

A FAST ALGORITHM FOR SOLVING LARGE SCALE MEAN-VARIANCE MODELS BY COMPACT FACTORIZATION OF COVARIANCE MATRICES

Hiroshi Konno Ken-ichi Suzuki
Tokyo Institute of Technology

(Received February 28, 1991; Revised October, 4, 1991)

Abstract A fast algorithm for solving large scale MV (mean-variance) portfolio optimization problems is proposed. It is shown that by using T independent data representing the rate of return of the assets, the MV model consisting of n assets can be put into a quadratic program with $n + T$ variables, T linear constraints and T quadratic terms in the objective function. As a result, the computation time required to solve this problem would increase very mildly as a function of n . This implies that a very large scale MV model can now be solved in a practical amount of time.

1. Introduction

Large-scale portfolio optimization is attracting more attention in recent years. In fact, many analysts are interested in solving Markowitz's MV (mean-variance) model consisting of more than one thousand assets to construct a portfolio with minimal risk or whose performance is as close as possible to that of a given index.

This makes a sharp contrast to the situation before early eighties, when people comfortably relied upon equilibrium relations implied by CAPM (capital asset pricing model) to construct a portfolio. It should be noted that CAPM was derived from the MV model by imposing several idealistic assumptions (See [5, 18] for details). One of the reasons why CAPM gained an unprecedented popularity is that the associated computational work is much less than that of solving a quadratic programming problem. Also, it was believed that the linear relations between the expected rate of return of individual assets and the market portfolio hold at least as a first order approximation. However, it was revealed through the detailed statistical analysis [6] that this relation is quite unstable as far as the stock data in Tokyo Stock Exchange are concerned. This embarrassing discovery motivated fund managers to look for more reliable models. Multi-factor (or multi-index) models and arbitrage pricing models served for this purpose to a large extent. However, they usually require considerably more computational work than CAPM.

The revival of Markowitz's model took place in early eighties with the declining popularity of CAPM and with the appearance of the path-breaking paper by Markowitz and Perold [12], in which they demonstrated that a large scale MV model can be solved efficiently. The most important aspect of their approach is the use of a multi-factor (index) model or a scenario model to generate a sparse representation together with sparse matrix techniques for solving large scale quadratic programming problems. Let us note that the computation time for solving quadratic programs depends upon the density, i.e., the percentage of nonzero elements included in the model as well as the number of variables. Since covariance matrix is usually completely dense, a large scale MV model in its original form requires a prohibitive amount of time. On the contrary, it was shown by Perold[14] that the use of factor models enables one to derive a sparse representation of the quadratic

program, for which efficient solution techniques are available [13]. Several additional features such as the appropriate handling of transaction/illiquidity costs helped his approach gain more popularity.

The purpose of this paper is to develop an even more compact representation of the MV model. This representation contains fewer quadratic terms in its objective function. As a result, the computation time for solving MV model using this new representation is expected to increase very mildly as a function of n , the number of assets. Thus we would be able to solve a very large scale portfolio optimization problem consisting of several thousand assets within a practical amount of time.

In section 2, we will summarize some of the previous efforts for solving large scale portfolio optimization problems. Section 3 will be devoted to an alternative representation of the MV model which is more amenable to sparse matrix techniques. In section 4, we will report the results of the preliminary computational experiments.

2. Review of Large Scale Portfolio Optimization

Let R_j be a random variable representing the rate of return (per period) of asset S_j , $j = 1, \dots, n$. Also, let x_j be the rate of money to be invested in S_j out of the total fund (normalized to 1).

The classical MV (mean-variance) model [11, 10] is defined as follows:

$$(2.1) \quad \left\{ \begin{array}{l} \text{minimize} \quad E[(\sum_{j=1}^n R_j x_j - E[\sum_{j=1}^n R_j x_j])^2] \\ \text{subject to} \quad E[\sum_{j=1}^n R_j x_j] \geq \rho \\ \sum_{j=1}^n x_j = 1 \\ x_j \geq 0, \quad j = 1, \dots, n \end{array} \right.$$

where $E[\cdot]$ stands for the expected value of the random variable in the bracket and ρ is a parameter representing the minimal required rate of return out of the portfolio.

Let us denote

$$(2.2) \quad r_j = E[R_j], \quad j = 1, \dots, n$$

$$(2.3) \quad \sigma_{ij} \equiv \text{cov}[R_i, R_j], \quad i = 1, \dots, n; \quad j = 1, \dots, n$$

Then (2.1) can be written as a convex quadratic programming problem:

$$(2.4) \quad \left\{ \begin{array}{l} \text{minimize} \quad \sum_{i=1}^n \sum_{j=1}^n \sigma_{ij} x_i x_j \\ \text{subject to} \quad \sum_{j=1}^n r_j x_j \geq \rho \\ \sum_{j=1}^n x_j = 1 \\ x_j \geq 0, \quad j = 1, \dots, n \end{array} \right.$$

If n is small, then this problem can be solved by any one of the standard algorithms [1, 3, 21, 22].

When n is larger than one thousand, most of these algorithms would require a prohibitive amount of computation. This is due to the facts that σ_{ij} 's are usually nonzero for all i, j 's

and that the number of arithmetic operations to solve (2.4) depends upon the number of non-zero coefficients contained in the model as well as the number of variables.

In the early sixties, several authors tried to alleviate this computational difficulty by employing linear approximation scheme[16, 17, 20]. Also, a single factor model[19] enabled one to reduce the computation time significantly. These efforts turned out to be successful at least for small scale problems. Unfortunately, research for extending these results to large scale problems were set aside until early eighties because of the widespread popularity of CAPM established in the mid 60's.

In 1981, A.Perold, with the collaboration of H.Markowitz[12], obtained an alternative representation of a quadratic programming problem by using a multi-factor model.

The key idea was to extend the single factor approach of the 60's [19] to the multi-factor (index) approach. He assumed the relation

$$(2.5) \quad R_i = \alpha_i + \beta_{i1}F_1 + \dots + \beta_{iK}F_K + \varepsilon_i, \quad i = 1, \dots, n$$

where F_k is the k -th random factor; α_i, β_{ij} are constants; ε_i is a random disturbance whose mean is zero and which is uncorrelated with F_k ($k = 1, \dots, K$) and ε_j ($j \neq i$).

Theorem 2..1 Let $\sigma_i^2 = E[\varepsilon_i^2]$, $f_{rs} = \text{cov}[F_r, F_s]$, then the relation (2.5) leads to the following expression

$$(2.6) \quad \sum_{i=1}^n \sum_{j=1}^n \sigma_{ij} x_i x_j = \sum_{i=1}^n \sigma_i^2 x_i^2 + \sum_{i=1}^n \sum_{r=1}^K \sum_{s=1}^K f_{rs} \beta_{ir} \beta_{is} x_i x_j$$

Proof. Can be proved by simple substitution. \square

From this, we obtain an alternative representation of (2.4):

$$(2.7) \quad \left. \begin{array}{l} \text{minimize} \quad \sum_{i=1}^n \sigma_i^2 x_i^2 + \sum_{r=1}^K \sum_{s=1}^K f_{rs} y_r y_s \\ \text{subject to} \quad \sum_{j=1}^n r_j x_j \geq \rho \\ \sum_{j=1}^n \beta_{jk} x_j - y_k = 0, \quad k = 1, \dots, K \\ \sum_{j=1}^n x_j = 1 \\ x_j \geq 0, \quad j = 1, \dots, n \end{array} \right\}$$

Usually, K is much less than n , so that (2.7) can be solved efficiently by using sparse matrix techniques[13, 14]. Also, it is argued in [14] that the matrix

$$B = (\beta_{jk}) \in R^{n \times K}$$

is sparse when K is large.

Additional feature of Perold's model is the introduction of transaction costs $c_j(x_j)$ associated with purchasing x_j units of the asset S_j . Also, he allows for the introduction of

linear institutional constraints. The resulting problem is given as follows:

$$(2.8) \quad \left\{ \begin{array}{l} \text{minimize} \quad \sum_{i=1}^n \sigma_i^2 x_i^2 + \sum_{r=1}^K \sum_{s=1}^K f_{rs} y_r y_s \\ \text{subject to} \quad \sum_{j=1}^n (r_j x_j - c_j(x_j)) \geq \rho \\ \sum_{j=1}^n \beta_{jk} x_j - y_k = 0, \quad k = 1, \dots, K \\ \sum_{j=1}^n x_j = 1 \\ \sum_{j=1}^n a_{ij} x_j \geq b_i, \quad i = 1, \dots, m \\ l_j \leq x_j \leq u_j, \quad j = 1, \dots, n \end{array} \right.$$

where the last two inequalities correspond to the institutional constraints.

The "Optimizer" code, a software developed by Perold is now widely used by practitioners. According to one of the frequent users, this code is very efficient once the representation (2.8) is given. For example, a problem (2.8) with $n=1,192$, $m \approx 10$, $K=4$ can be solved in less than a minute on VAX 8700 system when $c_j(\cdot) \equiv 0$, $j = 1, \dots, n$ and $f_{rs} = 0$, ($r \neq s$).

There are, however several problems associated with this approach when we consider a very large scale model consisting of more than a few thousand assets. First, we have to introduce as many as one hundred factors to obtain a good statistical fitting, which is very time consuming and somewhat dubious. In fact, it takes several hours to generate the problem (2.8) of the size referred to above. Second, though sparse, the problem (2.8) contains n quadratic terms. Thus it would not be easy to solve this problem when n is over a few thousand. Third, the optimal solution of (2.8) usually contains many positive x variables. Thus we need to spend some time eliminating a portion of variables to organize a portfolio which consists of a manageable number of assets, i.e., a portfolio with less than, say two hundred assets.

Another recent approach for solving a large scale portfolio optimization problem is the MAD (mean-absolute-deviation) model [7, 8], in which the standard deviation in (2.1) is replaced by the absolute deviation of the rate of return of the portfolio:

$$(2.9) \quad \left\{ \begin{array}{l} \text{minimize} \quad E\left[\left| \sum_{j=1}^n R_j x_j - E\left[\sum_{j=1}^n R_j x_j \right] \right| \right] \\ \text{subject to} \quad E\left[\sum_{j=1}^n R_j x_j \right] \geq \rho \\ \sum_{j=1}^n x_j = 1 \\ x_j \geq 0, \quad j = 1, \dots, n \end{array} \right.$$

It has been proved in [7] that both (2.1) and (2.9) lead to the same optimal solution if (R_1, \dots, R_n) is multi-variate normally distributed.

Let r_{jt} be the realization of the random variable R_j at time period t . Then a very straightforward approximation of the objective function of (2.9) is given by

$$(2.10) \quad \frac{1}{T} \sum_{t=1}^T \left| \sum_{j=1}^n (r_{jt} - \bar{r}_j) x_j \right|$$

Thus, instead of a quadratic program, it suffices to solve

$$(2.11) \quad \left\{ \begin{array}{l} \text{minimize} \quad \sum_{t=1}^T \left| \sum_{j=1}^n (r_{jt} - r_j) x_j \right| \\ \text{subject to} \quad \sum_{j=1}^n r_j x_j \geq \rho \\ \quad \quad \quad \sum_{j=1}^n x_j = 1 \\ \quad \quad \quad x_j \geq 0, \quad j = 1, \dots, n \end{array} \right.$$

Standard techniques can be used to convert this problem into a linear program:

$$(2.12) \quad \left\{ \begin{array}{l} \text{minimize} \quad \sum_{t=1}^T y_t \\ \text{subject to} \quad y_t - \sum_{j=1}^n (r_{jt} - r_j) x_j \geq 0, \quad t = 1, \dots, T \\ \quad \quad \quad y_t + \sum_{j=1}^n (r_{jt} - r_j) x_j \geq 0, \quad t = 1, \dots, T \\ \quad \quad \quad \sum_{j=1}^n r_j x_j \geq \rho \\ \quad \quad \quad \sum_{j=1}^n x_j = 1 \\ \quad \quad \quad x_j \geq 0, \quad j = 1, \dots, n \end{array} \right.$$

It has been shown in [8] that the portfolios generated by (2.12) and (2.4) using the same data are quite similar to each other. Some of the advantages of the MAD model (2.12) are:

- (a) It can be solved much faster than its counterparts (2.4) and (2.7) since it is a linear program.
- (b) Its optimal solution contains no more than $T+2$ assets with positive weight.
- (c) It can incorporate all the features included in (2.8).

3. A Compact Representation of the MV Model

In this section, we will derive a compact representation of the MV model, which is amenable to efficient implementation for large scale problems.

Let (r_{1t}, \dots, r_{nt}) , $t = 1, \dots, T$, be T independent samples of the random variable (R_1, \dots, R_n) . Also, let

$$(3.1) \quad \hat{r}_i = \frac{1}{T} \sum_{t=1}^T r_{it}, \quad i = 1, \dots, n$$

$$(3.2) \quad \hat{\sigma}_{ij} = \frac{1}{T} \sum_{t=1}^T (r_{it} - \hat{r}_i)(r_{jt} - \hat{r}_j), \quad i = 1, \dots, n; j = 1, \dots, n$$

Theorem 3.1 \hat{r}_i is an unbiased estimate of r_i . Also, $\hat{\sigma}_{ij}$ is an unbiased estimate of σ_{ij} .

Proof. See [15] □

Let us substitute σ_{ij} of (2.4) by its unbiased estimate $\hat{\sigma}_{ij}$. Then we obtain the following expression.

$$\begin{aligned}
 (3.3) \quad \sum_{i=1}^n \sum_{j=1}^n \hat{\sigma}_{ij} x_i x_j &= \sum_{i=1}^n \sum_{j=1}^n \left\{ \frac{1}{T} \sum_{t=1}^T (r_{it} - \hat{r}_i)(r_{jt} - \hat{r}_j) \right\} x_i x_j \\
 &= \frac{1}{T} \sum_{t=1}^T \sum_{i=1}^n \sum_{j=1}^n (r_{it} - \hat{r}_i) x_i (r_{jt} - \hat{r}_j) x_j \\
 &= \frac{1}{T} \sum_{t=1}^T \left\{ \sum_{i=1}^n (r_{it} - \hat{r}_i) x_i \right\}^2
 \end{aligned}$$

Let us introduce auxiliary variables

$$(3.4) \quad z_t = \sum_{i=1}^n (r_{it} - \hat{r}_i) x_i, \quad t = 1, \dots, T$$

We then have an alternative representation of the problem (2.4):

$$(3.5) \quad \left\{ \begin{array}{l} \text{minimize} \quad \sum_{t=1}^T z_t^2 \\ \text{subject to} \quad \sum_{j=1}^n \hat{r}_j x_j \geq \rho \\ z_t - \sum_{j=1}^n (r_{jt} - \hat{r}_j) x_j = 0, \quad t = 1, \dots, T \\ \sum_{j=1}^n x_j = 1 \\ x_j \geq 0, \quad j = 1, \dots, n \end{array} \right.$$

We can certainly add transaction/illiquidity cost and linear institutional constraints to obtain a problem corresponding to (2.8):

$$(3.6) \quad \left\{ \begin{array}{l} \text{minimize} \quad \sum_{t=1}^T z_t^2 \\ \text{subject to} \quad \sum_{j=1}^n \{ \hat{r}_j x_j - c_j(x_j) \} \geq \rho \\ z_t - \sum_{j=1}^n (r_{jt} - \hat{r}_j) x_j = 0, \quad t = 1, \dots, T \\ \sum_{j=1}^n x_j = 1 \\ \sum_{j=1}^n a_{ij} x_j \geq b_i, \quad i = 1, \dots, m \\ l_j \leq x_j \leq u_j, \quad j = 1, \dots, n \end{array} \right.$$

Let us discuss some of the advantages of (3.5) over (2.7) and (2.4) when we use historical data of the stock market. The reason why we consider historical data is that it is easy to access and that the direct use of historical data is claimed to be as good as other indirect statistical methods for estimating covariance coefficients [9].

The first advantage of (3.5) is that it can be constructed automatically, i.e., without any pre-processing of the data as opposed to the case of (2.4) or (2.7). The second advantage is that the objective function of (3.5) consists of only T quadratic terms, where T is usually less than one hundred. (2.7) and (2.4), on the other hand, contains $n + T^2$ and n^2 quadratic terms, respectively in its objective function. Hence the computation time for solving (3.5) would be significantly less than those of (2.7) and (2.4) when n is large.

Several additional comments are in order.

(a) If (3.5) has a feasible solution, then it has an optimal solution such that at most $T+2$ components of x_j 's are positive. To see this, let $(x_1^*, \dots, x_n^*, z_1^*, \dots, z_T^*)$ be an optimal solution and let us consider the linear system:

$$(3.7) \quad \left\{ \begin{array}{l} \sum_{j=1}^n a_{jt} x_j = z_t^*, \quad t = 1, \dots, T \\ \sum_{j=1}^n \hat{r}_j x_j \geq \rho \\ \sum_{j=1}^n x_j = 1 \\ x_j \geq 0, \quad j = 1, \dots, n \end{array} \right.$$

This system has a feasible solution and thus it has a basic feasible solution $(\hat{x}_1, \dots, \hat{x}_n)$ with only at most $T+2$ positive components. It is easy to see that $(\hat{x}_1, \dots, \hat{x}_n, z_1^*, \dots, z_T^*)$ is also an optimal solution of (3.5).

(b) We can apply standard piecewise linear approximation of a separable quadratic function for solving (3.5). In particular, if we replace $\sum_{t=1}^T z_t^2$ by $\sum_{t=1}^T |z_t|$, then we would have the problem (2.12) since minimizing $\sum_{t=1}^T |z_t|$ where

$$(3.8) \quad z_t = \sum_{j=1}^n (r_{jt} - \hat{r}_j) x_j, \quad t = 1, \dots, T$$

is equivalent to minimizing $\sum_{t=1}^T y_t$ under the constraint

$$(3.9) \quad \left\{ \begin{array}{l} y_t \geq \sum_{j=1}^n (r_{jt} - \hat{r}_j) x_j, \quad t = 1, \dots, T \\ y_t \geq -\sum_{j=1}^n (r_{jt} - \hat{r}_j) x_j, \quad t = 1, \dots, T \end{array} \right.$$

as is well known in linear programming (2.4).

This, together with (a), explains the remarkable similarities of the portfolio generated by MV and MAD models when we use the same set of data.

(c) (3.5) can be easily adapted to index matching problem:

$$(3.10) \quad \left\{ \begin{array}{l} \text{minimize} \quad E[(\sum_{j=1}^n R_j x_j - R_I)^2] \\ \text{subject to} \quad \sum_{j=1}^n \hat{r}_j x_j \geq \rho \\ \sum_{j=1}^n x_j = 1 \\ x_j \geq 0, \quad j = 1, \dots, n \end{array} \right.$$

where R_I is the rate of return of the index I. To do this, we simply replace \hat{r}_i of (3.4) by the realization of R_I at time period t .

4. Computational Experiments

Let us report some of the results of preliminary numerical experiments on the model (3.5) using two sets of historical data D_1 and D_2 which consist of 60 monthly data of Tokyo Stock Exchange from January '85 through December '89 and January '86 through December '90, respectively. First we compared the model (3.5) with the classical one (2.4) using the "LINDO" package installed on VAX 8700 system. This software computes the Karush-Kuhn-Tucker point of a quadratic programming problem by converting it to a linear complementarity problem.

The maximum size n of the problem solvable by using formulation (2.4) was 400 due to the memory restriction. On the other hand, we could solve problems of size up to $n=800$ by using the representation (3.5) in less than five minutes. Table 4.1 shows the number of pivoting operations required to solve (3.5).

Table 4.1

n	100	200	300	400	500	600	700	800
# of pivots	435	593	492	545	652	864	685	798

We see from this that the number of pivoting and thus, the computation time increases mildly as a function of the size of the problems. Both (2.4) and (3.5) generated identical portfolios up to $n=400$ in almost the same computation time.

To solve larger problems, we have to develop a special purpose algorithm by exploiting the special structure of the problem (3.5). We are now conducting research in this direction, whose results will be reported subsequently.

In the meantime, we applied piecewise linear approximation commonly used to convert a separable programming problem into a linear programming problem (See [4]). For this purpose, we choose L mesh points s_1, s_2, \dots, s_L where

$$(4.1) \quad s_1 = \min\{r_{jt} - \hat{r}_j | j = 1, \dots, n; t = 1, \dots, T\}$$

$$(4.2) \quad s_L = \max\{r_{jt} - \hat{r}_j | j = 1, \dots, n; t = 1, \dots, T\}$$

and approximate a quadratic function z_t^2 by a piecewise linear convex function by introducing L auxiliary variables ξ_{lt} ($l = 1, \dots, L; t = 1, \dots, T$):

$$(4.3) \quad z_t = \sum_{l=1}^L \xi_{lt} s_l$$

$$(4.4) \quad z_t^2 \cong \sum_{l=1}^L \xi_{tl} s_l^2$$

where

$$(4.5) \quad \sum_{l=1}^L \xi_{tl} = 1, \quad \xi_{tl} \geq 0, \quad l = 1, \dots, L$$

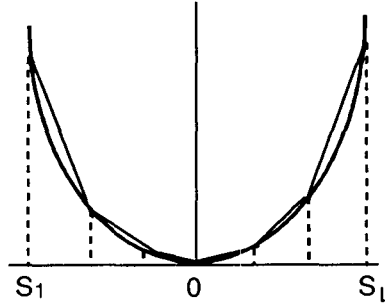


Fig.4.1

The resulting linear programming problem is given as follows:

$$(4.6) \quad \left\{ \begin{array}{l} \text{minimize} \quad \sum_{t=1}^T \sum_{l=1}^L \xi_{tl} s_l^2 \\ \text{subject to} \quad \sum_{l=1}^L \xi_{tl} s_l - \sum_{j=1}^n (r_{jt} - r_j) x_j = 0, \quad t = 1, \dots, T \\ \sum_{j=1}^n r_j x_j \geq \rho \\ \sum_{j=1}^n x_j = 1 \\ \sum_{l=1}^L \xi_{tl} = 1, \quad t = 1, \dots, T \\ x_j \geq 0, \quad j = 1, \dots, n \\ \xi_{tl} \geq 0, \quad l = 1, \dots, L; t = 1, \dots, T \end{array} \right.$$

This system consists of $n+T \cdot L$ variables and $2T+2$ constraints.

We approximated a quadratic function by choosing 13 mesh points listed in Table 4.2.

Table 4.2

s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}
-20	-10	-5	-3	-2	-1	0	1	2	3	5	10	20

Table 4.3 and 4.4 show the optimal portfolios generated by MV model (3.5), the PL (piecewise linear) model (4.6) and by MAD model (2.12) for the data sets D_1 and D_2 , respectively. The universe is 225 stocks included in NIKKEI 225 Index.

Table 4.3 Portfolio for D_1 ($\rho=2.0$)

Weight(%) and ranking				Weight(%) and ranking			
Code#	MV	PL	MAD	Code#	MV	PL	MAD
6	3.21	2.15	1.97	111	0.74	1.35	0.45
10	2.97	2.32	1.55	118	1.37	2.36	—
11	6.52 (4)	8.13 (2)	5.87 (6)	134	16.40 (1)	15.93 (1)	9.80 (1)
14	5.38 (5)	5.67 (6)	6.99 (4)	135	0.69	1.72	1.77
18	1.81	2.87	3.91	146	5.04 (6)	6.29 (4)	6.70 (5)
19	1.87	2.07	1.82	150	6.60 (3)	4.80 (7)	—
22	—	—	0.63	157	1.06	1.11	4.25 (10)
28	1.03	1.81	1.91	160	1.55	1.47	—
29	1.21	1.54	1.85	162	4.98 (7)	6.13 (5)	7.16 (3)
35	0.52	2.37	—	167	3.37 (10)	4.65 (8)	9.30 (2)
37	—	—	2.60	168	1.85	—	—
44	2.28	0.43	—	169	—	—	0.35
61	—	—	0.27	170	—	—	3.78
69	1.19	0.93	1.73	183	3.04	2.05	2.14
78	1.98	1.93	4.67 (8)	206	3.11	2.92 (10)	0.60
79	4.31 (9)	2.89	—	213	—	—	0.22
82	—	—	1.53	217	0.86	1.08	—
85	4.80 (8)	3.60 (9)	1.11	218	—	—	5.69 (7)
107	0.14	0.86	2.60	219	7.30 (2)	7.36 (3)	4.53 (9)
108	—	—	0.28	221	—	—	0.99
109	—	—	0.54	224	1.46	1.20	0.41

Table 4.4 Portfolio for D_2 ($\rho=2.0$)

Weight(%) and ranking				Weight(%) and ranking			
Code#	MV	PL	MAD	Code#	MV	PL	MAD
6	1.98	2.72	—	103	2.47	3.41 (10)	0.85
10	4.62 (8)	4.62 (8)	—	104	—	—	3.62 (9)
11	1.31	0.92	2.64	108	—	—	2.21
18	—	—	0.80	124	0.66	1.29	—
19	—	—	0.90	134	9.49 (4)	8.64 (5)	11.27 (3)
24	0.31	—	—	135	1.28	0.34	4.17 (7)
28	—	—	2.67	140	0.92	—	—
30	0.66	1.79	—	150	24.18 (1)	25.04 (1)	19.72 (1)
44	4.85 (6)	5.46 (6)	2.75	157	—	—	1.70
53	15.16 (2)	14.37 (2)	18.00 (2)	161	0.82	—	—
78	10.58 (3)	9.73 (3)	5.00 (5)	183	7.68 (5)	8.82 (4)	3.68 (8)
79	—	—	5.34 (4)	184	—	—	2.30
80	4.73 (7)	4.70 (7)	1.71	219	3.40 (9)	3.53 (9)	4.53 (6)
85	0.82	0.90	2.72	220	1.59	0.48	3.38 (10)
95	2.48 (10)	3.27	—				

We see from these tables that the optimal portfolio generated by a piecewise linear approximation (4.4) of a quadratic function (with 13 pieces of linearity) is similar to the MV portfolio. In particular, nine out of top ten stocks in terms of their weights are the

same for both portfolios. We would certainly be able to obtain a more accurate solution by successively refining the piecewise linear approximation of quadratic functions near the approximate optimal solution as is commonly done in separable programming.

The difference between MV portfolio and MAD portfolio is larger. This is natural since MAD can be viewed as a piecewise linear approximation with only two linear pieces instead of 13 pieces of PL portfolio.

Minimal values of the objective functions associated with optimal MV and PL portfolios are 228.8 and 239.5, respectively for the data set D_1 . On the other hand, they are 784.1 and 1253.3, respectively for the data set D_2 . This big difference is due to the very wild movement of stock prices in the year 1990. In fact, we experienced several big jumps of stock prices in the year 1990. The crude piecewise linear approximation for larger values of z_i 's accounts for the big difference. To obtain a better approximation, we need to apply successive refinement procedure referred to above. Finally, Table 4.5 show the number of simplex iterations required to solve the problem (4.6) when $L = 13$.

Table 4.5

n	# of iterations	# of assets with positive weights
1192	1239	58
1500	651	62
1800	573	61
2384	454	61

The reason for the decrease of the number of simplex iterations as a function of n is that we could attain the lower bound zero of the objective function for n greater than 1,500. This table shows that we are now able to solve a very large scale MV model efficiently for up to a few thousand variables. Unfortunately, however we cannot report the exact CPU time since the LINDO software installed on VAX system does not provide information about the CPU time.

We are now developing a software which automatically takes care of this successive refinement procedure as well as parametric right hand side procedure. The result of the experiments using this code will be reported subsequently.

References

- [1] E. M. L. Beale, "On quadratic programming," *Naval Research Logistics Quarterly*, 6:227–243, 1959.
- [2] V. Chvátal, *Linear Programming*. Freeman and Co., 1983.
- [3] R. W. Cottle, "Monotone solutions of the parametric linear complementarity problem," *Mathematical Programming*, 3:210–224, 1972.
- [4] G. B. Dantzig, *Linear Programming and Extensions*. Princeton University Press, 1963.
- [5] E. J. Elton and M. J. Gruber, *Modern Portfolio Theory and Investment Analysis (3rd Edition)*. John Wiley & Sons, Inc., 1987.
- [6] T. Kariya et al, *Distribution of Stock Prices in the Stock Market of Japan*. Toyo Keizai Publishing Co. (in Japanese), 1989.

- [7] H. Konno, "Piecewise linear risk functions and portfolio optimization," *Journal of the Operations Research Society of Japan*, 33:139–156, 1990.
- [8] H. Konno and H. Yamazaki, "A mean-absolute deviation portfolio optimization model and its application to tokyo stock market," *Management Science*, 37:519–531, 1991
- [9] H. Markowitz, Lecture delivered at the Tokyo Institute of Technology, Dec. 18, 1990.
- [10] H. Markowitz, *Mean-Variance Analysis in Portfolio Choice and Capital Markets*. Basil Blackwell, 1987.
- [11] H. Markowitz, *Portfolio Selection: Efficient Diversification of Investments*. John Wiley & Sons, 1959.
- [12] H. Markowitz and A. Perold, "Sparsity and piecewise linearity in large scale portfolio optimization problems," In *Sparse Matrices and Their Uses*, 89–108, I.S. Duff (ed.) Academic Press, New York, 1981.
- [13] J. S. Pang, "A new and efficient algorithm for a class of portfolio selection problems," *Operations Research*, 28:754–767, 1980.
- [14] A. Perold, "Large scale portfolio optimizations," *Management Science*, 30:1143–1160, 1984.
- [15] C. R. Rao, *Linear Statistical Inference and Its Applications (2nd ed.)*. John Wiley & Sons, 1965.
- [16] W. F. Sharpe, "A linear programming algorithm for a mutual fund portfolio selection," *Management Science*, 13:499–510, 1967.
- [17] W. F. Sharpe, "A linear programming approximation for the general portfolio selection problem," *Journal of Financial and Quantitative Analysis*, 6:1263–1275, 1971.
- [18] W. F. Sharpe, *Portfolio Theory and Capital Market*. McGraw Hill., 1970.
- [19] W. F. Sharpe, "A simplified model for portfolio analysis," *Management Science*, 9:277–293, 1963.
- [20] B. K. Stone, "A linear programming formulation of the general portfolio selection problem," *Journal of Financial and Quantitative Analysis*, 8:621–636, 1973.
- [21] B. Von Hohenbalken, "A finite algorithm to maximize certain pseudo-concave functions on polytopes," *Mathematical Programming*, 9:189–206, 1975.
- [22] P. Wolfe, "The simplex method for quadratic programming," *Econometrica*, 27:382–398, 1959.

Hiroshi KONNO:

Institute of Human and Social Sciences,
Tokyo Institute of Technology,
Oh-okayama, Meguro-ku, Tokyo,
152, Japan.