Decision Support Systems 51 (2011) 250–255

Contents lists available at ScienceDirect

Decision Support Systems

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

Comparative issues in large-scale mean–variance efficient frontier computation

Ralph E. Steuer [a,*], Yue Qi [b], Markus Hirschberger [c]

[a] *Terry College of Business, University of Georgia, Athens, GA 30602-6253, USA*
[b] *Department of Financial Management, College of Business, Nankai University, Tianjin, China*
[c] *Department of Mathematics, University of Eichstätt-Ingolstadt, Eichstätt, Germany*

ARTICLE INFO

Available online 25 November 2010

Keywords:
Mean–variance efficient frontiers
Portfolio selection
Hyperbolic segments
e-Constraint method
Parametric quadratic programming

ABSTRACT

One of the functions of a portfolio management system is to return quickly an efficient frontier. However, in the large-scale problems (1000 to 3000 securities) that are beginning to appear with greater frequency, the task of computing the mean–variance efficient frontier, even when all constraints are linear, can range from the significant to the prohibitive. For ease of reference, we call mean–variance problems with all linear constraints Markowitz problems. With little on the time to compute a Markowitz-problem efficient frontier in the literature, we conduct experiments that involve varying problem sizes, methods employed, and optimizers used to present an overall picture of the situation and establish benchmarks in the large-scale arena. One of the conclusions of the experiments is the superiority of the class of techniques that would fall under the title of parametric quadratic programming.

1. Introduction

Large institutional investors such as mutual funds and pension funds use portfolio management systems (also called portfolio selection systems, portfolio construction systems, asset allocation systems, and so forth) to assist in their asset allocations. Some are leased on a turnkey basis from vendors and others are built in-house. For related readings, see Zopounidis and Doumpos [29], Maginn et al. [11], and Xidonas et al. [27]. Typically, a system consists of several modules including a database management module, an input coefficient calculation module, and an optimization module. Our interest is in the optimizers in the optimization modules that compute and re-compute the efficient frontier any time the model undergoing study changes.

Generally, when using a system to achieve an asset allocation, considerable experimentation takes place as the user experiments with various constraint sets and pools of eligible securities in search of a portfolio that is both suitable and meets the criteria of the project. While in the beginning a user may even wish to test a few unorthodox ideas just to see what might result, in the end the experimentation generally tapers down to last-minute perturbations to confirm, re-confirm, and perfect a final solution. In this way, the optimizers in the optimization modules are often called upon to compute efficient frontier after efficient frontier. In order to not slow down a user, a rapid-fire response capability is an important attribute for a system to possess. On small problems, this is usually not an issue as the time to compute an efficient frontier is not of such a length that it makes much difference what optimizer is used. But on large problems (the domain of this paper), where turnaround times can become serious and inhibiting, the choice of optimizer can make a major difference. Consequently, the best way to keep turnaround times down in such cases is to know the situation and then try to manage it. Benchmarks would be very helpful.

Unfortunately, when it comes to benchmarks for the time required to compute an efficient frontier, the literature has been essentially a vacant area. This is despite the fact that the concept of an efficient frontier has been around since Markowitz [12] and that large-scale problems are now appearing with greater frequency. With data now widely available on securities from all over the world and with investors often open to the consideration of increased numbers of opportunities to produce added value for their portfolio, problems in the range of 1000 to 3000 securities no longer raise eyebrows.

Because of (a) the growing importance of large-scale problems (here defined to be in the range from 1000 to 3000 securities), (b) a lack of benchmarks, and (c) the time that could be wasted when attempting to compute an efficient frontier with a wrong optimizer, we have this paper. A sad thing about using a wrong optimizer in the absence of comparative computational experience is that a person might conclude that a given efficient frontier cannot be computed when in fact with a different optimizer the same frontier could be computed perhaps in reasonable time.

Thus, in this paper the attempt is to fill the gap and provide a frame of reference for those who find themselves confronting problems in large-scale portfolio selection. By conducting experiments on the times required to compute efficient frontiers across problems of different (large-scale) sizes using different methods and different optimizers, the goal is for a robust set of benchmarks to emerge. This is

[*] Corresponding author. Tel.: +1 706 542 3782; fax: +1 706 542 9434.
E-mail address: rsteuer@uga.edu (R.E. Steuer).

0167-9236/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.dss.2010.11.018

a contribution to the literature as no such set of benchmarks is known to exist. The optimizers used in the experiments are Risk Solver Platform for Excel [23], Matlab [17], LINGO [10], Cplex [3], and the research code CIOS by Hirschberger, Qi and Steuer [7].

Despite recent modeling innovations such as by Ben Abdelaziz et al. [1], Ehrgott et al. [4], Fang and Wang [5], Michalowski and Ogryczak [18], Xie et al. [28], Stummer et al. [25], Tseng et al. [26], and Köksalan and Tuncer [9], the mean–variance model with all linear constraints introduced by Markowitz in (1952) and studied further in his books [14–16] is still the best known model in portfolio selection. Furthermore, it is a starting point for many other models. For ease of reference, we henceforth refer to all mean–variance problems with all linear constraints as Markowitz problems. Because Markowitz problems are the standard bearer in portfolio selection, and because most portfolio problems possess dense covariance matrices in their natural state (before simplifying assumptions are applied to ease the computational burden), we will use the times experienced to compute the efficient frontiers of 100% dense covariance matrix problems to form the benchmark results reported in this paper.

The remainder of the paper is as follows. Section 2 discusses ways efficient frontiers can be displayed. Section 3 provides detail on the piecewise hyperbolic structure of the efficient frontiers computed in this paper. Section 4 presents comparative computational results across different problem sizes, methods, and optimizers. Section 5 closes the paper with concluding remarks.

2. Efficient frontier and its display

A typically-appearing mean–variance efficient frontier is as in Fig. 1. With standard deviation on the horizontal axis, one might wonder why we hear so often the term "mean–variance" in portfolio selection. It is because theory and computation are carried out in terms of variance, but results are displayed in terms of standard deviation (as standard deviation is more intuitive than variance).

Although the mean–variance efficient frontier of a Markowitz problem (hereafter simply called the efficient frontier) is a continuous curve, it is rarely rendered as such. Rather, efficient frontiers are customarily shown in the form of dotted representations as in Fig. 1. One can put aside the 10- to 20-dot representations seen in textbooks. In practice, a hundred or more dots are often standard. In this way, with so many dots, dotted representations nearly look like the continuous curves they are to represent. The difficulty usually is the amount of optimization that it takes to generate a representation.

A nice way for a user to interact with a dotted efficient frontier is with a mouse and cross hairs. By moving the cross hairs up and down a representation, different dots can be clicked to display their expected returns, standard deviations, and portfolio compositions. If the efficient frontier can indeed be displayed as the continuous curve that it is, then the cross hairs can move smoothly along the frontier and any point can be clicked. Otherwise, the cross hairs are only able to jump from point to point with only the given points available for clicking.

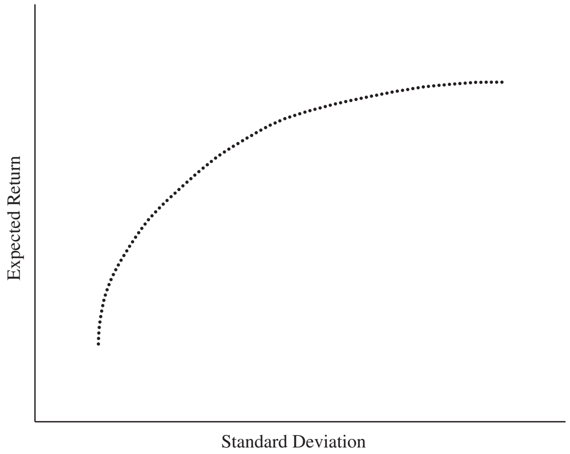

Fig. 1. A 100-dot representation of a typical Markowitz problem efficient frontier.

3. Problem and its piecewise hyperbolic frontier

In this section we describe the formulation of the problem used in this study along with the piecewise hyperbolic structure of its efficient frontier. With a Markowitz problem attempting to minimize variance and maximize expected return simultaneously, the problem in bi-criterion format is

$$\mathit{min}\ \mathbf{x}^T\Sigma\mathbf{x} \quad \text{variance} \tag{1.1}$$

$$\mathit{max}\ \mu^T\mathbf{x} \quad \text{expected return} \tag{1.2}$$

$$s.t.\ 1^T\mathbf{x} = 1 \tag{1.3}$$

$$\alpha \le \mathbf{x} \le \omega \tag{1.4}$$

where $\mathbf{x}\in\mathbb{R}^n$, n is the number of securities eligible for inclusion in a portfolio, and the x_i components of $\mathbf{x}$ are the proportions of capital to be allocated initially to security i. Also, Σ is the problem's $n\times n$ covariance matrix and μ is the vector of individual security expected returns. Constraint (1.3) assures that all x_i proportions sum to one and Eq. (1.4) is intended as a holding area for additional linear constraints (here only lower and upper bounds on the x_i).

Possessing more than one criterion, the problem has two feasible regions, $S\subset\mathbb{R}^n$ and $Z\subset\mathbb{R}^2$. S is the set of all feasible portfolios where $\mathbf{x}\in\mathbb{R}^n$ is a *portfolio* if it satisfies Eq. (1.3) and is *feasible* if it also satisfies Eq. (1.4). Z is the set of all image vectors (in terms of standard deviation and expected return) of the portfolios in S.

Now $\mathbf{x}\in S$ is an *efficient* portfolio if with respect to its image vector there is no image vector a member of Z with a lower standard deviation without a lower expected return or a greater expected return without a greater standard deviation. Then the set of all image vectors of all efficient portfolios forms the so-called efficient frontier.

A property of a Markowitz problem is that the image set of any linear line segment in S forms a hyperbolic line segment in Z. Then, considering all possible linear line segments in S, we can see why the efficient frontier, being a portion of the boundary of Z, is piecewise hyperbolic.

To illustrate, consider the three-security example whose data are in Table 1. With lower and upper bounds of 0 and 1 on all x_i, its S and Z are in Fig. 2. In this example, the efficient frontier consists of three hyperbolic segments. To see why, let us first look at the curves (partly dashed) connecting $\mathbf{z}^1$, $\mathbf{z}^2$ and $\mathbf{z}^2$, $\mathbf{z}^3$ in Z. As their inverse image sets are the straight lines connecting $\mathbf{x}^1$, $\mathbf{x}^2$ and $\mathbf{x}^2$, $\mathbf{x}^3$ in S, the two curves are hyperbolic. Hence, boundary segments $\mathbf{z}^5$, $\mathbf{z}^6$ and $\mathbf{z}^3$, $\mathbf{z}^4$, which correspond to $\mathbf{x}^5$, $\mathbf{x}^6$ and $\mathbf{x}^3$, $\mathbf{x}^4$, are hyperbolic. By the same token, boundary segment $\mathbf{z}^4$, $\mathbf{z}^5$ is hyperbolic as its inverse image set is $\mathbf{x}^4$, $\mathbf{x}^5$. In this way, efficient frontiers are piecewise hyperbolic and the (efficient) portfolios that generate them lie along piecewise linear paths in S. Markowitz calls the points in S that correspond to the

Table 1
Data defining the three-security illustrative example where σ is the vector of individual security standard deviations.

	1	2	3
σ	0.08	0.07	0.10
μ	0.01	0.08	0.10
Σ	0.0064	−0.0010	0.0040
		0.0049	0.0030
			0.0100

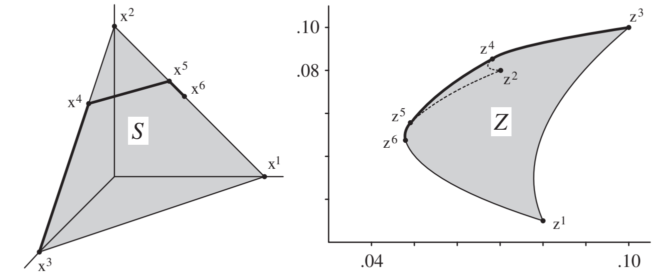

Fig. 2. Feasible regions S and Z of the three-security illustrative example along with the piecewise linear path in S that produces the piecewise hyperbolic efficient frontier in Z.

endpoints of the hyperbolic segments in Z that constitute the efficient frontier, such as $\mathbf{x}^3$, $\mathbf{x}^4$, $\mathbf{x}^5$, $\mathbf{x}^6$, *corner* portfolios.

For a method to be able to generate an exact specification of an efficient frontier, it must be able to generate information as in Tables 2 and 3. Consider row h of Table 2. The a_0, a_1, a_2 entries in that row are parameters that define the hyperbola upon which the h-th efficient frontier hyperbolic segment lies, and the μ^{lower} and μ^{upper} entries in that row specify the expected returns of the lower and upper endpoints of the hyperbolic segment, respectively. Rows h and $h+1$ of Table 3 specify the compositions of the (corner) portfolios corresponding to the endpoints of the hyperbolic segment.

As an example of how the tables can be deployed, suppose we are curious about a point $\mathbf{z}^*$ on the efficient frontier whose $\mu^*=.07$. According to Table 2, $\mathbf{z}^*$ is on the $\mathbf{z}^4$, $\mathbf{z}^5$ hyperbolic segment. To calculate $\mathbf{z}^*$'s standard deviation σ^*, we use the a_0, a_1, a_2 of the segment as follows

$$\begin{aligned}\sigma^* &= \sqrt{a_0 + a_1\mu^* + a_2(\mu^*)^2}\\ &= \sqrt{.00342-.0760(.07) + 1.0368(.07)^2}\\ &= .05639.\end{aligned}$$

To compute the composition of the (efficient) portfolio $\mathbf{x}^*$ whose image vector is $\mathbf{z}^*$, we use the μ^{lower} and μ^{upper} of the segment as follows

$$\begin{aligned}\mathbf{x}^* &= \frac{\mu^*-\mu^{lower}}{\mu^{upper}-\mu^{lower}}\mathbf{x}^4 + \frac{\mu^{upper}-\mu^*}{\mu^{upper}-\mu^{lower}}\mathbf{x}^5\\ &= \frac{.07-.05654}{.08660-.05654}\mathbf{x}^4 + \frac{.08660-.07}{.08660-.05654}\mathbf{x}^5\\ &= (.18506, .66725, .14769).\end{aligned}$$

4. Computational experiments and their results

In this section we describe and present the results of the computational experiments of the paper. The experiments are first categorized by approach, discrete or parametric. In the discrete category for generating dotted representations of the efficient frontier as in Fig. 1, two methods are treated (e-constraint method and λ-parameter method). In the parametric category for computing an exact specification of the efficient frontier as in Section 3, there are only parametric quadratic programming variants. As to be seen, there are vast differences in time among optimizers in the discrete category, and also between the discrete and parametric categories. In all experiments, the computer used was a 2.66 GHz Core 2 Duo Dell desktop at the University of Georgia.

Table 2
Expected return ranges of the hyperbolic segments of the efficient frontier along with parameters that specify the hyperbolas of the hyperbolic segments for the three-security illustrative example.

Hyperbolic segment	μ^{lower}	μ^{upper}	a_0	a_1	a_2
$\mathbf{z}^3$, $\mathbf{z}^4$	0.08660	0.10000	0.16250	−3.7500	22.250
$\mathbf{z}^4$, $\mathbf{z}^5$	0.05654	0.08660	0.00342	−0.0760	1.0368
$\mathbf{z}^5$, $\mathbf{z}^6$	0.04595	0.05654	0.00879	−0.2657	2.7143

Table 3
Compositions of the corner portfolios of the three-security illustrative example.

Corner portfolio	x_1	x_2	x_3
$\mathbf{x}^3$	0.0	0.0	1.0
$\mathbf{x}^4$	0.0	0.67017	0.32983
$\mathbf{x}^5$	0.33511	0.66489	0.0
$\mathbf{x}^6$	0.44361	0.55639	0.0

4.1. Discrete approaches: e-constraint method

Leading the list of discrete approaches is the “e-constraint” method. Let ρ_{min} and ρ_{max} denote the minimum and maximum expected return values over the efficient frontier. With regard to the construction of a discretized representation consisting of q dots, q values, denoted ρ_i, are selected from the interval $[\rho_{min},\rho_{max}]$ where $\rho_1:=\rho_{min}$, and $\rho_q:=\rho_{max}$. Often the ρ_i are chosen to be equally spaced. Then the e-constraint method solves the quadratic programming (QP) problem

$$\begin{aligned}&min\, \mathbf{x}^T\Sigma\mathbf{x}\\ &s.t.\quad \mu^T\mathbf{x}=\rho \quad \rho\in\left\{\rho_{min},\rho_2,\ldots,\rho_{q-1},\rho_{max}\right\}\\ &\mathbf{x}\in S\end{aligned} \tag{2}$$

once for each ρ-value in the list. By repetitively minimizing variance subject to different fixed levels of expected return, it is not necessary for one to be a mathematical expert to appreciate the construction of an efficient frontier, and this accounts for much of the e-constraint method's appeal. The fact that Eq. (2) only requires standard QP software (in plentiful supply) accounts for most of the rest. We use the term “e-constraint” because this is the term given to the method in multiple criteria optimization for computing efficient points after all objectives have been converted to constraints except one (see Miettinen[20]).

Suppose we were to compute a 50-dot representation of an efficient frontier using the e-constraint method. This would normally involve 51 optimizations: (a) a linear programming (LP) optimization to obtain the maximum expected return value ρ_{max} over the efficient frontier, (b) a QP optimization to make sure that the $\mathbf{z}$-vector of ρ_{max} with minimum variance is obtained, (c) a QP optimization to obtain a $\mathbf{z}$-vector of minimum variance from which ρ_{min} is to be extracted, and (d) 48 other QP optimizations to obtain the 48 “intermediate” points of the 50-dot representation. While multiple $\mathbf{z}$-vectors of maximum expected return will sometimes occur, users usually do not worry about multiple $\mathbf{z}$-vectors of minimum variance.[1] Since LP times are so small in relation to the QP times, we only track QP times in the experiments.

Table 4 shows the amount of time required on average to compute a single intermediate point along the efficient frontier using the e-constraint method (i.e., to solve one instance of the e-constraint QP), tabulated by problem size and optimizer employed. The four optimizers were selected because they are well known and have e-constraint examples in their illustrative materials. In particular, Risk Solver Platform (version 9.5) was

[1] In the event there is cause to worry, a second-order cone solver, to maximize expected return subject to an upper bound on variance (where the upper bound is the minimum variance obtained in (c)), is a tool that can be considered for resolving the situation.

Table 4
Average run times (sample size 5 in each case) for computing single intermediate points on the efficient frontiers of 100% dense covariance matrix Markowitz problems using the e-constraint method tabulated by problem size and optimizer.

Size	e-Constraint single intermediate point times			
	Risk solver	Matlab	LINGO	Cplex
$n=500$	8.3 s	51.7 s	34.9 s	2.4 s
$n=1000$	86.0 s	910.2 s	–	6.0 s
$n=1500$	313.7 s	–	–	22.2 s
$n=2000$	513.0 s	–	–	38.6 s
$n=3000$	–	–	–	118.2 s

chosen because it is the commercial edition of regular Solver that comes free with Excel. By the way, regular Solver is size limited to at most 200 variables in QP. Matlab (version 2009a) was selected because of the popularity of its financial toolbox in the financial services industry. LINGO (version 11) was chosen because of the degree to which it is well liked in academic circles. Cplex (default choice of version 11.1) was chosen because of the toughness of the standard that it presents. Since the times in Table 4 are for single points on the efficient frontier, they must be multiplied to take into account the number of points utilized to form a given discretized representation. Other than for the LINGO column, the blank cells are from when it is clear, without running, that the results would exceed 500 s, figuring that, with this much time consumption per point, it makes no sense to proceed.

As seen, there is considerable variation. Although Risk Solver Platform and default Cplex start out well at 500 securities, Cplex significantly widens its advantage until Cplex is over an order of magnitude faster than Risk Solver Platform after 1500 securities. On the other hand, Matlab's QUADPROG routine and LINGO's barrier algorithm are disappointments. The reason for the blank cells in the LINGO column is that we could not get LINGO to complete a run with more than 650 securities (perhaps Microsoft's fault when passing data) in any of the problems we tried with a dense covariance matrix.

All problems used in the experiments were created using the random covariance matrix generator described in Hirschberger, Qi and Steuer [6]. For data mining and potential utilization in finance, see Peng et al. [22].

4.2. Discrete approaches: λ-parameter method

Another way to obtain a dotted representation of an efficient frontier is to employ the λ-parameter method. In this method we solve the following QP

$$\begin{aligned} &\min\ \mathbf{x}^T\Sigma\mathbf{x}-\lambda\mu^T\mathbf{x} \qquad \lambda\in\left\{0,\lambda_2,\ldots,\lambda_{q-1},\lambda_{max}\right\} \\ &\mathbf{x}\in S \end{aligned} \tag{3}$$

once for each λ-value in the list. The greater the value of λ, the greater, if anything, the variance (and hence the expected return) of the point generated on the efficient frontier. Although the λ-parameter method only requires standard QP software as with the e-constraint method, this method, however, is more difficult to work with than the e-constraint method.

When solving Eq. (3) with $\lambda=0$, we solve for a minimum variance point. With regard to λ, this is clear. But when attempting to solve for the maximum expected return point on the efficient frontier we run into frustrations. This is because there is no simple way of knowing λ_{max} in advance (where λ_{max} is the smallest value of λ that uniquely computes the maximum expected return point on the efficient frontier). In some cases λ_{max} could be near infinity. In other cases it could be a number less than one. We have no way of knowing from the outside. If we guess too high when selecting values for λ, the last few λ-values may result in the same point. If we guess too low, we will not compute high enough up the efficient frontier. In the first case we run the risk of generating fewer points than desired. In the second case we fail to capture the whole frontier. Either way, there can be problems.

The λ-parameter method can also be disconcerting with regard to the spacing of the points generated along the efficient frontier. At least with the e-constraint method we know in advance the expected return component of each generated point, but with the λ-parameter method, we don't know either component until after the optimization is over.

As for single intermediate point run times with the λ-parameter method, we have Table 5. For Risk Solver Platform and Cplex, the times follow roughly the same patterns as in Table 4, but are about 30% faster. Matlab and LINGO, however, are little changed. Although not as intuitive as the e-constraint method, with the right optimizer, the method offers the opportunity to be faster than the e-constraint method if one can successfully deal with the likely irregular spacing of the points generated.

4.3. Parametric approaches: quadratic parametric programming

In a portfolio selection, the purpose of a parametric procedure is to compute the full continuous curve of the efficient frontier. In a parametric procedure, the problem is modeled with a parameter so that by smoothly varying the parameter the whole efficient frontier is exactly traced out. The first algorithm for doing this was the critical line method of Markowitz [13]. The critical line method, a parametric quadratic programming variant, views portfolio selection through the prism of the following formulation

$$\begin{aligned} &\min\ \mathbf{x}^T\Sigma\mathbf{x} \\ &s.t. \quad \mu^T\mathbf{x}=\rho \quad \rho\in[\rho_{min},\rho_{max}] \\ &\mathbf{x}\in S \end{aligned} \tag{4}$$

where ρ is to be continuously varied over the interval $[\rho_{min},\rho_{max}]$ of expected return values associated with the efficient frontier. Note that the e-constraint method can be seen as a discretized implementation of Eq. (4).

Even though Markowitz's other ideas have been of major success, his critical line method, called one of the “enigmas” of modern finance by Michaud [19], has not been widely adopted. In fact, other than for a few algorithmic pieces such as by Best [2], Stein et al. [24], Niedermayer and Niedermayer [21], and the parametric quadratic programming approach based upon the following formulation

$$\begin{aligned} &\min\ \mathbf{x}^T\Sigma\mathbf{x}-\lambda\mu^T\mathbf{x} \qquad \lambda\in[0,+\infty) \\ &\mathbf{x}\in S \end{aligned} \tag{5}$$

by Hirschberger et al. [7], little else has been available for citation on parametric approaches for efficient frontier computation over about the last four decades. Note that the λ-parameter method can be seen as a discretized implementation of Eq. (5).

Reasons for the above mentioned sparseness in the literature stem from the fact that (i) the critical line method in particular is not easy to

Table 5
Average run times (sample size 5 in each case) for computing single intermediate points on the efficient frontiers of 100% dense covariance matrix Markowitz problems using the λ-parameter method tabulated by problem size and optimizer.

Size	λ-Parameter single intermediate point times			
	Risk solver	Matlab	LINGO	Cplex
$n=500$	8.1 s	51.2 s	34.9 s	2.3 s
$n=1000$	55.1 s	936.6 s	–	5.4 s
$n=1500$	250.0 s	–	–	17.6 s
$n=2000$	347.8 s	–	–	29.3 s
$n=3000$	–	–	–	85.3 s

understand, (ii) the critical line method was not useful for much in its early days due to the CPU-time and memory limitations of computers at the time, and (iii) since Karmarkar in [8], the emphasis has been on interior-point algorithms of polynomial-time complexity as opposed to parametric procedures which are only of exponential-time complexity. Consequently, discrete methods using interior-point or barrier-type algorithms (such as in the experiments of Tables 4 and 5) are about all that is ever seen used in efficient frontier computation today.

Table 6 shows some remarkable results and that parametric methods, after all these years, deserve more than a second look. While interior-point optimizers may be faster on other types of problems, they certainly are not on the computation of the efficient frontiers of large-scale dense covariance matrix Markowitz problems. For instance in Table 6, CIOS takes 5.3 s on average to compute the entire efficient frontier of a problem with 1000 securities, whereas in Table 5, the fastest barrier or interior-point solver takes 5.4 s on average just to compute a single efficient frontier point. In the experiments of Table 6, we used CIOS, the code written to implement the parametric quadratic programming approach of Hirschberger et al. [7], as a stand-in for parametric methods in general, because, without any parametric procedures existing in any package of which we are aware, there is a shortage of such codes in practice.

To illustrate the impact of the numbers in Table 6, consider the following. While it would take Risk Solver Platform between 347.8 and 513.0 s times a factor as large as 100 and Cplex's interior-point routine between 29.3 and 38.6 s times a factor as large as 100 to construct a 100-point representation of an efficient frontier of a 2000-security problem, it would only take a parametric procedure represented by CIOS 23.1 s on average to compute the whole efficient frontier complete with all details about the piecewise linear/piecewise hyperbolic nature of the frontier as described in Section 3.

Also shown in Table 6 are the average numbers of hyperbolic segments per efficient frontier. In this way, there are two reasons why CPU-times increase as we go down the table. One of course is problem size. The other is because the average number of hyperbolic segments per efficient frontier (each of which must be computed) increases with problem size.

5. Concluding remarks

Suppose a user has always viewed efficient frontiers in dotted representation form, thinks they are more aesthetically pleasing that way, and has no interest in changing. Going along with this, it would still be faster to compute the whole frontier by a parametric procedure. Then, by postprocessing the information that is outputted to Tables like 2 and 3, points can be placed on the curve of the efficient frontier in any desired pattern. For example, if a user wanted a 50-point e-constraint portrayal of the efficient frontier of a 1500 security problem, instead of taking 22.2 or more seconds multiplied by a factor as high as 50 with an interior-point or barrier-type algorithm, CIOS with postprocessing could produce precisely the same results in about 12.2 s plus about 1 or 2 s more for postprocessing.

In summary, no commercial optimizers of which we are aware can compute a parametric specification of the efficient frontier of any non-trivial problem, let alone the large-scale problems addressed in this paper. All they can do is conduct repetitive optimizations to construct an efficient frontier in the form of a dotted representation. As indicated in Tables 4–6, one needs to be aware of the great variation among optimizers and approaches when attempting to carry out the task of computing an efficient frontier. Many users might not be aware of this and consequently waste a lot of valuable time as a result.

Table 6
Average run times (sample size 5 in each case) and numbers of hyperbolic segments encountered when computing the entire (continuous) curves of the efficient frontiers of 100% dense covariance matrix Markowitz problems using the parametric quadratic programming approach of Hirschberger, Qi and Steuer [7] tabulated by problem size.

Size	Whole continuous efficient frontier	
	Hyperbolic segments	CIOS time
$n=500$	179.6	1.5 s
$n=1000$	227.8	5.3 s
$n=1500$	249.4	12.2 s
$n=2000$	272.6	23.1 s
$n=3000$	294.6	34.3 s

But what Table 6 really says is two things. One is that interactivity that has not been possible in a portfolio system that uses an interior-point or barrier-type optimizer is possible if a parametric procedure is used, even though, ironically, the parametric procedure would only be of exponential-time complexity. The other is that there is really no need to construct the efficient frontier of a continuous-variable Markowitz problem by repetitive optimization anymore. All can be accomplished just as well with a parametric code like CIOS plus postprocessing in far less time, with the savings more pronounced the larger the problem.

References

[1] F. Ben Abdelaziz, B. Aouni, R. El-Fayedh, Multi-objective stochastic programming for portfolio selection, European Journal of Operational Research 177 (3) (2007) 1811–1823.

[2] M.J. Best, An algorithm for the solution of the parametric quadratic programming problem, in: B. Riedmüller, H. Fischer, S. Schäffler (Eds.), Applied Mathematics and Parallel Computing: Festschrift for Klaus Ritter, Physica-Verlag, 1996, pp. 57–76.

[3] Cplex, Cplex 11.1 User's Manual, ILOG, Inc., Mountain View, California, 2007.

[4] M. Ehrgott, C. Waters, R. Kasimbeyli, O. Ustun, Multiobjective programming and multiattribute utility functions in portfolio optimization, INFOR 47 (1) (2009) 31–42.

[5] Y. Fang, S. Wang, Fuzzy Portfolio Optimization: Theory and Methods, Publisher of High Level Education, Beijing, China, 2005.

[6] M. Hirschberger, Y. Qi, R.E. Steuer, Randomly generating portfolio-selection covariance matrices with specified distributional characteristics, European Journal of Operational Research 177 (3) (2007) 1610–1625.

[7] M. Hirschberger, Y. Qi, R.E. Steuer, Large-scale MV efficient frontier computation via a procedure of parametric quadratic programming, European Journal of Operational Research 204 (3) (2010) 581–588.

[8] N. Karmarkar, A new polynomial-time algorithm for linear programming, Combinatorica 4 (4) (1984) 373–395.

[9] M. Köksalan, C. Tuncer, A DEA-based approach to ranking multi-criteria alternatives, International Journal of Information Technology & Decision Making 8 (1) (2009) 29–54.

[10] LINGO Version 11, Lindo Systems, Inc, Chicago, 2008.

[11] J.L. Maginn, D.L. Tuttle, J.E. Pinto, D.W. McLeavy, Managing Investment Portfolios, 3rd edition John Wiley, New York, 2007.

[12] H.M. Markowitz, Portfolio selection, Journal of Finance 7 (1) (1952) 77–91.

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

[14] H.M. Markowitz, Portfolio Selection: Efficient Diversification in Investments, John Wiley, New York, 1959.

[15] H.M. Markowitz, Mean–Variance Analysis in Portfolio Choice and Capital Markets, Basil Blackwell, Oxford, 1987.

[16] H.M. Markowitz, G.P. Todd, Mean–Variance Analysis in Portfolio Choice and Capital Markets, Frank J. Fabozzi Associates, New Hope, Pennsylvania, 2000.

[17] Matlab, Optimization Toolbox for Use with Matlab, Version R2009a, Mathworks, Inc, Natick, Massachusetts, 2009.

[18] W. Michalowski, W. Ogryczak, Extending the MAD portfolio optimization model to incorporate downside risk aversion, Naval Research Logistics 48 (3) (2001) 185–200.

[19] R.O. Michaud, The Markowitz optimization enigma: is 'optimized' optimal? Financial Analysts Journal 45 (1989) 31–42.

[20] K.M. Miettinen, Nonlinear Multiobjective Optimization, Kluwer, Boston, 1999.

[21] A. Niedermayer, D. Niedermayer, Applying Markowitz's critical line algorithm, in: J.B. Guerard (Ed.), Handbook of Portfolio Construction, Springer-Verlag, Berlin, 2010, pp. 383–400.

[22] Y. Peng, G. Kou, Y. Shi, Z. Chen, A descriptive framework for the field of data mining and knowledge discovery, International Journal of Information Technology & Decision Making 7 (4) (2008) 639–682.

[23] Risk Solver Platform Version 9.5, Frontline Systems, Incline Village, Nevada, 2009.

[24] M. Stein, J. Branke, H. Schmeck, Efficient implementation of an active set algorithm for large-scale portfolio selection, Computers & Operations Research 35 (12) (2008) 3945–3961.

[25] C. Stummer, E. Kiesling, W.J. Gutjahr, A multicriteria decision support system for competence-driven project portfolio selection, International Journal of Information Technology & Decision Making 8 (2) (2009) 379–401.

[26] K.-J. Tseng, Y.-H. Liu, J.-F. Ho, An efficient algorithm for solving a quadratic programming model with application in credit card holders' behavior, International Journal of Information Technology & Decision Making 7 (3) (2008) 421–430.

[27] P. Xidonas, D. Askounis, J. Psarras, G. Mavrotas, Portfolio engineering using the IPSSIS multiobjective optimisation decision support system, International Journal of Decision Sciences, Risk and Management 1 (1/2) (2009) 36–53.

[28] S.X. Xie, Z.F. Li, S.Y. Wang, Continuous-time portfolio selection with liability: mean–variance model and stochastic LQ approach, Insurance, Mathematics & Economics 42 (3) (2008) 943–953.

[29] C. Zopounidis, M. Doumpos, INVESTOR: a decision support system based on multiple criteria for portfolio selection and composition, in: A. Colorni, M. Paruccini, B. Roy (Eds.), Multiple Criteria Decision Aiding, European Commission Joint Research Centre, Brussels, 2000, pp. 371–381.

Ralph E. Steuer is the Charles S. Sanford, Sr. Chair of Business in the Department of Banking and Finance of the Terry College of Business at the University of Georgia. He holds an electrical engineering degree from Brown University, an M.B.A. from Cornell University, and a Ph.D. in quantitative methods from the University of North Carolina. Prior to joining the University of Georgia, Dr. Steuer was on the faculty at the University of Kentucky (eight years) and a Visiting Associate Professor at Princeton University (one year). Dr. Steuer's research interests are in multiple-attribute portfolio theory, efficient sets and surfaces, multiple objective programming, and multiple criteria decision making.

Yue Qi is a Professor in the Department of Financial Management at Nankai University, Tianjin, China. His research interests are in financial management and multi-attribute portfolio analysis. Prior to joining Nankai University, Dr. Qi was for one year on the faculty of the International University of Monaco in the Principality of Monaco.

Markus Hirschberger has his Ph.D. in mathematics from the Katholische University of Eichstaett-Ingolstadt, Eichstaett, Germany. His dissertation was on parametric quadratic programming. His research interests are in various forms of risk analysis, multi-parametric extensions to quadratic programming, and the computation of efficient surfaces in portfolio selection. In addition, Dr. Hirschberger is Supervisor of Quantitative Research at MEAG MUNICH ERGO Asset Management GmbH in Munich.