Stochastic Portfolio Optimization with Proportional Transaction Costs: Convex Reformulations and Computational Experiments

Tiago P. Filomena
School of Business, Federal University of Rio Grande do Sul

Miguel A. Lejeune
Department of Decision Sciences
The George Washington University
Stochastic Portfolio Optimization with Proportional Transaction Costs: Convex Reformulations and Computational Experiments

Tiago P. Filomena  
School of Business, Federal University of Rio Grande do Sul  
855 Washington Luiz, Porto Alegre, Brazil  
Phone: 55-51-3308-3536 Fax: 55-51-3308-3991; Email: tpfilomena@ea.ufrgs.br

Miguel A. Lejeune  
Dept. of Decision Sciences, George Washington University  
2201 G Street, Suite 415, NW, Washington, DC, USA  
Phone: 1-202-994-6576; Fax: 1-202-994-2736; Email: mlejeune@gwu.edu

Abstract  
We propose a probabilistic version of the Markowitz portfolio problem with proportional transaction costs. We derive equivalent convex reformulations, and analyze their computational efficiency for solving large (up to 2000 securities) portfolio problems. There is a great disparity in the solution times. The time differential between formulations can reach several orders of magnitude for the largest instances. The second-order cone formulation in which the number of quadratic terms is invariant to the number of assets is the most efficient.

Keywords: stochastic portfolio optimization, probabilistic Markowitz, transaction costs, stochastic programming, estimation risk.

1 Introduction

Modern portfolio theory started with the mean-variance asset allocation model proposed by Markowitz [23] five decades ago. The model takes the form of a quadratic optimization problem and uses the variance of the portfolio as a risk measure representing the volatility of the market. Optimal portfolios are those offering the best trade-off between the expected return and the variance of the portfolio and are located on the mean-variance efficient frontier. A critic frequently voiced against the mean-variance model is that it considers the asset returns as deterministic parameters perfectly represented by a single point estimate, i.e., their mean, which leads to the so-called estimation risk [3]. Indeed, a number of empirical studies (see, e.g., [6, 24]) show that the optimal mean-variance portfolio is extremely sensitive to the estimation of the returns and that slight deviations of the return from its estimated mean value could lead to the construction of substantially different portfolios. Minor adjustments of the estimated parameters can lead to a major reshaping of the portfolio and to a large turnover ratio. Investors would rather trade off some return for more safety and construct a portfolio that performs well under a wide set of circumstances. With the objective of developing allocation strategy that are less subjected to the dangers of the estimation risk, models based on robust optimization [12, 32], stochastic dominance [10] and programming [4, 29, 31], Bayesian probability [28], or robust statistics [33], have been proposed.

In this paper, we use the probabilistic version of the Markowitz model proposed by Bonami and Lejeune [4] in which the incomplete knowledge of the return behavior is taken into account by modeling the asset returns as random variables. Transaction costs induced, for instance, by brokerage fees, liquidity costs, fund loans and tax [11, 18, 21] are incorporated in the model, and modeled as proportional to the amount traded [15, 21, 26].
Proportional transaction costs are often used to represent the gap between bid and ask prices. We refer the reader to [22] for a comprehensive overview of functional forms used for transaction costs.

The following notations will be used. We denote by \( n \) the total number of assets, \( R \) is the required return level, and \( w^0 \) is the portion of capital initially allocated to asset \( j, j = 1, \ldots, n \). Let \( w, y \) and \( z \) be \( n \)-dimensional vectors of decision variables: \( w_j \) is the portion of capital invested in asset \( j \) after the rebalancing of the initial portfolio, \( y_j \) is the portion of capital used to purchase asset \( j \) and \( z_j (z_j \leq w^0_j) \) is the portion of capital obtained by selling shares of asset \( j \). The cost rates incurred when purchasing and selling an asset are respectively \( c \) and \( d \). The asset returns are defined as stochastic variables: \( \xi \) is the \( n \)-dimensional vector of random asset returns. Finally, \( p \) is the specified probability level and \( P \) is a probability measure. We assess the volatility of the market with historical data of asset returns. We use the recordings of the past observed returns \( r_{jt} \) of each asset \( j \) at \( M(\ t = 1, \ldots, M) \) consecutive time-periods to derive empirical estimates of the mean return vector \( \mu \) and the matrix of variance-covariance \( \Sigma \). Note that this could be alternatively done by using factor models [8] or scenarios [14, 25]. The reader is referred to Chan et al. [7] for a description of the strengths and weaknesses of the methods generally employed to construct variance-covariance matrices.

The probabilistic Markowitz model with proportional transaction costs is the stochastic programming problem SPM

\[
\text{SPM : } \min \ w^T \Sigma w
\]

subject to \( P \left( \sum_{j=1}^{n} (\xi_j w_j - c y_j - d z_j) \geq R \right) \geq p \) \hspace{1cm} (1)

\[
\sum_{j=1}^{n} w_j + c \sum_{j=1}^{n} y_j + d \sum_{j=1}^{n} z_j = 1 \hspace{1cm} (2)
\]

\[
w_j = w^0_j + y_j - z_j, \ \forall j \hspace{1cm} (3)
\]

\[
w, y, z \in \mathbb{R}^+_n. \hspace{1cm} (4)
\]

Problem SPM minimizes the variance of the portfolio (1) and is subjected to a set of linear constraints (3)-(5) and a chance constraint (2). Constraint (3) is the budget constraint: the available capital is used in its entirety to cover the transaction costs \( \sum_{j=1}^{n} (c y_j + d z_j) \) and the shares of stocks included in the portfolio. Constraints (4) are balance constraints that ensure that the rebalanced position \( w_j \) in asset \( j \) is equal to the initial position \( w^0_j \) increased (resp., decreased) by the purchased (resp., sold) shares \( y_j \) (resp. \( z_j \)) of \( j \). Constraint (5) does not allow short-selling. Constraint (2) requires the net return (after deduction of the transaction costs) of the portfolio to exceed the specified return level \( R \) with probability at least \( p \). Clearly, the asset allocation model SPM follows a downside risk measure [27] that probabilistically prevents the net return of the portfolio to fall short the minimum threshold return. As explained in more details in [4], the risk measure is closely affiliated with Roy’s safety-first risk metric [30] (see also Kataoka’s model [17]).

Our contribution is twofold. First, we derive a series of deterministic and convex formulations equivalent to the probabilistic Markowitz problem with proportional transaction costs. The proposed models take the form of quadratically constrained or second-order cone programming problems. Second, we analyze the computational efficiency of the proposed formulations for their usage in solving large-scale (up to 2000 securities) problems.
portfolio optimization problems. Although all the problems share similar properties (convexity, inclusion of second-order cone constraint), the tests reveal a great disparity in the computational times needed by top-notch nonlinear/conic optimization solvers, such as Cplex, Ipopt, and Mosek, to solve them to optimality. The solution time differential between some of the formulations can reach several orders of magnitude for the largest problem instances. The ability to choose the most efficient formulation is particularly important for algorithmic trading in a high-frequency context [1, 16], when speed is at a premium, as well as when trading constraints involving integer decision variables need consideration [4, 15, 18]. Indeed, such integer problems require, at each of the many nodes of the branch-and-bound/cut algorithm, the solution of a convex nonlinear problem (i.e., the continuous relaxation at this node). Section 2 presents the proposed convex reformulations for problem SPM. The computational efficiency of the reformulated problems is analyzed on Section 3.

2 Convex Equivalent Reformulations

In the form (1)-(5), problem SPM cannot be handled by any optimization solvers. In this section, we provide a number of deterministic and convex reformulations that are equivalent to problem SPM. Problem equivalence is understood as the fact that the considered problems accept the same optimal solution.

2.1 Standard Convex Reformulation

It was previously shown [4] that

\[ P(\xi^T w \geq R) \geq p \iff \mu^T w + F^{-1}_{(w)}(1-p) \sqrt{w^T \Sigma w} \geq R, \]

where \( F^{-1} \) is the inverse of the probability distribution \( F \) of the normalized portfolio return \( \frac{(\xi-\mu)^T w}{\sqrt{w^T \Sigma w}} \) and \( F^{-1}_{(w)}(1-p) \) is its \((1-p)\)-quantile. The following theorem demonstrated in [4] permits to characterize the above quadratic constraint:

**Theorem 1** [4] Let \( \Sigma \) be positive semi-definite and \( p \in [0, 1) \). If the probability distribution of \( \xi^T w \) is symmetric or has positive skewness, \( F^{-1}_{(w)}(1-p) \) is a negative number and \( \mu^T w + F^{-1}_{(w)}(1-p) \sqrt{w^T \Sigma w} \geq R \) is a second-order cone constraint.

To ease the notation, we set \( F^{-1}_{(w)}(1-p) = -\kappa \), with \( \kappa > 0 \) under the encompassing assumptions of Theorem 1. It immediately follows from Theorem 1 and (6) that the stochastic net portfolio return constraint (2) can be rewritten as the second-order cone constraint:

\[ \mu^T w - \sum_{j=1}^{n} (cy_j + dz_j) \geq R + \kappa \sqrt{w^T \Sigma w} . \]

Constraint (7) requires the net expected portfolio return to exceed the targeted return threshold augmented by a penalty term \( \kappa \sqrt{w^T \Sigma w} \). The inflicted penalty is an increasing function of the required safety level (i.e., \( p \) and \( \kappa \) increase in unison) and of the volatility of the portfolio return. Substituting (7) for (2) in problem SPM gives the convex optimization problem CP0

\[ \text{CP0} : \quad \min (1) \]

subject to \( (3); (4); (5); (7) \)
that minimizes a quadratic objective function subjected to the feasibility of a set of linear constraints and one second-order cone constraint. Often, the probability distribution of the portfolio return is not or partially known and the exact value of its quantiles has to be approximated. The Cantelli, Chebychev [4] and Camp-Meidell [20] probability inequalities can be used for approximating the value of the quantile, which allows for the formulation of a convex inner approximation of the probabilistic Markowitz problem SPM.

2.2 Convex Reformulations with Cholesky Decomposition

In this section, we use the Cholesky decomposition to derive the convex programming problems CP1 and EP1. As compared to the LU decomposition, the Cholesky factorization can be computed roughly twice faster and its memory requirements are approximately half as large [5].

Assuming that the symmetric variance-covariance matrix $\Sigma$ is positive semi-definite, the Cholesky decomposition involves the computation of the lower triangular matrix $C$ such that $\Sigma = CC^T$. Replacing $\Sigma$ by $CC^T$ in the objective function and in (7), and introducing the positive decision variable $h$ (11), we obtain the convex problem CP1

$$\text{CP1 : } \min \|C^T w\|_2^2$$
$$\text{subject to } \mu^T w - \sum_{j=1}^n (c y_j + d z_j) - R \geq h$$
$$\kappa \|C^T w\|_2 \leq h$$
$$(3); (4); (5)$$
$$h \geq 0$$

where $\|x\|_2$ denotes the Euclidean norm of the vector $x$. Note that we incorporate the auxiliary variable $h$ for computational purposes and for facilitating the interaction with the optimization solvers. Should the estimated variance-covariance matrix not be positive definite, methods have been proposed [9, 13] for constructing a positive definite matrix that is “as close as possible“ from the original one.

We shall now put problem CP1 in its epigraph form. The epigraph formulation is based on the introduction of an auxiliary variable $v$ to reformulate the optimization problem $\min_x f_0(x)$ as $\min_{x,v} v$ subject to $v \geq f_0(x)$. As shown below, the epigraph formulation of problem CP1 is the canonical formulation EP1 of a second-order cone programming problem [2].

**Proposition 1** Problem EP1

$$\text{EP1 : } \min h$$
$$\text{subject to } (3); (4); (5); (9); (10); (11)$$

is equivalent to problem CP1.

**Proof** Problem

$$\left\{ \min_{h,w} \|C^T w\|_2^2 : \mu^T w - \sum_{j=1}^n (c y_j + d z_j) - R \geq h , \kappa \|C^T w\|_2 \leq h , h \geq 0 \right\}$$
is equivalent to
\[
\begin{aligned}
&\min_{h,w,v} \, \|C^T w\|_2^2 \leq v , \, \mu^T w - \sum_{j=1}^{n} (c y_j + d z_j) - R \geq h , \, \kappa \|C^T w\|_2 \leq h , \, h \geq 0 \\
&\min_{h,w} \, \|C^T w\|_2^2 \leq h \}, \, h \geq 0
\end{aligned}
\]  
(14)

In (14), we minimize \(v\) which is defined as an upper bound on the variance on the portfolio. Note that \(h^2/\kappa^2\) is also an upper bound on the variance of the portfolio. This implies that \(v \geq \|C^T w\|_2^2\) is redundant, that the decision variable \(v\) can be omitted, and thus that the optimal solution of problem (13) is the same as that of
\[
\begin{aligned}
&\min_{h,w} \, \mu^T w - \sum_{j=1}^{n} (c y_j + d z_j) - R \geq h , \, \kappa \|C^T w\|_2 \leq h , \, h \geq 0 \\
&\min h, w : \, \mu^T w - \sum_{j=1}^{n} (c y_j + d z_j) - R \geq h , \, \kappa \|C^T w\|_2 \leq h , \, h \geq 0
\end{aligned}
\]  
(15)

which is what was set out to prove.

\[\square\]

### 2.3 Convex Reformulations with Period-Separable Formulation of Variance

The model proposed in this section uses Proposition 2 derived in [19] to reformulate the functions involving the variance of the portfolio as the Euclidian norm of a vector comprising a number \(M\) of components equal to the number of data points, i.e., periods. We show that the variance of the portfolio is separable into (and can be rewritten as the weighted summation of) a number \(M\) of squared terms \(b_t\) representing each the part of the variance associated with period \(t\).

**Proposition 2** [19] Let \(r_{jt}\) be the observed return of \(j\) at \(t\) and define the auxiliary variables \(b_t, t = 1, \ldots, M\) as \(b_t = \sum_{j=1}^{n} (r_{jt} - \mu_j) w_j\). The variance of the portfolio’s return can be rewritten as:

\[
w^T \Sigma w = \frac{1}{M} \|b\|_2^2.
\]

**Proof** The estimated mean return of \(j\) is \(\mu_j = \frac{1}{M} \sum_{t=1}^{M} r_{jt}\) and the elements \(\sigma_{ij}\) of \(\Sigma\) representing the unbiased estimate of the covariance between the returns of \(i\) and \(j\) are computed as:

\[
\sigma_{ij} = \frac{1}{M} \sum_{t=1}^{M} (r_{it} - \mu_i) (r_{jt} - \mu_j), \, i, j = 1, \ldots, r.
\]

(16)

Using (16), we have that:

\[
w^T \Sigma w = \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} w_i w_j = \sum_{i=1}^{n} \sum_{j=1}^{n} \left\{ \frac{1}{M} \sum_{t=1}^{M} (r_{it} - \mu_i) (r_{jt} - \mu_j) \right\} w_i w_j
\]

\[
= \frac{1}{M} \sum_{t=1}^{M} \sum_{i=1}^{n} \sum_{j=1}^{n} (r_{it} - \mu_i) w_i (r_{jt} - \mu_j) w_j
\]

\[
= \frac{1}{M} \sum_{t=1}^{M} \left\{ \sum_{i=1}^{n} (r_{jt} - \mu_j) w_j \right\}^2 = \frac{1}{M} \sum_{t=1}^{M} b_t^2.
\]

\[\square\]
We introduce $M$ decision variables $b_t$ unrestricted in sign and apply Proposition 2 for the reformulation of the probabilistic Markowitz model with transaction costs as the convex optimization problem $CP2$:

$$CP2 : \quad \min \frac{1}{M} \|b\|_2^2$$
subject to

$$(3); (4); (5); (9); (11)$$
$$\frac{\kappa}{\sqrt{M}} \|b\|_2 \leq h$$
$$b_t - \sum_{j=1}^{n} (r_{jt} - \mu_j) w_j = 0, \forall t$$

Observe that the modeling of the variance in problems $CP0$, $CP1$ and $EP1$ require the a priori estimate of $\frac{n(n+1)}{2}$ covariance terms and can generate model specification issues, such as the obtaining of a variance-covariance matrix that is not positive semi-definite [9, 13]. The computation of the variance proposed in this section does not have such limitations and does not make any assumption on the form or rank of $\Sigma$.

As we did for $CP1$, we provide the epigraph formulation $EP2$ of problem $CP2$:

$$EP2 : \quad \min h$$
subject to

$$(3); (4); (5); (9); (11); (18); (19)$$

3 Computational Results

The monthly returns of 2570 stocks over the period January 1999 to December 2008 ($M=120$ data points) were extracted from the CRSP US Monthly Stock Data Base available through the Wharton Research Data Service. All the selected stocks are traded on NYSE, Amex and NASDAQ and have no missing observation for the data item "Monthly Price Alternate" over the period 1999-2008. The Monthly Price Alternate provides the last available stock price of the month and accounts for splits and dividends.

We built problems of five sizes, respectively containing $r = 100, 200, 500, 1000$ and 2000 assets. For each problem size, we created eight problem instances. The stocks included in each instance were selected randomly. The initial positions $w_j^0, j = 1, \ldots, r$ are obtained by solving the standard Markowitz problem

$$\min \quad \bar{\mu}^T \Sigma \bar{\mu}$$
subject to

$$\sum_{j=1}^{n} \bar{w}_j = 1$$
$$\bar{w} \in \mathbb{R}_n^+,$$

and by setting $w_0 = \bar{w}^*$, with $\bar{w}^*$ denoting the optimal solution of the above problem. Each problem instance was modeled with the AMPL modeling language and solved with the solvers Cplex 11.1, Ipopt 3.8 and Mosek 6.0 on a 64-bit Dell Precision T5400 Workstation with Quad Core Xeon Processor X5460 3.16GHz CPU and 4X2GB of RAM. We used the default options for each solver. The Cplex 11.1 solver required the reformulation of the second-order cone constraints (10) and (18) as the equivalent convex quadratic constraints:

$$\kappa^2 \|C^T \bar{w}\|_2^2 \leq h^2 \quad \text{and} \quad \frac{\kappa^2}{M} \|b\|_2^2 \leq h^2.$$
The results for the 600 problem instances solved are summarized in Tables 1 and 2 which respectively report the average computing time and standard deviation for each solver, model and instance size. In what follows, excerpts from Tables 1 and Table 2 are used to shed light on the computational tractability of the five model formulations and on the performance of the three considered solvers.

First, we compare the computational times needed by each solver for each problem formulation. Figure 1 displays the results for the 2000-asset problem instances. Except for problem CP0 (Cplex is not able to handle constraint (7)), it can be seen that Cplex 11.1 is the fastest regardless of the formulation. The same comment extends for problem sizes \( r = 100,200,500,1000 \). For 2000 assets, as compared to Cplex 11.1, Ipopt 3.8 takes on average from 1.5 times (EP1) to 88.9 times (CP2) longer to solve a problem, while Mosek 6.0 takes from 1.5 (EP2) to 21.7 (EP1) times longer. The solver Ipopt performs better than Mosek on the formulations (CP0), (CP1) and (EP1), while the conclusion is reversed for the formulations (CP2) and (EP2). The solver Mosek 6.0 has very similar solution times as Cplex 11.1 for formulation EP2. Table 2 also shows that the standard deviation of the computational times with Cplex 11.1 is smaller than the one for the other two solvers.

Next, we focus on the tractability of the various formulations with the apparently best performing solver Cplex 11.1. Table 1 shows that problem CP0, which cannot be solved by Cplex 11.1, is the one that is the most (or second-most) time-consuming model to solve for each solver. The remaining part of the computational analysis will thus be based on the formulations CP1, EP1, CP2 and EP2 and their solution with Cplex 11.1.

For problems with 100 assets (see Table 1), the solution of CP1 takes 1.4 times longer than this of EP1, while it takes 2.4 more times to solve CP2 than EP2. For 2000 assets, the ratio of the solution times for CP1 and EP1 on one hand and between CP2 and EP2 on the other are respectively 1.9 and 3.8. Clearly, Cplex solves the epigraph formulations EP1 and EP2 faster than their counterparts CP1 and CP2. The difference in solution time between CP1 and EP1 and between CP2 and EP2 probably finds its root in the form of the objective function which is quadratic in CP1 and CP2, but linear in EP1 and EP2.

Table 1 also shows that the EP2 formulation is solved the fastest for problem instances including more than 100 assets, while EP1 is the formulation that is the easiest to solve for the 100-asset instances. Similarly, it is faster to solve CP2 than CP1 when more than 100 assets are considered. The explanation resides in the number of quadratic terms included in the nonlinear constraints (10) for EP1 and CP1 and (18) for EP2 and CP2. While problems CP1 and EP1 include \( n \) quadratic terms, CP2 and EP2 have \( M \) quadratic terms. As the number \( n \) of assets increase and the difference \( (n - M) \) between the number of assets \( n \) and periods \( M (=120) \) grows, the time differential to obtain the solution of EP1 (resp., CP1) and EP2 (resp., CP2) increases exponentially. Whereas the ratio of the average solution times for EP2 and EP1 is 14.7 with the 1000-asset instances, it amounts to 95.9 (104.52 / 1.09) with the 2000-asset instances! Similarly, Mosek 6.0 takes on average 2271 seconds to solve the EP1 formulation for the 2000-asset problem instances, while it takes hardly more than one second to solve the EP2 formulation. Clearly, the differences in solution times between some of the proposed reformulations can reach several orders of magnitude.
Table 1: Average Computational Times (in seconds)

<table>
<thead>
<tr>
<th></th>
<th>CP0</th>
<th>CP1</th>
<th>EP1</th>
<th>CP2</th>
<th>EP2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cplex</td>
<td>Ipopt</td>
<td>Mosek</td>
<td>Cplex</td>
<td>Ipopt</td>
</tr>
<tr>
<td>100</td>
<td>–</td>
<td>0.42</td>
<td>0.59</td>
<td>0.06</td>
<td>0.12</td>
</tr>
<tr>
<td>200</td>
<td>–</td>
<td>2.18</td>
<td>3.95</td>
<td>0.22</td>
<td>0.49</td>
</tr>
<tr>
<td>500</td>
<td>–</td>
<td>17.80</td>
<td>32.31</td>
<td>1.78</td>
<td>5.50</td>
</tr>
<tr>
<td>1000</td>
<td>–</td>
<td>120.34</td>
<td>208.67</td>
<td>14.78</td>
<td>44.99</td>
</tr>
<tr>
<td>2000</td>
<td>–</td>
<td>1167.36</td>
<td>1782.12</td>
<td>200.63</td>
<td>346.99</td>
</tr>
</tbody>
</table>

Table 2: Standard Deviation of Computational Times (in seconds)

<table>
<thead>
<tr>
<th></th>
<th>CP0</th>
<th>CP1</th>
<th>EP1</th>
<th>CP2</th>
<th>EP2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cplex</td>
<td>Ipopt</td>
<td>Mosek</td>
<td>Cplex</td>
<td>Ipopt</td>
</tr>
<tr>
<td>100</td>
<td>–</td>
<td>0.05</td>
<td>0.04</td>
<td>0.01</td>
<td>0.02</td>
</tr>
<tr>
<td>200</td>
<td>–</td>
<td>0.13</td>
<td>0.32</td>
<td>0.03</td>
<td>0.04</td>
</tr>
<tr>
<td>500</td>
<td>–</td>
<td>1.10</td>
<td>2.13</td>
<td>0.45</td>
<td>0.55</td>
</tr>
<tr>
<td>1000</td>
<td>–</td>
<td>2.02</td>
<td>4.55</td>
<td>1.90</td>
<td>8.08</td>
</tr>
</tbody>
</table>

Figure 1: Computational Efficiency of Solvers for 2000-Asset Instances
Figure 2: Computational Time as a Function of the Number of Assets (with Cplex)

Figure 2 displays the relationship of the computational time as a function of the number of assets. The computational time for CP1 and EP1 increases at a roughly exponential pace with the number of assets. For instance, solving EP1 takes on average 0.04 second for the 100-asset instances, while it takes 104.52 seconds for the 2000-asset instances. In contrast to this, the computational time increases almost linearly in the number of assets for the model EP2. The average solution times are respectively equal to 0.07, 0.12, 0.24, 0.56 and 1.09 seconds when the number of assets is equal to 100, 200, 500, 1000, and 2000, respectively. Evidently, the formulations CP2 and in particular EP2 are those that scale best. This highlights the benefits of our separable formulation of the variance of the portfolio return that is used in problems CP2 and EP2. Within this approach, the variance is calculated as the sum of a number of squared terms equal to the number of data points and the number of quadratic terms is invariant to the number of assets. Clearly, the period-separable representation of the variance is extremely beneficial for large scale portfolio optimization problems where the number of assets typically exceed by far the number of data points used for parameter estimation purposes. The additional (i.e., as compared to EP1) number $M$ of linear constraints (19) included in EP2 does not play a significant role on the computational times.

It is worth noting that the probabilistic Markowitz model with transaction costs is not only applicable to rebalance an existing portfolio, but can also be used to build a new portfolio. In that case, the investor does not hold any initial position, which implies that, in problem SPM, $w^0_j = 0$, $z_j = 0$ and $w_j = y_j$, and that constraints (2), (3) and (5) can be rewritten accordingly, while (4) is not needed. Table 3 presents the average computational time and its standard deviation for each instance size solved with the formulation EP2B that allows the construction of new portfolios:

$$\text{EP2B : } \min h$$
$$\text{subject to } \sum_{j=1}^{n} (\mu_j - c)w_j - R \geq h$$
$$\quad (11); (18); (19)$$
$$\sum_{j=1}^{n} (1 + c)w_j = 1$$
$$w \in \mathbb{R}_{++}^n.$$
Table 3 shows that the solution of the largest problem instances takes on average hardly more than 1 second. The comparison between Table 3 and the last columns of Tables 1 and 2 indicates that the solution of the problem allowing for the construction of a new portfolio is slightly faster than that of the problem allowing for the rebalancing of an existing portfolio.

<table>
<thead>
<tr>
<th>Average Time</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.05</td>
</tr>
<tr>
<td>200</td>
<td>0.08</td>
</tr>
<tr>
<td>500</td>
<td>0.24</td>
</tr>
<tr>
<td>1000</td>
<td>0.43</td>
</tr>
<tr>
<td>2000</td>
<td>1.06</td>
</tr>
</tbody>
</table>

The results of this study can be extended to and are valuable for portfolio problems which requires the tackling of fixed or upper-bounded transaction costs as well as trading constraints modeled with integer decision variables. Indeed, such integer problems require, at each of node of the branch-and-bound tree, the solution of a convex nonlinear problem very similar to the one considered in this study. The gain in computational time will be then multiplied by the number of nodes and continuous relaxation problems solved. This study also opens new perspectives to design new algorithmic high-frequency trading strategies [1, 16].

Acknowledgments

Tiago Filomena thanks the CAPES/Fulbright commission for their support (Grant # 2465-05-1). Miguel Lejeune is thankful for the support of the Army Research Office (Grant # W911NF-09-1-0497).

References


