Math. Program., Ser. B 89: 233–250 (2001)

Digital Object Identifier (DOI) 10.1007/s101070000205

Hiroshi Konno · Annista Wijayanayake

Portfolio optimization problem under concave transaction costs and minimal transaction unit constraints

Received: July 15, 1999 / Accepted: October 1, 2000
Published online December 15, 2000 –

Abstract. We will propose a branch and bound algorithm for calculating a globally optimal solution of a portfolio construction/rebalancing problem under concave transaction costs and minimal transaction unit constraints. We will employ the absolute deviation of the rate of return of the portfolio as the measure of risk and solve linear programming subproblems by introducing (piecewise) linear underestimating function for concave transaction cost functions. It will be shown by a series of numerical experiments that the algorithm can solve the problem of practical size in an efficient manner.

Key words. portfolio optimization – concave transaction cost – rebalancing – minimal transaction unit – branch and bound algorithm – global optimization

1. Introduction

This paper is concerned with a portfolio optimization problem under nonconvex (concave) transaction costs and minimal transaction unit (MTU) constraints.

An investor has to pay a certain amount of fees when he/she purchases (invest) and/or sells (disinvest) assets. Let X_j be the amount of investment (or disinvestment) of the jth asset ($j = 1, \cdots, n$). The transaction cost associated with $X = (X_1, \cdots, X_n)$ is usually defined as the sum of the functions $\sum_{j=1}^{n} C_j(X_j)$ where each function $C_j(X_j)$ is a non-decreasing concave function up to certain point $\bar{X}_j$ as shown in Fig. 1. This is because the unit transaction cost is relatively large when X_j is small and it gradually decreases as X_j increases. However, the unit transaction cost increases beyond some point $\bar{X}_j$, due to the "illiquidity" effects. Let us briefly explain this phenomena.

Investment (disinvestment) of assets of an investor is associated with the disinvestment (investmet) of other investors. If X_j is large and if there is not enough supply (demand), then the unit price will increase (decrease), so that $C_j(X_j)$ becomes convex beyond point A_j.

We will assume in this paper that the amount of investment (disinvestment) is below the critical point, where the transaction cost is a well specified concave function

H. Konno: Department of Industrial Engineering & Management and Center for Research in Advanced Financial Technologies, Tokyo Institute of Technology, Japan. e-mail: `konno@me.titech.ac.jp`

A. Wijayanayake: Department of Industrial Engineering & Management and Tokyo Institute of Technology, Japan. e-mail: `anni@me.titech.ac.jp`

Mathematics Subject Classification (1991): 91B28

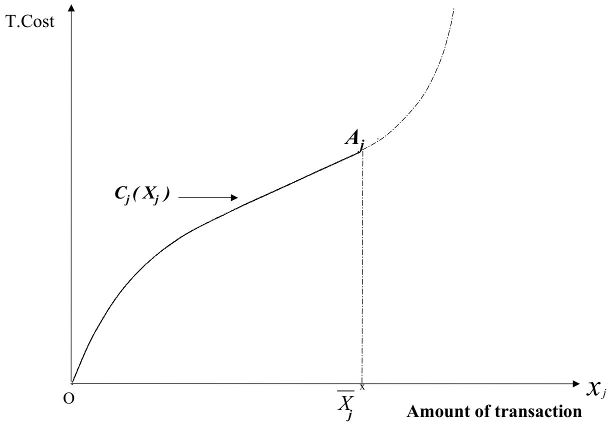

Fig. 1. Transaction cost function

calculated by the transaction cost table of the agent. The portfolio optimization problem to be treated in this paper becomes the minimization of a separable concave function under linear constraints. This type of problems are abundant in the literature of operation research. However, only a few problems have been solved to global optimality, at least until recently.

People often employ a linear (convex) approximation of a transaction cost function, or entirely ignore it and adjust the resulting solution in a heuristic manner. This manipulation may be adequate under certain circumstances, particularly if the amount of transaction is large when a linear approximation or heuristic approach may lead to a good solution. However, this approach may produce an erroneous result when the amount of investment is small and it is allocated to many assets in smaller fractions. Alternatively, one may apply integer programming approach by using piecewise linear approximation of nonconvex cost functions. However, as reported in [7], it takes a huge amount of computation time.

Fortunately, due to the recent progress in global optimization, one can now solve a fairly large scale linearly constrained concave minimization problem using the special structure of the problem (See [5,8,17] for such examples). This paper adds to the list of these successful examples.

The first serious effort in portfolio optimization under nonconvex cost structure was undertaken by Perold [15] in 1984. He proposed a piecewise linear convex approximation of the cost function and implemented it in the highly reputed software “Optimizer”. However, it is argued among the users of this software that this transaction cost routine requires a significant amount of computation time.

More recently, authors proposed a branch and bound algorithm for solving a concave cost portfolio optimization problem under the mean-absolute deviation framework. We used a linear underestimating function for the concave cost function and solved the resulting linear subproblems in a branch and bound using a well-designed problem reduction technique. We showed in [10] that this algorithm generates a good solution in a very efficient manner. We will show in this paper that the solution obtained by this approach always leads to a solution which is very close to the globally optimal solution of the original problem. Also, we extend this algorithm to the minimal cost rebalancing problems to be explained below.

Investors are inclined to “rebalance” the portfolio due to the change of investment environment. The current portfolio which was located on the efficient frontier calculated at the time of portfolio construction may deviate from the present efficient frontier. Therefore the investor may want to sell (disinvest) and/or purchase (invest) a small portion of the portfolio to push it back to the present efficient frontier.

We will formulate this problem as the piecewise-concave cost minimization problem and propose a branch and bound method using a piecewise linear convex underestimation strategy. We will extend this algorithm to even more difficult class of problem where the amount of transaction is constrained to be the integer multiple of minimal transaction units. In Tokyo Stock Exchange, minimal transaction unit is 1000 stocks with a few exception. This constraint may not be ignored when the amount of fund is small. The problem thus becomes a concave minimization problem with integer constraints on the variables.

We will show that the branch and bound algorithm can be modified in such a way that the generated solution consists of only a few (usually at most 3 or 4) nonintegral variables, regardless of the number of variables in the model. Thus we can obtain an integer solution by a simple rounding procedure, which is almost optimal in practical sense.

In the next section, we will formulate the portfolio optimization problem under concave transaction cost assuming that the risk function is given by the absolute deviation of the rate of return, instead of the standard deviation (or variance) employed by Markowitz [12]. The mean-absolute deviation (MAD) model is now widely used in large-scale portfolio optimization because it can be reduced to a linear programming problem instead of a quadratic programming problem in the case of mean-variance (MV) model. The success of our branch and bound algorithm critically depends upon the employment of the absolute deviation as the risk measure.

In Sect. 3, we will present the branch and bound method proposed in a recent paper [10]. Section 4 will be devoted to problem reduction technique and the treatment of minimal transaction unit (MTU) constraints. In Sect. 5, we will extend the algorithm presented in Sect. 3 to the rebalancing problem. Section 6 will be devoted to the result of numerical simulation of the algorithm. It will be shown that we can solve a fairly large scale problem in a reasonable amount of time.

2. Mean-absolute deviation model under concave transaction costs

Let there be n assets in the market and let R_j be the random variable representing the rate of return of jth asset $(j = 1, \cdots, n)$ without transaction cost. We will assume that the vector of random variables $(R_1, \cdots, R_n)$ is distributed over a set of finitely many points $\{(r_{1t}, \cdots, r_{nt}),\ t = 1, \cdots, T\}$ and that the probability

$$p_t = P_r\{(R_1, \cdots, R_n,) = (r_{1t}, \cdots, r_{nt})\},\ t = 1, \cdots, T, \tag{1}$$

is known. Then the expected rate of return r_j of jth asset (without transaction cost) is given by

$$r_j = \sum_{t=1}^{T} p_t r_{jt}. \tag{2}$$

Let

$$x_j = X_j / \sum_{k=1}^{n} X_k, \tag{3}$$

be the proportion of the fund to be allocated to jth asset. Then the expected rate of return of the portfolio $\boldsymbol{x}=(x_1, \cdots, x_n)$ is given by $\sum_{j=1}^{n} r_j x_j$. The actual expected rate of return under transaction costs is therefore, given by

$$r(\mathbf{x}) = \sum_{j=1}^{n} \left\{ r_j x_j - c_j(x_j) \right\}, \tag{4}$$

where $c_j(x_j)$ is a concave transaction cost associated with the investment into stock S_j. As in [10], we will employ the absolute deviation:

$$W[\boldsymbol{x}] = E[\,|R(\boldsymbol{x}) - E[R(\boldsymbol{x})]\,|\,] = \sum_{t=1}^{T} p_t \Big| \sum_{j=1}^{n} (r_{jt} - r_j) x_j \Big|, \tag{5}$$

as the measure of risk. The mean-absolute deviation (MAD) model under transaction cost is defined as the following convex maximization problem:

$$\left|\begin{array}{lll} \text{maximize} & f(\boldsymbol{x}) \equiv \sum_{j=1}^{n} \left\{ r_j x_j - c_j(x_j) \right\} & \\ \text{subject to} & \sum_{t=1}^{T} p_t \Big| \sum_{j=1}^{n} (r_{jt} - r_j) x_j \Big| & \leq w, \\ & \sum_{j=1}^{n} x_j & = 1, \\ & 0 \leq x_j \leq \alpha_j, \quad j = 1, \cdots, n. & \end{array}\right. \tag{6}$$

where w is the specified level of risk and α_j's are constants.

The MAD model (without transaction cost) was first proposed by one of the authors [9] as an alternative to the classical mean-variance (MV) model [2,12]. It has been demonstrated in [9] that this model can generate an optimal portfolio much faster than the MV model since it can be reduced to a linear programming problem. Also, it is shown in [9] that MV and MAD model usually generate similar portfolios. Further, it is proved in [14] that those portfolios on the MAD efficient frontier correspond to efficient portfolios in the sense of the second degree stochastic dominance, regardless of the distribution of the rate of return, which is not valid for the portfolios on the MV efficient frontier.

First let us introduce a set of nonnegative variables $y_t, z_t,\ \ (t = 1, \cdots, T)$ satisfying the following conditions.

$$y_t - z_t = p_t \sum_{j=1}^{n} (r_{jt} - r_j)x_j,\ t = 1, \cdots, T,$$
$$y_t z_t = 0\,;\ y_t \geq 0\,;\ z_t \geq 0, \quad t = 1, \cdots, T.$$

Then we have the following representation

$$|p_t \sum_{j=1}^{n} (r_{jt} - r_j)x_j| = y_t + z_t,\ \ t = 1, \cdots, T. \tag{7}$$

Therefore, the problem (6) can be rewritten as follows.

$$\left|\begin{array}{lll} \text{maximize } & f(\boldsymbol{x}) \equiv \sum_{j=1}^{n} \left\{ r_j x_j - c_j(x_j) \right\} & \\ \text{subject to} & \sum_{t=1}^{T} (y_t + z_t) \leq w, & \\ & y_t - z_t = p_t \sum_{j=1}^{n} (r_{jt} - r_j)x_j, & t = 1, \cdots, T, \\ & y_t z_t = 0, & t = 1, \cdots, T, \\ & \sum_{j=1}^{n} x_j = 1,\ \ 0 \leq x_j \leq \alpha_j, & j = 1, \cdots, n, \\ & y_t \geq 0,\ \ z_t \geq 0, & t = 1, \cdots, T. \end{array}\right. \tag{8}$$

Theorem 1. *The complementarity constraint* $y_t z_t = 0\ (t = 1, \cdots, T)$ *can be eliminated from (8).*

Proof. Let $(x_1^*, \cdots, x_n^*,\ y_1^*, \cdots, y_T^*,\ z_1^*, \cdots, z_T^*)$ be an optimal solution of the problem (8) without complementarity constraint and let us assume that $y_t^* z_t^* > 0, t \in I \subset \{1, \cdots, T\}$. For $t \in I$, let $(\hat{y}_t, \hat{z}_t) = (y_t^* - z_t^*, 0)$ if $y_t^* \geq z_t^* \geq 0$, and $(\hat{y}_t, \hat{z}_t) = (0, y_t^* - z_t^*,)$ if $z_t^* \geq y_t^* \geq 0$,. Then $(x_1^*, \cdots, x_n^*,\ \hat{y}_1, \cdots, \hat{y_T}\ \hat{z}_1, \cdots, \hat{z_T})$ satisfies all the constraints of (8). Also it has the same objective function value as $(x_1^*, \cdots, x_n^*,\ y_1^*, \cdots, y_T^*\ , z_1^*, \cdots, z_T^*)$.

□

In view of the relation

$$\sum_{t=1}^{T} (y_t - z_t) = \sum_{t=1}^{T} p_t \sum_{j=1}^{n} (r_{jt} - r_j)x_j = \sum_{j=1}^{n} (\sum_{t=1}^{T} p_t r_{jt} - r_j)x_j = 0,$$

we can eliminate $(z_1, \cdots, z_T)$ from (8) to obtain an alternative representation:

$$\left|\begin{array}{l} \text{maximize } f(\boldsymbol{x}) \equiv \sum_{j=1}^{n} \left\{ r_j x_j - c_j(x_j) \right\} \\ \text{subject to } \sum_{t=1}^{T} y_t \le w/2, \\ \qquad y_t \ge p_t \sum_{j=1}^{n} (r_{jt} - r_j) x_j, \quad y_t \ge 0,\ t = 1, \cdots, T, \\ \qquad \sum_{j=1}^{n} x_j = 1, \quad 0 \le x_j \le \alpha_j, \qquad j = 1, \cdots, n. \end{array}\right. \tag{9}$$

3. A branch and bound algorithm

Let us describe a branch and bound algorithm for solving the problem (9). Let $\boldsymbol{x} = (x_1, \cdots, x_n)$, $\boldsymbol{y} = (y_1, \cdots, y_T)$ and let

$$F = \{(\boldsymbol{x}, \boldsymbol{y}) | \sum_{t=1}^{T} y_t \le w/2, \ \ y_t - p_t \sum_{j=1}^{n} (r_{jt} - r_j) x_j \ge 0, \\ \sum_{j=1}^{n} x_j = 1, \quad y_t \ge 0, \quad t = 1, \cdots, T\}. \tag{10}$$

The problem (9) can be denoted as follows:

$$(P_0) \left|\begin{array}{l} \text{maximize } f(\boldsymbol{x}) = \sum_{j=1}^{n} \left\{ r_j x_j - c_j(x_j) \right\} \\ \text{subject to } (\boldsymbol{x}, \boldsymbol{y}) \in F, \quad \boldsymbol{0} \le \boldsymbol{x} \le \boldsymbol{\alpha}, \end{array}\right. \tag{11}$$

where $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_n)$. Let $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ be an optimal solution of (11) and let $f^* = f(\boldsymbol{x}^*)$.

Let us replace $c_j(x_j)$ by a linear underestimating function $\delta_j x_j$ as depicted in Fig. 2 and define a linear programming problem:

$$(Q_0) \left|\begin{array}{l} \text{maximize } g_0(\boldsymbol{x}) = \sum_{j=1}^{n} (r_j - \delta_j) x_j \\ \text{subject to } (\boldsymbol{x}, \boldsymbol{y}) \in F, \quad \boldsymbol{0} \le \boldsymbol{x} \le \boldsymbol{\alpha}. \end{array}\right. \tag{12}$$

Let $\boldsymbol{x}^0$ be an optimal solution of (Q_0). If

$$\sum_{j=1}^{n} \left\{ c_j\left(x_j^0\right) - \delta_j x_j^0 \right\} \le \varepsilon, \tag{13}$$

then $(\boldsymbol{x}^0, \boldsymbol{y}^0)$ is an approximate optimal solution of (P_0) with error less than ε.

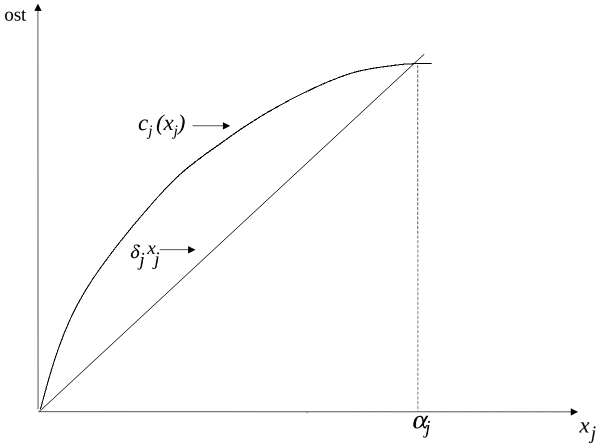

Fig. 2. A linear underestimating function

Theorem 2. *The following relation holds:*

$$g_0(\boldsymbol{x}^0) \geq f^* \geq f(\boldsymbol{x}^0). \tag{14}$$

Proof. We have

$$\begin{aligned} g_0(\boldsymbol{x}^0) &= \max x \,\{g_0(\boldsymbol{x}) \mid (\boldsymbol{x}, \boldsymbol{y}) \in F, \ \ \boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{\alpha}\} \\ &\geq \max x \,\{ f(\boldsymbol{x}) \mid (\boldsymbol{x}, \boldsymbol{y}) \in F, \ \ \boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{\alpha}\} \ = \ f^*. \end{aligned}$$

because $g_0(\boldsymbol{x}) \geq f(\boldsymbol{x}), \forall \boldsymbol{x} \in [\boldsymbol{0}, \boldsymbol{\alpha}]$. Also $f^* \geq f(\boldsymbol{x}^0)$, because $\boldsymbol{x}^0$ is a feasible solution of (12).

□

Let us consider the case when (13) does not hold. Let

$$c_s\left(x_s^0\right) - \delta_s x_s^0 = \max\left\{c_j\left(x_j^0\right) - \delta_j x_j^0 \,\middle|\, j = 1, \cdots, n\right\}, \tag{15}$$

and let

$$S_1 = \{\boldsymbol{x} | \, 0 \leq x_s \leq \alpha_s/2, \ 0 \leq x_j \leq \alpha_j \,, \ \ j \neq s\}, \tag{16}$$

$$S_2 = \{\boldsymbol{x} | \, \alpha_s/2 \ \leq x_s \leq \alpha_s, \ 0 \leq x_j \leq \alpha_j \,; \ \ j \neq s\}, \tag{17}$$

and define two subproblems

$$(P_1) \ \ \text{maximize} \ \{ f(\boldsymbol{x}) | \ (\boldsymbol{x}, \boldsymbol{y}) \in F, \ \ \boldsymbol{x} \in S_1\}, \tag{18}$$

$$(P_2) \ \ \text{maximize} \ \{ f(\boldsymbol{x}) | \ (\boldsymbol{x}, \boldsymbol{y}) \in F, \ \ \boldsymbol{x} \in S_2\}. \tag{19}$$

Corresponding to subproblem (P_1), we will define a piecewise linear function $g_1(\boldsymbol{x})$ underestimating $f(\boldsymbol{x})$ as shown in Fig. 3:

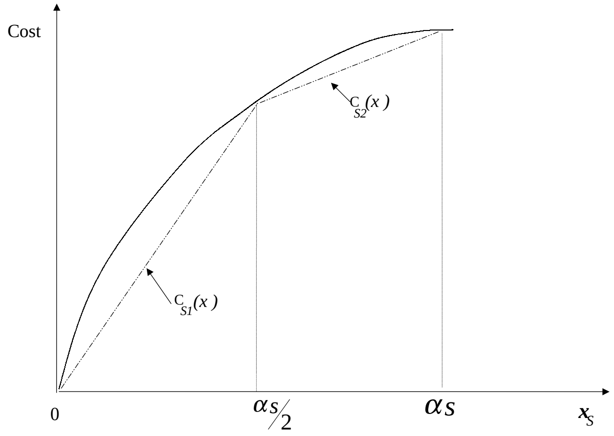

Fig. 3. Bisection scheme

$$g_1(\boldsymbol{x}) = \sum_{j \neq s} (r_j - \delta_j) x_j + (r_s x_s - c_{s1}(x_s)). \tag{20}$$

Define a linear programming problem:

$$(Q_1) \quad \text{maximize } \{ g_1(\boldsymbol{x}) \mid (\boldsymbol{x}, \boldsymbol{y}) \in F, \ \ \boldsymbol{x} \in S_1 \}. \tag{21}$$

If (Q_1) is infeasible, then (Q_1) can be fathomed. Otherwise, let $\hat{\boldsymbol{x}}^1$ be an optimal solution of (Q_1). If $\left| g_1(\hat{\boldsymbol{x}}^1) - f(\hat{\boldsymbol{x}}^1) \right| < \varepsilon$, then the problem has been solved with an approximate optimal solution $\hat{\boldsymbol{x}}^1$. Since $g_1(\boldsymbol{x}) \geq f(\boldsymbol{x}), \ \ \forall \, \boldsymbol{x} \in S_1$, we have

$$g_1(\hat{\boldsymbol{x}}^1) \geq f(\boldsymbol{x}^1) \geq f(\hat{\boldsymbol{x}}^1). \tag{22}$$

Therefore, if $g_1(\hat{\boldsymbol{x}}^1) \leq f(\boldsymbol{x}^0)$, then (P_1) can be fathomed since $f(\boldsymbol{x}^{\boldsymbol{1}}) \leq f(\boldsymbol{x}^0)$.

Algorithm 1 (Branch and bound method).

1° $\boldsymbol{P} = \{(P_0)\}, \ \hat{f} = -\infty, \ k = 0$.
2° If $\boldsymbol{P} = \{\phi\}$, then goto 9°; Otherwise goto 3°.
3° Choose a problem $(P_k) \in \boldsymbol{P}$:

$$(P_k) \left| \begin{array}{l} \text{maximize } \ f(\boldsymbol{x}) = \displaystyle\sum_{j=1}^{n} \left\{ r_j x_j - c_j(x_j) \right\} \\ \\ \text{subject to } \ (\boldsymbol{x}, \boldsymbol{y}) \in F, \quad \boldsymbol{\beta}^k \leq \boldsymbol{x} \leq \boldsymbol{\alpha}^k. \end{array} \right.$$

$$\boldsymbol{P} = \boldsymbol{P} \backslash \{(P_k)\}.$$

4° Let $c_j^k(x_j)$ be a linear underestimating function of $c_j(x_j)$ over the interval $\beta_j^k \le x_j \le \alpha_j^k$, $(j = 1, \cdots, n)$ and define a linear programming problem

$$(Q_k)\quad \left|\begin{array}{l} \text{maximize } g_k(\boldsymbol{x}) = \sum_{j=1}^{n} \left\{ r_j x_j - c_j^k(x_j) \right\} \\ \\ \text{subject to } (\boldsymbol{x}, \boldsymbol{y}) \in F, \quad \boldsymbol{\beta}^k \le \boldsymbol{x} \le \boldsymbol{\alpha}^k. \end{array}\right.$$

If (Q_k) is infeasible then go to 2°. Otherwise let $\boldsymbol{x^k}$ be an optimal solution of (Q_k).

If $|g_k(\boldsymbol{x^k}) - f(\boldsymbol{x^k})| > \varepsilon$ then goto 8°. Otherwise let $f_k = f(\boldsymbol{x^k})$.

5° If $f_k < \hat{f}$ then goto 7°; Otherwise goto 6°.

6° If $\hat{f} = f_k$; $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}^k$ and eliminate all the subproblems (P_t) for which $g_t(\boldsymbol{x^t}) \le \hat{f}$.

7° If $g_k(\boldsymbol{x^k}) \le \hat{f}$ then goto 2°. Otherwise goto 8°.

8° Let $c_s(x_s^k) - c_s^k(x_s^k) = \max\{c_j(x_j^k) - c_j^k(x_j^k) |\ j = 1, \cdots, n\}$,

$$S_{l+1} = S_k \cap \left\{ \boldsymbol{x} \,\middle|\, \beta_s^k \le x_s \le \left(\beta_s^k + \alpha_s^k\right)/2 \right\},$$
$$S_{l+2} = S_k \cap \left\{ \boldsymbol{x} \,\middle|\, \left(\beta_s^k + \alpha_s^k\right)/2 \le x_s \le \alpha_s^k \right\},$$

and define two subproblems:

$$(P_{l+1})\quad \text{maximize } \{ f(\boldsymbol{x}) |\ (\boldsymbol{x}, \boldsymbol{y}) \in F,\ \boldsymbol{x} \in S_{l+1} \}.$$
$$(P_{l+2})\quad \text{maximize } \{ f(\boldsymbol{x}) |\ (\boldsymbol{x}, \boldsymbol{y}) \in F,\ \boldsymbol{x} \in S_{l+2} \}.$$

$\boldsymbol{P} = \boldsymbol{P} \cup \{P_{l+1}, P_{l+2}\}$, $k = k + 1$ and goto 3°.

9° Stop: $\hat{\boldsymbol{x}}$ is an ε optimal solution of (P_0).

Theorem 3. *$\hat{\boldsymbol{x}}$ converges to an ε - optimal solution of (P_0) as $k \to \infty$.*

Proof. See e.g. [8,17]

□

To accelerate convergence, we may replace the bisection scheme by the ω - subdivision scheme [8], in which the interval $[\beta_s^k, \alpha_s^k]$ is divided into two subintervals $[\beta_s^k, x_s^k]$ and $[x_s^k, \alpha_s^k]$ where x_s^k is the sth component of the optimal solution $\boldsymbol{x^k}$ of (Q_k) as depicted in Fig. 4. This subdivision scheme usually accelerate convergence.

This type of branch and bound algorithm was first proposed in [4]. The elaborate version of the algorithm was implemented by Phong et al. [16] to obtain successful computational results for linearly constrained concave quadratic programming problems. Also, it has been applied to many other problems in recent years (see [7,10,11]).

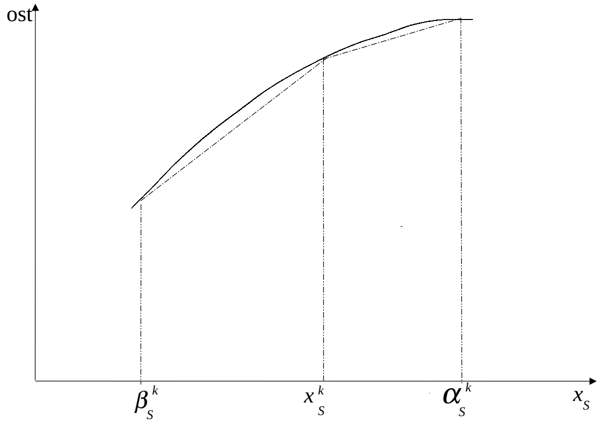

Fig. 4. ω - subdivision

4. Problem reduction and minimal transaction unit constraints

The problem (P_0) can, in principle, be solved by Algorithm 1. However, computation time is expected to increase rapidly as the number of assets increases. In this section, we will discuss a problem reduction scheme using the following theorem.

Theorem 4. *There exist an optimal solution* $\boldsymbol{x^0}$ *of* (Q_0)*, at most* $T+1$ *components of which satisfy* $0 < x_j^0 < \alpha_j$

Proof. Let $\hat{\boldsymbol{x}},\hat{\boldsymbol{y}}$ be an optimal solution of (Q_0). Then $\hat{\boldsymbol{x}}$ is an optimal solution of the following linear programming problem:

$$\left|\begin{array}{ll} \text{maximize } g_0(\boldsymbol{x}) \equiv \displaystyle\sum_{j=1}^{n}(r_j-\delta_j)x_j & \\ \text{subject to } \ p_t \displaystyle\sum_{j=1}^{n}(r_{jt}-r_j)x_j \le \hat{y}_t, \quad t=1,\cdots,T, & \\ \qquad\qquad \displaystyle\sum_{j=1}^{n} x_j = 1,\ \ 0 \le x_j \le \alpha_j,\ j=1,\cdots,n. & \end{array}\right. \tag{23}$$

By the fundamental theorem of linear programming [1], this problem has a basic optimal solution, say $\boldsymbol{x^0}$. It is easy to see that $\boldsymbol{x^0}$ contains at most $T+1$ components with property $0 < x_j^0 < \alpha_j$. Since $g(\hat{\boldsymbol{x}}) = g(\boldsymbol{x^0})$, the theorem follows.

□

This theorem implies that at least $n-(T+1)$ components of $\boldsymbol{x^0}$ attain their lower bound or upper bound. For these variable x_j's, we have the relation $c_j(x_j^0) = \delta_j x_j^0$. Therefore only these variables x_j' s such that $0 < x_j^0 < \alpha_j$ are subject to approximation error. Let us note that the number of such variables are at most $T+1$, usually less than $T/2$, so that the approximation error remains small even if n is very large.

Also, those assets with smaller amount of investment are subject to relatively large transaction cost, since the unit transaction cost is larger when the transaction amount is smaller. Therefore, there is a good reason to believe that those assets with $x_j^0 = 0$ are likely to satisfy $x_j^* = 0$ where $\boldsymbol{x}^*$ is an optimal solution of (P_0). These observations lead us to the following variable reduction scheme in the branch and bound procedure.

Let us assume without loss of generality that the first $J(\leq T+1)$ components of $\boldsymbol{x}^0$ are positive. Let us define the reduced problem:

$$(P_0') \quad \left| \begin{array}{ll} \text{maximize } g_0(\boldsymbol{x}) = \sum_{j=1}^{J} \{r_j x_j - c_j(x_j)\} & \\ \text{subject to} \quad (x_1, \cdots, x_J, 0, \cdots, 0, \ \boldsymbol{y}) \in F, & \\ \qquad\qquad 0 \leq x_j \leq \alpha_j, \quad j = 1, \cdots, J. & \end{array} \right. \tag{24}$$

We can solve this problem much faster than the problem (P_0) particularly when $n > T$, which is the case in the practical investment environment where $n \geq 1000$ and $T \leq 60$.

Theorem 4 plays more important role in the treatment of minimal transaction unit (MTU) constraints. We will naturally assume in this case that α_j' s are chosen as integer multiple of MTU. Then at least $n-(T+1)$ assets satisfy these constraints at $\boldsymbol{x}^0$. In the subsequent branch and bound scheme, we will replace the bisection or ω-subdivision scheme in such a way that the associated lower bound or upper bound to be equal to the nearest integer multiple of MTU. It will be shown in Sect. 6 that these strategies are remarkably effective in generating a nearly optimal solution of the problem (P_0) with or without MTU constraints.

5. Minimal cost rebalancing under concave transaction costs

The algorithm developed in Sect. 3 can be extended to the rebalancing problem. Let $\boldsymbol{x}^0$ be the portfolio constructed at time point 0 and assume that the investor wants to change the portfolio at some later time point, say 1 month or 3 months later, in such a way that the new portfolio $\boldsymbol{x}$ satisfies the condition that its expected rate of return $E[R(\boldsymbol{x})]$ is greater than some constant ρ and that the risk is $W[R(\boldsymbol{x})]$ is smaller than some constant w.

Let S be an investable set. Then the minimum cost rebalancing problem can be formulated as follows.

$$\left| \begin{array}{ll} \text{minimize } c(\boldsymbol{x}) & \\ \text{subject to} & E[R(\boldsymbol{x})] \geq \rho, \\ & W[R(\boldsymbol{x})] \leq w, \\ & \boldsymbol{x} \in S, \end{array} \right. \tag{25}$$

where $c(\boldsymbol{x})$ is the rebalancing cost and

$$S = \{\boldsymbol{x} |\, 0 \leq x_j \leq \alpha_j, \ \ j = 1, \cdots, n\}. \tag{26}$$

Let us introduce a new set of variables

$$\boldsymbol{v} = \boldsymbol{x} - \boldsymbol{x}^0 \tag{27}$$

Then the problem (25) can be represented as follows:

$$(H)\left|\begin{array}{ll} \text{minimize} & \sum_{j=1}^{n} c_j(v_j) \\ \text{subject to} & \sum_{j=1}^{n} r_j v_j \geq \rho - \sum_{j=1}^{n} r_j x_j^0, \\ & \sum_{t=1}^{T} |z_t| \leq w, \\ & z_t = p_t \sum_{j=1}^{n} [(r_{jt} - r_j)v_j + (r_{jt} - r_j)x_j^0], \; t = 1, \cdots, T, \\ & -x_j^0 \leq v_j \leq \alpha_j - x_j^0, \qquad j = 1, \cdots, n, \\ & \boldsymbol{v} \in V, \end{array}\right. \tag{28}$$

where V is the set of feasible $\boldsymbol{v}'$s corresponding to S, and $c_j(v_j)$ is the cost associated with purchasing v_j units (if $v_j > 0$) and selling v_j units (if $v_j < 0$) of jth asset. Let us assume again that $c_j(v_j)$ is piecewise concave and that $c_j(0) = 0$ for all j (Fig. 5).

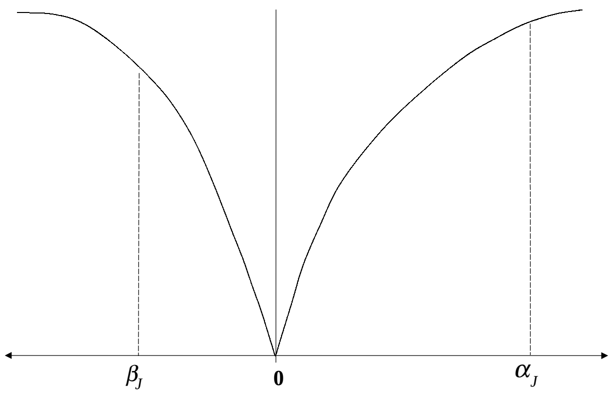

Fig. 5. Piecewise concave cost function

We can construct a branch and bound method similar to the one discussed in Sect. 3. Let (H_k) be a subproblem:

$$(H_k)\left|\begin{array}{ll} \text{minimize} & \sum_{j=1}^{n} c_j(v_j) \\ \text{subject to} & (\boldsymbol{v}, \boldsymbol{z}) \in G, \\ & \beta_j^k \leq v_j \leq \alpha_j^k, \; j = 1, \cdots, n, \end{array}\right.$$

where

$$G = \{(\boldsymbol{v}, \boldsymbol{z}) | \sum_{j=1}^{n} r_j v_j \geq \rho - \sum_{j=1}^{n} r_j x_j^0, \;\; \sum_{t=1}^{T} |z_t| \leq w,$$
$$z_t = p_t \sum_{j=1}^{n} \left[(r_{jt} - r_j) v_j + (r_{jt} - r_j) x_j^0 \right], \; t = 1, \cdots, T, \; \boldsymbol{v} \in V\}.$$

We will approximate the function $c_j(v_j)$ in the interval $[\alpha_j^k, \beta_j^k]$ by a piecewise-linear convex underestimating function $c_j^k(v_j)$ as depicted in Fig. 6 and define a relaxed

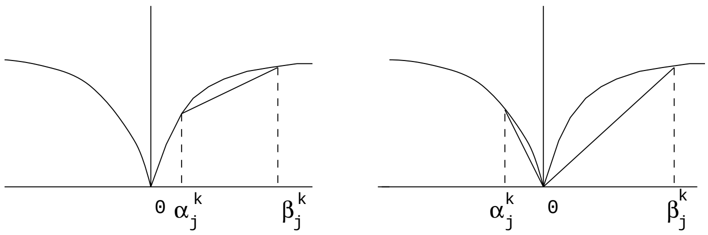

Fig. 6. Piecewise linear underestimating function

subproblem:

$$(\bar{H}_k) \left| \begin{array}{ll} \text{minimize} & \sum_{j=1}^{n} c_j^k(v_j) \\ \text{subject to} & (\boldsymbol{v}, \boldsymbol{z}) \in G, \\ & \beta_j^k \leq v_j \leq \alpha_j^k, \; j = 1, \cdots, n, \end{array} \right.$$

which can be reduced to a linear programming problem by using a standard method [1].

6. Computational experiments

We conducted numerical tests of the algorithm proposed in this paper using monthly data of 200 stocks chosen from NIKKEI 225 Index. The program was coded by C^{++} and was tested on Pentium Pro 450MHz with 256 Mbyte memory. We used the breadth first rule for choosing subproblems in Step 3^0 of the Algorithm 1 and $\omega-$ subdivision strategy throughout the test. Also we choose $\varepsilon = 10^{-5}$ in our computation. We tested the algorithm for three different levels of the investment M, namely $10^8, 3 \times 10^8$, and 10^9 yen and using the transaction cost table of a leading security company of Japan.

6.1. *Computational results of portfolio construction problem* (P_0)

Figure 7 shows computation time for solving the problem (P_0) by Algorithm 1 for different number of assets without using problem reduction strategy. We solved ten test problems corresponding to ten different sets of historical data and plotted the average computation time and its standard deviation.

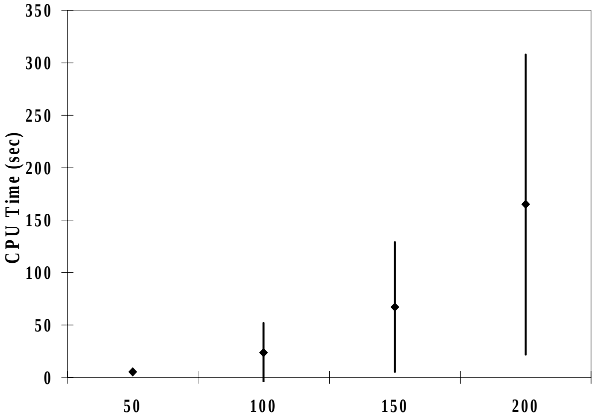

Fig. 7. CPU time for solving (P_0) for different n

We see from this figure that the average and the variance of the computation time increases as n increases, but not as rapidly as we expected. This is primarily due to the fact that the starting solution generated by (Q_0) is a very good feasible solution of (P_0), so that many subproblems are fathomed by bounds.

Figure 8 shows the computation time when the problem reduction strategy is employed. We see that the average computation time is around 15% of the Algorithm 1 without problem reduction. Also, the relative difference of the objective function value of the algorithm with and without problem reduction is at most 1.13%, usually much less (less than 0.5%). We conclude from this that problem reduction is a very effective strategy for solving problems with large n. The average computation time increases less mildly as n increases (see Fig. 7). Also it is very insensitive to the level of ε.

Figures 9 and 10 show the computation time as a function of T, the number of data when $n = 200$.We see that the increase of the average computation time is very mild. We can safely conclude that the problem up to $T = 60$ can be solved in less than a few minutes. Let us note that the maximal size of T is usually less than 60 in practical applications.

6.2. *Portfolio construction problem with MTU constraints*

We conducted similar experiments for the problem with MTU constraints. When the amount of investment M is fixed, there is no guarantee that there exists a so-

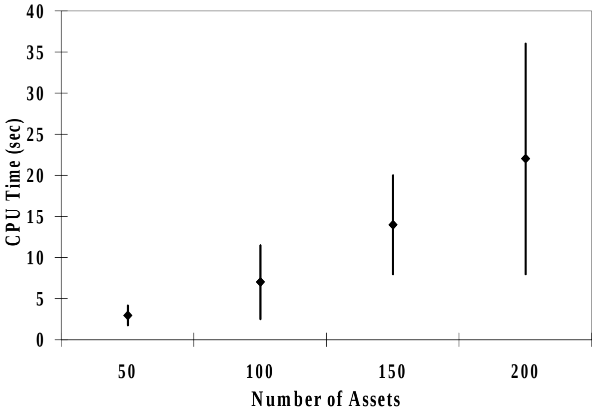

Fig. 8. CPU time for different n with problem reduction

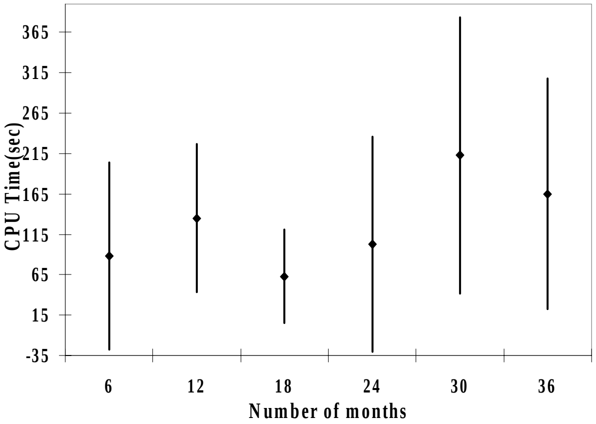

Fig. 9. CPU time for solving (P_0) for different T

lution satisfying MTU constraints. Therefore, we terminated computation as soon as the error decreased below ε and round the solution to the nearest solution satisfying MTU constraints. Table 1 shows the statistics of the computation time for $n = 200, T = 36, \alpha_j = 0.05, \& \, 0.1$ and for different level of investments, where the problem reduction strategy was employed. We see that the number of assets which do not satisfy the MTU constraints in the solution is at most 6 and the necessary amount of fund adjustment is less than 0.5% when $M = 10^8$, but it decreases to 0.01% when $M = 10^9$, which is almost negligible from the practical point of view.

6.3. *Minimal cost rebalancing problem*

We tested the branch and bound algorithm for rebalancing problem. We first calculate the starting portfolio by solving the problem (P_0') using 36 monthly data for fixed value

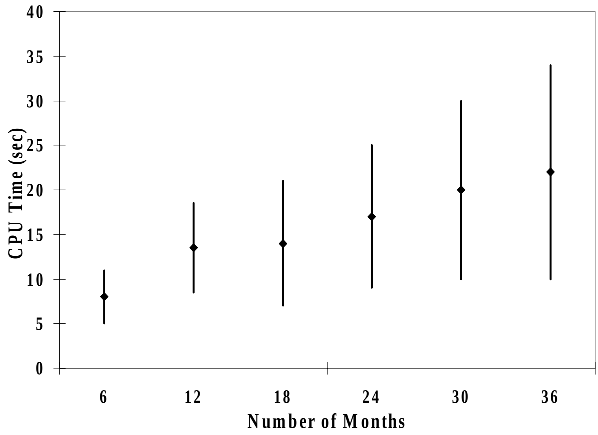

Fig. 10. CPU time for different T with problem reduction

Table 1. Statistical results for different level of α and M

α_j's	w	M million (yen)	ρ	# of branching	CPU time (Sec)	# of assets violating MTU constraints	# of assets in the portfolio	fund adjustment
0.05	1.8	100	0.2850	16	32	5	24	0.5025
	2.0	100	0.6148	10	18	6	24	−0.4261
	2.0	300	0.8341	14	28	4	23	−0.0975
	2.6	300	1.3200	12	22	2	21	0.0697
	2.0	1000	1.0470	32	79	5	22	−0.0152
0.1	1.8	100	0.6527	14	23	4	13	0.255
	2.0	100	0.9342	16	28	2	13	0.1806
	2.0	300	1.1169	22	37	5	12	0.309
	2.6	300	1.6435	6	9	2	11	0.0184
	2.0	1000	1.3857	14	22	4	11	−0.0767

of w. Then we choose appropriate level of ρ and w in view of the new efficient frontier calculated by a new set of 36 monthly data and solve the minimal cost rebalancing problem (28).

Figure 11 shows the average and the standard deviation of the computation time required for rebalancing after the elapse of months when $\alpha_j = 0.05\,(j = 1, \cdots, n)$. As expected, the performance of the algorithm is more or less the same as the that of portfolio construction problem reported in Sect. 6.1. It turns out that we sell/buy smaller number of assets when the terminal risk is the same as the original portfolio and the elapse of time is short enough. However, it increases as the discrepancy of ρ and w from the original portfolio increases.

We also compared alternative rebalancing strategies in terms of net ex-post performance of the portfolio. *i.e.* nominal return subtracted by transaction costs. Frequent rebalancing is associated with smaller transaction cost per rebalance. However, frequent rebalancing results in a larger total transaction costs. How frequent should we rebalance?

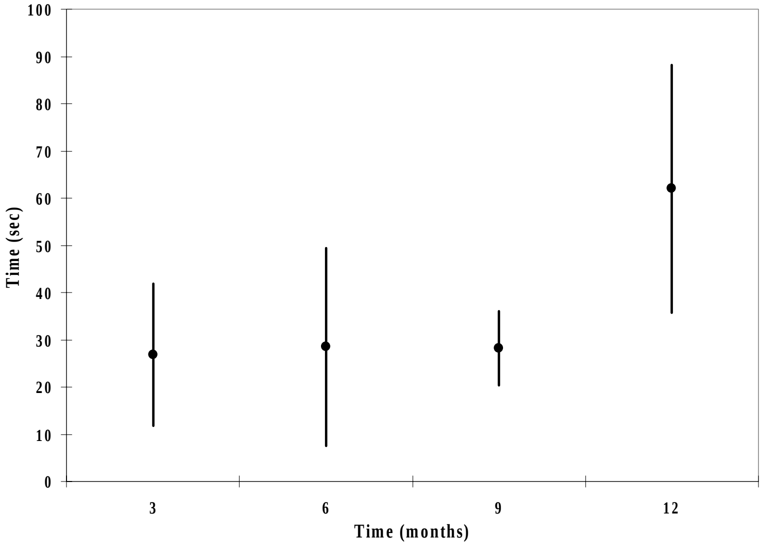

Fig. 11. CPU time for rebalancing

This is a very important problem from the practical point of view. However, we will not go into detail since it will be of little interest of mathematical programmers. More detailed analysis of rebalancing strategies will be reported in a subsequent article.

7. Conclusions

We showed in this paper that the portfolio construction/rebalancing problem under concave transaction costs can be solved in a practical amount of time. The success depends upon the use of mean absolute deviation model, elaboration of the classical branch and bound method using ω - subdivision strategy and the problem reduction strategy using the special structure of the problem. Let us emphasize that there are still a number of difficult (nonconvex) minimization problems in the field of financial optimization, some of which may be solved successfully by applying algorithm developed in the various fields of the mathematical programming.

Acknowledgements. This research was supported by Grant-in-Aid for Scientific Research B(2)(10450041) and B(2)(11558046) of the Ministry of Education, Science, and Culture. Also the first author acknowledges the generous support of IBJ-DL Financial Technologies, Co. and The Toyo Trust and Banking, Co., Ltd.

References

1. Chvátal, V. (1983): Linear programming. Freeman and Co.
2. Elton, J.E., Gruber, M.J. (1991): Modern Portfolio Theory and Investment Analysis, 4th edn. John Wiley & Sons
3. Falk, J.E., Hoffman, K.L. (1976): A successive underestimation method for concave minimization problem. Math. Oper. Res. **1**, 251–259

4. Falk, J.E., Soland, R.M. (1969): An algorithm for separable nonconvex programming problems. Manage. Sci. **15**, 550–569
5. Horst, R., Pardalos, P.M. (1995): Handbook of Global Optimization. Kluwer Academic Publishers
6. Konno, H., Suzuki, T., Kobayashi, D. (1998): A branch and bound algorithm for solving mean-risk-skewness model. Optim. Methods Software **10**, 297–317
7. Konno, H., Suzuki, K. (1995): A mean variance skewness portfolio optimaization model. J. Oper. Res. Soc. Japan **38**, 173–187
8. Konno, H., Thach, P.T., Tuy, H. (1997): Optimization on Low Rank Nonconvex Structures. Kluwer Academic Publishers
9. Konno, H., Yamazaki, H. (1991): Mean absolute deviation portfolio optimization model and its application to Tokyo stock market. Manage. Sci. **37**, 519–531
10. Konno, H., Wijayanayake, A. (1999): Mean-absolute deviation portfolio optimization model under transaction costs. J. Oper. Res. Soc. Japan **42**, 422–435
11. Kuno, T. (1999): A finite branch-and-bound algorithm for linear multiplicative programming. ISE-TR-99-159, Institute of Information Science and Electronics, University of Tsukuba
12. Markowitz, H.M. (1959): Portfolio Selection: Efficient Diversification of Investments. John Wiley & Sons
13. Mulvey, J.M. (1993): Incorporating transaction costs in models for asset allocation. In: Zenios, S. et al., eds., Financial Optimization. Cambridge University Press, pp. 243–259
14. Ogryczak, W., Ruszczynski, A. (1999): From stochastic dominance to mean-risk model. Eur. J. Oper. Res. **116**, 33–50
15. Perold, A.F. (1984): Large scale portfolio optimization. Manage. Sci. **30**, 1143–1160
16. Phong, T.Q., An, L.T.H., Tao, P.D. (1995): Decomposition branch and bound method for globally solving linearly constrained indefinite quadratic minimization problems. Oper. Res. Lett. **17**, 215–220
17. Tuy, H. (1998): Convex Analysis and Global Optimization. Kluwer Academic Publishers