A semidefinite programming heuristic for quadratic programming problems with complementarity constraints

Stephen Braun[1]
Warren & Selbert, Inc.
Santa Barbara, CA 93101
brauns2@alum.rpi.edu

John E. Mitchell[2]
Department of Mathematical Sciences,
Rensselaer Polytechnic Institute,
Troy, NY 12180.
mitchj@rpi.edu
http://www.rpi.edu/~mitchj

November 30, 2002
Revised: February 7, 2004

Abstract

The presence of complementarity constraints brings a combinatorial flavour to an optimization problem. A quadratic programming problem with complementarity constraints can be relaxed to give a semidefinite programming problem. The solution to this relaxation can be used to generate feasible solutions to the complementarity constraints. A quadratic programming problem is solved for each of these feasible solutions and the best resulting solution provides an estimate for the optimal solution to the quadratic program with complementarity constraints. Computational testing of such an approach is described for a problem arising in portfolio optimization.

Keywords: Complementarity constraints, quadratic programming, semidefinite programming.

[1]Research supported in part by the National Science Foundation's VIGRE Program (Grant DMS-9983646).

[2]Research partially supported by NSF Grant number CCR–9901822.

1 Introduction

Mathematical programming problems with complementarity constraints arise in many settings, including engineering and economics. Complementarity relations arise naturally from requirements for equilibrium. For surveys of applications see Ferris and Pang [6] and Luo et al. [24]. The example problem we will consider later is drawn from portfolio optimization. This problem can be stated as a quadratic programming problem with complementarity constraints. It has the form

$$\begin{array}{lrcl} \min & f(x,w,v) & & \\ \text{subject to} & g(x,w,v) & = & 0 \\ & w^T v & = & 0 \\ & x,w,v & \geq & 0 \end{array}$$

where x is a p-vector, w and v are n-vectors, $f(x,w,v)$ is a quadratic function, and $g(x,w,v)$ is a linear function. Approaches to solving mathematical programs with complementarity constraints include SQP methods (Fletcher et al. [7, 8], Fukushima et al. [9]), interior point methods (Luo et al. [24], see also Leyffer [22]), regularization schemes (Scholtes [32]), active set methods (Fukushima and Tseng [10] and Scholtes [31]), and nonsmooth optimization techniques (Outrata and Zowe [29]). See also Ferris et al. [5] for preprocessing and tests of nonlinear programming approaches. The difficulty with these problems comes from the complementarity constraints, which impose a combinatorial structure on the problem (Scholtes [33]). Scheel and Scholtes [30] discuss optimality conditions for mathematical programs with complementarity constraints.

One impractical method to solve these problems is explicit enumeration. The complementarity condition $w^T v = 0$ requires that at least one of w_i and v_i be zero, for each component i. Examining each possible choice for which n nonnegativity constraints should be active gives 2^n quadratic programming problems, each of which can be solved efficiently.

In this paper, we investigate a method for choosing a good subset of these 2^n possible subproblems. The method first solves a semidefinite programming relaxation of the original problem, and then uses the solution to the relaxation to try to determine a candidate set of variables that can be fixed equal to zero. If enough variables are fixed to zero, the number of possible quadratic programming problems is tractable and a solution can be found. This approach is heuristic, in that there is no guarantee that the quadratic programming subproblem corresponding to the optimal complementarity alignment will be selected. Nonetheless, the semidefinite programming relaxation provides a lower bound on the optimal value to the problem and the value of the best quadratic subproblem provides an upper bound.

Semidefinite programming problems are convex optimization problems. They consist of a linear objective function and linear constraints together with a constraint that the variables correspond to a positive semidefinite matrix. For example, the variables themselves might be the entries of a symmetric positive semidefinite matrix, and the objective function and other constraints are linear functions of the entries of this matrix. For more on semidefinite programming, see the website maintained by Helmberg [12], the survey papers by Todd [35] and Vandenberghe and Boyd [38], and the SDP handbook edited by Wolkowicz et al. [39].

In §2, we describe a semidefinite programming relaxation of a quadratic program with complementarity constraints. The algorithm is given in §3. Degenerate cases are considered in §5. Our example problem from portfolio optimization is the subject of §4, with computational results contained in §6.

2 A semidefinite programming relaxation

In order to simplify the notation we write our standard form quadratic program with complementarity constraints as

$$\begin{array}{lrcl} \min & \frac{1}{2}x^TQx + c^Tx & & \\ \text{s.t.} & Ax & = & b \\ & x & \in & C \\ & 0 \leq x & \leq & u \end{array} \qquad (QPCC)$$

where c, u, and x are n-vectors, A is an $m \times n$ matrix, Q is a symmetric positive semidefinite $n \times n$ matrix, b is an m-vector, and C is the set of n-vectors satisfying a complementarity relationship of the form $x_i x_j = 0$ for certain pairs of indices.

The standard technique for obtaining a semidefinite programming problem from a quadratic programming problem is to exploit the fact that the trace of a matrix product DE is equal to the trace of ED, provided both products are defined. It follows that $x^TQx = \text{trace}(Qxx^T)$. Writing $X = xx^T$, we have $\text{trace}(Qxx^T) = \text{trace}(QX)$. Since Q and X are symmetric, this is equal to the Frobenius inner product $Q \bullet X$, where

$$D \bullet E := \sum_{i=1}^{n} \sum_{j=1}^{n} D_{ij} E_{ij}.$$

The introduction of a homogenizing variable t lets us write the objective function to $(QPCC)$ equivalently as

$$\frac{1}{2}x^TQx + c^Tx = \frac{1}{2} \begin{bmatrix} Q & c \\ c^T & 0 \end{bmatrix} \bullet \left(\begin{bmatrix} x \\ t \end{bmatrix} \begin{bmatrix} x^T & t \end{bmatrix} \right) = \frac{1}{2} \bar{Q} \bullet Z$$

provided $t = 1$, and where

$$\bar{Q} := \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix}$$

and

$$Z := \begin{bmatrix} X & tx \\ tx^T & t^2 \end{bmatrix} \qquad (1)$$

with

$$X := xx^T. \qquad (2)$$

The lifting procedures of Balas et al. [1] and Lovász and Schrijver [23] and the reformulation-linearization technique of Sherali and Adams [34] all involve multiplying a linear constraint by a variable. Let $\hat{a}_i$ denote the ith row of A, written as a column vector. Multiplying the constraint $\hat{a}_i^T x = b_i$ by x_j gives a constraint that can be written as

$$(\hat{a}_i e_j^T + e_j \hat{a}_i^T) \bullet X - b_i x_j = 0 \tag{3}$$

where e_j denotes the jth unit vector of appropriate dimension. This constraint is valid for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. These constraints are implied by $Ax = b$ so they do not immediately provide additional information in the original variables. They will prove useful when we change variables later.

The constraints $Ax = b$ can be written equivalently as $Ax - bt = 0$ if $t = 1$. Defining the column vector $a_i := [\hat{a}_i^T, \ -b_i]^T$, we can express (3) as a constraint on Z:

$$(a_i e_j^T + e_j a_i^T) \bullet Z = 0 \tag{4}$$

This is valid for $i = 1, \ldots, m$ and $j = 1, \ldots, n+1$, so e_j now has dimension $n+1$. Setting $j = n+1$ gives back the constraints $Ax - bt = 0$.

The complementarity constraint $x_i x_j = 0$ can be expressed as $Z_{ij} + Z_{ji} = 0$, or equivalently as $(e_i e_j^T + e_j e_i^T) \bullet Z = 0$. This is similar to constraints arising in SDP formulations of the independent set problem [23].

The nonnegativity constraints on x can be written $(e_{n+1} e_j^T + e_j e_{n+1}^T) \bullet Z \geq 0$ for $i = 1, \ldots, n$. We also impose a nonnegativity requirement on the other off-diagonal entries in Z, with constraints of the form $(e_i e_j^T + e_j e_i^T) \bullet Z \geq 0$. The upper bound constraint $x_i \leq u_i$ is equivalent to $Z_{ii} \leq u_i^2$, which can be written $(e_i e_i^T) \bullet Z \leq u_i^2$. Requiring $t = 1$ can be expressed as $(e_{n+1} e_{n+1}^T) \bullet Z = 1$.

Putting all this together, we obtain an equivalent optimization problem to $(QPCC)$, expressed in terms of the variable Z:

$$\begin{array}{llcll}
\min & \frac{1}{2}\bar{Q} \bullet Z & & & \\
\text{s.t.} & (a_i e_j^T + e_j a_i^T) \bullet Z & = & 0 & i = 1, \ldots, m, \ j = 1, \ldots, n+1 \\
& (e_i e_j^T + e_j e_i^T) \bullet Z & = & 0 & (i,j) \in I \\
& (e_i e_i^T) \bullet Z & \leq & u_i^2 & i = 1, \ldots, n \\
& (e_i e_j^T + e_j e_i^T) \bullet Z & \geq & 0 & i = 1, \ldots, n, \ j = i+1, \ldots, n+1 \\
& (e_{n+1} e_{n+1})^T \bullet Z & = & 1 & \\
& Z & & & \text{is symmetric and has rank 1}
\end{array}$$

Here, I denotes the set of complementarity relationships corresponding to $x \in C$.

Relaxing the constraint that Z have rank one to the requirement that Z be positive semidefinite gives the semidefinite programming problem

$$\begin{array}{llcll}
\min & \frac{1}{2}\bar{Q} \bullet Z & & & \\
\text{s.t.} & (a_i e_j^T + e_j a_i^T) \bullet Z & = & 0 & i = 1, \ldots, m, \ j = 1, \ldots, n+1 \\
& (e_i e_j^T + e_j e_i^T) \bullet Z & = & 0 & (i,j) \in I \\
& (e_i e_i^T) \bullet Z & \leq & u_i^2 & i = 1, \ldots, n \\
& (e_i e_j^T + e_j e_i^T) \bullet Z & \geq & 0 & i = 1, \ldots, n, \ j = i+1, \ldots, n+1 \\
& (e_{n+1} e_{n+1})^T \bullet Z & = & 1 & \\
& Z & \succeq & 0 &
\end{array} \tag{SDP}$$

where $Z \succeq 0$ is the notation used to indicate that the matrix Z is required to be symmetric and positive semidefinite.

Note that the constraints $(e_i e_i^T) \bullet Z \leq u_i^2$, $(e_{n+1} e_{n+1}^T) \bullet Z = 1$, and $Z \succeq 0$ together imply that $Z_{ij} \leq u_i u_j$, so there is no need to impose upper bound constraints on the off-diagonal entries.

Problem (SDP) can be solved in polynomial time using an interior point method and we used SDPT3 [36, 37] for some of our computational testing. For larger scale problems, we used the spectral bundle method of Helmberg and Rendl [18, 14, 15]. Cutting plane and bundle methods for approximating the nonsmooth constraint $Z \succeq 0$ are discussed in Krishnan and Mitchell [21].

More general linear complementarity relationships can be captured in the (SDP) formulation. For example, if we have a constraint $(d^T x + \alpha)(g^T x + \beta) = 0$ then we can formulate the constraint

$$\begin{bmatrix} dg^T + gd^T & \alpha g + \beta d \\ \alpha g^T + \beta d^T & 2\alpha\beta \end{bmatrix} \bullet Z = 0.$$

If desired, a cutting plane approach can be employed to solve (SDP), so that only a subset of the constraints $(a_i e_j^T + e_j a_i^T) \bullet Z = 0$ are used initially and additional constraints are added as needed. The maxcut problem and other combinatorial optimization problems have been solved with this approach; for more details, see Helmberg and Rendl [17, 15, 16], Krishnan [20], and Mitchell [27].

3 An SDP-based algorithm

We solve (SDP) as the first stage of an enumerative algorithm. The solution matrix for (SDP) provides an estimate $\tilde{x}$ for the optimal solution vector x^*, and that estimate for x^* is used to deduce information about membership in the optimal active set, as discussed later in this section. Such information allows us to identify the most promising QP-subproblems which are then solved and the best observed solution is identified. Since $(QPCC)$ is an NP-hard problem, it cannot be guaranteed that the deduced information is accurate.

As a secondary and corrective step, the best identified solution vector is recursively examined for degeneracy. (We delay discussion of degeneracy until §5, so as to anchor it in terms of the portfolio optimization problem described in §4.) If degeneracy is present then additional QP-subproblems are also solved. The final result of this process is returned as the best found solution. Figure 1 presents this strategy using a flowchart.

One metaphor for complete enumeration of all choices for the active set of nonnegativity constraints is that of a tree. Speaking in those terms, the idea behind our strategy is that solving (SDP) gives us a way to immediately go deep into the tree. We hope to be able to correctly eliminate a large number of leaves from consideration. With the recursive degeneracy check, additional leaves can be considered as justified. Figure 2 illustrates our strategy using this tree metaphor.

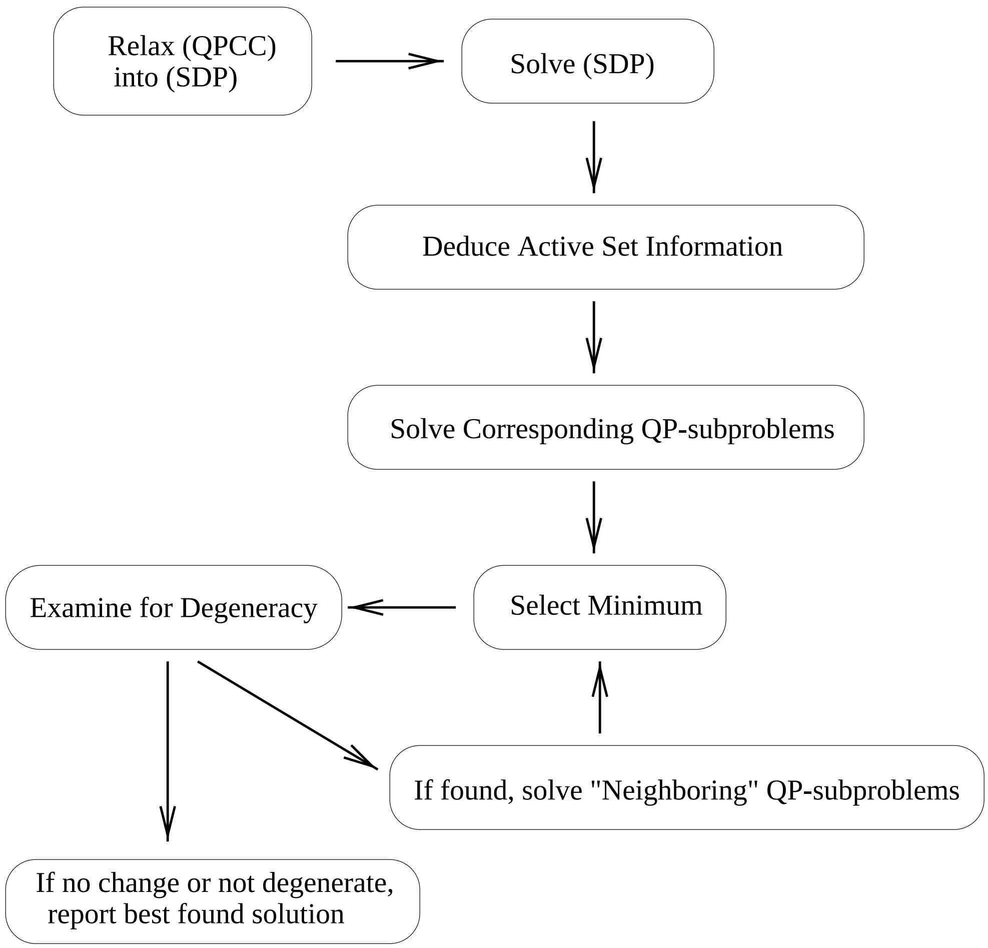

Figure 1: Flowchart illustrating the solution strategy used.

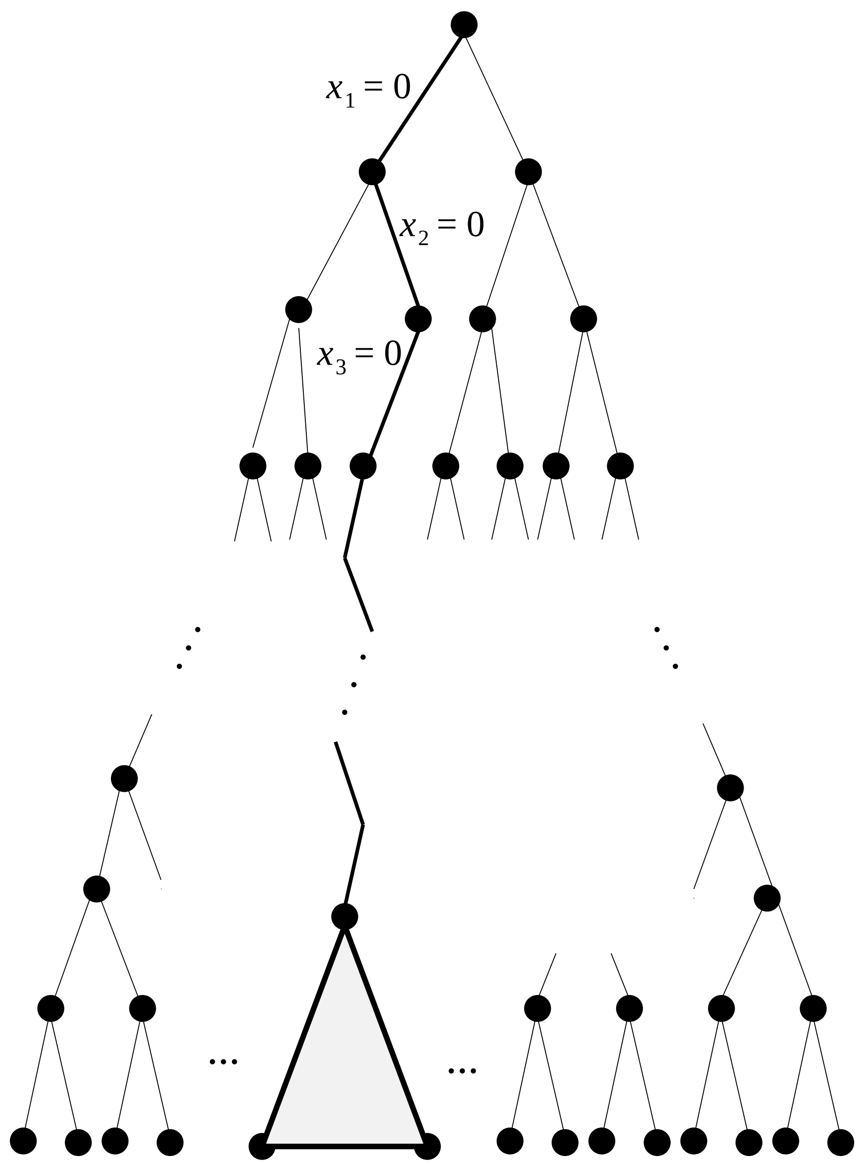

Figure 2: The figure uses the tree metaphor for our solution strategy. By solving (SDP) and attempting to determine members of the optimal active set, only the small shaded set of QP-subproblems need to be initially considered. Here the bold lines illustrate where activity decisions have been made within the tree.

Let Z^* denote a solution matrix returned for (SDP). This solution matrix may be an intermediate iterate or could be the solution returned as optimal from the SDP solver. The spectral decomposition can be applied. Analytically:

$$Z^* = \sum_{i=1}^{n+1} \lambda_i \xi_i {\xi_i}^T$$

where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_i \geq \ldots \geq \lambda_{n+1}$ are the eigenvalues of Z^* listed in order of decreasing magnitude. The eigenvector associated with each λ_i is denoted by ξ_i.

Let $Z^1 = \lambda_1 \xi_1 {\xi_1}^T$. This is certainly a rank-one matrix; in fact it has the attractive property of being the "closest" rank-one matrix to Z^* w.r.t. $\|\cdot\|_2$ (see Demmel [4] or Golub and Van Loan [11], for example). Given $Z_{ij} \geq 0 \ \forall i, j$, it follows from the theorem of Perron-Frobenius (see, eg, [19]) that every component of ξ_1 has the same sign. Without loss of generality, we assume $\xi_1 \geq 0$. As an aside, the second eigenvalue, λ_2, measures how close the full matrix is to being rank-one, analytically: $\|Z^* - Z^1\|_2 = \lambda_2$. Computationally, λ_2 was found to be a solver-independent measure of progress towards a rank-one solution matrix.

Our estimate $\tilde{x}$ comes directly from this rank-one matrix Z^1. The following analytic construction identifies the relationship that exists between $\tilde{x}$ and Z^1:

$$\begin{aligned} Z^1 &= \lambda_1 \xi_1 {\xi_1}^T \\ &= \left(\sqrt{\lambda_1}\, \xi_1\right) \left(\sqrt{\lambda_1}\, {\xi_1}^T\right) \\ &=: \begin{bmatrix} \tilde{x} \\ \tilde{t} \end{bmatrix} \begin{bmatrix} \tilde{x}^T & \tilde{t} \end{bmatrix} \end{aligned}$$

We compute $\tilde{x}$ after using the spectral decomposition to find the closest rank-one matrix, Z^1, to a solution of (SDP). Once $\tilde{x}$ has been determined, we compare $\tilde{x}_i$ and $\tilde{x}_j$ for $(i, j) \in I$ in order to determine whether to set $x_i = 0$ or $x_j = 0$. The complete procedure is given in Figure 3.

A heuristic which is perhaps better suited for larger problems is to rank the decisions and always generate QP-subproblems corresponding to the weakest N decisions where N is specified by the user. This single parameter N takes the place of the two decision tolerances, L_{tol} and U_{tol}. This idea has the computational advantage that the number of QP-subproblems is fixed. Such control over subproblem generation could perhaps be useful for balancing the computational load if subproblems are solved in parallel.

Since we are using (SDP) as the first stage in an enumerative algorithm, it is not necessarily the most productive investment of computation to solve (SDP) all the way to optimality. Stopping at some intermediate stage can still provide enough information to make valid activity decisions. The computational results presented in §6 exemplify the empirical accuracy of this remark.

Estimating $\tilde{x}$ via the spectral decomposition is numerically stable and makes fullest use of all information contained in the (SDP) solution matrix. Other estimates are possible, for example $\tilde{x} = \sqrt{diag\,(X^*)}$ where X^* is the first block of Z^*. However,

Step 1: Solve (SDP) until a user-specified stopping criterion is satisfied. Several examples of valid stopping criteria are: a fixed number of iterations, a time limit, a threshold on the second eigenvalue λ_2, or a relative precision optimality criterion.

Step 2: Take the solution returned from (SDP) and apply the spectral decomposition to find the closest rank-one matrix, $Z^1 = \lambda_1 \xi_1 {\xi_1}^T$.

Step 3: Estimate $\tilde{x} = \sqrt{\lambda_1}\ \xi_1(1\!:\!n)$. Here the notation $\xi_1(1\!:\!n)$ refers to the first n elements of ξ_1.

Step 4: Use $\tilde{x}$ to make activity decisions. Introduce tolerances U_{tol} and L_{tol}.

$$\text{If } \left\{ \tfrac{\tilde{x}_i}{u_i} \geq \mathrm{U}_{tol} \text{ and } \tfrac{\tilde{x}_j}{u_j} \leq \mathrm{L}_{tol} \right\} \text{ then } x_j = 0 \text{ should be considered active.}$$

$$\text{If } \left\{ \tfrac{\tilde{x}_i}{u_i} \leq \mathrm{L}_{tol} \text{ and } \tfrac{\tilde{x}_j}{u_j} \geq \mathrm{U}_{tol} \right\} \text{ then } x_i = 0 \text{ should be considered active.}$$

If neither criterion is satisfied then neither constraint is forced to be active.

Figure 3: Choosing an active set of constraints

these are only useful if (SDP) is solved to optimality. As mentioned previously, this is not necessarily the most productive investment of computation.

It may be possible to use sensitivity analysis to justify some of the activity decisions. Given a decision to make $x_i = 0$ active, it would be useful to determine lower bounds on the values of quadratic subproblems where this constraint is not forced to be active. If $(i, j) \in I$, such a bound can be determined by estimating the cost of forcing $x_j = 0$ to be active. The optimal value of (SDP) with the additional constraint $Z_{jj} = 0$ provides such a bound. Of course, solving this modified semidefinite programming is not attractive computationally if a number of variables are being considered, so Helmberg [13] describes a dual search procedure for underestimating the optimal value. The performance of variants of Helmberg's procedure in the setting of $(QPCC)$ is worthy of further investigation.

Our algorithm requires the solution of several related quadratic programming problems. Solution information to one problem, including the optimal solution and Lagrange multipliers, can sometimes be used to give a good warm start for a similar problem.

4 A portfolio optimization problem

Constructing a portfolio of investments is one of the most significant financial decisions facing individuals and institutions. A decision-making process must be developed which identifies the appropriate weight each investment should have within the portfolio. The portfolio must strike what the investor believes to be an acceptable balance between risk and reward. In addition, the costs incurred when setting up a new portfolio or rebalancing an existing portfolio must be included in any realistic analysis.

Essentially the standard portfolio optimization problem is to identify the optimal allocation of limited resources among a limited set of investments. Optimality is measured using a tradeoff between perceived risk and expected return. Expected future returns are based on historical data. Risk is measured by the variance of those historical returns.

Markowitz [25] first presented a quadratic programming model for choosing a portfolio, minimizing a quadratic risk measurement with a set of linear constraints specifying the minimum expected portfolio return, E_0, and enforcing full investment of funds. The decision variables x_i are the proportional weights of the i^{th} security in the portfolio. Here n securities are under consideration. Additionally, μ is the column vector of expected returns and V is the positive semidefinite covariance matrix. This formulation is:

$$\begin{array}{ll} \min_x & \frac{1}{2}x^T V x \\ \text{s.t.} & \mu^T x \geq E_0 \\ & \sum_{i=1}^n x_i = 1 \\ & x_i \geq 0 \quad \forall i. \end{array}$$

By varying the parameter E_0 and solving multiple instances of this problem, the set of efficient portfolios can be generated. This set, visualized in a risk/return plot, is called the efficient frontier. The return E_0 can be plotted on the horizontal axis and the risk $\frac{1}{2}x^T V x$ can be plotted on the vertical axis. An investor may decide where along the efficient frontier (s)he finds an acceptable balance between risk and reward.

Transactions are made to change an already existing portfolio, $\bar{x}$, into a new and efficient portfolio, x. A portfolio may need to be rebalanced periodically simply as updated risk and return information is generated with the passage of time. Further, any alteration to the set of investment choices would necessitate a rebalancing decision of this type.

We assume proportional transaction costs are paid each time a security is bought or sold. In addition to the obvious cost of brokerage fees/commissions, here are two examples of other transaction costs that can be modeled in this way:

1. Capital gains taxes are a security-specific selling cost that can be a major consideration for the rebalancing of a portfolio.

2. Another possibility would be to incorporate an investor's confidence in the risk/return forecast as a subjective "cost". Placing high buying and selling costs on a security would favor maintaining the current allocation $\bar{x}$. Placing a high selling cost and low buying cost could be used to express optimism that a security may outperform its forecast.

Let w and v denote the quantities of the securities that are bought and sold, respectively. In an efficient portfolio, a stock will not be both bought and sold, so we impose the complementarity constraint $w^T v = 0$. Let c_B and c_S denote the vectors of transaction costs incurred for buying and selling, respectively. We assume $c_B + c_S > 0$, $0 \leq c_B \leq e$ and $0 \leq c_S \leq e$, where e denotes the vector of ones.[3] The total amount paid in transaction costs is $c_B^T w + c_S^T v$ and the amount invested in each security is $x = \bar{x} + w - v$. The total amount invested after paying the transaction costs is $e^T x = e^T \bar{x} - c_B^T w - c_S^T v = 1 - c_B^T w - c_S^T v$. This can be expressed equivalently as a constraint on w and v, namely $(c_B + e)^T w + (c_S - e)^T v = 0$. We require that the expected return for x should be at least E_0. The calculation of the risk needs to be normalized by the square of the amount invested, giving a modified risk measurement:

$$\text{scaled risk } = \frac{1}{2} \frac{x^T V x}{(1 - c_B^T w - c_S^T v)^2}.$$

By using a change of variables $\hat{x} = sx$, $\hat{w} = sw$, and $\hat{v} = sv$, with $s = 1/(1 - c_B^T w - c_S^T v)$, we get a quadratic program with complementarity constraints:

$$\begin{array}{llcccccccl}
\min_{\hat{x},\hat{w},\hat{v},s} & \frac{1}{2}\hat{x}^T Q \hat{x} & & & & & & & & \\
\text{s.t.} & \mu^T \hat{x} & & & & & - & E_0 s & \geq & 0 \\
& \hat{x} & - & \hat{w} & + & \hat{v} & - & \bar{x} s & = & 0 \qquad (POCC) \\
& & & (c_B + e)^T \hat{w} & + & (c_S - e)^T \hat{v} & & & = & 0 \\
& & - & c_B^T \hat{w} & - & c_S^T \hat{v} & + & s & = & 1 \\
& & & & & \hat{w}^T \hat{v} & & & = & 0 \\
& & & & & \hat{x}, \hat{w}, \hat{v}, s & \geq & 0. & &
\end{array}$$

This change of variables is an extension of the standard method of Charnes and Cooper [3]. It is valid since the nonnegativity of w, v, c_B and c_S force s to be greater than one.

The criterion given in Figure 3 used upper bounds on the variables. Since we wish to avoid buying and selling the same security, we can impose the constraints $w \leq e - \bar{x}$ and $v \leq \bar{x}$. These do not translate into simple upper bounds on $\hat{w}$ and $\hat{v}$ in $(POCC)$. For our computational results, the costs c_B and c_S were small enough that we could use a slight weakening of the bounds in (SDP). We performed the calculation in the last step of Figure 3 with the original variables w and v.

In §6, we will discuss the computational performance of our algorithm when applied to this quadratic program with complementarity constraints. It should be noted that there are alternative methods to solve this problem; see Mitchell and Braun [28] for an exact algorithm. Other approaches to portfolio problems with transaction costs are surveyed in [28]; these include models with fixed costs per transaction, models placing transaction costs directly in the objective function, models with price breaks for different size transactions, dynamic rebalancing methods, stochastic programming approaches, and models for the related index tracking problem.

[3] If there exists a security for which the transaction costs are zero, then it is not necessary to introduce separate buy and sell variables for this security. To simplify the presentation, we assume that at least one of the transaction costs for each security is nonzero.

5 Degeneracy

A degenerate decision is one where both members of a complementary pair are zero in the best found solution to a quadratic programming subproblem. In the context of our solution strategy, degeneracy has the following effects. At the very least, degeneracy causes us to exclude more possibilities from consideration than can really be justified. At worst, degeneracy can be one obstacle that prevents us from identifying the optimal solution. Degeneracy relates to the accuracy — or better said — inaccuracy of our information regarding the optimal active set.

Our response to degenerate decisions will be to enumerate more possibilities. However, we obviously want to generate and solve as small a total number of QP-subproblems as possible. So if we respond to degeneracy by solving additional QP-subproblems, which problems are sensible? It turns out that the geometry of the feasible region gives us a rationale for which additional QP-subproblems should be considered.

The exploration of the feasible region has some similarities with the active set approach of Scholtes [31]. Scholtes also looks at different quadratic programming subproblems and looks to move between them by imposing varying combinations of active constraints. Typically the active constraints will include both indices in some complementary pairs and this imposed degeneracy is then relaxed based on consideration of Lagrange multipliers.

For the rest of this section, we will discuss degeneracy in the context of the portfolio optimization of §4. This discussion generalizes easily. A degenerate decision in the portfolio optimization problem is to neither buy nor sell a security i, so $x_i = \bar{x}_i$. Figure 4 illustrates a three security problem where decisions regarding x_1 and x_2 were originally made following solution of (SDP). Intersecting these decisions identifies the two faces which must be considered. The best observed solution in this set of QP-subproblems is located by the point A.

By inspection, point A contains a degenerate decision for x_2. Geometrically, this is recognizable since point A lies on the decision boundary $x_2 = \bar{x}_2$. The most conservative response is to simply "unmake" the degenerate decision for x_2. This would mean that the only enforceable decision boundary concerns x_1. Being to the left of the decision boundary $x_1 = \bar{x}_1$ means that one additional subproblem corresponding to the additional shaded face must be examined. Geometrically, that additional subproblem corresponds to a neighboring face since it shares a boundary.

To finish off our three dimensional example, Figure 5 presents the corresponding tree for the example of Figure 4. The same shading scheme is used and the QP-subproblem which gave rise to solution A is labeled. One observation is that although there are $2^3 = 8$ possible active sets, there are only 6 faces in the feasible region. The explanation is that a possible active set exists for each possible buy and sell combination. However, two of these active sets are immediately known to be infeasible. The infeasible active sets correspond to buying every security or selling every security. This cannot be consistent with the full investment constraint and has no geometrical realization. As a technical aside, the response to the degenerate decision

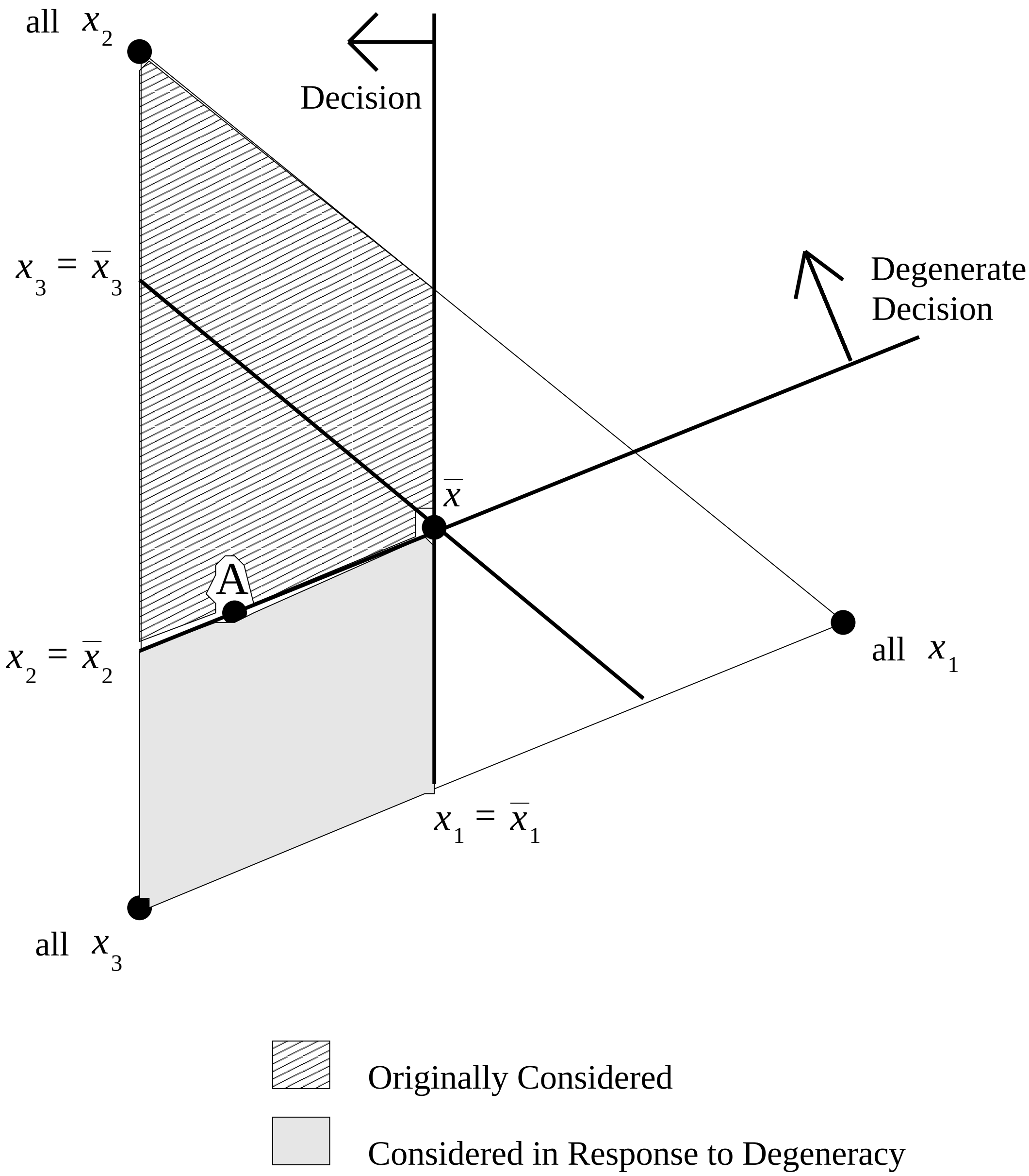

Figure 4: This figure idealizes the feasible region of the three security rebalancing problem. Point A is a degenerate solution which could be optimal. However, the additional neighboring face must be examined before that could be decided with any confidence.

regarding x_2 brings one of those infeasible QP-subproblems into consideration.

There are multiple ways that additional QP-subproblems can be generated in response to degeneracy. The approach which has already been discussed is conservative in the sense that it "unmakes" any decision that is found to be degenerate. This means that the number of additional subproblems grows exponentially each time a degenerate decision is found. Other approaches are more aggressive in the sense that fewer additional QP-subproblems are considered in response to degeneracy. The two approaches we will present here both rely on decision-making criteria which are similar to the one introduced in the final step of Figure 3. However for the recursive degeneracy check, the best observed QP-subproblem solution vector is supplied as the estimate x^*.

Before beginning our presentation of two specific degeneracy responses, an implicit characteristic of any QP-subproblem solution should be made explicit. Notice that the solution of every QP-subproblem has already had complementarity imposed. That is to say there are only two possible estimates for each complementary pair: either $(x_i^*, 0)$ or $(0, x_j^*)$ where $x_i^*, x_j^* \geq 0$ depends on which variable was assumed to be active for that subproblem. This allows us to construct a specialized decision criteria for each approach.

5.1 The conservative choice

The first approach is the most conservative way to grow the tree. It reverses any decision that is found to be degenerate and so doubles the number of QP-subproblems for each degenerate decision. Suppose an initial activity decision was made for the complementary pair (x_i, x_j). The conservative decision criteria can be expressed in the following manner:

> If $x_j \geq 0$ had been considered an active constraint but $x_i^* \approx 0$ then $x_j \geq 0$ should no longer be forced to be active.

The analytic statement $x_i^* \approx 0$ was implemented as $x_i^* \leq \epsilon$ where ϵ is a numerical tolerance, for example $\epsilon = 10^{-8}$.

5.2 A more aggressive approach

Once an optimal solution is found for any QP-subproblem, several useful pieces of information are known. First, that QP-subproblem objective value is an upper bound on the optimal value of $(QPCC)$ just as (SDP) was a lower bound on $(QPCC)$. Second — and of more direct interest — that solution vector constitutes an estimate of the complete optimal active set. Even with complementary pairs of variables for which no activity decision was made, decisions regarding what was bought and what was sold can be observed. The previous conservative approach discarded that information. This more aggressive approach makes use of that information.

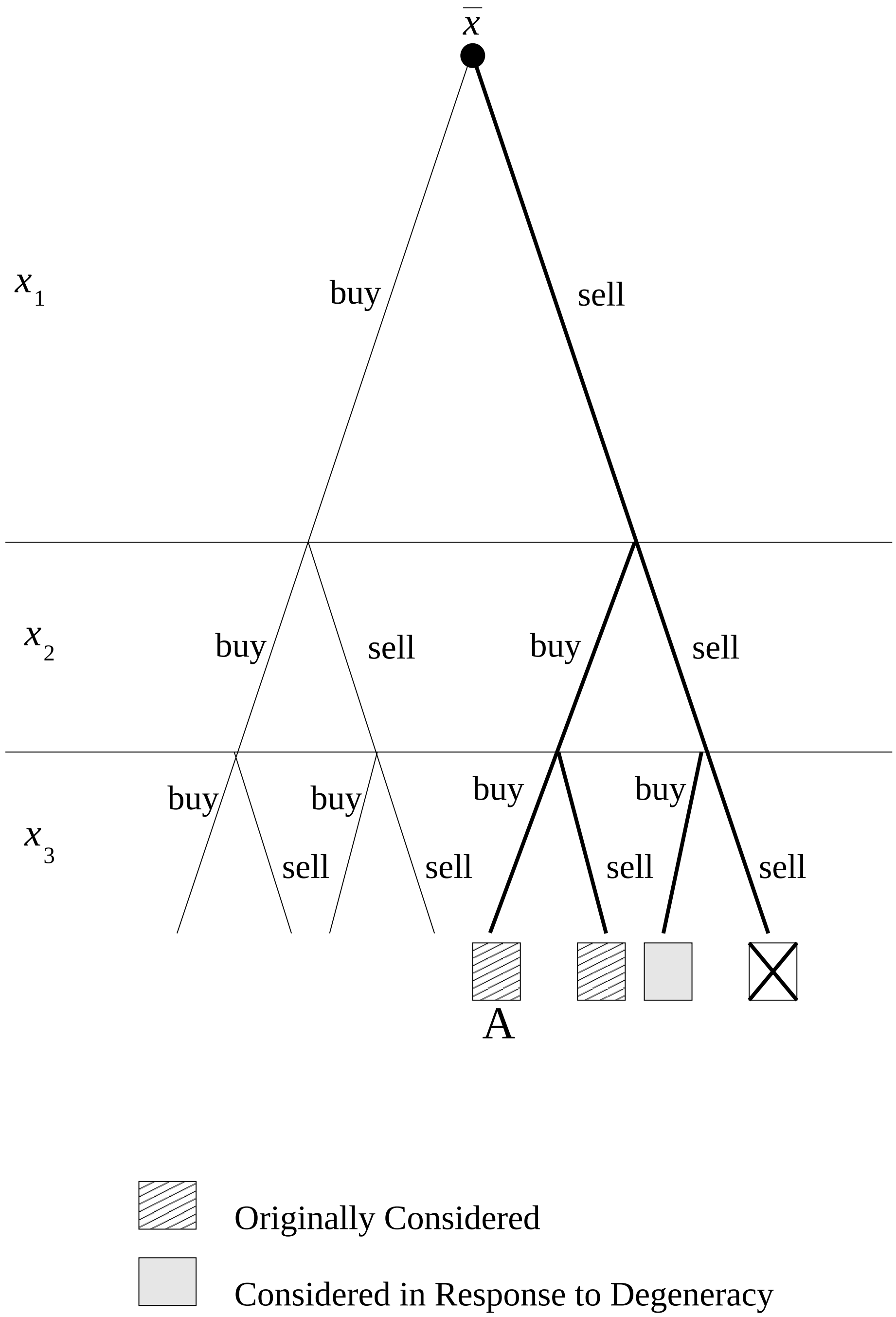

Figure 5: This figure shows the tree corresponding to Figure 4. Again, 2 subproblems are initially considered. In response to degeneracy, additional neighboring subproblems are considered. Bold lines indicate which branches were followed and connect to the leaves that were actually considered.

Again, let x^* be the best observed QP-subproblem optimal solution vector. All determinations of active constraints found by the conservative approach are observed. Additional requirements are imposed, in order to restrict the number of QP-subproblems. Suppose no initial activity decision was made for the complementary pair (x_i, x_j). The aggressive approach also uses (x_i^*, x_j^*) as its estimate and applies the following decision criteria:

If $\frac{x_i^*}{u_i} \geq \bar{U}_{tol}$ then $x_j \geq 0$ should now be considered an active constraint.

Here, $\bar{U}_{tol}$ is a user-specified tolerance. It is important to stress that nondegenerate decisions are never reversed by this criteria.

In simplest terms, the aggressive approach keeps all decisions which can be inferred from the best observed solution, while reversing any degenerate decisions made in Figure 3. Figures 6 and 7 show in sequence an example of how this approach works. With the aggressive approach, the number of additional QP-subproblems that will be enumerated in response to degeneracy grows linearly in the number of degenerate variables.

We are considering an NP-hard optimization problem so we cannot guarantee that our strategy with either a conservative or aggressive degeneracy response will always identify the optimal solution. Being aggressive by considering fewer subproblems is a systematic way to reduce the amount of computation but does not guarantee better performance. Being conservative systematically increases the number of subproblems brought into consideration but still does not provide a guarantee of optimality. However, in real-world situations where suboptimal solutions offering incremental improvement have their own value, this variety in approach is a useful feature of our strategy.

6 Computational results

We discuss two portfolios in this section, a nine-security one due to Markowitz [26] in §6.1, and a portfolio consisting of the thirty stocks in the Dow Jones Industrial Average in §6.2. More data on these problems can be found in Braun [2].

As the parameter E_0 is varied, an efficient frontier is traced out. When the transaction costs are zero, this gives the classical Markowitz efficient frontier. For nonzero transaction costs, we obtain what we refer to as a transaction cost efficient frontier, or TCEF.

6.1 Performance analysis for the Markowitz portfolio

Each point on a nonzero transaction cost efficient frontier corresponds to a separate instance of $(QPCC)$. Each instance is a separate test of how well our solution strategy performs. Looking at a specific frontier or even just at a specific point, performance information regarding our solution strategy can be presented.

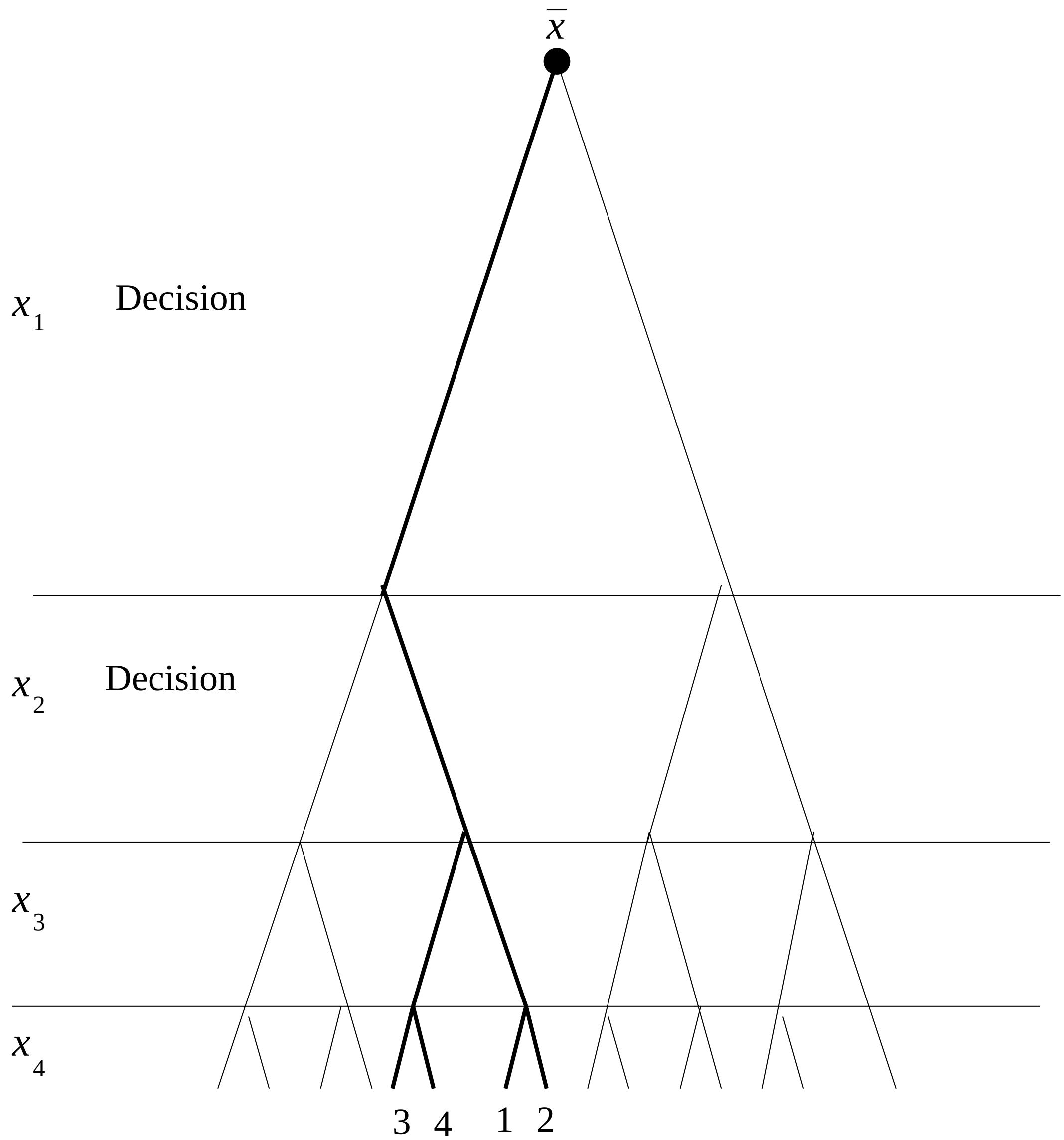

Figure 6: This figure presents the example of a four layer tree in which two decisions are initially made following the solution of (SDP). Again, bold lines indicate where decisions were made and which leaves are actually considered. The observed solutions are ranked by objective value. The best observed solution is numbered 1. Assume that x_3 and x_4 take on optimal values within $\mathrm{U}_{tol}\%$ of their respective upper bounds.

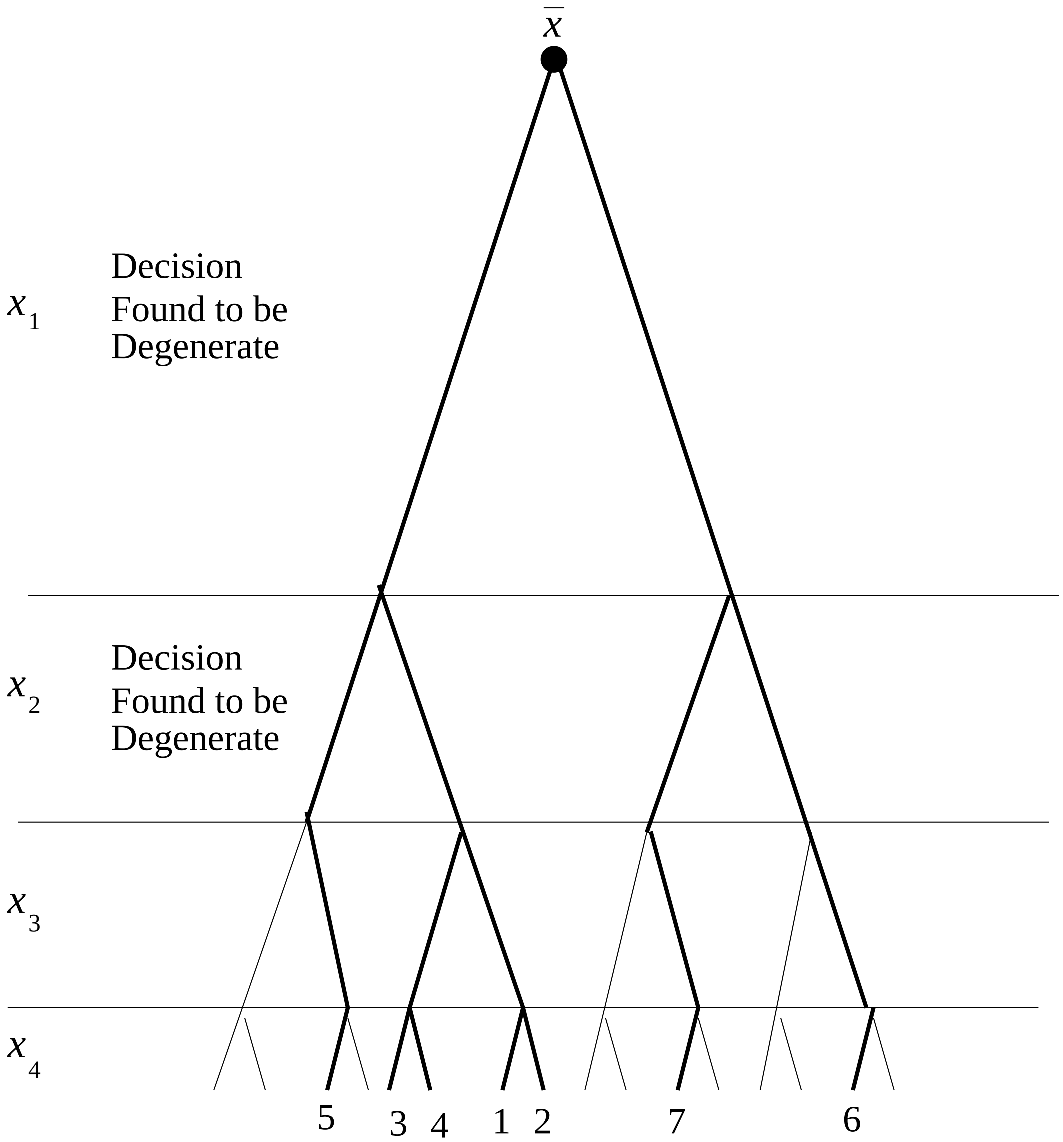

Figure 7: This figure shows the aggressive response to degeneracy. Since both decisions were found to be degenerate then the conservative approach would force consideration of all 16 leaves. Using the inferred nondegenerate decisions from solution #1, a branching structure for x_3 and x_4 can be translated to select only subproblems #5, 6, and 7. Notice that with the aggressive approach only 7 of the 16 subproblems are considered.

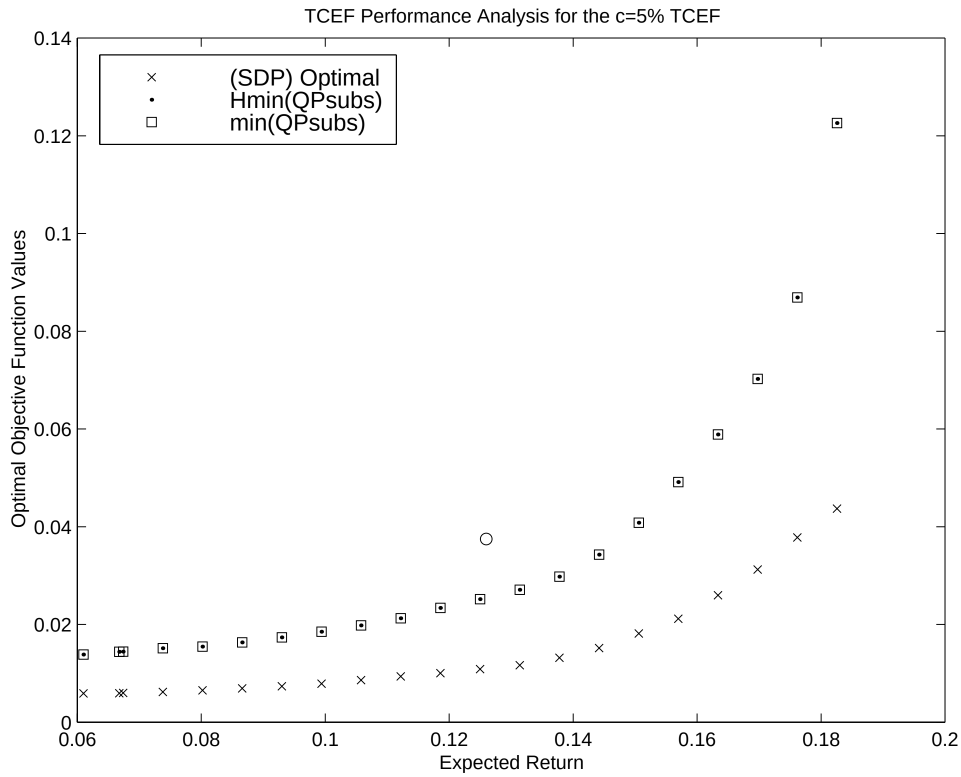

Figure 8: The circle locates the initial portfolio. This figure presents information regarding the performance of our solution strategy.

We assume the initial portfolio is equally weighted in the nine securities and that the costs are 5% for both buying and selling any security. Figure 8 shows the c=5% TCEF. For this TCEF, the Spectral Bundle solver [18] was allowed a 10 minute time limit to solve each instance of (SDP). For all instances, (SDP) was solved to optimality before that time limit was reached. A conservative set of tolerances ($L_{tol} = 10\%$ and $U_{tol} = 90\%$) was used to make activity decisions. The terms used in Figure 8 have the following meanings:

- "Hmin(QPsubs)" points are found using our solution strategy.
- "min(QPSubs)" points result from exhaustively considering all possibilities.
- Finally, the series "(SDP) Optimal" locates the optimal value of the relaxation.

Looking at Figure 8, several observations can be made. To provide orientation, comparisons can be made for each level of required expected return. This corresponds to vertical comparisons in the plot. First, notice that our solution strategy correctly identified the true optimal solution in every instance along this frontier. This is indicated by the "dot within a square." A second observation is that the optimal

Required Expected Return (%)	(SDP) Objective Value	Minimal QP-subproblem Objective Value	Number of QP-subproblems Considered
18.26	0.043725	0.122595	n.a.
17.62	0.037793	0.086903	16
16.98	0.031234	0.070248	16
16.34	0.025977	0.058898	16
15.70	0.021162	0.049131	16
15.06	0.018161	0.040815	16
14.42	0.015173	0.034284	16
13.78	0.013183	0.029781	16
13.14	0.011649	0.027103	16
12.50	0.010877	0.024602	16
11.86	0.010024	0.023381	32
11.22	0.009383	0.021267	32
10.58	0.008607	0.019820	32
9.94	0.007897	0.018517	32
9.30	0.007363	0.017363	32
8.66	0.006915	0.016326	32
8.02	0.006528	0.015471	16
7.38	0.006182	0.015153	16
6.74	0.005994	0.014462	16
6.68	0.005937	0.014426	16
6.10	0.005909	0.013850	8

Table 1: Since 9 securities were involved, the fourth column compares against a total of $2^9 = 512$ possible QP-subproblems. Notice that for most cases, correct decisions were made for 5 out of the 9 securities. No degenerate solutions were encountered during this calculation.

value of (SDP) does appear to be a consistently tight lower bound on the true optimum.

Table 1 contains the numerical results illustrated in Figure 8. It also quantifies the assertion that our strategy considered a relatively small number of QP-subproblems in order to find the optimal solution. To explain the lone "n.a." entry in Table 1, the highest returning point on any TCEF is directly solvable as a linear program. Therefore, equivalent performance information regarding the number of QP-subproblems is not applicable.

6.1.1 Examination of the QP-subproblems selected by the strategy

Table 1 demonstrates that our strategy selected a relatively small subset from the set of all possible QP-subproblems. However, the optimal subproblem was located

Optimal Objective Values for the 10 Best QP-subproblems	Optimal Objective Values for QP-subproblems Selected by the Strategy	Comment on Relationship
0.024602	0.024602	Strategy selected true optimal solution.
0.024703		Not selected, see Figure 9.
0.024722	0.024722	Selected by our strategy.
0.024737	0.024737	Selected by our strategy.
0.024875	0.024875	Selected by our strategy.
0.024877	0.024877	Selected by our strategy.
0.025168	0.025168	Selected by our strategy.
0.025186		
0.025238		
0.025240		
	0.033665	Feasible but not close to optimal.
	0.034617	Feasible but not close to optimal.
	0.035161	Feasible but not close to optimal.

Table 2: Notice that the QP-subproblems selected by our strategy are densely clustered near the true optimum. More specifically, the strategy selected the true optimal and five other subproblems located within 2% of the true optimum.

within that subset. Was this simply luck or can we provide some additional support for the performance of our strategy? Support can and does come from an analysis of the distribution of the QP-subproblems selected by our strategy against the set of all possible subproblems.

We now choose to focus on a single representative point rather then the entire Markowitz c=5% TCEF. The point chosen corresponds to a required expected return of 12.50%. This is close to the return already offered by the initial portfolio since $\mu^T \bar{x} = 12.60\%$. Exhaustively searching all 512 possible QP-subproblems identifies that only 290 subproblems are feasible. Each feasible QP-subproblem has an optimal objective value which we sort to identify the 10 Best QP-subproblems. Here, "Best" refers to the property that from the set of all possibilities, these QP-subproblems have the 10 smallest optimal objective values.

A similar examination of the 16 QP-subproblems that were selected by our strategy finds that 9 are feasible. The optimal objective values of these 9 QP-subproblems can also be sorted by objective value. Table 2 presents a side-by-side comparison of the results.

In Table 2, the "2^{nd} Best" QP-subproblem with objective value 0.024703 was not selected by our strategy. This would certainly seem to weaken this analysis. However, looking at the actual portfolios which are optimal in each subproblem is enlightening. Figure 9 below presents that information. The vertical ordering and side-by-side

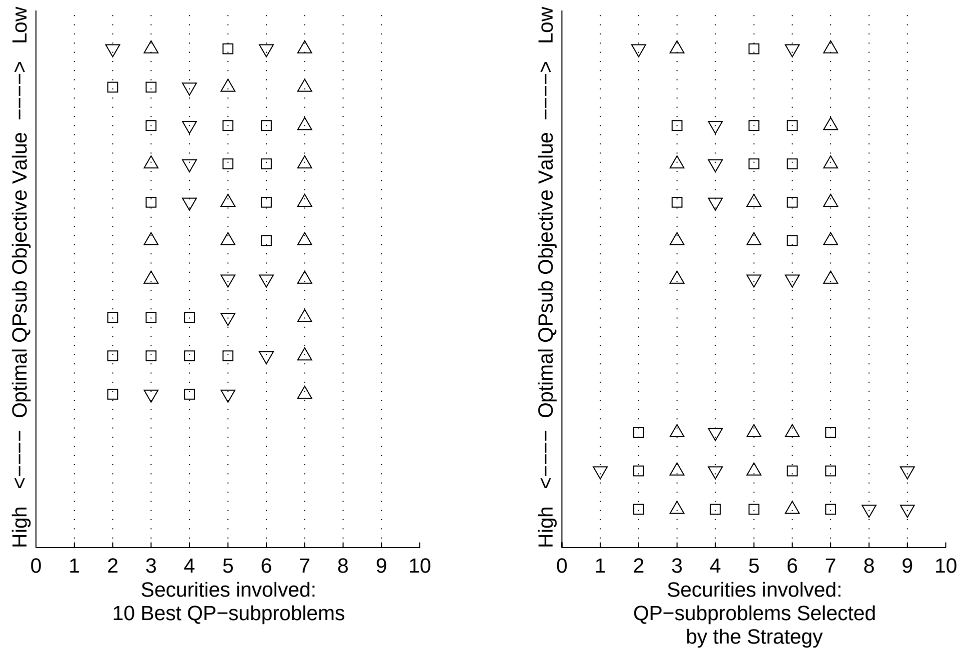

Figure 9: Securities numbered 1 through 9 exist. No activity decisions were made for securities #3, 5, 6, and 7. The first and second rows of the lefthand panel are of particular relevance.

comparison format of Table 2 is replicated in Figure 9. Symbols indicate whether the optimal allocation of a particular security was unchanged, increased, or decreased from its initial value. The symbols used are: bought $\triangle$, sold $\bigtriangledown$, unchanged $\square$. The absence of a symbol means that the entire allocation of that particular security was liquidated.

Notice that the optimal solution of the 2^{nd} Best QP-subproblem is feasible in the true optimum. Table 3 highlights this by showing the active set choices for these subproblems. This subproblem was not selected because it violated the activity decision to sell security #2. As evidenced by the true optimum, that sell decision was in fact correct. So, an analysis of the optimal solutions supports the performance of our strategy, at least for this instance.

To conclude this analysis, we observe that our strategy selected a subset of QP-subproblems which disproportionately populates the most promising region of the search space. This is exactly the behavior which was the original goal of our solution strategy.

6.1.2 Presentation of the Markowitz c=5% TCEF portfolios

As was mentioned earlier, the actual real-world rebalancing process requires information about the actual amounts of each security that should be bought or sold. Before leaving this TCEF, Figure 10 shows which securities are involved at each point along

Objective Values for the 2 Best QP-subproblems	Security Number 1	2	3	4	5	6	7	8	9
0.024602	Sell	Sell	Buy	Sell	Buy	Sell	Buy	Sell	Sell
0.024703	Sell	Buy	Buy	Sell	Buy	Sell	Buy	Sell	Sell

Table 3: This table contains the active set choices for the 2 Best QP-subproblems. Notice that the slightly suboptimal QP-subproblem solution is feasible in the true optimal. All decisions are the same except for security #2. Since this subproblem was constrained to buy security #2, the slightly suboptimal solution is a "Buy=0" decision.

the TCEF. We use the same symbology that was introduced for Figure 9. However, the vertical scale is now the minimum required expected return (ie. the horizontal coordinate) of the efficient frontier plots.

Figure 10 contains a great deal of useful information which can be accessed by visual inspection. Looking down a column, changes occurring in the portfolio as the required expected return is decreased can easily be seen. Also, horizontal comparisons between panels allow for comparisons to be made between the no-cost and c=5% frontiers. The portfolios along the TCEF are not simply related to the portfolios along the no transaction cost efficient frontier. Sometimes, entirely new securities are involved. Sometimes, buy and sell decisions are reversed. So Figure 10 highlights that the introduction of costs changes the portfolio rebalancing problem dramatically and that the optimal solutions are also quite different.

6.1.3 A degenerate case

The uniform portfolio allocation and cost structure is not the most realistic test for our solution strategy. Real portfolios change their relative composition over time as the better-returning securities grow more than those performing less well or those securities that decline outright. So nonuniform initial allocations are worth consideration. Also, to take full advantage of the flexibility of our cost model, nonuniform costs will be considered as well. The TCEF presented in this subsection also makes use of the Markowitz dataset. All nine securities are involved initially. The allocations were randomly selected as were the buying and selling costs. Those costs were kept on the order of a few percent. (These parameters are detailed in Braun [2].)

For one of the required returns for this instance, a degenerate optimal solution was obtained. In this solution, six of the nine nonnegativity constraints were determined to be active in the final step of Figure 3. For the conservative approach detailed in §5.1, 32 subproblems were solved altogether. With the aggressive approach of §5.2, only 11 subproblems needed to be solved. The optimal solution was found in each case, but only after examining the additional problems as detailed in §5.

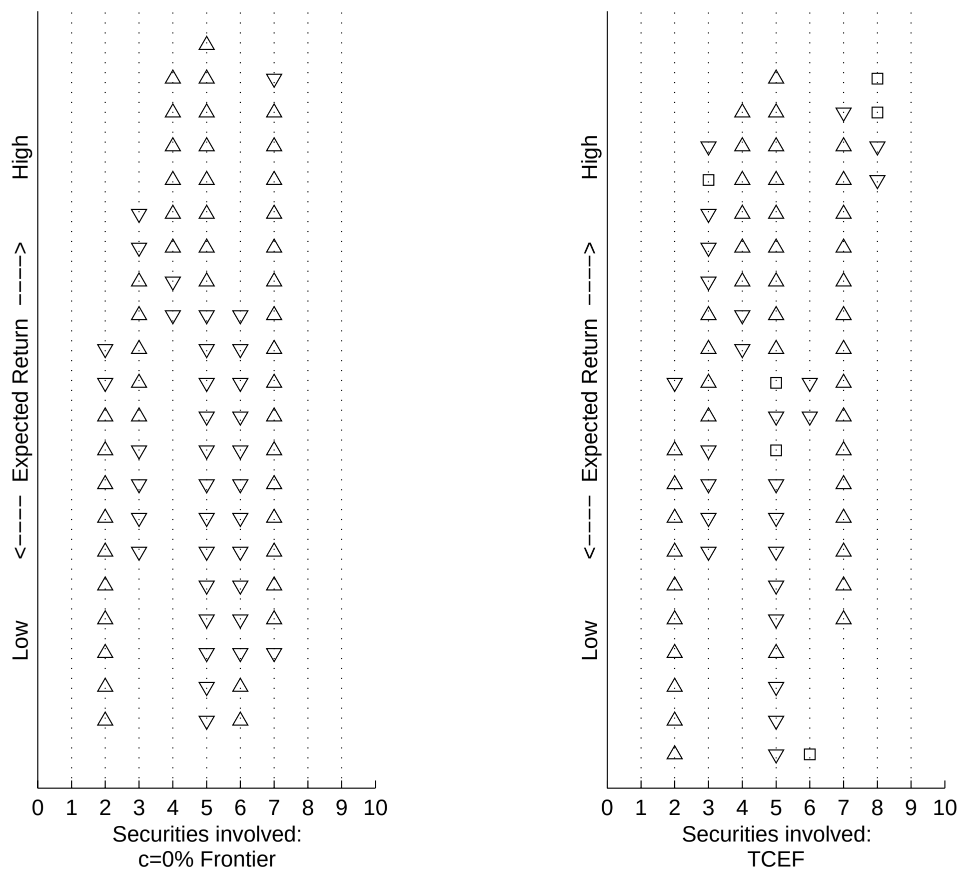

Figure 10: These panels show which securities are involved along the no-transaction cost efficient frontier (left) and the c=5% TCEF (right).

6.2 Rebalancing a Dow Jones portfolio

In this final set of computational results, we demonstrate the effectiveness of our strategy on a larger and so more realistic dataset. We applied our solution strategy to the problem of rebalancing portfolios composed of the 30 stocks which currently make up the Dow Jones Industrials Average. All securities were involved initially. With a portfolio of 30 securities, there are $2^{30} \sim 10^9$ or over a billion possible QP-subproblems. All securities were involved initially, with proportions varying from 1% to 5%. The buying and selling costs varied from security to security, from 0% to 5%. For the values of the parameters $\bar{x}$, c_B, and c_S, as well as for the return and risk data, see Braun [2].

For this TCEF, the Spectral Bundle solver was allowed 30 minutes to solve each instance of (SDP). However, no instance was solved to optimality before that time limit expired. As the caption of Figure 11 notes, this degrades how tight of a lower bound (SDP) provides for some points along the TCEF. However, the use of sub-optimal (SDP) solution matrices does not appear to have degraded the validity or performance of our decision-making process. Large numbers of activity decisions were made and no degenerate points were encountered. The same conservative set of decision tolerances was used as in §6.1. The performance of the strategy is illustrated in Figure 11, with the corresponding numerical results detailed in Table 4. Similar figures to Figure 10 can be constructed for this portfolio, allowing similar conclusions.

7 Conclusions

We have demonstrated an interesting approach to finding a good solution to a quadratic programming problem with complementarity constraints. The computational testing on portfolio optimization problems shows that the method holds promise. Further investigation of the method on larger problems arising from different applications is planned.

Acknowledgements

We would like to thank two anonymous referees for their constructive comments.

References

[1] E. Balas, S. Ceria, and G. Cornuéjols. A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Mathematical Programming*, 58:295–324, 1993.

[2] S. E. Braun. *Solving a quadratic programming problem subject to orthogonality constraints*. PhD thesis, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY, December 2001.

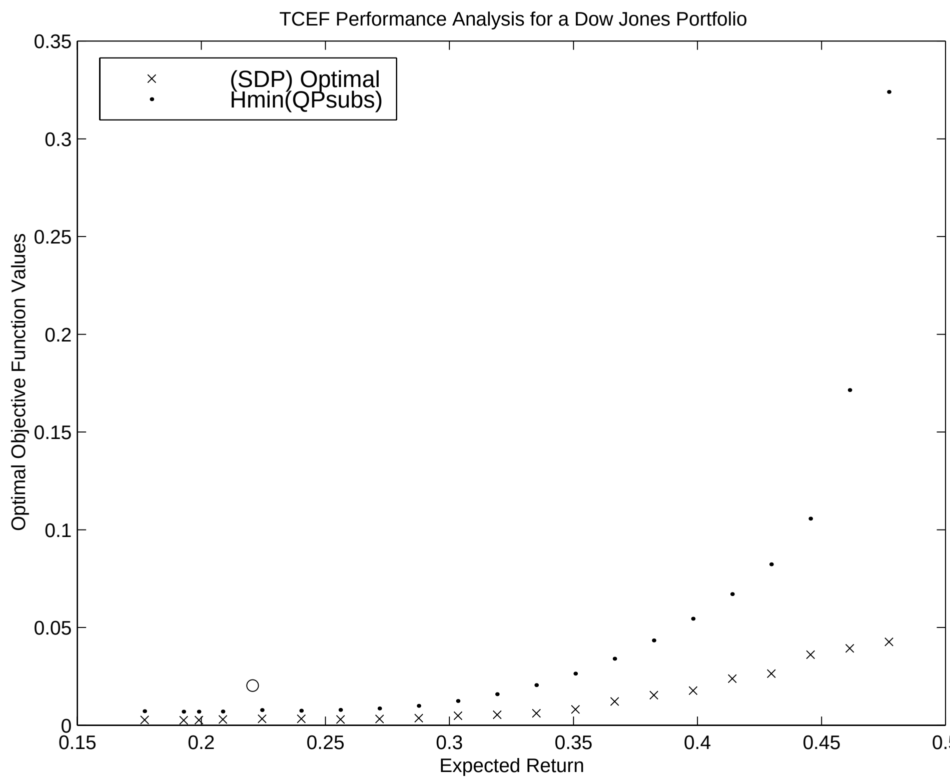

Figure 11: The circle again locates the initial portfolio. Notice that the (SDP) optimal objective value does not mirror the shape of the TCEF, particularly for higher required expected returns. This is a noticeable difference from Figure 8. However, the TCEF itself has no visible defects. This provides strong empirical support for our strategy.

Required Expected Return (%)	(SDP) Objective Value	Minimal QP-subproblem Objective Value	Number of QP-subproblems Selected by the Strategy
47.72	0.042696	0.324000	n.a.
46.14	0.039363	0.171472	256
44.56	0.036101	0.105743	256
42.98	0.026411	0.082315	256
41.40	0.023847	0.067093	256
39.82	0.017756	0.054479	256
38.24	0.015437	0.043378	256
36.66	0.012181	0.034012	512
35.08	0.008156	0.026464	1024
33.50	0.006154	0.020584	2048
31.92	0.005408	0.015929	1024
30.34	0.004849	0.012424	2048
28.77	0.003656	0.009964	256
27.19	0.003308	0.008580	512
25.61	0.003284	0.007910	256
24.03	0.003193	0.007751	2048
22.45	0.003009	0.007494	16384*
20.87	0.002962	0.007234	1024
19.90	0.002736	0.007011	256
19.29	0.002661	0.006952	256
17.71	0.002534	0.006923	256

Table 4: For this dataset, the fourth column compares against a total of $2^{30} \sim 10^9$ or over a billion possible QP-subproblems. For most cases, activity decisions were made for 22 out of the 30 securities. The third column corresponds to the "Hmin(QPsubs)" series in Figure 11. No degenerate solutions were encountered. For the point marked by the asterisk, the calculation was stopped after 2048 QP-subproblems were considered.

[3] A. Charnes and W. W. Cooper. Programming with linear fractional functionals. *Naval Research Logistics Quarterly*, 9:181–186, 1962.

[4] J. W. Demmel. *Applied Numerical Linear Algebra.* SIAM, Philadelphia, PA, 1997.

[5] M. C. Ferris, S. P. Dirkse, and A. Meeraus. Mathematical programs with equilibrium constraints: automatic reformulation and solution via constrained optimization. Technical Report 00–09, Computer Sciences Department, University of Wisconsin, Madison, WI, July 2002.

[6] M. C. Ferris and J. S. Pang. Engineering and economic applications of complementarity problems. *SIAM Review*, 39(4):669–713, 1997.

[7] R. Fletcher and S. Leyffer. Numerical experience with solving MPECs as NLPs. Technical Report NA 210, Department of Mathematics and Computer Science, University of Dundee, Dundee DD1 4HN, UK, August 2002.

[8] R. Fletcher, S. Leyffer, D. Ralph, and S. Scholtes. Local convergence of SQP methods for mathematical programs with equilibrium constraints. Technical Report NA 209, Department of Mathematics and Computer Science, University of Dundee, Dundee DD1 4HN, UK, May 2002.

[9] M. Fukushima, Z.-Q. Luo, and J. S. Pang. A globally convergent sequential quadratic programming algorithm for mathematical programs with linear complementarity constraints. *Computational Optimization and Applications*, 10:5–34, 1998.

[10] M. Fukushima and P. Tseng. An implementable active-set algorithm for computing a B-stationary point of the mathematical program with linear complementarity constraints. Technical report, Department of Mathematics, University of Washington, Seattle, WA, October 1999.

[11] G. H. Golub and C. F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, 1983.

[12] C. Helmberg. Semidefinite programming homepage, 1996. http://www-user.tu-chemnitz.de/~helmberg/semidef.html.

[13] C. Helmberg. Fixing variables in semidefinite relaxations. *SIAM Journal on Matrix Analysis and Applications*, 21(3):952–969, 2000.

[14] C. Helmberg. SBmethod: a C++ implementation of the spectral bundle method. Technical Report 00–35, TU Berlin, Konrad-Zuse-Zentrum, Berlin, October 2000.

[15] C. Helmberg. Semidefinite programming for combinatorial optimization. Technical Report ZR-00-34, TU Berlin, Konrad-Zuse-Zentrum, Berlin, October 2000. Habilitationsschrift.

[16] C. Helmberg. A cutting plane algorithm for large scale semidefinite relaxations. Technical Report 01–26, TU Berlin, Konrad-Zuse-Zentrum, Berlin, October 2001.

[17] C. Helmberg and F. Rendl. Solving quadratic (0,1)-problems by semidefinite programs and cutting planes. *Mathematical Programming*, 82:291–315, 1998.

[18] C. Helmberg and F. Rendl. A spectral bundle method for semidefinite programming. *SIAM Journal on Optimization*, 10(3):673–696, 2000.

[19] R. A. Horn and C. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, 1985.

[20] K. Krishnan. *Linear programming approaches to semidefinite programming problems*. PhD thesis, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, July 2002.

[21] K. Krishnan and J. E. Mitchell. A unifying framework for several cutting plane methods for semidefinite programming. Technical report, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, November 2002. Revised: December 15, 2003.

[22] S. Leyffer. The penalty interior point method fails to converge for mathematical programs with equilibrium constraints. Technical Report NA 208, Department of Mathematics and Computer Science, University of Dundee, Dundee DD1 4HN, UK, February 2002.

[23] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1(2):166–190, 1991.

[24] Z.-Q. Luo, J.-S. Pang, and D. Ralph. *Mathematical Programs with Equilibrium Constraints*. Cambridge University Press, Cambridge, 1996.

[25] H. Markowitz. Portfolio selection. *Journal of Finance*, 7:77–91, 1952.

[26] H. Markowitz. *Portfolio selection: efficient diversification of investments*. Blackwell, New York, 2nd edition, 1991.

[27] J. E. Mitchell. Restarting after branching in the SDP approach to MAX-CUT and similar combinatorial optimization problems. *Journal of Combinatorial Optimization*, 5(2):151–166, 2001.

[28] J. E. Mitchell and S. Braun. Rebalancing an investment portfolio in the presence of transaction costs. Technical report, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, November 2002.

[29] J. V. Outrata and J. Zowe. A numerical approach to optimization problems with variational inequality constraints. *Mathematical Programming*, 68(1):105–130, 1995.

[30] H. Scheel and S. Scholtes. Mathematical programs with complementarity constraints: stationarity, optimality, and sensitivity. *Mathematics of Operations Research*, 25(1):1–22, 2000.

[31] S. Scholtes. Active set methods for inverse linear complementarity problems. Technical report, Judge Institute of Management Science, Cambridge University, Cambridge CB2 1AG, November 1999.

[32] S. Scholtes. Convergence properties of a regularisation scheme for mathematical programs with complementarity constraints. *SIAM Journal on Optimization*, 11(4):918–936, 2001.

[33] S. Scholtes. Combinatorial structures in nonlinear programming. Technical report, Judge Institute of Management Science, Cambridge University, Cambridge CB2 1AG, April 2002.

[34] H. D. Sherali and W. D. Adams. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems.* Number 31 in Nonconvex Optimization and its Applications. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[35] M. J. Todd. Semidefinite optimization. *Acta Numerica*, 10:515–560, 2001.

[36] K. C. Toh, M. J. Todd, and R. Tutuncu. SDPT3 — a Matlab software package for semidefinite programming. *Optimization Methods and Software*, 11:545–581, 1999.

[37] K. C. Toh, M. J. Todd, and R. Tutuncu. SDPT3 — a Matlab software package for semidefinite-quadratic-linear programming, version 3.0. Technical report, Department of Mathematics, National University of Singapore, Singapore, August 2001.

[38] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38:49–95, 1996.

[39] H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.