Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2012, Article ID 105616, 25 pages
doi:10.1155/2012/105616

Research Article

On the Computation of the Efficient Frontier of the Portfolio Selection Problem

Clara Calvo, Carlos Ivorra, and Vicente Liern

Departamento de Matemáticas para la Economía y la Empresa, Universidad de Valencia, P.O. Box 46022, Valencia, Spain

Correspondence should be addressed to Carlos Ivorra, carlos.ivorra@uv.es

Received 22 December 2011; Revised 17 May 2012; Accepted 18 May 2012

Academic Editor: Yuri Sotskov

An easy-to-use procedure is presented for improving the ε-constraint method for computing the efficient frontier of the portfolio selection problem endowed with additional cardinality and semicontinuous variable constraints. The proposed method provides not only a numerical plotting of the frontier but also an analytical description of it, including the explicit equations of the arcs of parabola it comprises and the change points between them. This information is useful for performing a sensitivity analysis as well as for providing additional criteria to the investor in order to select an efficient portfolio. Computational results are provided to test the efficiency of the algorithm and to illustrate its applications. The procedure has been implemented in Mathematica.

1. Introduction

The portfolio selection problem consists of finding an efficient portfolio in the sense of obtaining a tradeoff between the expected return and the risk of the investment. Most portfolio selection models are based on the original Markowitz model [1, 2], in which the expected return of a given portfolio is measured by $\mathbf{e}^t\mathbf{x}$, where $\mathbf{e}$ is the vector of mean returns of the assets and $\mathbf{x}$ contains the weight of each asset in the portfolio. On the other hand, the risk is measured by $\mathbf{x}^t\mathbf{V}\mathbf{x}$, where $\mathbf{V}$ is the covariance matrix. In general, the matrix $\mathbf{V}$ is positive semidefinite, but we will assume that it is positive definite. This is the case if the returns of the assets are linearly independent as random variables.

In these terms, the Markowitz model can be formulated as the following quadratic programming problem, which we abbreviate as *continuous variable problem* (CP) as opposed

to the formulation with semi continuous variables to be introduced later:

$$\begin{aligned} (\mathrm{CP})\ \mathrm{Min}.\quad & \mathbf{x}^t\mathbf{V}\mathbf{x} \\ \text{s.t.}\quad & \mathbf{e}^t\mathbf{x} \geq r, \\ & \mathbf{1}^t\mathbf{x} = 1, \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned} \tag{1.1}$$

Here r is a minimum expected return specified by the investor. The portfolio selection problem can be thought of in a more natural way as a biobjective problem: to minimize risk and to maximize the expected return. Hence, an optimal portfolio selection must provide an *efficient portfolio,* that is, a portfolio providing the maximum expected return for a given admissible risk or—which is the same—the minimum risk for a given desired expected return. The risk-return pairs of all the efficient portfolios form the so-called *efficient frontier* of a given instance of the problem, and so the decision-support techniques designed to assist an investor in selecting a portfolio consist of computing and analyzing the efficient frontier in order to find the efficient portfolio best fitting the investor's preferences about the trade-off between acceptable risk and desired return.

The real world modern portfolio selection problems incorporate into the original Markowitz model many different kinds of additional constraints, reflecting both market conditions and further investor preferences (see, for instance, [3]). Here we address the problem of dealing with the two kinds among these constraints which make the corresponding model more involved from a computational point of view, namely, semicontinuous variable constraints and cardinality constraints. The main feature of models incorporating such constraints is that they are not quadratic (continuous) problems anymore, but become mixed integer (binary) problems. As it will be shown, the efficient frontier of such problems becomes more irregular and new specific computation techniques are required.

Moreover, these irregularities can make the optimal solution of the problem highly sensitive to small variations of the parameters fixed by the investor which are always very vague in nature. Cadenas et al. [4] deal with this issue by means of a fuzzy version of the portfolio selection problem in the continuous variable case. The techniques developed in the present paper make possible to apply those of [4] to the more general and complex problems we are considering here, in which the sensitivity analysis of the solutions is even more necessary. Sensitivity analysis on (continuous variable) quadratic problems has been studied from different points of view. For the specific case of the portfolio selection problem, the sensitivity on the estimations about expected returns and risk levels is dealt with, for instance, in Goldfarb and Iyengar [5]. A general analysis of the optimal value function in quadratic programming can be found in Hadigheh et al. [6]. See also Best and Grauer [7] for the portfolio selection case.

2. On the Computation of the Efficient Frontier

It is well known [2, 6] that the efficient frontier of (CP) is a continuous curve comprising a finite number of arcs of parabola. The usual way of determining it is the so-called ε-constraint method (EC) (see [8]), which can be described as a two-stage procedure.

(EC1) Calculate a sample of the efficient frontier, that is, solve the problem (CP) (or any of its extensions described below) for a sufficiently large number of values of r, ranging from the minimum to the maximum possible return of an efficient portfolio, which are calculated previously. So a "dotted" representation of the efficient frontier is obtained.

(EC2) Interpolate the pairs (risk, return) by any standard interpolation technique to obtain a continuous curve, or even a smooth one, depending on the specific interpolation technique used.

This is what most commercial packages actually do (see [8] for a review of the current software situation). Notice that what really matters is not just obtaining a picture of the efficient frontier but knowing the efficient portfolio corresponding to each of its points. In this way, the ε-constraint method also requires an interpolation of the efficient portfolios calculated in EC1 stage, and this is usually done by linear (vectorial) interpolation, even if the interpolation of EC2 has been nonlinear.

Although the ε-constraint method is the most extensively used procedure [8], it is clear that it provides a limited knowledge of the efficient frontier. In order to take a well-founded decision, it would be very useful to know the change points where an arc of parabola of the efficient frontier joins the next one, since they correspond to different portfolio compositions allowing a richer sensitivity analysis to be made than that provided by the Kuhn-Tucker multipliers (if known) and offering the investor the possibility of choosing among portfolios which are similar in risk and return but different in other characteristics (dividends, social responsibility, etc.) that could be considered decisive when the differences on risk and return are minimal.

In specific terms, an investor wishing for 4% of expected return would accept an efficient portfolio providing just 3.999% if it had better characteristics than the efficient portfolio corresponding to a return of 4%, for instance, if it had a substantially lower risk or a composition that made it preferable for other reasons not reflected in the model because of its secondary importance. This preferable alternative can exist if the initial choice of 4% is near a change point of the efficient frontier.

That is why some attempts can be found in the literature to obtain techniques for computing the exact efficient frontier, that is, for obtaining an analytical—instead of numerical—representation of the frontier, providing the exact efficient portfolio for each risk or return value, the equations of the arcs of parabola, the change points, the Kuhn-Tucker multipliers, and so forth. Markowitz himself provides in [2] the so-called *critical line algorithm,* a simplex-like procedure dealing with quadratic problems, which was distributed later in an excel implementation called *Optimizer,* limited to problems with at most 248 variables [9]. Later, Steuer et al. [8, 10, 11] proposed a completely different algorithm called MPQ (multiparametric quadratic programming) and showed that it is even more powerful than the previous method and can deal with very large instances of (CP). Finally, A. Niedermayer and D. Niedermayer presented in [12] a revised version of the Markowitz algorithm, improving MPQ.

However, all these exact methods are specifically designed for the problem (CP), but when additional constraints are incorporated, the efficient frontier is no longer continuous, and the set of possible risk-return pairs is not convex (see Figure 7 for a "typical" efficient frontier in this context). No method is known for computing the exact efficient frontier of such problems, and the ε-constraint method seems to be the only one available.

Table 1: Comparison between the ε-constraint and the MPQ method (taken from [8]). The first column contains the number of assets, the second one the average CPU-time in seconds for the ε-constraint method calculating a 20-point sample of the efficient frontier (the numbers in italics being estimates), and the third one contains the average CPU-time of the MPQ method calculating the exact efficient frontier.

n	200	400	600	800	1000	1200	1400	1600	1800
ε-constr	152.0	2069.3	*24689*	*54853*					
MPQ	3.7	50.2	237.5	685.5	1108.2	2585.2	3223.5	5478.7	8351.8

Table 2: CPU-times (in seconds per point) of the EC1 stage of the ε-constraint method in the linear and semicontinuous case as a function of the number of assets. The results are mean values obtained by calculating ten 20-point samples corresponding to ten different sets of (real) data, except for the numbers in italics, which have been obtained by solving a single instance of a single problem (computations have been made with GAMS).

Number of assets	100	200	400	600	800	1000
Linear	0.24	0.96	13.27	61.30	172.3	388.2
SC	0.91	14.3	*196*	*623*		

As Tables 1 and 2 (below) show, this method is useless in practice for large instances of (CP), and *a fortiori* for large instances of the much more complex problem with the additional constraints described. However, for medium-sized instances, the standard commercial packages like GAMS [13] or LINGO [14] happen to be powerful enough to deal with the EC1 stage of the ε-constraint method in a few minutes (for instance, for a 100-asset sample, GAMS takes about 7 minutes to calculate a 500-point sample). The purpose of this paper is to make a proposal regarding the EC2 stage.

The point is that all the interpolation methods used to this end vary from the linear interpolation (providing continuous nonsmooth curves) to other classical, relatively simple, interpolation methods providing smooth curves (see [15]). The main disadvantage of these methods is that they are good ways for approximating smooth curves by continuous or smooth curves, but looking for a smooth curve is not a good idea when we know that the true curve we are trying to capture is not even continuous.

More precisely, our proposal is an algorithm for calculating locally exact pieces of the efficient frontier around each point in the sample calculated at the EC1 stage of the procedure. It does provide a sequence of intervals together with the equations of the arcs of parabola composing the efficient frontier in each interval, as well as a pair of vectors parametrizing the corresponding efficient portfolio as a function of the expected return. It does not necessarily obtain the exact efficient frontier, but it provides an analytical interpolation of a given sample which is the best interpolation that can be obtained from it, in the sense that it is locally exact, that is, it is exact in a neighbourhood of each point of the sample. Moreover, for small problems it can be adapted to an enumeration algorithm providing the exact frontier.

3. The KTEF Procedure

Here we describe the kernel of the interpolation procedure that we propose as an alternative for the EC2 stage in the ε-constraint method. It is applied to the following variant of (CP),

where two vectors $\mathbf{l}$ and $\mathbf{u}$ of lower and upper bounds for the assets have been incorporated. Hence, we have a continuous bounded variable problem (CBP):

$$\begin{aligned} \text{(CBP)}\ \text{Min.}\quad & \mathbf{x}^t\mathbf{V}\mathbf{x} \\ \text{s.t.}\quad & \mathbf{e}^t\mathbf{x} \geq r \\ & \mathbf{1}^t\mathbf{x} = 1 \\ & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, \end{aligned} \tag{3.1}$$

Since it is a continuous (quadratic) problem, its optimal solution could be obtained theoretically by algebraically solving its Kuhn-Tucker conditions. In order to write them, we need to introduce the Lagrangian function:

$$\mathcal{L} = \mathbf{x}^t\mathbf{V}\mathbf{x} + \lambda\left(r - \mathbf{e}^t\mathbf{x}\right) + \mu\left(1 - \mathbf{1}^t\mathbf{x}\right) + \boldsymbol{\lambda}^t(\mathbf{l} - \mathbf{x}) + \boldsymbol{\mu}^t(\mathbf{u} - \mathbf{x}), \tag{3.2}$$

where λ, μ are real numbers and $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$ are vectors (λ, μ, λ_i and μ_i being the Kuhn-Tucker multipliers of the problem). Then the Kuhn-Tucker conditions are:

$$\begin{aligned} &\text{primal feasibility: } \mathbf{e}^t\mathbf{x} \geq r, \quad \mathbf{1}^t\mathbf{x} = 1, \quad \mathbf{l} \leq \mathbf{x}, \quad \mathbf{x} \leq \mathbf{u}, \\ &\text{dual feasibility: } \lambda \geq 0, \quad \boldsymbol{\lambda} \geq \mathbf{0}, \quad \boldsymbol{\mu} \leq \mathbf{0}, \\ &\text{stationary point: } 2\mathbf{V}\mathbf{x} - \lambda\mathbf{e} - \mu\mathbf{1} - \boldsymbol{\lambda} - \boldsymbol{\mu} = \mathbf{0}, \\ &\text{complementary slackness: } \lambda\left(r - \mathbf{e}^t\mathbf{x}\right) = 0, \quad \lambda_i(l_i - x_i) = 0, \quad \mu_i(u_i - x_i) = 0, \quad \forall i. \end{aligned} \tag{3.3}$$

We see that all of them are linear equalities or inequalities except for the complementary slackness ones. Each complementary slackness condition splits into two alternative linear equations that, when combined, give rise to $2 \cdot 4^n$ systems of linear equations and inequalities, where n is the number of assets considered (however, since the equations $x_i = l_i$ and $x_i = u_i$ cannot be satisfied simultaneously, they are immediately reduced to $2 \cdot 3^n$).

That is why the explicit resolution of the Kuhn-Tucker conditions is not a viable method, even for a small-sized problem of, for example, 10 variables, the amount of equation systems to be solved being exponentially high. Consequently, this approach is not dealt with in the literature except for very small instances of the problem (such as the two-asset case [16]), or in the simplest case consisting of problem (CP) without the sign constraints, that is, allowing short sales (see [17, 18]). One of the main ideas that we plan to exploit here is that if a solution of the Kuhn-Tucker conditions for a given value of r is known (in our context, as the result of the EC1 stage of the ε-constraint method), such a solution determines a specific case of the complementary slackness conditions, and in turn a single system of linear equations and inequalities that can be solved parametrically on r. The result is an exact piece of the efficient frontier of (CBP).

We have called KTEF the algorithm that (partially) solves in this sense the Kuhn-Tucker conditions to calculate a piece of the efficient frontier. In order to present it, we formulate some preliminary considerations.

Let us call r^- and r^+ the minimum and the maximum expected return of an efficient portfolio. For $r > r^+$ the portfolio selection problem becomes infeasible, whereas for $r < r^-$

the optimal solution is the same as for $r = r^-$. Hence, we can assume that $r^- \leq r \leq r^+$. Bearing in mind that we are assuming the variance-covariance matrix $\mathbf{V}$ to be positive definite, we know that for each level of return $r^- \leq r \leq r^+$ the problem has a unique optimal solution $\mathbf{x}$ (with expected return exactly equal to r), which is its only Kuhn–Tucker point. This implies that the first constraint in (3.1) is satisfied with an equality:

$$\mathbf{e}^t\mathbf{x} = r. \tag{3.4}$$

Hence, when stating the Kuhn-Tucker conditions, we can take this equation as the first primal feasibility condition and delete the first complementary slackness one.

For each variable x_i, the pair of conditions $\lambda_i(l_i - x_i) = 0$, $\mu_i(u_i - x_i) = 0$ gives rise to three possibilities:

$$x_i = l_i, \qquad \mu_i = 0, \qquad x_i = u_i, \qquad \lambda_i = 0, \qquad \lambda_i = \mu_i = 0. \tag{3.5}$$

Hence, in each case, the index set $I = \{1, \ldots, n\}$ splits into three disjoint subsets $I = L \cup U \cup N$, where

$$L = \{i \in I \mid x_i = l_i\}, \qquad U = \{i \in I \mid x_i = u_i\}, \tag{3.6}$$

and $N = I \setminus (L \cup U)$. Let us call one of these cases *degenerate* if it can provide a Kuhn-Tucker point for at most one value of r (considered as a parameter of the model). Notice that every case in which N contains at most one index is degenerate. Indeed, if $N = \emptyset$, the sets L and U determine the whole portfolio $\mathbf{x}$, so that r must be that determined by (3.4). If $N = \{i_0\}$, the value of x_{i_0} is determined by equation

$$\mathbf{1}^t\mathbf{x} = 1, \tag{3.7}$$

and r is again determined by (3.4). Since the Kuhn-Tucker conditions cannot provide an interpolation when the given case is degenerate, KTEF stops as soon as this situation is detected, in particular if N contains less than two indices. Otherwise, from the sets L and U, the KTEF procedure solves the Kuhn-Tucker conditions parametrically on r, that is, it calculates two vectors $\mathbf{g}$ and $\mathbf{h}$ such that the optimal portfolio is

$$\mathbf{x} = \mathbf{g} + r\mathbf{h}, \tag{3.8}$$

for all r varying in a certain interval $[r_{\min}, r_{\mathrm{mix}}]$, also determined by KTEF. Moreover, it also calculates the coefficients a, b, c such that the efficient frontier over the above-mentioned interval is the arc of parabola described by the quadratic equation $ar^2 + br + c$.

Inputs $\mathbf{V}$, $\mathbf{e}$, $\mathbf{l}$, $\mathbf{u}$, L, U.
Step 1 Set $N' = L \cup U$, $N = \{1, \ldots n\} \sim N'$, extract the vectors $\mathbf{e}_N$, $\mathbf{e}_{N'}$ and the submatrices $\mathbf{V}_0$, $\mathbf{W}$ and $\mathbf{Z}$ (see (A.1) in the Appendix), and the vector $\mathbf{b}$ of active bounds.
Step 2 If $\#(N) \leq 1$ the case is degenerate (STOP).
Step 3 Calculate the inverse matrix $\mathbf{V}_0^{-1}$.
Step 4 Calculate A, B, C, D, E, F.
Step 5 Calculate λ_0, λ_1, μ_0, μ_1 according to (A.14).
Step 6 Calculate $\mathbf{g}_N$, $\mathbf{h}_N$ according to (A.11), as well as $\mathbf{g} = (\mathbf{g}_N, \mathbf{b})$, $\mathbf{h} = (\mathbf{h}_N, \mathbf{0})$.
Step 7 Calculate $\boldsymbol{\lambda}_{L0}$, $\boldsymbol{\lambda}_{L1}$, $\boldsymbol{\mu}_{U0}$, $\boldsymbol{\mu}_{U1}$ according to (A.15).
Step 8 Define a set LB of lower bounds for r containing:
(i) $(-EC + A + AF)/C$,
(ii) $(\mathbf{l}_i - \mathbf{g}_i)/\mathbf{h}_i$ for $i \in N$ provided that $\mathbf{h}_i > 0$,
(iii) $(\mathbf{u}_i - \mathbf{g}_i)/\mathbf{h}_i$ for $i \in N$ provided that $\mathbf{h}_i < 0$,
(iv) $-\boldsymbol{\lambda}_{0i}/\boldsymbol{\lambda}_{1i}$ for $i \in L$ provided that $\boldsymbol{\lambda}_{1i} < 0$, (where $\boldsymbol{\lambda}_0 = (\boldsymbol{\lambda}_{L0}, \mathbf{0})$, $\boldsymbol{\lambda}_1 = (\boldsymbol{\lambda}_{L1}, 0)$),
(v) $-\boldsymbol{\mu}_{0i}/\boldsymbol{\mu}_{1i}$ for $i \in U$ provided that $\boldsymbol{\mu}_{1i} < 0$ (where $\boldsymbol{\mu}_0 = (\boldsymbol{\mu}_{U0}, \mathbf{0})$, $\boldsymbol{\mu}_1 = (\boldsymbol{\mu}_{U1}, \mathbf{0})$,
Step 9 Define $r_{\min} = \max LB$.
Step 10 Define a set UB of upper bounds for r containing:
(i) $(\mathbf{l}_i - \mathbf{g}_i)/\mathbf{h}_i$ for $i \in N$ provided that $\mathbf{h}_i < 0$,
(ii) $(\mathbf{u}_i - \mathbf{g}_i)/\mathbf{h}_i$ for $i \in N$ provided that $\mathbf{h}_i > 0$,
(iii) $-\boldsymbol{\lambda}_{0i}/\boldsymbol{\lambda}_{1i}$ for $i \in L$ provided that $\boldsymbol{\lambda}_{1i} < 0$,
(iv) $-\boldsymbol{\mu}_{0i}/\boldsymbol{\mu}_{1i}$ for $i \in U$ provided that $\boldsymbol{\mu}_{1i} > 0$,
Step 11 Define $r_{\max} = \min UB$.
Step 12 If $r_{\min} \geq r_{\max}$ the case is degenerate (STOP).
Step 13 Calculate a, b, c according to (A.19).
Outputs $r_{\min}$, $r_{\max}$, $\mathbf{g}$, $\mathbf{h}$, λ_0, λ_1, $\boldsymbol{\lambda}_{L0}$, $\boldsymbol{\lambda}_{L1}$, $\boldsymbol{\mu}_{U0}$, $\boldsymbol{\mu}_{U1}$, a, b, c.

Algorithm 1: The KTEF procedure.

See Algorithm 1 for the pseudocode of the KTEF-procedure. The details of the calculations, together with the justification that it actually solves the Kuhn-Tucker conditions, can be found in the Appendix. Notice that the output of the algorithm also contains the terms λ_0, λ_1, $\boldsymbol{\lambda}_{L0}$, $\boldsymbol{\lambda}_{L1}$, $\boldsymbol{\mu}_{U0}$, $\boldsymbol{\mu}_{U1}$ which determine the Khun-Tucker multipliers. See the Appendix for their specific meaning.

4. Computing the Efficient Frontier

In this section, we present a KTEF-based procedure, which we call KTEF-S (see Algorithm 2 for the pseudocode), for performing the second stage of the ε-constraint method (EC2) for the portfolio selection problem endowed with semicontinuous variable and cardinality constraints (additional linear constraints can also be included in our proposal without modifying it essentially, but we will not consider it in practice for the sake of clarity).

Semicontinuous Variables

Portfolios with many small nonzero weights are usually considered unacceptable by many investors and, on the other hand, the investor may also impose upper bounds for the sake of

Inputs $\widehat{\mathbf{V}}$, $\widehat{\mathbf{e}}$, $\widehat{\mathbf{l}}$, $\widehat{\mathbf{u}}$, $\{(\mathbf{x}_j, \mathbf{y}_j)\}_{j=1}^{k}$.
For each $\mathbf{j} = \mathbf{1}, \ldots, \mathbf{k}$
(a) Let $\mathbf{x_j}$, $\mathbf{e_j}$, $\mathbf{l_j}$, $\mathbf{u_j}$ be the vectors obtained from $\mathbf{x_j}$, $\widehat{\mathbf{e}}$, $\widehat{\mathbf{l}}$, $\widehat{\mathbf{u}}$, respectively, by deleting the components corresponding to indexes i such that $y_{ji} = 0$.
(b) Calculate L_j and U_j from $\mathbf{x_j}$ according to (3.6)
End
Eliminate the terms in the sequence $\{(\mathbf{x}_j, \mathbf{y}_j)\}_{j=1}^{k}$ giving rise to repeated terms in the sequence $\{(\mathbf{y}_j, L_j, U_j)\}_{j=1}^{k}$.
For each $\mathbf{j} = \mathbf{1}, \ldots, \mathbf{k}$
(a) Let $\mathbf{V_j}$ be the submatrix of $\widehat{\mathbf{V}}$ obtained by deleting the rows and columns for which $y_{ji} = 0$.
(b) Call KTEF ($\mathbf{V_j}$, $\mathbf{e_j}$, $\mathbf{l_j}$, $\mathbf{u_j}$, L_j, U_j), which provides an interval $[r_{\min j}, r_{\max\ j}]$and the coefficients (a_j, b_j, c_j) of the equation of an arc of parabola.
on error (KTEF has stoped in a degenerate case) discard the point.
End
(i) Define the functions $R_j(r)$ given by (4.3).
(ii) Let **Points** $= \{r_{\max\ j} | j = 1, \ldots, k\} \cup \{r_{\min}\}$, where $r_{\min}$ is the minimum of all $\{r_{\min j}\}_j$.
For $j = 1$ to $k - 1$
For $i = j + 1$ **to** k
(a) Let **Roots** be the set of real roots of (4.4).
(b) Append to **Points** any $r \in$ **Roots** satisfying (4.5).
(c) Let **Roots** be the set of real roots of (4.6)
(d) Append to **Points** any $r \in$ **Roots** satisfying $r_{\min j} \le r \le r_{\max\ j}$.
Next i.
Next j.
(i) Order the vector **Points** and eliminate repeated entries.
(ii) Let $m_l = (\mathbf{Points}_l + \mathbf{Points}_{l+1})/2$ for each l.
(iii) Calculate the vector **T** such that T_l is the index j where $\min_j\ R_j(m_l)$ is attained.
(iv) Let **Good** $= \{T_1\}$, let **Change** $= \{r_{\min T_1}\}$.
For $i = 1$ to the length of **T**
If $T_i \neq T_{i+1}$ append to **Good** the index T_{i+1} and append to **Change** the value $\mathbf{Points}_{i+1}$
Next i.
(i) Append to **Change** the last point of **Points**.
(ii) Set $(a_i, b_i, c_i) = (a_{g_i}, b_{g_i}, c_{g_i})$ (where g_i is a short for $\mathbf{Good}_i$).
Outputs **Change**, $\{(a_i, b_i, c_i)\}_{i=1}^{m}$.

Algorithm 2: The KTEF-S algorithm.

diversification. Since it would be absurd to force the portfolio to contain a minimum amount of each possible asset, we need to declare each weight x_i as a semicontinuous variable, that is, allow it to take the value 0 or, in another case, to vary within a given interval $[l_i, u_i]$.

Cardinality Constraints

They appear as diversification constraints, introducing into the model an investor's preferences about how many assets an acceptable portfolio must contain, or even how many assets it must contain from several fixed groups of assets.

Semicontinuous variables can be incorporated into the model by means of auxiliary binary variables, obtaining the following semicontinuous variable problem (SCP):

$$\begin{aligned}
(\text{SCP})\ \text{Min}.\quad & R = \mathbf{x}^t \widehat{\mathbf{V}} \mathbf{x} \\
\text{s.t.}\quad & \widehat{\mathbf{e}}^t \mathbf{x} \geq r \\
& \mathbf{1}^t \mathbf{x} = 1 \\
& \widehat{l}_i y_i \leq x_i \leq \widehat{u}_i y_i, \quad 1 \leq i \leq n, \\
& y_i \in \{0, 1\}.
\end{aligned} \tag{4.1}$$

Here y_i takes the value 1 if the ith asset appears in the portfolio and 0 otherwise. We have added hats to the problem data in order to keep the notation of KTEF when called later. The binary variables y_i can also be used to incorporate the cardinality constraints. For instance, we can impose

$$m \leq \sum_{i=1}^{n} y_i \leq M, \tag{4.2}$$

where m and M are, respectively, a lower and an upper bound on the number of assets composing the portfolio. Similarly, some bounds can be imposed on the number of assets taken from a specific subset. Any such cardinality constraint (i.e., any condition on the binary variables y_i) can be added without altering our method at all.

The input of the KTEF-S algorithm is the output of the first stage of the ε-constraint method (EC1), namely, a dotted sample $\{(\mathbf{x}_j, \mathbf{y}_j)\}_{j=1}^{k}$ of the efficient frontier calculated by means of any suitable procedure (for medium-sized problems, many commercial packages like GAMS or LINGO can be used).

For each point $(\mathbf{x}_j, \mathbf{y}_j)$, we apply KTEF to the instance $P_{\mathbf{y}}$ of the problem (CBP) obtained by removing the variables x_i from (SCP) (with any set of additional cardinality constraints) such that $y_i = 0$. This provides an interval $[r_{\min j}, r_{\max j}]$ and the coefficients (a_j, b_j, c_j) of the equation $a_j r^2 + b_j r + c_j$ of an arc of parabola, which is a piece of the exact efficient frontier of the problem $P_{\mathbf{y}}$. In order to compare the arcs defined on the possibly overlapping intervals $[r_{\min j}, r_{\max j}]$, we extend them to the functions

$$R_j(r) = \begin{cases} a_j r_{\min j}^2 + b_j r_{\min j} + c_j & \text{if } r < r_{\min j}, \\ a_j r^2 + b_j r + c_j & \text{if } r_{\min j} \leq r \leq r_{\max j}, \\ K & \text{if } r_{\max j} < r, \end{cases} \tag{4.3}$$

where K is a large enough number (greater than any possible level of risk). Function $R_j(r)$ provides the lowest level of risk that we can find for a given level of return r from the fact that we know that $(\mathbf{x}_j, \mathbf{y}_j)$ is an efficient portfolio for (SCP) (where $R(r) = K$ means that we cannot find any efficient portfolio from this fact). The best risk we can find for a given r is $R(r) = \min_j R_j(r)$. The last part of the KTEF-S calculates the function $R(r)$, which is the best approximation to the efficient frontier that we can get from the sample.

We want to calculate the function $R(r) = \min_j R_j(r)$ expressed as a sequence of lines and arcs of parabola on a respective sequence of intervals. The extreme points of these intervals (i.e., the points where the minimum of the functions R_j changes from being attained at one index j_0 to being attained at another one j_1) can be of three different kinds.

(1) The intersection point of two arcs of parabola corresponding to two different sample points, that is a point r satisfying

$$a_i r^2 + b_i r + c_i = a_j r^2 + b_j r + c_j. \tag{4.4}$$

Notice that it is also necessary for r to belong to the domains of both parabolas, that is,

$$r_{\min i} \le r \le r_{\max i}, \qquad r_{\min j} \le r \le r_{\max j}. \tag{4.5}$$

(2) The intersection point of an arc of parabola of an $R_j(r)$ with the first constant piece of another $R_{j'}(r)$, that is, a point r satisfying

$$a_j r^2 + b_j r + c_j = a_{j'} r_{\min j'}^2 + b_{j'} r_{\min j'} + c_{j'}, \tag{4.6}$$

with $r_{\min j} \le r \le r_{\max j}$, $j \ne j'$.

(3) The end point of an arc of parabola, that is, one of the points $r_{\max j}$.

Figure 1 shows an example of each type of change point. The KTEF-S procedure calculates the (finite) set *Points* of all points of type 1, 2, 3, so that the set of all change points will be found as a subset of *Points.* For technical reasons, we also include the minimum $r_{\min}$ of all $r_{\min j}$. To select the subset *Change* of change points from the set *Points,* we order *Points* $= \{p_1, p_2, \ldots, p_s\}$ and calculate the middle points $m_k = (p_k + p_{k+1})/2$. Let t_k be the index j where the minimum $\min_j R_j(m_k)$ is attained. Since $m_k < p_{k+1} < m_{k+1}$, if $t_k \ne t_{k+1}$, there must be a change point between m_k and m_{k+1}, which should be p_{k+1}, since it is the only member of *Points* in that interval. Hence the set *Change* can be obtained as the set of the points p_{k+1} such that $t_k \ne t_{k+1}$. Notice that we never check if p_1 really is a change point, but we clearly have $p_1 = r_{\min}$, which cannot be a change point. We also define a set *Good* containing the indexes t_{k+1} such that $t_k \ne t_{k+1}$. Hence, if we enumerate the elements of *Change* as $\{r_1, r_2, \ldots\}$ and those of *Good* as $\{g_1, g_2, \ldots\}$, we have that, for $r \in]r_i, r_{i+1}]$, the minimum $R(r) = \min_j R_j(r)$ is attained at $R_{g_i}(r)$. The data corresponding to indexes outside *Good* can be dismissed.

The output of the procedure consists of the sequence $\{r_i\}$ of change points together with the sequence $\{(a_i, b_i, c_i)\}$ of coefficients of parabola corresponding to the efficient frontier over the interval $]r_i, r_{i+1}]$.

5. Applying the KTEF Procedure to the Continuous Case

Although there are more efficient methods for computing the efficient frontier of a linear constrained (continuous) portfolio selection problem, it should be mentioned that in this case KTEF provides an interesting alternative to the usual ε-constraint method (i.e., the two-stage procedure described in the introduction) which also provides the exact efficient frontier.

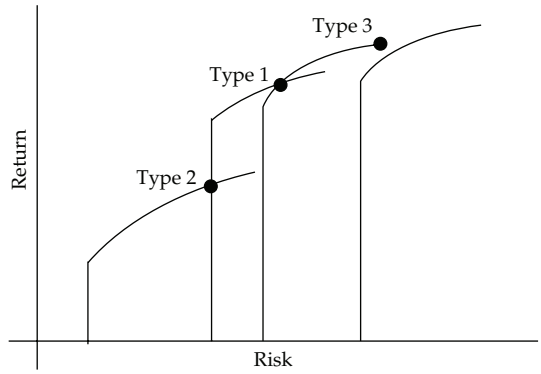

Figure 1: Types of change points.

Inputs **V**, **e**, **l**, **u**
  uses H, R, KTEF
    Set $P = 0$, $J = \{[r_{\min i}, r_{\max i}]\}_{i=1}^{P}$ = an empty sequence,
    $S = \emptyset$
    Set $r^- =$ **eH**(**V**, **e**, **l**, **u**, 0), $r^+ =$ R(**V**, **e**, **l**, **u**)
  (1) Set $k = 1$, $a = r^-$.
While $k \leq P$ **and** $a = r_{\min k}$ **do**
    $k = k + 1$, $a = r_{\max k}$
  If $a < r^+$ **then**
    If $k > P$ **then** $b = r^+$ **else** $b = r_{\min\ k}$
    $r = (a + b)/2$
    else stop
  (2) Set **x** = H(**V**, **e**, **l**, **u**, r)
    Calculate L and U from **x** according to (3.6)
    Call KTEF(**V**, **e**, **l**, **u**, L, U)
      onerror (KTEF has stoped in a degenerate case)
     set $r = (a + r)/2$ **go to (2)**
    Set $S = S \cup \{\text{ktef}(\mathbf{V}, \mathbf{e}, \mathbf{l}, \mathbf{u}, L, U)\}$
    Set $P = P + 1$
    Set $J = J \cup \{[r_{\min}, r_{\max}]\}$ ($r_{\min}$, $r_{\max}$ are part of the output of KTEF). The new interval should be inserted in the right place to preserve the increasing order of the sequence $J = \{[r_{\min i}, r_{\max i}]\}_{i=1}^{N}$
  goto (1)
Output S

Algorithm 3: The KTEF-C algorithm.

The idea is that instead of first calculating a sample for an arbitrary sequence of expected returns, the KTEF algorithm can guide the selection of the sample so that the number of calls to the solver that calculates the sample points is reduced to the minimum necessary to get the exact frontier.

Let us describe this procedure, which we have called KTEF-C, which applies KTEF to the continuous case (see Algorithm 3 for the pseudocode). Besides calling the KTEF procedure, it also uses a subroutine H whose inputs are the data **V**, **e**, **l**, **u** of the model together with a

level of return r and whose output is the efficient portfolio $\mathbf{x}$ for that r. As we have mentioned, the procedures implemented in the usual commercial standard optimization packages such as GAMS or LINGO are widely used for sampling efficient frontiers and they are capable of dealing with any reasonable problem.

At the beginning of the KTEF-C procedure, another subroutine R is called once in order to calculate r^+ as the maximum return that can be attained on the feasible set of (3.1) without the first constraint.

The output of the KTEF-C algorithm is a set S containing a sequence of outputs of KTEF, that is, of the form

$$\left(r_{\min}, r_{\max}, \mathbf{g}, \mathbf{h}, \lambda_0, \lambda_1, \boldsymbol{\lambda}_{L0}, \boldsymbol{\lambda}_{L1}, \boldsymbol{\mu}_{U0}, \boldsymbol{\mu}_{U1}, a, b, c\right), \tag{5.1}$$

where the intervals $[r_{\min}, r_{\max}]$ are almost disjoint (they have at most their endpoints in common) and cover the whole interval $[r^-, r^+]$ of the efficient frontier, the corresponding vectors $\mathbf{x} = \mathbf{g}+r\mathbf{h}$ parametrize the efficient portfolios, and the parabolas ar^2+br+c parametrize the efficient frontier. The rest of the data parametrize the Kuhn-Tucker multipliers.

Notice that the loop starting in the line labeled **(2)** must end after a finite number of iterations, since there is a finite number of possibilities for L and U and each degenerate case corresponds to at most one value of r. Hence, there is just a finite number of possible values for r giving rise to a degenerate case. In practice, the probability of choosing an r corresponding to a degenerate case is very small, so that the error case will never hold.

Each time the main loop (starting in **(1)**) is executed, a new nondegenerate interval $[r_{\min}, r_{\max}]$ is found. Since the number of such nondegenerate intervals is finite (because the number of nondegenerate possibilities for the sets L and U is also finite), the KTEF algorithm always stops, and the number of iterations is exactly the number of nondegenerate intervals composing the efficient frontier, that is, the least necessary number of iterations needed to compute the whole efficient frontier.

Finally, we note that the non-degenerate intervals cover the whole interval $[r^-, r^+]$, since if C denotes the union of such intervals, then C is a closed subset of $[r^-, r^+]$ whose complementary set is finite, and hence closed. The connectedness of the interval implies that $C = [r^-, r^+]$, that is, that all the Kuhn-Tucker points appear in the non-degenerate cases. That is why the degenerate cases can be disregarded.

6. Testing the Algorithms

In this section, we present some computational results in order to test the efficiency of our proposed algorithms. We have used a database of historical data of 1000 assets taken from the Russell 2000 stock market index [19]. The percentage of zero entries of all the covariance matrices considered in our computational proofs oscillates between 10% and 18%, so they are far from being sparse. The EC1 stage of the ε-constraint method has been handled with GAMS and our algorithms for the EC2 stage have been implemented in *Mathematica.*

For the continuous case, it is well known (see Section 1) that there are much more efficient procedures than ε-constraint. For instance, in Table 1 we reproduce a table from [8] comparing the ε-constraint method with the MPQ method proposed by Steuer, Qi and Hirschberger, which calculates the exact efficient frontier. We see that the CPU-times of the MPQ method are substantially better and also that the ε-constraint method becomes inviable for large instances of the problem. We also refer to (Table 12.1) in [12], where two

Table 3: CPU-time in seconds per case processed by the KTEF-S algorithm. The results are mean values obtained from ten 30-point samples corresponding to ten different sets of (real) data (computations have been made with *Mathematica*).

Number of assets	20	30	40	50	60	70	80	90	100	200
CPU-time	0.032	0.023	0.022	0.025	0.03	0.025	0.026	0.029	0.037	0.033

variants of the ε-constraint (one using the so-called "Wolfe-simplex algorithm" and a second one using Matlab) are compared with MPQ, Markowitz's critical line algorithm and the improved version of the latter proposed by those authors. The largest case considered for the ε-constraint method corresponds to a 500-asset instance and the reported Matlab CPU-time is 141.6 seconds per point.

On the other hand, for the semicontinuous case no alternative is known, and the CPU-times of solving mixed integer programs are much greater. Table 2 contains the mean CPU-time per point we have obtained for some instances with a different number of assets in the continuous and semicontinuous case. We have obtained better times than those of [8] for the continuous case (but presumably the MPQ results would be similarly improved by a faster computer and the CPU-times for MPQ in Table 1 are better than ours in any case). In the semicontinuous case, the EC1 stage of the ε-constraint method becomes inviable for 600-asset instances of the problem, and barely useful for 400-asset instances (for which a, say, 20-point sample requires about three and a half hours of computations).

However, these considerations concern to the EC1 stage of the ε-constraint method whereas our algorithms deal with the EC2 stage. Hence, once it is assumed that the ε-constraint method is to be used (because of its simplicity in the continuous case or out of necessity in the semicontinuous one), the only possible comparison would be with the usual interpolation methods. These methods vary from the simple piecewise linear interpolation (i.e., joining a given sequence of dots with straight lines) to methods that are a bit more sophisticated, guaranteeing that the resulting curve will be differentiable (like spline interpolation [15]). These methods are implemented in almost every commercial package, their computational time is negligible, and it is even disregarded when computing CPU-times of the ε-constraint method. Thus, it is obvious that our algorithms for interpolating a given sample of the efficient frontier by means of the Kuhn-Tucker conditions will take necessarily more time than the usual ones, which simply adjust small degree polynomials. Hence, we can only test our algorithms in the sense of granting that, for those instances of the problem for which the ε-constraint method is viable, the CPU-time added by our interpolation method is acceptable in view of the advantages it provides.

In this way, Table 3 contains the CPU-time which needs the KTEF-S algorithm to process one case (i.e., a given choice of sets L, U, and N) as a function of the number of assets. We need to deal with "time per case" because several points of a given sample can correspond to the same case, and hence even starting with equal length samples, the number of processed cases may differ.

Our computations show that the CPU-time of KTEF-S depends polynomially (quadratically, in fact) on the number of cases arising from the input sample. For instance, Figure 2 shows that this function fits almost exactly its quadratic least square approximation in a 50-asset example. We have observed the same almost exact fitting in all cases we have checked. All the obtained parabolas have a very small second derivative. Table 4 shows some equations of the interpolating parabolas we have obtained.

Table 4: Some least-square approximation of the CPU-time (in seconds) of KTEF-S as a function of the number of processed cases.

Number of assets	Least-square quadratic approximation
20	$0.056240 - 0.0101667n + 0.00111423n^2$
50	$0.629131 - 0.0499573n + 0.00111115n^2$
100	$0.051426 - 0.0008316n + 0.00088099n^2$

Table 5: Number of intervals (arcs of parabola)/portfolio compositions found from different samples for several instances of (SCP).

assets	Size of the sample													
	50		100		200		500		1000		3000		5000	
30	**15**	10	**19**	13	**20**	13	**21**	13	**22**	13	**27**	14	**30**	14
30	**23**	15	**37**	20	**51**	22	**62**	25	**67**	26	**73**	26	**80**	26
50	**29**	17	**37**	18	**46**	20	**49**	22	**51**	22	**58**	23	**61**	23
50	**39**	26	**59**	27	**73**	28	**89**	29	**97**	30	**103**	32	**105**	32
50	**33**	17	**42**	19	**52**	23	**63**	25	**65**	25	**69**	25	**70**	25
88	**31**	17	**47**	20	**61**	24	**77**	28	**84**	28	**96**	29	**100**	30
100	**33**	25	**44**	29	**57**	33	**63**	35	**65**	35	**72**	36	**89**	36
100	**20**	15	**23**	17	**27**	18	**33**	20	**34**	21	**36**	21	**38**	21

Let us also remark that the CPU-time corresponding to the calls to KTEF is just a minor percentage of the total CPU-time. For instance, from the 6.46 seconds used to process the 98 different cases generated from a 100-point sample in the instance used to generate Figure 2, only 0.11 correspond to the calls to KTEF. The rest corresponds to the computation of the change points.

7. Analysis of the Efficient Frontier

In this section, we present some examples illustrating the possibilities of analyzing the efficient frontier provided by our algorithms. The main idea is that when the efficient frontier is calculated by means of any of the usual interpolation methods, that is, mathematical techniques for obtaining in the simplest way a continuous or even smooth curve from a finite set of points, the only economical information contained in the result is a finite set of efficient portfolios, since the interpolating arcs have no economical meaning. This suffices to plot the frontier with enough accuracy so that an investor can choose a level of return taking into account the corresponding risk level. On the other hand, the interpolations made by means of our algorithms have a precise economical meaning since, starting from a sufficiently large sample of the frontier, they provide the exact frontier, specifically, a piecewise parametrization of the infinite set of efficient portfolios and in particular the change points, that is, the return values where the composition of the efficient portfolio changes.

This leads to the question of how many points are necessary in order to obtain the exact efficient frontier. In the continuous case, the KTEF-C algorithm determines the exact number of points that are needed, whereas in the semicontinuous case we cannot say anything *a priori.* Table 5 contains the number of arcs of parabola and the number of portfolio compositions

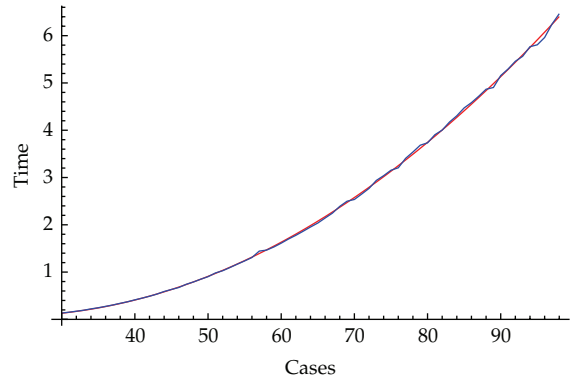

Figure 2: CPU-time in seconds taken by KTEF-S as a function of the number of processed cases for a 50-asset instance, with its least-square quadratic approximation superimposed.

found from different samples for different instances of (SCP). We have checked 50 different instances (they are taken from the database used in previous section, except for the 88-asset case, which is considered in Example 7.1) of several sizes but, since there is no obvious way of aggregating the results, we have opted for showing a few representative cases. Notice that Table 5 shows that the complexity of the frontier is not proportional to the number of assets.

In all the cases, we have considered the difference on the number of compositions found from a 1000-point sample and that found from a 5000-point sample does not exceed two additional cases. This means that in the frontier calculated in the first case, there are a few (very small) intervals where a slightly better composition exists. It is clear that finding these small corrections does not compensate the additional computational effort required by the EC1 stage of the ε-constraint method (we note also that the number of intervals found does not seem to stabilize, but this concerns to the set of constraints being active for each return level, which is of minor interest for an investor) . Hence, our computational results indicate that a 1000-point sample is a reasonable size, at least for 100-asset instances. However, we must remark that, in practice, it is more convenient to draw a rough version of the efficient frontier so that the investor can choose the particular zone where he or she would invest, according to the tradeoff between risk and return he or she considers acceptable, and then calculate a, say, 200- or 500-point sample of that particular zone, providing easily a more accurate description of it that what we could obtain for the whole frontier from a 5000-point sample. In this way, larger instances of SCP can be handled in a reasonable time. In any case, the fact is that postprocessing the sample by using the KTEF-S procedure instead of a typical interpolation offers many advantages with only a small additional CPU-time. The most immediate one is obtained at the very first step of the algorithm, where the sample is filtered to retain just one point for each Kuhn-Tucker complementary slackness case. For instance, as Table 5 shows, a 1000-point sample is immediately reduced to a subsample with less than 100 points without any loss of information at all, since the Kuhn-Tucker conditions applied to the reduced sample provide a representation of the efficient frontier which exactly interpolates all the removed points. Hence, our algorithm makes the previously discussed convenience of working with medium-sized or large samples viable. Moreover, our algorithm not only greatly simplifies the output of the standard ε-constraint

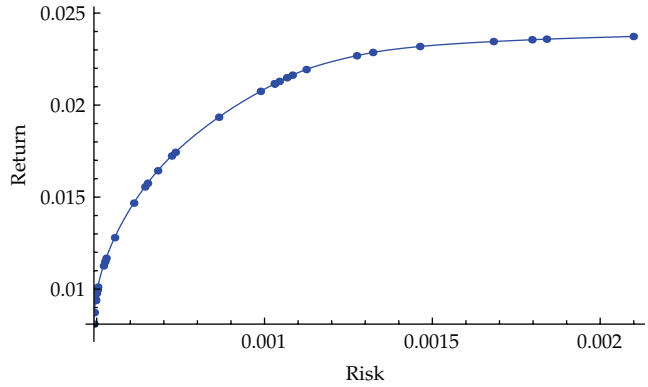

Figure 3: The exact efficient frontier of a problem with 88 securities.

method, but in fact it also structures and analyzes it, providing a functional structured efficient frontier, with exact values for the equations of the arcs of parabola, change points, Kuhn-Tucker multipliers, and so forth. Let us illustrate these facts by means of two specific examples.

Example 7.1. We have computed the efficient frontier of a portfolio selection problem with 88 assets. We have used monthly data over the period January 2001–December 2008 from the *Spanish Stock Exchange Interconnection System* (SIBE) [20], which integrates the four existing security exchanges in Barcelona, Bilbao, Madrid, and Valencia (for the experiment we have used 88 assets that have quoted every month from January 2001 to December 2008. Specifically, the assets are the following: ABE, ABG, ACS, ACX, ADZ, AGS, ALB, AMP, ANA, AND, ASA, AZK, BAY, BBVA, BDL, BES, BKT, BMA, BTO, BVA, CAF, CEP, CPF, CPL, CUN, DGI, DIN, EAD, ECR, ELE, ENC, EVA, FAE, FCC, FER, FUN, GAM, GAS, GCO, GUI, IBE, IBG, IDO, IDR, ITI, JAZ, LGT, MAP, MCM, MDF, MLX, MVC, NAT, NEA, NHH, OHL, PAC, PAS, PAT, POP, PRS, PSG, PVA, RDM, REE, REP, RIO, SAN, SED, SNC, SOL, SOS, SPS, STG, SYV, TEC, TEF, TST, TUB, TUD, UBS, UNF, UPL, URA, VID, VIS, ZEL, ZOT).

Continuous Case

We have considered a continuous instance in which each weight is bounded in the interval $[0, 0.2]$. Figure 3 shows the exact efficient frontier resulting. It comprises 32 arcs of parabola over the intervals $[0.00809875, 0.0237277]$ of expected returns and $[0.000491689, 0.00209849]$ of risk levels.

After applying our algorithm, not only do we have the picture of the efficient frontier but also all the related data about the efficient portfolios and Kuhn-Tucker multipliers. This information can be used to perform a sensitivity analysis of a given solution. For instance, if we set a return level $r = 0.01$, the optimal portfolio contains the following 16 assets: BBVA, BDL, CAF, CEP, CUN, IBE, POP, REE, RIO, SOS, STG, TEF, TST, UNF, UPL, and ZEL. However, in Table 6, we see that this solution is only valid over a very small interval

Table 6: Sensitivity analysis of an efficient portfolio.

Return	Sensitivity interval	Changes in the portfolio	Efficient frontier
	[0.87242, 0.937596]	BBVA exits	$2.39395r^2 - 0.0380155r + 0.000642$
0.9%	[0.937596, 0.942124]	IBG enters	$2.11613r^2 - 0.0328059r + 0.000617$
	[0.942124, 0.97665]	MCM enters	$3.17967r^2 - 0.0528456r + 0.000712$
0.99%	[0.97665, 0.995887]	AND enters	$3.3637r^2 - 0.0564403r + 0.000729$
1%	[0.995887. 1.000181]		$3.63485r^2 - 0.061841r + 0.000756$
1.01%	[1.00018, 1.01099]	None	$2.92292r^2 - 0.0475998r + 0.000685$
1.1%	[1.01099, 1.12637]	ITI enters	$2.8056r^2 - 0.0452276r + 0.000673$

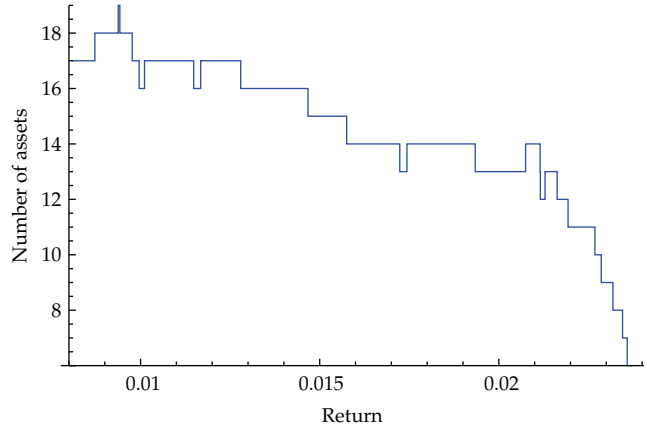

Figure 4: Number of assets in the optimal portfolio.

of returns, namely $[0.995887, 1.00018]$. For r in this interval, the efficient portfolio is given by the expression $\mathbf{x} = \mathbf{g} + r\mathbf{h}$, where

$$\begin{aligned}\mathbf{g} = {} & \{0.21428, -0.105007, -0.122125, -0.0503289, 0.2, 0.131733, 0.207833, 0.024451, 0.2,\\ & -0.0202532, 0.0376479, 0.0820534, 0.0968558, 0.0419524, 0.0223051, 0.0386029\},\\ \mathbf{h} = {} & \{-13.816, 11.0731, 13.6804, 15.9865, 0, -9.47368, -18.3797, 2.93055,\\ & 0, 12.0842, 2.2981, -6.22358, -3.76073, -3.91436, 0.587565, -3.07234\}.\end{aligned} \tag{7.1}$$

For a return level $r = 0.9\%$, the efficient portfolio differs from the original one in four assets. This could be checked just by simply solving the problem for this value of r. However, our additional computations allow us to trace the changes in the efficient portfolio as the return decreases. This is shown in Table 6, where we see that assets AND, MCM, and IBG enter the portfolio successively and that finally asset BBVA exits. On the other hand, the number of assets in the efficient portfolio also grows if we increase the return level, but this is just a local behavior, since, as Figure 4 shows, the number of assets globally decreases as the return increases. Our method guarantees that the analysis is exact and we can see that there are many unstable portfolios in the sense that a small change in r may produce a change in the composition of the portfolio. This analysis can also be used to study the convenience of introducing cardinality constraints into the model. Moreover, the equations parametrizing

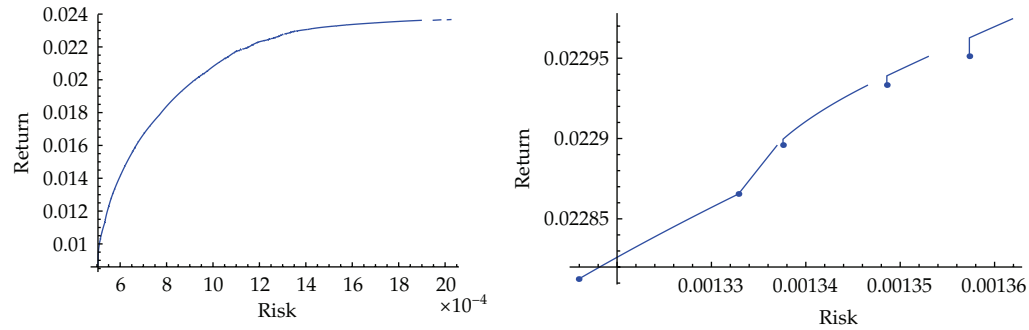

Figure 5: The whole efficient frontier of the problem and a zoom of a part of it.

the efficient frontier (also given in Table 6) provide a sensitivity analysis of the risk with respect to the return level.

Semicontinuous Case

Next we deal with the same data, but considering semicontinuous variables in the range $[0.05, 0.2]$ and cardinality constraints specifying that the total number of assets in a portfolio must vary within the range 5–10. We apply KTEF-S to equally spaced samples of the efficient frontier. The number of intervals found is indicated in Table 5. Figure 5(a) shows the whole efficient frontier calculated from a 600-point sample. The calculations from a 30-point sample provide an almost equal picture, but the larger the sample, the more accurate is the structure we obtain. For instance, Figure 5(b) shows an enlargement of the neighborhood of the return value $r = 0.0229$. We see that the convexity and the continuity that the frontier shows on a large scale fail when examined more closely, and these details are missed when considering a smaller sample.

We note that the economic theory about the portfolio selection problem relies partially on the continuity and convexity of the frontier, which is granted in the continuous case, but fails in the semicontinuous one, and hence it is relevant to know to what extent it can fail in the specific zone of the frontier where the investor intends to choose an efficient portfolio. On the other hand, if there are different portfolio compositions with similar levels of risk and return, an investor could prefer one of them for other reasons beyond these two values. Hence, knowing the variation in portfolio structures along the frontier is also relevant to making a sensitivity analysis of the problem.

Example 7.2. We now consider five assets from the historical data introduced by Markowitz [2], namely, American Tobacco, AT&T, United States Steel, General Motors and Atcheson, and Topeka & Santa Fe. We have established the bounds $0.1 \leq x_i \leq 0.3$.

Continuous Case

For this kind of small problem there is no need to call any optimization package. We can apply KTEF to an enumeration of all possible cases for the sets L and U. More precisely, from the $3^5 = 243$ cases, many of them can be removed *a priori* since they are degenerate leaving

Table 7: Non-degenerate cases.

L	U	$[R_{\min}, R_{\max}]$	$[R_{\min}, R_{\max}]$	Frontier equation
$\{4,5\}$	$\{2\}$	$[0.103289, 0.107478]$	$[0.0400728, 0.0426276]$	$11.6675r^2 - 1.84921r + 0.1066$
$\{5\}$	$\{1,2\}$	$[0.107478, 0.113689]$	$[0.0426276, 0.047546]$	$17.0478r^2 - 2.97854r + 0.165827$
$\{3,5\}$	$\{2\}$	$[0.113689, 0.115711]$	$[0.047546, 0.0505051]$	$122.894r^2 - 26.7285r + 1.49786$
$\{3,5\}$	$\{4\}$	$[0.115711, 0.119233]$	$[0.0505051, 0.057026]$	$27.5594r^2 - 4.62357 + 0.216509$
$\{3\}$	$\{1,4\}$	$[0.119233, 0.1236]$	$[0.057026, 0.067734]$	$84.3641r^2 - 18.0342 + 1.00793$
$\{2,3\}$	$\{3\}$	$[0.1236, 0.124444]$	$[0.067734, 0.0743335]$	$2171.02r^2 - 530.695 + 32.495$

Table 8: An optimal solution.

Risk/return	Investment	Nonnull multipliers
$r = 0.11$	$x_1 = x_2 = 0.3$	$\lambda = 0.771973$
$R = 0.0444663$	$x_3 = 0.159392$	$\lambda_1 = -0.0032799$
	$x_4 = 0.140608$	$\lambda_2 = -0.0369954$
	$x_5 = 0.1$	$\mu_5 = 0.00427664$

just 131 cases. After applying the algorithm, only 6 provide a piece of the efficient frontier. Figure 6 shows the efficient frontier of the problem in which the six intervals are highlighted with dots. These are listed in Table 7 together with their corresponding sets L and U, as well as the equation of the corresponding piece of the efficient frontier.

From these equations, we can calculate the derivative of the frontier or, alternatively, notice that it is just the Kuhn-Tucker multiplier $\lambda(r)$ associated with the return constraint, which is also given by the algorithm. This derivative allows us in turn to study the smoothness of the frontier. Figure 6 shows also the derivative in the present example, and we see that it is discontinuous at all the points where the equation changes, which means that the efficient frontier is not smooth at these points. For instance, the left and right derivatives at the first change point are, respectively

$$R'_{-}(0.107478) = 0.658789, \qquad R'_{+}(0.107478) = 0.685976. \tag{7.2}$$

The difference between them is small, so the discontinuity could be difficult to detect without an exact procedure. On the other hand, the jumps can also be large, as at the last change point, where we have

$$R'_{-}(0.1236) = 2.8206, \qquad R'_{+}(0.1236) = 5.98189. \tag{7.3}$$

This means that starting from a return level near to $r = 0.1236$, the risk of the optimal portfolio is especially sensitive to a small change in r.

The algorithm allows us to calculate the Kuhn-Tucker multipliers. For instance, Table 8 gives the optimal solution for a desired return $r = 0.11$. It contains the optimal values for the variables x_i as well as the minimum risk $R = 0.0444663$. The last row contains the (nontrivial) multipliers, for example λ is the multiplier of the return constraint, that is, the ratio between

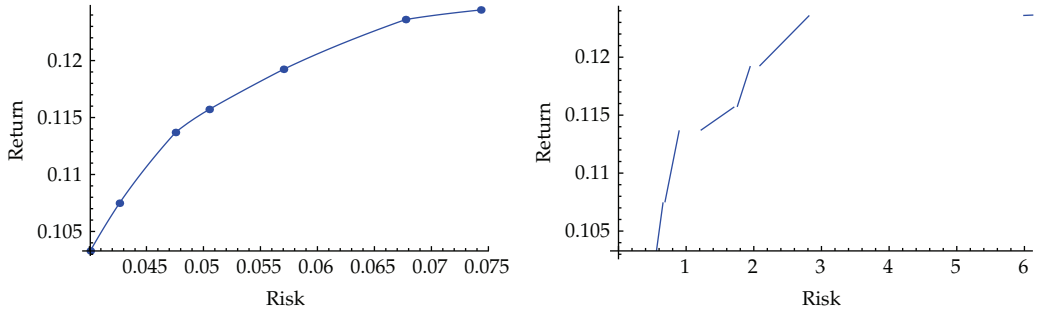

Figure 6: The exact efficient frontier of a five-asset problem (a) and its derivative (b).

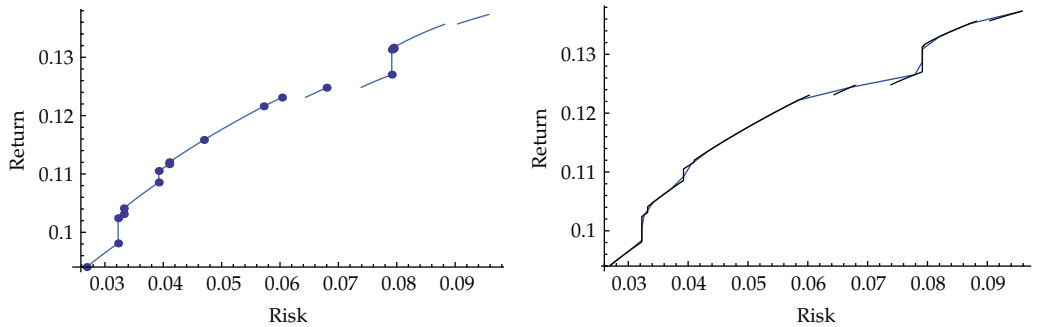

Figure 7: (a) The true efficient frontier. (b) The true frontier and a standard approximation.

the increase in the minimum risk and the increase in the specified desired return. Notice that the multiplier μ of the capital constraint is of no interest since the constant 1 on the right-hand side cannot be modified.

Semicontinuous Case

We have considered semicontinuous variables with bounds $\mathbf{l} = (0.2, 0.3, 0.2, 0.3, 0.2)$, $\mathbf{u} = (0.6, 0.6, 0.6, 0.6, 0.6)$ and the cardinality constraint (4.2) with $m = 2$, $M = 5$. Applying KTEF-S to an equally spaced 20-point sample, we get the frontier shown in Figure 7(a). We know that this is in fact the true frontier since we obtain the same result if we apply the exact algorithm consisting of changing the first loop of KTEF-S by an enumeration of all the possibilities for $\mathbf{V}$, $\mathbf{e}$, $\mathbf{l}$, $\mathbf{u}$, L, U. Figure 7(b) shows the true frontier together with a standard interpolation of the sample, and we see that there are some remarkable differences. The frontier consists of 12 arcs of parabola with 19 change points, such that the interval between two of them corresponds to an arc or to a vertical line.

8. Conclusions

Solving the Kuhn-Tucker conditions is a theoretical way of tackling the portfolio selection problem which can be used in very restrictive cases in practice, since it gives rise to nonacceptable exponential CPU-times. Our results show that, however, the Kuhn-Tucker conditions can be efficiently used as an interpolation procedure for the final stage of the ε-constraint method, since the CPU-times remain quadratic and, moreover, they turn out to be very small when compared with the CPU-time needed for the first stage.

The interpolation algorithms proposed here are very simple conceptually, and they can be implemented by short codes in any general purpose application like Mathematica, Matlab, and so forth (the most complicated operation to perform is the computation of an inverse matrix). This feature, together with the popularity of the ε-constraint method for graphing efficient frontiers, makes our method competitive even with the existing alternatives for the continuous case, since they require more complex implementations not easily available for the economist user that is not a specialist in computation tasks, which, with the aid of our proposal, can get much more than a graph with a relatively small additional CPU-time.

Moreover, our interpolation method has been shown to remain effective when applied to problems with semicontinuous variable and cardinality constraints. For this kind of problems, the ε-constraint is the only known applicable method, and it requires large samples to provide faithful graphs of the frontier. We have shown that, by means of our nontrivial interpolation method, large samples of about 1000 points of the efficient frontier are reduced to a set of less than 100 intervals with its corresponding equations, containing even more information than the original sample.

In general, our procedure provides a simple, structured, analytical expression for the efficient frontier, which is easier to handle than the sample it is calculated from.

For small-sized problems, the analytical description of the frontier obtained with our method is exact, whereas for medium-sized instances, we have shown that a 1000-point sample of the whole frontier (or a proportionally reduced sample of a part of it) provides a reasonable approximation of the exact frontier in the sense that larger samples provide very few additional portfolio compositions (valid for very short intervals) that do not compensate the additional computational effort.

In the continuous case, our method determines the minimal sample that is needed to obtain the exact frontier.

For very small instances (with no more than seven or eight assets), it can be adapted to an exact enumeration algorithm (not to be confused with KTEF-C or KTEF-S) that does not require any sample of the efficient frontier. This can be useful for academic purposes.

We have illustrated by two examples the advantages and possibilities provided by our proposal. In general, the computational results show that the shape of the efficient frontier is different in small and medium-sized instances.

(i) For small-sized problems (notice that many private investors are interested in selecting portfolios from a small-sized set of assets), although they are computationally simple to handle, the shape of the efficient frontier can present many irregularities (discontinuities and sudden changes of slope) which must be taken into account since the risk of an efficient portfolio can be very sensitive to the selected expected return. Hence, the information provided by our method could make an investor move his or her choice to a safer or a more profitable one.

(ii) As the number of assets increases, the efficient frontier becomes more regular at a large scale, and hence its "microscopic" irregularities are not relevant. However, very small changes in the selected expected return can alter the composition of the corresponding optimal portfolio. In this case, our procedure provides the investor with the intervals where each composition is efficient, so that he or she can select according to additional preferences among several options practically indistinguishable with respect to risk and return.

Appendix

Justification of the KTEF Procedure

Let us decompose an arbitrary vector $\mathbf{v} = (\mathbf{v}_N, \mathbf{v}_{N'})$, where $\mathbf{v}_N$ is the vector consisting of the components v_i with $i \in N$ and $\mathbf{v}_{N'}$ consists of those with $i \in L \cup U$. In particular, we have $\mathbf{x} = (\mathbf{x}_N, \mathbf{b})$, where $\mathbf{b}$ is the vector of active bounds, that is, $\mathbf{b}_i$ is l_i (resp., u_i) if $i \in L$ (resp. $i \in U$). Similarly, we decompose

$$\mathbf{V} = \left(\begin{array}{c|c} \mathbf{V}_0 & \mathbf{W} \\ \hline \mathbf{W}^t & \mathbf{Z} \end{array}\right), \tag{A.1}$$

where $\mathbf{V}_0$ contains the rows and columns of $\mathbf{V}$ corresponding to the indexes $i \in N$, $\mathbf{Z}$ contains those corresponding to indexes $i \in L \cup U$ and $\mathbf{W}$ has the rows of those $i \in N$ and the columns of those $i \in L \cup U$. In these terms, (3.4) and (3.7) become

$$\begin{aligned} \mathbf{e}_N^t \mathbf{x}_N &= r - \mathbf{e}_{N'}^t \mathbf{b}, \\ \mathbf{1}_N^t \mathbf{x}_N &= 1 - \mathbf{1}_{N'}^t \mathbf{b}. \end{aligned} \tag{A.2}$$

Similarly, the stationary point conditions for indexes $i \in N$ become

$$2\mathbf{V}_0 \mathbf{x}_N + 2\mathbf{W}\mathbf{b} - \lambda \mathbf{e}_N - \mu \mathbf{1}_N = \mathbf{0}. \tag{A.3}$$

Solving this equation gives

$$\mathbf{x}_N = \frac{\lambda}{2} \mathbf{V}_0^{-1} \mathbf{e}_N + \frac{\mu}{2} \mathbf{V}_0^{-1} \mathbf{1}_N - \mathbf{V}_0^{-1} \mathbf{W}\mathbf{b}. \tag{A.4}$$

Premultiplying by $\mathbf{e}_N^t$ and $\mathbf{1}_N^t$ and using (A.2), we obtain, respectively,

$$B\lambda + A\mu = r + E \tag{A.5}$$

$$A\lambda + C\mu = 1 + F, \tag{A.6}$$

where

$$\begin{aligned} A &= \frac{1}{2}\ \mathbf{1}_N^t \mathbf{V}_0^{-1}\mathbf{e}_N = \frac{1}{2}\ \mathbf{e}_N^t \mathbf{V}_0^{-1}\mathbf{1}_N, \\ B &= \frac{1}{2}\ \mathbf{e}_N^t \mathbf{V}_0^{-1}\mathbf{e}_N, \\ C &= \frac{1}{2}\ \mathbf{1}_N^t \mathbf{V}_0^{-1}\mathbf{1}_N, \\ E &= -\mathbf{e}_{N'}^t \mathbf{b} + \mathbf{e}_N^t \mathbf{V}_0^{-1}\mathbf{W}\mathbf{b}, \\ F &= -\mathbf{1}_{N'}^t \mathbf{b} + \mathbf{1}_N^t \mathbf{V}_0^{-1}\mathbf{W}\mathbf{b}. \end{aligned} \tag{A.7}$$

Solving (A.5) and (A.6) gives

$$\lambda = \frac{rC + EC - A - AF}{D}, \qquad \mu = \frac{B + BF - Ar - AE}{D}, \tag{A.8}$$

where

$$D = BC - A^2. \tag{A.9}$$

Of course, this requires D not to be 0. In this way, notice first that since V_0^{-1} is positive-definite, $B > 0$ and $C > 0$. Also

$$(A\mathbf{e}_N - B\mathbf{1}_N)^t \mathbf{V}_0^{-1}(A\mathbf{e}_N - B\mathbf{1}_N) = BD. \tag{A.10}$$

Hence, we will have $D > 0$ provided that $A\mathbf{e}_N - B\mathbf{1}_N = 0$. But clearly, $A\mathbf{e}_N - B\mathbf{1}_N = \mathbf{0}$ only if $\mathbf{e}_N = e_0\mathbf{1}$ for a certain $e_0 \in \mathbb{R}$, and therefore (A.2) give $e_0(1 - \mathbf{1}_{N'}^t \mathbf{b}) = r - \mathbf{e}_{N'}^t \mathbf{b}$.

Thus, we see that we are considering a degenerate case, since it can only provide a Kuhn-Tucker point for at most one value of r (that given by the previous equation). Therefore, we can assume that $D > 0$.

Incorporating (A.8) into (A.4), we obtain $\mathbf{x}_N = \mathbf{g}_N + r\mathbf{h}_N$, where

$$\begin{aligned} \mathbf{g}_N &= \left(\frac{EC - A - AF}{2D}\right)\mathbf{V}_0^{-1}\mathbf{e}_N + \left(\frac{B + BF - AE}{2D}\right)\mathbf{V}_0^{-1}\mathbf{1}_N - \mathbf{V}_0^{-1}\mathbf{W}\mathbf{b}, \\ \mathbf{h}_N &= \frac{C}{2D}\ \mathbf{V}_0^{-1}\mathbf{e}_N - \frac{A}{2D}\mathbf{V}_0^{-1}\mathbf{1}_N. \end{aligned} \tag{A.11}$$

Therefore, (3.8) is fulfilled taking $\mathbf{g} = (\mathbf{g}_N, \mathbf{b})$, $\mathbf{h} = (\mathbf{h}_N, \mathbf{0})$. Moreover, decomposing $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_L, \mathbf{0})$ and $\boldsymbol{\mu} = (\boldsymbol{\mu}_U, \mathbf{0})$, where $\boldsymbol{\lambda}_L$ (resp., $\boldsymbol{\mu}_U$) denotes the part of $\boldsymbol{\lambda}$ (resp., $\boldsymbol{\mu}$) corresponding to the indexes of L (resp. U), we can calculate $\boldsymbol{\lambda}_L$ and $\boldsymbol{\mu}_U$ from the stationary point conditions corresponding to the indexes in L and U, respectively,

$$\begin{aligned} \boldsymbol{\lambda}_L &= 2(\mathbf{V}\mathbf{x})_L - \lambda\mathbf{e}_L - \mu\mathbf{1}_L, \\ \boldsymbol{\mu}_U &= 2(\mathbf{V}\mathbf{x})_U - \lambda\mathbf{e}_U - \mu\mathbf{1}_U. \end{aligned} \tag{A.12}$$

So we have obtained a unique point $(\mathbf{x}, \lambda, \mu, \boldsymbol{\lambda}, \boldsymbol{\mu})$ satisfying the Kuhn-Tucker equalities for the fixed case. However, it must also satisfy the inequalities to be a Kuhn-Tucker point. In order to make these conditions explicit, let us show how the multipliers depend on r. From (A.8) and incorporating (3.8) into (A.12), we obtain

$$\lambda = \lambda_0 + r\lambda_1, \qquad \mu = \mu_0 + r\mu_1, \qquad \boldsymbol{\lambda}_L = \boldsymbol{\lambda}_{L0} + r\boldsymbol{\lambda}_{L1}, \qquad \boldsymbol{\mu}_U = \boldsymbol{\mu}_{U0} + r\boldsymbol{\mu}_{U1}, \tag{A.13}$$

where

$$\lambda_0 = \frac{EC - A - AF}{D}, \qquad \lambda_1 = \frac{C}{D}, \qquad \mu_0 = \frac{B + BF - AE}{D}, \qquad \mu_1 = -\frac{A}{D}, \tag{A.14}$$

$$\begin{aligned} \boldsymbol{\lambda}_{L0} &= 2(\mathbf{Vg})_L - \lambda_0 \mathbf{e}_L - \mu_0 \mathbf{1}_L, \\ \boldsymbol{\lambda}_{L1} &= 2(\mathbf{Vh})_L - \lambda_1 \mathbf{e}_L - \mu_1 \mathbf{1}_L, \\ \boldsymbol{\mu}_{U0} &= 2(\mathbf{Vg})_U - \lambda_0 \mathbf{e}_U - \mu_0 \mathbf{1}_U, \\ \boldsymbol{\mu}_{U1} &= 2(\mathbf{Vh})_U - \lambda_1 \mathbf{e}_U - \mu_1 \mathbf{1}_U. \end{aligned} \tag{A.15}$$

The Kuhn-Tucker inequalities are

$$\begin{gathered} \mathbf{l}_N \le \mathbf{g}_N + r\mathbf{h}_N, \qquad \mathbf{g}_N + r\mathbf{h}_N \le \mathbf{u}_N, \\ \lambda_0 + r\lambda_1 \ge 0, \qquad \boldsymbol{\lambda}_{L0} + r\boldsymbol{\lambda}_{L1} \ge \mathbf{0}, \qquad \boldsymbol{\mu}_{U0} + r\boldsymbol{\mu}_{U1} \le \mathbf{0}. \end{gathered} \tag{A.16}$$

Equivalently,

$$\begin{gathered} r\mathbf{h}_N \ge \mathbf{l}_N - \mathbf{g}_N, \qquad r\mathbf{h}_N \le \mathbf{u}_N - \mathbf{g}_N, \\ r \ge \frac{-EC + A + AF}{C}, \qquad r\boldsymbol{\lambda}_{L1} \ge -\boldsymbol{\lambda}_{L0}, \qquad r\boldsymbol{\mu}_{U1} \le -\boldsymbol{\mu}_{U0}. \end{gathered} \tag{A.17}$$

These inequalities provide a finite set of lower and upper bounds for r, which in turn determine a closed interval $[r_{\min}, r_{\max}]$. We conclude that the current case provides a Kuhn–Tucker point just for $r_{\min} \le r \le r_{\max}$. If $r_{\max} \le r_{\min}$, we will again have a degenerate case. Finally, the minimum risk for a given return within the interval $[r_{\min}, r_{\max}]$ is given by

$$R(r) = (\mathbf{g} + r\mathbf{h})^t \mathbf{V}(g + r\mathbf{h}) = ar^2 + br + c, \tag{A.18}$$

where

$$a = \mathbf{h}^t \mathbf{V}\mathbf{h}, \qquad b = 2\mathbf{g}^t \mathbf{V}\mathbf{h}, \qquad c = \mathbf{g}^t \mathbf{V}\mathbf{g}. \tag{A.19}$$

The KTEF procedure consists of these calculations arranged as the algorithm described before. Its inputs are the data of the model (3.1) together with the sets of indices L and U, and it provides among its outputs a parametrization $ar^2 + br + c$ of a piece of the efficient frontier of the problem together with the (possibly empty) interval $[r_{\min}, r_{\max}]$ in which it is valid.

Acknowledgment

This paper has been partially supported by project TIN2008-06872-C04-02 from the Ministerio de Ciencia e Innovación of Spain.

References

[1] H. M. Markowitz, "Portfolio selection," *Journal of Finance*, vol. 7, pp. 79–91, 1952.
[2] H. M. Markowitz, *Portfolio Selection: Efficient Diversification of Investments*, vol. 16 of *Cowles Foundation for Research in Economics at Yale University*, John Wiley & Sons, New York, NY, USA, 1959.
[3] W. Hallerbach, H. Ning, A. Soppe, and J. A. Spronk, "A framework for managing a portfolio of socially responsible investments," *European Journal of Operational Research*, vol. 153, no. 2, pp. 517–529, 2004.
[4] J. M. Cadenas, J. V. Carrillo, M. C. Garrido, C. Ivorra, and V. Liern, "Exact and heuristic procedures for solving the fuzzy portfolio selection problem," *Fuzzy Optimization and Decision Making*, vol. 11, no. 1, pp. 29–46, 2012.
[5] D. Goldfarb and G. Iyengar, "Robust portfolio selection problems," *Mathematics of Operations Research*, vol. 28, no. 1, pp. 1–38, 2003.
[6] A. G. Hadigheh, O. Romanko, and T. Terlaky, "Sensitivity analysis in convex quadratic optimization: simultaneous perturbation of the objective and right-hand-side vectors," *Algorithmic Operations Research*, vol. 2, no. 2, pp. 94–111, 2007.
[7] M. J. Best and R. R. Grauer, "Sensitivity analysis for mean-variance portfolio problems," *Management Science*, vol. 37, no. 8, pp. 980–989, 1991.
[8] R. E. Steuer, Y. Qi, and M. Hirschberger, "Portfolio optimization: new capabilities and future methods," *Zeitschrift für Betriebswirtschaft*, vol. 76, no. 2, pp. 199–219, 2006.
[9] H. M. Markowitz and G. P. Todd, *Mean-Variance Analysis in Portfolio Choice and Capital Markets*, Frank J. Fabrozzi Associates, New Hope, Pa, USA, 2000.
[10] M. Hirschberger, Y. Qi, and R. E. Steuer, "Quadratic parametric programming for portfolio selection with random problem generation and computational experience," Working Paper, Departament of Banking and Finance, University of Georgia, Athens, Ga, USA, 2006.
[11] R. E. Steuer, Y. Qi, and M. Hirschberger, "Suitable-portfolio investors, nondominated frontier sensitivity, and the effect of multiple objectives on standard portfolio selection," *Annals of Operations Research*, vol. 152, no. 1, pp. 297–317, 2007.
[12] A. Niedermayer and D. Niedermayer, "Applying Markowitz's critical line algorithm," in *Handbook of Portfolio Construction*, chapter 12, pp. 383–400, Springer, New York, NY, USA, 2010.
[13] http://www.gams.com/dd/docs/bigdocs/gams2002/mccarlgamsuserguide.pdf.
[14] http://www.lindo.com/downloads/PDF/LINGO12.pdf.
[15] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK, 3rd edition, 2007.
[16] H. H. Müller, "Modern portfolio theory: some main results," *ASTIN Bulletin*, vol. 19S, pp. 9–27, 1989.
[17] R.C. Grinold and R. N. Kahn, *Active Portfolio Management*, McGraw-Hill, New York, NY, USA, 1999.
[18] C.-F. Huang and R. H. Litzenberger, *Foundations for Financial Economics*, North Holland, Amsterdam, The Netherlands, 1988.
[19] http://www.russell.com/indexes/data/fact_sheets/us/russell_2000_index.asp.
[20] http://www.sbolsas.es/.