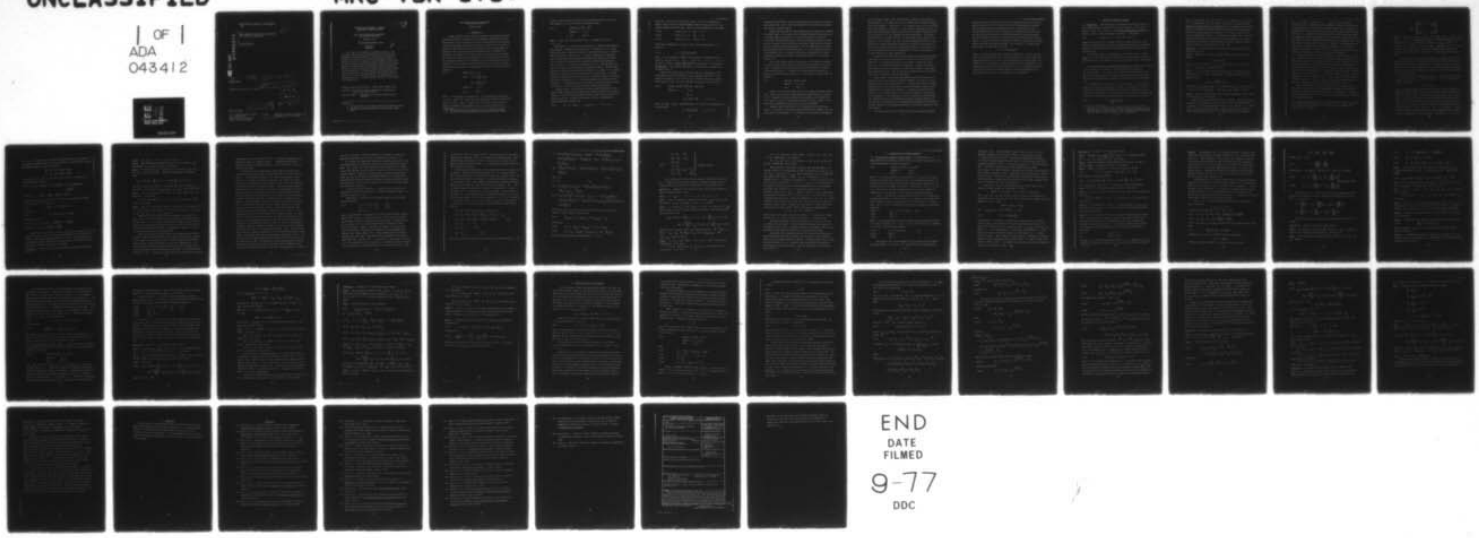
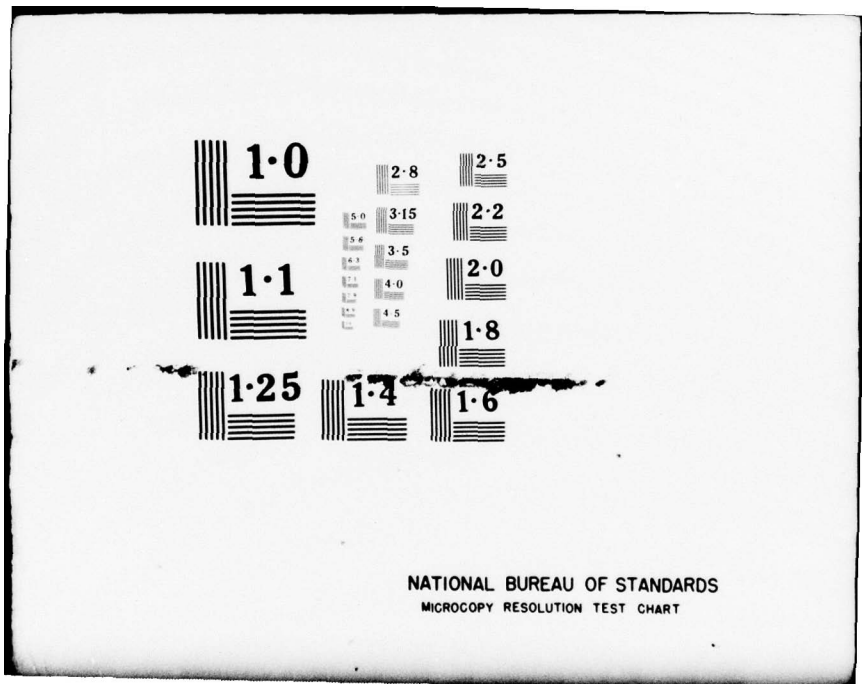
AD-A043 412 WISCONSIN UNIV MADISON MATHEMATICS RESEARCH CENTER F/G 9/2
SOME NEW AND EFFICIENT ALGORITHMS FOR PORTFOLIO ANALYSIS.(U)
MAR 77 J PANG DAAG29-75-C-0024
UNCLASSIFIED MRC-TSR-1738 NL

1 OF 1
ADA
043412



END
DATE
FILMED
9-77
DDC


1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8
3·15
3·5
4·0
4·5
5·0
5·6
NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

MRC Technical Summary Report, #1738

SOME NEW AND EFFICIENT ALGORITHMS FOR PORTFOLIO ANALYSIS.

Jong-Shi Pang

March 1977

48 p.

MRC-TSR-1738

DAAG29-75-C-0024
NSF-MCS-75-17385

(Received February 23, 1977)

Sponsored by

U. S. Army Research Office
P. O. Box 12211
Research Triangle Park
North Carolina 27709

and

National Science Foundation
Washington, D. C. 20550

UNIVERSITY OF WISCONSIN - MADISON
MATHEMATICS RESEARCH CENTER

SOME NEW AND EFFICIENT ALGORITHMS FOR PORTFOLIO ANALYSIS

Jong-Shi Pang

Technical Summary Report # 1738

March 1977

ABSTRACT

This paper first describes a modified version of the parametric principal pivoting algorithm [4] for the class of parametric linear complementarity problems with P-matrices. The modified version of the parametric principal pivoting algorithm is then used to develop a new and efficient algorithm to solve the class of portfolio analysis problems with positive definite (or equivalently, nonsingular) covariance matrices. Extension of the new algorithm to handle explicit upper-bounds is also established. The new algorithm and its extension are then specialized to the "index models" introduced by Sharpe [28], [29]. In these specializations, the algorithms are particularly effective, achieving dramatic savings in both storage and computations.

AMS(MOS) Subject Classification - 65K05, 90B99, 90C00, 90C50

Key Words - Portfolio analysis, index models, mathematical programming, factorization.

Work Unit Number 5 - Mathematical Programming and Operations Research

---

Sponsored by

1) the United States Army under Contract No. DAAG29-75-C-0024
2) the National Science Foundation under Grant No. MCS75-17385.

SOME NEW AND EFFICIENT ALGORITHMS FOR PORTFOLIO ANALYSIS

Jong-Shi Pang

1. INTRODUCTION

In 1952, Markowitz [19] suggested formulating the general portfolio analysis problem as a parametric (convex) quadratic programming problem. The quadratic term in the objective function corresponds to the predicted variance of portfolio return which is used as a measure of risk; the linear term in the objective function corresponds to the expected value of portfolio return; the parameter of the problem which is assumed to be nonnegative and appears in the linear term of the objective is a measure of the relative importance of reduced risk and increased return; finally, the constraints consist of the nonnegativity of the proportions of an investor's fund to be invested in the securities and the sum of the proportions being equal to unity. Specifically, the general portfolio analysis problem is

$$
\begin{aligned}
(1.1) \qquad & \text{minimize } \tfrac{1}{2} V_p - \theta E_p \\
& \text{with} \quad V_p = \mathrm{Var}\Big(\sum_{i=1}^{n} x_i R_i\Big) \\
& \qquad\quad E_p = \mathrm{Exp}\Big(\sum_{i=1}^{n} x_i R_i\Big) \\
& \text{subject to} \quad x \geq 0 \\
& \text{and} \quad\quad e^T x = 1
\end{aligned}
$$

where n is the number of securities; θ is the nonnegative parameter; $x = (x_1,\ldots,x_n)^T \in R^n$ is the vector of proportions invested in the securities; $R = (R_1,\ldots,R_n)^T$ is the vector of returns on the securities which are assumed to be stochastic; and e is the n-vector of 1's .

Sponsored by
1) the United States Army under Contract No. DAAG29-75- C-0024
2) the National Science Foundation under Grant No. MCS75- 17385.

Letting V be the covariance matrix of the security returns, we can rewrite problem (1.1) in the following equivalent format,

$$(1.2) \qquad \text{minimize} \quad \frac{1}{2} x^T V x + \theta r^T x$$

$$\text{subject to} \quad x \geq 0$$

$$\text{and} \quad e^T x = 1$$

where $r = (r_i) \in R^n$ is the vector with $r_i = -\text{Exp}(R_i)$ for every $i = 1,\ldots,n$.

The matrix V, being a covariance matrix, is symmetric and positive semi-definite. Therefore, for every fixed θ, problem (1.2) is a convex quadratic program. Methods for solving (1.2) parametrically (i.e. obtaining a solution for every value of θ) include, among others, Markowitz' critical line method [20] and Wolfe's simplex-like parametric algorithm [35]. Markowitz' method starts by finding a portfolio with maximum expected return (i.e. a solution corresponding to $\theta = \infty$), then decreases the value of θ successively and proceeds in parametric manner. The algorithm stops when it reaches the minimum risk portfolio (i.e. when $\theta = 0$). Wolfe's parametric algorithm operates in precisely the reverse order. It starts by using the unparametarized version of the algorithm to obtain an initial solution corresponding to $\theta = 0$ and then increases the value of θ until it reaches ∞. Both methods (and a few others) require the storage of the entire matrix V and an extensive amount of computational effort.

Recognizing the informational and computational complexity of the general model, Sharpe [28] introduced some simplified models for portfolio analysis. These are the "index models" which assume that the security returns are expressed in terms of some market indices. Specifically, a typical index model assumes that

$$(1.3) \qquad R_i = \alpha_i + \beta_{1i} I_1 + \ldots + \beta_{mi} I_m + \varepsilon_i \qquad i = 1,\ldots,n$$

where the α's and β's are constant scalars, the I's are appropriate market indices which are random variables with mean zero, and the ε's are random variables with mean zero. There are three assumptions to the model:

(1.4a) $\quad \mathrm{Cov}(I_i, I_j) = 0 \quad$ for $\quad i \neq j$

(1.4b) $\quad \mathrm{Cov}(\varepsilon_i, \varepsilon_j) = 0 \quad$ for $\quad i \neq j$

(1.4c) $\quad \mathrm{Cov}(\varepsilon_i, I_j) = 0 \quad$ for all $\quad i,j$.

Under these assumptions, as shown in [30], the covariance matrix V is given by

$$(1.5) \qquad V = \sum + \sum_{j=1}^{m} \mathrm{Var}(I_j)\beta_j\beta_j^T$$

where $\sum$ is a nonnegative diagonal matrix whose i-th diagonal entry is equal to $\mathrm{Var}(\varepsilon_i)$, and $\beta_j = (\beta_{j1},\ldots,\beta_{jn})^T$ for $j = 1,\ldots,m$. If $m = 1$, the model is called a <u>single index model</u>, and in general, it is called an <u>m-index model</u>. For practical purposes, it is reasonable to assume that m is smaller than n .

One of the simplifications resulting from the introduction of the index models was the transformation of the quadratic term x^TVx in the objective function of (1.2) into one with a diagonal matrix (i.e. into a sum of square terms). By (1.5), problem (1.2) can be cast in the form:

$$(1.6) \qquad \text{minimize} \quad \frac{1}{2}\begin{pmatrix} x \\ y \end{pmatrix}^T \begin{pmatrix} \Sigma & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \theta \begin{pmatrix} r \\ 0 \end{pmatrix}^T \begin{pmatrix} x \\ y \end{pmatrix}$$

$$\text{subject to} \qquad x \geq 0$$

$$e^Tx = 1$$

$$y_j = \mathrm{Var}(I_j)\ \beta_j^Tx \qquad j = 1,\ldots,m .$$

where $y = (y_1,\ldots,y_m)^T$. By treating the y's as separate variables and the equalities

$$y_j = \mathrm{Var}(I_j)\ \beta_j^Tx \qquad j = 1,\ldots,m$$

as additional constraints, special purpose quadratic programming codes have been written [12], [34] to take full advantage of the fact that the objective now has a diagonal matrix.

Sharpe [31] and Stone [32] have proposed another approach to simplify the general model. This involves first diagonalizing the covariance matrix V by some numerical procedure, e.g. the one described in Dantzig [7, p. 491-492], then approximating the resulting diagonalized objective function by a piecewise linear function, and finally, applying linear programming to solve the approximated problem. With the index models, the first step of diagonalizing the covariance matrix V is not required because of the formulation (1.6). Of course, the solution obtained by this linear programming approach gives only an approximated solution to the general problem (1.2).

Due to some legal or personal constraints, it is common in practice to impose explicit upper bounds on the proportions invested in the securities. An example of this was described in [29] concerning the mutual fund portfolio analysis. To formulate the general portfolio analysis problem with explicit upper bounds, let a be a positive vector, then the problem is

$$
\begin{aligned}
(1.7) \qquad & \text{minimize} && \tfrac{1}{2}\, x^T V x + \theta r^T x \\
& \text{subject to} && a \geq x \geq 0 \\
& \text{and} && e^T x = 1 \quad .
\end{aligned}
$$

Assuming a single index model and that the bounds are rather restrictive, Sharpe [29] proposed a linear programming approximation to problem (1.7). Although the solution procedure then becomes a very simple one, (as a matter of fact, the approximated problem is almost trivial) it produces only an approximate solution to the original problem.

Recently, Jucker and de Faro [13], assuming that the covariance matrix V is diagonal, described a simple algorithm for problem (1.2) for fixed value of the parameter θ and extended the algorithm to the problem

(1.7) with upper bounds. They also provided a graphical interpretation to their algorithm. As a matter of fact, the problem they considered, namely, (1.2) with diagonal V and fixed θ, is a special case of the single index model corresponding to $\mathrm{Var}(I_1) = 0$. More recently, Alexander [1] has reported some computational experience with the use of Lemke's algorithm [17] to solve problem (1.2) in a somewhat different formulation. The largest problem solved in the paper contains 90 securities and was solved in an average of 190.3 seconds (CPU time). Finally, Rosenberg and Rudd [24] has reported some rather encouraging computational results with the use of a new and general-purpose quadratic programming algorithm in portfolio analysis. The algorithm is due to Von Hohenbalken [33] and is not parametric. It uses techniques in linear programming.

Despite the fact that there are many quadratic programming algorithms that can be used to solve the problems (1.2) and (1.7), few of them are successful in solving problems with a large number of securities (say, when n is in the order of hundreds or thousands). A major limitation in the application of these algorithms to large-scale problems is the excessive computational cost. This is very often caused by the lack of efficiency in the implementation of the algorithms.

In this paper, we propose a fresh approach and develop some new algorithms to solve the classes of portfolio problems (1.2) and (1.7) with positive definite covariance matrices V. We will also discuss efficient implementation of these algorithms, with special emphasis placed on their implementation for large-scale problems. It should be pointed out that since V is a covariance matrix, it is positive definite if and only if it is nonsingular. The essential tool used to develop the new algorithms is a modified version of the parametric principal pivoting algorithm [4] for the class of parametric linear complementarity problems with P-matrices. We shall explain the terminology and describe the modified algorithm in the next section.

The new algorithms that we shall propose for the general portfolio

analysis problem in Section 3 have two basic properties which are extremely important and useful for the application of these algorithms to problems with specially structured covariance matrices. These properties are: (1) the algorithms can take full advantage of the special structures; and (2) they preserve the special structures throughout the solution process. Of course, without either one of the properties, the other would not be so useful. In order to provide an example of a specially structured covariance matrix, consider the single index model (and in general, a multiple index model also). We know from (1.5) that

$$V = \Sigma + \mathrm{Var}(I_1)\beta_1 \beta_1^T$$

which is a diagonal matrix plus a rank one modification. This is special! Matrices of this class (or even its generalization (1.5)) have certain nice properties which can be used in the new algorithms to facilitate their implementation. In Section 4, we shall specialize the new algorithms to the general index models where we drop assumption (1.4a). Our purpose there is to exploit the special structures inherent in these index models and to show how they can be used in the algorithms most effectively.

2. PARAMETRIC PRINCIPAL PIVOTING

2.1. Background. This subsection is concerned with a brief review with the background materials required for the entire paper. For a given n-vector q and $n \times n$ matrix M, the linear complementarity problem, denoted by (q,M) is that of finding an n-vector x satisfying

$$q + Mx \geq 0, \quad x \geq 0 \quad \text{and} \quad x^T(q + Mx) = 0 . \tag{2.1}$$

Recently, there has been an extensive amount of papers published on the theory and applications of the linear complementarity problem. We mention three most recent survey papers on this subject, Cottle [3], Kaneko [14] and Lemke [18].

It has been proved (see Samelson et al. [25]) that if M is a P-matrix i.e. a real square matrix with positive principal minors, then the linear complementarity problem (q,M) has a unique solution for every vector q.

The parametric linear complementarity problem, denoted by $\{(q + \lambda p,M): \lambda \in R\}$ consists of the family of linear complementarity problems $(q + \lambda p,M)$ where λ is a scalar, to be varied over the whole real line, and p is a given n-vector. For some introductory exposition to the parametric linear complementarity problem, we refer to Cottle [4] and Kaneko [15].

Throughout this entire section, we assume that M is a P-matrix[1], which is not necessarily symmetric, and that p is positive[2]. It then follows that the linear complementarity problem $(q + \lambda p,M)$ has a unique solution $\bar{x}(\lambda)$ for every λ. In fact $\bar{x}(\lambda)$ is a continuous piecewise linear function of λ with finitely many breakpoints (see [4] e.g.). If we let

$$\bar{\lambda} = \max_{1 \leq i \leq n} (-q_i/p_i) , \tag{2.2}$$

---

1. See also the discussion in the last paragraph of this subsection.
2. Of course, we can then assume for simplicity and without loss of generality, that p is the vector of 1's. However, we do not asssume this for reasons which will become clear in Section 4.

then it is clear that $q + \lambda p \geq 0$ for all $\lambda \geq \bar{\lambda}$. Thus $\bar{x}(\lambda) = 0$ for all $\lambda \geq \bar{\lambda}$. To determine $\bar{x}(\lambda)$ for those λ less than $\bar{\lambda}$, we resort to the parametric principal pivoting algorithm which is a parametarized version of Graves' lexicographic principal pivoting algorithm [10]. The parametric principal pivoting algorithm is valid if M is either a P-matrix or a positive semi-definite matrix. It is described in Cottle [4]. We reformulate the algorithm for the case of a P-matrix as follows. For those readers who are unfamiliar with the theory of pivotal algebra, we advise them to consult on the references [2], [5], [10].

Algorithm 1: Original version of the parametric principal pivoting algorithm for $M \in P$ [1].

Step 0. Determine $\bar{\lambda}$ by (2.2). Let the initial canonical tableau be given by

$$u = q + \lambda p + Mx$$

where $q + \lambda p \geq 0$ for all $\lambda \geq \bar{\lambda}$. Let $(\bar{u}(\lambda), \bar{x}(\lambda)) = (q + \lambda p, 0)$ for $\lambda \geq \bar{\lambda}$. Put $\lambda_{old} = \bar{\lambda}$ and go to Step 1.

Step 1. If $p \leq 0$ (this cannot occur initially), let $(\bar{u}(\lambda), \bar{x}(\lambda)) = (q + \lambda p, 0)$ on $(-\infty, \lambda_{old})$ and terminate. Otherwise determine the new critical value

$$\lambda_{new} = \max\{-\frac{q_i}{p_i}: \ p_i > 0\}$$

and let the new critical index k be a maximizing index. Put $(\bar{u}(\lambda), \bar{x}(\lambda)) = (q + \lambda p, 0)$ on $[\lambda_{old}, \lambda_{new}]$. Set $\lambda_{old} = \lambda_{new}$ and to to Step 2.

Step 2. Pivot on m_{kk} and let u, x, q, p and M correspond to the resulting canonical form. Go to Step 1.

Algorithm 1 may be explained briefly as follows. Initially, the determination of $\bar{\lambda}$ such that $q + \lambda p \geq 0$ for all $\lambda \geq \bar{\lambda}$ is possible because p is positive. This then produces the solution $\bar{x}(\lambda)$ for all $\lambda \geq \bar{\lambda}$ immediately. Next, one wants to decrease λ and to determine the solution

1. We sometimes use the letter P to denote the class of P-matrices.

$\bar{x}(\lambda)$ for those λ less than $\bar{\lambda}$. This is done in precisely the same way as in parametric linear programming, namely, by a ratio test. In this instance, it is a maximum ratio test because one is decreasing λ . Having obtained the maximum ratio (which gives the largest extent for the decrease of λ , keeping the current basic solution satisfy the nonnegativity conditions) and the maximizing index, one performs a principal pivot to replace the current "blocking" basic variable by its nonbasic pair. One then repeats the ratio test to decrease λ further. The algorithm terminates when the current p-vector becomes non-positive. In this case, a further decrease of λ can result only in an increase of the values of the current basic variables. Therefore the algorithm terminates.

Several remarks are important to Algorithm 1 and subsequent development. Firstly, the pivots in Step 2 are always possible because of the fact that the matrix M after each such pivot remains a P-matrix (see [2] e.g.) thus has positive diagonal entries. Secondly, by incorporating a lexicographic rule to choose the maximizing index in Step 1, like the one used in Graves' algorithm [10], the above algorithm will compute the solution $\bar{x}(\lambda)$ for all values of λ , in a finite number of steps. The lexicography is used to avoid degeneracy. Lastly, each pair of old and new critical values of λ , namely, λ_{old} and λ_{new}, actually corresponds to a pair of consecutive breakpoints of the solution curve $\bar{x}(\lambda)$.

As mentioned earlier, the parametric principal pivoting algorithm was valid for a parametric linear complementarity problem with either a P-matrix or a positive semi-definite matrix. In this paper, we restrict our consideration to the P-matrix case and leave the other case as the subject for another report.

2.2. Modification and extension. Before proceeding further, we explain some notations to be used later. Let M be an $n \times n$ matrix and let $I, J \subseteq \{1, \ldots n\}$. We define

$$M_{IJ} = \begin{bmatrix} m_{i_1 j_1} & \cdots & m_{i_1 j_t} \\ \vdots & & \vdots \\ m_{i_s j_1} & \cdots & m_{i_s j_t} \end{bmatrix}$$

where $I = \{i_1,\ldots,i_s\}$ and $J = \{j_1,\ldots,j_t\}$ with $1 \le i_1 < \ldots < i_s \le n$ and $1 \le j_1 < \ldots < j_t \le n$. In particular, M_{II} is a principal submatrix of M. Similarly, if q is an n-vector, we define $q_I = (q_{i_1},\ldots,q_{i_s})^T$. If $I = \{1,\ldots,n\}$, we write $M_{.J}$ to mean M_{IJ}. Similarly, if $J = \{1,\ldots,n\}$, we write $M_{I.}$ to mean M_{IJ}. Any vector indexed by the empty set is meant to be the zero vector. The letter e always denotes the vector of 1's.

In the sequel, we first modify Algorithm 1 in such a way that it can be efficiently adopted to large-scale applications and then extend it to allow for upper bounds on the variables. In order to refrain us from repeating, we hereby declare that all the algorithms described in this paper are finite provided that degeneracy or cycling is treated appropriately by some lexicographic rule.

We now explain how we modify Algorithm 1. Following the usual language in pivotal algebra, we call the vectors u and x in the system of linear equations

$$u = q + \lambda p + Mx$$

basic and non-basic vectors respectively and their components basic and non-basic variables respectively. Now, suppose that Step 2 of Algorithm 1 has been performed several times and that the algorithm is not yet terminated. The components of the current basic vector will then consist of some components of the original x-vector together with the rest of the components of the original u-vector. More precisely, there are two disjoint index sets I and J with $I \cup J = \{1,\ldots,n\}$ such that I and J consist of, respectively, those indices of the components of the original u- and x-vectors that are currently basic. (Initially, $I = \{1,\ldots,n\}$

and $J = \phi$.) Furthermore, the current canonical tableau can be obtained by performing a block pivot in the initial tableau. In other words, writing the initial tableau in partitioned form as

$$u_I = q_I + \lambda p_I + M_{II}x_I + M_{IJ}x_J$$

$$u_J = q_J + \lambda p_J + M_{IJ}x_J + M_{JJ}x_J$$

and solving for u_I and x_J in terms of x_I and u_J, we obtain the current canonical tableau:

$$(2.3a)\quad u_I = (q_I - M_{IJ}M_{JJ}^{-1}q_J) + \lambda(p_I - M_{IJ}M_{JJ}^{-1}p_J) + (M_{II} - M_{IJ}M_{JJ}^{-1}M_{JI})x_I + M_{IJ}M_{JJ}^{-1}u_J$$

$$(2.3b)\quad x_J = -M_{JJ}^{-1}q_J - \lambda M_{JJ}^{-1}p_J - M_{JJ}^{-1}M_{JI}x_I + M_{JJ}^{-1}u_J .$$

Letting g_J and h_J be the (unique) solutions to the following systems of linear equations respectively,

$$(2.4)\qquad M_{JJ}g_J = q_J \quad \text{and} \quad M_{JJ}h_J = p_J ,$$

and defining

$$(2.5)\qquad c_I = q_I - M_{IJ}g_J \quad \text{and} \quad d_I = p_I - M_{IJ}h_J$$

we may write (2.3) as

$$(2.6a)\quad u_I = c_I + \lambda d_I + (M_{II} - M_{IJ}M_{JJ}^{-1}M_{JI})x_I + M_{IJ}M_{JJ}^{-1}u_J$$

$$(2.6b)\quad x_J = -g_J - \lambda h_J - M_{JJ}^{-1}M_{JI}x_I + M_{JJ}^{-1}u_J .$$

Noting that the ratio test in Step 1 of Algorithm 1 depends solely on the vectors c_I, d_I, g_J and h_J and that performing the pivot in Step 2 is actually equivalent to adjusting the sets I and J suitably and then to obtaining the corresponding *canonical* tableau (2.6), we can readily state the modified version of Algorithm 1 as follows.

Algorithm 2: Modified version of the parametric principal pivoting for M c P .

Step 0. Let $\lambda_{old} = \infty$. Let $I = \{1,\ldots,n\}$ and $J = \phi$.

Step 1. Solve the systems of equations in (2.4) for g_J and h_J and then compute c_I and d_I by (2.5). Go to Step 2.

Step 2. If $d_I \leq 0$ and $h_J \geq 0$, set $\bar{x}_I(\lambda) = 0$ and $\bar{x}_J(\lambda) = -g_J - \lambda h_J$ for all $\lambda \leq \lambda_{old}$ and terminate. Otherwise, determine the new critical value

$$\lambda_{new} = \max\{ \max\{ -\frac{c_i}{d_i} : d_i > 0,\ i \in I\},\ \max\{-\frac{g_j}{h_j} : h_j < 0,\ j \in J\}\}$$

and let the new critical index k be a maximizing index. Put $\bar{x}_I(\lambda) = 0$ and $\bar{x}_J(\lambda) = -g_J - \lambda h_J$ for $\lambda \in [\lambda_{new}, \lambda_{old}]$. Set $\lambda_{old} = \lambda_{new}$ and go to Step 3.

Step 3. If $k \in I$, then replace I and J by $I \setminus \{k\}$ and $J \cup \{k\}$ respectively. Go to Step 1.

If $k \in J$, then replace I and J by $I \cup \{k\}$ and $J \setminus \{k\}$ respectively. Go to Step 1.

Remark. The set J will always contain at least one element except initially. This is clear from the way the critical index is chosen in Step 2 and the fact that $h_J > 0$ if J is a singleton.

Algorithm 2 is basically the same as Algorithm 1 except that the pivot step in the latter algorithm has been replaced by the updating of the two index sets I and J and by solving the two systems of linear equations in (2.4) which always have unique solutions. The ratio test is performed in Step 2 without any change. The updating of the sets I and J is clear and requires no further explanation.

The idea of replacing the pivot steps by solving systems of linear equation in the above modification seems as though fairly simple, nevertheless it suggests a very efficient way to actually implement the original algorithm, and is especially useful in large-scale applications. As a matter of fact, the same idea has been applied to Lemke's algorithm [17] for certain class of linear complementarity problems. The application was described in [26]. It should be pointed out that the modification described

in Algorithm 2 is not contained in [26] and is obtained independently here. Indeed, the author is indebted to Professor Stephen Robinson who has called his attention to the reference [26] after learning the results obtained by the author.

A large part of the computational effort required in Algorithm 2 consists of solving the systems of linear equations in (2.4) for various sets J. Of course these systems can be solved independently of each other by, for instance, factorization methods. But solving the systems in this way is definitely not most efficient because one is not making use of any old information to generate new data. Here the updating approach described in [26] can be applied. The idea is to avoid solving the whole current systems of linear equations in (2.4) from scratch, but instead, by taking advantage of the information one has about the previous systems. First note that each time (2.4) is invoked, the set J changes by one element, or more precisely, increases or decreases by the new critical index. This means that the submatrix M_{JJ} either increases or decreases by one row and the corresponding column. If one uses orthogonal (QR) factorization to solve the systems of equations in (2.4) (see [11], e.g.), then based on the fact about the change of the submatrix M_{JJ}, it is possible to develop fast numerical procedures to update the QR factors of the new M_{JJ}. These updating procedures can then be used to facilitate the computation of the solution vectors g_J and h_J in (2.4). The amount of operations required to compute the vectors g_J and h_J by this approach is of order ℓ^2 where ℓ is the size of the current M_{JJ}. This is better than solving the systems of equations in (2.4) directly by an order of magnitude. The updating procedures are described in more detail in [26]. We would also like to mention that updating formulas for the inverses of M_{JJ} are given in [27], [11]. The approach of using updated inverses to solve the systems of equations in (2.4) is slightly inferior (although of the same order of computational effort) than that of using updated QR factors

(see [26]). Finally, we point out that the total amount of storage required by using either one of these two approaches is of order n^2.

It is clear that Algorithm 2 has the two properties mentioned in the introduction when it is applied to parametric linear complementarity problems $\{(q + \lambda p, M): \lambda \in R\}$ with specially structured matrices M, namely, it can take advantage of and preserve the special structures throughout the solution process. Sometimes, certain special structure allows the solutions g_J and h_J in (2.4) to be computed readily (or even explicitly) without the need of going through the updating procedures mentioned in the last paragraph. This is true for example, in the case of the index models as we shall see in Section 4.

In the rest of this subsection, we extend the above analysis and Algorithm 2 to deal with upper bounds. A similar extension of Lemke's algorithm is described in [26].

Specifically, we consider the following parametric linear complementarity problem:

$$
\begin{aligned}
(2.7) \qquad & u = q + \lambda p + Mx + y \geq 0 \qquad x \geq 0 \\
& v = a \qquad - x \qquad \geq 0 \qquad y \geq 0 \\
& u^T x = v^T y = 0
\end{aligned}
$$

where λ, q, p and M are as above and a is a given positive vector of upper bounds on the variables. Problems of the type (2.7) arise, for example, from quadratic minimization problems with upper bounds. For fixed value of λ, problem (2.7) is a linear complementarity problem with vector $\begin{pmatrix} q + \lambda p \\ a \end{pmatrix}$ and matrix $\begin{pmatrix} M & I \\ -I & 0 \end{pmatrix}$. Therefore, by varying λ over the whole real line, problem (2.7) becomes the parametric linear complementarity problem $\{\begin{pmatrix} q \\ a \end{pmatrix} + \lambda \begin{pmatrix} p \\ 0 \end{pmatrix}, \begin{pmatrix} M & I \\ -I & 0 \end{pmatrix}): \lambda \in R\}$. Linear complementarity problems of the type (2.7) with fixed λ have been studied intensively in two recent papers [6], [21]. See also [16]. In particular, it was proved in [6] that if M is a P-matrix, then for every fixed λ, problem (2.7) has a solution $\bar{x}^a(\lambda)$ (from which the corresponding $\bar{y}^a(\lambda)$ can be generated easily). The

uniqueness of $\bar{x}^a(\lambda)$ was proved in [21]. Our objective here is to extend Algorithm 2 to determine $\bar{x}^a(\lambda)$ as a function of λ parametrically. For a unified study on problems (2.7) with $\lambda = 0$ and (2.1), we refer to [22] where the author treats the vector a of upper bounds as having some components equal to infinity.

The reason that Algorithm 2 is not directly applicable to problem (2.7) is obvious: $\begin{pmatrix} M & I \\ -I & 0 \end{pmatrix}$ is not a P-matrix even if M is. Suppose that we have obtained some value $\bar{\lambda}$ of λ for which we know the solutions $\bar{x}^a(\lambda)$ for all $\lambda \geq \bar{\lambda}$ and now we want to decrease λ to complete the solution process. In the solution $\bar{x}^a(\lambda)$, some components will be basic and others nonbasic. Among the basic components, some will at their upper bounds. Let I and J denote respectively, the sets of indices of the now basic u- and x-variables, with J consisting of J_1 and J_2 where J_2 corresponds to the subset of J such that the basic x-variables with indices in J_2 are at their upper bounds and J_1 is the complement of J_2 in J. (Initially, $\bar{\lambda}$ is given by (2.2), $I = \{1,\ldots,n\}$, and $J_1 = J_2 = \phi$.) With the sets I, J_1 and J_2, we can write the initial canonical tableau in partitioned form as:

$$
\begin{aligned}
u_I &= q_I + \lambda p_I + M_{II}x_I + M_{IJ_1}x_{J_1} + M_{JJ_2}x_{J_2} + y_I \\
u_{J_1} &= q_{J_1} + \lambda p_{J_1} + M_{J_1I}x_I + M_{J_1J_1}x_{J_1} + M_{J_1J_2}x_{J_2} + y_{J_1} \\
u_{J_2} &= q_{J_2} + \lambda p_{J_2} + M_{J_2I}x_I + M_{J_2J_1}x_{J_1} + M_{J_2J_2}x_{J_2} + y_{J_2} \\
v_I &= a_I - x_I \\
v_{J_1} &= a_{J_1} - x_{J_1} \\
v_{J_2} &= a_{J_2} - x_{J_2} ,
\end{aligned}
$$

and the intermediate canonical tableau before the further decrease of λ as:

$$u_I = [q_I - M_{IJ_1} M_{J_1J_1}^{-1} (q_{J_1} + M_{J_1J_2} a_{J_2}) + M_{IJ_2} a_{J_2}] + \lambda (p_I - M_{IJ_1} M_{J_1J_1}^{-1} p_{J_1})$$
$$+ (M_{II} - M_{IJ_1} M_{J_1J_1}^{-1} M_{J_1I}) x_I + M_{IJ_1} M_{J_1J_1}^{-1} u_{J_1} - (M_{IJ_2} - M_{IJ_1} M_{J_1J_1}^{-1} M_{J_1J_2}) v_{J_2} + y_I$$
$$- M_{IJ_1} M_{J_1J_1}^{-1} y_{J_1}$$

$$x_{J_1} = -M_{J_1J_1}^{-1} (q_{J_1} + M_{J_1J_2} a_{J_2}) - \lambda M_{J_1J_1}^{-1} p_{J_1} - M_{J_1J_1}^{-1} M_{J_1I} x_I + M_{J_1J_1}^{-1} u_{J_1} + M_{J_1J_1}^{-1} M_{J_1J_2} v_{J_2}$$
$$- M_{J_1J_1}^{-1} y_{J_1}$$

$$x_{J_2} = a_{J_2} \qquad - v_{J_2}$$

$$v_I = a_I \qquad - x_I$$

$$v_{J_1} = [a_{J_1} + M_{J_1J_1}^{-1} (q_{J_1} + M_{J_1J_2} a_{J_2})] + \lambda M_{J_1J_1}^{-1} p_{J_1} + M_{J_1J_1}^{-1} M_{J_1I} x_I - M_{J_1J_1}^{-1} u_{J_1}$$
$$- M_{J_1J_1}^{-1} M_{J_1J_2} v_{J_2} + M_{J_1J_1}^{-1} y_{J_1}$$

$$y_{J_2} = -[q_{J_2} - M_{J_2J_1} M_{J_1J_1}^{-1} (q_{J_1} + M_{J_1J_2} a_{J_2}) + M_{J_2J_2} a_{J_2}] - \lambda (p_{J_2} - M_{J_2J_1} M_{J_1J_1}^{-1} p_{J_1})$$
$$- (M_{J_2I} - M_{J_2J_1} M_{J_1J_1}^{-1} M_{J_1I}) x_I - M_{J_2J_1} M_{J_1J_1}^{-1} u_{J_1} + (M_{J_2J_2} - M_{J_2J_1} M_{J_1J_1}^{-1} M_{J_1J_2}) v_{J_2} + u_{J_2}$$
$$+ M_{J_2J_1} M_{J_1J_1}^{-1} y_{J_1} .$$

Letting g_{J_1} and h_{J_1} denote the (unique) solutions to the following systems of linear equations respectively,

(2.8) $$M_{J_1J_1} g_{J_1} = q_{J_1} + M_{J_1J_2} a_{J_2} \quad \text{and} \quad M_{J_1J_1} h_{J_1} = p_{J_1}$$

and defining

(2.9a) $$c_I = q_I - M_{IJ_1} g_{J_1} + M_{IJ_2} a_{J_2}, \quad d_I = p_I - M_{IJ_1} h_{J_1}$$

(2.9b) $$c_{J_2} = q_{J_2} - M_{J_2J_1} g_{J_1} + M_{J_2J_2} a_{J_2}, \quad d_{J_2} = p_{J_2} - M_{J_2J_1} h_{J_1}$$

we may rewrite the tableau corresponding to $\lambda = \bar{\lambda}$ as

$$
\left.
\begin{aligned}
u_I &= c_I + \lambda d_I \\
x_{J_1} &= -g_{J_1} - \lambda h_{J_1} \\
x_{J_2} &= a_{J_2} \\
v_I &= a_I \\
v_{J_1} &= (a_{J_1} + g_{J_1}) + \lambda h_{J_1} \\
y_{J_2} &= -c_{J_2} - \lambda d_{J_2}
\end{aligned}
\right\} + \text{nonbasic portion.}
\tag{2.10}
$$

It is now clear from (2.10) how to proceed to determine the decrease of λ. Based on (2.10) and the same idea as before, we deduce the next algorithm which is an extension of Algorithm 2 to solve problem (2.7).

Algorithm 3: Extension of the parametric principal pivoting algorithm to deal with upper bounds for $M \in P$.

Step 0. Let $\lambda_{old} = \infty$. Let $I = \{1,\ldots,n\}$ and $J_1 = J_2 = \phi$.

Step 1. Solve the systems of equations in (2.8) for g_{J_1} and h_{J_1} and then compute c_I, d_I, c_{J_2} and d_{J_2} by (2.9). Go to Step 2.

Step 2. If $d_I \leq 0$, $h_{J_1} = 0$ and $d_{J_2} \geq 0$, set $\bar{x}^a(\lambda) = a$ for all $\lambda \leq \lambda_{old}$ and terminate (in fact, I and J_1 will then be empty). Otherwise determine the new critical value

$$
\lambda_{new} = \max\{ \max\{-\frac{c_i}{d_i}: d_i > 0,\ i \in I\},\ \max\{-\frac{g_{j_1}}{h_{j_1}}: h_{j_1} < 0,\ j_1 \in J_1\},
$$

$$
\max\{-\frac{a_{j_1}+g_{j_1}}{h_{j_1}}: h_{j_1} > 0, j_1 \in J_1\},\ \max\{-\frac{c_{j_2}}{d_{j_2}}: d_{j_2} < 0,\ j_2 \in J_2\}\}
$$

and let the new critical index k be a maximizing index. Put $\bar{x}^a_I(\lambda) = 0$, $\bar{x}^a_{J_1}(\lambda) = -g_{J_1} - \lambda h_{J_1}$ and $\bar{x}^a_{J_2}(\lambda) = a_{J_2}$ for $\lambda \in [\lambda_{old}, \lambda_{new}]$. Set $\lambda_{old} = \lambda_{new}$ and go to Step 3.

Step 3. If $k \in I$, then replace I and J_1 by $I \setminus \{k\}$ and $J_1 \cup \{k\}$ respectively. Go to Step 1.

If $k \in J_2$, then replace J_1 and J_2 by $J_1 \cup \{k\}$ and $J_2 \setminus \{k\}$ respectively. Go to step 1.

If $k \in J_1$ and $h_k < 0$, then replace I and J_1 by $I \cup \{k\}$ and $J_1 \setminus \{k\}$ respectively. Go to Step 1.

If $k \in J_1$ and $h_k > 0$, then replace J_1 and J_2 by $J_1 \setminus \{k\}$ and $J_2 \cup \{k\}$ respectively. Go to Step 1.

The termination of Algorithm 3 deserves some more discussion. According to Step 2 of the algorithm, we have at termination $d_I \leq 0$, $h_{J_1} = 0$ and $d_{J_2} \geq 0$, and for all $\lambda \leq \lambda_{old}$, all the components of the solution $\hat{x}^a(\lambda)$ are at their upper bounds. In fact, $h_{J_1} = 0$ if and only if $J_1 = \phi$. Therefore since p is positive, we must have $I = \phi$. On the other hand, the fact that the solution $\bar{x}^a(\lambda) = a$ for all $\lambda \leq \lambda_{old}$ is to be expected and can be justified directly from the original formulation (2.7) of the problem. As we know, the solution $\bar{x}(\lambda)$ is a piecewise linear function of λ. Therefore, there exists a value $\underline{\lambda}$ (which is in fact, λ_{old} at termination of Algorithm 3) such that $\bar{x}^a(\lambda)$ is linear for all $\lambda \leq \underline{\lambda}$. Now since $\bar{x}^a(\lambda)$ is bounded above and below, it must be independent of λ for $\lambda \leq \underline{\lambda}$, i.e. $\bar{x}^a(\lambda)$ is equal to some constant vector for $\lambda \leq \underline{\lambda}$. (This is not true in the case without the upper bounds.) Suppose the i-th component of this vector lies strictly below its upper bound. Then we must have, by complementarity

$$0 \leq u_i = q_i + \lambda p_i + (M\bar{x}^a(\lambda))_i$$

which certainly can not be true for all negative λ. Therefore it follows that for all $\lambda \leq \underline{\lambda}$, $\bar{x}^a(\lambda)$ is equal to the vector a of upper bounds.

It should be pointed out that every time Step 3 is passed, the set J_1 changes (by one element). It may sometimes be empty (as it must be at termination). Furthermore, the updating procedures described earlier can be used to solve the systems of equations in (2.8) more efficiently.

We conclude this subsection and also this section by remarking that under suitable modification, Algorithm 3 can be applied to the situation where some of the upper bounds are infinity. Roughly speaking, this is done by restricting the set J_2 to include only those indices for which the upper bounds are less than infinity. See the discussion in [22] for the case of fixed λ.

3. APPLICATIONS TO PORTFOLIO ANALYSIS

3.1. The case without explicit upper bounds. In this subsection, we apply Algorithm 2 to develop a new and efficient algorithm (Algorithm 5) for the following parametric quadratic program.

$$\text{(3.1)} \qquad \begin{aligned} &\text{minimize} && \tfrac{1}{2}\, x^T M x + (q + \theta r)^T x \\ &\text{subject to} && x \geq 0 \\ &\text{and} && p^T x = 1 \end{aligned}$$

where M is a given symmetric positive definite $n \times n$ matrix; p,q and r are given n-vectors with p positive; and θ is a nonnegative parameter. We recall that the general portfolio analysis problem with a nonsingular covariance matrix is a problem of this type. Furthermore we have $q = 0$ and $p = e$ (see (1.2)). The first identification $(q = 0)$ has certain advantage in the implementation of the first algorithm proposed below.

Since p is positive, the variables in problem (3.1) are always implicitly bounded above. Therefore since M is symmetric and positive definite, problem (3.1) has a unique solution $\hat{x}(\theta)$ for every θ. Furthermore, $\hat{x}(\theta)$ is also the unique vector satisfying the following Kuhn-Tucker conditions

$$\text{(3.2a)} \qquad u = q + \lambda p + \theta r + Mx \geq 0, \quad x \geq 0$$

$$\text{(3.2b)} \qquad u^T x = 0$$

$$\text{(3.2c)} \qquad p^T x = 1$$

where λ is the Lagrange multiplier associated with the equality constraint (3.2c) and is dependent on θ. Also λ is not restricted in sign. Consider $\theta = 0$ first. The resulting problem is

$$\text{(3.3a)} \qquad u = q + \lambda p + Mx \geq 0, \quad x \geq 0$$

$$\text{(3.3b)} \qquad u^T x = 0$$

$$\text{(3.3c)} \qquad p^T x = 1 .$$

Now problem (3.3) can be regarded as the parametric linear complementarity problem $\{(q + \lambda p, M): \lambda \in R\}$ coupled with the single equality

constraint (3.3c). In other words, to obtain $\hat{x}(0)$, we can let λ be a parameter varying over the real line and aim at determining a suitable λ^* such that the solution $\bar{x}(\lambda^*)$ to the linear complementarity problem $(q + \lambda^* p, M)$ satisfies condition (3.3c) also. It is then clear that $\hat{x}(0)$ is given precisely by $\bar{x}(\lambda^*)$. One way to find such a λ^* and the solution $\bar{x}(\lambda^*)$ is to solve the parametric linear complementarity problem $\{(q + \lambda p, M): \lambda \in R\}$ and to develop a systematic way to check if the solution $\bar{x}(\lambda)$ satisfies condition (3.3c). Algorithm 2 can be applied here to obtain $\bar{x}(\lambda)$ for every λ and in fact, it also provides an easy way to check (3.3c). In what follows, we explain how this latter part can be done. Roughly speaking, it is achieved by linear interpolation.

Recall that $\bar{x}(\lambda)$ is a continuous piecewise linear function of λ. Indeed, according to Step 2 in Algorithm 2, we have

$$\bar{x}_I(\lambda) = 0 \quad \text{and} \quad \bar{x}_J(\lambda) = -g_J - \lambda h_J \text{ for } \lambda \in [\lambda_{new}, \lambda_{old}]$$

where λ_{new} and λ_{old} correspond to a pair of consecutive breakpoints of the solution curve $\bar{x}(\lambda)$; g_J and h_J are solutions to the systems of linear equations in (2.4). Thus,

$$p^T\bar{x}(\lambda) = -p_J^T g_J - \lambda p_J^T h_J ,$$

for $\lambda \in [\lambda_{new}, \lambda_{old}]$. Setting $p^T\bar{x}(\lambda) = 1$ gives

$$\bar{\lambda} = -(1 + p_J^T g_J)/p_J^T h_J , \tag{3.4}$$

provided that J is nonempty. Geometrically, the quantity $\bar{\lambda}$ given by (3.4) is precisely that value of λ where the (possibly extended) line segment of the solution curve $\bar{x}(\lambda)$ in the interval $[\lambda_{new}, \lambda_{old}]$ meets the line $p^T\bar{x}(\lambda) = 1$. So if this quantity $\bar{\lambda}$ lies within the interval (i.e. if the intersection occurs within the interval), we have found $\lambda^* = \bar{\lambda}$. Otherwise we decrease λ and proceed to the next interval. Eventually in a finite number of steps, this process will terminate with a λ^* and of course, the solution $\bar{x}(\lambda^*)$ $(= \hat{x}(0))$. We summarize the discussion in the next algorithm which solves problem (3.1) with $\theta = 0$.

Algorithm 4: QP with $\theta = 0$ and no upper bounds.

Step 0. Determine $\lambda_{old} = \max_{1 \le i \le n} \{-\frac{q_i}{p_i}\}$ and let k be a maximizing index. Let $I = \{1,\ldots,n\} \setminus \{k\}$ and $J = \{k\}$. Go to Step 1.

Step 1. Solve the systems of linear equations in (2.4) for g_J and h_J and then compute c_I and d_I by (2.5). Go to Step 2.

Step 2. Compute $\bar{\lambda}$ by (3.4). If $d_I \le 0$ and $h_J \ge 0$, then set

$$\lambda^* = \bar{\lambda}, \quad \hat{x}_I(0) = 0 \quad \text{and} \quad \hat{x}_J(0) = -g_J - \lambda^* h_J \tag{3.5}$$

and terminate. Otherwise determine

$$\lambda_{new} = \max\{\max\{-\frac{c_i}{d_i} : d_i > 0,\ i \in I\},\ \max\{-\frac{g_i}{h_j} : h_j < 0,\ j \in J\}\} \ . \tag{3.6}$$

If $\bar{\lambda} \in [\lambda_{new}, \lambda_{old}]$, then define λ^* and $\hat{x}(0)$ by (3.5) and terminate. Otherwise let k be a maximizing index in (3.6). Go to Step 3.

Step 3. If $k \in I$, replace I and J by $I \setminus \{k\}$ and $J \cup \{k\}$ respectively. Go to Step 1.

If $k \in J$, replace I and J by $I \cup \{k\}$ and $J \setminus \{k\}$ respectively. Go to Step 1.

Remark 1. As in Algorithm 2, the set J always contains at least one element, so that the quantity $\bar{\lambda}$ in (3.4) is always well-defined by the positive definiteness of M.

Remark 2. If q happens to be zero, as in the portfolio analysis problem (1.2), then it is easy to see that λ_{old} and λ_{new} will always be zero. Therefore the ratio test (3.6) and the one in Step 0 can be eliminated. Furthermore, the maximizing index k can be chosen "arbitrarily" (subject to non-cycling) provided either one of the following two conditions is satisfied:

(1) $$d_k > 0, \quad k \in I \ ,$$

(2) $$h_k < 0, \quad k \in J \ .$$

In Step 0, k can be chosen arbitrarily in $\{1,\ldots,n\}$. Finally, if $q = 0$, Algorithm 4 terminates if and only if $d_I \le 0$ and $h_J \ge 0$ for some suitable I and J.

Remark 3. The assumption that M is positive definite is essential for the success of Algorithm 4 to produce the solution $\hat{x}(0)$ to problem (3.1) with $\theta = 0$. This is because, on the one hand, the assumption guarantees the existence and uniqueness of $\hat{x}(0)$ which also satisfies conditions (3.2a)-(3.2c) for $\theta = 0$ and suitable λ; on the other hand, it guarantees that for every λ, the linear complementarity problem $(q + \lambda p, M)$ has a unique solution. Consequently, by solving the parametric linear complementarity problem $\{(q + \lambda p, M): \lambda \in R\}$, one is bound to arrive at the λ^* and the solution $\hat{x}(0)$ given in (3.5).

Using Algorithm 4 as a start, we may proceed to develop a parametric algorithm for problem (3.1), or equivalently, problem (3.2). Observe that there are "two" parameters λ and θ in (3.2), with λ depending θ. It will be shown in what follows that λ can always be eliminated by means of the equality constraint (3.2c) and expressed in terms of the values of the current basic x-variables and the real parameter θ. Hence we are left with one parameter and can therefore proceed as before.

Let I and J be the index sets at termination of Algorithm 4. Then we can write

$$u = q + \lambda p + \theta r + Mx$$

in canonical form with respect to I and J as:

$$(3.7a) \quad u_I = c_I + \lambda d_I + \theta b_I + (M_{II} - M_{IJ}M_{JJ}^{-1}M_{JI})x_I + M_{IJ}M_{JJ}^{-1}u_J$$

$$(3.7b) \quad x_J = -g_J - \lambda h_J - \theta f_J - M_{JJ}^{-1}M_{JI}x_I + M_{JJ}^{-1}u_J$$

where f_J, g_J and h_J are (unique) solutions to the systems of linear equations

$$(3.8) \quad M_{JJ}(f_J, g_J, h_J) = (r_J, q_J, p_J)$$

respectively; c_I and d_I are given in (2.5); and finally

$$b_I = r_I - M_{IJ}f_J .$$

Setting the nonbasic variables x_I and u_J at zero, we have

$$p^T x = -p_J^T g_J - \lambda p_J^T h_J - \theta p_J^T f_J \quad .$$

Setting $p^T x = 1$ gives

$$\lambda = -\frac{1+p_J^T g_J}{p_J^T h_J} - \theta \frac{p_J^T f_J}{p_J^T h_J} \quad , \tag{3.9}$$

provided that J is nonempty. Substituting λ into (3.7), we obtain

$$\left.\begin{aligned} u_I &= (c_I - \frac{1+p_J^T g_J}{p_J^T h_J} d_I) + \theta(b_I - \frac{p_J^T f_J}{p_J^T h_J} d_I) \qquad (3.10a) \\ x_J &= -(g_J - \frac{1+p_J^T g_J}{p_J^T h_J} h_J) - \theta(f_J - \frac{p_J^T f_J}{p_J^T h_J} h_J) \qquad (3.10b) \end{aligned}\right\} + \text{nonbasic portion}$$

Starting with $\theta = 0$, we may use tableau (3.10) to determine the increase of θ and proceed as before. Recalling the definition of c_I, d_I and b_I, we may write

$$c_I - \frac{1+p_J^T g_J}{p_J^T h_J} d_I = (q_I - \frac{1+p_J^T g_J}{p_J^T h_J} p_I) - M_{IJ}(g_J - \frac{1+p_J^T g_J}{p_J^T h_J} h_J)$$

$$b_I - \frac{p_J^T f_J}{p_J^T h_J} d_I = (r_I - \frac{p_J^T f_J}{p_J^T h_J} p_I) - M_{IJ}(f_J - \frac{p_J^T f_J}{p_J^T h_J} h_J) \quad .$$

Now we can formulate the parametric algorithm for problem (3.1) as follows.

Algorithm 5: Parametric QP with no upper bounds.

Step 0. Use Algorithm 4 to obtain $\hat{x}(0)$ and let I and J be the two index sets at termination of the algorithm. Let $\theta_{old} = 0$.

Step 1. Solve the systems of linear equations in (3.8) for f_J, g_J and h_J . Compute

(3.11) $$\bar{\lambda} = -(1 + p_J^T g_J)/p_J^T h_J, \quad \bar{\mu} = -p_J^T f_J/p_J^T h_J$$

(3.12) $$s_J = g_J + \bar{\lambda} h_J, \quad t_J = f_J + \bar{\mu} h_J$$

(3.13) $$c_I = (q_I + \bar{\lambda} p_I) - M_{IJ} s_J, \quad d_I = (r_I + \bar{\mu} p_I) - M_{IJ} t_J \ .$$

Step 2. If $d_I \geq 0$ and $t_J \leq 0$, set $\hat{x}_I(\theta) = 0$ and $\hat{x}_J(\theta) = -s_J$ for $\theta \geq \theta_{old}$ and terminate (in fact, t_J will then be zero). Otherwise determine

(3.14) $$\theta_{new} = \min\{\min\{-\frac{c_i}{d_i}: d_i < 0, \ i \in I\}, \min\{-\frac{s_j}{t_j}: t_j > 0, \ j \in J\}\}$$

and let k be a minimizing index. Put $\hat{x}_I(\theta) = 0$ and $\hat{x}_J(\theta) = -s_J - \theta t_J$ for $\theta \in [\theta_{old}, \theta_{new}]$. Set $\theta_{old} = \theta_{new}$ and go to Step 3.

Step 3. If $k \in I$, replace I and J by $I \setminus \{k\}$ and $J \cup \{k\}$ respectively. Go to Step 1.

If $k \in J$, replace I and J by $I \cup \{k\}$ and $J \setminus \{k\}$ respectively. Go to Step 1.

Remark 1. The set J will never be empty because of the fact that $t_J = 0$ for singleton J and also of the way the minimizing index k is chosen. Therefore, the quantities $\bar{\lambda}$ and $\bar{\mu}$ in (3.11) are always well-defined.

Remark 2. This time, a minimum ratio test is performed in Step 2 because we are increasing θ .

Remark 3. The following two identities are clear from (3.11) and (3.12)

(3.15) $$p_J^T s_J = -1 \quad \text{and} \quad p_J^T t_J = 0 \ .$$

They guarantee that for every θ, the solution $\hat{x}(\theta)$ defined in Step 2 satisfies $p^T \hat{x}(\theta) = 1$.

Remark 4. As different from the last algorithm, the condition $(q = 0)$ does not induce too much simplication in Algorithm 5 except possibly in the computations (3.11)-(3.13).

It is worth pointing out two rather interesting consequences at the termination of Algorithm 5. The first consequence is that the solution $\hat{x}(\theta)$ for all θ greater than or equal to the value of θ at termination is equal to some constant vector which is independent of the parameter θ. The direct justification for this is similar to that for Algorithm 3 which we have discussed earlier. The second consequence reflects certain relationship between the solution $\hat{x}(\theta)$ at termination and the ratios $(r_i/p_i)_{i=1}^n$. According to Step 2, we have $d_I \geq 0$ and $t_J \leq 0$ at termination. Since p is positive, (3.15) then implies that indeed $t_J = 0$. Recalling the definitions of d_I and t_J, we may deduce that

$$r_I + \bar{\mu} p_I \geq 0 \quad \text{and} \quad r_J + \bar{\mu} p_J = 0 \tag{3.16}$$

or equivalently,

$$\min_{i \in I} \{\frac{r_i}{p_i}\} \geq -\bar{\mu} = \frac{r_j}{p_j} \quad \text{for every} \quad j \in J . \tag{3.16)'}$$

This latter relation says that if θ is large enough, then component j of the solution $\hat{x}(\theta)$ can not be positive unless the corresponding ratio r_j/p_j is no greater than all the other ratios (r_i/p_i).

3.2. The case with explicit upper bounds. In this subsection, we extend the analysis and algorithms in the previous subsection to the problem with explicit upper bounds:

$$\begin{aligned} &\text{minimize } \frac{1}{2} x^T M x + (q + \theta r)^T x \\ &\text{subject to} \quad a \geq x \geq 0 \\ &\text{and} \quad p^T x = 1 \end{aligned} \tag{3.17}$$

where p, q, r and M are as above and a is a positive vector such that $a_i < 1/p_i$ for every i. This latter condition is imposed on the vector a so that the upper bounds will not be redundant. Furthermore, in order for the problem (3.17) to be feasible, we must have $p^T a \geq 1$. If $p^T a = 1$, then every feasible vector x must satisfy $x = a$ because p is positive. The problem (3.17) thus becomes trivial. Consequently, we assume $p^T a > 1$

throughout the discussion below. The portfolio analysis problem with upper bounds (1.7) is of the form (3.17) with $q = 0$ and $p = e$.

For every fixed θ, let $\hat{x}^a(\theta)$ denote the (unique) solution to problem (3.17). Since M is symmetric and positive definite, problem (3.17) is actually equivalent to its Kuhn-Tucker optimality conditions

$$u = q + \lambda p + \theta r + Mx + y \geq 0, \qquad x \geq 0 \tag{3.18a}$$

$$v = a - x \geq 0, \qquad y \geq 0 \tag{3.18b}$$

$$u^T x = v^T y = 0 \tag{3.18c}$$

$$p^T x = 1 . \tag{3.18d}$$

Consider $\theta = 0$. Then condition (3.18a)-(3.18c) can be regarded as a parametric linear complementarity problem of the form (2.7) with λ as the parameter. Therefore, in order to compute $\hat{x}^a(0)$ which must satisfy conditions (3.18a)-(3.18d) for some suitable λ, we may use Algorithm 3 and incorporate a rule to check condition (3.18d). The idea is exactly the same for the case with no explicit upper bounds; therefore we omit the detail. The next algorithm extends Algorithm 4 to solve problem (3.17) with $\theta = 0$.

Algorithm 6: QP with $\theta = 0$ and explicit upper bounds.

Step 0. Determine $\lambda_{old} = \max_{1 \leq i \leq n} \{-q_i/p_i\}$ and let k be a maximizing index. Let $I = \{1,\ldots,n\} \setminus \{k\}$, $J_1 = \{k\}$ and $J_2 = \phi$. Go to Step 1.

Step 1. Solve the systems of linear equations in (2.8) for g_{J_1} and h_{J_1} and then compute c_I, d_I, c_{J_2} and d_{J_2} by (2.9). Go to Step 2.

Step 2. Determine

$$\lambda_{new} = \max\{\max\{-\frac{c_i}{d_i}: d_i > 0, i \in I\}, \max\{-\frac{g_{j_1}}{h_{j_1}}: h_{j_1} < 0, j_1 \in J_1\}, \tag{3.19}$$

$$\max\{-\frac{a_{j_1}+g_{j_1}}{h_{j_1}}: h_{j_1} > 0, j_1 \in J_1\}, \max\{-\frac{c_{j_2}}{d_{j_2}}: d_{j_2} < 0, j_2 \in J_2\}\} .$$

Case (i) $J_1 \neq \phi$. Put

$$\bar{\lambda} = -(1 + p_{J_1}^T g_{J_1} - p_{J_2}^T a_{J_2})/p_{J_1}^T h_{J_1} .$$

If $\bar{\lambda} \in [\lambda_{new}, \lambda_{old}]$, set $\lambda^* = \bar{\lambda}$,

$$\hat{x}_I^a(0) = 0, \quad \hat{x}_{J_1}^a(0) = -g_{J_1} - \lambda^* h_{J_1} \quad \text{and} \quad \hat{x}_{J_2}^a(0) = a_{J_2}$$

and terminate. Otherwise let k be a maximizing index in (3.19). Set $\lambda_{old} = \lambda_{new}$ and go to Step 3.

Case (ii) $J_1 = \phi$ (or equivalently, $h_{J_1} = 0$). If $p_{J_2}^T a_{J_2} = 1$, set $\lambda^* = \lambda_{new}$

$$\hat{x}_I^a(0) = 0 \quad \text{and} \quad \hat{x}_{J_2}^a(0) = a_{J_2}$$

and terminate. Otherwise, let k be a maximizing index in (3.19). Set $\lambda_{old} = \lambda_{new}$ and go to Step 3.

Step 3. If $k \in I$, replace I and J_1 by $I \setminus \{k\}$ and $J_1 \cup \{k\}$ respectively. Go to Step 1.

If $k \in J_2$, replace J_1 and J_2 by $J_1 \cup \{k\}$ and $J_2 \setminus \{k\}$ respectively. Go to Step 1.

If $k \in J_1$ and $h_k < 0$, replace I and J_1 by $I \cup \{k\}$ and $J_1 \setminus \{k\}$ respectively. Go to Step 1.

If $k \in J_1$ and $h_k > 0$, replace J_1 and J_2 by $J_1 \setminus \{k\}$ and $J_2 \cup \{k\}$ respectively. Go to Step 1.

The existence of a maximizing index k in (3.19) still requires justification. However, this is clear because if $d_I \leq 0$, $h_{J_1} = 0$ and $d_{J_2} \geq 0$, then according to what we have remarked earlier, these conditions indicate that in fact, $I = J_1 = \phi$ and that we have arrived at the last breakpoint of the solution curve $\bar{x}^a(\lambda)$. These latter two facts together would imply $\hat{x}^a(0) = a$, contradicting $p^T a \neq 1$. Therefore a maximizing index k in (3.19) always exists.

Finally, we present the parametric algorithm to solve problem (3.17). It is an extension of Algorithm 5 to deal with explicit upper bounds.

Algorithm 7: Parametric QP with explicit upper bounds.

Step 0. Use Algorithm 6 to obtain $\hat{x}^a(0)$ and let I, J_1 and J_2 be the three index sets at termination of the algorithm. Let $\theta_{old} = 0$. If $J_1 = \phi$, set $\bar{\lambda} = \lambda^*$, $\bar{\mu} = 0$ and go directly to (3.23). Otherwise go to Step 1.

Step 1. Solve the systems of linear equations

$$M_{J_1J_1}(f_{J_1}, g_{J_1}, h_{J_1}) = (r_{J_1}, q_{J_1} + M_{J_1J_2}a_{J_2}, p_{J_1}) \tag{3.20}$$

for f_{J_1}, g_{J_1} and h_{J_1}. Compute

$$\bar{\lambda} = -(1 + p_{J_1}^T g_{J_1} - p_{J_2}^T a_{J_2})/p_{J_1}^T h_{J_1}, \quad \bar{\mu} = -p_{J_1}^T f_{J_1}/p_{J_1}^T h_{J_1} \tag{3.21}$$

$$s_{J_1} = g_{J_1} + \bar{\lambda} h_{J_1}, \quad t_{J_1} = f_{J_1} + \bar{\mu} h_{J_1} \tag{3.22}$$

$$c_I = (q_I + \bar{\lambda} p_I) + M_{IJ_2}a_{J_2} - M_{IJ_1}s_{J_1}, \quad d_I = (s_I + \bar{\mu} p_I) - M_{IJ_1}t_{J_1} \tag{3.23}$$

$$c_{J_2} = (q_{J_2} + \bar{\lambda} p_{J_2}) + M_{J_2J_2}a_{J_2} - M_{J_2J_1}s_{J_1}, \quad d_{J_2} = (r_{J_2} + \bar{\mu} p_{J_2}) - M_{J_2J_1}t_{J_1}. \tag{3.24}$$

Step 2. If $d_I \geq 0$, $t_{J_1} = 0$ and $d_{J_2} \leq 0$, set $\hat{x}_I^a(\theta) = 0$, $\hat{x}_{J_1}^a(\theta) = -s_{J_1}$ and $\hat{x}_{J_2}^a(\theta) = a_{J_2}$ for $\theta \geq \theta_{old}$ and terminate. Otherwise determine

$$\theta_{new} = \min\{\min\{-\frac{c_i}{d_i}: d_i < 0,\ i \in I\}, \min\{-\frac{s_{j_1}}{t_{j_1}}: t_{j_1} > 0,\ j_1 \in J_1\}, \tag{3.25}$$

$$\min\{-\frac{a_{j_1} + s_{j_1}}{t_{j_1}}: t_{j_1} < 0,\ j_1 \in J_1\}, \min\{-\frac{c_{j_2}}{d_{j_2}}: d_{j_2} > 0,\ j_2 \in J_2\}\}$$

and let k be a minimizing index. But $\hat{x}_I^a(\theta) = 0$, $\hat{x}_{J_1}^a(\theta) = -s_{J_1} - \theta t_{J_1}$ and $\hat{x}_{J_2}^a(\theta) = a_{J_2}$ for $\theta \in [\theta_{old}, \theta_{new}]$. Set $\theta_{old} = \theta_{new}$ and go to Step 3.

Step 3. If $k \in I$, replace I and J_1 by $I \setminus \{k\}$ and $J_1 \cup \{k\}$ respectively. Go to Step 1.

If $k \in J_2$, replace J_1 and J_2 by $J_1 \cup \{k\}$ and $J_2 \setminus \{k\}$ respectively. Go to Step 1.

If $k \in J_1$ and $t_k > 0$, replace I and J_1 by $I \cup \{k\}$ and $J_1 \setminus \{k\}$ respectively. Go to Step 1.

If $k \in J_1$ and $t_k < 0$, replace J_1 and J_2 by $J_1 \setminus \{k\}$ and $J_2 \cup \{k\}$ respectively. Go to Step 1.

Remark 1. The set J_1 will never be empty except possibly at the termination of Algorithm 6. The reason is the same as in the case with no upper bounds. Furthermore (3.15) holds if J is replaced by J_1.

Remark 2. The following analog of relation (3.16) holds at termination of Algorithm 7:

$$(3.26) \qquad r_I + \bar{\mu} p_I \geq 0, \quad r_{J_1} + \bar{\mu} p_{J_1} = 0 \quad \text{and} \quad r_{J_2} + \bar{\mu} p_{J_2} \leq 0$$

or equivalently,

$$(3.26)' \qquad \min_{i \in I} \{\frac{r_i}{p_i}\} \geq -\bar{\mu} = \frac{r_{j_1}}{p_{j_1}} \geq \max_{j_2 \in J_2} \{\frac{r_{j_2}}{p_{j_2}}\} \quad \text{for every } j_1 \in J_1 .$$

This latter relation (3.26)' indicates how the solution $\hat{x}^a(\theta)$ is related to the ratios $(r_i/p_i)_{i=1}^n$ for sufficiently large θ.

4. SPECIALIZATIONS TO INDEX MODELS

In this section, we apply the algorithms developed in the last section to some specific models in portfolio analysis, namely, the index models. Recall that in an m-index model, the returns of the securities are given by (1.3) and there are three assumptions (1.4a)-(1.4c) concerning various covariances. Since it is rather unreasonable in practice to assume that no two indices are correlated, we will drop the assumption (1.4a) throughout the following discussion. We denote by C the covariance matrix of the indices, i.e. $C = (c_{ij})_{m \times m}$ where

$$c_{ij} = \mathrm{cov}(I_i, I_j) \quad \text{for every} \quad i,j = 1,\ldots,m \ .$$

Under this setting, it is easy to deduce that the covariance matrix V of portfolio returns is given by

$$V = \Sigma + (\beta_1,\ldots,\beta_m)\, C\, (\beta_1,\ldots,\beta_m)^T \tag{4.1}$$

where Σ is the nonnegative diagonal matrix whose i-th diagonal element is equal to $\mathrm{Var}(\varepsilon_i)$ and $\beta_j = (\beta_{j1},\ldots,\beta_{jn})^T$ for $j = 1,\ldots,m$. If C is diagonal, then (4.1) reduces to (1.5) mentioned in the introduction. Note that $(\beta_1,\ldots,\beta_m)$ is an $n \times m$ matrix.

In the sequel, we add the following assumption to the model:

(A) the diagonal entries of Σ are all positive, i.e. $\mathrm{Var}(\varepsilon_i) > 0$ for $i = 1,\ldots,n$.

Since the ε_i's can be viewed as the differences between the actual security returns and those obtained by taking linear combinations of the indices, the assumption (A) says that there are always errors in measuring the actual security returns by linear combinations of the indices. Fortunately, this is not a too unreasonable assumption because the coefficients of combinations, i.e. the β_{ij}'s, are usually obtained by statistical estimation based on past history in the stock market, thus are themselves subject to errors. Presumably, taking linear combinations involving these erroneous

β_{ij}'s would provide only approximate and not exact values of the actual security returns. Unfortunately, assumption (A) will exclude riskless securities in the model.

If the index models are of any significance in reducing the computational and informational complexity of the general portfolio analysis problem, it would seem very reasonable not to include too many indices in the models; or in other words, we may consider m relatively small compared with n. This latter condition is particularly meaningful when n is very large.

Under assumption (A), the matrix V given in (4.1) is symmetric and positive definite. Therefore the algorithms developed in Section 3 are applicable. Of course, (A) is only a sufficient condition for V to be positive definite. Making the change of variables,

$$z = \Sigma^{\frac{1}{2}} x$$

where $\Sigma^{\frac{1}{2}}$ is the diagonal matrix whose diagonal entries are the square roots of the corresponding diagonal entries of Σ, we may reformulate problem (1.2) ((1.7)) as

(4.2)
$$\begin{aligned} &\text{minimize} && \tfrac{1}{2} z^T M z + \theta r'^T z \\ &\text{subject to} && (a' \geq) z \geq 0 \\ &\text{and} && p'^T z = 1 \end{aligned}$$

where

(4.3) $$M = I + (\beta_1', \ldots, \beta_m') \, C (\beta_1', \ldots, \beta_m')^T$$

(4.4a) $$\beta_i' = \Sigma^{-\frac{1}{2}} \beta_i \qquad i = 1, \ldots, m$$

(4.4b) $$a' = \Sigma^{\frac{1}{2}} a$$

(4.4c) $$r' = \Sigma^{-\frac{1}{2}} r \quad \text{and} \quad p' = \Sigma^{-\frac{1}{2}} e .$$

If the n standard deviations $(\mathrm{Var}(\varepsilon_i)^{1/2})_{i=1}^n$ are given, then it requires $n \times m$ divisions to compute all the β_i's, n multiplications to obtain a' and $2n$ divisions to obtain r' and p'.

To facilitate practical computations, it is convenient to factorize C into

$$C = GG^T \tag{4.5}$$

where G is an $m \times m$ lower triangular matrix. The factorization is possible because C is symmetric and positive semi-definite. It can be achieved for example, by the scheme described in [7, p. 491-492]. The amount of operations (i.e. multiplications and additions) required by the scheme is $O(m^3)$. Since m is small, this factorization will, presumably, present no computational difficulty. With (4.5), we then compute the $n \times m$ matrix

$$\Gamma = (\beta_1', \ldots, \beta_m')G \ . \tag{4.6}$$

The computation of Γ requires $n \times m$ multiplications and additions. By (4.5) and (4.6), (4.3) can be rewritten as

$$M = I + \Gamma\Gamma^T \ . \tag{4.7}$$

As we shall see, there is no need to compute the product $\Gamma\Gamma^T$. The total amount of operations to transform the original problem (1.2) (or (1.7)) with a covariance matrix given by (4.1) to the equivalent problem (4.2) with the matrix M given by (4.7) is $O(n) + O(m \times n) + O(m^3)$.

Recall that we have mentioned some updating procedures in Section 2 which can be used to execute Step 1 in each of the four algorithms in Section 3. These procedures usually require $O(n^2)$ storage and are operating on matrices with varying sizes which sometimes can be very large. On the other hand, by exploiting the special structure of the matrix M given in (4.7), we can achieve enormous savings in both the storage and operations of the algorithms. More specifically, we can reduce total storage to $O(n) + O(m \times n) + O(m^2)$ and in Step 1 we need to operate on matrices of order m only. These are indeed savings because m is usually very small compared with n. In what follows, we illustrate how these savings can be achieved in the case with upper bounds and omit the easier case without the upper bounds.

In the sequel, we put subscripts in the identity matrix I to denote its dimensionalities and write $|S|$ to mean the cardinality of a set S. Let Γ be an $n \times m$ matrix and

$$M = I_n + \Gamma\Gamma^T . \tag{4.7}$$

Let p,q,r and a be n-vectors. Let I, J_1 and J_2 be subsets of $\{1,\ldots,n\}$, disjoint from each other and such that $I \cup J_1 \cup J_2 = \{1,\ldots,n\}$. Then we have

$$M_{J_1J_1} = I_{|J_1|} + \Gamma_{J_1\cdot}(\Gamma_{J_1\cdot})^T$$

whose inverse can be computed by the Sherman-Morrison-Woodbury formula ([11, p. 79])

$$M^{-1}_{J_1J_1} = I_{|J_1|} - \Gamma_{J_1\cdot}[I_m + (\Gamma_{J_1\cdot})^T(\Gamma_{J_1\cdot})]^{-1}(\Gamma_{J_1\cdot})^T .$$

For each $\delta \in R^{|J_1|}$, let σ^δ be the (unique) solution of

$$[I_m + (\Gamma_{J_1\cdot})^T(\Gamma_{J\cdot})]\sigma^\delta = (\Gamma_{J_1\cdot})^T\delta . \tag{4.8}$$

Now if f_{J_1}, g_{J_1} and h_{J_1} are the solutions of the systems of equations in (3.20), then

$$f_{J_1} = r_{J_1} - \Gamma_{J_1\cdot}\sigma^{r_{J_1}} \text{ and } h_{J_1} = p_{J_1} - \Gamma_{J_1\cdot}\sigma^{p_{J_1}} . \tag{4.9}$$

To compute g_{J_1}, it is necessary to evaluate $M^{-1}_{J_1J_1}M_{J_1J_2}a_{J_2}$. We have

$$M_{J_1J_2}a_{J_2} = \Gamma_{J_1\cdot}(\Gamma_{J_2\cdot})^T a_{J_2} .$$

Thus,

$$\begin{aligned} M^{-1}_{J_1J_1}M_{J_1J_2}a_{J_2} &= \Gamma_{J_1\cdot}(\Gamma_{J_2\cdot})^T a_{J_2} - \Gamma_{J_1}[I_m+(\Gamma_{J_1\cdot})^T(\Gamma_{J_1\cdot})]^{-1}(\Gamma_{J_1\cdot})^T(\Gamma_{J_1\cdot})(\Gamma_{J_2\cdot})^T a_{J_2} \\ &= \Gamma_{J_1\cdot}[I_m-\{I_m+(\Gamma_{J_1\cdot})^T(\Gamma_{J_1\cdot})\}^{-1}(\Gamma_{J_1\cdot})^T(\Gamma_{J_1\cdot})](\Gamma_{J_2\cdot})^T a_{J_2} \\ &= \Gamma_{J_1\cdot}[I_m+(\Gamma_{J_1\cdot})^T(\Gamma_{J_1\cdot})]^{-1}(\Gamma_{J_2\cdot})^T a_{J_2} . \end{aligned}$$

Letting $\tau^{a_{J_2}}$ be the (unique) solution to

$$[I_m + (\Gamma_{J_1.})^T (\Gamma_{J_1.})]\tau^{a_{J_2}} = (\Gamma_{J_2})^T a_{J_2} , \tag{4.10}$$

we deduce that

$$g_{J_1} = q_{J_1} - \Gamma_{J_1.} (\sigma^{q_{J_1}} - \tau^{a_{J_2}}) . \tag{4.11}$$

If s_{J_1}, t_{J_1}, c_I, d_I, c_{J_2} and d_{J_2} are the vectors given by (3.22), (3.23) and (3.24), they can be computed in the following way. By (4.8), (4.9) and (4.11), we have

$$\begin{aligned} s_{J_1} &= g_{J_1} + \bar{\lambda} h_{J_1} \\ &= (q_{J_1} + \bar{\lambda} p_{J_1}) - \Gamma_{J_1.} (\sigma^{q_{J_1}+\bar{\lambda}p_{J_1}} - \tau^{a_{J_2}}) \end{aligned} \tag{4.12a}$$

and

$$\begin{aligned} t_{J_1} &= f_{J_1} + \bar{\mu} h_{J_1} \\ &= (r_{J_1} + \bar{\mu} p_{J_1}) - \Gamma_{J_1.} (\sigma^{r_{J_1}+\bar{\mu}p_{J_1}}) . \end{aligned} \tag{4.12b}$$

Therefore,

$$\begin{aligned} & M_{IJ_2} a_{J_2} - M_{IJ_1} s_{J_1} \\ &= \Gamma_{I.} (\Gamma_{J_2.})^T a_{J_2} - \Gamma_{I.} (\Gamma_{J_1.})^T (q_{J_1}+\bar{\lambda}p_{J_1}) + \Gamma_I (\Gamma_{J_1.})^T \Gamma_{J_1.} (\sigma^{q_{J_1}+\bar{\lambda}p_{J_1}} - \tau^{q_{J_2}}) \\ &= \Gamma_{I.} [(\Gamma_{J_2.})^T a_{J_2} - (\Gamma_{J_1.})^T (\Gamma_{J_1.})\tau^{a_{J_2}} - (\Gamma_{J_1.})^T (q_{J_1}+\bar{\lambda}p_{J_1}) + (\Gamma_{J_1.})^T (\Gamma_{J_1.}) \sigma^{q_{J_1}+\bar{\lambda}p_{J_1}}] \\ &= -\Gamma_{I.} (\sigma^{q_{J_1}+\bar{\lambda}p_{J_1}} - \tau^{a_{J_2}}) \end{aligned}$$

where the last equality follows from (4.8) and (4.10). Hence,

$$c_I = (q_I + \bar{\lambda} p_I) - \Gamma_{I.} (\sigma^{q_{J_1}+\bar{\lambda}p_{J_1}} - \tau^{a_{J_2}}) . \tag{4.13a}$$

Similarly we may deduce

$$d_I = (r_I + \bar{\mu} p_I) - \Gamma_{I.} (\sigma^{r_{J_1}+\bar{\mu}p_I}) \tag{4.13b}$$

(4.14a) $$c_{J_2} = (q_{J_2} + \bar{\lambda}p_{J_2}) - \Gamma_{J_2.}(\sigma^{q_{J_1}+\bar{\lambda}p_{J_1}} - \tau^{a_{J_2}}) + a_{J_2}$$

(4.4b) $$d_{J_2} = (r_{J_2} + \bar{\mu}p_{J_2}) - \Gamma_{J_2.}(\sigma^{r_{J_1}+\bar{\mu}p_{J_2}}) \quad .$$

If we define the n-vectors s and t by

(4.15a) $$s = (q + \bar{\lambda}p) - \Gamma(\sigma^{q_{J_1}+\bar{\lambda}p_{J_1}} - \tau^{a_{J_2}})$$

(4.15b) $$t = (r + \bar{\mu}p) - \Gamma(\sigma^{r_{J_1}+\bar{\mu}p_{J_1}})$$

then we can immediately generate all the vectors $s_{J_1}, t_{J_1}, c_I, d_I, c_{J_2}$ and d_{J_2}. In practice the vectors $\sigma^{q_{J_1}}, \sigma^{r_{J_1}}, \sigma^{p_{J_1}}$ and $\tau^{a_{J_2}}$ are generated by solving the following three systems of equations

(4.16) $$[I_m + (\Gamma_{J_1.})^T(\Gamma_{J_1.})](\sigma^{q_{J_1}} - \tau^{a_{J_2}}, \sigma^{r_{J_1}}, \sigma^{p_{J_1}}) =$$

$$= ((\Gamma_{J_1.})^T a_{J_1} - (\Gamma_{J_2.})^T a_{J_2}, (\Gamma_{J_1.})^T r_{J_1}, (\Gamma_{J_1.})^T p_{J_1}) \quad .$$

Note that there is no need to generate $\sigma^{q_{J_1}}$ and $\tau^{a_{J_2}}$ separately. It is their difference that is required. Furthermore, the matrix $I_m + (\Gamma_{J_1.})^T(\Gamma_{J_1.})$ is symmetric, positive definite and of order m which is independent of $|J_1|$. Now, suppose that J_1 changes by one element, then it is easy to see that $I_m + (\Gamma_{J_1.})^T(\Gamma_{J_1.})$ is affected by a rank-one modification, i.e.

(4.17) $$\text{new matrix} = \text{old matrix} \pm \gamma\gamma^T$$

where γ^T is a suitable row in the matrix Γ. If we solve (4.16) by applying Cholesky factorization (see [11] e.g.) on the matrix $I_m + (\Gamma_{J_1.})^T(\Gamma_{J_1.})$, then we can update the Cholesky factors very quickly by taking advantage of the relation (4.17). An updating procedure is described in [8, Algorithm C1]. See also [9]. The number of operations necessary to compute the modified

factorization using the procedure is $m^2 + O(m)$ multiplications and $m^2 + O(m)$ additions [8]. Furthermore, the procedure is numerically stable because for every J_1, the matrix $I_m + (\Gamma_{J_1.})^T(\Gamma_{J_1.})$ is sufficiently positive definite, its smallest eigenvalue is greater than or equal to 1. We note that if $m = 1$, (corresponding to the single index model), then $I_m + (\Gamma_{J_1.})^T(\Gamma_{J_1.})$ becomes a scalar and the updating process reduces to trivial.

Based on the above analysis, we may now formulate the specializations of Algorithms 6 and 7 to problem (4.2) with the upper bounds. To keep the notations simple, we delete the primes in all the vectors in (4.2). Moreover we assume that the vector q does not vanish in the objective function of (4.2), i.e. we consider the objective function to be $\frac{1}{2} z^T M z + (q + \theta r)^T z$. Furthermore, we assume $p^T a \neq 1$ in order to eliminate the trivial case. Finally, we remark that the vectors $g_{J_1}, h_{J_1}, c_I, d_I, c_{J_2}$ and d_{J_2} required in Algorithm 6 can be obtained easily from (4.15) by some simple modifications. See (4.18) and (4.19) below.

Algorithm 8. Specialization of Algorithm 6 with M given by (4.7).

Step 0. Determine $\lambda_{old} = \max_{1 \le i \le n} \{-q_i/p_i\}$ and let k be a maximizing index. Let $I = \{1,\ldots,n\} \setminus \{k\}$, $J_1 = \{k\}$ and $J_2 = \phi$. Go to Step 1.

Step 1. Solve the systems of linear equations

$$[I_m + (\Gamma_{J_1.})^T(\Gamma_{J_1.})]\,(\sigma^{q_{J_1}} - \tau^{a_{J_2}}, \sigma^{p_{J_1}}) = \tag{4.18}$$

$$= ((\Gamma_{J_1.})^T q_{J_1} - (\Gamma_{J_2.})^T a_{J_2},\ (\Gamma_{J_1.})^T p_{J_1})$$

and compute

$$s = q - \Gamma(\sigma^{q_{J_1}} - \tau^{a_{J_2}}),\quad t = p - \Gamma\sigma^{p_{J_1}}\ . \tag{4.19}$$

Step 2. Determine

$$\lambda_{new} = \max\{\max\{-\frac{s_i}{t_i}: t_i > 0,\ i \in I\},\ \max\{-\frac{s_{j_1}}{t_{j_1}}: t_{j_1} < 0,\ j_1 \in J_1\}\ , \tag{4.20}$$

$$\max\{-\frac{a_{j_1}+s_{j_1}}{t_{j_1}} : t_{j_1} > 0,\ j_1 \in J_1\},\ \max\{-\frac{a_{j_2}+s_{j_2}}{t_{j_2}} : t_{j_2} < 0, j_2 \in J_2\}\}$$

Case (i) $J_1 \neq \phi$. Put

$$\bar{\lambda} = -(1 + p_{J_1}^T s_{J_1} - p_{J_2}^T a_{J_2})/p_{J_1}^T t_{J_1} \ .$$

If $\bar{\lambda} \in [\lambda_{new}, \lambda_{old}]$, set $\lambda^* = \bar{\lambda}$

$$\hat{x}_I^a(0) = 0,\ \hat{x}_{J_1}^a(0) = -s_{J_1} - \lambda^* t_{J_1} \text{ and } \hat{x}_{J_2}^a(0) = a_{J_2}$$

and terminate. Otherwise, let k be a maximizing index in (4.20). Set $\lambda_{old} = \lambda_{new}$ and go to Step 3.

Case (ii) $J_1 = \phi$. If $p_{J_2}^T a_{J_2} = 1$, set $\lambda^* = \lambda_{new}$.

$$\hat{x}_I^a(0) = 0 \text{ and } \hat{x}_{J_2}^a(0) = a_{J_2}$$

and terminate. Otherwise, let k be a maximizing in (4.20). Set $\lambda_{old} = \lambda_{new}$ and go to Step 3.

Step 3. If $k \in I$, replace I and J by $I \setminus \{k\}$ and $J_1 \cup \{k\}$ respectively. Go to Step 1.

If $k \in J_2$, replace J_1 and J_2 by $J_1 \cup \{k\}$ and $J_2 \setminus \{k\}$ respectively Go to Step 1 .

If $k \in J_1$ and $t_k < 0$, replace I and J_1 by $I \cup \{k\}$ and $J_1 \setminus \{k\}$ respectively. Go to Step 1.

If $k \in J_1$ and $t_k > 0$, replace J_1 and J_2 by $J_1 \setminus \{k\}$ and $J_2 \cup \{k\}$ respecively. Go to Step 1.

Algorithm 9: Specialization of Algorithm 7 with M given by (4.7).

Step 0. Use Algorithm 8 to obtain $\hat{x}^a(0)$ and let I, J_1 and J_2 be the three index sets at termination of the algorithm. Let $\theta_{old} = 0$. If

$J_1 = \phi$ set $s = q + \lambda^* p$ and $t = r$, go to Step 2. Otherwise go to Step 1.

Step 1. Solve the systems of equations in (4.16). Compute

$$\eta^T = p_{J_1}^T \Gamma_{J_1}.$$

$$PG = p_{J_1}^T q_{J_1} - \eta^T (\sigma^{q_{J_1}} - \tau^{q_{J_2}})$$

$$PF = p_{J_1}^T r_{J_1} - \eta^T \sigma^{r_{J_1}}$$

$$PH = p_{J_1}^T p_{J_1} - \eta^T \sigma^{p_{J_1}}$$

$$\bar{\lambda} = -(1 + PG - p_{J_2}^T a_{J_2})/PH, \quad \bar{\mu} = -PF/PH$$

and the vectors s and t by (4.15).

Step 2. If $t_I \geq 0$, $t_{J_1} = 0$, $t_{J_2} \leq 0$ set $\hat{x}_I^a(\theta) = 0$, $\hat{x}_{J_1}^a(\theta) = -s_{J_1}$ and $\hat{x}_{J_2}^a(\theta) = a_{J_2}$ for $\theta \geq \theta_{old}$ and terminate. Otherwise determine

$$\theta_{new} = \min\{\min\{-\frac{s_i}{t_i}: t_i < 0, i \in I\}, \min\{-\frac{s_{j_1}}{t_{j_1}}: t_{j_1} > 0, j_1 \in J_1\},$$

$$\min\{-\frac{a_{j_1} + s_{j_1}}{t_{j_1}}: t_{j_1} < 0, j_1 \in J_1\}, \min\{-\frac{a_{j_2} + s_{j_2}}{t_{j_2}}: t_{j_2} > 0, j_2 \in J_2\}\}$$

and let k be a minimizing index. Put $\hat{x}_I^a(\theta) = 0$, $\hat{x}_{J_1}^a(\theta) = -s_{J_1} - t_{J_1}$ and $\hat{x}_{J_2}^a(\theta) = a_{J_2}$ for $\theta \in [\theta_{old}, \theta_{new}]$. Set $\theta_{old} = \theta_{new}$ and go to Step 3.

Step 3. Same as Step 3 in Algorithm 7.

The systems of linear equations in (4.16) or (4.18) should best be solved by Cholesky factorization together with Algorithm C1 in [8] to update the Cholesky factors. Because of the fact that J_1 starts initially in

Algorithm 8 as a singleton, there is no need to compute the product $(\Gamma_{J_1.})^T(\Gamma_{J_1.})$ throughout. Finally, whenever J_1 becomes a singleton, it is more convenient to compute the Cholesky factors of the matrix $I_m + (\Gamma_{J_1.})^T(\Gamma_{J_1.})$ from scratch. Formulas for computing such factors are given in [8].

Of course, the above two algorithms are applicable to the general portfolio analysis problem (1.7) with the covariance matrix V given by (4.1), provided that we first obtain the transformed problem (4.2) with the transformed data (4.3) and (4.4) by the process described earlier. The total storage required in this application consists of the n standard deviations $\text{Var}(\varepsilon_i)^{1/2}$, the $m \times m$ covariance matrix C, the $m \times m$ matrix G, (C and G each require $\frac{m(m+1)}{2}$ storage because of symmetry), the $n \times m$ matrices $(\beta_1, \ldots, \beta_m)$ and Γ, the n-vectors r,a and x, finally, some $\frac{m(m+1)}{2}$ storage for the Cholesky factors. These add up to $O(n) + O(m^2) + O(n \times m)$. This latter amount is best described as minimal.

We conclude this section by remarking that the matrix V given by (4.1) appeared also in an application of the parametric linear complementarity problem to structural engineering. In fact, it was shown in [15] (see also the references there) that the behavior of a reinforced concrete beam can be described in terms of the parametric linear complementarity problem $\{(q + \lambda p, M): \lambda \in R\}$ where M is precisely of the form (4.1). Of course, Algorithm 2 is applicable in this instance. Moreover, the analysis developed in this section can be used to increase the overall efficiency of the algorithm. See [23] for a more detailed discussion of this subject.

5. CONCLUSION

In this paper, we have developed several new algorithms for portfolio analysis and discussed their efficient implementation. We have also shown how their specializations to the index models will result in dramatic savings in storage and computations. In a forthcoming paper [23], we shall report our computational experience with the algorithms in solving some portfolio problems.

REFERENCES

[1] Alexander, G. J., "The derivation of efficient sets", Journal of Financial and Quantitative Analysis, December 1976, 817-829.

[2] Cottle, R. W., "Principal pivoting method for quadratic programming", Mathematics of the Decision Sciences, Part I (G. B. Dantzig and A. F. Veinott, Jr. eds.) Amer. Math. Society, Providence, R. I. 1968.

[3] Cottle, R. W., "Complementarity and variational inequalities", Tech. Report SOL 74-6, Systems Optimization Laboratory, Dept. of Operations Research, Stanford University, May 1974, to appear in Symposia Mathematica.

[4] Cottle, R. W., "Monotone solutions of the parametric linear complementarity problem", Mathematical Programming, 3 (1972), 210-224.

[5] Cottle, R. W. and G. B. Dantzig, "Complementarity pivot theory of mathematical programming", Linear Algebra and Its Applications, 1 (1968), 103-125.

[6] Cottle, R. W. and M. Goheen, "On a special class of large quadratic programs", Tech. Report SOL 76-7, Systems Optimization Laboratory, Dept. of Operations Research, Stanford University, March 1976.

[7] Dantzig, G. B., "Linear programming and extension", Princeton University Press, 1963.

[8] Gill, P. E., G. H. Golub, W. Murray and M. A. Saunders, "Methods for modifying matrix factorizations", Mathematics of Computation, 28 (1974), 505-535.

[9] Gill, P. E., W. Murray and M. A. Saunders, "Methods for computing and modifying the LDV factors of a matrix", Mathematics of Computation, 29 (1975), 1051-1077.

[10] Graves, R. L., "A principal pivoting simplex algorithm for linear and quadratic programming", Operations Research, 15 (1967), 482-494.

[11] Householder, A. S., "Principles of numerical analysis", McGraw-Hill Book Company, Inc., 1953.

[12] International Business Machines, Inc., "1401 Portfolio selection program (1401-F1-04X) program reference manual", New York: International Business Machines, Inc., 1965.

[13] Jucker, J. V. and C. de Faro, "A simple algorithm for Stone's version of the portfolio selection problem", Journal of Financial and Quantitative Analysis, December 1975, 859-870.

[14] Kaneko, I., "Linear complementarity problems and characterizations of Minkowski matrices", Tech. Report 76-9, Dept. of Industrial Engineering, University of Wisconsin-Madison, July 1976.

[15] Kaneko, I., "The parametric linear complementarity problem in the De Donato-Maier analysis of reinforced concrete beams", Tech. Report SOL 75-13, Dept. of Operations Research, Stanford University, May 1975.

[16] Kaneko, I., "The linear complementarity problem with a n by 2n "P"-matrix", Technical report 76-4, Dept. of Industrial Engineering, University of Wisconsin-Madison, February 1976.

[17] Lemke, C. E., "Bimatrix equilibrium points and mathematical programming", Management Science, 11 (1965), 681-689.

[18] Lemke, C. E., "Recent results on complementarity problems", in Symposium on Nonlinear Programming, ed. by J. B. Rosen, O. L. Mangasarian and K. Ritter, Acad. Press 1970.

[19] Markowitz, H., "Portfolio selection", The Journal of Finance, March 1952, 77-91.

[20] Markowitz, H., "The optimization of a quadratic function subject to linear constraints", Naval Research Logistics Quarterly, March-June 1965, 111-133.

[21] Pang, J. S., "On a class of least-element complementarity problems", Tech. Report SOL 76-10, Systems Optimization Laboratory, Dept. of Operations Research, Stanford University, June 1976.

[22] Pang, J. S., "Least element complementarity theory", Ph.D. dissertation, Dept. of Operations Research, Stanford University, September 1976.

[23] Pang, J. S. and I. Kaneko, "On solving some parametric linear complementarity problems arising from portfolio analysis and structural engineering", forthcoming.

[24] Rosenberg, B. and A. Rudd, "Portfolio optimization algorithms: A progress report", Working paper No. 42, Research Program in Finance, Graduate School of Business Administration, University of California, Berkeley.

[25] Samelson, H., R. M. Thrall and O. Wesler, "A partitioning theorem for Euclidean n-space", Proceedings Amer. Math. Soc., 9 (1958), 805-807.

[26] Sargent, R. W. H., "An efficient implementation of the Lemke algorithm and its entension to deal with upper and lower uppers", paper presented at the IXth International Symposium on Mathematical Programming, Budapest, August 1976.

[27] Sargent, R. W. H. and B. A. Murtagh, "Projection methods for nonlinear programming", Mathematical Programming, 4 (1973), 245-268.

[28] Sharpe, W. F., "A simplified model for portfolio analysis", Managemnt Science, January 1963, 277-293.

[29] Sharpe, W. F., "A linear programming algorithm for mutual fund portfolio selection", Management Science, March 1967, 499-510.

[30] Sharpe, W. F., "Portfolio theory and capital markets", New York, McGraw-Hill Book Company, Inc., 1970.

[31] Sharpe, W. F., "A linear programming approximation for the general portfolio analysis problem", Journal of Financial and Quantitative Analysis, December 1971, 1263-1275.

[32] Stone, B. K., "A linear programming formulation of the general portfolio analysis problem", Journal of Financial and Quantitative Analysis, September 1973, 621-636.

[33] Von Hohenbalken, B., "A finite algorithm to maximize certain pseudo-concave functions on polytopes", Research Paper No. 18, Edmonton: Department of Economics, University of Alberta (March). To appear in Mathematical Programming.

[34] Williamson, J. P. and D. H. Downes, "Manuals for computer programs in finance and investments", Hanover, New Hampshire: Dartmouth College, 1970.

[35] Wolfe, P., "The simplex method for quadratic programming", Econometrica, 27 (1959), 382-398.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 1738	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SOME NEW AND EFFICIENT ALGORITHMS FOR PORTFOLIO ANALYSIS		5. TYPE OF REPORT & PERIOD COVERED Summary Report - no specific reporting period 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Jong-Shi Pang		8. CONTRACT OR GRANT NUMBER(s) DAAG29-75-C-0024 MCS75-17385
9. PERFORMING ORGANIZATION NAME AND ADDRESS Mathematics Research Center, University of Wisconsin 610 Walnut Street Madison, Wisconsin 53706		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Work Unit Number 5 Math. Prog. and Oper. Res.
11. CONTROLLING OFFICE NAME AND ADDRESS See Item 18 below.		12. REPORT DATE March 1977 13. NUMBER OF PAGES 44
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

U.S. Army Research Office
P.O. Box 12211
Research Triangle Park
North Carolina 27709

National Science Foundation
Washington, D. C. 20550

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Portfolio analysis, index models, mathematical programming, factorization

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper first describes a modified version of the parametric principal pivoting algorithm [4] for the class of parametric linear complementarity problems with P-matrices. The modified version of the parametric principal pivoting algorithm is then used to develop a new and efficient algorithm to solve the class of portfolio analysis problems with positive definite (or equivalently, nonsingular) covariance matrices. Extension of the new algorithm to handle explicit upper-bounds is also established. The new algorithm and its

next page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

extension are then specialized to the "index models" introduced by Sharpe [28], [29]. In these specializations, the algorithms are particularly effective, achieving dramatic savings in both storage and computations.