An Efficient Optimization Approach for a Cardinality-Constrained Index Tracking Problem

Fengmin Xu *   Zhaosong Lu †   Zongben Xu ‡

October 20, 2014 (Revised: March 16, 2015)

Abstract

In the practical business environment, portfolio managers often face business-driven requirements that limit the number of constituents in their tracking portfolio. A natural index tracking model is thus to minimize a tracking error measure while enforcing an upper bound on the number of assets in the portfolio. In this paper we consider such a cardinality-constrained index tracking model. In particular, we propose an efficient nonmonotone projected gradient (NPG) method for solving this problem. At each iteration, this method usually solves several projected gradient subproblems. We show that each subproblem has a closed-form solution, which can be computed in linear time. Under some suitable assumptions, we establish that any accumulation point of the sequence generated by the NPG method is a local minimizer of the cardinality-constrained index tracking problem. We also conduct empirical tests to compare our method with the hybrid evolutionary algorithm [28] and the hybrid half thresholding algorithm [30] for index tracking. The computational results demonstrate that our approach generally produces sparse portfolios with smaller out-of-sample tracking error and higher consistency between in-sample and out-of-sample tracking errors. Moreover, our method outperforms the other two approaches in terms of speed.

Keywords: Index tracking, cardinality constraint, nonmonotone projected gradient method.

*School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, 710049, China (Email: `fengminxu@mail.xjtu.edu.cn`. This author was supported by China NSFC projects (No.11101325) and (No.71371152).

†Department of Mathematics, Simon Fraser University, Burnaby, BC, V5A 1S6, Canada (Email: `zhaosong@sfu.ca`). This author was supported in part by NSERC Discovery Grant.

‡School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, 710049, China (Email: `zbxu@mail.xjtu.edu.cn`.) This author was supported by National 973 Program of China (No.2007CB311002) and China NSFC projects (No.70531030).

1 Introduction

Index tracking aims at replicating the performance and risk profile of a given market index, and constructs a tracking portfolio such that the performance of the portfolio is as close as possible to that of the market index. Index tracking problem has received a great deal of attention in the literature (see, for example, [1, 2, 6, 8, 9, 10, 14, 15, 19, 21, 24, 25, 26, 28, 18]). An obvious approach is by full replication of the index. It, however, can cause high administrative and transaction costs. Also, in the practical business environment, portfolio managers often face business-driven requirements that limit the number of constituents in their tracking portfolio. Therefore, index tracking can reduce transaction costs and avoid detaining small and illiquid assets for the index with a large number of constituents.

In this paper we consider a natural model for index tracking, which minimizes a quadratic tracking error while enforcing an upper bound on the number of assets in the portfolio. When short selling is not allowed, this model can be formulated mathematically as

$$\min_{x\in\Delta_r^u} TE(x) := \|y - Rx\|^2/T. \tag{1.1}$$

Here, $x \in \Re^n$ is the weight vector of n index constituents; $y \in \Re^T$ is a sample vector of portfolio returns over a period of length T; $R \in \Re^{T\times n}$ consists of the sample returns of index constituents over the same period,

$$\Delta_r^u := \left\{ x \in \Re^n : \begin{array}{l} \sum_{i=1}^n x_i = 1, \ \|x\|_0 \leq r \\ 0 \leq x_i \leq u, \ i = 1, \ldots, n \end{array} \right\}, \tag{1.2}$$

$\|x\|_0$ denotes the number of nonzero entries of x; and $u \in [1/r, 1]$ is an upper bound on the weight of each index constituent. The sum of error squares is used here to measure the tracking error between the returns of the index and the returns of a portfolio. We shall mention that another possible tracking error measure is the weighted sum of error squares. Recently, Gao and Li [18] studied a related but different cardinality constrained portfolio selection model, which minimizes the variance of the portfolio subject to a given expected return and a cardinality restriction on the assets. They developed some efficient lower bounding schemes and proposed a branch-and-bound algorithm to solve the model.

Index tracking problem (1.1) involves a cardinality constraint and is generally NP-hard. It is thus highly challenging to find a global optimal solution to this problem. Recently, Fastrich et al. [16] studied a relaxation of (1.1) by replacing the cardinality constraint in (1.1) by imposing an upper bound on the l_q-norm $(0 < q < 1)$ [12] of the vector of portfolio weights. Xu et al. [30] considered a special case of this relaxation with $q = 1/2$ and proposed a hybrid half thresholding algorithm for solving this $l_{1/2}$ regularized index tracking model. Lately, Chen et al. [11] proposed a new relaxation of problem (1.1), which minimizes the l_q-norm regularized tracking error. They also proposed an interior point method to solve the model. On the other hand, a local optimal solution of (1.1) can be found by the penalty decomposition method and the iterative hard thresholding method that were proposed in [22, 23], respectively. However, they are generic methods for a more general class of cardinality-constrained optimization problems. When applied to problem (1.1), these methods may not be efficient since they cannot exploit the specific structure of the feasible region of problem (1.1).

Nonmonotone projected gradient (NPG) methods have widely been studied in the literature, which incorporate the nonmonotone line search technique proposed in [20] into projected gradient methods. For example, Birgin et al. [7] studied the convergence of an NPG method for minimizing a smooth function over a closed convex set. Dai and Fletcher [13] studied a NPG method for solving a box-constrained quadratic programming in which Barzilai and Borwein's scheme [4] is used to choose initial stepsize. Recently, Francisco and Bazán [17] proposed an NPG method for minimizing a smooth objective over a general nonconvex set and showed that it converges a generalized stationary point that is a fixed point of a certain proximal mapping. It is known that NPG methods generally outperform the classical (monotone) projected gradient methods in terms of speed and/or solution quality (see, for example, [7, 13, 3, 27]). In this paper, we propose a simple NPG method for solving problem (1.1). At each iteration, our method usually solves several projected gradient subproblems. By exploiting the specific structure of the feasible region of problem (1.1), we show that each projected gradient subproblem has a closed-form solution, which can be computed in *linear* time. Moreover, we show that any accumulation point of the sequence generated by our method is an optimal solution of a related convex optimization problem. Under some suitable assumption, we further establish that such an accumulation point is a local minimizer of problem (1.1). We also conduct empirical tests to compare our method with the other two approaches proposed in [28, 30] for index tracking. The computational results demonstrate that our approach generally produces sparse portfolios with smaller out-of-sample tracking error and higher consistency between in-sample and out-of-sample tracking errors. Moreover, our method outperforms the other two approaches in terms of speed.

The rest of the paper is organized as follows. In section 2 we propose a nonmonotone projected gradient method for solving a class of optimization problems that include problem (1.1) as a special case and establish its convergence. In section 3 we conduct empirical tests to compare our method with the other two existing approaches for index tracking. We present some concluding remarks in section 4.

2 Nonmonotone projected gradient method

In this section we propose a nonmonotone projected gradient (NPG) method for solving the problem

$$\min_{x \in \Delta_r^u} f(x), \tag{2.1}$$

where Δ_r^u is defined in (1.2) and $f : \Re^n \to \Re$ is Lipschitz continuously differentiable, that is, there is a constant $L_f > 0$ such that

$$\|\nabla f(x) - \nabla f(y)\| \le L_f \|x - y\| \quad \forall x, y \in \Re^n. \tag{2.2}$$

Throughout this paper, $\|\cdot\|$ denotes the standard Euclidean norm. It is clear to see that problem (2.1) includes (1.1) as a special case. Therefore, the NPG method proposed below can be suitably applied to solve problem (1.1).

Nonmonotone projected gradient (NPG) method for (2.1)

Let $0 < L_{\min} < L_{\max}$, $\tau > 1$, $c > 0$, integer $M \ge 0$ be given. Choose an arbitrary $x^0 \in \Delta_r^u$ and set $k = 0$.

1) Choose $L_k^0 \in [L_{\min}, L_{\max}]$ arbitrarily. Set $L_k = L_k^0$.

   1a) Solve the subproblem

   $$x^{k+1} \in \operatorname{Arg}\min_{x \in \Delta_r^u} \left\{ \nabla f(x^k)^T (x - x^k) + \frac{L_k}{2} \|x - x^k\|^2 \right\} \tag{2.3}$$

   1b) If

   $$f(x^{k+1}) \le \max_{[k-M]_+ \le i \le k} f(x^i) - \frac{c}{2} \|x^{k+1} - x^k\|^2 \tag{2.4}$$

   is satisfied, then go to step 2).

   1c) Set $L_k \leftarrow \tau L_k$ and go to step 1a).

2) Set $k \leftarrow k + 1$ and go to step 1).

end

Remark.

(i) When $M = 0$, the sequence $\{f(x^k)\}$ is monotonically decreasing. Otherwise, it may increase at some iterations and thus the above method is generally a nonmonotone method.

(ii) A popular choice of L_k^0 is by the following formula proposed by Barzilai and Borwein [4] (see also [7]):

$$L_k^0 = \max \left\{ L_{\min}, \min \left\{ L_{\max}, \frac{(s^k)^T y^k}{\|s^k\|^2} \right\} \right\},$$

where $s^k = x^k - x^{k-1}$, $y^k = \nabla f(x^k) - \nabla f(x^{k-1})$.

■

We first show that for each outer iteration of the above NPG method, the number of its inner iterations is finite.

Theorem 2.1 *For each $k \geq 0$, the inner termination criterion* (2.4) *is satisfied after at most*

$$\max\left\{\left\lfloor \frac{\log(L_f + c) - \log(L_{\min})}{\log \tau} + 1 \right\rfloor, 1\right\}$$

inner iterations.

Proof. Let $\bar{L}_k$ denote the final value of L_k at the kth outer iteration, and let n_k denote the number of inner iterations for the kth outer iteration. We divide the proof into two separate cases.

Case 1): $\bar{L}_k = L_k^0$. It is clear that $n_k = 1$.

Case 2): $\bar{L}_k < L_k^0$. Let $H(x)$ denote the objective function of (2.3). By the definition of x^{k+1}, we know that $H(x^{k+1}) \leq H(x^k)$, which implies that

$$\nabla f(x^k)^T(x^{k+1} - x^k) + \frac{L_k}{2}\|x^{k+1} - x^k\|^2 \leq 0.$$

In addition, it follows from (2.2) that

$$f(x^{k+1}) \ \leq \ f(x^k) + \nabla f(x^k)^T(x^{k+1} - x^k) + \frac{L_f}{2}\|x^{k+1} - x^k\|^2.$$

Combining these two inequalities, we obtain that

$$f(x^{k+1}) \leq f(x^k) - \frac{L_k - L_f}{2}\|x^{k+1} - x^k\|^2.$$

Hence, (2.4) holds whenever $L_k \geq L_f + c$. This together with the definition of $\bar{L}_k$ implies that $\bar{L}_k/\tau < L_f + c$, that is, $\bar{L}_k < \tau(L_f + c)$. In view of the definition of n_k, we further have

$$L_{\min}\tau^{n_k - 1} \leq L_k^0 \tau^{n_k - 1} = \bar{L}_k < \tau(L_f + c).$$

Hence, $n_k \leq \left\lfloor \frac{\log(L_f+c) - \log(L_{\min})}{\log \tau} + 1 \right\rfloor$.

Combining the above two cases, we see that the conclusion holds. ∎

We next establish convergence of the outer iterations of the NPG method.

Theorem 2.2 *Let $\{x^k\}$ be the sequence generated by the above NPG method. There hold:*

(1) $\{f(x^k)\}$ converges and $\{\|x^k - x^{k-1}\|\} \to 0$.

(2) Let x^* be an arbitrary accumulation point of $\{x^k\}$ and $J^* = \{j : x_j^* \neq 0\}$. Then x^* is a stationary point of the problem

$$\begin{array}{ll} \min\limits_{x} & f(x) \\ s.t. & \sum_{i=1}^n x_i = 1, \ 0 \leq x_j \leq u, \ j \in J^*; \\ & x_j = 0, \ j \notin J^*. \end{array} \tag{2.5}$$

Suppose further that f is convex. Then

(2a) x^* *is a local minimizer of problem* (2.1) *if* $\|x^*\|_0 = r$*;*

(2b) x^* *is a minimizer of problem* (2.5) *if* $\|x^*\|_0 < r$*.*

Proof. (1) Notice that f is continuous in $\Delta = \{x \in \Re^n : \sum_{i=1}^n x_i = 1,\ 0 \le x_i \le u\ \forall i\}$. Since $\{x^k\} \subset \Delta$, it follows that $\{f(x^k)\}$ is bounded below. Let $\ell(k)$ be an integer such that $[k-M]_+ \le \ell(k) \le k$ and

$$f(x^{\ell(k)}) = \max_{[k-M]_+ \le i \le k} f(x^i).$$

It is not hard to observe from (2.4) that $f(x^{\ell(k)})$ is decreasing. Hence, $\lim_{k\to\infty} f(x^{\ell(k)}) = \hat{f}$ for some $\hat{f} \in \Re$. Using this relation, (2.4), and a similar induction argument as used in [29], one can show that for all $j \ge 1$,

$$\lim_{k\to\infty} d^{l(k)-j} = 0, \qquad \lim_{k\to\infty} f(x^{l(k)-j}) = \hat{f},$$

where $d^k = x^{k+1} - x^k$ for all $k \ge 0$. In view of these equalities, the uniform continuity of f over Δ, and a similar argument in [29], we can conclude that $\{f(x^k)\}$ converges and $\{\|x^k - x^{k-1}\|\} \to 0$.

(2) Let x^* be an arbitrary accumulation point of $\{x^k\}$. Then there exists a subsequence $\mathcal{K}$ such that $\{x^k\}_{\mathcal{K}} \to x^*$, which together with $\|x^k - x^{k-1}\| \to 0$ implies that $\{x^{k-1}\}_{\mathcal{K}} \to x^*$. By considering a convergent subsequence of $\mathcal{K}$ if necessary, assume without loss of generality that there exists some index set J such that $x_j^k = 0$ for every $j \notin J, k \in \mathcal{K}$ and $x_j^k > 0$ for all $j \in J, k \in \mathcal{K}$. Let $\bar{L}_k$ denote the final value of L_k at the kth outer iteration. From the proof of Theorem 2.1, we know that $\bar{L}_k \in [L_{\min}, \tau(L_f + c)]$. By the definition of x^k, one can see that x^k is a minimizer of the problem

$$\min_{x \in \Delta_r^u} \left\{ \nabla f(x^{k-1})^T (x - x^{k-1}) + \frac{\bar{L}_{k-1}}{2} \|x - x^{k-1}\|^2 \right\}.$$

Using this fact and the definition of J, one can observe that x^k is also the minimizer of the problem

$$\min_{x \in \Omega} \left\{ \nabla f(x^{k-1})^T (x - x^{k-1}) + \frac{\bar{L}_{k-1}}{2} \|x - x^{k-1}\|^2 \right\}, \tag{2.6}$$

where

$$\Omega = \left\{ x \in \Re^n : \begin{array}{l} \sum_{i=1}^n x_i = 1,\ 0 \le x_j \le u,\ j \in J, \\ x_j = 0,\ j \notin J. \end{array} \right\}.$$

By the first-order optimality conditions of (2.6), we have

$$-\nabla f(x^{k-1}) - \bar{L}_{k-1}(x^k - x^{k-1}) \in \mathcal{N}_\Omega(x^k) \quad \forall k \in \mathcal{K}, \tag{2.7}$$

where $\mathcal{N}_\Omega(x)$ denotes the normal cone of Ω at x. Using $\bar{L}_{k-1} \in [L_{\min}, \tau(L_f + c)]$, $\{x^{k-1}\}_{\mathcal{K}} \to x^*$, $\|x^k - x^{k-1}\| \to 0$, outer continuity of $\mathcal{N}_\Omega(\cdot)$, and taking limits on both sides of (2.7) as $k \in \mathcal{K} \to \infty$, one can obtain that

$$-\nabla f(x^*) \in \mathcal{N}_\Omega(x^*). \tag{2.8}$$

Let $\tilde{\Omega}$ be the feasible region of problem (2.5). Clearly, $J^* \subseteq J$ and hence $\tilde{\Omega} \subseteq \Omega$, which implies that $\mathcal{N}_{\Omega}(x^*) \subseteq \mathcal{N}_{\tilde{\Omega}}(x^*)$. It then follows from (2.8) that $-\nabla f(x^*) \in \mathcal{N}_{\tilde{\Omega}}(x^*)$. Hence, x^* is a stationary point of problem (2.5).

We next prove statements (2a) and (2b) under the assumption that f is convex.

(2a) Suppose that $\|x^*\|_0 = r$ and f is convex. We will show that x^* is a local minimizer of problem (2.1). Let $\epsilon = \min\{x_j^* : j \in J^*\}$,

$$\tilde{\mathcal{O}}(x^*;\epsilon) = \{x \in \tilde{\Omega} : \|x - x^*\| < \epsilon\}, \qquad \mathcal{O}(x^*;\epsilon) = \{x \in \Delta_r^u : \|x - x^*\| < \epsilon\},$$

where $\tilde{\Omega}$ is defined above. Since f is convex and x^* is a stationary point of (2.5), one can conclude that x^* is a minimizer of problem (2.5), which implies that $f(x) \geq f(x^*)$ for all $x \in \tilde{\mathcal{O}}(x^*;\epsilon)$. In addition, using the definition of ϵ and $|J^*| = r$, it is not hard to observe that $\mathcal{O}(x^*;\epsilon) = \tilde{\mathcal{O}}(x^*;\epsilon)$. It then follows that $f(x) \geq f(x^*)$ for all $x \in \mathcal{O}(x^*;\epsilon)$, which implies that x^* is a local minimizer of problem (2.1).

(2b) Suppose that $\|x^*\|_0 < r$ and f is convex. Recall from above that x^* is a stationary point of (2.5). Moreover, notice that problem (2.5) becomes a convex optimization problem when f is convex. Therefore, the conclusion of this statement immediately follows. ∎

One can observe that problem (2.3) is equivalent to

$$x^{k+1} \in \operatorname{Arg}\min_{x \in \Delta_r^u} \left\{ \left\| x - \left(x^k - \frac{1}{L_k} \nabla f(x^k) \right) \right\|^2 \right\},$$

which is a special case of a more general problem

$$\min_{x \in \Delta_r^u} \|x - a\|^2 \tag{2.9}$$

for some $a \in \Re^n$. In the remainder of this section we will show that problem (2.9) has a closed-form solution, and moreover, it can be found in linear time. Before proceeding, we review a technical lemma established in [22].

Lemma 2.1 *Let $\mathcal{X}_i \subseteq \Re$ and $\phi_i : \Re \to \Re$ for $i = 1, \ldots, n$ be given. Suppose that r is a positive integer and $0 \in \mathcal{X}_i$ for all i. Consider the following l_0 minimization problem:*

$$\min \left\{ \phi(x) = \sum_{i=1}^n \phi_i(x_i) : \|x\|_0 \leq r, \ x \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_n \right\}. \tag{2.10}$$

Let $\tilde{x}_i^* \in \operatorname{Arg}\min\{\phi_i(x_i) : x_i \in \mathcal{X}_i\}$ and $I^* \subseteq \{1, \ldots, n\}$ be the index set corresponding to the r largest values of $\{v_i^*\}_{i=1}^n$, where $v_i^* = \phi_i(0) - \phi_i(\tilde{x}_i^*)$ for $i = 1, \ldots, n$. Then x^* is an optimal solution of problem (2.10), where x^* is defined as follows:

$$x_i^* = \begin{cases} \tilde{x}_i^* & \text{if } i \in I^*; \\ 0 & \text{otherwise}, \end{cases} \qquad i = 1, \ldots, n.$$

We are now ready to establish that problem (2.9) has a closed-form solution that can be computed efficiently.

Theorem 2.3 *Given any $a \in \Re^n$, let $I^* \subseteq \{1, \ldots, n\}$ be the index set corresponding to the r largest values of $\{a_i\}_{i=1}^n$. Suppose that $\lambda^* \in \Re$ is such that*

$$\sum_{i \in I^*} \Pi_{[0,u]}(a_i + \lambda^*) = 1, \tag{2.11}$$

where

$$\Pi_{[0,u]}(t) = \begin{cases} 0 & \text{if } t \leq 0; \\ t & \text{if } 0 < t < u; \\ u & \text{if } t \geq u \end{cases} \qquad \forall t \in \Re.$$

Then x^* is an optimal solution of problem (2.9), where x^* is defined as follows:

$$x_i^* = \begin{cases} \Pi_{[0,u]}(a_i + \lambda^*) & \text{if } i \in I^*; \\ 0 & \text{otherwise}, \end{cases} \qquad i = 1, \ldots, n.$$

Proof. Let $d(x)$ and d^* denote the objective function and the optimal value of (2.9), respectively, and x^* be defined above. We can observe that $\|x^*\|_0 \leq r$, $\sum_{i=1}^n x_i^* = 1$ and $0 \leq x_j^* \leq u$ for all j, which implies that x^* is a feasible solution of (2.9), namely, $x^* \in \Delta_r^u$. Hence, $d(x^*) \geq d^*$. Let $\psi(t) = t^2 - (t - \Pi_{[0,u]}(t))^2$ for every $t \in \Re$. It is not hard to see that ψ is differentiable, and moreover,

$$\psi'(t) = 2t - 2(t - \Pi_{[0,u]}(t)) = 2\Pi_{[0,u]}(t) \geq 0.$$

Hence, $\psi(t)$ is increasing in $(-\infty, \infty)$. Let $\phi_i(x_i) = (x_i - a_i - \lambda^*)^2$, $\mathcal{X}_i = [0, u]$, $\tilde{x}_i^* = \arg\min\{\phi_i(x_i) : x_i \in \mathcal{X}_i\}$ and $v_i^* = \phi_i(0) - \phi_i(\tilde{x}_i^*)$ for all i. One can observe that $\tilde{x}_i^* = \Pi_{[0,u]}(a_i + \lambda^*)$ and $v_i^* = \psi(a_i + \lambda^*)$ for all i. By the definition of I^* and the monotonicity of ψ, we conclude that I^* is the index set corresponding to the r largest values of $\{v_i^*\}_{i=1}^n$. In view of Lemma 2.1 and the definitions of x^* and $\tilde{x}^*$, one can see that x^* is an optimal solution to the problem

$$\underline{d}^* = \min_{0 \leq x \leq u, \|x\|_0 \leq r} \left\{ \|x - a\|^2 - 2\lambda^* (\sum_{i=1}^n x_i - 1) \right\},$$

and hence,

$$\underline{d}^* = \|x^* - a\|^2 - 2\lambda^* (\sum_{i=1}^n x_i^* - 1) = \|x^* - a\|^2 = d(x^*).$$

In addition, we can observe that $d^* \geq \underline{d}^*$. It then follows that $d^* \geq d(x^*)$. Recall that $d(x^*) \geq d^*$. Hence, we have $d(x^*) = d^*$. Using this relation and $x^* \in \Delta_r^u$, we conclude that x^* is an optimal solution of problem (2.9). ■

We next show that a λ^* satisfying (2.11) can be computed in linear time, which together with Theorem 2.3 implies that problem (2.9) can be solved in linear time as well.

Theorem 2.4 *For any $a \in \Re^n$ and $u \geq 1/n$, the equation*

$$h(\lambda) := \sum_{i=1}^{n} \Pi_{[0,u]}(a_i + \lambda) - 1 = 0. \tag{2.12}$$

has at least a root λ^*, and moreover, it can be computed in $O(n)$ time.

Proof. One can observe that h is continuous in $(-\infty, \infty)$, and moreover, $h(\lambda) = -1$ when λ is sufficiently small and $h(\lambda) = nu - 1 \geq 0$ when λ is sufficiently large. Hence, (2.12) has at least a root λ^*.

We next show that a root λ^* to (2.12) can be computed in $O(n)$ time. Indeed, it is not hard to observe that h is a piecewise linear increasing function in $(-\infty, \infty)$ with breakpoints $\{-a_1, \ldots, -a_n, -a_1 + u, \ldots, -a_n + u\}$. Suppose that only k of these breakpoints are distinct and they are arranged in strictly increasing order $\{\lambda_1 < \ldots < \lambda_k\}$. The value of h at each λ_i and the slope of each piece of h can be evaluated iteratively. Indeed, let $\lambda_0 = -\infty$. Observe that $h(\lambda) = -1$ for all $\lambda \leq \lambda_1$. Hence, h has slope $s_0 = 0$ in $(-\infty, \lambda_1]$ and $h(\lambda_1) = -1$. Suppose that h has slope s_{i-1} in $(\lambda_{i-1}, \lambda_i]$, and that $h(\lambda_i)$ is already computed, and also that there are m_i number of $\{-a_1, \ldots, -a_n\}$ and n_i number of $\{-a_1 + u, \ldots, -a_n + u\}$ equal to λ_i. Then the slope of h in $(\lambda_i, \lambda_{i+1}]$ is $s_i = s_{i-1} + m_i - n_i$, which yields $h(\lambda_{i+1}) = h(\lambda_i) + s_i(\lambda_{i+1} - \lambda_i)$ for $i = 1, \ldots, k-1$. Since $h(\lambda_1) = -1$, $h(\lambda_k) = nu - 1 \geq 0$ and h is increasing, there exists some $1 \leq j < k$ such that $h(\lambda_j) < 0$ and $h(\lambda_{j+1}) \geq 0$. If $h(\lambda_{j+1}) = 0$, then $\lambda^* = \lambda_{j+1}$ is a root to (2.12). Otherwise, $\lambda^* \in (\lambda_j, \lambda_{j+1})$ and $h(\lambda^*) = 0$. Using these facts and the relation $h(\lambda) = h(\lambda_j) + s_j(\lambda - \lambda_j)$ for $\lambda \in (\lambda_j, \lambda_{j+1})$, we can have

$$\lambda^* = \lambda_j - h(\lambda_j)/s_j.$$

In addition, one can observe that the arithmetic operation cost of this root-finding procedure is $O(n)$. ∎

3 Numerical results

In this section, we conduct numerical experiments to compare the performance of the NPG method proposed in Section 2 with the hybrid evolutionary algorithm [28] and the hybrid half thresholding algorithm [30] for solving index tracking problems. It shall be mentioned that the NPG method solves the l_0 constrained model (1.1) with $u = 0.5$ while the hybrid evolutionary algorithm solves a mixed integer programming model and the hybrid half thresholding algorithm [30] solves an $l_{1/2}$ regularized index tracking model. These three methods were coded in Matlab, and all computations were performed on a HP dx7408 PC (Intel core E4500 CPU, 2.2GHz,1GB RAM) with Matlab 7.9 (R2009b).

The data sets used in our experiments are selected from the standard ones in OR-library [5] and the CSI 300 index from China Shanghai-Shenzhen stock market. For the standard data sets, weekly prices of the stocks from 1992 to 1997 of Hang

Seng (Hong Kong), DAX 100 (Germany), FTSE (Great Britain), Standard and Poor's 100 (USA), the Nikkei index (Japan), the Standard and Poor's 500 (USA), Russell 2000 (USA) and Russell 3000 (USA) are used. For CSI 300 index, the daily prices of 300 stocks from 2011 to 2013 in China stock market are considered. According to the sample scale, we divide the above data sets into two categories: small data sets including Hang Seng, DAX 100, FTSE , Standard and Poor's 100, the Nikkei index; and large data sets including CSI 300, Standard and Poor's 500, Russell 2000 and Russell 3000. As in Torrubiano and Alberto [28], each data set is partitioned into two subsets: a training set and a testing set. The training set, called in-sample set, consists of the first half of the data and is used to compute the optimal index tracking portfolio. We also use the in-sample set and the formula for TE given in (1.1) to calculate the tracking error, which is called in-sample tracking error (TEI) of the portfolio. The testing set, called out-of-sample set, contains the rest of the data and is used to test the performance of the resulting optimal index tracking portfolio. In particular, we use the formula for TE in (1.1) with (R, y) replaced by the out-of-sample set to calculate the tracking error, which is called out-sample tracking error (TEO) of the portfolio. In addition, we denote the true sparsity of the optimal output generated by each method by S_{true}.

For the NPG method, we set $L_{\min} = 10^{-8}$, $L_{\max} = 10^{8}$, $\tau = 2$, $c = 10^{-4}$, and $M = 3$ for small data sets and $M = 5$ for large data sets. For the hybrid half thresholding algorithm, the lower and upper bounds are chosen to be 0 and 0.5, respectively. We terminate these methods when the absolute change of the approximate solutions over two consecutive iterations is below 10^{-6} or the maximum iteration is $10,000$. For the hybrid evolutionary algorithm, we set the lower bound to 0, the upper bound to 0.5, initial population size to 100, mutation probability to 1%, cross probability to 50%, and maximum iterations to $10,000$. In addition, we randomly choose a feasible point of problem (1.1) as a common initial point for these three methods.

In order to measure the out-of-sample performance and the consistency between in-sample and out-of-sample, we introduce the following two criteria.

- Consistency: The consistency between in-sample and out-of-sample tracking errors of a portfolio given by a method A is defined as

  $$Cons(A) = |TEI_A - TEO_A|,$$

  where TEI_A and TEO_A are the in-sample and out-of-sample tracking errors of a portfolio generated by the method A. Clearly, the smaller value of $Cons(A)$ means that the portfolio by A has more consistency between in-sample and out-of-sample tracking errors and thus it is more robust (or less sensitive) with respect to the sample data used for model (1.1).

- Superiority of out-of-sample: We define

  $$SupO(A, B) = \frac{TEO_B - TEO_A}{TEO_B} \times 100\%,$$

where TEO_A and TEO_B are out-of-sample tracking error of the portfolio by methods A and B, respectively. One can see that if $SupO(A, B) > 0$, TEO_A is smaller than TEO_B, i.e., the portfolio by method A is superior to that by method B in terms of out-of-sample tracking error; and it is very likely that the portfolio by A has a smaller expected tracking error and thus it is a better estimation to the underlying statistical regression model.

3.1 Results on small data sets

In this subsection, we compare the performance of the NPG method with the hybrid evolutionary algorithm [28] and the hybrid half thresholding algorithm [30] on five small data sets, which are Hang Seng, DAX 100, FTSE, Standard and Poor's 100, and Nikkei 225. For convenience of presentation, we abbreviate these three approaches as l_0, MIP and $l_{1/2}$ since they are the methods for l_0, MIP and $l_{1/2}$ models, respectively. In order to compare fairly the performance of these methods, we tailor their model parameters so that the resulting portfolios have same density (i.e., same number of nonzero entries).

Numerical results are presented in Tables 1 and 2, where N denotes the number of assets in a data set. In particular, we report in Table 1 in-sample error and out-of sample error of the portfolios generated by the aforementioned three methods. In Table 2, we report the consistency between in-sample and out-of-sample errors, and the superiority of out-of-sample errors for the portfolios generated by these methods. The number of nonzero portfolios given by these methods is listed in the column named "density". From Table 2, we can make the following observations.

(i) The l_0-based method (i.e., NPG method) generally has higher consistency between in-sample error and out-of-sample error than the MIP- and $l_{1/2}$-based methods (namely, hybrid evolutionary and half thresholding algorithms) since $Cons(l_0) < Cons(MIP)$ holds for 100% (30/30) instances and $Cons(l_0) < Cons(l_{1/2})$ holds for 77.3% (22/30) instances.

(ii) The l_0-based method is generally superior to the MIP- and $l_{1/2}$-based methods in terms of out-of-sample error since $SupO(l_0, MIP) > 0$ holds for 90% (27/30) instances and $SupO(l_0, l_{1/2}) > 0$ holds for 93.3% (28/30) instances.

3.2 Results on large data sets

In this subsection, we compare the performance of the l_0-based method (i.e., NPG method) with the MIP- and $l_{1/2}$-based methods (namely, hybrid evolutionary and half thresholding algorithms) on four large data sets, which are Standard and Poor's 100, Russell 2000, Russell 3000 and the Chinese index CSI 300. For a fair comparison of the performance of these methods, we tailor their model parameters so that the resulting portfolios have same density (i.e., same number of nonzero entries).

Table 1: The in-sample and out-of-sample tracking errors on small data sets.

Index	Density	l_0			MIP			$l_{1/2}$		
		TEI	TEO	S_{true}	TEI	TEO	S_{true}	TEI	TEO	S_{true}
Hang	5	6.23e-5	5.17e-5	5	5.69e-5	8.87e-5	5	8.36e-5	7.07e-5	5
Seng	6	4.29e-5	3.45e-5	6	4.85e-5	7.82e-5	6	8.58e-5	7.19e-5	6
(N=31)	7	2.37e-5	3.83e-5	7	3.26e-5	5.38e-5	7	6.45e-5	4.59e-5	7
	8	2.38e-5	2.50e-5	8	2.06e-5	3.09e-5	8	3.20e-5	2.95e-5	8
	9	2.00e-5	2.16e-5	9	1.95e-5	2.80e-5	9	3.96e-5	2.44e-5	9
	10	1.58e-5	1.55e-5	10	1.86e-5	2.77e-5	10	2.33e-5	2.34e-5	10
DAX	5	4.10e-5	1.08e-4	5	2.21e-5	1.02e-4	5	4.88e-5	1.18e-4	5
(N=85)	6	3.07e-5	1.00e-4	6	1.82e-5	9.43e-5	6	3.86e-5	1.13e-4	6
	7	2.56e-5	9.68e-5	7	1.47e-5	1.02e-4	7	2.47e-5	1.04e-4	7
	8	1.68e-5	8.71e-5	8	1.48e-5	8.78e-5	8	2.66e-5	9.36e-5	8
	9	1.54e-5	8.23e-5	9	1.05e-5	8.63e-5	9	3.44e-5	9.72e-5	9
	10	1.88e-5	8.11e-5	10	8.21e-6	7.76e-5	10	2.23e-5	1.03e-4	10
FTSE	5	1.05e-4	8.43e-5	5	6.92e-5	9.87e-5	5	1.22e-4	8.80e-5	5
(N=89)	6	7.29e-5	8.74e-5	6	5.50e-5	9.14e-5	6	1.04e-4	8.78e-5	6
	7	6.83e-5	8.18e-5	7	4.15e-5	1.02e-4	7	6.70e-5	9.67e-5	7
	8	5.81e-5	6.00e-5	8	3.50e-5	7.44e-5	8	6.11e-5	7.10e-5	8
	9	6.51e-5	5.67e-5	9	2.49e-5	8.59e-5	9	7.08e-5	5.72e-5	9
	10	6.70e-5	6.94e-5	10	2.18e-5	8.01e-5	10	5.43e-5	7.27e-5	10
S&P	5	8.74e-5	8.94e-5	5	4.50e-5	1.14e-4	5	1.02e-4	1.14e-4	5
(N=98)	6	5.87e-5	8.47e-5	6	3.37e-5	1.01e-4	6	7.93e-5	8.88e-5	6
	7	3.51e-5	7.69e-5	7	3.36e-5	8.93e-5	7	6.70e-5	7.58e-5	7
	8	5.50e-5	5.75e-5	8	2.51e-5	7.35e-5	8	6.41e-5	6.58e-5	8
	9	3.71e-5	5.09e-5	9	2.11e-5	5.92e-5	9	5.78e-5	6.56e-5	9
	10	2.93e-5	4.57e-5	10	1.85e-5	5.10e-5	10	5.22e-5	5.07e-5	10
Nikkei	5	1.34e-4	1.32e-4	5	6.02e-5	1.44e-4	5	1.22e-4	1.43e-4	5
(N=225)	6	9.48e-5	9.92e-5	6	5.13e-5	1.20e-4	6	8.26e-5	9.71e-5	6
	7	7.72e-5	9.77e-5	7	3.93e-5	1.11e-4	7	6.89e-5	1.11e-4	7
	8	9.24e-5	8.70e-5	8	3.12e-5	1.18e-4	8	7.09e-5	9.09e-5	8
	9	4.87e-5	7.68e-5	9	2.78e-5	1.18e-4	9	4.52e-5	8.22e-5	9
	10	6.39e-5	6.75e-5	10	2.36e-5	8.25e-5	10	5.37e-5	6.77e-5	10

Numerical results are reported in Tables 3 and 4, where N denotes the number of assets in a data set. In particular, we present in Table 3 in-sample error and out-of sample error of the portfolios generated by the above three methods. In Table 4, we present the CPU time of these methods and superiority of out-of-sample errors of the portfolios given by these methods. The number of nonzero portfolios given by these methods is listed in the column named "density". We can have the following observations from Table 4.

(i) The l_0-based method (i.e., NPG method) generally has higher consistency between in-sample error and out-of-sample error than the MIP- and $l_{1/2}$-based methods (namely, hybrid evolutionary and half thresholding algorithms) since $Cons(l_0) < Cons(MIP)$ holds for 100% (28/28) instances and $Cons(l_0) < Cons(l_{1/2})$ holds for 89.3% (25/28) instances.

(ii) The l_0-based method is generally superior to the MIP- and $l_{1/2}$-based methods in terms of out-of-sample error since $SupO(l_0, MIP) > 0$ holds for all instances

Table 2: The comparison on small data sets.

Index	Density	$Cons(l_0)$	$Cons(MIP)$	$Cons(l_{1/2})$	$SupO(l_0, MIP)$	$SupO(l_0, l_{1/2})$
Hang	5	1.05e-5	3.18e-5	1.29e-5	41.7	26.8
Seng	6	8.37e-6	2.97e-5	1.39e-5	55.9	52.1
(N=31)	7	1.46e-5	2.13e-5	1.86e-5	28.8	16.4
	8	1.23e-6	1.03e-5	2.43e-6	19.0	15.3
	9	1.66e-6	8.50e-6	1.52e-5	22.9	11.4
	10	**3.54e-7**	9.15e-6	**8.50e-8**	44.3	33.9
DAX	5	6.72e-5	7.97e-5	6.94e-5	**-6.28**	8.47
(N=85)	6	6.95e-5	7.61e-5	7.49e-5	**-6.27**	11.7
	7	7.12e-5	8.69e-5	7.96e-5	4.72	7.26
	8	**7.03e-5**	7.30e-5	**6.70e-5**	0.79	6.96
	9	**6.69e-5**	7.58e-5	**6.28e-5**	4.68	15.3
	10	6.23e-5	6.94e-5	8.11e-5	**-4.52**	21.6
FTSE	5	2.11e-5	2.95e-5	3.40e-5	14.6	4.27
(N=89)	6	1.45e-5	3.64e-5	1.66e-5	4.41	0.42
	7	1.35e-5	6.05e-5	2.98e-5	19.8	15.4
	8	1.85e-6	3.94e-5	9.95e-6	19.3	15.5
	9	8.39e-6	6.11e-5	1.36e-5	34.0	0.74
	10	2.46e-6	5.83e-5	1.85e-5	13.3	4.52
S&P	5	2.10e-6	6.93e-5	1.17e-5	21.7	21.3
(N=98)	6	**2.60e-5**	6.70e-5	**9.48e-6**	15.9	4.66
	7	**4.18e-5**	5.57e-5	**8.80e-6**	13.9	**-1.40**
	8	**2.58e-6**	4.83e-5	**1.70e-6**	21.7	12.6
	9	**1.38e-5**	3.81e-5	**7.81e-6**	14.0	22.4
	10	**1.64e-5**	3.25e-5	**1.49e-6**	10.4	9.96
Nikkei	5	2.10e-6	8.39e-5	2.14e-5	8.28	7.81
(N=225)	6	4.38e-6	6.83e-5	1.46e-5	17.0	**-2.11**
	7	2.05e-5	7.16e-5	4.19e-5	11.9	11.8
	8	5.40e-6	8.64e-5	2.00e-5	26.1	4.29
	9	2.81e-5	8.98e-5	3.70e-5	34.8	6.60
	10	3.60e-6	5.89e-5	1.39e-5	18.1	0.23

and $SupO(l_0, l_{1/2}) > 0$ holds for 92.9% (26/28) instances.

(iii) The l_0-based method also generally outperforms the MIP- and $l_{1/2}$-based methods in terms of speed.

4 Concluding remarks

In this paper we proposed an index tracking model with budget, no-short selling and a cardinality constraint. Also, we developed an efficient nonmonotone projected gradient (NPG) method for solving this model. At each iteration, this method usually solves several projected gradient subproblems. We showed that each subproblem has a closed-form solution, which can be computed in linear time. Under some suitable assumptions, we showed that any accumulation point of the sequence generated by the NPG method is a local minimizer of the cardinality-constrained index tracking problem. We also conducted empirical tests on the data sets from OR-library [5]

Table 3: The in-sample and out-of-sample tracking errors on large data sets.

Index	Density	l_0			MIP			$l_{1/2}$		
		TEI	TEO	S_{true}	TEI	TEO	S_{true}	TEI	TEO	S_{true}
	5	3.34e-5	2.19e-5	5	1.21e-5	2.43e-5	5	2.39e-5	1.99e-5	5
CSI 300	6	2.34e-5	2.11e-5	6	1.17e-5	2.37e-5	6	1.91e-5	2.11e-5	6
(N=300)	7	1.86e-5	1.98e-5	7	7.84e-6	2.36e-5	7	1.51e-5	2.09e-5	7
	8	1.67e-5	1.68e-5	8	7.68e-6	2.04e-5	8	1.42e-5	1.92e-5	8
	9	1.67e-5	1.54e-5	9	7.23e-6	1.85e-5	9	1.26e-5	1.63e-5	9
	10	1.13e-5	1.21e-5	10	6.42e-6	1.51e-5	10	1.32e-5	1.33e-5	10
	20	6.29e-6	7.29e-6	20	2.92e-6	7.65e-6	20	6.40e-6	7.64e-6	20
	30	3.72e-6	5.14e-6	30	2.07e-6	5.20e-6	30	4.15e-6	5.55e-6	30
	40	2.39e-6	4.17e-6	40	1.58e-6	7.63e-6	40	3.05e-6	5.30e-5	40
	50	2.87e-6	3.28e-6	50	1.90e-6	5.00e-6	50	2.03e-6	4.53e-6	50
	80	2.85e-6	7.82e-5	80	2.65e-6	9.98e-5	80	1.37e-5	9.85e-5	80
S&P	90	2.43e-6	7.52e-5	90	3.01e-6	1.24e-4	90	1.08e-5	9.98e-5	90
(N=457)	100	2.13e-6	7.39e-5	100	2.50e-6	9.69e-5	100	9.08e-6	1.04e-4	100
	120	1.66e-6	7.59e-5	120	2.58e-5	1.04e-4	120	6.42e-6	9.35e-5	120
	150	1.52e-6	7.95e-5	150	5.64e-6	1.25e-4	150	5.18e-6	1.07e-4	150
	200	1.57e-6	7.94e-5	200	2.13e-6	9.80e-5	200	2.72e-6	9.09e-5	200
	80	4.02e-6	2.07e-4	80	3.62e-6	2.89e-4	80	2.92e-5	2.34e-4	80
Russell 2000	90	3.51e-6	2.08e-4	90	4.95e-6	2.76e-4	90	2.76e-5	2.45e-4	90
(N=1318)	100	3.18e-6	1.70e-4	100	2.61e-6	2.60e-4	100	2.09e-5	2.13e-4	100
	120	2.32e-6	1.68e-4	120	2.80e-6	2.49e-4	120	1.71e-5	2.61e-4	120
	150	1.99e-6	1.94e-4	150	1.16e-5	2.68e-4	150	1.20e-5	2.66e-4	150
	200	9.83e-7	2.28e-4	200	1.42e-6	3.31e-4	200	6.89e-6	3.18e-4	200
	80	6.24e-6	1.34e-4	80	3.90e-6	1.70e-4	80	2.62e-5	1.64e-4	80
Russell 3000	90	5.49e-6	1.14e-4	90	3.33e-6	1.21e-4	90	1.99e-5	1.47e-4	90
(N=2151)	100	4.10e-6	1.05e-4	100	3.48e-6	1.05e-4	100	1.87e-5	1.37e-4	100
	120	2.78e-6	9.82e-5	120	3.01e-6	1.06e-4	120	1.66e-5	1.26e-4	120
	150	1.63e-6	1.00e-4	150	2.48e-6	1.10e-4	150	1.46e-5	1.23e-4	150
	200	1.41e-6	1.06e-4	200	3.22e-6	1.09e-4	200	1.03e-5	1.57e-4	200

and the CSI 300 index from China Shanghai-Shenzhen stock market to compare our method with the hybrid evolutionary algorithm [28] and the hybrid half thresholding algorithm [30] for index tracking. The computational results demonstrate that our approach generally produces sparse portfolios with smaller out-of-sample tracking error and higher consistency between in-sample and out-of-sample tracking errors. Moreover, our method outperforms the other two approaches in terms of speed.

We shall mention that the proposed NPG method in this paper can be used to solve the subproblems arising in the penalty method or augmented Lagrangian method when applied to solve more general problem

$$\begin{array}{ll} \min\limits_{x \in \Delta_r^u} & f(x) \\ \text{s.t.} & g(x) \leq 0, \ h(x) = 0 \end{array}$$

for some $g : \Re^n \to \Re^p$ and $h : \Re^n \to \Re^q$, where Δ_r^u is given in (1.2).

Table 4: The comparison on large data sets.

Index	Density	Time l_0	Time MIP	Time $l_{1/2}$	$Cons(l_0)$	$Cons(MIP)$	$Cons(l_{1/2})$	$SupO$ (l_0, MIP)	$SupO$ $(l_0, l_{1/2})$
	5	0.0114	26.7	1.96	4.05e-6	1.22e-5	5.65e-6	18.2	**-5.24**
CSI 300	6	0.0113	36.0	2.17	**2.32e-6**	1.20e-5	**1.96e-6**	10.9	**-0.08**
(N=300)	7	0.0039	42.1	2.31	1.22e-6	1.58e-5	5.83e-6	16.3	5.41
	8	0.0097	13.5	2.18	1.91e-7	1.27e-5	4.94e-6	17.3	12.1
	9	0.0078	17.1	2.50	1.35e-6	1.12e-5	3.66e-6	16.7	5.44
	10	0.0053	14.6	2.71	**7.95e-7**	8.67e-6	**1.28e-7**	19.7	8.72
	20	0.0078	2.84	4.30	1.00e-6	4.73e-6	1.23e-6	4.65	4.49
	30	0.0060	1.97	6.47	**1.42e-6**	3.13e-6	**1.40e-6**	1.21	7.43
	40	0.0064	2.20	6.85	1.78e-7	6.05e-6	2.25e-6	45.4	21.4
	50	0.0083	1.76	7.65	4.10e-7	3.11e-6	2.50e-6	34.5	27.8
	80	0.0271	63.6	8.64	7.53e-5	9.72e-5	8.48e-5	21.7	20.7
S&P	90	0.0207	49.0	10.2	7.28e-5	1.21e-4	8.90e-5	39.1	24.6
(N=457)	100	0.0199	77.0	15.3	7.17e-5	9.44e-5	9.47e-5	23.7	28.8
	120	0.0187	86.9	13.3	7.42e-5	1.02e-4	8.71e-5	27.3	18.8
	150	0.0184	58.7	13.5	7.80e-5	1.20e-4	1.01e-4	36.6	25.4
	200	0.0197	689.3	13.7	7.78e-5	9.58e-5	8.82e-5	19.0	12.7
	80	0.153	577.7	35.7	2.03e-4	2.85e-4	2.05e-4	28.3	11.6
Russell 2000	90	0.137	352.6	27.5	2.04e-4	2.71e-4	2.17e-4	24.7	15.0
(N=1318)	100	0.148	657.8	38.4	1.67e-4	2.58e-4	1.92e-4	34.6	20.1
	120	0.149	449.1	47.2	1.65e-4	2.46e-4	2.44e-4	32.6	35.6
	150	0.113	50.6	56.5	1.92e-4	2.56e-4	2.54e-4	27.6	27.3
	200	0.095	1352.7	46.4	2.27e-4	3.29e-4	3.11e-4	30.9	28.2
	80	0.626	861.1	37.1	1.28e-4	1.66e-4	1.38e-4	21.0	18.6
Russell 3000	90	0.267	1039.5	47.9	1.08e-4	1.18e-4	1.27e-4	6.00	22.3
(N=2151)	100	0.269	913.1	48.5	1.01e-4	1.02e-4	1.19e-4	0.05	23.5
	120	0.248	658.7	88.0	9.54e-5	1.03e-4	1.09e-4	7.26	21.8
	150	0.216	878.7	74.9	9.83e-5	1.08e-4	1.09e-4	9.34	18.9
	200	0.342	1999.9	97.9	1.05e-4	1.05e-4	1.47e-4	2.30	32.4

Acknowledgment

The authors would like to thank the two anonymous referees for their constructive comments which substantially improved the presentation of the paper.

References

[1] C. Alexander. Optimal hedging using cointegration. *Philosophical Transactions of the Royal Society of London, Series A. Mathematical, Physical and Engineering Sciences*, 357(1758): 2039–2058, 1999.

[2] M. Ammann and H. Zimmermann. Tracking error and tactical asset allocation. *Financial Analysts Journal*, 57(2): 32–43, 2001.

[3] A. Auslender, P. J. S. Silva and M. Teboulle. Nonmonotone projected gradient methods based on barrier and Euclidean distances. *Computational Optimization and Applications*, 8(3): 305–327, 2007.

[4] J. Barzilai and J. Borwein. Two-point step size gradient methods. *IAM Journal of Numerical Analysis*, 8(1): 141–148, 1988.

[5] J. E. Beasley. OR-Library: Distributing test problems by electronic mail. *Journal of the Operational Research Society,* 41(11): 1069–1072, 1990.

[6] J. E. Beasley, N. Meade and T. J. Chang. An evolutionary heuristic for the index tracking problem. *European Journal of Operational Research*, 148(3): 621–643, 2003.

[7] E. G. Birgin, J. M. Martinez and J. A. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal of Optimization*, 10(4): 1196–1211, 2000.

[8] J. Brodie, I. Daubechies, C. DeMol, D. Giannone and D. Loris. Sparse and stable markowitz portfolios. *Proceedings of the National Academy of Sciences*, 106(30): 12267–12272, 2009.

[9] I. R. C. Buckley and R. Korn. Optimal index tracking under transaction costs and impulse control. *International Journal of Theoretical and Applied Finance*, 1(03): 315–330, 1998.

[10] N. A. Canakgoz and J. E. Beasley. Mixed-integer programming approaches for index tracking and enhanced indexation. *European Journal of Operational Research,* 196(1): 384–399, 2008.

[11] C. Chen, X. Li, C. Tolman, S. Wang and Y. Ye. Sparse portfolio selection via quasi-norm regularization. arXiv:1312.6350, 2013.

[12] X. Chen, F. Xu and Y. Ye. Lower bound theory of nonzero entries in solutions of l2-lq minimization. *SIAM Journal on Scientific Computing*, 32(5): 2832–2852, 2010.

[13] Y. H. Dai and R. Fletcher. Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming. *Numerische Mathematik*, 100(1): 21–47, 2005.

[14] V. DeMiguel, L. Garlappi, J. Nogales and R. Uppal. A generalized approach to portfolio optimization: Improving performance by constraining portfolio norms. *Management Science*, 55(5): 798–812, 2009.

[15] J. Fan, J. Zhang and K. Yu. Asset allocation and risk assessment with gross exposure constraints for vast portfolios. *Available at SSRN 1307423*, 2008.

[16] B. Fastrich, S. Paterlini and P. Winker. Cardinality versus q-norm constraints for index tracking. *Quantitative Finance*, (ahead-of-print): 1–14, 2012.

[17] J. B. Francisco, F. S. Viloche Bazán. Nonmonotone algorithm for minimization on closed sets with application to minimization on Stiefel manifolds. *Journal of Computational and Applied Mathematics*, 236 (10): 2717–2727, 2012.

[18] J. Gao, D. Li. Optimal cardinality constrained portfolio selection. *Operations Research*, 61(3): 745-761, 2013.

[19] M. Gilli and E. Kellezi. Threshold accepting for index tracking. *Financial Engineering, E-Commerce, and Supply Chain* (Eds. P. Pardalos and V.K. Tsitsiringos), Kluwer Applied Optimization Series, 1–18, 2002.

[20] L. Grippo, F. Lampariello and S. Lucidi. A nonmonotone line search technique for Newton's method. *SIAM J on Numerical Analysis*, 23, 707-716, 1986.

[21] M. Lobo, M. Fazel and S. Boyd. Portfolio optimization with linear and fixed transaction costs. *Annals of Operations Research*, 152(1): 341–365, 2007.

[22] Z. Lu and Y. Zhang. Sparse approximation via penalty decomposition methods. September 2013, *SIAM Journal on Optimization*, 23(4):2448–2478, 2013.

[23] Z. Lu. Iterative hard thresholding methods for l_0 regularized convex cone programming. *Mathematical Programming*, 147(1-2): 277-307, 2014.

[24] R. A. Roll. Mean-variance analysis of tracking error. *The Journal of Portfolio Management*, 18(4): 13–22, 1992.

[25] M. Rudolf, H. J. Wolter and H. Zimmermann. A linear model for tracking error minimization. *Journal of Banking and Finance*, 23(1): 85-103, 1999.

[26] Y. Tabata and E. Takeda. Bicriteria optimization problem of designing and index fund. *Journal of the Operational Research Society*, 46(8): 1023–1032, 1995.

[27] R. Tavakoli and H. Zhang. A nonmonotone spectral projected gradient method for large-scale topology optimization problems. *Numerical Algebra, Control and Optimization*, 2(2): 395–412, 2012.

[28] P. R. Torrubiano and S. Alberto. A hybrid optimization approach to index tracking. *Annals of Operation Research*, 166(1): 57–71, 2009.

[29] S. J. Wright, R. Nowak, and M. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7): 2479–2493, 2009.

[30] F. Xu, Z. Xu and H Xue. Sparse index tracking: An $L_{1/2}$ regularization based model and solution. Submitted, 2012.