MANAGEMENT SCIENCE
Vol. 51, No. 11, November 2005, pp. 1676–1688
ISSN 0025-1909 | EISSN 1526-5501 | 05 | 5111 | 1676

DOI 10.1287/mnsc.1050.0418

An Algorithm for Portfolio Optimization with Transaction Costs

Michael J. Best
Department of Combinatorics and Optimization, Faculty of Mathematics, University of Waterloo, Waterloo, Ontario, N2L 3G1 Canada, mjbest@math.uwaterloo.ca

Jaroslava Hlouskova
Department of Economics and Finance, Institute for Advanced Studies, Stumpergasse 56, A-1060 Vienna, Austria, jaroslava.hlouskova@ihs.ac.at

We consider the problem of maximizing an expected utility function of n assets, such as the mean-variance or power-utility function. Associated with a change in an asset's holdings from its current or target value is a transaction cost. This cost must be accounted for in practical problems. A straightforward way of doing so results in a $3n$-dimensional optimization problem with $3n$ additional constraints. This higher dimensional problem is computationally expensive to solve. We present a method for solving the $3n$-dimensional problem by solving a sequence of n-dimensional optimization problems, which accounts for the transaction costs implicitly rather than explicitly. The method is based on deriving the optimality conditions for the higher-dimensional problem solely in terms of lower-dimensional quantities. The new method is compared to the barrier method implemented in Cplex in a series of numerical experiments. With small but positive transaction costs, the barrier method and the new method solve problems in roughly the same amount of execution time. As the size of the transaction cost increases, the new method outperforms the barrier method by a larger and larger factor.

Key words: convex programming; portfolio optimization; transaction costs
History: Accepted by Thomas M. Liebling, mathematical programming and networks; received September 20, 2001. This paper was with the authors 2 years and 5 months for 6 revisions.

1. Introduction

In a financial setting, holdings of assets may be bought or sold according to some criterion, typically the investor's utility function. Two well-known such utility functions are the mean-variance utility function (Bodie et al. 1999, Luenberger 1998, Markowitz 1959) and the power-utility function (Grauer and Hakansson 1993). In response to changed data, the utility function implies a change should be made in an investor's holdings. In practical situations, there is a cost associated with buying or selling an asset. This is called the transaction cost.

Intuitively, if a transaction cost for a particular asset is very large, it may not be advantageous to change the holdings of that asset, and the holdings of that asset remain at their initial values. Alternatively, if the transaction cost is quite small, it may be advantageous to make the trade and incur the transaction cost.

A solution for a portfolio optimization problem with linear transaction costs is given in Best and Hlouskova (2003). Their model problem assumes a diagonal covariance matrix, the budget constraint and upper bounds on all assets' holdings. See also Schattman (2000) for an overview of transaction costs in a variety of settings.

In this paper we present an algorithm to maximize an expected utility function of n asset holdings while accounting for transaction costs. As we shall show, transaction costs can be accounted for by solving an optimization problem with $2n$ additional variables and $3n$ additional linear constraints. However, the method we present will require the solution of a number of n-dimensional problems without the additional linear constraints and thus with corresponding savings in computer time and storage. The key idea is to treat the transaction costs *implicitly* rather than explicitly.

Throughout this paper prime ($'$) denotes transposition. All vectors are column vectors unless primed. The notation z_i or $(z)_i$ will be used to denote the ith component of the vector z.

Consider the following problem

$$\begin{aligned} &\text{minimize:} && f(x) \\ &\text{subject to:} && Ax \le b, \quad d \le x \le e, \end{aligned} \tag{1.1}$$

where A is an $m \times n$ matrix, b is an m-vector, x is an n-vector of asset holdings, d and e are n-vectors of lower and upper bounds on x, respectively, and $-f(x)$ is an expected utility function.[1] The constraints $Ax \le b$

[1] We prefer a minimization model. Minimizing $f(x)$ gives the same solution as maximizing $-f(x)$.

represent general linear constraints on the asset holdings, and the constraints $d \le x \le e$ impose explicit bounds on the asset holdings. Let

$$S \equiv \{x \in \Re^n \mid Ax \le b,\ d \le x \le e\} \tag{1.2}$$

denote the feasible region for (1.1), where $\Re^n$ is the set of all real n-vectors. We assume that a target n-vector $\hat{x}$ is given. $\hat{x}$ could represent the current holdings of assets so that purchase transaction costs are measured by the amount increased from $\hat{x}$ and sales transaction costs are measured by a decrease from $\hat{x}$. Alternatively, $\hat{x}$ could represent an index fund being followed, such as the Standard & Poor's 500.

In optimizing the expected utility function with no transaction costs, there is the possibility of making a large change in the holdings with only a small change in the utility function. The actual transaction cost for such a large move would be quite significant with very little gain in utility. The inclusion of transaction costs in the model precludes this possibility.

Let x^+ and x^- denote n-vectors of amounts purchased and sold, respectively. These quantities are related to the actual holdings x according to

$$x = \hat{x} + x^+ - x^-, \quad x^+ \ge 0,\ x^- \ge 0. \tag{1.3}$$

For $i = 1, \ldots, n$ we assume the purchase cost for x_i^+ is given by $p_i(x_i^+)$ and the sales cost for x_i^- is given by $q_i(x_i^-)$. Thus, the transaction costs are assumed to be separable. In addition, we shall assume that each $p_i(x_i^+)$ and $q_i(x_i^-)$ is a convex and increasing[2] function of its argument for all $x_i^+ \ge 0$ and $x_i^- \ge 0$, respectively. The total cost of the transactions is thus

$$p(x^+) + q(x^-),$$

where $p(x^+) = \sum_{i=1}^n p_i(x_i^+)$ and $q(x^-) = \sum_{i=1}^n q_i(x_i^-)$.

For the case of a mean-variance utility function we have

$$f(x) = -t\mu' x + \tfrac{1}{2} x' C x,$$

where μ is an n-vector of expected returns, C is an $n \times n$ covariance matrix and t is a fixed scalar parameter.[3] The expected return of the portfolio is $\mu' x$ and its variance is $x'Cx$. One way to incorporate transaction costs into this model is to reduce the portfolio expected return by the transaction cost. This approach was taken in Best and Kale (2000). The problem to be solved is then

$$\begin{aligned} \text{minimize:} \quad & -t\mu' x + \tfrac{1}{2} x' C x + tp(x^+) + tq(x^-) \\ \text{subject to:} \quad & x - x^+ + x^- = \hat{x}, \quad Ax \le b, \\ & d \le x \le e, \\ & x^+ \ge 0, \quad x^- \ge 0. \end{aligned} \tag{1.4}$$

For the case of the power-utility function (Grauer and Hakansson 1993) for a single period,

$$f(x) = -\sum_s \pi_s \frac{1}{\gamma}(1 + r_s' x)^\gamma, \tag{1.5}$$

where r_s is the vector of returns for state s, π_s is the probability that state s occurs, and γ is a parameter such that $\gamma < 1$. As γ tends to zero, the power-utility function tends to the logarithmic utility function.

The Taylor's series for (1.5) taken about the origin is

$$f(x) = -\frac{1}{\gamma}\sum_s \pi_s - \sum_s \pi_s r_s' x + O(\|x\|^2).$$

The linear part of this is the negative of the expected return of the portfolio. Similar to the mean-variance case we can incorporate transaction costs into this model by reducing the expected return with the transaction costs. The problem to be solved is

$$\begin{aligned} \text{minimize:} \quad & -\sum_s \pi_s \frac{1}{\gamma}(1 + r_s' x)^\gamma + p(x^+) + q(x^-) \\ \text{subject to:} \quad & x - x^+ + x^- = \hat{x}, \quad Ax \le b, \\ & d \le x \le e, \\ & x^+ \ge 0, \quad x^- \ge 0. \end{aligned} \tag{1.6}$$

We assume that the general constraints $Ax \le b$ incorporate constraints $1 + r_s' x \ge \epsilon_s$, $\epsilon_s > 0$, with ϵ_s sufficiently small. This implies that the expected power-utility function is well defined (see Grauer and Hakansson 1993).

More generally, let $-f(x)$ be any concave twice-differentiable expected utility function such that $-\nabla f(0)' x$ is a nonnegative multiple θ of the expected return of the portfolio. Then the corresponding version of (1.4) or (1.6) will have the same constraints and the objective function

$$f(x) + \theta p(x^+) + \theta q(x^-), \tag{1.7}$$

where the transaction costs have the interpretation of decreasing the expected return of the portfolio. Note that both the mean-variance and power-utility functions satisfy this property.

A basic difficulty of incorporating transaction costs is that it triples the number of problem variables

[2] One of the authors is a consultant to several North American financial institutions for algorithms to solve portfolio optimization problems. In his experience, increasing piecewise linear transaction costs (which are convex) are quite popular in practice.

[3] As t is varied from 0 to $+\infty$, the plot of the corresponding optimal portfolio means and variances gives the well-known efficient frontier.

and requires the addition of $3n$ linear constraints. This gives an optimization problem that is considerably more time consuming to solve. In §2, we will derive relationships between the n- and $3n$-dimensional problems expressed solely in terms of n-dimensional quantities. In §3, we will use the analytical results of §2 to derive an algorithm for the solution of the $3n$-dimensional problem solely in terms of n-dimensional quantities and discuss its efficiency compared to any algorithm applied directly to the $3n$-dimensional problem. Section 4 presents computational results, and §5 gives conclusions.

2. Reduction of Problem Dimensionality

The addition of transaction costs to (1.1) resulted in the two example Problems (1.4) and (1.6) being $3n$-dimensional. Solving a $3n$-dimensional problem is computationally considerably more expensive than solving an n-dimensional problem, and in practical problems n can be quite large. It is the purpose of this section to establish relationships between the n- and $3n$-dimensional problems. The major result of this section is Theorem 2.1, which establishes optimality conditions for the $3n$-dimensional problem solely in terms of n-dimensional quantities.

Each of (1.4) and (1.6) is a special case of the following $3n$-dimensional model problem

$$\begin{aligned} \text{minimize:}\quad & f(x)+p(x^+)+q(x^-) \\ \text{subject to:}\quad & x-x^++x^-=\hat{x}, \quad Ax\le b, \\ & d\le x\le e, \\ & x^+\ge 0, \quad x^-\ge 0, \end{aligned} \tag{2.1}$$

where A, b, x, d, and e are as in (1.1) and $\hat{x}, x^+, x^-\in\Re^n$. We shall use (2.1) as our model problem and in §3 develop a solution algorithm for it. Note that $\theta p(x^+)$ and $\theta q(x^-)$ in (1.7) have been replaced by $p(x^+)$ and $q(x^-)$ in (2.1). Multiplication by a positive scalar will not change the assumptions required below.

Our analysis will require the following two assumptions.

Assumption 2.1. $d<\hat{x}<e$.

Assumption 2.2. *Let x, x^+, and $x^-\in\Re^n$.*

(i) *$f(x)$ is a twice differentiable convex function;*

(ii) *$p(x^+)$ and $q(x^-)$ are separable functions; that is, $p(x^+)=\sum_{i=1}^n p_i(x_i^+)$ and $q(x^-)=\sum_{i=1}^n q_i(x_i^-)$;*

(iii) *$p_i(x_i^+)$ and $q_i(x_i^-)$ are convex functions of a single variable for $i=1,\ldots,n$;*

(iv) *p, q are twice differentiable functions;*

(v) *$\nabla p(x^+)\ge 0$ and $\nabla q(x^-)\ge 0$; that is, $p_i(x_i^+)$ and $q_i(x_i^-)$ are increasing functions for $i=1,\ldots,n$.*

Definition 2.1. Let $x\in\Re^n$. The index sets of x are defined as follows:

$$\begin{aligned} I^+(x)&=\{i\mid x_i>\hat{x}_i\} \\ I(x)&=\{i\mid x_i=\hat{x}_i\} \\ I^-(x)&=\{i\mid x_i<\hat{x}_i\}. \end{aligned}$$

Definition 2.2. Let $x, y\in\Re^n$.

(i) $x^+\in\Re^n$ is the positive portion of x with respect to y if

$$x_i^+=\begin{cases} x_i-y_i, & i \text{ such that } x_i\ge y_i, \\ 0, & i \text{ such that } x_i<y_i. \end{cases}$$

(ii) $x^-\in\Re^n$ is the negative portion of x with respect to y if

$$x_i^-=\begin{cases} 0, & i \text{ such that } x_i>y_i, \\ y_i-x_i, & i \text{ such that } x_i\le y_i. \end{cases}$$

Note that if x^+ and x^- are the positive and negative portions of x with respect to $\hat{x}$, then $(x', (x^+)', (x^-)')'$ satisfies the constraints $x^+\ge 0$, $x^-\ge 0$, and $x-x^++x^-=\hat{x}$.

We will formulate an algorithm for the solution of (2.1) in terms of solving a sequence of subproblems. The subproblems to be solved depend on two n-vectors $\tilde{d}$ and $\tilde{e}$ as follows:

$$\text{SUB}(\tilde{d},\tilde{e}):\ \min\{f(x)+\tilde{c}(\tilde{d},\tilde{e},x)\mid Ax\le b,\ \tilde{d}\le x\le\tilde{e}\}.$$

The vectors $\tilde{d}$ and $\tilde{e}$ are to be specified. They will always satisfy

$$\begin{aligned} &\tilde{d}_i=d_i,\ \tilde{e}_i=\hat{x}_i \quad\text{or}\quad \tilde{d}_i=\hat{x}_i,\ \tilde{e}_i=e_i, \quad\text{or} \\ &\qquad\qquad \tilde{d}_i=\tilde{e}_i=\hat{x}_i. \end{aligned} \tag{2.2}$$

In addition, $\tilde{c}(\tilde{d},\tilde{e},x)=\sum_{i=1}^n\tilde{c}_i(\tilde{d}_i,\tilde{e}_i,x_i)$, where for $i=1,\ldots,n$

$$\tilde{c}_i(\tilde{d}_i,\tilde{e}_i,x_i)=\begin{cases} p_i(x_i-\hat{x}_i), & \text{if } \tilde{d}_i=\hat{x}_i \text{ and } \tilde{e}_i=e_i, \\ 0, & \text{if } \tilde{d}_i=\tilde{e}_i=\hat{x}_i, \\ q_i(\hat{x}_i-x_i), & \text{if } \tilde{d}_i=d_i \text{ and } \tilde{e}_i=\hat{x}_i. \end{cases} \tag{2.3}$$

The subproblem SUB$(\tilde{d},\tilde{e})$ is an n-dimensional problem with linear constraints and a convex, twice-differentiable, nonlinear objective function. There are many algorithms with demonstrably rapid convergence rates to solve it. See, for example, Best and Ritter (1976).

Remark 2.1. (a) The feasible region for SUB$(\tilde{d},\tilde{e})$ is a compact set. From Assumption 2.2(i)(iv), the objective function for SUB$(\tilde{d},\tilde{e})$ is continuous. These two facts imply the existence of a solution for SUB$(\tilde{d},\tilde{e})$.

(b) Note that for (2.1) the upper and lower bounds on x and the continuity of $f(x)$ (from Assumption 2.2(i)) imply $f(x)$ is bounded from below over the feasible region of (2.1). Furthermore, from Assumption 2.2(ii), (v), $p(x^+) + q(x^-)$ is also bounded from below over the feasible region of (2.1). Therefore, the objective function for (2.1) is bounded from below over the feasible region of (2.1).

Let $(x', (x^+)', (x^-)')'$ be any feasible solution for (2.1). Then $(x', (x^+ + \delta l)', (x^- + \delta l)')'$, where δ is any nonnegative scalar and l is the n-vector of ones, is also a feasible solution for (2.1). The following lemma shows that if (2.1) possesses an optimal solution, then there is an optimal solution for (2.1), for which no pair of components can be strictly positive.

Let $G(x, x^+, x^-) \equiv f(x) + p(x^+) + q(x^-)$ denote the objective function for (2.1).

LEMMA 2.1. *Let Assumption* 2.2(ii)(v) *be satisfied and assume* (2.1) *possesses at least one optimal solution. Then there exists an optimal solution* $(x', (x^+)', (x^-)')'$ *for* (2.1) *such that both* x_k^+ *and* x_k^- *can not be strictly positive for* $k = 1, \ldots, n$.

PROOF. Let $(y', (y^+)', (y^-)')'$ be any optimal solution for (2.1). Suppose for some k with $1 \le k \le n$ that

$$y_k^+ \ge y_k^- > 0. \tag{2.4}$$

Let $(x', (x^+)', (x^-)')'$ be obtained from $(y', (y^+)', (y^-)')'$ as follows: $x = y$ and $x_i^+ = y_i^+$, $x_i^- = y_i^-$ for all $i = 1, \ldots, n$, $i \ne k$, $x_k^+ = y_k^+ - y_k^-$, $x_k^- = 0$. Clearly, $(x', (x^+)', (x^-)')'$ is feasible for (2.1). Then from Assumption 2.2(ii) it follows that

$$G(x, x^+, x^-) = G(y, y^+, y^-) + (p_k(y_k^+ - y_k^-) - p_k(y_k^+)) + (q_k(0) - q_k(y_k^-)). \tag{2.5}$$

From Assumption 2.2(v)

$$p_k(y_k^+ - y_k^-) - p_k(y_k^+) \le 0 \quad \text{and} \quad q_k(0) - q_k(y_k^-) \le 0. \tag{2.6}$$

There are two cases to be considered.

Case 1. $p_k(y_k^+ - y_k^-) - p_k(y_k^+) < 0$ or $q_k(0) - q_k(y_k^-) < 0$. In this case it follows from (2.5) and (2.6) that $G(x, x^+, x^-) < G(y, y^+, y^-)$, which contradicts the optimality of $(y', (y^+)', (y^-)')'$. Thus, Case 1 cannot occur. From (2.6), the remaining possibility is Case 2.

Case 2. $p_k(y_k^+ - y_k^-) - p_k(y_k^+) = 0$ and $q_k(0) - q_k(y_k^-) = 0$. In this case $G(x, x^+, x^-) = G(y, y^+, y^-)$ and the modified solution is an alternate optimal solution. Furthermore, the modified solution has the property that not both x_k^+ and x_k^- can be strictly positive.

This modification can be performed on all other components, which satisfy (2.4). Furthermore, an analogous modification can be used for the case $y_k^- \ge y_k^+ > 0$. Repeated use of these modifications will produce an optimal solution having the property in the statement of the lemma. This completes the proof of the lemma. □

DEFINITION 2.3. If $(x', (x^+)', (x^-)')'$ is a feasible solution for (2.1) and satisfies the property that not both of x_i^+, x_i^- are strictly positive for $i = 1, \ldots, n$ then we call $(x', (x^+)', (x^-)')'$ a proper feasible solution. Additionally, if $(x', (x^+)', (x^-)')'$ is a proper feasible solution, which is optimal for (2.1), then we call $(x', (x^+)', (x^-)')'$ a proper optimal solution.

In light of Lemma 2.1, we restrict our search for an optimal solution for (2.1) to a proper optimal solution.

LEMMA 2.2. *Let* $(x', (x^+)', (x^-)')'$ *have the following properties*: (i) $x_i^+ x_i^- = 0$, *for* $i = 1, \ldots, n$; (ii) $x - x^+ + x^- = \hat{x}$; (iii) $x^+ \ge 0$, $x^- \ge 0$. *Then* x^+ *and* x^- *are positive and negative portions of* x *with respect to* $\hat{x}$, *respectively.*

PROOF. Let $(x', (x^+)', (x^-)')'$ have Properties (i)–(iii) stated in Lemma 2.2. Let $k \in I^+(x)$. Then according to Definition 2.1, $x_k > \hat{x}_k$. From Properties (ii) and (iii) it follows that $x_k^+ > 0$. This last and Property (i) imply

$$x_k^- = 0, \quad \text{for } k \in I^+(x). \tag{2.7}$$

It follows from (2.7) and Property (ii) that

$$x_k^+ = x_k - \hat{x}_k, \quad \text{for } k \in I^+(x). \tag{2.8}$$

Analogously, it can be shown that

$$x_k^+ = 0 \quad \text{and} \quad x_k^- = \hat{x}_k - x_k, \quad \text{for } k \in I^-(x). \tag{2.9}$$

Let $k \in I(x)$; that is, according to Definition 2.1, $x_k = \hat{x}_k$. This and Property (ii) imply that $x_k^+ = x_k^-$. From this last and Property (i) it follows that

$$x_k^+ = x_k^- = 0, \quad \text{for } k \in I(x). \tag{2.10}$$

Finally, it follows from Definition 2.2 and (2.7)–(2.10) that x^+ and x^- are the positive and negative portions of x with respect to $\hat{x}$, respectively. □

COROLLARY 2.1. *If* $(x', (x^+)', (x^-)')'$ *is a proper feasible solution for* (2.1), *then* x^+ *and* x^- *are the positive and negative portions of* x *with respect to* $\hat{x}$, *respectively.*

PROOF. This follows directly from Definition 2.3 and Lemma 2.2. □

LEMMA 2.3. *If* x^* *is an optimal solution for* SUB$(\tilde{d}, \tilde{e})$, *then* x^* *is optimal solution for* SUB$(\hat{d}, \hat{e})$, *where*

$$\hat{d}_i = \begin{cases} \tilde{d}_i, & i \in \{1, \ldots, n\} - (I(x^*) - J), \\ \hat{x}_i, & i \in I(x^*) - J, \end{cases} \tag{2.11}$$

$$\hat{e}_i = \begin{cases} \tilde{e}_i, & i \in \{1, \ldots, n\} - (I(x^*) - J), \\ \hat{x}_i, & i \in I(x^*) - J, \end{cases} \tag{2.12}$$

and

$$J = \{i \mid \tilde{d}_i = \tilde{e}_i = \hat{x}_i\}. \tag{2.13}$$

PROOF. Let x^* be an optimal solution for SUB($\tilde{d}, \tilde{e}$) and $\tilde{S}$ and $\hat{S}$ be sets of feasible solutions of SUB($\tilde{d}, \tilde{e}$) and SUB($\hat{d}, \hat{e}$), respectively; that is,

$$\begin{aligned} \tilde{S} &= \{x \mid Ax \le b,\ \tilde{d} \le x \le \tilde{e}\}, \\ \hat{S} &= \{x \mid Ax \le b,\ \hat{d} \le x \le \hat{e}\}. \end{aligned} \tag{2.14}$$

Let $\tilde{F}(x)$ be the objective function of SUB($\tilde{d}, \tilde{e}$) and $\hat{F}(x)$ be the objective function of SUB($\hat{d}, \hat{e}$). Then according to (2.3), (2.11), and (2.12),

$$\tilde{F}(x) = \hat{F}(x) + \sum_{i \in \tilde{J}_1} p_i(x_i - \hat{x}_i) + \sum_{i \in \tilde{J}_2} q_i(\hat{x}_i - x_i), \tag{2.15}$$

where $\tilde{J}_1 = \{i \mid i \in I(x^*) - J,\ \tilde{d}_i = \hat{x}_i,\ \tilde{e}_i = e_i\}$ and $\tilde{J}_2 = \{i \mid i \in I(x^*) - J,\ \tilde{d}_i = d_i,\ \tilde{e}_i = \hat{x}_i\}$.

From the definitions of $\hat{d}$ and $\hat{e}$ (see (2.11)–(2.13)) and because x^* is an optimal solution for SUB($\tilde{d}, \tilde{e}$), then

$$\hat{S} \subset \tilde{S}.$$

This last and the fact that x^* is an optimal solution for SUB($\tilde{d}, \tilde{e}$) imply that

$$\tilde{F}(x^*) \le \tilde{F}(x), \quad \text{for all } x \in \hat{S}. \tag{2.16}$$

From this and (2.15), it follows that for all $x \in \hat{S}$

$$\begin{aligned} &\hat{F}(x^*) + \sum_{i \in \tilde{J}_1} p_i(x_i^* - \hat{x}_i) + \sum_{i \in \tilde{J}_2} q_i(\hat{x}_i - x_i^*) \\ &\quad \le \hat{F}(x) + \sum_{i \in \tilde{J}_1} p_i(x_i - \hat{x}_i) + \sum_{i \in \tilde{J}_2} q_i(\hat{x}_i - x_i). \end{aligned}$$

Thus, Definition 2.1 and (2.11)–(2.14) imply for all $x \in \hat{S}$

$$\hat{F}(x^*) + \sum_{i \in \tilde{J}_1} p_i(0) + \sum_{i \in \tilde{J}_2} q_i(0) \le \hat{F}(x) + \sum_{i \in \tilde{J}_1} p_i(0) + \sum_{i \in \tilde{J}_2} q_i(0).$$

Thus,

$$\hat{F}(x^*) \le \hat{F}(x) \quad \text{for all } x \in \hat{S},$$

which implies that x^* is an optimal solution for SUB($\hat{d}, \hat{e}$). This completes the proof of the lemma. □

Assumption 2.2(i), (iii), and (iv), together with the linearity of the constraints, imply that the Karush-Kuhn-Tucker (KKT) conditions for (2.1) are both necessary and sufficient for optimality (see Mangasarian 1969). The KKT conditions for (2.1) are

$$\begin{aligned} -\nabla f(x) &= z + A'u + u_1 - u_2, \\ & \qquad u \ge 0,\ u_1 \ge 0,\ u_2 \ge 0, \\ -\nabla p(x^+) &= -z - v, \qquad v \ge 0, \\ -\nabla q(x^-) &= z - w, \qquad w \ge 0, \\ (x^+)'v &= 0, \qquad (x^-)'w = 0, \quad u'(Ax - b) = 0, \\ & \qquad u_1'(e - x) = 0, \quad u_2'(x - d) = 0, \\ x - x^+ + x^- &= \hat{x}, \qquad Ax \le b, \\ & \qquad d \le x \le e,\ x^+ \ge 0,\ x^- \ge 0, \end{aligned} \tag{2.17}$$

where u, u_1, u_2, z, v, and w are the vectors of multipliers for the constraints $Ax \le b$, $x \le e$, $x \ge d$, $x - x^+ + x^- = \hat{x}$, $x^+ \ge 0$, and $x^- \ge 0$, respectively.

The multipliers u, v, and w play a special role in our analysis. The next result shows that they always have a certain form.

LEMMA 2.4. *Let Assumptions* 2.1 *and* 2.2(i)–(iv) *be satisfied. If* $((x^*)', (x^{*+})', (x^{*-})')'$ *is a proper optimal solution for* (2.1), *then the multipliers for the constraints* $x - x^+ + x^- = \hat{x}$, $x^+ \ge 0$, *and* $x^- \ge 0$ *are* z, v, *and* w, *respectively, where*

$$z_i = \begin{cases} \dfrac{dp_i(x_i^* - \hat{x}_i)}{dx_i^+}, & i \in I^+(x^*), \\ -(\nabla f(x^*))_i - (A'u)_i, & i \in I(x^*), \\ -\dfrac{dq_i(\hat{x}_i - x_i^*)}{dx_i^-}, & i \in I^-(x^*), \end{cases}$$

$$v_i = \begin{cases} 0, & i \in I^+(x^*), \\ (\nabla f(x^*))_i + (A'u)_i + \dfrac{dp_i(0)}{dx_i^+}, & i \in I(x^*), \\ \dfrac{dp_i(0)}{dx_i^+} + \dfrac{dq_i(\hat{x}_i - x_i^*)}{dx_i^-}, & i \in I^-(x^*), \end{cases}$$

$$w_i = \begin{cases} \dfrac{dp_i(x_i^* - \hat{x}_i)}{dx_i^+} + \dfrac{dq_i(0)}{dx_i^-}, & i \in I^+(x^*), \\ -((\nabla f(x^*))_i + (A'u)_i) + \dfrac{dq_i(0)}{dx_i^-}, & i \in I(x^*), \\ 0, & i \in I^-(x^*), \end{cases} \tag{2.18}$$

with u *being the* m*-vector of multipliers for the constraints* $Ax \le b$.

PROOF. According to Assumption 2.2(i)–(iv), (2.1) is a convex programming problem and thus the KKT conditions for (2.1), namely (2.17), are sufficient for optimality. Thus, there exist u, u_1, u_2, z, v, and w, which are the vectors of multipliers for the constraints $Ax \le b$, $x \le e$, $x \ge d$, $x - x^+ + x^- = \hat{x}$, $x^+ \ge 0$, and $x^- \ge 0$, respectively, such that (2.17) is satisfied for $x = x^*$, $x^+ = x^{*+}$, and $x^- = x^{*-}$. According to Assumption 2.1, $(u_1)_i = (u_2)_i = 0$ for $i \in I(x^*)$. Corollary 2.1 implies that x^{*+} and x^{*-} are positive and negative portions of x^* with respect to $\hat{x}$, respectively. From this last, Assumption 2.2(ii), and (2.17), it follows that v, w, and z defined by (2.18) are the multipliers for the constraints $x^+ \ge 0$, $x^- \ge 0$, and $x - x^+ + x^- = \hat{x}$, respectively. This completes the proof of the lemma. □

The following key result relates an optimal solution for SUB($\tilde{d}, \tilde{e}$) with an optimal solution for (2.1).

THEOREM 2.1. (a) *Let Assumptions* 2.1 *and* 2.2 *be satisfied. Let* x^* *be an optimal solution for* SUB($\tilde{d},\tilde{e}$), *let* u *be the* m*-vector of multipliers for the constraints* $Ax\le b$, *and let* x^{*+} *and* x^{*-} *be the positive and negative portions of* x^* *with respect to* $\hat{x}$, *respectively. Then* $((x^*)',(x^{*+})',(x^{*-})')'$ *satisfies all the KKT conditions for* (2.1) *with the possible exception of nonnegativity of some multipliers for the constraints* $x_i^+\ge 0$ *or* $x_i^-\ge 0$ *for* $i\in I(x^*)$. *For all* $i\in I(x^*)$ *these multipliers are*

$$v_i\equiv(\nabla f(x^*))_i+(A'u)_i+\frac{dp_i(0)}{dx_i^+},\tag{2.19}$$

$$w_i\equiv-((\nabla f(x^*))_i+(A'u)_i)+\frac{dq_i(0)}{dx_i^-}.\tag{2.20}$$

If $v_i\ge 0$ *and* $w_i\ge 0$ *for all* $i\in I(x^*)$, *then* $((x^*)',(x^{*+})',(x^{*-})')'$ *is a proper optimal solution for* (2.1).

(b) *Let Assumption* 2.2(ii), (v) *be satisfied. Then if* x^* *is an optimal solution for* SUB($\tilde{d},\tilde{e}$), *then* $((x^*)',(x^{*+})',(x^{*-})')'$ *is a proper optimal solution for* (2.1) *where the constraints* $d\le x\le e$ *are replaced by the constraints* $\tilde{d}\le x\le\tilde{e}$ *of* SUB($\tilde{d},\tilde{e}$) *and where* x^{*+} *and* x^{*-} *are the positive and negative portions of* x^* *with respect to* $\hat{x}$, *respectively.*

PROOF. (a) Let x^* be an optimal solution of SUB($\tilde{d},\tilde{e}$). Then according to Lemma 2.3, x^* is also optimal for SUB($\hat{d},\hat{e}$), where $\hat{d}$ and $\hat{e}$ are given by (2.11) and (2.12), respectively. (2.2) and (2.3) imply that the KKT conditions for SUB($\hat{d},\hat{e}$) can be written as follows:

$$\begin{aligned}
-(\nabla f(x))_i-\frac{dp_i(x_i-\hat{x}_i)}{dx_i^+}&=(A'u)_i+(\tilde{u}_1)_i-(\tilde{u}_2)_i,\\
\hat{d}_i=\hat{x}_i,\quad \hat{e}_i=e_i,&\\
-(\nabla f(x))_i&=(A'u)_i+(\tilde{u}_1)_i-(\tilde{u}_2)_i,\\
\hat{d}_i=\hat{e}_i=\hat{x}_i,&\\
-(\nabla f(x))_i+\frac{dq_i(\hat{x}_i-x_i)}{dx_i^-}&=(A'u)_i+(\tilde{u}_1)_i-(\tilde{u}_2)_i,\\
\hat{d}_i=d_i,\quad \hat{e}_i=\hat{x}_i,&\\
Ax\le b,\quad \hat{d}\le x\le\hat{e},&\\
u\ge 0,\ \tilde{u}_1\ge 0,\ \tilde{u}_2\ge 0,&\\
u'(Ax-b)=0,\ \tilde{u}_1'(\hat{e}-x)=0,&\\
\tilde{u}_2'(x-\hat{d})=0,&
\end{aligned}\tag{2.21}$$

where u is the m-vector of multipliers for the constraints $Ax\le b$ and $\tilde{u}_1\ge 0$, and $\tilde{u}_2\ge 0$ are the n-vectors of multipliers for the constraints $x\le\hat{e}$ and $x\ge\hat{d}$, respectively.

From (2.2), (2.11), (2.12), and Definition 2.1 we obtain

$$\begin{aligned}
i\in I^-(x^*)\quad&\text{if and only if } \hat{d}_i=d_i,\hat{e}_i=\hat{x}_i,\\
i\in I^+(x^*)\quad&\text{if and only if } \hat{d}_i=\hat{x}_i,\hat{e}_i=e_i,\\
i\in I(x^*)\quad&\text{if and only if } \hat{d}_i=\hat{e}_i=\hat{x}_i.
\end{aligned}\tag{2.22}$$

For $i\in I(x^*)$ define

$$y_i\equiv(\tilde{u}_1)_i-(\tilde{u}_2)_i.\tag{2.23}$$

Definition 2.1, (2.22), the complementarity part of the KKT conditions (2.21), namely $\tilde{u}_1'(\hat{e}-x)=0$, and $\tilde{u}_2'(x-\hat{d})=0$ imply the following:

$$\begin{aligned}
(\tilde{u}_1)_i=0\quad&\text{for } i\in I^-(x^*),\quad\text{and}\\
(\tilde{u}_2)_i=0\quad&\text{for } i\in I^+(x^*).
\end{aligned}\tag{2.24}$$

Finally, let u_1 and u_2 be defined as follows:

$$(u_1)_i\equiv\begin{cases}(\tilde{u}_1)_i, & i\in I^+(x^*),\\ 0, & i\in I(x^*)\cup I^-(x^*),\end{cases}\tag{2.25}$$

$$(u_2)_i\equiv\begin{cases}(\tilde{u}_2)_i, & i\in I^-(x^*),\\ 0, & i\in I^+(x^*)\cup I(x^*).\end{cases}\tag{2.26}$$

Then it follows from (2.22)–(2.26) that the KKT conditions for SUB($\hat{d},\hat{e}$), namely (2.21), can be written as

$$\begin{aligned}
-(\nabla f(x))_i-\frac{dp_i(x_i-\hat{x}_i)}{dx_i^+}&=(A'u)_i+(u_1)_i,\ \ i\in I^+(x^*),\\
-(\nabla f(x))_i&=(A'u)_i+y_i,\ \ i\in I(x^*),\\
-(\nabla f(x))_i+\frac{dq_i(\hat{x}_i-x_i)}{dx_i^-}&=(A'u)_i-(u_2)_i,\ \ i\in I^-(x^*),\\
Ax\le b,\ \ \hat{d}\le x\le\hat{e},&\\
u\ge 0,\ \ u_1\ge 0,\ \ u_2\ge 0,&\\
u'(Ax-b)=0,\ \ u_1'(\hat{e}-x)=0,&\\
u_2'(x-\hat{d})=0.&
\end{aligned}\tag{2.27}$$

Because, according to Assumption 2.2(i)–(iv), SUB($\hat{d},\hat{e}$) is a convex programming problem with linear constraints, the KKT conditions (2.27) are both necessary and sufficient for optimality (see Mangasarian 1969). Thus, if x^* is an optimal solution for SUB($\hat{d},\hat{e}$), then there exist u, $\tilde{u}_1$, and $\tilde{u}_2$ and thus also y_i, $i\in I(x^*)$, u_1 and u_2, defined by (2.23), (2.25), and (2.26), such that the KKT conditions (2.27) are satisfied for $x=x^*$.

Let $x=x^*$, u, u_1, and u_2 be as defined in (2.27) and let $x^+=x^{*+}$ and $x^-=x^{*-}$ be the positive and negative portions of x^* with respect to $\hat{x}$, respectively. Additionally, let v, w, and z be defined by (2.18); that is, according to Corollary 2.1 and Lemma 2.4, v, w, and z are multipliers for the constraints $x^+\ge 0$, $x^-\ge 0$, and $x-x^++x^-=\hat{x}$ in (2.1). Assumption 2.2(v) and (2.18) imply that

$$v_i\ge 0\quad\text{and}\quad w_i\ge 0,\quad\text{for all } i\in I^+(x^*)\cup I^-(x^*).\tag{2.28}$$

Then from (2.25)–(2.28) it follows that all the KKT conditions for (2.1), namely (2.17), are satisfied, with the

possible exception of nonnegativity of some multipliers for the constraints $x_i^+ \geq 0$ or $x_i^- \geq 0$ for $i \in I(x^*)$. If in addition $v_i \geq 0$ and $w_i \geq 0$ for all $i \in I(x^*)$, then all the KKT conditions for (2.1) are satisfied. As (2.1) is a convex programming problem, the KKT conditions (2.17) are sufficient for optimality. This implies that $((x^*)', (x^{*+})', (x^{*-})')'$ is an optimal solution for (2.1). Furthermore, by construction $((x^*)', (x^{*+})', (x^{*-})')'$ is a proper optimal solution for (2.1). This completes the proof of part (a) of the theorem.

(b) Let D be the set of proper feasible solutions of (2.1). Thus according to Corollary 2.1, the set D is defined as

$$D \equiv \{(x', (x^+)', (x^-)')' \mid Ax \leq b,\ \tilde{d} \leq x \leq \tilde{e},\ x^+ \text{ and } x^- \text{ are positive and negative portions of } x \text{ with respect to } \hat{x}\}. \quad (2.29)$$

Additionally, let $\tilde{F}(x)$ be the objective function of SUB($\tilde{d}, \tilde{e}$). Then (2.3) implies that

$$\tilde{F}(x) = f(x) + \sum_{i \in J_1} p_i(x_i - \hat{x}_i) + \sum_{i \in J_2} q_i(\hat{x}_i - x_i), \quad (2.30)$$

where $J_1 = \{i \mid \tilde{d}_i = \hat{x}_i, \tilde{e}_i = e_i\}$ and $J_2 = \{i \mid \tilde{d}_i = d_i, \tilde{e}_i = \hat{x}_i\}$.

We refer to (P) as (2.1) with the constraints $d \leq x \leq e$ being replaced by the constraints $\tilde{d} \leq x \leq \tilde{e}$ of SUB($\tilde{d}, \tilde{e}$). Assumption 2.2(ii) and (v) and Lemma 2.1, which is also valid for (P), imply that for the proper optimality of (P) it is sufficient to show that there exists $(y', (y^+)', (y^-)')' \in D$ such that

$$f(y) + p(y^+) + q(y^-) \leq f(x) + p(x^+) + q(x^-) \quad \text{for all } (x', (x^+)', (x^-)')' \in D. \quad (2.31)$$

The definition of D, SUB($\tilde{d}, \tilde{e}$), (2.2), (2.3), and (2.30) imply that for all $(x', (x^+)', (x^-)')' \in D$

$$\begin{aligned} f(x) + p(x^+) + q(x^-) &= f(x) + \sum_{i \in I^+(x)} p_i(x_i - \hat{x}_i) + \sum_{i \in I(x) \cup I^-(x)} p_i(0) \\ &\quad + \sum_{i \in I^-(x)} q_i(\hat{x}_i - x_i) + \sum_{i \in I^+(x) \cup I(x)} q_i(0) \\ &= f(x) + \sum_{i \in J_1} p_i(x_i - \hat{x}_i) + \sum_{i \in J \cup J_2} p_i(0) \\ &\quad + \sum_{i \in J_2} q_i(\hat{x}_i - x_i) + \sum_{i \in J_1 \cup J} q_i(0) \\ &= \tilde{F}(x) + \sum_{i \in J \cup J_2} p_i(0) + \sum_{i \in J_1 \cup J} q_i(0), \end{aligned} \quad (2.32)$$

where $J = \{i \mid \tilde{d}_i = \tilde{e}_i = \hat{x}_i\}$. If x^* is the optimal solution for SUB($\tilde{d}, \tilde{e}$), then (2.32) and the fact that the feasible region of SUB($\tilde{d}, \tilde{e}$) coincides with the x portion of D imply the validity of (2.31) for $y = x^*$, $y^+ = x^{*+}$, and $y^- = x^{*-}$, where x^{*+} and x^{*-} are positive and negative portions of x^* with respect to $\hat{x}$, respectively. This completes the proof of part (b) of the theorem. □

3. A Solution Algorithm

In this section, we formulate a solution method for (2.1) solely in terms of n-dimensional quantities. This method treats the variables x^+ and x^-, the constraint $x - x^+ + x^- = \hat{x}$, as well as the constraints $x^+ \geq 0$, $x^- \geq 0$ *implicitly* rather than explicitly. At each iteration j, each x_i is restricted according to one of the possibilities: (i) $d_i \leq x_i \leq \hat{x}_i$, (ii) $x_i = \hat{x}_i$, or (iii) $\hat{x}_i \leq x_i \leq e_i$. The objective function for SUB($\tilde{d}, \tilde{e}$) is created from the objective function of (1.1) by the addition of transaction costs terms according to (i)–(iii) as follows: In the case of (i) this term is $q_i(\hat{x}_i - x_i)$, which is the transaction cost for selling the amount $\hat{x}_i - x_i$ of asset i. In the case of (ii) this term is zero. In the case of (iii) this term is $p_i(x_i - \hat{x}_i)$, which is the transaction cost for buying the amount $x_i - \hat{x}_i$ of asset i.

At the jth iteration, SUB($\tilde{d}, \tilde{e}$) is solved to produce optimal solution x^{j+1} and multipliers u^{j+1} for the constraints $Ax \leq b$, then for each $i \in I(x^{j+1})$ the multipliers v_i^{j+1} and w_i^{j+1} can be calculated from Theorem 2.1(a). If these are all nonnegative, then from Theorem 2.1(a), $((x^{j+1})', (x^+)', (x^-)')'$ is optimal for (2.1), where x^+ and x^- are the positive and negative portions of x^{j+1} with respect to $\hat{x}$, respectively.

Otherwise, suppose $v_{k_1}^{j+1}$ is the smallest of these multipliers. For the next iteration, the upper bound on x_{k_1} is changed to e_{k_1} and the lower bound on it is changed to $\hat{x}_{k_1}$. If $w_{k_2}^{j+1}$ is the smallest such multiplier, then for the next iteration, the lower bound on x_{k_2} is changed to d_{k_2} and the upper is changed to $\hat{x}_{k_2}$. From Lemma A.1 (see appendix), the objective function value for (2.1) for the next iteration will be strictly less than the present one.

Next we give a detailed formulation of the algorithm.

Algorithm
Model Problem: Problem (2.1) under
Assumptions 2.1, 2.2, and $S \neq \varnothing$, where S is given by (1.2).
Begin
Initialization:
Start with any $x^0 \in S$. Construct the initial bounds $\tilde{d}^0$, $\tilde{e}^0$ as follows:
do for $i = 1, \ldots, n$
if $x_i^0 > \hat{x}_i$ set $\tilde{d}_i^0 = \hat{x}_i$, $\tilde{e}_i^0 = e_i$
elseif $x_i^0 < \hat{x}_i$ set $\tilde{d}_i^0 = d_i$, $\tilde{e}_i^0 = \hat{x}_i$
else set $\tilde{d}_i^0 = \tilde{e}_i^0 = \hat{x}_i$
endif
enddo
Set $j = 0$ and go to Step 1.
Step 1: Solution of Subproblem
Solve SUB($\tilde{d}^j, \tilde{e}^j$) to obtain optimal solution x^{j+1} and the multiplier vector u^{j+1} for the constraints $Ax \leq b$. Go to Step 2.

Step 2: Update and Optimality Test

For $i \in I(x^{j+1})$ compute

$$v_i^{j+1} = (\nabla f(x^{j+1}) + A'u^{j+1})_i + \frac{dp_i(0)}{dx_i^+},$$

$$w_i^{j+1} = -(\nabla f(x^{j+1}) + A'u^{j+1})_i + \frac{dq_i(0)}{dx_i^-}.$$

Further, compute k_1 and k_2 such that

$$v_{k_1}^{j+1} = \min\{v_i^{j+1} \mid i \in I(x^{j+1})\},$$
$$w_{k_2}^{j+1} = \min\{w_i^{j+1} \mid i \in I(x^{j+1})\}.$$

if $v_{k_1}^{j+1} \geq 0$ and $w_{k_2}^{j+1} \geq 0$, then STOP with a proper optimal solution $((x^{j+1})', (x^{j+1,+})', (x^{j+1,-})')'$ for (2.1), where $x^{j+1,+}$ and $x^{j+1,-}$ are positive and negative portions of x^{j+1} with respect to $\hat{x}$, respectively.

elseif $v_{k_1}^{j+1} \leq w_{k_2}^{j+1}$ then set

$$\tilde{d}_i^{j+1} = \begin{cases} \tilde{d}_i^j, & i \in \{1, \ldots, n\} - I(x^{j+1}), \\ \hat{x}_i, & i \in I(x^{j+1}), \end{cases}$$

$$\tilde{e}_i^{j+1} = \begin{cases} \tilde{e}_i^j, & i \in \{1, \ldots, n\} - I(x^{j+1}), \\ \hat{x}_i, & i \in I(x^{j+1}) - \{k_1\}, \\ e_{k_1}, & i = k_1, \end{cases}$$

replace j with $j+1$ and go to Step 1.

else set

$$\tilde{d}_i^{j+1} = \begin{cases} \tilde{d}_i^j, & i \in \{1, \ldots, n\} - I(x^{j+1}), \\ \hat{x}_i, & i \in I(x^{j+1}) - \{k_2\}, \\ d_{k_2}, & i = k_2, \end{cases}$$

$$\tilde{e}_i^{j+1} = \begin{cases} \tilde{e}_i^j, & i \in \{1, \ldots, n\} - I(x^{j+1}), \\ \hat{x}_i, & i \in I(x^{j+1}), \end{cases}$$

replace j with $j+1$ and go to Step 1.

endif

End

Remark 3.1. (a) In light of Remark 2.1, the algorithm needs no provision for the possibility that (2.1) is unbounded from below.

(b) Consecutive subproblems differ in that one or more pairs of bounds have been replaced by others and corresponding changes have been made in the convex separable part of the objective function. Furthermore, the optimal solution for SUB($\tilde{d}^j, \tilde{e}^j$) is feasible for SUB($\tilde{d}^{j+1}, \tilde{e}^{j+1}$) and may be used as a starting point for it. Thus, if $S \neq \varnothing$, then the feasible region of any subproblem solved by the algorithm is nonempty.

The algorithm is demonstrated in the following two examples, which also show the values of the objective function for (2.1) at each iteration.

Example 3.1. $f(x) = -6x_1 + x_1^2 - 2x_2 + x_2^2$, $\hat{x} = (1, 1)'$, $p(x^+) = 10x_1^+ + x_2^+$, $q(x^-) = x_1^-$, A and b are null, $d = (0, 0)'$, $e = (2, 2)'$.

Initialization: We choose $x^0 = (2, 1)'$ as the starting point for the subproblem. Then $\tilde{d}^0 = (1, 1)'$, $\tilde{e}^0 = (2, 1)'$, and thus $\tilde{c}^0(x) = 10(x_1 - 1)$. In addition, $G(x^0, x^{0,+}, x^{0,-}) = 1$, where $x^{0,+} = (1, 0)'$, $x^{0,-} = (0, 0)'$.

Step 1: SUB($\tilde{d}^0, \tilde{e}^0$) is precisely the problem:

$$\min\{4x_1 + x_1^2 - 2x_2 + x_2^2 - 10 \mid 1 \leq x_1 \leq 2,\ x_2 = 1\},$$

which has optimal solution $x^1 = (1, 1)'$.

Step 2: $I(x^1) = \{1, 2\}$, $v^1 = (6, 1)'$, $w^1 = (5, 0)'$. Since $v^1 \geq 0$ and $w^1 \geq 0$, $((x^1)', (x^{1,+})', (x^{1,-})')'$ is optimal for (2.1) with the given data, where $x^{1,+} = (0, 0)'$ and $x^{1,-} = (0, 0)'$. Furthermore, $G(x^1, x^{1,+}, x^{1,-}) = -6$.

Example 3.2. We obtain a second example from Example 3.1 by leaving $f(x)$, $\hat{x}$, $q(x^-)$, x^0, A, b, d, and e unchanged but taking $p(x^+) = 3x_1^+ + x_2^+$.

Initialization: Then $\tilde{d}^0 = (1, 1)'$, $\tilde{e}^0 = (2, 2)'$ and thus $\tilde{c}^0(x) = 3(x_1 - 1)$. In addition, $G(x^0, x^{0,+}, x^{0,-}) = -6$, where $x^{0,+} = (1, 0)'$, $x^{0,-} = (0, 0)'$.

Step 1: SUB($\tilde{d}^0, \tilde{e}^0$) is precisely the problem:

$$\min\{-3x_1 + x_1^2 - 2x_2 + x_2^2 - 3 \mid 1 \leq x_1 \leq 2,\ x_2 = 1\},$$

which has optimal solution $x^1 = (\frac{3}{2}, 1)'$.

Step 2: $I(x^1) = \{2\}$, $v_2^1 = 1$, $w_2^1 = 0$. Since $v_2^1 \geq 0$ and $w_2^1 \geq 0$, $((x^1)', (x^{1,+})', (x^{1,-})')'$ is optimal for (2.1) with the given data, where $x^{1,+} = (\frac{1}{2}, 0)'$, $x^{1,-} = (0, 0)'$. Furthermore, $G(x^1, x^{1,+}, x^{1,-}) = -25/4$.

Definition 3.1. A point $X = (x', (x^+)', (x^-)')'$ is degenerate for (2.1) if it is feasible for (2.1) and the gradients of those constraints active at X are linearly dependent.

The finite termination property of the algorithm is established in the following theorem.

Theorem 3.1. *Let Assumptions* 2.1 *and* 2.2 *be satisfied and let* $S \neq \varnothing$, *where* S *is given by* (1.2). *Beginning with any* $x^0 \in S$, *let the algorithm be applied to* (2.1) *and let* $x^1, x^2, \ldots, x^j, \ldots$ *be the points so obtained. Let* $x^{j,+}$ *and* $x^{j,-}$ *be the positive and negative portions of* x^j *with respect to* $\hat{x}$, *respectively. Assume each* $((x^j)', (x^{j,+})', (x^{j,-})')'$ *is nondegenerate. Then* $G(x^{j+1}, x^{j+1,+}, x^{j+1,-}) < G(x^j, x^{j,+}, x^{j,-})$ *for* $j = 1, 2, \ldots$, *where* $G(x, x^+, x^-)$ *is the objective function of* (2.1). *In addition, the algorithm will solve* (2.1) *in a finite number of steps; that is, there is a* k, *such that* $((x^k)', (x^{k,+})', (x^{k,-})')'$ *is a proper optimal solution for* (2.1).

Proof. At the end of iteration $j-1$, Theorem 2.1 asserts that $((x^j)', (x^{j,+})', (x^{j,-})')'$ satisfies all of the KKT conditions for (2.1), with the possible exception of the nonnegativity of the dual variables for the constraint $(x^+)_i \geq 0$ or $(x^-)_i \geq 0$, $i \in I(x^j)$. Furthermore, from Theorem 2.1, these multipliers are v_i^j and w_i^j for all $i \in I(x^j)$ as determined in Step 2 of the algorithm.

Thus, if $v^j_{k_1} \geq 0$ and $w^j_{k_2} \geq 0$ where k_1, k_2 are defined as in Step 2 of the algorithm, then according to Theorem 2.1(a) all of the KKT conditions for (2.1) are satisfied and $((x^j)', (x^{j,+})', (x^{j,-})')'$ is a proper optimal solution for (2.1).

If $v^j_{k_1} \leq w^j_{k_2}$ and $v^j_{k_1} < 0$, then according to Step 2 of the algorithm the constraints for the jth subproblem are obtained from those of the subproblem $j-1$ by replacing the constraint $x_{k_1} = \hat{x}_{k_1}$ with $\hat{x}_{k_1} \leq x_{k_1} \leq e_{k_1}$ as well as imposing $x_i = \hat{x}_i$ for all $i \in I(x^j) - \{k_1\}$. From Lemma A.1 it follows that

$$G(x^{j+1}, x^{j+1,+}, x^{j+1,-}) < G(x^j, x^{j,+}, x^{j,-}). \tag{3.1}$$

The case for $w^j_{k_2} < v^j_{k_1}$ and $w^j_{k_2} < 0$ is similar and also leads to (3.1). Therefore, (3.1) is satisfied for all iterations.

In each subproblem, each variable x_i, $i = 1, \ldots, n$, is restricted according to $d_i \leq x_i \leq \hat{x}_i$ or $x_i = \hat{x}_i$ or $\hat{x}_i \leq x_i \leq e_i$. Thus, there is a finitely number of subproblems and according to (3.1) and Theorem 2.1(b), none can be repeated. Thus, the algorithm terminates in a finite number of steps and by Theorem 2.1, it will terminate with an optimal solution for (2.1).

Note that according to Assumption 2.2(i), (ii), (iv), Remark 3.1, and as $S \neq \varnothing$, SUB($\tilde{d}^j, \tilde{e}^j$) solved in Step 1 of the algorithm possesses a solution. This concludes the proof of the theorem. □

Remark 3.2. Theorem 3.1 asserts that the algorithm solves (2.1) in a finitely number of steps. If the utility function is quadratic, then the subproblems required by the algorithm are quadratic programming problems and can be solved in finitely many steps so that the entire solution procedure is finite. If the utility function is a more general nonlinear function, then solution procedures for the subproblems produce sequences of points that converge to an optimal solution.

Remark 3.3. A consequence of Theorem 3.1 is that under Assumptions 2.1 and 2.2 an optimal solution for (2.1) does indeed exist.

The computational efficiency of the algorithm is based on the following observations. Consecutive subproblems differ in that a single asset[4] has been allowed to move from its target and some assets that have reached their target are required to stay there and the $\tilde{c}(\tilde{d}, \tilde{e}, x)$ part of the objective function corresponding to these assets has been modified. The optimal solution for the previous subproblem, together with its associated data structure, can be used as starting data for the current subproblem (see also Remark 3.1(b)). Thus, one would expect consecutive subproblems to be solved very quickly.

Consider the following situation: (2.1) is to be solved for a sequence of time periods, where the data defining the expected utility function change only a small amount between consecutive time periods. Suppose x^* is the x portion of the optimal solution for (2.1) for the previous time period and for the next time period we take $\hat{x} = x^*$ in (2.1). It could happen that because of the transaction costs, the optimal solution of the revised problem (together with its positive and negative portions) is in fact optimal for (2.1) with the revised data.[5] In this case the algorithm would determine this in the first iteration and immediately terminate. If $((x^*)', (x^+)', (x^-)')'$, where x^+ and x^- are the positive and negative portions of x with respect to $\hat{x}$, respectively, is not optimal for the revised problem, one would expect it to be close to the optimal solution of (2.1). The optimal solution would be obtained by moving just a few assets off of their target values in just a few iterations of our algorithm, when using this point as a starting point in the next iteration. The fact that our algorithm deals only with n-dimensional quantities would give it considerable advantage over a general-purpose method for (2.1).

Assume that we use the same general-purpose algorithm to solve (2.1) as we use to solve SUB($\tilde{d}, \tilde{e}$). Our algorithm and a general-purpose algorithm for the solution for (2.1) differ in two ways. First, a general-purpose algorithm sees a problem with $3n$ variables and an additional $3n$ linear constraints, whereas our algorithm is dealing with n-dimensional quantities and equality constraints. The fact that some assets in the subproblem are fixed at their target values makes SUB($\tilde{d}, \tilde{e}$) easier to solve. Second, the general-purpose algorithm deals simultaneously with all the constraints of (2.1), including $x^+ \geq 0$ and $x^- \geq 0$, until all the multipliers for the inequality constraints are nonnegative. By contrast, our algorithm fixes some assets at their target values and optimizes over the remaining constraints using SUB($\tilde{d}, \tilde{e}$). Only after this is done are the multipliers for the active components $x^+ \geq 0$ and $x^- \geq 0$ examined and dealt with. Thus, the two methods differ only in the order in which the constraints are dealt with. Consequently, it is reasonable to make the assumption that the total number of iterations required to solve all the subproblems is comparable to the number of iterations to solve (2.1) directly. This implies that our algorithm will offer computational advantage because it deals only with n-dimensional quantities.

[4] Technically speaking, we should use "asset holdings" rather than "asset." However, for the sake of brevity, we will use the term asset to mean asset holdings.

[5] This would be the case if the multipliers v_i and w_i, $i = 1, \ldots, n$, from Theorem 2.1 were all nonnegative.

4. Computational Results

To test the new algorithm, we programmed it to solve the following problem with linear transaction costs:

$$\begin{aligned} \text{minimize:} \quad & -t(\mu' x - p' x^+ - q' x^-) + \tfrac{1}{2} x' C x \\ \text{subject to:} \quad & x - x^+ + x^- = \hat{x}, \quad x_1 + \cdots + x_n = 1, \\ & d \le x \le e, \\ & x^+ \ge 0, \quad x^- \ge 0. \end{aligned} \tag{4.1}$$

We compared the results with those obtained by solving the same problem with the barrier method implemented in ILOG Cplex, which is a commercial version of an interior point method (IPM).

We use random numbers to generate data for (4.1) so that the generated problems are similar to portfolio optimization problems. We construct an (n, n) matrix Q, whose elements are random numbers in the range $(-1, 1)$ and then form $C = Q'Q/1{,}000$ (which is positive semidefinite and positive definite, provided Q has full rank). C could then be interpreted as a covariance matrix.[6] The vector of expected returns, μ, is composed of random numbers in the range $(0, 1.3)$. The components of the target vector $\hat{x}$ were constructed from random numbers in the interval $(\epsilon, 1/n - \epsilon)$, where ϵ is a tolerance (typically $\epsilon = 10^{-10}$).

The risk-aversion parameter t was set to unity ($t = 1$). For the transaction costs, we use vectors p and q, all of whose components are set equal to a common value, P and Q, respectively. By solving several problems with differing values of $P = Q$, we were able to assess how the size of P affected the efficiency of our new method.

In the portfolio optimization context, the target portfolio may represent the optimal holdings for the previous time period, and large transaction costs may force the optimal holdings of many assets for the present period to the associated component of the target portfolio. In this case, the target portfolio is a good starting point for the new method presented here. However, in practical problems, $\hat{x}$ may not be feasible for the present problem. In this case, we can set $(x_0)_i = \hat{x}_i$ for most assets i and choose the remaining to satisfy feasibility. To emulate this situation, the initial feasible point x_0 for our new method is constructed according to $(x_0)_i = \hat{x}_i$, $i = 3, \ldots, n$, $(x_0)_2$ is a random number in the interval $(\epsilon, \hat{x}_2 - \epsilon)$, and $(x_0)_1 = 1 - ((x_0)_2 + \cdots + (x_0)_n)$. Note that the last $n - 2$ components of x_0 are identical with those of $\hat{x}$. For the lower and upper bounds, we use $d = 0$ and $e_i = (x_0)_1 + \epsilon$, $i = 1, \ldots, n$, respectively. By construction, $\hat{x}$ satisfies Assumption 2.1. The general linear constraints are taken to be only the budget constraint $x_1 + \cdots + x_n = 1$.

For the following number of assets $n = 500, 700, 1{,}000, 1{,}500, 2{,}000$, the random data are constructed and the identical problem is solved by both Cplex and our new method. For both methods, we solve the problem 10 times (using different random numbers), and the figures reported in Table 4.1 are average execution times.

Our method requires an algorithm to solve the quadratic programming (QP) subproblems. An appropriate method is an upper bounded variation of the van de Panne and Whinston/Dantzig symmetric form of the simplex method for QP (van de Panne and Whinston 1969, Dantzig 1963). The model problem for this method requires linear equality constraints as well as upper and lower bounds on all variables and is thus well suited for solving our subproblems. The method requires that the Hessian matrix of the objective function be positive semidefinite.

Our algorithm is implemented in Fortran 77 and both the barrier method implemented in Cplex 9.0 and our method were run on a SunFire V240 using SunOS 5.8. The execution times are summarized in Table 4.1, where the label "New" refers to the algorithm formulated in this paper and the label "Cplex" refers to using the barrier method in Cplex to solve the $3n$-dimensional problem (4.1) directly.

In Table 4.1, for each number of assets n, several values of P are specified, namely $P = 0.1, \ldots, 0.7$. For each such P, the execution times (in seconds) for Cplex and our new method are given. Also given are the ratio ("Cplex/New") of the Cplex to the new method execution times, the number of assets at the target ("On target"), and the average percentage of assets held on their targets at the optimal solution ("Assets at target (%)"). Note that the execution times presented in Table 4.1 have been rounded for clearer presentation and that the ratios "Cplex/New" have been calculated with more precise execution times. Table 4.1 shows that for $P = 0.1$ the ratio of Cplex to the new method execution time varies from 0.92 to 1.22. Cplex was faster than the new method for $n = 1{,}500$ and 2,000. For the case when $P > 0.1$ the execution time of the new method was always smaller than the execution time of Cplex. The ranges of Cplex to the new method execution time ratios are: 1.09–1.43, 1.37–1.78, 1.78–2.52, 2.75–4.18, 8.38–12.06, and 47.68–78.78 for $P = 0.2, \ldots, 0.7$, respectively, depending on the number of assets, n. Thus, it can be seen that with increasing transaction costs, the ratios of Cplex to the new method execution time increase as well; that is, the new method is becoming faster than Cplex when transaction costs are increased. This is because with bigger transaction costs, fewer assets move from their

[6] The size of the random numbers used to generate C is not relevant, as it can be accounted for in the risk aversion factor t.

Table 4.1 Execution Times for New Method vs. Cplex

$P = Q$	0.1	0.2	0.3	0.4	0.5	0.6	0.7
$n = 500$							
Cplex (time, secs)	11.21	11.53	11.61	11.85	11.55	10.91	9.63
New (time, secs)	11.14	9.05	7.41	5.40	3.35	1.01	0.19
Cplex/New	1.01	1.27	1.57	2.20	3.45	10.81	51.01
On target	78	163	228	305	381	465	494
$n = 700$							
Cplex (time, secs)	39.85	40.24	39.90	39.65	39.64	38.42	35.04
New (time, secs)	35.92	30.04	23.69	17.44	10.43	3.42	0.46
Cplex/New	1.11	1.34	1.68	2.27	3.80	11.22	75.73
On target	110	215	323	425	538	647	693
$n = 1{,}000$							
Cplex (time, secs)	131.71	130.15	129.79	131.42	134.09	126.85	116.02
New (time, secs)	108.39	91.04	72.87	52.08	32.10	10.52	1.47
Cplex/New	1.22	1.43	1.78	2.52	4.18	12.06	78.78
On target	157	305	453	18	766	924	989
$n = 1{,}500$							
Cplex (time, secs)	457.71	466.78	467.74	479.07	485.46	470.81	446.30
New (time, secs)	486.07	394.59	311.26	227.54	142.06	47.59	6.68
Cplex/New	0.94	1.18	1.50	2.11	3.42	9.89	66.83
On target	230	465	697	927	1,158	1,390	1,484
$n = 2{,}000$							
Cplex (time, secs)	1,086	1,131	1,143	1,121	1,083	1,000	1,032
New (time, secs)	1,185	1,039	835.27	628.59	394.17	119.41	21.65
Cplex/New	0.92	1.09	1.37	1.78	2.75	8.38	47.68
On target	303	616	927	1,224	1,532	1,850	1,978
Assets at target (%)	16	31	46	61	77	91	99

targets at the optimal solution. This can also be seen in the last line of Table 4.1, where only 16% of all assets stay at their target at the optimal solution for small transaction costs ($P = 0.1$), and nearly all assets remain at their target when transaction costs are large (in this case, when $P = 0.7$).

Our new method is increasingly effective with higher P and Q. However, we believe it will be very effective even when P and Q are quite small, provided the problem is modeled to more closely reflect a true portfolio scenario. In our computational results, we used a target vector that was mostly random and bore no relationship with the optimal solution of the problem. However, in real-life situations, the target vector would be the optimal solution for the previous time period and as such should be close to the optimal solution for the present time period, and small transaction costs would tend to hold most assets at their respective targets. In this situation, our method would find the optimal solution very quickly. This type of situation will be investigated in our further research.[7]

It is straightforward to show that if the sparsity of (4.1) is taken into account in the implemented IPM, then the complexity is maximum $O(n^3)$ per iteration of the IPM.[8] On the other hand, the complexity of the New method when a Whinston–van de Panne/Dantzig symmetric form of the simplex method for QP is used, is $O(n_o n^2)$, where n_o is the number of outer iterations.[9] It can be seen numerically that with increasing transaction costs the number of outer iterations (n_o) decreases. Thus, it can be seen that for larger transaction costs the New method will be superior.

The results, as they are now, are less favorable when compared to an interior point method implemented in Mosek. As it might be known, the interior point method implemented in Mosek is better using the sparsity structure and thus has shorter execution times than Cplex. In this case, for problems generated in our simulations, Mosek will perform better for small and also medium transaction costs. For higher transaction costs the new method will be superior. Thereby, even though the IPM implemented in Cplex outperforms the new method only for no transaction costs and very small transaction costs, an efficiently implemented IPM should do a better job. Thus, practitioners might consider this alternative as well.

[7] In addition, higher transaction costs can be justified in the sensitivity analysis of mean-variance portfolios.

[8] The complexity analysis of an IPM for Problem (4.1) can be obtained from the authors on request.

[9] The details of the complexity analysis for our specific problems can be obtained from the authors on request.

5. Conclusion

We have considered the problem of maximizing an expected utility function of n assets. Two possible utility functions are the mean-variance and power-utility functions. Transaction costs must be accounted for when changing an asset's holdings from a target value. Transaction costs can be modeled by introducing an additional $2n$ variables giving a $3n$-dimensional optimization problem. When n is large, this $3n$-dimensional problem may be computationally very expensive to solve. We have developed an algorithm for its solution in terms of a sequence of n-dimensional subproblems with corresponding savings in computer time and storage. Our method was developed by deriving optimality conditions for the higher dimensional problem solely in terms of n-dimensional quantities. The key idea was to treat the transaction costs implicitly rather than explicitly.

We compared our new method to the barrier method implemented in Cplex on a number of test problems. It solved the problems approximately 1.09 to 79 times as fast as the barrier method, depending on the magnitude of the transaction costs (0.2–0.7). However, even when the barrier method implemented in Cplex was more superior only for no transaction costs or low transaction costs and large number of assets, the efficiently implemented interior point method (i.e., in Mosek) would give a better performance with respect to the new method.

Acknowledgments

This research was supported by the National Science and Engineering Research Council of Canada and the Institute for Advanced Studies, Vienna, Austria. The authors acknowledge the assistance of Milan Vyskrabka, Zhengzheng Zhou, the associate editor, two anonymous referees, and the thoughtful comments and remarks of Margareta Halicka.

Appendix

Suppose we solve a convex minimization problem with "less than or equal to" constraints ("$\le$") and at least one equality constraint. Suppose the optimal solution for this, call it x^*, is nondegenerate and the multiplier for the equality constraint is strictly negative. Then we will show that the objective function can be strictly reduced by relaxing the equality to a "$\le$" constraint and restricting those "$\le$" constraints to the equality constraints that are active at x^*. This result is used to ensure the finite termination of the algorithm (see Theorem 3.1).

To this end, consider the problem

$$\min\{f(x) \mid a_i'x \le b_i,\ i=1,\dots,m-1,\ a_m'x = b_m\}, \qquad \text{(A.1)}$$

where $f(x)$ is any twice-differentiable convex function, $a_1,\dots,a_m$ are n-vectors, and $b_1,\dots,b_m$ are scalars. Suppose x^* is an optimal solution for (A.1). Then KKTs for (A.1) assert that there exists an m-vector $u=(u_1,\dots,u_m)'$ such that

$$\begin{aligned} &a_i'x^* \le b_i, \quad a_m'x^* = b_m, \quad i=1,\dots,m-1,\\ &-\nabla f(x^*) = u_1a_1+\cdots+u_ma_m, \quad u_i \ge 0, \quad i=1,\dots,m-1,\\ &u_i(a_i'x^*-b_i)=0, \quad i=1,\dots,m-1. \end{aligned} \qquad \text{(A.2)}$$

Without loss of generality assume that the first $k-1$ inequality constraints are active at x^* and denote

$$R \equiv \{x \mid a_i'x = b_i,\ i=1,\dots,k-1,\ a_i'x \le b_i,\ i=k,\dots,m\}, \qquad \text{(A.3)}$$

where $k-1<m$. The detailed result is formulated in the following lemma.

Lemma A.1. *Let R be defined by* (A.3), *x^* be a nondegenerate optimal solution for* (A.1), *u_m be the multiplier for constraint m, $u_m<0$, and $a_i'x^*=b_i$ for $i=1,\dots,k-1$. Then there exist points $\tilde{x}\in R$ for which $f(\tilde{x})<f(x^*)$.*

Proof. Because x^* is nondegenerate, $a_1,\dots,a_{k-1}$ and a_m are linearly independent. Let $d_{k+1},\dots,d_n$ be any vectors such that

$$D' = [a_1,\dots,a_{k-1},a_m,d_{k+1},\dots,d_n]$$

is nonsingular. Let

$$D^{-1} = [c_1,\dots,c_k,c_{k+1},\dots,c_n],$$

where c_i, denotes the ith column of D^{-1} for $i=1,\dots,n$. Let $s=c_k$. By definition of the inverse matrix,

$$a_i's = 0, \quad i=1,\dots,k-1 \qquad \text{(A.4)}$$

and

$$a_m's = 1. \qquad \text{(A.5)}$$

Consider points of the form $x^*-\sigma s$, where σ is a nonnegative scalar. From (A.4)

$$a_i'(x^*-\sigma s) = a_i'x^* = b_i, \quad i=1,\dots,k-1,$$

so the first $k-1$ constraints remain active at $x^*-\sigma s$ for all $\sigma\ge 0$. Furthermore, from (A.5),

$$a_m'(x^*-\sigma s) = b_m-\sigma < b_m, \quad \text{for } \sigma>0.$$

Thus, constraint m becomes inactive at $x^*-\sigma s$ for all strictly positive σ. Since constraints $k,\dots,m-1$ are inactive at x^*, this implies that

$$x^*-\sigma s \in R \quad \text{for all positive sufficiently small } \sigma. \qquad \text{(A.6)}$$

From (A.2) and the fact that constraints $k,\dots,m-1$ are inactive at x^* we have

$$-\nabla f(x^*) = u_1a_1+\cdots+u_{k-1}a_{k-1}+u_ma_m.$$

From this, (A.4), (A.5) and the hypothesis that $u_m<0$, it follows that

$$\nabla f(x^*)'s = -u_m > 0. \qquad \text{(A.7)}$$

From Taylor's series

$$f(x^*-\sigma s) = f(x^*)-\sigma\nabla f(x^*)'s+\tfrac{1}{2}\sigma^2 s'H(\xi)s, \qquad \text{(A.8)}$$

where H denotes the Hessian of $f(x)$ and ξ is on the

line segment joining x^* and $x^* - \sigma s$. It now follows from (A.6)–(A.8) that $f(x^* - \sigma s) < f(x^*)$ for all positive sufficiently small σ. This completes the proof of the lemma. □

References

Best, M. J., J. Hlouskova. 2003. Portfolio selection and transaction costs. *Comput. Optim. Appl.* **24**(1) 95–116.

Best, M. J., J. K. Kale. 2000. Quadratic programming for large-scale portfolio optimization. J. Keyes, ed. *Financial Services Information Systems*. CRC Press, Boca Raton, 513–529.

Best, M. J., K. Ritter. 1976. A class of accelerated conjugate direction methods for linearly constrained minimization problems. *Math. Comput.* **30**(135) 478–504.

Bodie, Z., A. Kane, A. J. Marcus. 1999. *Investments*, 4th ed. Irwin McGraw-Hill, Boston, MA.

Dantzig, G. B. 1963. *Linear Programming and Extensions*. Princeton University Press, Princeton, NJ.

Goldfarb, D., S. Liu. 1991. An $O(n^3L)$ primal interior point algorithm for convex quadratic programming. *Math. Programming* **49** 325–340.

Grauer, R. R., N. H. Hakansson. 1993. On the use of mean-variance and quadratic approximations in implementing dynamic investment strategies: A comparison of returns and investment policies. *Management Sci.* **39** 856–871.

Luenberger, D. G. 1998. *Investment Science*. Oxford University Press, New York.

Mangasarian, O. L. 1969. *Nonlinear Programming*. McGraw–Hill, New York.

Markowitz, H. M. 1959. *Portfolio Selection: Efficient Diversification of Investment*. John Wiley, New York.

Monteiro, R. D. C., I. Adler. 1989. Interior path following primal-dual algorithms. Part II: Convex quadratic programming. *Math. Programming* **44** 43–66.

Monteiro, R. D. C., I. Adler, M. G. C. Resende. 1990. A polynomial-time primal-dual affine scaling algorithm for linear and convex quadratic programming and its power series extension. *Math. Oper. Res.* **15**(2) 191–214.

Murty, K. G. 1995. *Operations Research: Deterministic Optimization Models*. Prentice Hall, Englewood Cliffs, NJ.

van de Panne, C., A. Whinston. 1969. The symmetric formulation of the simplex method for quadratic programming. *Econometrica* **37** 507–527.

Schattman, J. B. 2000. Portfolio selection under non-convex transaction costs and capital gains taxes. Unpublished doctoral dissertation, Rutgers Center for Operations Research, Rutgers University, Piscataway, NJ.