1}^{S} N_{\sigma(s)} \right).$$

Step 4. (Upper Bound Estimation) Calculate an upper bound by substituting a feasible solution $(a', \bar{\boldsymbol{e}}, u', \bar{\boldsymbol{x}})$ into the objective function (4. a):

$$\mathrm{UB} := (1-\lambda)u' - \frac{\lambda}{S} \sum_{s=1}^{S} \sum_{i=1}^{I} \left(R_i^{(s)} \bar{x}_i - \sum_{\ell=0}^{L} \bar{e}_{i\ell} \mathcal{C}_i(X_{i\ell}) \right).$$

the solution $(\bar{a}, \bar{\boldsymbol{e}}, \bar{u}, \bar{\boldsymbol{x}})$ violates the CVaR constraint (4. b). Accordingly,

$$\begin{aligned} \bar{u} &< \bar{a} + \frac{1}{(1-\beta)S} \sum_{s=1}^{S} \left[-\sum_{i=1}^{I} \left(R_i^{(s)} \bar{x}_i - \sum_{\ell=0}^{L} \bar{e}_{i\ell} \mathcal{C}_i(X_{i\ell}) \right) - \bar{a} \right]_+ \\ &= \bar{a} + \frac{1}{(1-\beta)S} \sum_{s \in \mathcal{J}_k} \left(-\sum_{i=1}^{I} \left(R_i^{(s)} \bar{x}_i - \sum_{\ell=0}^{L} \bar{e}_{i\ell} \mathcal{C}_i(X_{i\ell}) \right) - \bar{a} \right) \\ &= \mathcal{F}(\bar{a}, \bar{\boldsymbol{e}}, \bar{\boldsymbol{x}}; \mathcal{J}_k), \end{aligned}$$

where

$$\mathcal{J}_k := \left\{ s \;\middle|\; s = 1, 2, ..., S, \ -\sum_{i=1}^{I} \left(R_i^{(s)} \bar{x}_i - \sum_{\ell=0}^{L} \bar{e}_{i\ell} \mathcal{C}_i(X_{i\ell}) \right) - \bar{a} > 0 \right\}. \tag{14}$$

Thus, the constraint $u \geq \mathcal{F}(a, \boldsymbol{e}, \boldsymbol{x}; \mathcal{J}_k)$ cuts off the solution $(\bar{a}, \bar{\boldsymbol{e}}, \bar{u}, \bar{\boldsymbol{x}})$ from the feasible set. Our basic cutting plane algorithm is summarized in Algorithm 2 (BCPA).

Our algorithm is basically a direct application of the subgradient-based cutting plane algorithm to problem (4), as in Ahmed [2], whereas the algorithm in [17] is based on the L-shaped decomposition method [25] for solving a two-stage stochastic linear programming problem. In addition to solving MILP problems, Algorithm 1 (BCPA) is slightly different from [2, 17] in that it improves upper bounds by means of a sorting-based procedure, Algorithm 1 (UBE).

Similarly to [2, 17], we can show the finite convergence of our algorithm.

Algorithm 2 (BCPA): Basic Cutting Plane Algorithm for Solving Problem (4)

Step 0. (Initialization) Let $\varepsilon \geq 0$ be a tolerance for optimality. Define $\mathcal{Z}_1$ as (12). Set $\mathrm{UB}_0 := \infty$ and $k \leftarrow 1$.

Step 1. (Relaxed MILP Problem) Solve the problem:

$$\begin{array}{|ll} \underset{a,\, \boldsymbol{e},\, u,\, \boldsymbol{x}}{\text{minimize}} & (1-\lambda)u - \dfrac{\lambda}{S}\displaystyle\sum_{s=1}^{S}\sum_{i=1}^{I}\left(R_i^{(s)}x_i - \sum_{\ell=0}^{L} e_{i\ell}\mathcal{C}_i(X_{i\ell})\right) \\ \text{subject to} & (a, \boldsymbol{e}, u, \boldsymbol{x}) \in \mathcal{Z}_k. \end{array} \tag{15}$$

Let $(\bar{a}, \bar{\boldsymbol{e}}, \bar{u}, \bar{\boldsymbol{x}})$ be an optimal solution, and LB_k be the corresponding objective function value.

Step 2. (Solution Update) Execute Algorithm 1 (UBE). If the obtained upper bound UB is less than UB_{k-1}, then $\mathrm{UB}_k := \mathrm{UB}$ and $(\hat{a}, \hat{\boldsymbol{e}}, \hat{u}, \hat{\boldsymbol{x}}) \leftarrow (a', \bar{\boldsymbol{e}}, u', \bar{\boldsymbol{x}})$. Otherwise, $\mathrm{UB}_k := \mathrm{UB}_{k-1}$.

Step 3. (Termination Condition) If $\mathrm{UB}_k - \mathrm{LB}_k < \varepsilon$, terminate the algorithm with the ε-optimal solution: $(\hat{a}, \hat{\boldsymbol{e}}, \hat{u}, \hat{\boldsymbol{x}})$.

Step 4. (Cut Generation) Set $\mathcal{J}_k$ as (14), and

$$\mathcal{Z}_{k+1} := \mathcal{Z}_k \cap \{(a, \boldsymbol{e}, u, \boldsymbol{x}) \mid u \geq \mathcal{F}(a, \boldsymbol{e}, \boldsymbol{x}; \mathcal{J}_k)\}.$$

Then, set $k \leftarrow k+1$, and return to Step 1.

Theorem 3.1 *For any $\varepsilon \geq 0$,* Algorithm 2 (BCPA) *terminates in a finite number of iterations.*

Proof. As explained above, a solution $(\bar{a}, \bar{\boldsymbol{e}}, \bar{u}, \bar{\boldsymbol{x}}) \in \mathcal{Z}_k$ is separated from the set $\mathcal{Z}_k$ in every iteration; therefore, the same cut is never appended twice. As a result, after 2^S iterations, all the constraints of the cutting plane representation (11) are imposed on problem (15). This problem is equivalent to the original problem (4), and the algorithm terminates because $\mathrm{UB}_k = \mathrm{LB}_k$. ■

3.2 Two-phase cutting plane algorithm

Algorithm 2 (BCPA) needs to repeatedly solve the relaxed MILP problems (15). This is very different from the algorithm in [17], in which transaction costs are not taken into consideration and relaxed LP problems are repeatedly solved. Since solving many MILP problems takes a substantial amount of time, we will reduce the number of problems to be solved. To accomplish this, we will select and append effective cuts from the cutting plane representation (11) before starting the MILP-based cutting plane algorithm.

Let us consider problem (4) without SOS2 constraints (4. f):

$$\begin{array}{|ll} \underset{a,\, e_{i\ell},\, u,\, x_i}{\text{minimize}} & (1-\lambda)u - \dfrac{\lambda}{S}\displaystyle\sum_{s=1}^{S}\sum_{i=1}^{I}\left(R_i^{(s)}x_i - \sum_{\ell=0}^{L} e_{i\ell}\mathcal{C}_i(X_{i\ell})\right) \\ \text{subject to} & \text{Constraints (4. b), (4. c), (4. d), (4. e), (4. g).} \end{array} \tag{16}$$

In this problem, every combination of $e_{i\ell}$ can be nonzero; consequently, as shown in [28, 29], the transaction cost $\sum_{\ell=0}^{L} e_{i\ell}\mathcal{C}_i(X_{i\ell})$ is a convex underestimator of the piecewise linear transaction cost function (see Figure 4). Problem (16) becomes an efficiently solvable LP problem once the CVaR constraint (4. b) has been removed. A cutting plane algorithm for solving problem (16) enjoys this computational benefit. After solving problem (16) by means of our cutting plane algorithm, we restore SOS2 constraints (4. f) and restart the algorithm. Now a number of effective constraints are already imposed, and they greatly decrease the number of MILP problems to be solved. This two-phase cutting plane algorithm is presented as Algorithm 3 (2PCPA).

Algorithm 3 (2PCPA): Two-Phase Cutting Plane Algorithm for Solving Problem (4)

Step 0. (Initialization) Let $\varepsilon \geq 0$ be a tolerance for optimality. Define $\mathcal{Z}_1$ as

$$\mathcal{Z}_1 := \{(a, \boldsymbol{e}, u, \boldsymbol{x}) \mid (4.\,\mathrm{c}), (4.\,\mathrm{d}), (4.\,\mathrm{e}), (4.\,\mathrm{g}),\ u \geq U^{\min}\}.$$

Set $\mathrm{UB}_0 := \infty$ and $k \leftarrow 1$.

Step 1. (Phase One: LP-Based CPA) Start Algorithm 2 (BCPA) from Step 1.

Step 2. (Restoration of SOS2 Constraints) Set

$$\mathcal{Z}_{k+1} := \mathcal{Z}_k \cap \{(a, \boldsymbol{e}, u, \boldsymbol{x}) \mid (4.\,f)\},$$

$\mathrm{UB}_k := \infty$, and $k \leftarrow k + 1$.

Step 3. (Phase Two: MILP-Based CPA) Restart Algorithm 2 (BCPA) from Step 1.

We can prove the finite convergence of this algorithm similarly to Algorithm 2 (BCPA).

Corollary 3.1 *For any $\varepsilon \geq 0$, Algorithm 3 (2PCPA) terminates in a finite number of iterations.*

Proof. This proof is similar to the one of Theorem 3.1. ■

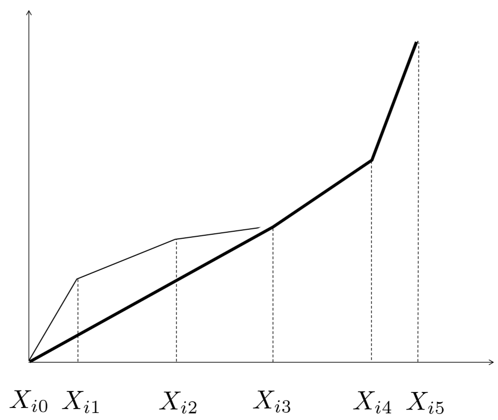


Figure 4: Convex underestimator of piecewise linear transaction cost function

4 Computational Experiments

The computational results reported in this section assess the efficiency of our algorithms in solving mean-CVaR portfolio optimization problems with nonconvex transaction costs.

4.1 Problem setting

We supposed that the total investment amount is five million yen. The lower and upper limits, $X_i^{\min}$ and $X_i^{\max}$, of the investment proportion were set to 0 and 0.2, respectively, for each $i = 1, 2, ..., I$. We also used the transaction cost function provided by a leading Japanese securities company for all assets (see Figure 5). Note that a fixed cost is imposed on a transaction of one yen. In addition, we did not consider the market impact cost, and hence, the purchases cost and sales cost were equal. For this reason, we solved the reduced problem (9) as a rebalancing problem.

Scenario set $\{\boldsymbol{R}^{(s)} = (R_1^{(s)}, R_2^{(s)}, ..., R_I^{(s)})^\top \mid s = 1, 2, ..., S\}$ was generated as follows:

$$(\ln(1 + \tilde{r}_1), \ln(1 + \tilde{r}_2), ..., \ln(1 + \tilde{r}_I))^\top \sim \mathbf{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}),$$

where $\mathbf{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a multivariate normal distribution with mean vector $\boldsymbol{\mu} \in \mathbb{R}^I$ and variance-covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{I \times I}$. We collected monthly stock prices of the 225 Japanese companies composing the Nikkei 225 from 2003 to 2012 from Yahoo finance Japan[2]. The values of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ were estimated with these historical data. Five scenario sets were generated for each (I, S), and the average performance on these sets was evaluated.

The trade-off parameter, λ, was set to 0.5. We chose the number of assets, $I \in \{20, 100, 200\}$, which corresponds to the I largest companies by market value in the Nikkei 225. We set the number of scenarios, $S \in \{1{,}000, 10{,}000, 100{,}000\}$. In our cutting plane algorithms, the tolerance for optimality, ε, was set to 10^{-4}, and $U^{\min}$ was set to $\min\{-R_i^{(s)} \mid i = 1, 2, ..., I,\ s = 1, 2, ..., S\}$.

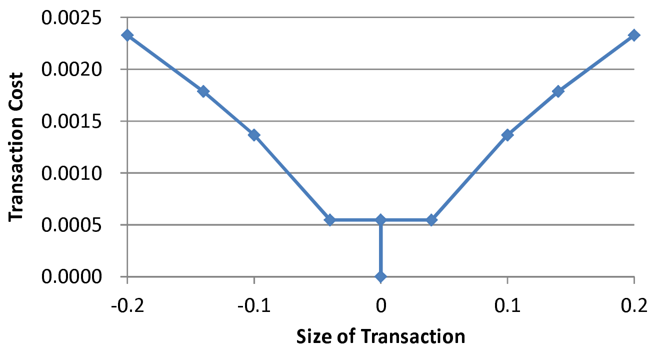


Figure 5: Actual transaction cost function

[2]http://finance.yahoo.co.jp

The computational experiments compared the performance of our cutting plane algorithms with Lifting Representation and Problem Reduction. For problem (4), Lifting Representation directly solves the following MILP problem:

$$\left|\begin{array}{ll} \underset{a,\,\boldsymbol{e},\,u,\,\boldsymbol{x}}{\text{minimize}} & (1-\lambda)u - \dfrac{\lambda}{S}\displaystyle\sum_{s=1}^{S}\sum_{i=1}^{I}\left(R_i^{(s)}x_i - \sum_{\ell=0}^{L} e_{i\ell}\mathcal{C}_i(X_{i\ell})\right) \\ \text{subject to} & \text{Constraints } (10), (4.\,\text{c}), (4.\,\text{d}), (4.\,\text{e}), (4.\,\text{f}), (4.\,\text{g}), \end{array}\right. \tag{17}$$

where the lifting representation (10) is imposed instead of the CVaR constraint (4. b).

Problem Reduction is a heuristic optimization algorithm based on continuous relaxation, and similar algorithms were implemented in previous studies [4, 28, 29]. Specifically, for the MILP problem (17), this algorithm first solves the relaxed LP problem without SOS2 constraints (4. f),

$$\left|\begin{array}{ll} \underset{a,\,\boldsymbol{e},\,u,\,\boldsymbol{x}}{\text{minimize}} & (1-\lambda)u - \dfrac{\lambda}{S}\displaystyle\sum_{s=1}^{S}\sum_{i=1}^{I}\left(R_i^{(s)}x_i - \sum_{\ell=0}^{L} e_{i\ell}\mathcal{C}_i(X_{i\ell})\right) \\ \text{subject to} & \text{Constraints } (10), (4.\,\text{c}), (4.\,\text{d}), (4.\,\text{e}), (4.\,\text{g}) \end{array}\right.$$

and obtains a solution $(\bar{a}, \bar{\boldsymbol{e}}, \bar{u}, \bar{\boldsymbol{x}})$. Next, we limit the investable assets to ones satisfying $\bar{x}_i \neq x_i^0$ and solve the following MILP problem with SOS2 constraints (4. f):

$$\left|\begin{array}{ll} \underset{a,\,\boldsymbol{e},\,u,\,\boldsymbol{x}}{\text{minimize}} & (1-\lambda)u - \dfrac{\lambda}{S}\displaystyle\sum_{s=1}^{S}\sum_{i=1}^{I}\left(R_i^{(s)}x_i - \sum_{\ell=0}^{L} e_{i\ell}\mathcal{C}_i(X_{i\ell})\right) \\ \text{subject to} & \text{Constraints } (10), (4.\,\text{c}), (4.\,\text{d}), (4.\,\text{e}), (4.\,\text{f}), (4.\,\text{g}) \\ & x_i = x_i^0, \ \forall i \in \mathcal{I}, \end{array}\right.$$

where $\mathcal{I} := \{i \mid i = 1, 2, ..., I, \ \bar{x}_i = x_i^0\}$.

Tables 1 and 2 use the following abbreviations: The row labeled "#Scenarios" is the number of scenarios, S. The row labeled "Time [sec.]" is the computation time in seconds. Note that a computation was terminated if it took more than 1,800 seconds. "OT(*)" indicates terminated computations, where "*" is the number of terminations out of five due to this time limit. Similarly, "OM(*)" indicates terminations due to memory shortages, where "*" is the number of memory shortages out of five. For Lifting Representation and Problem Reduction, the row labeled "Obj.Val." is the obtained objective function value. For Algorithm 2 (BCPA) and Algorithm 3 (2PCPA), "Obj.Val.LB" and "Obj.Val.UB" are respectively the obtained lower and upper bounds of the optimal objective function value. In addition, "Opt.Gap [%]" is the optimality gap, i.e., 100×(Obj.Val.UB − Obj.Val.LB)/Obj.Val.LB. "Time [sec.]", "Obj.Val.", "Obj.Val.LB" and "Obj.Val.UB" are average values for the five scenario sets.

All computations were conducted on a Linux computer with an Intel Xeon CPU (2.80GHz) and 12GB memory. Gurobi Optimizer[3] 4.5 was used to solve the (MI)LP problems.

4.2 Computational results for the initial investment problem

Table 1 shows the results of solving the initial investment problem (4). We begin by looking

[3] `http://www.gurobi.com/`

Table 1: Results of solving the initial investment problem (4)

$I = 20$	Lifting Representation			Problem Reduction		
#Scenarios	1,000	10,000	100,000	1,000	10,000	100,000
Time [sec.]	3.2	50.4	OT(5)	1.9	28.7	OT(4)
Obj.Val.	0.0466	0.0483	0.0481	0.0466	0.0483	0.0480
	Algorithm 2 (BCPA)			Algorithm 3 (2PCPA)		
#Scenarios	1,000	10,000	100,000	1,000	10,000	100,000
Time [sec.]	26.6	35.4	22.8	12.7	4.9	5.2
Obj.Val.UB	0.0467	0.0483	0.0480	0.0467	0.0483	0.0480
Obj.Val.LB	0.0466	0.0482	0.0479	0.0466	0.0482	0.0479
Opt.Gap [%]	0.18	0.19	0.19	0.19	0.20	0.20
$I = 100$	Lifting Representation			Problem Reduction		
#Scenarios	1,000	10,000	100,000	1,000	10,000	100,000
Time [sec.]	107.0	OT(4)	OM(5)	20.8	919.9	OM(5)
Obj.Val.	0.0359	0.0379	—	0.0359	0.0379	—
	Algorithm 2 (BCPA)			Algorithm 3 (2PCPA)		
#Scenarios	1,000	10,000	100,000	1,000	10,000	100,000
Time [sec.]	OT(5)	OT(5)	OT(5)	OT(5)	OT(5)	OT(5)
Obj.Val.UB	0.0360	0.0382	0.0381	0.0359	0.0380	0.0379
Obj.Val.LB	0.0354	0.0370	0.0368	0.0357	0.0375	0.0374
Opt.Gap [%]	1.90	3.18	3.38	0.60	1.33	1.37
$I = 200$	Lifting Representation			Problem Reduction		
#Scenarios	1,000	10,000	100,000	1,000	10,000	100,000
Time [sec.]	349.4	OM(5)	OM(5)	26.9	OT(1)	OM(5)
Obj.Val.	0.0369	—	—	0.0369	0.0378	—
	Algorithm 2 (BCPA)			Algorithm 3 (2PCPA)		
#Scenarios	1,000	10,000	100,000	1,000	10,000	100,000
Time [sec.]	OT(5)	OT(5)	OT(5)	OT(5)	OT(5)	OT(5)
Obj.Val.UB	0.0372	0.0381	0.0382	0.0371	0.0379	0.0380
Obj.Val.LB	0.0362	0.0367	0.0366	0.0367	0.0374	0.0373
Opt.Gap [%]	2.75	3.52	4.08	1.10	1.41	1.73

at the case of $I = 20$. Here, we can see that Lifting Representation and Problem Reduction took over 1,800 seconds to solve the problem with $S = 100{,}000$. Meanwhile, Algorithm 2 (BCPA) and Algorithm 3 (2PCPA) took less than one minute to solve it. Moreover, Algorithm 3 (2PCPA)'s computation time was sharply reduced compared with Algorithm 2 (BCPA). For instance, when $S = 100{,}000$, Algorithm 2 (BCPA) and Algorithm 3 (2PCPA) took 22.8 seconds and 5.2 seconds, respectively. In addition, the computation times of our cutting plane algorithms were nearly independent of the number of scenarios. These algorithms select and use only the necessary constraints from the cutting plane representation; consequently, they performed well regardless of the number of scenarios.

Turning now to the cases of $I = 100$ and 200, Algorithm 2 (BCPA) and Algorithm 3 (2PCPA) required over 1,800 seconds to solve the problems. For this reason, the solutions obtained by Lifting Representation or Problem Reduction were slightly better than those of the cutting plane algorithms. For instance, when $(I, S) = (200, 1{,}000)$, the objective function values of Lifting Representation and Problem Reduction were 0.0369, while those of Algorithm 2 (BCPA) and Algorithm 3 (2PCPA) were 0.0372 and 0.0371. However, the resultant optimality gap of Algorithm 3 (2PCPA) was always less than 2%, which is sufficiently small. This implies that Algorithm 3 (2PCPA) solved the problems with satisfactory accuracy in only 1,800 seconds.

4.3 Computational results for the rebalancing problem

Table 2 shows the results of solving the rebalancing problem (9). Here, the current portfolio $\boldsymbol{x}^0$ was constructed so that the investment proportions in the ten largest companies by market value were all 0.1. Table 2 reveals that our cutting plane algorithms have clear advantages over Lifting Representation and Problem Reduction. Specifically, they solved the problems with $I = 20$ in a few seconds. Moreover, they solved the problems with $I = 200$ within 1,800 seconds. On average, Algorithm 2 (BCPA) took two to three times longer than Algorithm 3 (2PCPA). Furthermore, our cutting plane algorithms solved the problems much faster than Lifting Representation and Problem Reduction did when the number of scenarios was 10,000 and 100,000. This is because the current portfolio $\boldsymbol{x}^0$ was relatively close to the optimal portfolio, and therefore, the cutting plane algorithm needed fewer iterations to arrive at an ε-optimal solution. For practical purposes, portfolios should be periodically rebalanced in view of the latest data on asset returns. In this case, it is unlikely that a portfolio would be significantly changed by rebalancing, and that means our cutting plane algorithms should be a practical way of solving the rebalancing problems.

4.4 Evaluation of the sampling error

Tables 1 and 2 indicate that our cutting plane algorithms were very effective especially when there were many scenarios (over 10,000). Here, we would like to emphasize that a large number of scenarios are required to estimate CVaR accurately.

Takeda and Kanamori [26] demonstrated that a huge number of scenarios are needed to calculate CVaR accurately via the scenario-based approximation. In view of their theoretical results, we show in Figure 6 the standard deviation of the objective function values, i.e.,

Table 2: Results of solving the rebalancing problem (9)

***I* = 20**	Lifting Representation			Problem Reduction		
#Scenarios	1,000	10,000	100,000	1,000	10,000	100,000
Time [sec.]	2.9	31.6	OM(3)	2.6	35.0	1228.3
Obj.Val.	0.0458	0.0470	—	0.0458	0.0470	0.0468
	Algorithm 2 (BCPA)			Algorithm 3 (2PCPA)		
#Scenarios	1,000	10,000	100,000	1,000	10,000	100,000
Time [sec.]	2.5	1.9	4.1	0.8	0.5	2.6
Obj.Val.UB	0.0458	0.0471	0.0468	0.0458	0.0471	0.0468
Obj.Val.LB	0.0458	0.0470	0.0467	0.0457	0.0470	0.0467
Opt.Gap [%]	0.14	0.19	0.18	0.16	0.17	0.18
***I* = 100**	Lifting Representation			Problem Reduction		
#Scenarios	1,000	10,000	100,000	1,000	10,000	100,000
Time [sec.]	74.0	621.6	OM(5)	48.3	605.7	OM(5)
Obj.Val.	0.0417	0.0427	—	0.0417	0.0427	—
	Algorithm 2 (BCPA)			Algorithm 3 (2PCPA)		
#Scenarios	1,000	10,000	100,000	1,000	10,000	100,000
Time [sec.]	283.0	93.6	120.8	115.2	46.4	55.6
Obj.Val.UB	0.0417	0.0427	0.0426	0.0417	0.0427	0.0426
Obj.Val.LB	0.0416	0.0426	0.0425	0.0416	0.0426	0.0425
Opt.Gap [%]	0.22	0.22	0.23	0.22	0.21	0.23
***I* = 200**	Lifting Representation			Problem Reduction		
#Scenarios	1,000	10,000	100,000	1,000	10,000	100,000
Time [sec.]	103.8	OM(1)	OM(5)	66.1	656.2	OM(5)
Obj.Val.	0.0421	—	—	0.0421	0.0428	—
	Algorithm 2 (BCPA)			Algorithm 3 (2PCPA)		
#Scenarios	1,000	10,000	100,000	1,000	10,000	100,000
Time [sec.]	435.1	157.9	250.5	139.4	54.9	83.0
Obj.Val.UB	0.0421	0.0428	0.0426	0.0421	0.0428	0.0426
Obj.Val.LB	0.0420	0.0427	0.0425	0.0420	0.0427	0.0425
Opt.Gap [%]	0.22	0.21	0.22	0.22	0.21	0.23

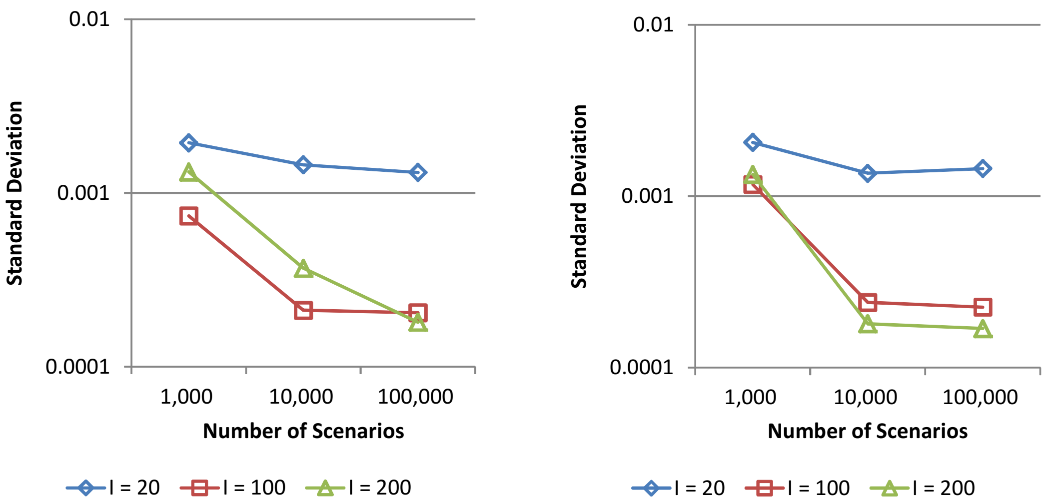
Obj.Val.UB obtained by Algorithm 3 (2PCPA) for five scenario sets. A small standard deviation implies that the sampling error in scenario-based CVaR is small. From Figure 6, we can see that the standard deviation becomes smaller as the number of scenarios increases. In other words, this figure suggests that by increasing the number of scenarios, we can lessen the effect of the sampling error on scenario-based CVaR. Figure 6 also indicates that it is desirable to use over 10,000 scenarios to solve the portfolio optimization problems in our problem setting.

5 Conclusions

This paper dealt with the scenario-based mean-CVaR portfolio optimization problem with non-convex transaction costs. Through the use of a lifting representation and piecewise linear approximation, this problem can be posed as an MILP problem with SOS2 constraints. This approach does not work on large-scale problems, however, so we developed a specialized cutting plane algorithm. Moreover, we devised a two-phase cutting plane algorithm with even higher computational efficiency.

The computational results indicated that our cutting plane algorithms were very effective at solving the problems with a large number of scenarios. Specifically, our algorithms attained a near-optimal solution to the initial investment problem in a reasonable amount of time. They had clear advantages over lifting representation and problem reduction heuristics in solving the rebalancing problem. Moreover, we would like to stress that our algorithms have a guarantee of global optimality, unlike heuristic optimization algorithms. We also demonstrated that using a large number of scenarios mitigated the adverse effect of the sampling error on scenario-based CVaR.

A future direction of study is to apply our cutting plane algorithms to other decision-making



(a) initial investment problem (4) (b) rebalancing problem (9)

Figure 6: Standard deviation of objective function value

problems subject to uncertainty. Many practical problems are framed as scenario-based MILP problems, and our cutting plane algorithms would be useful for solving them.

Acknowledgments

The third author was supported by a Grant-in-Aid for JSPS Fellows.

References

[1] K.C. Ágoston, "CVaR Minimization by the SRA Algorithm," *Central European Journal of Operations Research*, Vol.20, No.4, pp.623–632 (2012).

[2] S. Ahmed, "Convexity and Decomposition of Mean-Risk Stochastic Programs," *Mathematical Programming*, Vol.106, No.3, pp.433–446 (2006).

[3] S. Alexander, T.F. Coleman, and Y. Li, "Minimizing CVaR and VaR for a Portfolio of Derivatives," *Journal of Banking & Finance*, Vol.30, No.2, pp.583–605 (2006).

[4] E. Angelelli, E. Mansini, and M.G. Speranza, "A Comparison of MAD and CVaR Models with Real Features," *Journal of Banking & Finance*, Vol.32, No.7, pp.1188–1197 (2008).

[5] P. Artzner, F. Delbaen, J.M. Eber, and D. Heath, "Coherent Measures of Risk," *Mathematical Finance*, Vol.9, No.3, pp.203–228 (1999).

[6] E.M.L. Beale and J.A. Tomlin, "Special Facilities in a General Mathematical Programming System for Non-Convex Problems Using Ordered Sets of Variables," *Proceedings of the 5th International Conference on Operations Research*, pp.447–454 (1970).

[7] G. Beliakov and A. Bagirov, "Non-Smooth Optimization Methods for Computation of the Conditional Value-at-Risk and Portfolio Optimization," *Optimization*, Vol.55, No.5-6, pp.459–479 (2006).

[8] D. Bertsimas, C. Darnell, and R. Soucy, "Portfolio Construction through Mixed-Integer Programming at Grantham, Mayo, Van Otterloo and Company," *Interfaces*, Vol.29, No.1, pp.49–66 (1999).

[9] C.I. Fábián, "Handling CVaR Objectives and Constraints in Two-Stage Stochastic Models," *European Journal of Operational Research*, Vol.191, No.3, pp.888–911 (2008).

[10] C.I. Fábián and A. Veszpémi, "Algorithms for Handling CVaR-Constraints in Dynamic Stochastic Programming Models with Applications to Finance," *Journal of Risk*, Vol.10, No.3, pp.111–131 (2008).

[11] G. Iyengar and A.K.C. Ma, "Fast Gradient Descent Method for Mean-CVaR Optimization," *Annals of Operations Research*, Vol.205, No.1, pp.203–212 (2013).

[12] M. Kaut, H. Vladimirou, S.W. Wallace, and S.A. Zenios, "Stability Analysis of Portfolio Management with Conditional Value-at-Risk," *Quantitative Finance*, Vol.7, No.4, pp.397–409 (2007).

[13] H. Kellerer, R. Mansini, and M.G. Speranza, "Selecting Portfolios with Fixed Costs and Minimum Transaction Lots," *Annals of Operations Research*, Vol.99, No.1–4, pp.287–304 (2000).

[14] H. Konno, H. Waki, and A. Yuuki, "Portfolio Optimization under Lower Partial Risk Measures," *Asia-Pacific Financial Markets*, Vol.9, No.2, pp.127–140 (2002).

[15] H. Konno and R. Yamamoto, "Global Optimization versus Integer Programming in Portfolio Optimization under Nonconvex Transaction Costs," *Journal of Global Optimization*, Vol.32, No.2, pp.207–219 (2005).

[16] H. Konno and R. Yamamoto, "Integer Programming Approaches in Mean-Risk Models," *Computational Management Science*, Vol.2, No.4, pp.339–351 (2005).

[17] A. Künzi-Bay and J. Mayer, "Computational Aspects of Minimizing Conditional Value-at-Risk," *Computational Management Science*, Vol.3, No.1, pp.3–27 (2006).

[18] C. Lim, H.D. Sherali, and S. Uryasev, "Portfolio Optimization by Minimizing Conditional Value-at-Risk via Nondifferentiable Optimization," *Computational Optimization and Applications*, Vol.46, No.3, pp.391–415 (2008).

[19] M.S. Lobo, M. Fazel, and S. Boyd, "Portfolio Optimization with Linear and Fixed Transaction Costs," *Annals of Operations Research*, Vol.152, No.1, pp.341–365 (2007).

[20] H. Markowitz, "Portfolio Selection," *Journal of Finance*, Vol.7, No.1, pp.77–91 (1952).

[21] A.F. Perold, "Large-Scale Portfolio Optimization," *Management Science*, Vol.30, No.10, pp.1143–1160 (1984).

[22] G.Ch. Pflug, "Some Remarks on the Value-at-Risk and the Conditional Value-at-Risk," In S. Uryasev (ed.), *Probabilistic Constrained Optimization: Methodology and Applications*, pp.272–281 (Kluwer Academic Publishers, Dordrecht, 2000).

[23] R.T. Rockafellar and S. Uryasev, "Optimization of Conditional Value-at-Risk," *Journal of Risk*, Vol.2, No.3, pp.21–42 (2000).

[24] R.T. Rockafellar and S. Uryasev, "Conditional Value-at-Risk for General Loss Distributions," *Journal of Banking & Finance*, Vol.26, No.7, pp.1443–1471 (2002).

[25] R.M. Van Slyke and R. Wets, "L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming," *SIAM Journal on Applied Mathematics*, Vol.17, No.4, pp.638–663 (1969).

[26] A. Takeda and T. Kanamori, "A Robust Approach Based on Conditional Value-at-Risk Measure to Statistical Learning Problems," *European Journal of Operational Research*, Vol.198, No.1, pp.287–296 (2009).

[27] 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*, Vol.216, No.6, pp.1723–1740 (2010).

[28] R. Yamamoto and H. Konno, "An Efficient Algorithm for Solving a Mean-Variance Model under Nonconvex Transaction Costs," *Pacific Journal of Optimization*, Vol.2, No.2, pp.385–394 (2006).

[29] R. Yamamoto and H. Konno, "Rebalance Schedule Optimization of a Large Scale Portfolio under Transaction Cost," *Journal of the Operations Research Society of Japan*, Vol.56, No.1, pp.26–37 (2013).