Optimization Methods and Software
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/goms20

A concave optimization-based approach for sparse portfolio selection

D. Di Lorenzo [a], G. Liuzzi [b], F. Rinaldi [c], F. Schoen [a] & M. Sciandrone [a]

[a] Dipartimento di Sistemi e Informatica, Università di Firenze, Via di Santa Marta 3, 50139, Firenze, Italy

[b] Istituto di Analisi dei Sistemi e Informatica, Consiglio Nazionale delle Ricerche, Viale Manzoni 30, 00185, Roma, Italy

[c] Dipartimento di Informatica e Sistemistica, Sapienza Università di Roma, via Ariosto, 25, 00185, Roma, Italy

Available online: 15 Aug 2011

To cite this article: D. Di Lorenzo, G. Liuzzi, F. Rinaldi, F. Schoen & M. Sciandrone (2011): A concave optimization-based approach for sparse portfolio selection, Optimization Methods and Software, DOI:10.1080/10556788.2011.577773

To link to this article: http://dx.doi.org/10.1080/10556788.2011.577773

Optimization Methods & Software
iFirst, 2011, 1–18

Taylor & Francis
Taylor & Francis Group

A concave optimization-based approach for sparse portfolio selection

D. Di Lorenzo[a]*, G. Liuzzi[b], F. Rinaldi[c], F. Schoen[a] and M. Sciandrone[a]

[a]*Dipartimento di Sistemi e Informatica, Università di Firenze, Via di Santa Marta 3, 50139 Firenze, Italy;* [b]*Istituto di Analisi dei Sistemi e Informatica, Consiglio Nazionale delle Ricerche, Viale Manzoni 30, 00185 Roma, Italy;* [c]*Dipartimento di Informatica e Sistemistica, Sapienza Università di Roma, via Ariosto, 25, 00185 Roma, Italy*

(*Received 8 February 2010; final version received 26 March 2011*)

This paper considers a portfolio selection problem in which portfolios with minimum number of active assets are sought. This problem is motivated by the need of inducing sparsity on the selected portfolio to reduce transaction costs, complexity of portfolio management, and instability of the solution. The resulting problem is a difficult combinatorial problem. We propose an approach based on the definition of an equivalent smooth concave problem. In this way, we move the difficulty of the original problem to that of solving a concave global minimization problem. We present as global optimization algorithm a specific version of the monotonic basin hopping method which employs, as local minimizer, an efficient version of the Frank–Wolfe method. We test our method on various data sets (of small, medium, and large dimensions) involving real-world capital market from major stock markets. The obtained results show the effectiveness of the presented methodology in terms of global optimization. Furthermore, also the out-of-sample performances of the selected portfolios, as measured by Sharpe ratio, appear satisfactory.

Keywords: zero-norm programming; concave programming; Frank–Wolfe method; basin hopping method

1. Introduction

Portfolio selection theory studies how to allocate an investor's available capital into a prefixed set of assets with the aims of maximizing the expected return and minimizing the investment risk.

We denote by n the number of available assets, by $\mu \in R^n$ the vector of expected returns of the assets, and by $Q \in R^{n\times n}$ the symmetric positive semidefinite matrix whose generic element q_{ij} is the variance of returns of assets i and j. Usually, both the vector μ and the matrix Q are not known analytically but can be estimated using historical data.

Let us assume that one unit of capital is available and that we want it to be fully invested. Then, let $x \in R^n$ be the vector of decision variables, where x_i is the fraction of the available capital to be invested into asset i, with $i = 1, \ldots, n$. Since the available capital is to be entirely used for

*Corresponding author. Email: dilorenzo@dsi.unifi.it

ISSN 1055-6788 print/ISSN 1029-4937 online

DOI: 10.1080/10556788.2011.577773
http://www.informaworld.com

investment and no short positions are allowed, vector x must satisfy the constraints

$$\mathrm{e}^{\mathrm{T}}x = 1, \quad x \geq 0,$$

where $e \in R^n$ denotes the column vector of all ones. Then, by this notation, $\mu^{\mathrm{T}}x$ is the expected return of the portfolio and $x^{\mathrm{T}}Qx$ is the variance of the portfolio which can be used as a measure of the risk connected with the investment [20]. Hence, the classical Markowitz portfolio selection model [20] seeks for solutions that minimize the risk ($x^{\mathrm{T}}Qx$) while maximizing the expected return ($\mu^{\mathrm{T}}x$) of the portfolio.

In the traditional Markowitz portfolio selection model [20], this bi-objective problem is transformed into a single-objective optimization problem, where the objective function is the risk of the portfolio ($x^{\mathrm{T}}Qx$), and the expected return ($\mu^{\mathrm{T}}x$) of the assets is fixed to a chosen value. Formally, the optimization problem is stated as the following convex quadratic programme

$$\begin{aligned} \min_{x \in R^n} \quad & x^{\mathrm{T}}Qx \\ \text{s.t.} \quad & \mu^{\mathrm{T}}x = \beta \\ & \mathrm{e}^{\mathrm{T}}x = 1 \\ & x \geq 0, \end{aligned} \tag{1}$$

where β is the desired expected return of the portfolio. The main drawback of problem (1) is that means and covariances of the assets are not sufficiently accurate since they have to be estimated from historical (and typically) noisy data. Indeed, it is extremely difficult to estimate the mean returns to working precision and this is a known phenomenon referred to as *mean blur* [18,21]. Besides, the mean–variance model (1) is very sensitive to distributional input parameters. As a result, the model amplifies any estimation error, thus yielding extreme portfolios which, as it can be seen, perform poorly in out-of-sample tests [5,10,12,22]. Several variants of the Markowitz model have been proposed and many attempts have been undertaken to ease the mentioned amplification of estimation errors and yield more stable portfolios. In [3], Bayesian estimation of means and covariances has been employed. In [8,16], additional portfolio constraints have been imposed in the model in order to guide the optimization process towards more diversified portfolios. In [9], the use of a James–Steiner estimator for the means has been proposed which steers the optimal allocations towards the minimum-variance portfolio. The employment of robust estimators has been investigated in [12]. In [2,6,11,13], an important class of portfolio selection problems has been defined by limiting the number of assets to be held in the portfolio so as to reduce both the transaction costs and the complexity of portfolio management. Such a constraint, as argued in [6], helps inducing sparsity of the selected portfolio and can be a remedy to the high instability of classic methods for portfolio selection.

In particular, in [1,2,7], the Markowitz model has been modified by adding to problem (1) a constraint on the number of assets that can be held in the portfolio. This kind of problems, usually called *cardinality constrained portfolio selection problems*, are stated as follows:

$$\begin{aligned} \min_{x \in R^n} \quad & x^{\mathrm{T}}Qx \\ \text{s.t.} \quad & \mu^{\mathrm{T}}x = \beta \\ & \mathrm{e}^{\mathrm{T}}x = 1 \\ & x \geq 0, \\ & \|x\|_0 \leq K, \end{aligned} \tag{2}$$

where

- $\|x\|_0$ is the so-called zero-norm of x and indicates the number of nonzero components of x;
- the parameter K is the chosen limit of assets to be held in the portfolio.

Optimal portfolios with a limited number of assets can be achieved by fixing parameter K in (2) to a sufficiently small value. A different approach for the search of sparse portfolios consists in replacing the l_0 norm in (2) with the more tractable l_1 norm [11], or (equivalently) in adding, as a tunable penalty term, the l_1 norm of x to the objective function (1) [6].

Summarizing, we can say that modern portfolio selection problems involve three partially conflicting objectives:

(a) the risk connected with the portfolio selection ($x^{\mathrm{T}}Qx$) to be minimized,
(b) the expected return of the portfolio ($\mu^{\mathrm{T}}x$) to be maximized,
(c) the number of assets ($\|x\|_0$) held in the portfolio to be minimized.

As such, we have different alternatives to model the portfolio selection problem as a single-objective optimization problem.

In this work, we focus on the *sparsest portfolio*, that is, on the following nonsmooth optimization problem

$$\begin{aligned} \min_{x\in R^n} \quad & \|x\|_0 \\ \text{s.t.} \quad & \mu^{\mathrm{T}}x = \beta \\ & x^{\mathrm{T}}Qx \le \alpha \\ & \mathrm{e}^{\mathrm{T}}x = 1 \\ & x \ge 0, \end{aligned} \tag{3}$$

where β is the desired expected return of the portfolio and α is the maximum acceptable level of risk.

As shown in [2], problem (3) is a difficult, in fact NP-hard, combinatorial problem. Following [19,24], we choose to tackle it by replacing the nonsmooth objective function $\|x\|_0$ with a suitable smooth concave approximating function. This leads to an 'equivalent' (in a sense to be specified later) concave optimization problem. In this way, we move the difficulty of solving (3) to that of solving a concave global minimization problem. As a global optimization strategy, we adopt the monotonic basin hopping (MBH) method [15,17] employing as local minimization procedure an efficient version of the Frank–Wolfe method [14], useful to solve large-dimensional problems.

We observe that the use of the l_1 norm in place of the objective function $\|x\|_0$ can guarantee the recovery of sparse solutions. However, as shown in [4,24], the solutions which are obtained this way, although easily obtainable thanks to the convex nature of the problem, are far less sparse than those obtainable via concave approximations.

Summarizing, the main contributions of the work are:

- the development of a theoretical analysis aimed to prove the equivalence between a general class of zero-norm minimization problems (including (3)) and smooth concave minimization problems; in particular, a new equivalence result has been proven which extends previous results to nonpolyhedral feasible sets;
- the design and implementation of a specific version of the MBH global optimization method that can be applied to efficiently solve, as pointed out by the numerical experiments, the class of portfolio selection problems here considered; in particular, this version of the MBH method is different from previously known ones as it is specifically tailored to this class of problems:

a specific perturbation operator, which is the crucial part of any MBH algorithm, has been designed for this problem and has greatly improved the performance of the method.

The paper is organized as follows. In Section 2, we present the concave optimization-based approach to transform the combinatorial problem (3) into a theoretical equivalent smooth problem. In Section 3, we briefly describe a version [23] of the well-known Frank–Wolfe method employed as local optimizer within our global optimization framework. The proposed global optimization algorithm is presented in Section 4. The results of the numerical experiments are reported in Section 5. Finally, Section 6 contains some concluding remarks.

2. Concave formulations of zero-norm minimization problems

In this section, we describe an approach for transforming a zero-norm minimization problem, which is nonsmooth, into an equivalent (in some sense) smooth concave optimization problem. The approach used here is very general and can be applied not only to the portfolio selection problem, but to any optimization problem which involves the minimization of the zero-norm function over a compact set.

Hence, let us consider the problem

$$\begin{aligned} \min_{x \in R^n} \quad & \|x\|_0 \\ & x \in S, x \geq 0, \end{aligned} \tag{4}$$

where we assume that $S \subset R^n$ is a compact set.

In order to illustrate the idea underlying the concave approach, we observe that the objective function of problem (4) can be written as follows:

$$\|x\|_0 = \sum_{i=1}^{n} s(|x_i|),$$

where $s : R \to R^+$ is the *step function* such that $s(t) = 1$ for $t > 0$ and $s(t) = 0$ for $t \leq 0$. The approach was originally proposed in [19] and is based on the idea of replacing the discontinuous step function by a continuously differentiable concave function $1 - \mathrm{e}^{-\alpha t}$, with $\alpha > 0$, thus obtaining a problem of the form

$$\begin{aligned} \min_{x \in R^n} \quad & \sum_{i=1}^{n} (1 - \mathrm{e}^{-\alpha x_i}) \\ & x \in S, x \geq 0. \end{aligned} \tag{5}$$

It has been shown in [19] that, by assuming that S is a *polyhedral set*, the approximating problem (5) is equivalent to the given nonsmooth problem (4), that is, for α sufficiently large, there exists a solution of (5) which yields a solution of the original problem (4).

A similar concave optimization-based approach has been proposed in [25], where the idea is that of using the logarithm function instead of the step function. The adoption of the logarithm function is practically motivated by the fact that, due to the form of the logarithm function, it is better to increase one variable while setting to zero another one rather than doing some compromise between both, and this should facilitate the computation of a sparse solution. This

leads to a concave smooth problem of the form

$$\begin{aligned} \min_{x\in R^n} \quad & \sum_{i=1}^{n} \log(\epsilon + x_i) \\ & x \in S, x \geq 0. \end{aligned} \tag{6}$$

The equivalence of (6) with (4), namely that for ϵ sufficiently small there exists a solution of (6) which yields a solution of the original problem (4), has been proved in [24] under the assumption that S is a polyhedral set.

Here, we remove the assumption that S is a polyhedral set, and we study the equivalence between problem (4) and a problem of the form

$$\begin{aligned} \min_{x\in R^n} \quad & \sum_{i=1}^{n} f^u(x_i) \\ & x \in S, x \geq 0, \end{aligned} \tag{7}$$

where $f^u : R^+ \to R$ is a smooth function depending on a parameter $u \in U \subseteq R$.

To this aim, we introduce the following assumptions on the parametrized function f^u. There exists $\bar{u} \in U$ such that, for any infinite sequence $\{u_k\} \to \bar{u}$, we have that:

(i) for each $x_i \geq 0$, $\lim_{k\to\infty} f^{u_k}(x_i)$ is well defined,
(ii) for each $x_i > 0$, it follows $f^{u_k}(0) < f^{u_k}(x_i)$ and

$$\lim_{k\to\infty} f^{u_k}(0) < \lim_{k\to\infty} f^{u_k}(x_i) < \infty;$$

(iii) for any $\bar{x}_i > 0$, and for any sequence $\{x_i^k\} \to \bar{x}_i$, we have

$$\lim_{k\to\infty} f^{u_k}(x_i^k) = \lim_{k\to\infty} f^{u_k}(\bar{x}_i);$$

(iv) for each $x_i \geq 0$, one of the following conditions holds: either

$$\lim_{k\to\infty} f^{u_k}(x_i) = \begin{cases} 1 & \text{if } x_i > 0 \\ 0 & \text{if } x_i = 0 \end{cases} \tag{8}$$

or

$$\lim_{k\to\infty} f^{u_k}(0) = -\infty. \tag{9}$$

It can be shown that, setting $U = R^+$, we have that assumptions (i)–(iv) are satisfied, for instance:

- by the function $f^u(x_i) = (1 - \mathrm{e}^{-ux_i})$, with $\bar{u} = +\infty$;
- by the function $f^u(x_i) = \log(u + x_i)$, with $\bar{u} = 0$.

In particular, the function $f^u(x_i) = (1 - \mathrm{e}^{-ux_i})$ satisfies condition (8), and function $f^u(x_i) = \log(u + x_i)$ satisfies condition (9). Note that whenever condition (8) holds, we have

$$\lim_{k\to\infty} \sum_{i=1}^{n} f^{u_k}(x_i) = \|x\|_0. \tag{10}$$

Let $\{u^k\}$ be any sequence convergent to $\bar{u}$. For each k, let x^k be a solution of (7) with $u = u^k$. Thus, by definition, we have for all k and for each $x \in S, x \geq 0$

$$\sum_{i=1}^{n} f^{u^k}(x_i^k) \leq \sum_{i=1}^{n} f^{u^k}(x_i). \tag{11}$$

We prove that any limit point of $\{x^k\}$ is a solution of the original problem (4). In this way, we provide a theoretical justification regarding the transformation of (4) into the class of smooth problems defined by (7).

THEOREM 2.1 *Let $\{u^k\}$ be a sequence such that $\lim_{k\to\infty} u^k = \bar{u}$. Let $\{x^k\}$ be a sequence such that x^k solves problem* (7) *with $u = u^k$. Then, the sequence $\{x^k\}$ admits accumulation points, and all of them solve problem* (4).

Proof Since, for all k, x^k solves problem (7), that is, in particular, $x^k \in S, x^k \geq 0$. Thus, compactness of S implies that $\{x^k\}$ admits accumulation points. Now, let $\bar{x}$ be a limit point of $\{x^k\}$ and $x^\star$ be a solution of (4). By compactness of S, we have that $\bar{x} \in S, \bar{x} \geq 0$.

Assume by contradiction that $\bar{x}$ is not a solution of (4), that is,

$$\|\bar{x}\|_0 \geq \|x^\star\|_0 + 1. \tag{12}$$

Consider any $i \in \{1, \ldots, n\}$ such that $\bar{x}_i > 0$. From assumption (iii), it follows that

$$\lim_{k\to\infty} f^{u_k}(\bar{x}_i) = \lim_{k\to\infty} f^{u_k}(x_i^k) = l_i. \tag{13}$$

Then, given any positive ϵ such that $n\epsilon < 1$, two positive integers $k_1(\epsilon)$ and $k_2(\epsilon)$ exist such that

$$f^{u_k}(\bar{x}_i) \leq l_i + \frac{\epsilon}{2}, \quad \text{for all } k \geq k_1(\epsilon),$$
$$f^{u_k}(x_i^k) \geq l_i - \frac{\epsilon}{2}, \quad \text{for all } k \geq k_2(\epsilon).$$

Thus, for k sufficiently large, we obtain

$$f^{u_k}(\bar{x}_i) \leq f^{u_k}(x_i^k) + \epsilon. \tag{14}$$

Now, let us consider any index $i \in \{1, \ldots, n\}$ such that $\bar{x}_i = 0$. Using assumption (ii), we have, for all k,

$$f^{u^k}(\bar{x}_i) \leq f^{u^k}(x_i^k). \tag{15}$$

From (14) and (15), we get that for k sufficiently large, we can write

$$\sum_{i=1}^{n} f^{u^k}(\bar{x}_i) \leq \sum_{i=1}^{n} f^{u^k}(x_i^k) + n\epsilon. \tag{16}$$

Conditions (11) and (16) imply, as $x^\star \in S, x^\star \geq 0$,

$$\sum_{i=1}^{n} f^{u^k}(\bar{x}_i) \leq \sum_{i=1}^{n} f^{u^k}(x_i^\star) + n\epsilon. \tag{17}$$

Now let us distinguish the two cases.

Case I: Suppose that condition (8) holds. Using (10), we have

$$\lim_{k\to\infty}\sum_{i=1}^{n} f^{u^k}(\bar{x}_i) = \|\bar{x}\|_0$$

$$\lim_{k\to\infty}\sum_{i=1}^{n} f^{u^k}(x_i^\star) = \|x^\star\|_0.$$

Hence, taking the limits for $k \to \infty$ in (17), we obtain

$$\|\bar{x}\|_0 \le \|x^\star\|_0 + n\epsilon.$$

From the above relation and (12), it follows

$$\|x^\star\|_0 + 1 \le \|x^\star\|_0 + n\epsilon,$$

which contradicts the fact that $n\epsilon < 1$.

Case II: Suppose that condition (9) holds. First, we rewrite relation (17) as follows:

$$\sum_{i:\bar{x}_i>0} f^{u_k}(\bar{x}_i) + (n - \|\bar{x}\|_0) f^{u_k}(0) \le \sum_{i:x_i^\star>0} f^{u_k}(x_i^\star) + (n - \|x^\star\|_0) f^{u_k}(0) + n\epsilon,$$

from which, we obtain

$$(\|x^\star\|_0 - \|\bar{x}\|_0) f^{u_k}(0) \le \sum_{i:x_i^\star>0} f^{u_k}(x_i^\star) - \sum_{i:\bar{x}_i>0} f^{u_k}(\bar{x}_i) + n\epsilon.$$

Taking limits for $k \to \infty$, using (12) and condition (9), we get that the left member of the above relation tends to $+\infty$, while the right member tends to a finite value (see assumption (ii)), thus yielding a contradiction. ■

3. Frank–Wolfe method as local optimizer

In this section, we describe an efficient version of the Frank–Wolfe algorithm for minimizing a concave function over a compact convex set and recall some theoretical results about its global convergence (see [23] for further details and proofs). The main motivations for using the Frank–Wolfe algorithm as a local minimizer are the following:

- no need to make a line search when minimizing a concave function over a compact convex set (see Proposition 3.1);
- possibility to reduce the problem dimension at each step of the algorithm, which leads to significant savings in the computational time (see Propositions 3.2 and 3.3).

Let us consider the problem

$$\begin{aligned} \min \quad & f(x) \\ & x \in S, \end{aligned} \tag{18}$$

where $S \subset R^n$ is a nonempty compact convex set having the following form

$$S = \{x \in R^n : \mu^{\mathrm{T}} x = \beta, x^{\mathrm{T}} Q x \le \alpha, \mathrm{e}^{\mathrm{T}} x = 1, 0 \le x\}, \tag{19}$$

and $f : R^n \to R$ is a concave, continuously differentiable function, bounded below on S. A version of the Frank–Wolfe algorithm with unitary stepsize can be described as follows.

Frank–Wolfe-unitary stepsize (FW1) algorithm

1. Let $x^0 \in R^n$ be the starting point;

2. For $k = 0, 1, \ldots,$
if $x^k \notin \arg\min_{x \in S} \nabla f(x^k)^{\mathrm{T}} x$ then compute a solution x^{k+1} of

$$\min_{x \in S} \nabla f(x^k)^{\mathrm{T}} x \tag{20}$$

else exit.

The global convergence property of FW1 algorithm is stated in the following proposition [19].

PROPOSITION 3.1 *Let $\{x^k\}$ be a sequence generated by the Frank–Wolfe unitary stepsize algorithm. Then every limit point $\bar{x}$ of $\{x^k\}$ is a stationary point of problem* (18).

Now consider the problem

$$\begin{aligned} \min \quad & f(x) = \sum_{j=1}^{n} \phi_j(x_j) \\ & x \in S, \end{aligned} \tag{21}$$

where $\phi_j : R \to R$, for $j = 1, \ldots, n$ are concave, continuously differentiable functions. We assume that f is bounded below on S.

We observe that problem (21) includes as special cases the concave programming problems presented in Section 2.

The next proposition shows that, under suitable conditions on the concave functions ϕ_j, the algorithm does not change a variable once it has been fixed to zero.

PROPOSITION 3.2 *Let $\{x^k\}$ be any sequence generated by the Frank–Wolfe algorithm. There exists a value M such that, if*

$$\phi_i'(0) \geq M$$

then we have that for $k \geq 1$

$$x_i^k = 0 \quad \text{implies} \quad x_i^{k+1} = 0.$$

On the basis of Proposition 3.2, it is possible to define the following version of the Frank–Wolfe algorithm with unitary stepsize, where the problems to be solved are of reduced dimension. In particular, whenever a variable is set to zero at an iteration, the method removes this variable for all the following ones.

Frank–Wolfe–unitary stepsize-reduced dimension (FW1-RD) algorithm

1. Let $x^0 \in R^n$ be the starting point.
2. Let $I^0 = \emptyset$, $S^0 = S$ and for $k = 1, \ldots,$ let $I^k = \{i : x_i^k = 0\}$, $S^k = \{x \in S : x_i = 0 \ \forall i \in I^k\}$
if $x^k \notin \arg\min_{x \in S^k} \nabla f(x^k)^{\mathrm{T}} x$ then compute a solution x^{k+1} of

$$\min_{x \in S^k} \nabla f(x^k)^{\mathrm{T}} x \tag{22}$$

else exit.

The special treatment given in the algorithm to the case $k = 0$ accounts for the possibility that the method is started from an infeasible point. Note that problem (22) is equivalent to a problem

of dimension $n - |I^k|$ and that $I^k \subseteq I^{k+1}$, so that the problems to be solved are of nonincreasing dimensions. This yields obvious advantages in terms of computational time. Since algorithm FW1-RD is different from the standard Frank–Wolfe method, its convergence properties cannot be derived from the known result given by Proposition 3.1. Next proposition shows the convergence of the algorithm to a stationary point.

PROPOSITION 3.3 *Let $\{x^k\}$ be a sequence generated by the FW1-RD algorithm. Suppose there exists a value M such that $\phi_j'(0) \geq M$ for $j = 1, \ldots, n$, then every limit point $\bar{x}$ of $\{x^k\}$ is a stationary point.*

Concerning the separable concave objective functions of problems (5) and (6), we have, for $j = 1, \ldots, n$ and $\alpha, \epsilon > 0$,

- $\phi_j(x_j) = f^\alpha(x_j) = 1 - \mathrm{e}^{-\alpha x_j}$ and $\phi_j'(0) = \alpha$;
- $\phi_j(x_j) = f^\epsilon(x_j) = \ln(\epsilon + x_j)$ and $\phi_j'(0) = 1/\epsilon$.

Therefore, the assumption of Proposition 3.3 holds for suitable values of the parameters of the above concave functions, so that algorithm FW1-RD can be applied to solve problems (5) and (6).

4. Global optimization using the basin hopping method

In this section, we present the global optimization method we used to solve the concave optimization problem. The problem introduced in the previous sections is the minimization of a concave objective function over a compact convex region, that is,

$$\begin{aligned} \min_{x \in R^n} \quad & f(x) = \sum_{i=1}^{n} f^u(x_i) \\ \text{s.t.} \quad & \mu^{\mathrm{T}} x = \beta \\ & x^{\mathrm{T}} Q x \leq \alpha \\ & \mathrm{e}^{\mathrm{T}} x = 1 \\ & x \geq 0, \end{aligned} \tag{23}$$

where $f^u : R^+ \to R$ is a concave function depending on a parameter u and satisfying assumptions (i)–(iv) of Section 2.

Although concave minimization is quite a special case of global optimization, with many important properties which may guide towards designing good optimization algorithm, it remains nonetheless an NP-hard problem. Concavity in particular implies that the global optimum will be located at an extreme point of the feasible region; as we saw, the Frank–Wolfe algorithm can be employed to find a stationary point which surely belongs to the frontier of the feasible set. Note that each iteration of the Frank–Wolfe algorithm applied to (23) requires to solve a problem with linear objective function, one convex quadratic constraint, and linear constraints, and this can be efficiently performed by using modern solvers.

However, local optimization is not enough, as the problem under consideration has, in general, many local optima which are not global. Thus, there is a need to employ some global optimization strategy in order to fully exploit the interesting properties of the proposed Frank–Wolfe local method. The most elementary strategy for global optimization is Multistart, which merely consists

of repeatedly running a local optimization method from randomly chosen starting points and retaining the best local optimum found. A few experiments within the context of the problem described in this paper quickly showed that Multistart is too inefficient, leading to an extremely slow and computationally expensive convergence to the global optimum. Our choice was to try a slightly more elaborated method, MBH [15,17]. This is a simple iterated local search algorithm which consists in applying a perturbation to the current locally optimal solution and starting a local search from the perturbed point. If the local search leads to an improvement, then the current local optimum is updated; otherwise, it is left unchanged and the procedure is repeated until, for a prefixed number of iterations, no improvement is observed. This MBH procedure might be considered as a refined local search method and included in a Multistart framework. Thus, several runs of MBH are performed starting from randomly selected points. In order to define a MBH-based method, some procedures have to be defined and some parameters chosen. In particular, we need:

- a procedure $\mathcal{G}()$, which generates a random starting point;
- a procedure $\mathcal{P}(x)$, which generates a perturbed solution in a prescribed neighbourhood of the current point x;
- a procedure $\mathcal{L}(f, X, x)$, which, starting from a point x, produces a local minimum of the objective function f over the feasible set X.

Among the parameters to be chosen, most relevant are the number N of Multistart trials to be performed and the number MNI of iterations without improvement after which the current MBH run is stopped. A simplified scheme for Multistart/MBH is the following:

```
(1)  let f_best := ∞
(2)  for i = 1, ..., N                                  // N Multistart runs
(3)        let x := G()                                 // a random starting point
(4)        let x* := L(x, X, f)                         // a local optimum
(5)        let k := 0
(6)        while (k < MNI) do:                          // start of MBH
(7)              let y := P(x*)                         // perturbation
(8)              let y* := L(y, X, f)                   // local optimization
(9)              if f(y*) < f(x*) then                  // improvement
(10)                   let x* := y*
(11)                   let k := 0
(12)             else                                   // no improvement
(13)                   let k := k + 1
(14)             end if
(15)       end while                                    // end of an MBH run
(16)       if f(x*) < f_best then
(17)             let f_best := f(x*)
(18)             let x_best := x*
(19)       end if
(20) end for
```

The choice of the perturbation procedure $\mathcal{P}$ is one of the keys for the success of MBH; in fact, choosing too small a neighbourhood makes MBH get stuck at a local optimum, wasting local searches. On the other hand, choosing too large a neighbourhood makes MBH behaves like pure Multistart, thus loosing efficiency.

The most frequently used perturbation consists in choosing a radius r and uniformly generating the perturbed point y in a ball of radius r centred at $x^\star$. In this case, it is crucial to choose

a radius r which is neither too small nor too large, the ideal being somewhat larger than the radius (i.e. half the diameter) of the region of attraction of the current local optimum $x^\star$. This strategy, with many different choices for the radius r, has been attempted for the concave portfolio optimization problem, but only with very limited success. A possible explanation might be that the local optima obtained through the Frank–Wolfe procedure belong to quite a large basin of attraction which includes large portions of the boundary of the feasible region; thus, usually, a new local optimization from a nearby point will most likely lead to the same local optimum. More refined generation procedures might be tried, exploiting the fact that perturbed points should be preferably located on the boundary but not too close to the current one. In this paper, however, we preferred a more combinatorial perturbation mechanism which turned out to be quite efficient for the problem under consideration. At each iteration, we choose to 'swap' the values of some variables of the current solution $x^\star$ which are nonzero, with other variables which are currently null. This way we performed a perturbation, in some sense, in a space more closely related to the zero-norm. In the experiments, we choose to swap $\min\{20, 0.5\|x^\star\|_0\}$ pairs.

For what concerns the other procedures, we choose as $\mathcal{L}$ the FW1-RD procedure described before. Finally, some care had to be taken also for the initial random generation $\mathcal{G}$. In fact, the feasible set is the intersection of a simplex with an ellipsoid and with an half-space: generating a uniform point in such a set might not be a trivial task. In our experiments, we choose to uniformly generate a point in the unit simplex surface by means of a standard procedure which consists in the uniform generation of $n-1$ points in the interval $[0, 1]$: the vector composed of the n lengths of the partition of the unit interval induced by these points turns out to be a uniform point in the n-dimensional simplex surface. After this, feasibility with respect to the other constraints was checked and the whole procedure repeated until a feasible point was eventually produced. Thus, the generation tool was a mix of a rigorous generator in the unit simplex coupled with an acceptance/rejection method for the remaining constraints. Apart from pathological cases, this procedure turned out to be quite efficient.

It might be observed that in the generation phase we choose to generate feasible points (and this is motivated from some experiments which showed a significant improvement obtained thanks to feasible point initialization). However, in the perturbation phase, we usually obtain an unfeasible starting point (which, in any case, satisfies the unit simplex constraints). This was considered not to be too harmful as, after a perturbation, usually the local optimization algorithm was able to restore feasibility by moving in the neighbourhood of the current point.

For what concerns other algorithmic choices, we choose a relatively small value for the MNI parameter (equal to 10) and ran Multistart until either 500 calls to the FW1-RD procedure have been made, or until 1 h of CPU time has passed.

5. Numerical results

The aims of the experiments are the following:

(a) to evaluate the efficiency of the proposed algorithm (called MBH/FW1-RD algorithm) in terms of global optimization and of computational time; to this aim, we use a state-of-the-art solver which is able to produce a certificate of optimality;
(b) to assess, on out-of-sample data, the performance of the portfolios obtained from our model and to compare them with classical Markowitz model (1).

The results concerning point (a) will be presented in Section 5.2, while those regarding point (b) will be shown in Section 5.3.

As a concluding remark, we highlight that the original problem (3) can be easily formulated as a mixed-integer quadratically constrained programming problem (MIQCP), which can be solved exactly by a number of solvers. Let us consider the following formulation:

$$\begin{aligned} \min_{x\in R^n, y\in\{0,1\}^n} \quad & \sum_{i=1}^{n} y_i \\ \text{s.t.} \quad & \mu^{\mathrm{T}} x = \beta \\ & x^{\mathrm{T}} Q x \le \alpha \\ & \mathrm{e}^{\mathrm{T}} x = 1 \\ & y_i \ge x_i \ \forall i \\ & x \ge 0. \end{aligned} \tag{24}$$

It can be easily shown that problem (3) is equivalent to problem (24).

5.1 *Implementation details and test problems*

We considered formulation (23) with the logarithmic concave function, namely the problem

$$\begin{aligned} \min_{x\in R^n} \quad & \sum_{i=1}^{n} \log(\epsilon + x_i) \\ \text{s.t.} \quad & \mu^{\mathrm{T}} x = \beta \\ & x^{\mathrm{T}} Q x \le \alpha \\ & \mathrm{e}^{\mathrm{T}} x = 1 \\ & x \ge 0, \end{aligned} \tag{25}$$

with $\epsilon = 10^{-6}$.

We employed CPLEX 12.0 to solve the subproblem defined at each iteration of the Frank–Wolfe method.

In the computational experiments, we used seven publicly available data sets [7,11] for mean–variance portfolio optimization. These test problems refer to the following capital market indexes:

- *6FF* [11] ($n = 6$): six Fama and French (1992) portfolios of firms sorted by size and book-to-market;
- *10IND* [11] ($n = 10$): 10 industry portfolios representing the US stock market;
- *25FF* [11] ($n = 25$): 25 Fama and French (1992) portfolios of firms sorted by size and book-to-market;
- *48IND* [11] ($n = 48$): 48 industry portfolios representing the US stock market;
- *FTSE 100* [7] ($n = 79$): it is a share index published since 1984 of the 100 most highly capitalized UK companies listed on the London stock exchange;
- *S&P 500* [7] ($n = 476$): it is a free-float capitalization-weighted index published since 1957 of the prices of 500 large-cap common stocks actively traded in the USA;
- *NASDAQ* [7] ($n = 2196$): it is a stock market index published since 1971 of all of the common stocks and similar securities listed on the NASDAQ stock market (about 3000 components).

For each test problem, we have

- expected returns;
- covariance matrix.

For the three biggest problems, we also have

- name list of stocks;
- weekly stock price data from March 2003 to March 2008;
- weekly stock return data from March 2003 to March 2008.

Note that, expected returns and covariance matrices for problems FTSE 100, S&P 500 and NASDAQ are calculated using the stock data from March 2003 to March 2007. The remaining data, for the period April 2007–March 2008, are used as out-of-sample data to evaluate the performance of the portfolios obtained with our method (see below for further details).

For each problem, we generated different instances by varying the parameters α and β as follows. For what concerns parameter β, following [7], we define an interval $[\beta_{\min}, \beta_{\max}]$ and then select N_β values equally spaced in the interval, there including the extreme values. The following table reports, for each problem, the values $\beta_{\min}$, $\beta_{\max}$, and N_β.

Data set	$\beta_{\min}$	$\beta_{\max}$	N_β
6FF	1.0425	1.5693	100
10IND	0.9254	1.1574	100
25FF	1.0466	1.6805	100
48IND	0.9217	1.4463	100
FTSE	3.4691×10^{-3}	1.6146×10^{-2}	245
S&P	2.4970×10^{-3}	1.6721×10^{-2}	314
NASDAQ	2.4970×10^{-3}	3.7239×10^{-3}	22

As regards the choice of parameter α, for each value of β, we select the corresponding value of α as the solution of problem (2), where we fix $K = 2$ for data set 6FF, $K = 3$ for data sets 10IND and 25FF, $K = 8$ for data set 8IND. In this way, 100 instances were generated for each of these data sets.

For the remaining three problems FTSE 100, S&P 500, and NASDAQ, used also to evaluate the out-of-sample performance, for each value of β we select five equally spaced values of α in the interval $[\alpha_\beta^{\min}, \alpha_\beta^{\max}]$ where

- $\alpha_\beta^{\min}$ is the minimum risk value, according to the classical Markowitz model, when the expected return is equal to β;
- $\alpha_\beta^{\max}$ is the solution reported in [7] when the expected return is equal to β and $K = 10$, that is a value that allows to obtain solutions with no more than 10 active assets.

This choice seems reasonable since, on the one hand, we are interested in sparse portfolios with a limited risk (as near as possible to the Markowitz lower bound) and, on the other hand, we are interested in comparing the sparsity of our solutions to that of the Markowitz ones. Summarizing, for each of portfolio optimization problems FTSE 100, S&P 500, and NASDAQ, a grid of $5N_\beta$ (α, β) pairs was used to generate the test instances of the experimentation.

5.2 *Global optimization results and comparison with MIQC formulation*

In this section, we compare our method, in term of optimal solution value and of computational time, with a state-of-the-art exact solver for MIQCP problems, that is, CPLEX 12.0.

First, we solved the small-dimensional portfolio problems described in [11] and compared the solutions obtained by algorithm MBH/FW1-RD with those obtained by means of the MIQCP solver CPLEX.

Comparing the results, it turned out that the algorithm was generally able to find the global optima. More specifically,

- for three problems over four (namely 6FF, 10IND, and 48IND), we obtained the certified optimal solution in all the instances;
- for problem 25FF, we obtained the certified optimal solution in 80% of the instances, and a slightly less sparse solution than the one found by CPLEX in all the other cases.

These results confirm the good ability of our method in finding the sparsest portfolio. The computational time spent for solving these small problems was very low both for algorithm MBH/FW1-RD and for CPLEX.

Then we considered the difficult portfolio problems described in [7]. For each of the three test problems, we attempted to solve all the instances of (24) obtained with the values of parameters α and β as described in Section 5.1.

CPLEX was able to exactly solve in a reasonable amount of time only the instances of FTSE problem. We have verified that in all instances of FTSE problem, the solutions found by CPLEX and MBH/FW1-RD algorithm were the same. This behaviour was also quite robust, as only 4.2% of the local minimizations in the Multistart framework returned a solution with a zero-norm value worse than the optimal one.

In order to evaluate the efficiency of our solver, we choose to compare the time our algorithm needs to compute the solution with the time spent by CPLEX to solve problem (24). However, a straightforward comparison would not produce meaningful results, as CPLEX is designed to prove the optimality of the solution, a task which is computationally very expensive and which our algorithm is unable to perform. In order to overcome such difficulty, we adopted the following procedure. For every chosen value of α and β, we execute the algorithm described in Section 4, and we record the best solution $x^*_{\alpha,\beta}$ found and the time spent to find it. Then, we execute the CPLEX solver on the same problem using the formulation described in (24). However, we terminate the programme execution as soon as the current objective function value becomes equal or less than $\|x^*_{\alpha,\beta}\|_0$, and we record again the time spent by the algorithm to find such objective function value. We execute this procedure on the FTSE, S&P, and NASDAQ problems. At the end of such runs, we obtain the time needed by both algorithms in order to find the same objective function value $\|x^*_{\alpha,\beta}\|_0$ on the same problems. Then, we can finally compare the average time spent by the two algorithms which are displayed, in seconds, on the following table.

Data set	Dimension	MBH/FW1-RD	CPLEX integer formulation
FTSE	79	0.24	6.1
S&P	476	80.1	$>$1 day
NASDAQ	2196	2860	$>$1 day

Clearly, MBH/FW1-RD algorithm outperforms CPLEX in terms of computational time. We did not have any exact time for integer formulation on the S&P and NASDAQ problems as, for every tested value of α and β, after 1 day of execution CPLEX was not able to reach the same $K_{\alpha,\beta}$ found by MBH/FW1-RD, so we decided to interrupt its execution. For S&P and NASDAQ problems, we also tried to warm-start CPLEX by supplying it with the best solution found by our

method; however, it turns out that this technique does not help CPLEX in solving the problem to global optimality.

5.3 *Result validation on out-of-sample data*

In this section, we evaluate the out-of-sample performance of the sparse portfolios obtained by using the proposed concave programming strategy. Moreover, we compare these solutions with reference to portfolio solutions obtained by using the classical Markowitz model.

First, let us briefly recall how the historical data have been used in the experimentation. For a given set of assets, available historical data refer to a period of 264 weeks extending from March 2003 to March 2008. The covariance matrix Q and the vector of expected returns μ are computed by using the first 212 weeks of the data (the training period), thus leaving out 52 weeks (from April 2007 to March 2008, the so-called out-of-sample data that we reference to as the testing period) that we use for testing. In particular, the out-of-sample data are used to evaluate the obtained sparse portfolios and to compare them against other reasonable ones (e.g. the Markowitz solutions). We assess the quality of a portfolio solution by means of a one-parameter performance measure. More specifically, let $1, \ldots, P$ be the weeks of the testing period, and let $\mu_1^{\mathrm{T}}x, \ldots, \mu_P^{\mathrm{T}}x$ be the actual returns obtained by portfolio x. We indicate by $\hat{\mu}$ the average of the returns of the testing period, i.e.

$$\hat{\mu}(x) = \frac{1}{P}\sum_{k=1}^{P}\mu_k^{\mathrm{T}}x,$$

and by σ the standard deviation, i.e.

$$\sigma(x) = \sqrt{\frac{1}{P-1}}\sum_{k=1}^{P}(\mu_k^{\mathrm{T}}x - \hat{\mu})^2.$$

Then, to evaluate investment x, we employ the Sharpe ratio (SR)

$$\mathrm{SR}(x) = \frac{\hat{\mu}(x)}{\sigma(x)},$$

which yields a measure of the performance of the portfolio compared to the risk taken (i.e. the higher the Sharpe ratio, the better the performance and the greater the profits for taking on additional risk).

Let us denote, for a given value of β such that (1) is feasible, by x^β the solution of problem (1), i.e. the classical Markowitz problem. Furthermore, for given values of the parameters α and β defining the risk and return constraints, respectively, let us denote by $x^*_{\alpha,\beta}$ the approximate solution of problem (4) determined by the proposed method. In the following, we make a comparison between the solutions x^β of the Markowitz problem for varying values of parameter β and our corresponding sparse solutions $x^*_{\alpha,\beta}$ for varying values of the risk parameter α.

In Figure 1, for each data set and for each $\|x^\beta\|_0$, we report by a squared marker the Markowitz solution which yields the highest SR on the testing period. Similarly, in Figure 1 we report, for each data set and for each value of $\|x^*_{\alpha,\beta}\|_0$ obtained by our method, the solution which yields the highest SR on the testing period. The figures show that for small and medium size data sets, i.e. data sets FTSE and S&P, our solutions and those provided by the Markowitz model are comparable in terms of the number of active assets. As for the quality of the solutions, measured by the Sharpe ratio, we note that our solutions are comparable with the Markowitz ones on the small data set but perform better on the medium one. On the large data set considered, NASDAQ, our solutions

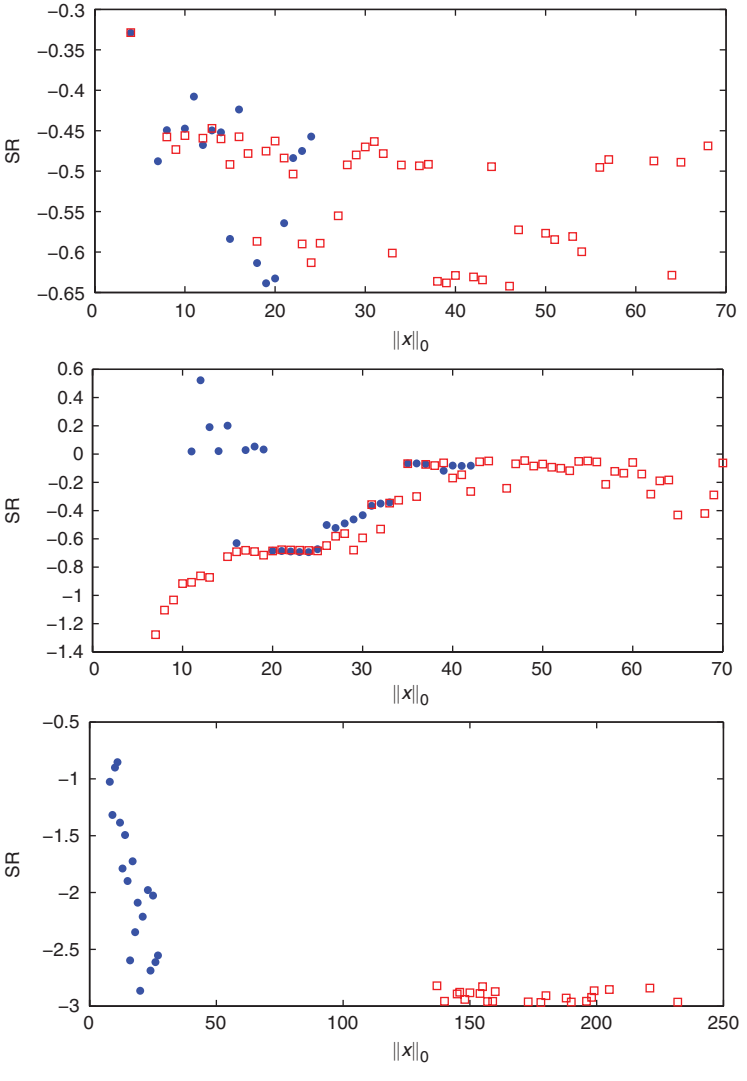

Figure 1. Out-of-sample comparison on the data sets (top to bottom) FTSE, S&P, and NASDAQ between our solutions, denoted by dotted markers, and those provided by the classical Markowitz model, denoted by squared markers.

clearly outperform the Markowitz ones both in terms of achieved sparsity and of portfolio quality as measured by the SR coefficient.

The computational results show that the proposed approach provides sparse and efficient portfolios for suitable values of the expected return and allowable risk. Therefore, it may be a useful tool for financial experts, who are responsible for selecting appropriate values of β and α to obtain a sparse portfolio yielding good performance on out-of-sample data.

6. Concluding remarks

In this paper, we have shown how to design a novel algorithm to solve, in a quite efficient way, the problem of choosing the sparsest portfolio which guarantees prefixed expected return and risk.

From the point of view of optimization algorithms, the proposed approach can be seen as a general method to deal with sparse optimization, not necessarily related to portfolio optimization. Starting from a continuous equivalent formulation of the original discrete problem, the tools employed are Frank–Wolfe method as a local optimizer and iterated local search as a global method. It has been shown that an efficient algorithm can be defined by exploiting a concave formulation. Applying this computational scheme to the sparsest portfolio problem, it has been shown that the achieved computational efficiency is orders of magnitude higher than the one obtained by using a state-of-the-art solver for an equivalent mixed-integer formulation. So the first result in this paper is a new optimization technique whose efficiency is significant for the class of sparse optimization problems over the intersection of a polytope with an ellipsoid is introduced. Moreover, it has been shown that, when applied to optimal portfolio problems, this approach is capable of producing high-quality portfolios, achieving a very good Sharpe ratio, in particular when applied to data sets consisting of a large number of assets.

Thus, it can be safely affirmed that an efficient method capable of producing sparse solutions for the minimum asset portfolio problem has been proposed. Since the method is an heuristic, there is no theoretical guarantee that it can find solutions of good quality and it also does not provide a lower bound for the problem. Anyway, the numerical experimentation on small to medium size problems evidenced that the method is able to find the same optimal solutions found by CPLEX. Future research directions might include the application of sparse optimization tools in different contexts, like, for example, that of image restoration. Further research directions are also possible for portfolio problems in order to deal with more general models.

Acknowledgement

The research of the authors from Università di Firenze has been partially supported by Progetto PRIN 'Nonlinear Optimization, Variational Inequalities and Equilibrium Problems'.

References

[1] D. Bertsimas and R. Shioda, *Algorithms for cardinality-constrained quadratic optimization*, Comp. Optim. Appl. 43 (2009), pp. 1–22.
[2] D. Bienstock, *Computational study of a family of mixed-integer quadratic programming problems*, Math. Program. 74 (1996), pp. 121–140.
[3] F. Black and R. Litterman, *Asset allocation: Combining investor views with market equilibrium*, J. Fixed Income 1 (1991), pp. 7–18.
[4] P.S. Bradley and O.L. Mangasarian, *Feature selection via concave minimization and support vector machines*, in *Machine Learning Proceedings of the Fifteenth International Conference (ICML '98)*, J. Shavlik, ed., Morgan Kaufmann, San Francisco, CA, 1998, pp. 82–90.
[5] M. Broadie, *Computing efficient frontiers using estimated parameters*, Ann. Oper. Res. 45 (1993), pp. 21–58.
[6] J. Brodie, I. Daubechies, C. De Mol, D. Giannone, and I. Loris, *Sparse and stable Markovitz portfolios*, European Central Bank Working Paper Series, No. 36, 2008.
[7] F. Cesarone, A. Scozzari, and F. Tardella, *Efficient algorithms for mean–variance portfolio optimization with hard real-world constraints*, Giornale dell'Istituto Italiano degli Attuari LXXII (2009), pp. 37–56.
[8] V.K. Chopra, *Improving optimization*, J. Invest. 2 (1993), pp. 51–59.
[9] V.K. Chopra, C.R. Hensel, and A.L. Turner, *Massaging mean–variance inputs: Return from alternative investment strategies in the 1980s*, Manag. Sci. 39 (1993), pp. 845–855.
[10] V.K. Chopra and W.T. Ziemba, *The effect of errors in means, variances and covariances on optimal portfolio choice*, J. Portfolio Manag. 19 (1993), pp. 6–11.
[11] V. DeMiguel, L. Garlappi, F.J. Nogales, and R. Uppal, *A generalized approach to portfolio optimization: Improving performance by constraining portfolio norms*, Manag. Sci. 55(5) (2009), pp. 798–812.
[12] V. DeMiguel and F.J. Nogales, *Portfolio selection with robust estimation*, Oper. Res. 57 (2009), pp. 560–577.
[13] L. El Ghaoui, M. Oks, and F. Oustry, *Worst-case value-at-risk and robust portfolio optimization: A conic programming approach*, Oper. Res. 51(4) (2003), pp. 543–556.
[14] M. Frank and P. Wolfe, *An algorithm for quadratic programming*, Naval Res. Logist. Quart. 3 (1956), pp. 95–110.
[15] A. Grosso, M. Locatelli, and F. Schoen, *A population based approach for hard global optimization problems based on dissimilarity measures*, Math. Program. 110 (2007), pp. 373–404.

[16] R. Jagannathan and T. Ma, *Risk reduction in large portfolios: Why imposing the wrong constraints helps*, J. Finance 58 (2003), pp. 1651–1684.
[17] R.H. Leary, *Global optimization on funneling landscapes*, J. Global Optim. 18 (2000), pp. 367–383.
[18] D.G. Luenberger, *Investment Science*, Oxford University Press, New York, NY, 1998.
[19] O.L. Mangasarian, *Machine learning via polyhedral concave minimization*, in *Applied Mathematics and Parallel Computing – Festschrift for Klaus Ritter*, H. Fischer, B. Riedmueller, and S. Schaeffler, eds., Physica-Verlag, Germany, 1996, pp. 175–188.
[20] H.M. Markovitz, *Portfolio selection*, J. Finance 7 (1952), pp. 77–91.
[21] R.C. Merton, *On estimating the expected return on the market: An exploratory investigation*, J. Financ. Econ. 8 (1980), pp. 323–361.
[22] R.O. Michaud, *Efficient Asset Management: A Practical Guide to Stock Portfolio Management and Asset Allocation*, Oxford University Press, New York, NY, 2001.
[23] F. Rinaldi, *Concave programming for finding sparse solutions to problems with convex constraints*, Tech. Rep. 1–8, Dipartimento di Informatica e Sistemistica, Sapienza Università di Roma, 2011, Optimization Methods and Software, in press.
[24] F. Rinaldi, F. Schoen, and M. Sciandrone, *Concave programming for minimizing the zero-norm over polyhedral sets*, Comput. Optim. Appl. 46 (2010), pp. 467–486.
[25] J. Weston, A. Elisseef, and B. Scholkopf, *Use of the zero-norm with linear models and kernel model*, J. Mach. Learn. Res. 3 (2003), pp. 1439–1461.