Randomized Portfolio Selection, with Constraints

Jiang XIE Simai HE Shuzhong ZHANG *

December 20, 2006

Abstract

In this paper we propose to deal with the combinatorial difficulties in mean-variance portfolio selection, caused by various side constraints, by means of randomization. As examples of such side constraints, we consider in this paper the following two models. In the first model, an investor is interested in a small, compact portfolio, in the sense that it involves only a small number of securities. The second model explicitly requires that each security involved in the portfolio need to have a substantial presence if it is present at all, implicitly restricting the number of them given the budget constraints. These constraints are motivated by practical considerations in the face of management and informational costs in investment. By incorporating such side constraints, however, the mean-variance model becomes very hard to solve. We resort to the method of randomization for finding good approximation solutions. Extensive numerical experiments show that randomization is indeed a viable alternative for solving such hard investment models, for which the combinatorial complexity in the constraints makes it quite hopeless to find an exact solution, while good approximate solutions in fact already serve the purpose quite well given the approximative nature of the models.

Keywords: mean-variance model, randomization method, SDP relaxation, approximation ratio.

MSC subject classification: 91B28, 68W25, 68W20.

*Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, Hong Kong. Email: zhang@se.cuhk.edu.hk. Research supported by Hong Kong RGC Earmarked Grant CUHK418406.

1 Introduction

The classical Markowitz mean-variance portfolio selection model (see [7]) has been the cornerstone for the portfolio theory in the last half century. In spite of its many shortcomings, principles underlying the simple mean-variance model still guide the theory and practice of portfolio selection till this day. Among a significant number of papers in the literature refining the original mean-variance model, we mention the following modifications: Markowitz [8] proposed to replace variance by semi-variance as a more plausible measure for risk; Zhou and Li [11] extended the model to a continuous framework; Goldfarb and Iyengar [4] studied the data sensitivity issue of the original model and consequently proposed a robust mean-variance model. The aim of the current paper is quite different. Instead of stretching the model, we shall squeeze it. One practical issue for an investor is often the management costs of maintaining a large and complicated portfolio, not least from an information-gathering point of view. Thus, a desirable feature of an investment portfolio is that its composition should be compact, sensible, and manageable. Blog, Van der Hoek, Rinnooy Kan, and Timmer [3] called a portfolio with a small number of securities as a *small portfolio* – a term we shall borrow here – and discussed solution methods for such problems. In a similar vein, Bienstock [2], and more recently Bertsimas and Shioda [1], studied the *exact* solution methods for solving the problem, based on either the dynamic programming principle or some clever branch-and-bound schemes. Our approach in this paper is quite different. As investment is a business where uncertainties and ambiguities are a part of the nature, it is arguable that any model can only be a rough approximation of the reality. However, if a model is inexact, then it makes sense to treat it no more than a guiding reference point, rather than an unbendable iron object. In light of this, the method of randomization becomes attractive. This paper is devoted to studying the application of randomization methods for selecting a portfolio with some sort of compactness (thus combinatorially hard) constraints. In Section 2, we consider the problem of choosing a portfolio with a small number of assets, or, for brevity, the problem of choosing a small portfolio, in the spirit of [3]. Two slightly different methods are introduced, with different flavors for randomization. We further consider in Section 3 the problem of choosing a *clean and substantial* portfolio, in that once an asset is present then its presence must be substantial, say, no less than 10% of the entire volume in the absolute term. With these considerations, we attempt to strike a balance among three factors in investment: (1) the expected return; (2) the risk (to be controlled by diversification); (3) the management costs of the investment. In Section 4, we present the numerical results using simulated data, in order to evaluate the performance of these randomization methods. The notations to be used are as follows:

E:	mathematical expectation of a random variable or random vector;
Var:	the variance of a random variable;
Cov:	the covariance matrix of two random vectors;
$\circ$:	the Hadamard product;
$\mathrm{sign}(x)$:	the sign function, i.e. $\mathrm{sign}(x) = 1$ if $x > 0$, $\mathrm{sign}(x) = -1$ if $x < 0$, and $\mathrm{sign}(0) = 0$;
e:	the all-one vector with an appropriate dimension;
C_n^k:	the combinatorial number of choosing k elements from the set of total n elements;
$\mathcal{S}^n$:	the set of all $n \times n$ symmetric matrices.

2 Selecting a 'Small' Portfolio

Consider the following portfolio selection problem, where there are in total n available securities, and the investment budget is scaled to be 1. Suppose that the first two moments of the return of the assets are r (expected return) and Q (covariance matrix) respectively. Furthermore, we assume that short-selling is allowed. As in the standard mean-variance model, we use the variance of the portfolio return as the risk measure, and we set μ to be the target expected rate of return. The only difference is that we are now only interested in a 'small' portfolio ([3]), i.e., a portfolio with no more than k ($k << n$) securities. The model now becomes:

$$\begin{array}{llll} (MV_s) & \min & x^T Q x & \\ & \text{s.t.} & r^T x \geq \mu, & \\ & & e^T x \leq 1, & \\ & & \sum\limits_{i=1}^{n} |\text{sign}(x_i)| \leq k. & \end{array} \tag{1}$$

The last notion is sometimes known as the L_0-norm of x, i.e., $\|x\|_0 := \sum_{i=1}^n |\text{sign}(x_i)|$, and so the constraint can also be written as $\|x\|_0 \leq k$. In other words, a portfolio is called 'small' if its L_0-norm is small. Note, however, that the so-called L_0-norm is not actually a norm; it is actually non-convex.

Clearly, the last constraint introduces a great deal of combinatorial complexity, and in fact makes the problem NP-hard. From a practicality point of view, however, this constraint is almost indispensable. Instead of attempting to solve (1) to optimality either by dynamic programming or by branch-and-bound (see e.g. [3, 2, 1]), we take an entirely different approach, viz. we shall consider some kind of randomization schemes to deal with the combinatorial complexity issue encountered in solving (1) to optimality. Suppose that we have setup a scheme to randomly pick up a small portfolio. Then, the question may be posed as: what will be the quantity of each security once it is chosen? In other words, as an approximation we may decompose the problem into two independent decision processes: (1) which assets to pick; and (2), what quantity to buy once an asset is selected. The following lemma is useful in our subsequent analysis.

Lemma 1. *Suppose that ξ^1 and ξ^2 are two mutually independent n dimensional random vectors, with finite first two moments. It holds that*

$$\text{Cov}(\xi^1 \circ \xi^2) = \text{Cov}(\xi^1) \circ \text{Cov}(\xi^2) + \text{Cov}(\xi^1) \circ (\mathsf{E}\xi^2 (\mathsf{E}\xi^2)^T) + \text{Cov}(\xi^2) \circ (\mathsf{E}\xi^1 (\mathsf{E}\xi^1)^T).$$

Proof. Observe for any index pair $i \leq i, j \leq n$ that

$$\begin{aligned} & \mathsf{E}(\xi^1 \circ \xi^2)_{ij} = \mathsf{E}(\xi_i^1 \xi_i^2 \xi_j^1 \xi_j^2) = \mathsf{E}(\xi_i^1 \xi_j^1) \mathsf{E}(\xi_i^2 \xi_j^2) \\ = \; & (\text{Cov}(\xi^1)_{ij} + \mathsf{E}(\xi^1)_i \mathsf{E}(\xi^1)_j) \times (\text{Cov}(\xi^2)_{ij} + \mathsf{E}(\xi^2)_i \mathsf{E}(\xi^2)_j) \\ = \; & \text{Cov}(\xi^1)_{ij} \text{Cov}(\xi^2)_{ij} + \mathsf{E}(\xi^1)_i \mathsf{E}(\xi^1)_j \times (\text{Cov}(\xi^2)_{ij} \\ & + \mathsf{E}(\xi^2)_i \mathsf{E}(\xi^2)_j \text{Cov}(\xi^1)_{ij} + \mathsf{E}(\xi^1)_i \mathsf{E}(\xi^1)_j \mathsf{E}(\xi^2)_i \mathsf{E}(\xi^2)_j. \end{aligned}$$

Rearranging yields the desired result. □

In the next two subsections we consider two slightly different randomization schemes. The first one, to be presented in Subsection 2.1, takes the view that the cardinality constraint is not necessary a strict constraint, in the sense that there might be some flexibility in enforcing such constraint, and thus focuses on a selection process which will generate a small portfolio with a high probability. In Subsection 2.2, on the other hand, we shall discuss how to impose the cardinality constraint strictly if necessary.

2.1 The Bernoulli Style Randomization

Consider the following simple way to make an investment decision: toss a coin and decide whether or not to invest on asset i, i.e.,

$$\eta_i := \begin{cases} x_i, & \text{with probability } \frac{k}{n}, \\ 0, & \text{with probability } 1-\frac{k}{n}, \end{cases}$$

where the vector $x \in \Re^n$ is deterministic vector, which will be determined by an optimization process later. Suppose that η_i's are generated independently, with $i = 1, ..., n$. Then the first moment of η is $\frac{k}{n}x$, and the second moment, the covariance matrix, is

$$\mathrm{Cov}(\eta) = \frac{k(n-k)}{n^2}\,\mathrm{diag}(x^2).$$

Let ξ be the rate of return on these n given securities, with the known mean vector r and covariance matrix Q. For brevity, let us denote $\xi \sim (r, Q)$. We consider a randomized portfolio on these assets as given by η. In fact, the return on the portfolio is given by $\xi^T\eta$, or, equivalently $e^T(\xi \circ \eta)$. Thus, the portfolio problem can be written as

$$\begin{array}{ll} \min & \mathrm{Var}(\xi^T\eta) \\ \text{s.t.} & \mathsf{E}(\xi^T\eta) \geq \mu, \\ & \mathsf{E}(e^T\eta) \leq 1, \\ & \eta_i := \begin{cases} x_i, & \text{with probability } \frac{k}{n} \\ 0, & \text{with probability } 1-\frac{k}{n}, \end{cases} \quad \text{for } i = 1, 2, \cdots, n, \end{array}$$

where the decision variable is x. Since ξ and η are independent, we conclude, using Lemma 1, that

$$\mathsf{E}e^T(\xi \circ \eta) = \frac{k}{n}r^Tx,$$

and

$$\begin{array}{rl} & \mathrm{Var}(e^T(\xi \circ \eta)) \\ = & e^T\mathrm{Cov}(\xi \circ \eta)e \\ = & \mathrm{Cov}(\xi) \bullet \mathrm{Cov}(\eta) + \mathsf{E}(\eta)^T\mathrm{Cov}(\xi)\mathsf{E}(\eta) + \mathsf{E}(\xi)^T\mathrm{Cov}(\eta)\mathsf{E}(\xi) \\ = & \dfrac{k(n-k)}{n^2}\displaystyle\sum_{i=1}^n q_{ii}x_i^2 + \dfrac{k(n-k)}{n^2}r^T\mathrm{diag}(x^2)r + \left(\dfrac{k}{n}\right)^2 x^TQx. \end{array}$$

By a variable transformation, $x := \frac{k}{n}$, the portfolio problem is turned into:

$$\begin{array}{lll} \text{(RP1)} & \min & \frac{n-k}{k} \sum_{i=1}^{n} (q_{ii} + r_i^2) x_i^2 + x^T Q x \\ & \text{s.t.} & r^T x \geq \mu, \\ & & e^T x \leq 1. \end{array} \tag{2}$$

Compared with the original problem (MV_s), (RP1) is now a convex quadratic program, and thus is easy solvable. Theorem 3 below shows that the so-formed portfolio is indeed highly probable to be 'small'. To show this, we first note a technical lemma, involving a useful estimation on binary random variables, which can be found, for instance, in [5].

Lemma 2. *Let $p \in (0,1)$ and*

$$X_i = \begin{cases} 1, & \textit{with probability } p \\ 0, & \textit{with probability } 1-p, \end{cases}$$

are i.i.d. binary random variables, $i = 1, ..., n$. Then, for any $q \in (0,1)$ it holds that

$$\text{Prob}\left\{\sum_{i=1}^{n} X_i > qn\right\} \leq \exp\left(-n \log\left(q \log \frac{p}{q} + (1-q) \log \frac{1-q}{1-p}\right)\right).$$

Theorem 3. *Suppose that $k \leq n/2$. It holds that*

$$\text{Prob}\left\{\|\eta\|_0 \leq k\right\} = \sum_{j=0}^{k} C_n^j \left(\frac{k}{n}\right)^j \left(\frac{n-k}{n}\right)^{n-j} > \frac{1}{2},$$

and for any $\alpha > 0$,

$$\text{Prob}\left\{\|\eta\|_0 > (1+\alpha)k\right\} \leq e^{-k} \left(\frac{e}{1+\alpha}\right)^{(1+\alpha)k}.$$

PROOF. Let us introduce

$$f_{k,n}(x) = \sum_{i=0}^{k} C_n^i (\frac{x}{n})^i (1 - \frac{x}{n})^{n-i}.$$

Clearly, Prob $\{\|\eta\|_0 \leq k\} = f_{k,n}(k)$, and the derivative of $f_{k,n}(x)$ is as follows

$$\begin{aligned} f_{k,n}(x)' &= \sum_{i=1}^{k} C_n^i \left(\frac{i}{n}\right) \left(\frac{x}{n}\right)^{i-1} \left(1 - \frac{x}{n}\right)^{n-i} - \sum_{i=0}^{k} C_n^i \left(\frac{n-i}{n}\right) \left(\frac{x}{n}\right)^i \left(1 - \frac{x}{n}\right)^{n-i-1} \\ &= \sum_{i=0}^{k-1} C_n^{i+1} \left(\frac{i+1}{n}\right) \left(\frac{x}{n}\right)^i \left(1 - \frac{x}{n}\right)^{n-i-1} - \sum_{i=0}^{k} C_n^i \left(\frac{n-i}{n}\right) \left(\frac{x}{n}\right)^i \left(1 - \frac{x}{n}\right)^{n-i-1} \\ &= -C_{n-1}^k \left(\frac{x}{n}\right)^k \left(1 - \frac{x}{n}\right)^{n-k-1} \\ &= -n^{1-n} C_{n-1}^k x^k (n-x)^{n-k-1}, \end{aligned}$$

where we used $C_n^{i+1}(\frac{i+1}{n}) = C_{n-1}^i$, and $C_n^i(\frac{n-i}{n}) = C_{n-1}^i$.

Let $g(x) = x^k(n-x)^{n-k-1}$, and then

$$\begin{aligned} g(x)' &= kx^{k-1}(n-x)^{n-k-1} - (n-k-1)x^k(n-x)^{n-k-2} \\ &= (n-1)x^{k-1}(n-x)^{n-k-2}\left(\frac{kn}{n-1} - x\right). \end{aligned}$$

For $0 < x < n$, $g(x)$ attains its maximum at

$$\hat{x} = \frac{kn}{n-1} = k + \frac{k}{n-1},$$

where $k < \hat{x} < k+1$. We see that $g(x)$ is a increasing for $x \in [k, \hat{x}]$ and decreasing for $x \in [\hat{x}, k+1]$. Moreover,

$$\frac{g(k)}{g(k+1)} = \left(\frac{k}{k+1}\right)^k \left(1 + \frac{1}{n-k-1}\right)^{n-k-1}.$$

Since

$$\left(1 + \frac{1}{n-k-1}\right)^{n-k-1} \geq \left(1 + \frac{1}{k}\right)^k,$$

whenever $n-k-1 \geq k$, or $k \leq (n-1)/2$, therefore, if $k \leq (n-1)/2$ and $x \in [k, k+1]$, then

$$g(x) \geq \min\{g(k), g(k+1)\} = g(k+1) = (k+1)^k(n-k-1)^{n-k-1}.$$

By the mean value theorem, there exists some $\tau \in [k, k+1]$ such that

$$\begin{aligned} f_{k,n}(k) - f_{k,n}(k+1) &= -f_{k,n}(\tau)' \\ &= \frac{1}{n^{n-1}} C_{n-1}^k \tau^k (n-\tau)^{n-k-1} \\ &> \frac{1}{n^{n-1}} C_{n-1}^k (k+1)^k (n-k-1)^{n-k-1} \\ &= C_n^{k+1}\left(\frac{k+1}{n}\right)^{k+1}\left(1 - \frac{k+1}{n}\right)^{n-k-1}. \end{aligned}$$

In general,

$$f_{j,n}(j) > f_{j,n}(j+1) + C_n^{j+1}\left(\frac{j+1}{n}\right)^{j+1}\left(1 - \frac{j+1}{n}\right)^{n-j-1} = f_{j+1,n}(j+1)$$

whenever $j < (n-1)/2$. This means that $f_{k,n}(k) \geq f_{\lceil \frac{n}{2} \rceil, n}(\lceil \frac{n}{2} \rceil) \geq 1/2$. This completes the first part of the theorem.

Now let us consider the probability of large deviation. Let us denote $X_i = \operatorname{sign} \eta_i$; that is,

$$X_i = \begin{cases} 1, & \text{with probability } k/n \\ 0, & \text{with probability } 1 - k/n, \end{cases}$$

where $i = 1, ..., n$. For $0 < t < n/k$, according to Lemma 2, letting $p = k/n$ and $q = tk/n$, it follows that

$$\begin{aligned}
\text{Prob}\left\{\sum_{i=1}^{n} X_i \geq tk\right\} &\leq \exp\left(-tk\log t - n(1-tk/n)\log\frac{n-tk}{n-k}\right) \\
&= t^{-tk}\left(1+\frac{(t-1)k}{n-tk}\right)^{n-tk} \\
&\leq t^{-tk}e^{(t-1)k} = e^{-k}\left(\frac{e}{t}\right)^{tk}.
\end{aligned}$$

Letting $t = 1 + \alpha$ completes the proof. □

2.2 Picking Exact Number of Assets

The Bernoulli style selection may leave some space for fluctuations in terms of the number of assets in the portfolio. This feature may or may not be desirable. Certainly, in some circumstance, one may wish to pick an exact number of securities in the portfolio. To implement this scheme, we observe that it is straightforward to select k out of n assets in a uniform fashion. The procedure is as follows. We start by picking one asset uniformly from n assets. Then, removing this assets and pick up another asset uniformly from the remaining $n-1$ assets. This procedure continues until k assets are picked up. To derive an explicit form for optimization let us denote M to be a 0-1 matrix with n rows, and each column of M consists of exactly k number of 1's, implying that the number of columns in M is C_n^k. Let θ be uniformly selected from the columns of M. Therefore,

$$\mathsf{E}(\theta) = \frac{C_{n-1}^{k-1}}{C_n^k}\, e = \frac{k}{n}\, e,$$

and

$$\text{Cov}(\theta) = \frac{1}{C_n^k}\, MM^T - \left(\frac{k}{n}\right)^2 ee^T.$$

Moreover, we have

$$MM^T = \left(C_{n-1}^{k-1} - C_{n-2}^{k-2}\right) I + C_{n-2}^{k-2} ee^T.$$

Hence, after some calculations we get that

$$\text{Cov}(\theta) = \frac{k(n-k)}{n(n-1)}\left(I - \frac{1}{n}ee^T\right).$$

If we follow this new randomization process, and select a portfolio as $x \circ \theta$, then the return on this portfolio will be $e^T(x \circ \theta \circ \xi)$, or $x^T(\theta \circ \xi)$. According to Lemma 1, we have

$$\begin{aligned}
\text{Cov}(\xi \circ \theta) &= \frac{k(n-k)}{n(n-1)} Q \circ \left(I - \frac{1}{n}ee^T\right) + \left(\frac{k}{n}\right)^2 Q \circ (ee^T) \\
&\quad + \frac{k(n-k)}{n(n-1)}\left(I - \frac{1}{n}ee^T\right) \circ (rr^T).
\end{aligned}$$

After rearrangements, one obtains that the corresponding optimization problem can be formulated as (after replacing $\frac{k}{n}x$ by x):

$$\begin{array}{lll}(\mathrm{RP2}) & \min & \frac{n(n-k)}{k(n-1)}\sum_{i=1}^{n}(q_{ii}+r_i^2)x_i^2+\frac{n(n-k)}{k(n-1)}x^TQx-\frac{n-k}{k(n-1)}(r^Tx)^2 \\ & \text{s.t.} & r^Tx\geq\mu, \\ & & e^Tx\leq 1.\end{array}$$

3 Selecting a 'Clean' Portfolio

By a clean portfolio, we mean a portfolio with a *substantiated* position (long or short) for each asset involved. That is to say, there is a threshold, and the portfolio does not contain any insignificant amount of asset whose value is below this threshold. This is motivated by a practical consideration in investment, since both information and management come with a cost, and so it is only economical to buy or sell, say a financial security, once deemed to do so, with a substantial quantity.

Assume that x is a portfolio with n risky assets and x_f is the proportion invested in riskless asset. We formulate this problem with the above mentioned constraints on the risky assets, which we shall call the threshold constraint from now on. That is, for each risky stock i, $1\leq i\leq n$, we assume $|x_i|$ is either 0, i.e., excluding stock i in the portfolio, or at least a_i (>0); i.e., at least taking a_i position (long or short) in asset i. Below is the mathematical programming formulation for the investment problem with threshold constraints:

$$\begin{array}{lll}(MV_t) & \min & x^TQx \\ & \text{s.t.} & r^Tx+r_fx_f\geq\rho, \\ & & e^Tx+x_f=1, \\ & & |x_i|=\left\{\begin{array}{c}0,\\ \text{or}\\ \geq a_i,\end{array}\right. \quad \text{for } i=1,\cdots,n,\end{array} \tag{3}$$

where Q is the covariance matrix for the risky assets, r_f is the return rate of the riskfree asset, and $a_i>0$, are given parameters, $i=1,...,n$.

The threshold constraints in asset selection are realistic factors for consideration, taking into account the cost of management for the assets, as there are transaction costs, commission fees and other costs. These costs lead us to consider investing in only a few stocks which then should have some sizeable quantity for investment. The threshold requirement is reasonable and practical, which partially limit the total number of stocks we actually invest in. Clearly, the model (MV_t) as represented by (3) is NP-hard from a computational complexity point of view.

The asset selection problem (MV_t) as represented by (3) with threshold constraints can be reformulated in the following equivalent form by scaling:

$$
\begin{array}{lll}
(MV_q) & \min & x^T Q x \\
& \text{s.t.} & x^T Q_i x = \begin{cases} 0, & \\ \text{or} & \text{for } i = 1, \cdots, n, \\ \geq 1, & \end{cases} \\
& & x^T Q_0 x \geq 1,
\end{array}
\tag{4}
$$

where $Q_0 = (r - r_f e)(r - r_f e)^T / (\rho - r_f)^2 \in \mathcal{S}^n$ is a semidefinite matrix, and Q_i is zero everywhere except $Q_i(i,i) = 1/a_i^2$, $i = 1, ..., n$.

3.1 The SDP Relaxation

In this subsection, if a minimization SDP problem has no feasible solution, then we denote its optimal value to be ∞ and its optimal solution as ∞I_n, where I_n is an $n \times n$ identity matrix. Let us consider a general nonconvex quadratic programming problem as following, which is denoted as (QA):

$$
\begin{array}{lll}
(QA) & \min & x^T Q x \\
& \text{s.t.} & x^T Q_i x = 0 \text{ for } i = 1, ..., s, \\
& & x^T Q_i x = \begin{cases} 0, & \\ \text{or} & \text{for } i = s+1, ..., k, \\ \geq 1, & \end{cases} \\
& & x^T Q_i x \geq 1 \text{ for } i = k+1, ..., m,
\end{array}
\tag{5}
$$

where Q is positive definite matrix and Q_i is symmetric positive semidefinite matrix, for $i = 1, ..., m$. Here, s can be assumed as either 0 or 1, otherwise, if we have $s > 1$, then by adding all the Q_is together we can combine all of them to be one zero constraint. And $k < m$, otherwise 0 is the optimal solution, which makes the problem trivial. We also assume that $s < k$. The problem (MV_t) is just a special case of the quadratic problem (QA), and in the subsequent analysis we shall study the more general nonconvex quadratic problem (QA). The NP-hardness of the problem (QA) is immediate as it is a generalization of both (MV_t) and a model in Luo *et al.* [6].

The SDP relaxation for the above problem (QA) is (QB), to be defined as

$$
\begin{array}{lll}
(QB) & \min & Q \bullet X \\
& \text{s.t.} & Q_i \bullet X = 0 \text{ for } i = 1, ..., s, \\
& & Q_i \bullet X = \begin{cases} 0, & \\ \text{or} & \text{for } i = s+1, ..., k, \\ \geq 1, & \end{cases} \\
& & Q_i \bullet X \geq 1 \text{ for } i = k+1, ..., m, \\
& & X \succeq 0,
\end{array}
\tag{6}
$$

where X is a $n \times n$ positive semidefinite matrix. Clearly we know that

$$
v(QB) \leq v(QA).
$$

For each $j \in \{s+1, s+2, ..., k\}$, let us define an SDP problem (QC_j) as following:

$$
\begin{array}{llll}
(QC_j) & \min & Q \bullet X & \\
 & \text{s.t.} & Q_i \bullet X = 0 \text{ for } i = 1, ..., s, & \\
 & & Q_j \bullet X \geq 1, & \qquad (7) \\
 & & Q_i \bullet X \geq 1 \text{ for } i = k+1, ..., m, & \\
 & & X \succeq 0, &
\end{array}
$$

which is an ordinary semidefinite program.

There is a close relationship between problem (QB) and (QC_j), $j \in \{s+1, s+2, ..., k\}$, as the following lemma shows.

Lemma 4. *If X is an optimal solution for (QB), and $Q_j \bullet X \neq 0$, for any one $j \in \{s+1, ..., k\}$, then $v(QC_j) \leq v(QB)$.*

Proof. By the definition of (QB), we know that $Q_j \bullet X \geq 1$ when $Q_j \bullet X \neq 0$, which automatically implies that X a feasible solution for (QC_j). Because X is optimal for (QB), we know that $v(QB) = Q \bullet X$. Thus we have that $v(QC_j) \leq Q \bullet X = v(QB)$. □

This also means that if $v(QC_j) > v(QB)$ then for any optimal solution X of (QB), we have that $Q_j \bullet X = 0$, where j is any one of $\{s+1, ..., k\}$.

3.2 The screening algorithm

We need to find a good feasible solution for the problem (QB) because (QB) is still an NP-hard problem which needs to be tackled first. For this purpose we propose a method to be called a *screening algorithm*, which works as follows.

Screening Algorithm for the SDP relaxation problem (QB)

Step 1 Set $r = 1$.

Step 2 Solve the following SDP problem (QD_r):

$$
\begin{array}{lll}
(QD_r) & \min & Q \bullet X \\
 & \text{s.t.} & Q_i \bullet X = 0 \text{ for } i = 1, ..., s, \\
 & & Q_i \bullet X \geq 1 \text{ for } i = s+1, ..., m, \\
 & & X \succeq 0.
\end{array}
$$

Denote the optimal solution of this problem by X_r and the optimal value by ν_r. If the problem (QD_r) is infeasible, then set $\nu_r = \infty$ and $X_r = \infty I_n$.

Step 3 If $s = k$, set $r_{\max} := r$, then exit to Step 5; otherwise, solve problems (QC_j) for all $s+1 \leq j \leq k$,

$$\begin{array}{lll} (QC_j) & \min & Q \bullet X \\ & \text{s.t.} & Q_i \bullet X = 0 \text{ for } i = 1, ..., s, \\ & & Q_j \bullet X \geq 1, \\ & & Q_i \bullet X \geq 1 \text{ for } i = k+1, ..., m, \\ & & X \succeq 0, \end{array}$$

and denote the optimal value of each problem (QC_j) by t_j^r and the solutions for them by X_j^r. If any one of (QC_j) is unsolvable, then set the optimal value t_j^r to be ∞ and the optimal solution to be ∞I_n. Sort and rename the indices from $s+1$ to k, to ensure that t_j^rs are in nonincreasing order. Suppose that the biggest ones among them are $t_{\max}^r := t_{j+1}^r = \cdots = t_{s'}^r$. If there are more than one t_j^r equal to ∞, count them all equal and all are the maximal items.

Step 4 Set $s = s'$ and $r = r+1$. Problem (QB) is changed to (QB_r) which is:

$$\begin{array}{lll} (QB_r) & \min & Q \bullet X \\ & \text{s.t.} & Q_i \bullet X = 0 \text{ for } i = 1, ..., s, \\ & & Q_i \bullet X = \begin{cases} 0, & \\ \text{or} & \text{for } i = s+1, ..., k, \\ \geq 1, & \end{cases} \\ & & Q_i \bullet X \geq 1 \text{ for } i = s+1, ..., m, \\ & & X \succeq 0. \end{array}$$

Go back to Step 2.

Step 5 Compare all ν_r in the end, and choose the smallest one among them and report the correspondent X_r as the solution. If $X_r = \infty I_n$, then we conclude that the original problem is infeasible.

Theorem 5. *The above screening algorithm has an approximation ratio no more than k-s for (QB).*

Proof. We discuss two cases separately.

(i) If we have $t_{\max}^r \leq v(QB)$ for some iteration r, then $t_j^r \leq v(QB)$ for any $s+1 \leq j \leq k$. Thus by setting $X' = \sum_{j=s+1}^{k} X_j^r$, we have that

$$Q \bullet X' = \sum_{j=s+1}^{k} Q \bullet X_j^r \leq (k-s)\, v(QB),$$

which means that at least we have found one good feasible solution for the SDP relaxation problem (QB). We have that $Q_j \bullet X' \geq Q_j \bullet X_j^r \geq 1$ for any $s+1 \leq j \leq k$. Also it is easy to check that X' satisfies the other constraints of (QD_r) as well. Thus X' is a feasible solution for (QD_r), by the definition of X_r we know that

$$Q \bullet X_r = v(QD_r) \le Q \bullet X' \le (k-s)\, v(QB).$$

(ii) If we have $t^r_{\max} > v(QB)$ for all iteration r, then for any index j which is removed from the set $\{s+1, ..., k\}$ at this iteration, it holds that $t^r_j > v(QB)$. At the first iteration, i.e., $r = 1$, it directly follows from Lemma 4 that $Q_j \bullet X = 0$ as long as X is an optimal solution for (QB), thus $v(QB_1) = v(QB)$. Similarly, for any $r \ge 1$, we know that $Q_j \bullet X = 0$ for any X that is an optimal solution for (QB_{r+1}) by using Lemma 4. Thus

$$v(QB_{r+1}) = v(QB_r) = \cdots = v(QB_1) = v(QB).$$

Thus by induction we have $v(QB_{r_{\max}}) = v(QB)$. This means that $X_{r_{\max}}$ is an optimal solution for (QB) too, i.e., $Q \bullet X_{r_{\max}} = v(QB)$.

As $k - s \le m$, in either case we have that $Q \bullet X_r = v(QD_r) \le m\, v(QB)$, where X_r is a feasible solution for (QB) obtained from Step 5 of the screening algorithm. □

3.3 The worst-case performance ratio

We now propose a rounding algorithm analogous to that in Luo *et al.* [6] for quadratic optimization with homogeneous quadratic constraints. Upon obtaining an approximative solution X for (QB), we construct a feasible solution for (QA) using the following randomized procedure:

Randomized Rounding Algorithm

Step 1 Generate a random vector $\xi \in \Re^n$ from the real-valued normal distribution $N(0, X)$;

Step 2 Let

$$x = \xi / \sqrt{\min\{\xi^T Q_i \xi \mid s+1 \le i \le m, \text{ and } \xi^T Q_i \xi \ne 0\}}. \tag{8}$$

At Step 2 of this randomized procedure, we denote $\Psi := \{i : \xi^T Q_i \xi \ne 0, s+1 \le i \le m\}$. We will use x to analyze the performance of the SDP relaxation. The worst-case performance analysis is similar to the procedure in [6]. For completeness, we will include those key lemmas here.

Lemma 6. *Let $Q \in \mathcal{S}^n$, $X \in \mathcal{S}^n$ be two symmetric positive semidefinite matrices. Assume $\xi \in \Re^n$ and $\xi \sim N(0, X)$. Then, for any $\gamma > 0$,*

$$\text{Prob}\left\{\xi^T Q \xi < \gamma \mathsf{E}(\xi^T Q \xi)\right\} \le \max\left\{\sqrt{\gamma}, \frac{2(\bar{r}-1)\gamma}{\pi - 2}\right\}, \tag{9}$$

where $\bar{r} := \min\{\text{rank }(Q), \text{rank }(X)\}$.

The above lemma is quoted from [6] (Lemma 1). Another useful lemma for our analysis is an adopted form of Lemma 2 in [6].

Lemma 7. *If X is an optimal solution from Step 5 of the Screening Algorithm for the SDP relaxation problem (QB) and x is generated by the randomized rounding algorithm described earlier. Then, with probability 1, x is well defined and feasible for (QA). Moreover, for every $\gamma > 0$ and $\mu > 0$,*

$$\text{Prob}\left\{\min_{i\in\Psi} \xi^T Q_i \xi \geq \gamma,\, \xi^T Q \xi \leq \mu Q \bullet X\right\} \geq 1 - m\max\{\sqrt{\gamma}, \frac{2(\hat{r}-1)\gamma}{\pi-2}\} - \frac{1}{\mu}, \tag{10}$$

where $\hat{r} :=$ rank (X).

Proof. By re-naming the indices, we can assume that for an index s', which satisfies $s+1 \leq s' \leq k$, we have

$$\begin{aligned} &Q_i \bullet X = 0 \text{ for } i = 1, ..., s', \\ &Q_i \bullet X \geq 1 \text{ for } i = k+1, ..., m. \end{aligned}$$

Then $\mathsf{E}(\xi^T Q_i \xi) = Q_i \bullet X = 0$ for $1 \leq i \leq s'$, which means $\xi^T Q_i \xi = 0$ for all $1 \leq i \leq s'$, thus it is the same for $x^T Q_i x$. Moreover, $x^T Q_i x \geq 1$ for $s' \leq i \leq m$ follows from the definition of x (see (8)). The feasibility of x is easily verified.

$$\begin{aligned} & \text{Prob}\,\{\min_{i\in\Psi} \xi^T Q_i \xi \geq \gamma, \xi^T Q \xi \leq \mu Q \bullet X\} \\ = \;& \text{Prob}\,\{\xi^T Q_i \xi \geq \gamma, \forall i \in \Psi,\ \text{and}\ \xi^T Q \xi \leq \mu Q \cdot X\} \\ \geq \;& \text{Prob}\,\{\xi^T Q_i \xi \geq \gamma Q_i \bullet X, \forall i \in \Psi,\ \text{and}\ \xi^T Q \xi \leq \mu Q \bullet X\} \\ = \;& \text{Prob}\,\{\xi^T Q_i \xi \geq \gamma \mathsf{E}(\xi^T Q_i \xi), \forall i \in \Psi,\ \text{and}\ \xi^T Q \xi \leq \mu \mathsf{E}(\xi^T Q \xi)\} \\ = \;& 1 - \text{Prob}\,\{\xi^T Q_i \xi < \gamma \mathsf{E}(\xi^T Q_i \xi) \text{ for some } i \in \Psi \text{ or } \xi^T Q \xi > \mu \mathsf{E}(\xi^T Q \xi)\} \\ \geq \;& 1 - \sum_{i\in\Psi} \text{Prob}\,\{\xi^T Q_i \xi < \gamma \mathsf{E}(\xi^T Q_i \xi)\} - \text{Prob}\,\{\xi^T Q \xi > \mu \mathsf{E}(\xi^T Q \xi)\} \\ > \;& 1 - m\max\left\{\sqrt{\gamma}, \frac{2(\hat{r}-1)\gamma}{\pi-2}\right\} - \frac{1}{\mu}, \end{aligned}$$

where in the last step we used Lemma 6, and also Markov's inequality. ☐

We now use these lemmas to bound the performance of the SDP relaxation.

Theorem 8. *The screening algorithm and the randomized rounding algorithm provide an $O(m^3)$ approximation with probability of at least 7.5%.*

Proof. By applying a suitable rank reduction procedure if necessary, we can assume that the rank $\hat{r}$ of the optimal SDP solution X satisfies $\hat{r}(\hat{r}+1)/2 \leq m$; see e.g. [9]. Thus $\hat{r} < \sqrt{2m}$. If $m \leq 2$, then $\hat{r} = 1$, implying that $X = x^*(x^*)^T$ for some $x^* \in \Re^n$ and it is readily seen that x^* is an optimal solution of (QA). Otherwise, we apply the randomized procedure to X. We also choose

$$\mu = 3,\ \gamma = \frac{\pi}{4m^2}\left(1 - \frac{1}{\mu}\right)^2 = \frac{\pi}{9m^2}.$$

Then, it is easily verified using $\hat{r} < \sqrt{2m}$ that

$$\sqrt{\gamma} \geq \frac{2(\hat{r}-1)\gamma}{\pi - 2}, \forall m = 1, 2, \ldots$$

Plugging these choices of γ and μ into (10), we see that there is a positive probability of at least

$$1 - m\sqrt{\gamma} - \frac{1}{\mu} = 1 - \frac{\sqrt{\pi}}{3} - \frac{1}{3} = 0.0758\ldots$$

and ξ generated by the randomized procedure satisfies

$$\min_{i \in \Psi} \xi^T Q_i \xi \geq \frac{\pi}{9m^2}, \text{ and } \xi^T Q \xi \leq 3(Q \bullet X).$$

Let ξ be any vector satisfying these two conditions. Then x is a feasible solution for (QA), so that

$$x^T Q x = \frac{\xi^T Q \xi}{\min_{i \in \Psi} \xi^T Q_i \xi} \leq \frac{3(Q \bullet X)}{\pi / 9m^2} \leq \frac{27m^3}{\pi} v(QB),$$

where the last inequality uses $Q \bullet X = v(QD_r) \leq m\, v(QB)$. □

4 Numerical Experiments

It remains to test the effects of the randomization methods as we have introduced in the previous sections. Our numerical experiments are organized as follows. Subsections 4.1 and 4.2 are devoted to the numerical performance of $(RP1)$ and $(RP2)$ for the small portfolio selection problem, while Subsection 4.3 is devoted to the clean portfolio problem. Subsection 4.4 discusses an extension of the clean portfolio model, which is interesting as an optimization model on its own. For the small portfolio selection problem, the study boils down to comparing the optimal values of (MV_s), by means of either branch and bound or other implicit enumeration methods, and the optimal values of $(RP1)$ and $(RP2)$. A further division of the study distinguishes whether or not a risk-free asset is considered, and we shall discuss in Subsections 4.1 and 4.2 separately.

4.1 Risky assets only

In this subsection, we consider the portfolio selection problem with risky assets only and the equality constraints are considered. The problem we used for our computational experiments are randomly generated, i.e., the mean return rate of each individual stock and the covariance matrix of stock are randomly generated. Table 1 and Figure 1 present the comparison of the optimal value among problems (MV_s), $(RP1)$ and $(RP2)$ for different cardinalities, where we assume that there are in total 10 stocks for investment. In Table 1, the first column reports the number of stocks we selected in our portfolio, i.e., choosing 2 stocks to 10 stocks from 10 stocks, and the following three columns report the minimal variance (optimum) for different models, i.e., (MV_s), $(RP1)$ and $(RP2)$, respectively.

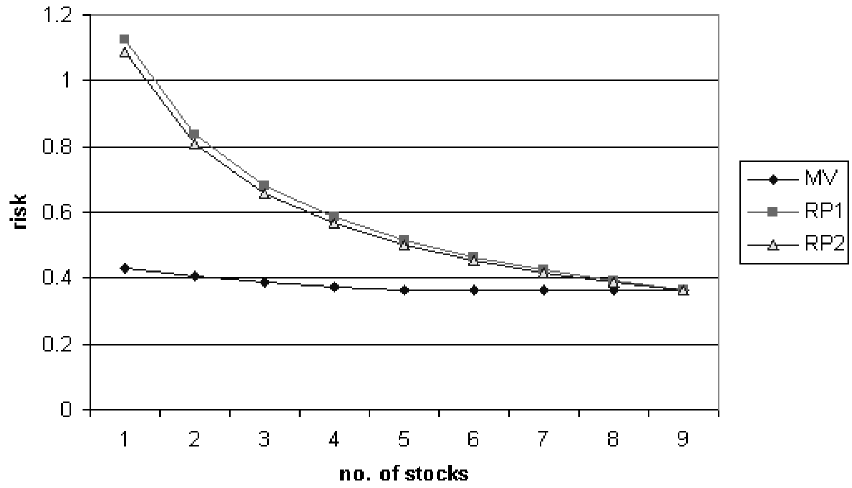

Figure 1: Cardinality vs. Variance (return = 5%)

Our first observation is that the optimal values of problem $(RP1)$ and $(RP2)$ are almost the same, and they are naturally larger than the optimal value of problem (MV_s). Another interesting phenomenon is that the optimal values of problems $(RP1)$ and $(RP2)$ are decreasing and converging to the optimal value of (MV_s) while the desired number of stocks selected grows from 2 stocks to the entire 10 stocks, which is shown in Figure 1.

k	MV	RP1	RP2
2	0.4292	1.1259	1.0882
3	0.4073	0.8339	0.8062
4	0.3861	0.6808	0.6587
5	0.3731	0.5842	0.5661
6	0.3656	0.5164	0.5016
7	0.3625	0.4653	0.4535
8	0.3621	0.4248	0.416
9	0.362	0.3911	0.3859
10	0.3619	0.3619	0.3619

Table 1: Results on Cardinality vs. Variance.

Table 2 reports the frontier for problems (MV_s), $(RP1)$ and $(RP2)$ respectively, and we select 2 stocks from 10 alternative stocks. The first column presents the desired expected return rate from 0.5% to 10% and the the following three columns report the minimal variance (optimum) for different models, i.e., (MV_s), $(RP1)$ and $(RP2)$ with respect to each desired expected return rate. Figure 2 depicts that the frontiers of both problems $(RP1)$ and $(RP2)$ are very close and higher than that of problem (MV_s). The optimal values of $(RP1)$ and $(RP2)$ are always within three times of the optimal value of (MV_s) for all these desired return rates.

u	MV	RP1	RP2
0.005	0.9338	2.6239	2.8439
0.01	0.8419	2.2238	2.3849
0.015	0.677	1.882	1.9918
0.02	0.5567	1.5987	1.6647
0.025	0.4811	1.3738	1.4036
0.03	0.4502	1.2074	1.2085
0.035	0.4639	1.0994	1.0794
0.04	0.5222	1.0498	1.0163
0.045	0.4751	1.0586	1.0193
0.05	0.4292	1.1259	1.0882
0.055	0.4637	1.2516	1.2231
0.06	0.5984	1.4357	1.424
0.065	0.8334	1.6783	1.6908
0.07	1.1591	1.9793	2.0237
0.075	1.3149	2.3387	2.4226
0.08	1.3705	2.7565	2.8875
0.085	1.4535	3.2328	3.4184
0.09	1.5602	3.7675	4.0153
0.095	1.6925	4.3606	4.6781
0.1	1.8503	5.0122	5.407

Table 2: Results on efficient frontier.

Figure 3 presents the frontier of the stock selection problem (MV_s) which has 4 frontiers, such as $MV(2)$, $MV(3)$, $MV(4)$ and $MV(5)$, where $MV(2)$ denotes the frontier for only choosing 2 stocks from the whole 10 stocks to form the optimal portfolio. We can find that the more stocks we chose to form the portfolio, the less risk the portfolio in the frontier. For problems ($RP1$) and ($RP2$), we also have this result that the more stocks selected in portfolio, the less risk incurred in the optimal portfolio in the frontier, refer to Figure 4 and Figure 5. For these figures, we also observe that the more stocks selected into our portfolio, the smoother the curve in the frontier.

Finally we present Figure 6, which shows that the efficient frontiers for those models with inequality constraints also have similar properties as the models with equality constraints, which we tested and discussed earlier. For the problems with inequality constraints, the frontiers naturally lie below the corresponding ones with equality constraints, since the additional flexibility helps to reduce the risk.

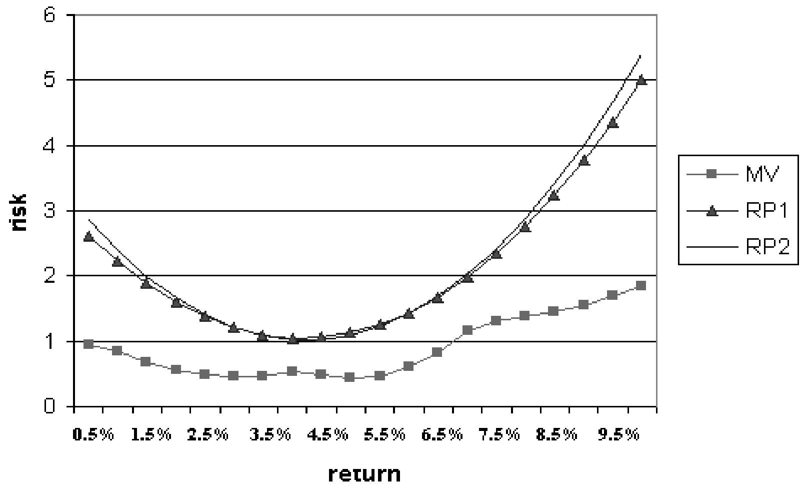

Figure 2: Efficient Frontier (k=2, n=10)

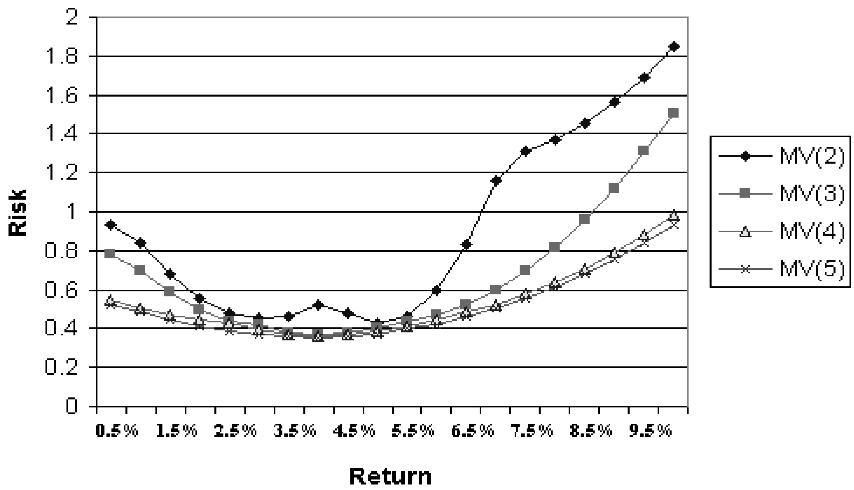

Figure 3: Frontier of (MV_s)

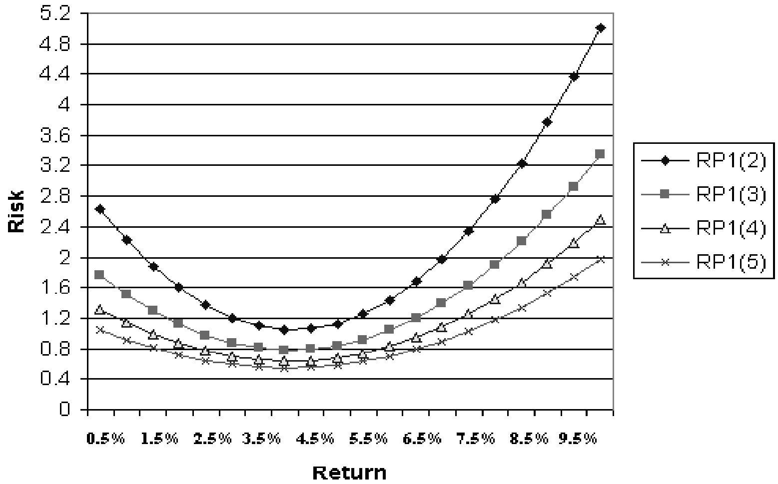

Figure 4: Frontier of (RP1)

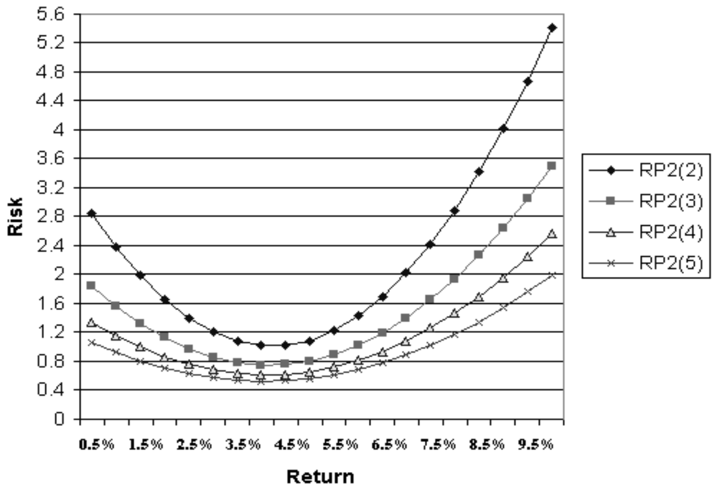

Figure 5: Frontier of (RP2)

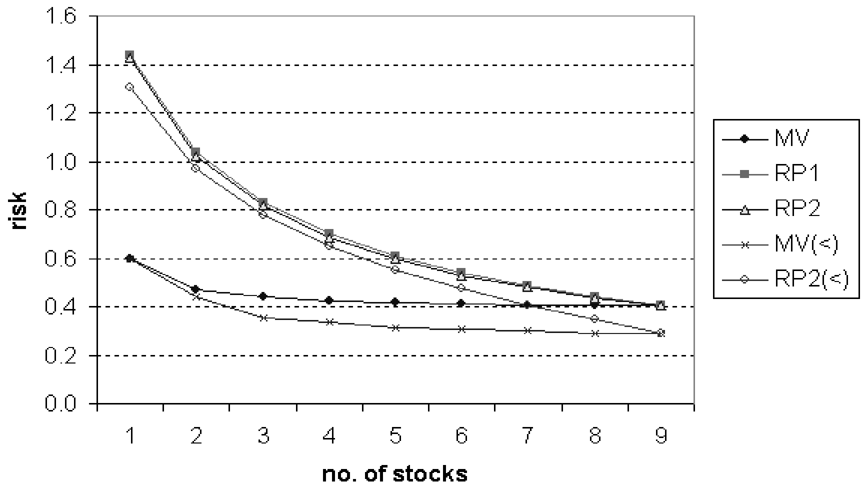

Figure 6: Frontier with equality and inequality constraints (return = 6%).

4.2 Riskfree asset included

In this subsection, we present the computational results on the mean-variance model (MV_s) and the two randomized procedures, ($RP1$) and ($RP2$), with riskfree asset included. Figure 7 shows that the efficient frontier (with the riskfree asset included) is similar to the one without the riskless asset, except that the minimal variance portfolio with solely the risk free asset is zero. We experiment

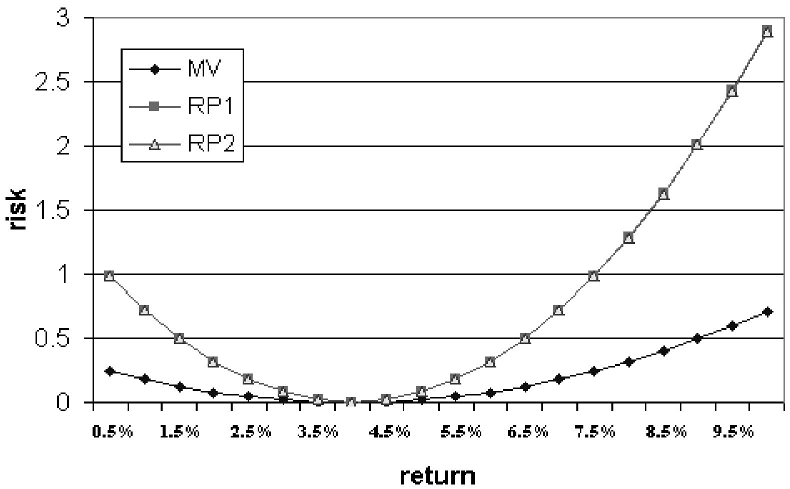

Figure 7: Frontier with riskfree asset.

with more randomized portfolio selection instances with different cardinality constraints and/or the varying risk free return rates. We observe that the efficient frontier moves up if the risk free rate is higher which is easy to explain as the higher return rate of riskless asset reduce the total variance for the corresponding portfolios exposed at the frontier. Clearly, if more assets are allowed then the less risk (though higher management cost) there will be in the optimal portfolio.

4.3 The threshold model

In the following, we mainly consider some numerical experiments on the clean portfolio selection problem, which is (see (MV_t) as represented in (3)):

$$
\begin{array}{ll}
\min & x^T Q x \\
\text{s.t.} & x^T Q_i x = \left\{ \begin{array}{c} 0, \\ \text{or} \\ \geq 1, \end{array} \right. \qquad \text{for } i = 1, ..., n, \\
& x^T Q_0 x \geq 1, \\
& x \in \Re^n,
\end{array}
\tag{11}
$$

where $Q_0 = (r - r_f e)(r - r_f e)^T / (\rho - r_f)^2 \in \mathcal{S}^n$ is a positive semidefinite matrix and Q_i is a matrix that is zero everywhere except for $Q_i(i,i) = 1/a_i^2$, $i = 1, ..., n$. We solve this problem in the following procedure:

Algorithm:

Step 1 Finding a feasible solution for SDP relaxation problem (QB) by running the Screening Algorithm described in Section 3.2;

Step 2 A feasible solution is generated for the nonconvex quadratic programming problem (QA) by the Randomized Rounding Algorithm in Section 3.3, which provides an upper bound for the problem (QA);

Step 3 Running a branch-and-bound algorithm with the upper bound given in Step 2 to find the optimal solution for (QA).

Table 3 presents some randomly generated instances for the model (MV_t). All these instances are 10 dimensional problems with 11 constraints. The first row lists the instances names. From column 2 to 7, We summarize some numerical results for 6 randomly generated examples as follows:

a. The second row 'OPT' reports the optimal value of the problem (MV_t) solved by the branch-and-bound algorithm.

b. The third row 'Round' shows the objective value of each instance obtained by the randomized rounding procedure, i.e., generating 1000 normal random variable with the covariance matrix being the SDP solution which is solved by the screening procedure in Section 3.2.

c. The 4th and 5th row, i.e., 'Upper' and 'Lower', are the upper bound and lower bound of the SDP relaxation problem (QB).

d. The 6th row 'Bound' reports the ratio of the objective value by random rounding to the lower bound of the SDP relaxation problem (QB), which are all better than the theoretical ratio, i.e., O(1000), where the best one is 3.77 times the lower bound and the worst one is 70.43 times the lower bound.

e. The 7th row 'Visit' reports the visit ratio that we totally visit in our searching tree, we only visit very small number of nodes in the branching tree to find the optimal solution, in these instances, the worst one is 0.3% of the total nodes and the best one, i.e., instances 'q10a' and 'q10e', only visits 0.03% of all the nodes in the branching tree.

f. The 8th row 'Cut' lists how many nodes we have cut in view of the total nodes visited, which we call it cut ratio. In these test examples, the cut ratios are mostly around 60%.

g. The 9th row 'Leaf' shows how many leaf nodes we have visited in the searching tree. The leaf ratio is around 7% to 10%.

h. The 10th row 'SDP' reports the computing time in seconds to run the screening algorithm.

i. The 11th row 'BnB' reports the computing time in seconds using branch-and-bound algorithm with the upper bound given by the random rounding procedure in which the covariance matrix is provided by the screening algorithm. The time to finish the branch-and-bound are around 1 to 4 seconds.

j. The 12th row 'BnB(na)' reports the computing time in seconds using branch-and-bound algorithm if we randomly select a large upper bound. In our case, we set the upper bound to be 10^{10} at the beginning. This is a naive branch-and-bound algorithm which needs more time to finish searching, i.e., almost 3-4 times slower than our branch-and-bound method 'BnB'.

	q10a	q10b	q10c	q10d	q10e	q10f
OPT	6.4	15.1	18.4	24.0	0.092	4.7
Round	6.5	22.0	19.9	29.8	0.092	15.7
Upper	27.6	33.7	33.9	110.8	6.650	34.0
Lower	1.4	9.6	11.7	19.3	0.002	0.7
Bound	4.82	2.29	1.70	1.54	45.30	23.87
Visit	0.03%	0.17%	0.22%	0.09%	0.03%	0.30%
Cut	60.0%	59.2%	56.6%	60.7%	60.0%	59.3%
Leaf	10.0%	8.2%	10.6%	7.1%	10.0%	7.8%
SDP	44.29	45.84	47.16	43.52	42.98	47.61
BnB	1.29	2.38	3.11	1.83	1.30	3.96
BnB(na)	7.98	5.72	10.46	6.43	9.89	7.28

Table 3: Results for Branch-and-Bound (QP, 10-dim).

	q50a1	q50a2	q50b1	q50b2
OPT	0.105	0.105	0.033	0.033
Round	0.430	1.249	0.071	0.779
Bound1	4.1	11.9	2.2	23.6
Bound2	29.76	86.445	4.9115	53.9
Visit	9.70E-20	1.00E-19	3.40E-20	4.60E-20
Cut	66.56%	66.49%	66.52%	65.85%
Leaf	0.11%	0.18%	0.15%	0.82%
SDP	3.46	0.82	3.72	0.96
BnB	1.50	1.59	0.54	0.99
Total	4.96	2.42	4.26	1.95
BnB(na)	3.90	-	2.68	-

Table 4: Results for Branch-and-Bound (QP, 50-dim).

In Table 4, we test some examples randomly generated with 50 dimensions. We test the same one problem in the 2nd and 3rd column with different accuracy for SeDuMi. Similarly, we test another problem in the 4th and 5th column with different accuracy for SeDuMi. For ‘q50a1’ and ‘q50b1’, we use the default accuracy to compute the SDP solution in SeDuMi. For ‘q50a2’ and ‘q50b2’, we use a relaxed accuracy, i.e., we set pars.eps=0.1 in SeDuMi, to compute the SDP solution. ‘Bound1’ is the bound ratio of the rounded value to the optimal value, and ‘Bound2’ is the bound ratio of the rounded value to the lower bound value of the SDP relaxation problem. All visit ratios are in the 10^{-20} magnitude. All cut ratios are around 66%. No Leaf ratio is greater than 1% in all these cases. ‘Total’ is the total computation time to run the screening procedure and the branch-and-bound algorithm. We observe that the total time is greater than that of ‘BnB(na)’ in the default accuracy case, as in ‘q50a1’, ‘Total’ needs 4.96 hours but ‘BnB(na)’ needs only 3.90. But if we relax the accuracy a bit more, such as to 0.1 in SeDuMi, then the time to compute the screening procedure (mainly solve SDP problem using SeDuMi) is decreased by a large amount, from 3.46 hours to 0.82 hours in ‘q50a1’ and ‘q50a2’. In other words, we have saved the computational time by about 75%, although the time to complete ‘BnB’ increases slightly. If we set the accuracy parameter to be 0.1 in SeDuMi, we only need 2.42 hours in ‘Total’ but we need 3.9 hours to complete ‘BnB(na)’ in ‘q50a’, i.e., we saved about 40%. For ‘q50b’, we have the same observation that the total time decreases from 2.68 hours to 1.95 hours, i.e., approximate 30% saving in computational time.

In Tables 5 and 6, we test some instances from stock markets around the world, such as, Hong Kong, Shanghai, Shenzhen, Singapore and NASDAQ stock market. In Table 5, ‘HSI33’ includes 33 component stocks of the Hang Seng Index, ‘SH49’ includes 49 stocks from the Shanghai 50 Index, ‘SGIT29a’ and ‘SGIT29b’ include 29 stocks from the IT stocks listed in Singapore stock market. In Table 6, ‘SZB55’ includes 55 stocks of the Shenzhen B share, ‘NASDAQ30a’, ‘NASDAQ30b’ and ‘NASDAQ38c’ include 30, 30 and 38 different stocks from the NASDAQ stock market respectively.

	HSI33	SH49	SGIT29a	SGIT29b
OPT	0.719	1.896	0.277	0.2767
Round	0.721	1.986	20.143	0.2773
Upper	0.836	63.831	3.966	3.966
Lower	0.702	0.218	0.239	0.239
Bound	1.03	9.09	84.25	1.16
Visit	6.03E-13	1.62E-19	1.13E-09	1.52E-11
Cut	66.45%	66.60%	44.21%	65.39%
Leaf	0.24%	0.07%	22.46%	1.34%
SDP	0.69	3.20	0.46	0.39
BnB	0.04	0.39	0.93	0.01
BnB(na)	0.43	43.04	0.92	0.93

Table 5: Results for asset selection.

	SZB55	NASDAQ30a	NASDAQ30b	NASDAQ38c
OPT	0.8741	1.623	1.762	3.792
Round	0.8742	1.623	6.128	3.853
Upper	76.92	47.94	50.66	59.87
Lower	0.311	0.360	0.348	1.651
Bound	2.81	4.51	17.62	2.33
Visit	2.00E-22	2.18E-12	2.65E-10	1.09E-15
Cut	66.23%	66.37%	65.85%	66.30%
Leaf	0.44%	0.45%	0.82%	0.41%
SDP	6.61	0.33	0.36	0.91
BnB	0.33	9.57(sec)	0.39	0.01
BnB(na)	114.99*	0.51	0.78	10.13

Table 6: Results for asset selection (cont').

We observe from Tables 5 and 6 that the objective value given by the random rounding procedure is very close to the optimal value except for two instances, i.e, 'SGIT29a' and 'NASDAQ30b'. For 'NASDAQ30a', the rounding result is even equal to the optimal value. For these data, we find that the randomized rounding procedure does provide a good feasible solution. The bound is less than 10 times in most cases. The visit ratios are very small and the cut ratios are around 66% except for one instance, i.e., 'SGIT29a'. The Leaf ratio has the same properties as the cut ratio, all ratios are lower than or around 1% except for 'SGIT29a'. In these two tables, 'SDP', 'BnB' and 'BnB(na)' report the time in hours. For all of these examples, the computation time of the naive branch-and-bound method listed in 'BnB(na)' is longer than that of our branch-and-bound algorithm listed in 'BnB', except for 'SGIT29a'. The reason is the rounding result is not so good for 'SGIT29a'. For 'NASDAQ30b', the visit ratio of the naive branch-and-bound method increases by 53.58% with respect to the visit ratio

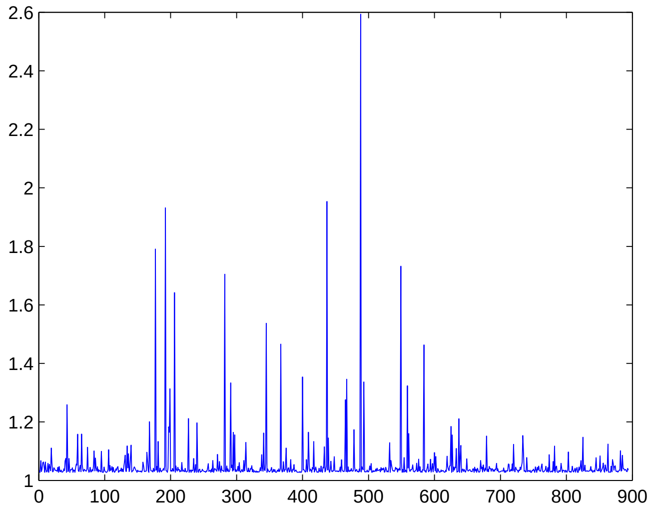

Figure 8: upper bound on v(qp)/v(sdp) for n=33, m=34, 1000 realizations-HSI.

of our branch-and-bound algorithm. For 'SZB55', we run a naive branch-and-bound algorithm for 114.99 hours, after which 99% of the total nodes are still not explored yet.

For 'SGIT29a' and 'SGIT29b', we observe that our branch-and-bound algorithm runs fast if the random rounding procedure provides a good feasible solution. In 'SGIT29a', the rounding procedure gives a solution with bound 84.25, but in 'SGIT29b', we get a rounding solution with bound 1.16. Thus, the computational time for 'SGIT29a' is almost one hour, but the computational time for 'SGIT29b' is less than one minute. The cut ratio increases by 50%, from 44.21% to 65.39%. For 'SGIT29a', the total computational time of 'SDP' and 'BnB' is greater than that of 'BnB(na)'. But for 'SGIT29b', the total computation time of 'SDP' and 'BnB' is less than half of the computation time of 'BnB(na)'.

Figure 8 shows the randomized rounding results for problem 'HSI33' which is relaxed to an SDP problem and solved efficiently by SeDuMi [10]. Then we use the random rounding procedure to generate some feasible solutions for the quadratic program problem 'HSI33', for example, we generate 1000 feasible solutions for 'HSI33'. Figure 9 shows that 97.65% of all the realizations has the bound lower than 1.2.

4.4 The discrete SDP problem

In this subsection, we shall present some experimental results about the SDP problem (QB) as defined in (6). We also randomly generate some 10-dimensional instances. The first 2 instances in Table 7 show that the optimal solution by branch-and-bound is the same as the one obtained by

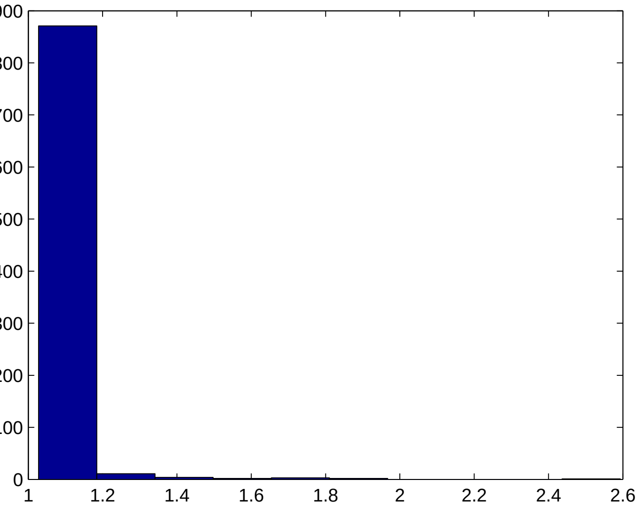

Figure 9: Histogram of the outcomes in Figure 8.

the SDP relaxation screening procedure. For the last 4 instances, the branch-and-bound algorithm finds the optimal solutions of these SDP problems quickly. In Figure 10, we observe as the accuracy

Table 7: Results for Branch-and-bound (SDP,10-dim).

	s10a	s10b	s10c	s10d	s10e	s10f
OPT	6.31	5.88	2.63	23.33	15.83	1.39
SDP-relax	6.31	5.88	3.68	25.16	16.33	2.33
Upper	8.25	8.08	18.04	56.35	21.69	1.64
Lower	0.33	0.84	0.14	2.4	0.23	0.11
Visit	6.16%	3.62%	2.35%	3.62%	5.08%	1.76%
Cut	47.62%	48.65%	47.92%	43.24%	47.12%	41.67%
Leaf	3.17%	2.70%	4.17%	8.11%	3.85%	11.11%
SDP	16.9	19.5	16.5	25.6	16.6	17.4
BnB	49.8	32.4	14.6	34.3	35.5	8.7

requirement relaxes in the SDP subroutines the computational time for screening procedure decreases drastically. However, the time to complete the entire branch-and-bound algorithm increases as a result of a worsened (inaccurate) SDP bounds.

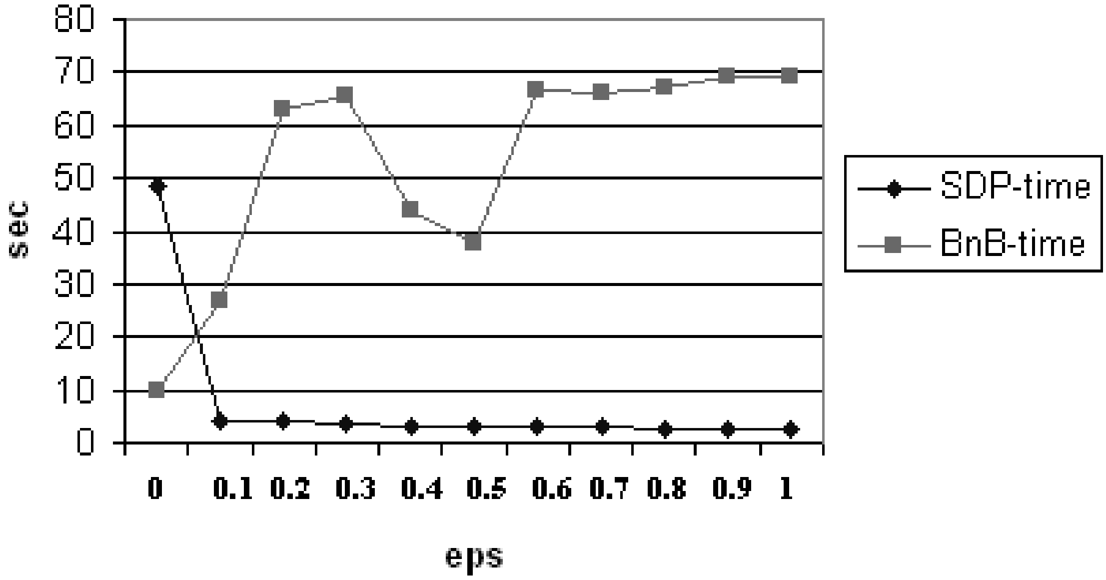

Figure 10: Accuracy in SeDuMi.

5 Conclusions

In this paper, we examine the asset selection problem with various constraints. First we considered the mean variance model with the constraint on the L_0 norm of the portfolio. Such portfolio is termed *small* in our discussion. For that model we introduced two different types of randomization procedures to deal with the combinatorial complexity. Numerical results show that these procedures are effective indeed. In particular, we conducted computational experiments on the comparisons between the two randomization models, $(RP1)$ and $(RP2)$, and compared them with (MV_s), revealing the tradeoffs (the risk vs. the return) that they offered in the context of efficient frontiers. Then we considered the mean-variance model with another kind of combinatorial complication, however practically plausible, which posed a minimum level of involvement once it is involved. Such portfolio is termed *clean* for ease of reference. This problem is relaxed to a semidefinite program and then we applied a screening procedure in Section 3.2 to find a good feasible solution which is bounded by $O(m)$ of the optimal value of the relaxation SDP problem. Similar to Luo *et al.* [6], we randomly generate enough solutions for (MV_t), which will give us a feasible solution with a worst-case performance bound of $O(m^3)$. Furthermore, we conducted extensive numerical experiments on problem (MV_t) and (QA). We show that we only need to visit a small percentage of nodes in the branching tree in order to find the optimal solution, owing to the good feasible solution generated for the quadratic program (MV_t) and the good feasible solution for the (discrete) SDP relaxation problem (QA) by the screening procedure.

References

[1] D. Bertsimas and R. Shioda. An Algorithm for Cardinality Constrained Quadratic optimization Problem. Working Paper, 2004.

[2] D. Bienstock. Computational Study on Families of Mixed-Integer Quadratic Programming Problems, *Mathematical Programming*, Vol. 74, pp. 121 – 140, 1996.

[3] B. Blog, G. van der Hoek, A.H.G. Rinnooy Kan, and G.T. Timmer. The Optimal Selection of Small Portfolios, *Management Science*, Vol. 29(7), pp. 792 – 798, 1983.

[4] D. Goldfarb and G. Iyengar. Robust portfolio selection problems, *Mathematics of Operations Research*, Vol. 28, pp. 1 – 38, 2003.

[5] H.W. Jensen. Chernoff Bounds. Lecture Notes:
`http://seed.ucsd.edu/~cse190/media/chernoff.pdf`, 2006.

[6] Z.Q. Luo, N.D. Sidiropoulos, P. Tseng, and S. Zhang. Approximation Bounds for Quadratic Optimization with Homogeneous Quadratic Constraints. To appear in *SIAM Journal on Optimization*

[7] H.M. Markowitz. Portfolio Selection, *Journal of Finance*, Vol. 7(1), 77–91, 1952.

[8] H.M. Markowitz. *Portfolio Selection*. J. Wiley & Sons Inc., New York, 1959.

[9] G. Pataki. On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues, *Math. Oper. Res.*, Vol. 23 pp. 339 – 358, 1998.

[10] J.F. Sturm, Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones, *Optimization Methods and Software*, Vol. 11 – 12, pp. 625-653, 1999. Special issue on Interior Point Methods (CD supplement with software).

[11] X.Y. Zhou and D. Li. Continuous-Time Mean-Variance Portfolio Selection: A Stochastic LQ Framework, *Applied Mathematics Optimization*, Vol. 42, pp. 19 – 33, 2000.