INFOR, Vol. 47, No. 1, Month 2009, pp. 15–21
ISSN 0315-5986 | EISSN 1916-0615

Dotted Representations of Mean-Variance Efficient Frontiers and their Computation

Yue Qi
Hedge Fund Research Institute, International University of Monaco, Principality of Monaco, e-mail: yorkche@nankai.edu.cn

Markus Hirschberger
Department of Mathematics, University of Eichstätt-Ingolstadt, Eichstätt, Germany, e-mail: markus.hirschberger@gmx.de

Ralph E. Steuer
Terry College of Business, University of Georgia, Athens, Georgia 30602-6253 USA, e-mail: rsteuer@uga.edu

Abstract—This paper is about dotted representations of efficient frontiers. Dotted representations, as in portfolio selection, can often be the most practical way of communicating an efficient frontier. The most popular method is to minimize variance subject to fixed levels of expected return. However, even when the fixed levels are evenly dispersed, one can not count on the resulting dots being evenly dispersed. Another method uses fixed values of a risk tolerance parameter, but with this method the resulting dots are even less controllable. In this paper we develop a third approach applicable to what we call Markowitz problems (mean-variance problems with all linear constraints). The approach utilizes the results of algorithms that can compute all hyperbolic segments of a Markowitz efficient frontier. Then the approach can place dots on the hyperbolic segments of the efficient frontier in a variety ways including equally spaced. The advantage of the approach is the speed at which dotted representations can be produced and modified, particularly on large applications.

Keywords Portfolio selection, mean-variance efficient frontiers, hyperbolic segments, e-constraint method, trapezoidal rule, parametric quadratic programming.

1. INTRODUCTION

In finance, for about thirty years (until about the mid 1980s), mean-variance models and their resultant efficient frontiers were a hot topic in portfolio selection. See, for instance, Elton, Gruber, Brown and Goetzmann (2007). But after a somewhat less active period, interest in portfolio selection, since about 2000, has shown signs of resurgence. Reasons include dramatically faster computers, improved algorithmic techniques, and new ideas such as those suggested by Ben Abdelaziz, Aouni and El-Fayedh (2007), Arenas Parra, Bilbao Terol and Rodríguez Uría (2001), Bana e Costa and Soares (2004), Best and Hlouskova (2005), Ehrgott, Klamroth and Schwehm (2004), Hallerbach, Ning, Soppe and Spronk (2004), Mansini, Ogryczak and Speranza (2003), Ruszczyński and Vanderbei (2003), Stein, Branke and Schmeck (2007), Fang and Wang (2005), and others.

Received November 2007; Revision August 2008; Accepted May 2009

While there can be many kinds of mean-variance problems, the focus in this paper is on problems with all linear constraints. We call such problems "Markowitz problems" because of the extent studied by Markowitz (1959, 1987, etc.). As seen later, the approach described herein for creating dotted representations of efficient frontiers is especially effective on large-scale Markowitz problems. We know of no other paper that has dealt with the intricacies of presenting efficient frontiers as we have in this paper.

In bi-criterion format, let us assume the following Markowitz problem formulation

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} x_i \sigma_{ij} x_j \qquad \text{variance} \tag{1.1}$$

$$\max \sum_{i=1}^{n} \mu_i x_i \qquad \text{expected return} \tag{1.2}$$

$$s.t. \quad \sum_{i=1}^{n} x_i = 1 \tag{1.3}$$

$$\alpha_i \leq x_i \leq \omega_i \tag{1.4}$$

where

(a) n is the number of securities in the pool of securities eligible for use in a portfolio
(b) x_i is the proportion of a sum to be initially invested in security i
(c) μ_i is the expected return of security i
(d) σ_{ij} is the covariance of the returns of securities i and j
(e) constraint (1.3) is mandatory to assure that all x_i proportions sum to one
(f) (1.4) is for additional linear constraints (here just lower and upper bounds on the x_i)

Whereas an $(x_1, \ldots, x_n)$-vector is a *portfolio* if and only if it satisfies (1.3), a portfolio is *feasible* if and only if it satisfies (1.4).

A feasible portfolio $\mathbf{x} \in \mathbb{R}^n$ has a point $\mathbf{z} \in \mathbb{R}^2$ on the efficient frontier if and only if there is no other feasible portfolio that has a higher expected return without a higher variance or has a lower variance without a lower expected return. Hereafter, when using the term "efficient frontier," it will always be in reference to Markowitz problems as above. A typical efficient frontier is shown in Figure 1. Although not easy to discern with the naked eye, efficient frontiers typically consist of a series of hyperbolic line segments

Despite the prevalence of the term "mean-variance," efficient frontiers are typically displayed with standard deviation on the horizontal axis. This is because theory and computation are carried out in terms of variance, but when communicating an efficient frontier to a user, standard deviation is commonly employed because of its more intuitive interpretation.

The paper is organized as follows. Section 2 reviews the hyperbolic structure of an efficient frontier and illustrates the information necessary for a full specification of an efficient frontier's hyperbolic segments. Section 3 discusses computational experience with the two methods mentioned in the abstract for constructing dotted representations of an efficient frontier. Section 4 reviews a method by which the arc-length of an efficient frontier can be compiled in a cumulative sense. Section 5 presents a routine for creating points on a hyperbolically specified efficient frontier in a variety of ways including equally spaced. Section 6 ends the paper with concluding remarks.

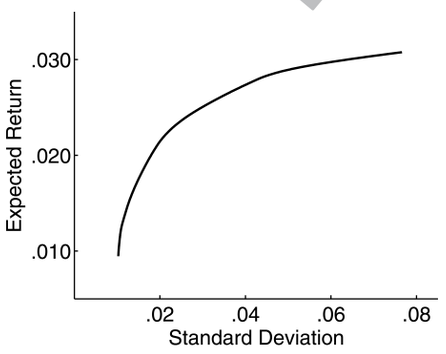

Figure 1. Efficient frontier of a Markowitz problem with $n = 500$ securities. Note that an efficient frontier isn't always perfectly smooth as so often seen in textbooks

2. STRUCTURE OF THE EFFICIENT FRONTIER

To note the hyperbolic nature of the efficient frontier, we begin by observing two things.

(1) In portfolio selection, as in any problem with more than one criterion, there are two versions of the feasible region. One is $S \subset \mathbb{R}^n$ and the other is $Z \subset \mathbb{R}^2$. In our case, S is the set of all feasible portfolios and Z is the set of all (standard deviation, expected return) image vectors resulting from the portfolios in S.
(2) Any straight line segment in S has as its set of resulting (standard deviation, expected return) image vectors a segment of a hyperbola.

Viewing S as the set of all straight lines connecting all pairs of feasible portfolios, it is not hard to see why the efficient frontier, as a portion of the boundary of Z, is piecewise hyperbolic.

To illustrate, consider the three-security example taken from Rockafellar and Uryasev (2000) of Table 1 in which $\boldsymbol{\Sigma}$ is the covariance matrix of the σ_{ij} as in (1.1) and $\boldsymbol{\mu}$ is the vector of μ_i as in (1.2). With lower and upper bounds of 0 and 1 on all of the x_i, S and Z for this problem are in Figure 2. Note the

TABLE 1.
Data for Rockafellar-Uryasev three-security example

	1	2	3
$\boldsymbol{\Sigma}$	.0032465	.0002298	.0042040
	.0002298	.0004994	.0001925
	.0042040	.0001925	.0076410
$\boldsymbol{\mu}$	.0101110	.0043532	.0137058

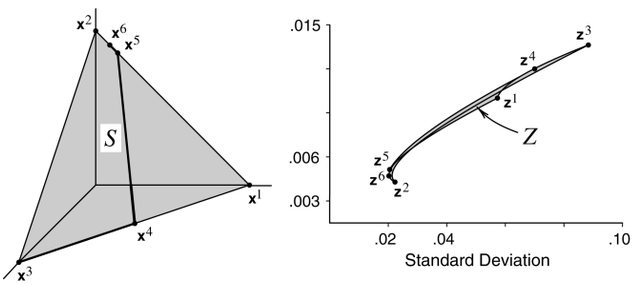

Figure 2. Rockafellar-Uryasev three-security example

DOI 10.3138/infor.47.1.015

curve (part of which is interior to Z) connecting $\mathbf{z}^1$ and $\mathbf{z}^2$ in Z. This is a portion of a hyperbola as its inverse image set is the straight line connecting $\mathbf{x}^1$ and $\mathbf{x}^2$ in S. By the same token, the curves connecting $\mathbf{z}^4$ and $\mathbf{z}^5$, and $\mathbf{z}^5$ and $\mathbf{z}^6$, are hyperbolic as their inverse image sets, $\mathbf{x}^4$ and $\mathbf{x}^5$, and $\mathbf{x}^5$ and $\mathbf{x}^6$, are straight lines. In this way, an efficient frontier is piecewise hyperbolic while its set of inverse image portfolios is piecewise linear. We will call points such as $\mathbf{x}^3, \mathbf{x}^4, \mathbf{x}^5, \mathbf{x}^6 \in S$ that correspond to the endpoints of the hyperbolic segments *corner* portfolios.

For a mathematical description of an efficient frontier, technical information about the piecewise linear path in S and all of the efficient frontier's hyperbolic segments in Z must be specified. For the three-security Rockafellar and Uryasev example, this is given in Tables 2 and 3.

To appreciate the tables, suppose we are curious about a point $\mathbf{z}^*$ on the efficient frontier whose expected return $\mu^* = .010$. This places $\mathbf{z}^*$ on the middle hyperbolic segment. To calculate the standard deviation σ^* of the point, we use the a_0, a_1, a_2 of the hyperbolic segment as follows

$$\begin{aligned}\sigma^* &= \sqrt{a_0 + a_1\mu^* + a_2(\mu^*)^2} \\ &= \sqrt{.00244 - 0.82108(.010) + 86.7464(.010)^2} \\ &= .05389\end{aligned}$$

TABLE 2.
From the top of the efficient frontier down, this table specifies the expected returns of all hyperbolic segment endpoints along with three parameters for specifying each hyperbola for the Rockafellar-Uryasev three-security example

Hyperbolic Segment	μ^{upper}	μ^{lower}	a_0	a_1	a_2
$\mathbf{z}^3$ to $\mathbf{z}^4$	.01370	.01190	.01747	−3.34735	191.8733
$\mathbf{z}^4$ to $\mathbf{z}^5$	.01190	.00507	.00244	−0.82108	86.7464
$\mathbf{z}^5$ to $\mathbf{z}^6$	.00507	.00483	.00278	−0.95669	99.1275

TABLE 3.
From the top of the frontier down, this table specifies the compositions of the corner portfolios (whose images are the endpoints of the hyperbolic segments) for the Rockafellar-Uryasev three-security example

Corner Portfolio	x_1	x_2	x_3
$\mathbf{x}^3$	.0	.0	1.0
$\mathbf{x}^4$	.50176	.0	.49824
$\mathbf{x}^5$	.12396	.87604	.0
$\mathbf{x}^6$	.08204	.91796	.0

To calculate the composition of the inverse image portfolio $\mathbf{x}^*$ of $\mathbf{z}^*$, we use the μ^{upper} and μ^{lower} of the hyperbolic segment as follows

$$\begin{aligned}\mathbf{x}^* &= \frac{\mu^* - \mu^{lower}}{\mu^{upper} - \mu^{lower}}\,\mathbf{x}^4 + \frac{\mu^{upper} - \mu^*}{\mu^{upper} - \mu^{lower}}\,\mathbf{x}^5 \\ &= \frac{.010 - .00507}{.01190 - .00507}\,\mathbf{x}^4 + \frac{.01190 - .010}{.01190 - .00507}\,\mathbf{x}^5 \\ &= (.39666,\ .24370,\ .35964)\end{aligned}$$

3. APPROACHES FOR COMPUTING EFFICIENT FRONTIERS

The "e-constraint" and "risk tolerance factor" methods for producing dotted representations of efficient frontiers are as follows. Both involve the repetitive optimization of a quadratic programming problem. Let us assume, that a q-dot representation is to be computed.

The "e-constraint" method employs formulation

$$\begin{aligned}&\min \mathbf{x}^T\boldsymbol{\Sigma}\,\mathbf{x} \\ &s.t.\quad \boldsymbol{\mu}^T\mathbf{x} \geq \rho \\ &\mathbf{x} \in S\end{aligned} \tag{1}$$

The idea is to obtain solutions to (1) for q different values of ρ

$$\rho_1 < \rho_2 < \ldots < \rho_q$$

where ρ_q causes (1) to produce the maximum expected return point on the efficient frontier, ρ_1 causes (1) to produce the minimum variance point on the efficient frontier, and the other ρ_i cause (1) to produce $q - 2$ in-between points. The method is a little more difficult than it looks because one or two preliminary optimizations may be necessary to set the values for ρ_1 and ρ_q in the first place. We use the term "e-constraint" because this is the term attributed to the method in multiple criteria optimization for generating efficient points by converting all constraints to objectives except one (see for instance Miettinen (1999)).

The "risk tolerance factor" method employs formulation

$$\begin{aligned}&\min \mathbf{x}^T\boldsymbol{\Sigma}\,\mathbf{x} - \lambda\boldsymbol{\mu}^T\mathbf{x} \\ &\mathbf{x} \in S\end{aligned} \tag{2}$$

The idea is to obtain solutions to (2) for q different values of λ

$$\lambda_1 < \lambda_2 < \ldots < \lambda_q$$

where λ_q causes (2) to produce the maximum expected return point on the efficient frontier, $\lambda_1 = 0$ causes (2) to produce the minimum variance point on the efficient frontier, and the other λ_i cause (2) to produce $q - 2$ in-between points. A weighting approach, this method works in the context of this paper as S is polyhedral and $\boldsymbol{\Sigma}$ is positive semi-definite (being a covariance matrix). Frustrations with the method are

ISSN 0315-5986 | EISSN 1916-0615 DOI 10.3138/infor.47.1.015

that a few preliminary optimizations may be necessary to ascertain a good value to use for λ_q and that it is difficult to know how to space the λ_i to obtain nicely spaced points on the efficient frontier.

With small problems, there is no difficulty in solving (1) or (2). But when the number of securities begins to exceed more than a few hundred, implementation can be a disappointment. For instance, with Premium Solver Platform (Frontline Systems, 2007), we encountered run times of about 7 minutes[1] per dot at 400 securities (roughly 2 1/3 hours for a 20-dot representation). With Matlab's QUADPROG routine (Matlab, n.d.), we encountered run times of about 11 minutes per dot at 1,000 securities (roughly 3 1/2 hours for a 20-dot representation). With LINGO (Schrage, 2007), which is faster than Matlab, but reads from an Excel spreadsheet, most of the time we could not get beyond 600 securities with a dense covariance matrix because of freeze ups from Excel 2003 on data read in. But with Cplex (2007), things are much faster. With a dense covariance matrix and 2,000 securities, we were able to obtain run times of about 70 seconds per dot (roughly 20–25 minutes for a 20-dot representation).

However, dotted representations can generally be created much more quickly. The proposed approach involves utilizing an algorithm, such as implemented in the CIOS code described in Hirschberger, Qi and Steuer (2008), to generate information as in Tables 2 and 3 about all of an efficient frontier's hyperbolic segments. As reported in that paper, CIOS is of such a speed that it can often compute all of an efficient frontier's hyperbolic segments in less time than it takes Cplex to compute just a single dot. Although CIOS is the only code currently demonstrated to operate at such a speed, research by Niedermayer and Niedermayer (2007) indicates that other well-tested codes in the same speed-class could soon be available. With CIOS available, self-contained (doesn't involve components from any external packages), and written in Java for transportability, we use CIOS to illustrate.

In the following two sections we (a) describe how we compute the arc length of an efficient frontfier in a cumulative fashion and (b), with this accomplished, describe a routine for developing and displaying various dotted representations of an efficient frontier.

4. TRAPEZOIDAL PROCESS

Fashioned to a hyperbolic segment, a classical method for computing arc length is as follows. Let μ denote expected return. With respect to a point $(\sigma(\mu), \mu)$ on hyperbolic segment i,

$$\sigma(\mu) = \sqrt{a_{i,0} + a_{i,1}\mu + a_{i,2}\mu^2} \qquad \mu \in [\mu^{lower}, \mu^{upper}] \quad (3)$$

[1] All run times in the paper pertain to a 2.66 GHz Dell Core 2 Duo Desktop with 3GB of RAM at the University of Georgia.

We now note that when displaying efficient frontiers, the scale of the vertical axis is typically "magnified" with respect to that of the horizontal. For instance, suppose a graph's aspect ratio is $6/5$. That is, its horizontal axis is 20% longer than its vertical axis. Also, suppose that the range of the scale on horizontal axis is from .02 to .16 while that on the vertical axis is from .01 to .04. Thus, the *magnification factor* would be given by

$$m = \frac{5}{6}\,\frac{(.16 - .02)}{(.04 - .01)} = 3.88$$

Then, taking this into account, the Euclidean norm of the gradient $(\sigma'(\mu), m)$ on hyperbolic segment i is

$$l_i(r) = \sqrt{\frac{(a_{i,1} + 2a_{i,2}\mu)^2}{4(a_{i,0} + a_{i,1}\mu + a_{i,2}\mu^2)} + m^2}$$

and the hyperbolic segment's arc length on the graph is given by

$$L_i = \int_{\mu^{lower}}^{\mu^{upper}} l_i(\mu)\, d\mu$$

Unfortunately, there is no closed form representation of L_i. However, L_i may be approximated numerically by the trapezoid formula ($t \in \mathbb{N}$)

$$L_i \approx \frac{1}{2}\, h_i\, l_i(\mu^{lower}) + \sum_{k=1}^{t-1} h_i\, l_i(\mu^{lower} + kh_i) + \frac{1}{2} h_i\, l_i(\mu^{upper}) \quad (4)$$

where

$$h_i = \frac{\mu^{upper} - \mu^{lower}}{t}$$

Letting t be a number like 100 and then accumulating the terms of (4) one-by-one, one hyperbolic segment after the other, a schedule for accumulated displayed arc length as we proceed along the efficient frontier, indexed by both μ and σ, can be constructed. Using this schedule of accumulated μ, σ, and arc length information at many points along the efficient frontier, dots can be placed on the efficient frontier according to virtually any μ, σ, or arc-length pattern.

5. ROUTINE

The routine we have prepared to carry out this paper is written in Matlab. Perhaps, a more permanent version could be programmed in C++ or Java. It begins by asking for the paths of two input files. One file contains expected return range and a_0, a_1, a_2 information for each of the efficient frontier's

DOI 10.3138/infor.47.1.015

hyperbolic segments as in Table 2. The other contains all corner portfolio **x**-vectors as in Table 3. Described in terms of parameters to configure a given run, we have

(1) **frontierType**: Type of frontier to be outputted
  $= 0$ smooth curve only
  $= 1$ hyperbolic segment endpoints only
  $= 2$ equally spaced dots by expected return
  $= 3$ equally spaced dots by standard deviation
  $= 4$ equally spaced dots by displayed arc length
(2) **numDots**: Number of dots used to represent efficient frontier when frontierType ≥ 2.
(3) **yL** and **yU**: range lower and upper bounds of the values on the vertical axis
(4) **xL** and **xU**: range lower and upper bounds of the values on the horizontal axis
(5) **aspectRatio**: length of the vertical axis divided by the length of the horizontal axis

To demonstrate the approach of the paper, let us first solve for the efficient frontier of a 500-security problem using CIOS in the form of Tables 2 and 3. Then, after reading in the information in Tables 2 and 3, the routine is run with the following settings.

$$\begin{aligned} \textbf{frontierType} &= 0, 1, 2, 3, 4 \\ \textbf{numDots} &= 30 \\ \textbf{yU} &= .035 \\ \textbf{yL} &= .005 \\ \textbf{xL} &= .000 \\ \textbf{xU} &= .085 \\ \textbf{aspectRatio} &= 5/4 \end{aligned}$$

This produces five output graphs. Setting frontierType $= 0$ produces the efficient frontier in smooth form. The graph produced is actually the one in Figure 1. Setting frontierType $= 1$ produces Figure 3 which is the dotted representation resulting from the efficient frontier's 55 hyperbolic segment endpoints.

Setting frontierType $= 2$ produces Figure 4 which a 30-dot equally spaced by expected return representation of the efficient frontier, and setting frontierType $= 3$ produces Figure 5 which is a 30-dot equally spaced by standard deviation representation of the efficient frontier. Note the consequent relative sparseness of the dots at the top of the frontier in Figure 4 and the relative sparseness of the dots at the bottom of the frontier in Figure 5.

Using the magnification factor of $m = 2.266$ which is computed inside the routine, setting frontierType $= 4$ produces the 30-dot equally spaced by displayed arc-length representation of the efficient frontier as shown in Figure 6.

With regard to CPU run time requirements, let us begin by commenting on the 30-dot equally spaced by expected return representation of Figure 4. Whereas it takes Cplex, as shown in Hirschberger, Qi and Steuer (2008), about 3 seconds to solve for each dot (roughly 90 seconds for the 30-dot representation), it takes the approach of this paper only about 1.5 seconds total (about 1 second for CIOS to generate all hyperbolic information and about 0.5 seconds for the routine). Furthermore, the time difference tends to become more pronounced the larger the application. For instance, on a 1,500-security dense covariance

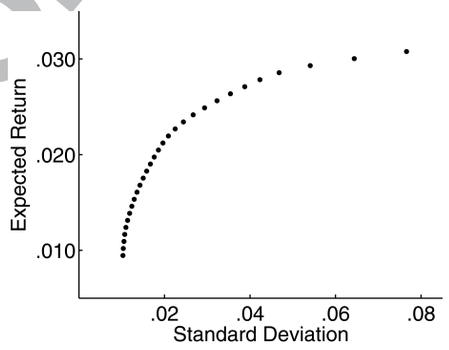

Figure 4. Resulting from frontierType $= 2$, this is the 30-dot equally spaced by expected return representation of the efficient frontier of the $n = 500$ problem

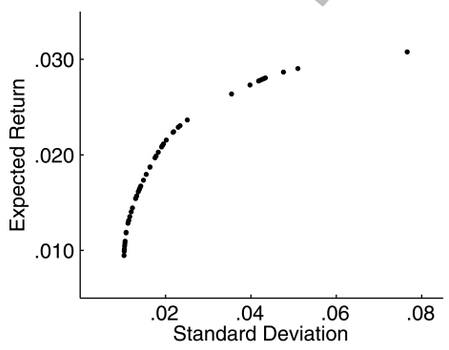

Figure 3. Resulting from frontierType $= 1$, this is the 55-dot representation of the efficient frontier of the $n = 500$ problem given by the endpoints of the problem's 54 hyperbolic segments

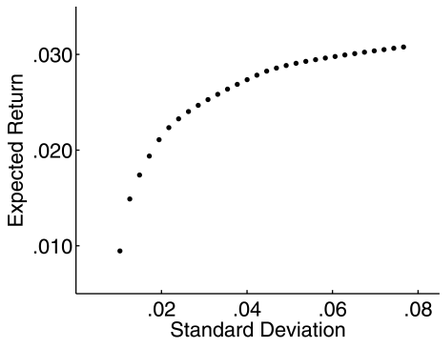

Figure 5. Resulting from frontierType $= 3$, this is the 30-dot equally spaced by standard deviation representation of the efficient frontier of the $n = 500$ problem

ISSN 0315-5986 | EISSN 1916-0615 DOI 10.3138/infor.47.1.015

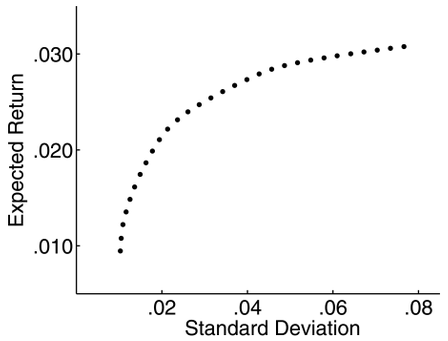

Figure 6. Resulting from frontierType = 4, utilizing the magnification factor, this is the 30-dot equally spaced by displayed arc-length representation of the efficient frontier of the $n = 500$ problem

matrix problem experiment that we conducted, it takes Cplex about 40 seconds to solve for each dot (roughly 20 minutes for a 30-dot representation), but it only takes the approach of this paper about 26 seconds total (about 25 seconds for CIOS and about 1 second for the routine).

With regard to the 30-dot equally spaced by standard deviation representation of Figure 5, the difference is even more pronounced as Cplex would have to use its second-order cone solver which takes a little longer per dot. Note that both Premium Solver Platform and Matlab essentially fall out of the picture at about 400 and 800 securities, respectively, because of excessive run time.

With regard to the 30-dot equally spaced by displayed arc-length representation of Figure 6, this can only be carried out by the approach of this paper as it would be impossible to know how to set up (1) or (2) beforehand to produce such a result.

6. CONCLUDING REMARKS

Dotted representations can often be the most practical way of communicating an efficient frontier. But creating dotted representations of an efficient frontier has been a laborious process, typically taking at least one quadratic optimization per dot. As discussed in Sections 3 and 5, this can take considerable time as problem size increases. But with the approach of this paper, CPU time requirements can be reduced by at least an order of magnitude.

In addition, the approach of this paper allows certain things to be done that either cannot be done with the e-constraint or risk tolerance factor methods, or would take considerable extra time. One is that it is possible to create dotted distributions, like equally spaced by arc length, that are not possible with either the e-constraint or risk tolerance factor methods. Another pertains to flexibility. Suppose that after running CIOS and creating a 30-dot representation, one decides that a 40-dot representation would be better. Because few dots would likely be reusable for the 40-dot representation, this would essentially require the e-constraint or risk tolerance factor methods to start again from scratch. But from the information generated by CIOS, dotted distributions can be created and modified virtually at will with only very little consequence with regard to computer run time, even on portfolio problems with up to thousands of securities.

REFERENCES

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

Arenas Parra, M., Bilbao Terol, A., and Rodríguez Uría, M. V. (2001), "A fuzzy goal programming approach to portfolio selection", *European Journal of Operational Research*, 133(2): 287–297.

Bana e Costa, C. A. and Soares, J. O. (2004), "A multicriteria model for portfolio Management", *European Journal of Finance*, 10(3): 198–211.

Best, M. J. and Hlouskova, J. (2005), "An algorithm for portfolio optimization with transaction costs", *Management Science*, 51(11): 1676–1688.

Cplex. (2007), "*Cplex 11.1 User's Manual*", ILOG, Inc., Mountain View, California.

Ehrgott, M., Klamroth, K., and Schwehm, C. (2004), "An MCDM approach to portfolio optimization", *European Journal of Operational Research*, 155(3): 752–770.

Elton, E. J., Gruber, M. J., Brown, S. J., and Goetzmann, W. N. (2007), *Modern Portfolio Theory and Investment Analysis*, 7th edition, John Wiley, New York.

Fang, Y. and Wang, S. (2005), *Fuzzy Portfolio Optimization: Theory and Methods*, Publisher of High Level Education, Beijing, China.

Frontline Systems (2007), *Premium Solver Platform 7.1*, Incline Village, Nevada.

Hallerbach, W. G., Ning, H., Soppe, A., and Spronk, J. (2004), "A framework for managing a portfolio of socially responsible investments", *European Journal of Operational Research*, 153(2): 517–529.

Hirschberger, M., Qi, Y., and Steuer, R. E. (2008), *Mean-varince efficient frontier computation via parametric quadratic programming*, Department of Banking and Finance, University of Georgia, Athens.

Mansini, R., Ogryczak, W., and Speranza, M. G. (2003), "On LP solvable models for portfolio selection", *Informatica*, 14(1): 37–62.

Markowitz, H. M. (1959), *Portfolio Selection: Efficient Diversification in Investments*, John Wiley, New York.

Markowitz, H. M. (1987), *Mean-Variance Analysis in Portfolio Choice and Capital Markets*, Basil Blackwell, Oxford.

Matlab. (n.d.), *Optimization Toolbox for Use with Matlab*, Version R2008a, Mathworks, Inc., Natick, Massachusetts.

Miettinen, K. M. (1999), *Nonlinear Multiobjective Optimization*, Kluwer, Boston.

Niedermayer, A. and Niedermayer, D. (2007), *Applying Markowitz's critical line algorithm*, Diskussionsschriften dp0602, Volkswirtschaftliches Institut, Universität Bern, Switzerland.

ISSN 0315-5986 | EISSN 1916-0615 DOI 10.3138/infor.47.1.015

Rockafellar, R. T. and Uryasev, S. (2000), "Optimization of conditional value-at-Risk", *Journal of Risk*, 2: 21–41.

Ruszczyński, A. and Vanderbei, R. J. (2003), "Frontiers of stochastuically non-dominated portfolios", *Econometrica*, 71(4): 1287–1297.

Schrage, L. (2007), *LINGO User's Guide (for LINGO 11.0)*, Lindo Publishing, Chicago.

Stein, M., Branke, J., and Schmeck, H. (2007), "Efficient implementation of an active set algorithm for large-scale portfolio selection", *Computers & Operations Research*.

ISSN 0315-5986 | EISSN 1916-0615 DOI 10.3138/infor.47.1.015