European Journal of Operational Research 313 (2024) 628–636

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Decision Support

Computing cardinality constrained portfolio selection efficient frontiers via closest correlation matrices

Ralph E. Steuer[a,*], Yue Qi[b], Maximilian Wimmer[c]

[a] Department of Finance, University of Georgia, Athens, GA 30606, USA
[b] Department of Financial Management, Business School, Nankai University, Tianjin 300071, China
[c] Center for Finance, University of Regensburg, 93040 Regensburg, Germany

ARTICLE INFO

Article history:
Received 28 January 2023
Accepted 15 August 2023

Keywords:
Portfolio optimization
Efficient frontiers
Closest correlation matrices
Buyin thresholds
Cardinality constraints

ABSTRACT

In this paper, we demonstrate a completely new approach for computing cardinality constrained mean-variance efficient frontiers. By cardinality constrained, it is meant that if there is to be investment in a security, it is to be of at least some minimum amount (a buyin threshold), and that there is also a specification on the number of securities to be held in a portfolio (called a cardinality constraint). Whereas the usual strategy, as such problems are *NP*-hard, is to take the original exact problem and apply heuristics to solve, in this paper the strategy is to perturb the original problem and then apply exact procedures to solve. The advantages of the approach are that the perturbations are tiny, they are only applied to the problem's correlation matrix, and they allow for the accurate computation of cardinality constrained efficient frontiers in problems with up to at least 1000 securities in remarkably little time. Moreover, the simplicity of the approach is such that it can be inserted into existing portfolio management systems without requiring any re-training beyond what a typical portfolio analyst would already know.[1]

1. Introduction

We start by overviewing the problem of portfolio selection and Markowitz's mean-variance procedure for solving it to show how interest in cardinality constrained versions of the basic problem has evolved within this framework. By cardinality constrained, it is meant that if there is to be investment in a security, it is to be of at least some minimum amount (a buyin threshold), and that there is also a restriction on the number of securities to be held in a portfolio (called a cardinality constraint). Despite the fact that Markowitz portfolio selection has been around for over 70 years, it has been in only the last third of this period that interest in cardinality constrained versions of the mean-variance approach has come to the forefront.

Why it did not start earlier was because it took until this time for the dream of being able to solve mean-variance portfolio selection problems of almost any size to be achieved – only to find out that in large problems, virtually all generated solutions are unusable. What is meant by this? Assume an investor is trying to build a portfolio containing 20 securities. With the solutions produced by Markowitz's mean-variance approach containing, say, 35 to 40 securities (typical numbers), the solutions are not useful to the investor. As another example, suppose you are the manager of a mutual fund that traditionally contains 100 securities. Again, solutions with cardinalities in the range of 35 to 40 are of no use to you when in need of a number like 100 or so. Even if you were looking for a solution in the 35 to 40 range, it is highly likely there would be buyin threshold violations (more on this as we go) which wouldn't satisfy your situation either.

Indeed, cardinality constrained formulations of the mean-variance problem can be constructed whose solutions would be able to provide cardinality constrained users with what they would be looking for, if it were only possible to solve the problem, as such problems have eluded solution over the years. However, the contribution of this paper is the approach, the likes of which are different from any other approach, that enables the solution of cardinality constrained mean-variance portfolio selection problems with up to at least 1000 securities in size, in remarkably little time.

1.1. Background

As for history, while the idea of portfolio selection is widely known, that term and others relating to it that we take for granted really only go back to 1952. That was the year in which the

* Corresponding author.
E-mail addresses: rsteuer@uga.edu (R.E. Steuer), yorkche@nankai.edu.cn (Y. Qi), maximilian.wimmer@ur.de (M. Wimmer).

[1] The authors have benefited from associations with Markus Hirschberger and Sebastian Utz in this research.

https://doi.org/10.1016/j.ejor.2023.08.026
0377-2217/

paper by Markowitz (1952) was published for which he won the Nobel Prize. That paper has been of such impact that it can be viewed as dividing finance into two parts, with the first being before Markowitz (1952), and the second part being after.

It takes effort to imagine the field of investments prior to 1952 as there was no notion of an efficient portfolio, there was no notion of an efficient frontier, risk had not been settled upon, and diversification was for all practical purposes not putting all of one's eggs in the same basket. The focus of the era was on identifying good individual stocks to buy. But how to do this and know how much to invest in each was unclear as the only techniques available were mostly drawn from experience and common sense (such as in Graham & Dodd, 1934) as there was, beyond discounted cashflows, no formal theory of investments at the time. More on what investing was like before Markowitz can be found in Beattie, Brown, & Kvilhaug (2022) and Cifuentes (2022).

But all changed with Markowitz (1952). By introducing the concept of an efficient portfolio, Markowitz was able to tie together the best of what had been out there into an overarching theory expressible within the structure of a mathematical programming problem. The features of that paper are (a) its optimization model (almost certainly the most written about optimization problem of all time) by which efficient solutions are generated and (b) its decision-making prescription by which investors are to interact with the efficient solutions generated to identify one's optimal portfolio. As for (a), the optimization model is

$$\begin{aligned} &\min \ \mathbf{x}^T \mathbf{\Sigma}\, \mathbf{x} && \text{(M)}\\ &s.t. \ \boldsymbol{\mu}^T \mathbf{x} \geq \rho \qquad \rho \in [\rho_{\min}, \rho_{\max}]\\ &\mathbf{x} \in S \end{aligned}$$

where

(1) S is the feasible region defined by $x_i \geq 0$ for all i along with $\mathbf{1}^T\mathbf{x} = 1$, and at the discretion of the investor, other linear constraints as proposed in Markowitz (1956),
(2) n is the number of securities eligible for use in constructing a portfolio,
(3) $\mathbf{x} \in S$ is a portfolio whose components x_i (often called weights) are the proportions of capital invested in the n different securities,
(4) $\boldsymbol{\mu} \in \mathbb{R}^n$ is the vector of the individual security expected returns,
(5) $\mathbf{\Sigma}$ is the $n \times n$ covariance matrix of security returns whose elements σ_{ij} specify the covariances between the returns of securities i and j,
(6) $\mathbf{x}^T\mathbf{\Sigma}\mathbf{x}$ is the matrix expression of portfolio return variance,
(7) $\boldsymbol{\mu}^T\mathbf{x}$ is expected portfolio return, and
(8) (M) is to be solved for all ρ in the interval indicated where $\rho_{\min}$ is the value of expected portfolio return at the solution of minimum variance and $\rho_{\max}$ the value of expected portfolio return at the solution of maximum return.

This is the mean-variance model of Markowitz portfolio selection, where "mean" is expected portfolio return $\boldsymbol{\mu}^T\mathbf{x}$ and "variance" is short for portfolio return variance $\mathbf{x}^T\mathbf{\Sigma}\mathbf{x}$, the quantity that measures risk. Note how the model's emphasis is on portfolios rather than on individual securities as was earlier the case.

With a portfolio $\mathbf{x} \in S$ being *efficient* if and only if there exists no other portfolio whose risk is better without its expected return being worse (or whose expected return is better without its risk being worse), this means that when (M) is solved for all $\rho \in [\rho_{\min}, \rho_{\max}]$, the set of all efficient portfolios is returned. The importance of this set, the *efficient set*, is that it contains, and only contains, all portfolios that could be optimal for an investor. This means that the investor's most preferred portfolio in the set is the investor's optimal portfolio. But since efficient sets typically contain many points, it would be nice if some kind of assistance were available. This is where (b), the decision-making part of Markowitz (1952), comes in. Based upon the *efficient frontier*, which is the curve that results from plotting the mean-variance combinations of all portfolios in the efficient set, the usefulness of the curve is that by selecting the most preferred mean-variance combination on it, the investor will then have, after taking its inverse image, his or her optimal portfolio.

This provides an accommodating environment for an investor. It allows different investors to have different optimal solutions depending upon where along the efficient frontier their most preferred mean-variance combination lies. Furthermore, the environment allows investors, when picking their most preferred mean-variance combination, to see all other candidates for optimality at the same time. Thus, when picking one's most preferred point, the decision maker is aware that this is a global choice.

While computers didn't have the slightest chance of being able to solve even a toy version of (M) in 1952, this didn't stop the paper's theoretical content from having effect. In contrast to before, Markowitz (1952) presented a concise and insightful view of what the portfolio selection problem is, how it is to be solved ideally, and how the solution process can be tempered to the needs of different investors. Moreover, due to the material's teachability, this has enabled universities for many years now to have taught the essentials of Markowitz portfolio selection to large numbers of students around the world without getting the students caught up in the model's implementation difficulties.

This is fortunate as there have always been implementation difficulties with the model of Markowitz portfolio selection. Despite the impact of Markowitz (1952) on finance, there has always been the question that if Markowitz portfolio selection is so important, how come its model is not used more in practice? In their review of the 60 years of portfolio selection since 1952, Kolm, Tütüncü, & Fabozzi (2014) point out that one of the reasons for this has been the computational challenges that have forever dogged Markowitz portfolio selection since its introduction.

In the beginning it was not enough computing power nor enough storage. Storage was involved because (M), a quadratic programming problem, is not an ordinary quadratic programming problem. In ordinary quadratic programs, only small percentages of the objective function coefficients are nonzero, but in (M) it is often the case that all coefficients are nonzero as this is what typically results when computing $\mathbf{\Sigma}$ from historical data. While storage is no longer a problem, the density of $\mathbf{\Sigma}$ still causes (M) to be among the most difficult of quadratic programs to solve.

In addition, there is the issue of solving (M) for all $\rho \in [\rho_{\min}, \rho_{\max}]$. Due to the difficulties in doing this over a continuous interval, it has been welcomed as satisfactory from early on to solve (M) repetitively for a sufficiently large number of discrete values of ρ coming from $[\rho_{\min}, \rho_{\max}]$ and the world has become used to this. The idea is to solve (M) for enough values of ρ so that after plotting the mean-variance combinations that result, nicely dotted versions of the efficient frontier, such as commonly seen in textbooks, are produced. However, with the computing power and algorithmic advancements that had come along by year 2000, it was then possible to compute well-dotted representations of the efficient frontiers of dense covariance matrix mean-variance portfolio selection problems with well up to 1000 securities in non-prohibitive time.

One might think that this would be just what was needed so that Markowitz portfolio selection would no longer be consigned to the mostly smaller problems it had been limited to in the first part of its existence. Unfortunately, this did not turn out to be the case. It turned out that in overcoming the challenge of size, this exposed two other challenges that had been laying low in small problems as they did not cause trouble there, but they are major challenges in larger problems. One is cardinality as discussed

earlier. The other has to do with very small weights that can be anticipated to show up in the efficient portfolios of large problems. Because of the overhead each security adds to the administration of a portfolio, investors almost universally only want to hold a security in at least some buyin threshold amount or not at all. Thus when a weight is smaller than its security's threshold amount, a situation that only becomes more prevalent the larger the problem, this makes it frustrating on the part of an investor to know what to do with it. If you round down, where do you distribute the weight freed up? If you round up, where do you get the extra weight from? With it not uncommon for more than one such frustration to occur in a large-problem efficient portfolio (Steuer, Hirschberger, & Deb, 2015), junctures can be easily reached at which dealing with Markowitz portfolio selection can be more trouble than it is worth.

Note the perverseness of the situation. Earlier, we couldn't use mean-variance portfolio selection on large problems because of computer technology, but now after computer technology is no longer a limitation, we can't use mean-variance portfolio selection on large problems because of cardinality and buyin threshold difficulties. Either way, we are blocked, with the situation only the worse the larger the problem.

The resulting challenge is how to prevent solutions with not-wanted cardinalities and security weights positive but below their buyin thresholds from being computed in the first place. This is accomplished by equipping (M) with (a) semi-continuous variables by which either nothing or at least some threshold amount is to be invested in a security and (b) a cardinality constraint. In this way, if we can solve the problem with semi-continuous variables and a cardinality constraint, we get the cardinalities we want and the difficulty of insufficiently large weights plaguing efficient solutions is eliminated.

In such a fashion, with semi-continuous variables and a cardinality constraint, it is hard to imagine more realistic versions of (M) for general use. Also, with a cardinality constraint, all portfolios that make up a problem's efficient frontier can be made to possess the same cardinality, which is all but impossible with continuous variables. With semi-continuous variables and a cardinality constraint integrated into (M), this gives us the following cardinality constrained mean-variance portfolio selection problem formulation

$$\begin{aligned} &\min \ \mathbf{x}^T \boldsymbol{\Sigma} \mathbf{x} && \text{(CC)}\\ &s.t. \ \ \boldsymbol{\mu}^T \mathbf{x} \geq \rho \\ &\mathbf{1}^T \mathbf{x} = 1 \\ &\sum_{i=1}^{n} y_i = K \\ &Ly_i \leq x_i \leq Uy_i \quad \text{for all } i \\ &y_i \in \{0, 1\} \quad \text{for all } i \end{aligned}$$

where

(1) feasible region S of (CC) is specified by the last four lines of the formulation,
(2) the x_i variables are no longer continuous, but are now semi-continuous,
(3) L is the buyin threshold and U is the upper bound on an x_i when $x_i \neq 0$,
(4) the y_i are binary variables used to implement both the semi-continuous variable and cardinality constraint parts of the problem,
(5) K is the cardinality of a portfolio (the number of positive x_i-values defining the portfolio), and
(6) it is implied that the formulation is to be solved repetitively for a sufficiently dispersed and large number of ρ from $[\rho_{\min}, \rho_{\max}]$.

Unfortunately, cardinality constrained mean-variance portfolio selection problems are *NP*-hard (Bienstock, 1996 and Gao & Li, 2013). With this and the runaway solution times that are so often the case when working with *NP*-hard problems, this has caused scores of researchers to pursue heuristics to develop approximations to the solutions of (CC) that couldn't be computed exactly. Note how (CC) embraces the minimum transaction lot problems studied in Mansini & Speranza (1999) and Lin & Liu (2008).

Whereas in the approaches of the many researchers who have addressed (CC) the strategy has been to take the original exact problem and then apply heuristics to solve, in this paper the strategy is to perturb the original problem and then apply exact procedures to solve. By "perturbing" the original problem it is meant to make adjustments to the correlation matrix of a problem's $\boldsymbol{\Sigma}$ covariance matrix to make it near positive definite or perhaps even positive definite. Then the adjusted correlation matrix is used to construct a re-constituted version of the original covariance matrix (which is only slightly different from the original covariance matrix) to be used in place of the original $\boldsymbol{\Sigma}$ in (CC). Not only enabling (CC) to run on state-of-the-art solvers, because the adjustments are so small, the approach hardly affects solution accuracy as we will see. In addition, the approach is easy to implement, it is easy to understand, and it allows cardinality constrained problems with up to at least 1000 securities to run in very rapid time.

The organization of the rest of this paper is as follows. In Section 1.2, we review the literature on the variety of heuristics that have been proposed for cardinality constrained mean-variance portfolio selection problems. In Section 2 we describe the *closest correlation matrix covariance matrix* approach of the paper and describe the random portfolio selection problem generator used to generate the many test problems needed for the experiments of this paper. In Section 3 experiments attesting to the smallness of the perturbations and the accuracy of the method are presented, and in Section 4 the results of our experiments in a production environment are reported. In Section 5 we end the paper with concluding remarks.

1.2. Literature review

Although interest in special methods for developing solutions to the mean-variance cardinality constrained portfolio selection problem can be traced back to the late 1990s, research did not begin to appear in volume on the subject until after the appearance of the article by Chang, Meade, Beasley, & Sharaiha (2000). The significance of that paper is that it showed in very clear terms the relevance of the mean-variance cardinality constrained portfolio selection problem, its issues, and the potential of inexact methods (i.e., heuristics) to solve for the efficient frontier of such problems given that state-of-the-art solvers can not be counted on to be effective on such problems. Thus, the rush was on for who could develop the best heuristic for solving the mean-variance cardinality constrained portfolio selection problem.

To not repeat what already has been done, it is pointed out that two reviews of the literature on the mean-variance cardinality constrained portfolio selection problem (CC) already exist. The first, contained in the article by Woodside-Oriakhi, Lucas, & Beasley (2011), covers 22 articles published between the time of the Chang et al. (2000) and 2010, grouped by type of heuristic into three categories. The categories are: exact procedure modification approaches (5 papers), single metaheuristic approaches (11 papers), and multiple metaheuristic approaches (6 papers). In the exact procedure modification category, the heuristics are constructed out of exact method tools such as special terms and cuts as for example in the papers of Shaw, Liu & Kopman (2008) and Bertsimas & Shioda (2009). In the single metaheuris-

tic category, the papers employ single heuristics such as simulated annealing, neural networks, and multiobjective evolutionary algorithms. Examples are found in the neural networks paper of Fernández & Gómez (2007) and in the multiobjective evolutionary paper by Branke, Scheckenbach, Stein, Deb, & Schmeck (2009). In the multiple metaheuristic category, where the papers involve two or more metaheuristics, sometimes in combination and other times for use in comparing against one another, we have for example Ehrgott, Klamroth, & Schwehm (2004) and Ruiz-Torrubiano & Suárez (2010) in which simulated annealing, tabu search, and genetic algorithms are used.

The second review is contained in Kalayci, Ertenlice, Akyer, & Aygoren (2017) and covers heuristics published between the time of Woodside-Oriakhi et al. (2011) through 2015. In this review are listed an additional 38 papers with 5 falling into the exact procedure modification category in which Cesarone, Scozzari, & Tardella (2013) and Cui, Zheng, Zhu, & Sun (2012) are examples; 24 falling into the single metaheuristic category in which Sadjadi, Gharakhani, & Safari (2012) and Lwin, Qu, & Kendall (2014) are examples; and 9 falling into the multiple metaheuristic category in which Anagnostopoulos & Mamanis (2011) and Baykasoğlu, Yunusoglu, & Özsoydan (2015) are examples. It is interesting to note that out of the 24 in the single metaheuristic category, 17 used particle swarm, ant colony, artificial bee colony, and firefly optimization strategies (as in Deng & Lin, 2010; Deng, Lin, & Lo, 2012; and Kalayci et al., 2017) in contrast to only one swarm optimization heuristic (Cura, 2011) showing up in the first review.

Since the Kalayci et al. (2017) review, another 13 articles presenting heuristics of which we are aware have appeared such as by Bertsimas & Cory-Wright (2022), Cesarone, Scozzari, & Tardella (2015), and Juszczuk, Kaliszewski, Miroforidis and Podkopaev (2022), Liagkouras & Metaxiotis (2018a,b) and Salehpoor & Molla-Alizadeh-Zavardehi (2019) to name five. By the way, Salehpoor & Molla-Alizadeh-Zavardehi (2019) also contains an excellent bibliography.

With all of the 73 papers alluded to above, and with a few of them proposing more than one, this brings us to over 70 heuristics that have be proposed. A problem with all of the heuristics is that they involve intricate procedures that would require a specialist in any procedure in question to operate. Consequently, they would not be an easy sell to Wall Street (that is, financial markets in general) due to the considerable amounts of often esoteric detail that one would have to have knowledge about to operate. Moreover, other than for only a few of the heuristics such as by Cesarone et al. (2013) and Bertsimas & Cory-Wright (2022) that have been subjected to considerable testing, the computational experience reported in support of the bulk of the heuristics has been limited to only a handful of small test problems often from Beasley (2000), so we don't yet know what to expect from them over the long haul and on larger problems. In addition to all of this, run times are not short and this would get in the way of them being used in practice. In summary, there is not enough standard, straightforward, and familiar about them for one to feel that many might be on the verge of being used in practice.

However, the approach of this paper is different. With the approach only involving minute changes being made to the covariance matrix, and with that being easily carried out in Matlab, the approach does not require a specialist to operate like the heuristics just discussed. Actually, its simplicity is such that it would not be beyond a portfolio manager to carry out the approach himself or herself if it came down to it. Also, the approach would rank high in being explainable to clients having any kind of interest in the process. Furthermore, the speed at which the approach can be carried out, the size of the problems that can be addressed, and the accuracy of the solutions generated is not matched by any of the 70+ heuristics.

To instill confidence in our approach, all meaningful aspects of the approach have been subjected to extensive computational testing to provide an understanding about what to expect from the approach in practice as will follow shortly. But first, in the next section, we algorithmically describe the approach.

2. The approach and test problem generator

While the driving force behind the 70+ heuristics recapped in the previous section has been the *NP*-hardness of (CC) brought on by the binary variables, it is not out of the question that other things could be at play. For instance, in Cplex, if $\mathbf{\Sigma}$ is not of full or near full rank, Cplex will most likely return the error message "Covariance Matrix Not Convex" and not start. In using Gurobi, another state-of-the-art solver (our favorite), one might get the message "Objective Q not PSD". In our analysis, it is not *NP*-hardness that is the most serious cause of solvers like Cplex and Gurobi not being able to solve instances of (CC). It is the degree to which $\mathbf{\Sigma}$ is not of full rank that is the culprit. That is, with $\mathbf{\Sigma}$ usually 120 or less in rank when constructed from historical return data, success in problems with more than a few percent more in securities becomes hard to come by.

Thus, rather than *NP*-hardness, this has caused us to think in the direction of the covariance matrix with our approach for enabling the solution of (CC). Under the prospect that a non-positive definite covariance matrix can have its rank improved by making small adjustments to its elements, we pursue an approach for problems with non-positive definite covariance matrices that endeavors to improve the rank of the covariance matrix without changing the matrix much as follows.

We first take the problem's non-positive definite covariance matrix and decompose it into its standard deviation vector and its correlation matrix. Then, using a routine drawn from the research of Qi & Sun (2006), which in turn was inspired by research from Higham (1988, 2002), we strive to compute the positive definite correlation matrix that is as close as we are able to compute to the correlation matrix of the original covariance matrix. (This is because the set of all positive definite correlation matrices is open.) Then, by pre- and post-multiplying the computed correlation matrix by the diagonal matrix along whose main diagonal is the standard deviation vector, we form a slightly different version of the original covariance matrix that we call the *closest correlation matrix covariance matrix*, whose rank should be much improved. With this matrix used in place of the original covariance matrix, we take (CC) to a state-of-the-art solver, which in this paper is Gurobi, for solution. Of course, the solutions generated when using the closest correlation matrix covariance matrix will not be precisely optimal, but as we will see, they will be remarkably close. Step-by-step, the closest correlation matrix covariance matrix routine is as follows.

(1) Let $\mathbf{\Sigma}$ be the starting covariance matrix.
(2) Compute its standard deviation vector and correlation matrix **Corr**. This is done by means of `cov2corr` in Matlab. Then form the diagonal matrix **Std** whose main diagonal is the standard deviation vector.
(3) Compute **pdCorr** where it is to be the positive definite correlation matrix "closest" to **Corr**. This is done by invoking the procedure of Qi & Sun (2006) in the form of the Matlab executable `CorNewton1` downloaded from the first author's website several years ago. By *closest*, it is meant in accordance with the Frobenius norm. With the Frobenius norm of a matrix being the square root of the sum of the squares of all elements in the matrix, **pdCorr** is then the matrix **X**

Table 1
Parameter settings used by the random portfolio selection problem generator to generate the covariance matrices $\mathbf{\Sigma}$ and expected return vectors $\boldsymbol{\mu}$ used in the problems of the experiments of this paper.

Parameter	Quantity	Value
$\mathbf{\Sigma}$ main diagonal elements	mean	0.00554
	stdev	0.00667
$\mathbf{\Sigma}$ off-diagonal elements	mean	0.00124
	stdev	0.00115
$\boldsymbol{\mu}$-vector elements	mean	0.00899
	stdev	0.00938

obtained by solving

$$\begin{aligned} \min \quad & \frac{1}{2}\,\|\mathbf{Corr} - \mathbf{X}\|^2 \\ s.t. \quad & x_{ii} = 1 \quad i = 1 \ldots n \\ & \mathbf{X} \in \mathbb{S}_+^n \end{aligned} \tag{1}$$

where $\mathbb{S}_+^n$ is the set of all symmetric $n \times n$ positive definite matrices.

(4) Upon computing **rSig** = **Std** ∗ **pdCorr** ∗ **Std**, **rSig** is the recomputed covariance matrix, called the *closest correlation matrix covariance matrix* in this paper.

(5) With **rSig** used in place of $\mathbf{\Sigma}$, (CC) is sent to a state-of-the-art solver such as Cplex or Gurobi for solution.

Since $\mathbb{S}_+^n$, the set of all symmetric $n \times n$ positive definite matrices, is an open set, program (1) is only run until a tolerance is satisfied such as 1.0e-4 (default value in `CorNewton1`). While **pdCorr** is to be positive definite, there is no exact guarantee because of the tolerance. But in our experiments, we have found that **rSig** normally computes to over 95% of full rank as shown in the next section, and that has been enough to enable the computation of all 100-point efficient frontiers of all of the problems of this paper.

Because there are not enough real-world portfolio selection problems to supply the test problem needs of this paper, we call upon a random portfolio selection problem generator. The random problem generator employed is patterned after the random portfolio selection problem generator used in Hirschberger, Qi, & Steuer (2010). In accordance with this, the covariance matrices $\mathbf{\Sigma}$ of the problems generated are produced by the routine described in Hirschberger, Qi, & Steuer (2007).

For realism in the generation of the $\mathbf{\Sigma}$ and μ in the problems generated for this paper, the random problem generator was parameterized with data drawn from the stocks that were in the S&P500 over the 5-year period January 1, 2015 through December 31, 2019, similar to as was done in Steuer & Utz (2022) but with different seeds. These parameter values are as given in Table 1.

Using these parameter values, we have in Table 2 the steps of the random problem generator for generating the problem situations needed in this paper. With regard to Step 1, in this paper we experiment with problem sizes of 250, 500, 750 and 1000 securities to show how the approach works in problems of these sizes. Note that in the random problem generator, we create two versions of each instance of (CC) generated. In the first version, to generate the covariance matrices $\mathbf{\Sigma}$ in accordance with the S&P500 data, the main diagonal elements of the covariance matrices are designed to average 0.00554 with a standard deviation of 0.00667, and the off-diagonal elements of the covariance matrices are designed to average 0.00124 with a standard deviation of 0.00115. This is done in Step 2. To generate the expected return vectors $\boldsymbol{\mu}$ in accordance with the S&P500 data, their elements are designed to average 0.00899 with a standard deviation of 0.00938. This is done in Step 3. After this in Step 4, values for L, U and K are supplied to complete the specification of the instance of (CC) generated.

That completes the first version.

To create the second version of the instance of (CC) just generated, we compute $\mathbf{\Sigma}$'s closest correlation matrix covariance matrix **rSig** in Step 5, Then in Step 6, instead of $\mathbf{\Sigma}$, we use **rSig** as the covariance matrix in the second version of the instance of (CC). This is done so that the effectiveness of the closest correlation matrix covariance matrix approach can be tested problem by problem.

CPU times per problem for carrying out the construction of **rSig** from its $\mathbf{\Sigma}$ using Steps **(2)** through **(4)** of the closest correlation matrix covariance matrix routine specified earlier in the section are given in Table 3. For instance, the time taken to convert a covariance matrix $\mathbf{\Sigma}$ of size 750 to its **rSig** shows an average of 7.14 seconds.[2]

3. Physical and operational closeness of the rSig

In this section we show how close a problem's closest correlation matrix covariance matrix **rSig** is to its original covariance matrix $\mathbf{\Sigma}$ physically, and how close the results of using **rSig** vs. $\mathbf{\Sigma}$ test out in an operational setting. Then we discuss the differences between the ranks of **rSig** and $\mathbf{\Sigma}$ followed by some comments about the use of state-of-the-art solvers Cplex and Gurobi. Consider Table 4.

The first four rows show how close the elements of the closest correlation matrix covariance matrices **rSig** are to their corresponding elements in their randomly generated $\mathbf{\Sigma}$'s when computed by the approach. On average, in rows 1 and 2, the numbers show that corresponding diagonal elements in the two types of covariance matrices only begin differing from one another in the seventh place to the right of the decimal point, and on average, in rows 3 and 4, the numbers show that corresponding off-diagonal elements only begin differing from one another in the eighth place to the right of the decimal point. In this way, with the differences very small, the **rSig** are very close *physically* to their $\mathbf{\Sigma}$, with the relationship holding steady across covariance matrix sizes.

In rows 5 to 7, we show the results of testing the $\mathbf{\Sigma}$ and their **rSig** in an *operational* fashion. We do this by using the $\mathbf{\Sigma}$ and **rSig** to compute their regular (i.e., continuous-variable) efficient frontiers. The efficient frontiers providing rows 5 and 6 were obtained by running (CC) with $L = 0$, $U = 0.03$ (a reasonable value), and the cardinality constraint dropped. Known as early as Markowitz (1956), the obtained efficient frontiers are piecewise parabolic. That is, each consists of a connected string of parabolic line segments. In our experiments, the mean number of parabolic line segments making up the continuous-variable efficient frontiers when run with $\mathbf{\Sigma}$ range from 184.16 to 269.12 as seen in row 5. Row 6 shows the degree to which the number of parabolic line segments can vary from one problem to the next.

With regard to row 7, what the 5-tuples mean is as follows. Over the experiments, only five types of discrepancies were observed between the two continuous-variable efficient frontiers, that is, when the **rSig** efficient frontiers had 4 fewer, 2 fewer, 1 fewer, 1 more, and 2 more parabolic line segments than their corresponding $\mathbf{\Sigma}$ efficient frontiers. This is what the $(-4, -2, -1, 1, 2)$-tuple is all about. In this way, the (0,2,0,0,0) in the covariance matrix size 750 column means that out of the 25 covariance matrices of this size tested, 23 had identical numbers of parabolic line segments with only two having discrepancies, both with the **rSig** efficient frontier having 2 fewer parabolic line segments. In this way, looking at the other 5-tuples, the efficient frontiers produced by the **rSig**'s of the approach are very close in this sense *operationally* to the efficient frontiers produced by their $\mathbf{\Sigma}$'s.

With all of the covariance matrices produced by the random problem generator having a rank of 23, row 9 shows the ranks to

[2] All of the computational work in this paper was run on an HP Z2 Tower G5 Desktop with 32GB of RAM at the University of Georgia.

Table 2
Steps for generating two versions of randomly generated instances of (CC), one with $\mathbf{\Sigma}$ as its covariance matrix, and the other with $\mathbf{\Sigma}$'s closest correlation matrix covariance matrix **rSig** as its covariance matrix.

1. Input the portfolio selection problem's size n.
2. In accordance with the first four parameters of Table 1, randomly generate covariance matrix $\mathbf{\Sigma}$.
3. In accordance with the fifth and sixth parameters of Table 1, randomly generate expected return vector $\boldsymbol{\mu}$.
4. Equip instance of (CC) with $\mathbf{\Sigma}$, $\boldsymbol{\mu}$, and values for L, U and K.
5. Employing the approach of this paper, compute $\mathbf{\Sigma}$'s closest correlation matrix covariance matrix **rSig**.
6. Equip a second instance of (CC) with all the same except for **rSig** replacing $\mathbf{\Sigma}$.

Table 3
Average times (in seconds) to convert a covariance matrix to its closest correlation matrix covariance matrix counterpart using the Matlab routine of `CorNewton1` by Qi & Sun (2006). Sample size 100 for each covariance matrix size.

Covariance matrix size	250	500	750	1000
Conversion time (secs)	0.75	3.18	7.14	12.31

Table 4
In the first 4 rows, corresponding elements between **rSig** and $\mathbf{\Sigma}$ are compared. In terms of parabolic line segments, rows 5 to 7 show that there is hardly any difference between the continuous-variable efficient frontiers when run with $\mathbf{\Sigma}$ vs. **rSig**. In rows 8 and 9, the differences in rank between the originally generated $\mathbf{\Sigma}$'s and their **rSig**'s are shown. Sample size 25 for each covariance matrix size.

row	Covariance matrix size	250	500	750	1000
1	AveDiagElemDiff	0.000001302	0.000001370	0.000001394	0.000001408
2	AveDiagElem%Diff	0.0245%	0.0258%	0.0263%	0.0266%
3	AveOffDiagElemDiff	0.000000544	0.000000569	0.000000574	0.000000578
4	AveOffDiagElem%Diff	0.2077%	0.3706%	0.4787%	0.4267%
5	MeanNumParabolicSegs	184.16	221.60	237.77	269.12
6	StdDevNumParabolicSegs	16.52	23.83	17.97	20.82
7	Discrepancies $(-4,-2,-1,1,2)$	(0,1,0,1,3)	(0,0,2,1,3)	(0,2,0,0,0)	(3,3,0,0,4)
8	AveRank($\mathbf{\Sigma}$, 0.0000005)	23	23	23	23
9	AveRank(**rSig**, 0.0000005)	244.64	488.80	731.32	974.20

Table 5
By size, the percentages of randomly generated $\mathbf{\Sigma}$'s that cause Gurobi to respond with the message "Objective Q not PSD". Sample size 500 for each covariance matrix size.

Covariance matrix size	250	500	750	1000
Gurobi Q not PSD percentage	5.4%	5.6%	7.4%	8.0%

which they have been raised by the closest correlation matrix covariance matrix approach of the paper. That is, they were all transformed into **rSig**'s with ranks averaging over 97.4% of full rank.[3]

We now need to talk about Cplex and Gurobi. With Cplex having been introduced in 1988 and Gurobi only in 2008, Cplex is the best known and most widely used. Reflecting this, in the papers of the 70+ heuristics, whenever a reference is made to a state-of-the-art solver, it has always been to Cplex. But now there is Gurobi. While many users have a lot invested in Cplex and would have no interest in changing, we have found Gurobi to have advantages in cardinality constrained mean-variance portfolio optimization. The situation is this. Because of the lowness of the ranks of $\mathbf{\Sigma}$ relative to their n, Cplex is not able to solve any of the randomly generated problems of this paper. But when equipped with **rSig**'s as computed by the approach, they all run. However, with Gurobi, most problems run with just $\mathbf{\Sigma}$, while those that do not run all run when (CC) is equipped with **rSig**.

Testing how often randomly generated covariance matrices (all of which do not run on Cplex) don't run on Gurobi either yields Table 5. For instance, over a sample of 500 randomly generated covariance matrices with 750 securities, Gurobi only bounced 37 back with the message "Objective Q not PSD" and refused to start. Note how the percentages in the table only gradually increase with problem size. But by applying the approach, i.e., by computing the problem's **rSig** and then re-running, solutions were obtained to all (CC) in the same high-speed fashion as with the problems Gurobi had no problem solving in the first place.

In Table 6 we take another look at the operational closeness of the two covariance matrices. In the tests of this table, we compare the minimum variances achieved when using the $\mathbf{\Sigma}$ as opposed to when using their **rSig**. This is done by solving (CC) with its $\boldsymbol{\mu}^T\mathbf{x} \geq \rho$ constraint deleted for the 16 different (problem size, K, L, U) combinations indicated. In the table, the avminVar are the average minimum variances of the problems whose covariance matrix is $\mathbf{\Sigma}$, and the avAbsDif show the average absolute differences between the minimum variances of problems solved with $\mathbf{\Sigma}$ as the covariance matrix and those of the same problems but solved with **rSig** as the covariance matrix. Note how small the avAbsDif are relative to their avminVar, showing once again how very close operationally the **rSig** are to their $\mathbf{\Sigma}$.

4. Experiments

In Table 7, for 10 different cardinalities K and buyin thresholds and upper bounds appropriate to these K-values, we report on experiments for computing 100-point cardinality constrained mean-variance efficient frontiers for problems of sizes 250, 500, 750 and 1000 securities. In rows 4 and 5 of each panel are the means and standard deviations of the Inner run times experienced where Inner run time is Gurobi 9.5 Runtime as Gurobi was the solver applied. In row 8 of each panel are the means of the Outer run times experienced where Outer run time is Inner run time plus the time to make **rSig** and all other overhead involved in the 100-point efficient frontier construction process. For the 100 points of each cardinality constrained efficient frontier, 100 optimizations were conducted (one for each of the 100 equally-spaced ρ-values taken from the problem's $[\rho_{\min}, \rho_{\max}]$) with the indexing such that optimization 1 starts with $\rho_{\min}$ and optimization 100 finishes with $\rho_{\max}$.

[3] As assessed by the rank function in Matlab with tolerance 0.0000005.

Table 6
Statistics showing that the use of **rSig**'s has virtually no effect on solution accuracy. Sample size 25 for each (problem size, K, L, U) combination.

	K	10	20	40	70
	$[L, U]$	[0.05,0.20]	[0.03,0.10]	[0.02,0.06]	[0.01,0.03]
250	avminVar	0.000166967	0.000195995	0.000270676	0.000347326
	avAbsDif	0.000000011	0.000000043	0.000000220	0.000001039
500	avminVar	0.000112421	0.000132217	0.000182528	0.000234018
	avAbsDif	0.000000005	0.000000139	0.000000494	0.000001685
750	avminVar	0.000089572	0.000104147	0.000145572	0.000185384
	avAbsDif	0.000000005	0.000000172	0.000000895	0.000002289
1000	avminVar	0.000079270	0.000090526	0.000124956	0.000160779
	avAbsDif	0.000000005	0.000000160	0.000001100	0.000001572

Table 7
Run times (in seconds) and other statistics for computing 250, 500, 750 and 1000-security, 100-point cardinality constrained mean-variance efficient frontiers for different values of K. Sample size 25 for each of the forty (n, K, U, L) combinations in the panels.

Panel A $n = 250$ $pts = 100$											
1	K	3	5	10	15	20	30	40	70	100	150
2	U	0.50	0.40	0.30	0.25	0.20	0.15	0.10	0.07	0.04	0.03
3	L	0.10	0.08	0.05	0.04	0.03	0.02	0.01	0.005	0.004	0.003
4	MeanInnerRuntime	21.21	10.98	7.17	7.79	7.43	6.21	6.07	3.36	3.74	4.03
5	StDevInnerRuntime	7.67	4.21	9.73	0.57	0.65	0.59	8.96	0.28	0.38	0.42
6	AvOptMaxIdx	25.92	27.16	41.92	37.32	38.16	48.96	64.00	64.40	72.92	76.96
7	OptMaxToAve	2.08	2.04	1.78	1.79	1.83	1.84	9.09	2.01	1.76	1.57
8	MeanOuterRuntime	22.07	12.65	8.77	9.40	9.03	7.85	7.61	4.86	5.37	5.66
Panel B $n = 500$ $pts = 100$											
1	K	3	5	10	15	20	30	40	70	100	150
2	U	0.50	0.40	0.30	0.25	0.20	0.15	0.10	0.07	0.04	0.03
3	L	0.10	0.08	0.05	0.04	0.03	0.02	0.01	0.005	0.004	0.003
4	MeanInnerRuntime	89.36	41.30	27.79	29.13	26.61	22.48	13.21	12.24	14.50	17.64
5	StDevInnerRuntime	40.22	11.48	2.73	2.95	2.28	7.52	1.07	1.16	1.74	1.84
6	AvOptMaxIdx	21.76	32.28	41.60	41.48	49.84	55.60	64.92	64.12	59.00	43.60
7	OptMaxToAve	2.84	1.99	1.66	2.13	1.69	4.44	2.50	2.26	2.09	1.91
8	MeanOuterRuntime	95.68	50.20	36.56	37.72	35.13	30.99	21.63	20.68	22.97	26.14
Panel C $n = 750$ $pts = 100$											
1	K	3	5	10	15	20	30	40	70	100	150
2	U	0.50	0.40	0.30	0.25	0.20	0.15	0.10	0.07	0.04	0.03
3	L	0.10	0.08	0.05	0.04	0.03	0.02	0.01	0.005	0.004	0.003
4	MeanInnerRuntime	382.05	88.74	66.51	70.08	60.13	48.07	29.82	26.21	31.39	37.09
5	StDevInnerRuntime	983.86	17.40	7.19	5.91	7.18	5.33	2.30	2.19	3.07	2.93
6	AvOptMaxIdx	14.92	18.40	31.08	43.04	52.28	75.68	63.32	66.60	61.80	60.36
7	OptMaxToAve	5.44	4.51	1.73	1.62	1.77	2.11	2.68	2.52	2.31	2.15
8	MeanOuterRuntime	400.56	101.28	78.53	82.21	72.00	59.92	41.43	37.92	39.16	49.00
Panel D $n = 1000$ $pts = 100$											
1	K	3	5	10	15	20	30	40	70	100	150
2	U	0.50	0.40	0.30	0.25	0.20	0.15	0.10	0.07	0.04	0.03
3	L	0.10	0.08	0.05	0.04	0.03	0.02	0.01	0.005	0.004	0.003
4	MeanInnerRuntime	474.82	187.13	120.16	117.71	101.09	77.38	45.28	43.58	50.70	60.49
5	StDevInnerRuntime	266.11	78.05	10.86	10.86	8.24	6.54	3.53	2.73	4.02	4.21
6	AvOptMaxIdx	21.72	22.92	24.28	35.68	47.20	66.32	67.16	51.52	53.88	51.96
7	OptMaxToAve	4.74	3.07	1.66	1.68	1.83	2.21	3.17	2.43	2.30	2.32
8	MeanOuterRuntime	504.94	204.89	152.64	149.36	133.24	109.31	76.64	74.99	83.74	93.44

Whether looking at Inner or Outer run times, the times facilitated by the approach for whole cardinality constrained efficient frontiers are remarkably small, being in the single digits for most of Panel A, in the double digits for almost all of Panels B and C, and only in the triple digits in Panel C for $K \leq 5$ and in panel D for $K \leq 30$. Illustrating, mean Inner run time for $n = 250$ when $K = 100$ is 3.74 seconds, mean Inner run time for $n = 500$ when $K = 40$ is 13.21 seconds, and mean Inner run time for $n = 1000$ when $K = 5$ is 187.13 seconds. As for cardinality constrained efficient frontier run times being at their longest, say when $K \leq 30$, this is consistent with what the progenitors of the 70+ heuristics have reported, that is, that the longest run times are with the smallest values of K. However, even at their longest, none of these run times are long at all.

Note how Inner and Outer run times decrease with increased cardinality until they bottom out for all problem sizes at $K = 70$, after which they begin a gradual rise. For instance, mean Outer run time when $n = 1000$ bottoms out at $K = 70$ at 74.99 seconds. It seems logical that this would be due to the portfolios making up the continuous-variable efficient frontiers of the problems having cardinalities that place the least amount of stress on the cardinality constraint in (CC) when set to 70. But since there could be other explanations, this is something that needs to be researched further before more is said.

With regard to rows 6 and 7 in the panels, consider the optimizations that have the longest Inner run times out of the 100 of each cardinality constrained efficient frontier. AvOptMaxIdx is then the average of the indices of these optimizations over the 25 problems in the sample for each (n, K, U, L) combination in the panels. OptMaxToAve is then for each (n, K, U, L) combination the average of the ratios of a problem's longest Inner run time to the mean of the Inner run times of the problem's other 99 optimizations. We include these two rows because of a comment in Bertsimas & Cory-Wright (2022) about the optimizations forming the lower

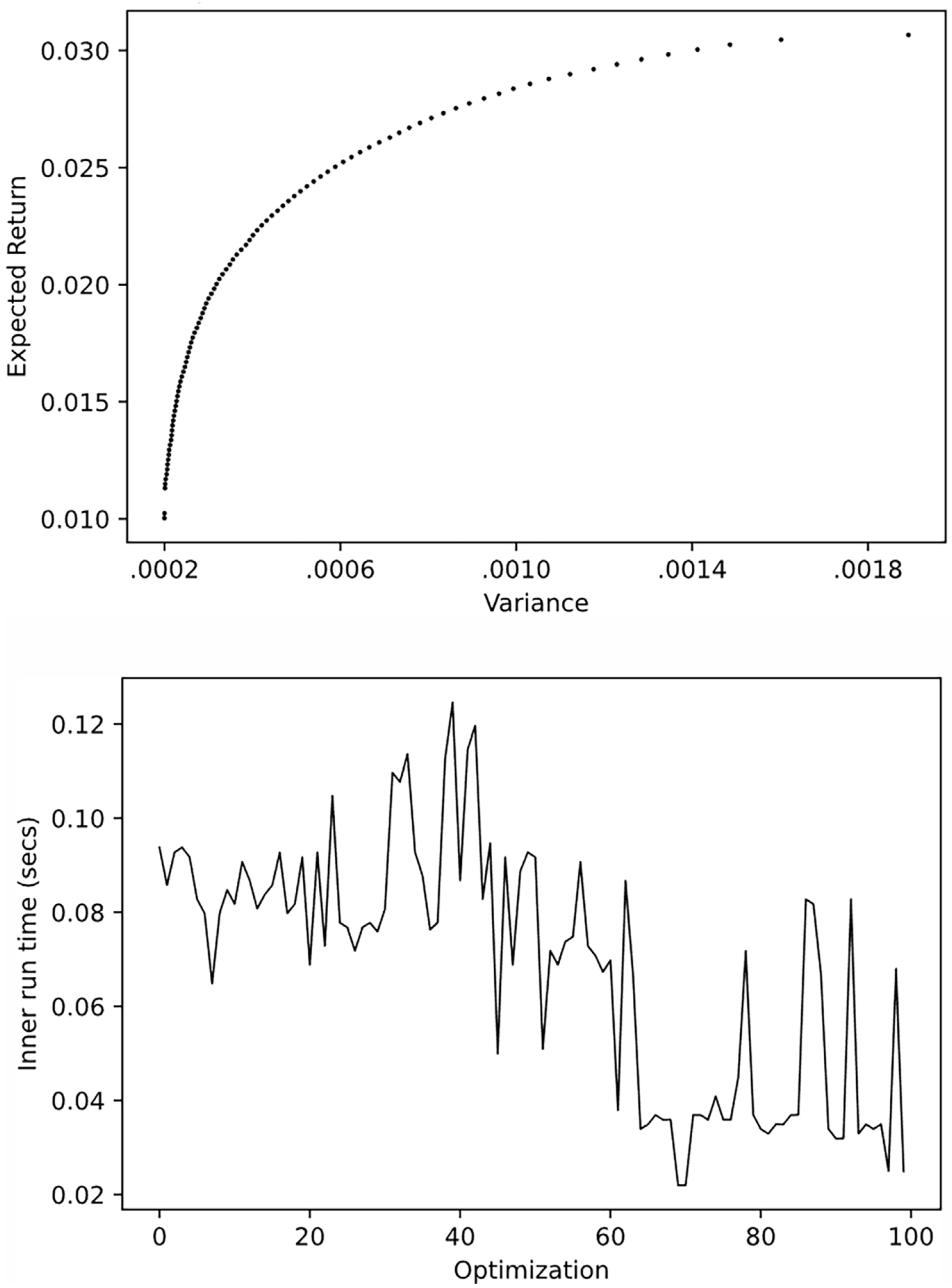

Fig. 1. Using the 100 equally-spaced ρ-values from the problem's interval, at the top (a) is the $K = 10$, 100-point cardinality constrained efficient frontier of a 250-security problem, and at the bottom (b) is a plot of the Inner run times of the 100 optimizations used to construct it.

part of the efficient frontier likely taking the longest. While we do not see this across the board, this effect, however, is observed in all four of our panels for $K \leq 20$.

To illustrate, let us first consider Fig. 1(a). In this figure we have the $K = 10$, 100-point cardinality constrained efficient frontier of a 250-security problem whose $U = 0.30$ and $L = 0.05$. In problems of the sizes of this paper, it is to be noted that their cardinality constrained efficient frontiers, apart from a little granular roughness here and there and perhaps a small gap in the near vertical lower part of the frontier, follow very closely the sweeping nature of the efficient frontiers that we have long been familiar with in continuous-variable portfolio optimization. This is in contrast to the irregularly shaped cardinality constrained efficient frontiers shown, for example, in Chang et al. (2000), Jobst, Horniman, Lucas, & Mitra (2001), and Woodside-Oriakhi et al. (2011) which can only occur when both K and n are small. But as soon as K and n take on larger values as they would in more realistic situations, their cardinality constrained efficient frontiers quickly morph into shapes with which we are familiar with all but small traces of the irregularities seen in small problems long left behind.

In Fig. 1(b) we have the Inner run times of the 100 optimizations used to construct the frontier. In the plot the greatest Inner run time is 1.837 times the average time of the other 99 optimizations. With the greatest run time occurring on optimization 39, this is consistent with the AvOptMaxIdx entry in the $K = 10$ column of Panel A in Table 7. However, in our experiments, the phenomenon of longest optimization run times gradually shifts to the upper portions of the efficient frontiers as K increases as tabulated in row 6 of all panels.

5. Conclusions

What we have been able to develop is a simple, straightforward, and easy to implement approach that enables one to solve cardinality constrained mean-variance portfolio selection problems and deliver useable portfolio selection solutions to problems with up to 1000 securities. Our finding is that it is not the *NP*-hard nature of (CC), but the covariance matrix, that is the stumbling block. Fortunately, troubles with the covariance matrix can be taken care of by making only minute adjustments to it. With the adjustments to the covariance matrix having virtually no effect on solution accuracy, the complexities thought to be involved in solving cardinality constrained portfolio problems are avoided. In this way, with the cardinality constrained portfolio problem now so easily solved, efficient frontiers consisting of many points representing them can be computed in problems with up to 1000 securities in times not believed possible before.

In summary, the approach paves ground for a new era in portfolio selection. On Wall Street, there is seldom a problem without buyin thresholds and cardinality targets. Instead of having to compromise with continuous variables, such problems can now be solved in a clear, concise and rapid fashion.

References

Anagnostopoulos, K. P., & Mamanis, G. (2011). Multiobjective evolutionary algorithms for complex portfolio optimization problems. *Computational Management Science, 8*, 259–279.

Baykasoğlu, A., Yunusoglu, M. G., & Özsoydan, F. B. (2015). A GRASP based solution approach to solve cardinality constrained portfolio optimizations problems. *Computers & Industrial Engineering, 74*, 339–351.

Beasley, J. E. (2000). Distributing test problems by electronic mail. *Journal of the Operational Research Society, 41*, 1069–1072.

Beattie, A., Brown, J. R., & Kvilhaug, S. (2022). *Understanding the history of the modern portfolio*. Investopedia. https://www.investopedia.com.

Bertsimas, D., & Cory-Wright, R. (2022). A scalable algorithm for sparse portfolio selection. *INFORMS Journal on Computing, 34*, 1489–1511.

Bertsimas, D., & Shioda, R. (2009). Algorithm for cardinality-constrained quadratic optimization. *Computational Optimization and Applications, 43*, 1–22.

Bienstock, D. (1996). Computational study of a family of mixed-integer quadratic programming problems. *Mathematical Programming, 74*, 121–140.

Branke, J., Scheckenbach, B., Stein, M., Deb, K., & Schmeck, H. (2009). Portfolio optimization and an envelope-based multi-objective evolutionary algorithm. *European Journal of Operational Research, 199*, 684–693.

Cesarone, F., Scozzari, A., & Tardella, F. (2013). A new method for mean-variance portfolio optimization with cardinality constraints. *Annals of Operations Research, 205*, 213234.

Cesarone, F., Scozzari, A., & Tardella, F. (2015). Linear vs. quadratic portfolio selection models with hard real-world constraints. *Computational Management Science, 12*, 345–370.

Chang, T.-J., Meade, N., Beasley, J. E., & Sharaiha, Y. M. (2000). Heuristics for cardinally constrained portfolio optimisation. *Computers & Operations Research, 27*, 1271–1302.

Cifuentes, A. (2022). *Timeless relevance of Markowitz's ideas for finance and investing*. The Financial Times. https://www.ft.com.

Cui, X. T., Zheng, X. J., Zhu, S. S., & Sun, X. L. (2012). Convex relaxations and MIQCQP reformulations for a class of cardinality-constrained portfolio selection problems. *Journal of Global Optimization, 56*, 1409–1423.

Cura, T. (2011). Particle swarm optimization approach to portfolio optimization. *Nonlinear Analysis: Real World Applications*, 2396–2406.

Deng, G.-F., & Lin, W.-T. (2010). Ant colony optimization for Markowitz mean-variance portfolio model. *Lecture Notes in Computer Science, 6466*, 238–245.

Deng, G.-F., Lin, W.-T., & Lo, C.-C. (2012). Markowitz-based portfolio selection with cardinality constraints using improved particle swarm optimization. *Expert Systems with Applications, 39*, 4558–4566.

Ehrgott, M., Klamroth, K., & Schwehm, C. (2004). An MCDM approach to portfolio optimization. *European Journal of Operational Research, 155*, 752–770.

Fernández, A., & Gómez, S. (2007). Portfolio selection using neural networks. *Computers & Operations Research, 34*, 1177–1191.

Gao, J., & Li, D. (2013). Optimal cardinality constrained portfolio selection. *Operations Research, 61*, 745–761.

Graham, B., & Dodd, D. L. (1934). *Security analysis*. New York: McGraw Hill Book Co..

Higham, N. J. (1988). Computing a nearest symmetric positive semidefinite matrix. *Linear Algebra and its Applications, 103*, 103–118.

Higham, N. J. (2002). Computing the nearest correlation matrix - a problem from finance. *IMA Journal of Numerical Analysis, 22*, 329–343.

Hirschberger, M., Qi, Y., & Steuer, R. E. (2007). Randomly generating portfolio-selection covariance matrices with specified distributional characteristics. *European Journal of Operational Research, 177*, 1610–1625.

Hirschberger, M., Qi, Y., & Steuer, R. E. (2010). Large-scale MV efficient frontier computation via a procedure of parametric quadratic programming. *European Journal of Operational Research, 204*, 581–588.

Jobst, N. B., Horniman, M. D., Lucas, C. A., & Mitra, G. (2001). Computational aspects of alternative portfolio selection models in the presence of discrete asset choice constraints. *Quantitative Finance, 1*, 1–13.

Juszczuk, P., Kaliszewski, I., Miroforidis, J., & Podkopaev, D. (2022). Mean return-standard deviation efficient frontier approximation with low-cardinality portfolios in the presence of the risk-free asset. *International Transactions in Operational Research*. https://doi.org/10.1111/itor.13121.

Kalayci, C., Ertenlice, O., Akyer, H., & Aygoren, H. (2017). An artificial bee colony algorithm with feasibility enforcement and infeasibility toleration procedures for cardinality constrained portfolio optimization. *Expert Systems with Applications, 85*, 61–75.

Kolm, P. N., Tütüncü, R., & Fabozzi, F. J. (2014). 60 Years of portfolio optimization: Practical challenges and current trends. *European Journal of Operational Research, 234*, 356–371.

Liagkouras, K., & Metaxiotis, K. (2018). Handling the complexities of the multi–constrained portfolio optimization problem with the support of a novel MOEA. *Journal of the Operational Research Society, 69*, 1607–1627.

Liagkouras, K., & Metaxiotis, K. (2018). A new efficiently encoded multiobjective algorithm for the solution of the cardinality constrained portfolio optimization problem. *Annals of Operations Research, 267*, 281–319.

Lin, C. C., & Liu, Y. T. (2008). Genetic algorithms for portfolio selection problems with minimum transaction lots. *European Journal of Operational Research, 185*, 393–404.

Lwin, K., Qu, R., & Kendall, G. (2014). A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization. *Applied Soft Computing Journal, 24*, 757–772.

Mansini, R., & Speranza, M. G. (1999). Heuristic algorithms for the portfolio selection problem with minimum transaction lots. *European Journal of Operational Research, 114*, 219–233.

Markowitz, H. (1952). Portfolio selection. *Journal of Finance, 7*, 77–91.

Markowitz, H. M. (1956). The optimization of a quadratic function subject to linear constraints. *Naval Research Logistics Quarterly, 3*, 111–133.

Qi, H., & Sun, D. (2006). A quadratically convergent Newton method for computing the nearest correlation matrix. *SIAM Journal on Matrix Analysis and Applications, 28*, 360–385.

Ruiz-Torrubiano, R., & Suárez, A. (2010). Hybrid approaches and dimensionality reduction for portfolio selection problems with cardinality constraints. *IEEE Computational Intelligence Magazine*. https://doi.org/10.1109/MCI.2010.936308.

Sadjadi, S. J., Gharakhani, M., & Safari, E. (2012). Robust optimization framework for cardinality constrained portfolio problem. *Applied Soft Computing Journal, 12*, 91–99.

Salehpoor, I. B., & Molla-Alizadeh-Zavardehi, S. (2019). A constrained portfolio selection model at considering risk-adjusted measure by using hybrid meta-heuristic algorithms. *Applied Soft Computing Journal, 75*, 233–253.

Shaw, D. X., Liu, S., & Kopman, L. (2008). Lagrangian relaxation procedure for cardinality-constrained portfolio optimization. *Optimization Methods and Software, 23*, 411–420.

Steuer, R. E., Hirschberger, M., & Deb, K. (2015). Extracting from the relaxed for large-scale semi-continuous variable nondominated frontiers. *Journal of Global Optimization, 64*, 33–48.

Steuer, R. E., & Utz, S. (2022). Non-contour efficient fronts for identifying most preferred portfolios in sustainability investing. *European journal of operational research, 306*, 742–753.

Woodside-Oriakhi, M., Lucas, C., & Beasley, J. E. (2011). Heuristic algorithms for the cardinality constrained efficient frontier. *European journal of operational research, 213*, 538–550.