

Threshold Accepting for Index Tracking¹

Manfred Gilli

Department of Econometrics, University of Geneva, 1211 Geneva 4, Switzerland
`Manfred.Gilli@metri.unige.ch`

Evis Këllezi

Department of Econometrics and FAME, University of Geneva, 1211 Geneva 4,
Switzerland
`Evis.Kellezi@metri.unige.ch`

Abstract

In this paper we investigate the performance of the threshold accepting heuristic for the index tracking problem. The index tracking problem consists in minimizing the tracking error between a portfolio and a benchmark. The objective is to replicate the performance of a given index upon the condition that the number of stocks allowed in the portfolio is smaller than the number of stocks in the benchmark index. Transaction costs are incurred each time that the portfolio is rebalanced.

We find the composition of a portfolio that tracks the performance of the benchmark during a given period in the past and compare it with the performance of the portfolio in a subsequent period. We report computational results in the cases where the benchmarks are market indices tracked by a small number of assets. We find that the threshold accepting heuristic is an efficient optimization technique for this problem.

Keywords: Threshold Accepting, Heuristic Optimization, Index Tracking, Passive Fund Management.

JEL codes: C61, C63

¹Preliminary and incomplete version (July 20, 2001). Future updates will be available at <http://www.unige.ch/ses/metri/gilli>. We thank Peter Winker, Nick Webber and Agim Khaja for insightful comments. Financial support from the Swiss National Science Foundation (project 12-5248.97) is gratefully acknowledged.

1 Introduction

Approaches to portfolio management can be divided into two broad categories – active and passive. Active strategies rely on the belief that skillful investors can out-perform the market by exercising activities such as market timing and stock picking. In recent years passive investment strategies have become very popular, especially among mutual fund managers and pension funds. These strategies are adopted by investors who believe that financial markets are efficient and it is therefore impossible to consistently beat the aggregate market return.

A passive strategy that attempts to reproduce as closely as possible the performance of a theoretical index representing the market, is called an *index tracking* strategy. Index tracking can be done by investing in all constituents of the index proportional to their share in the index, or by selecting a smaller subset of the assets in the index such that the resulting portfolio optimally (by some criteria) tracks the performance of the chosen index. This second approach is called partial replication and is generally preferred in practice, given that full replication involves high transaction costs and difficulties in rebalancing the portfolio when the weights in the tracked index change.

Despite the increasing popularity of passive investment strategies, the attention given in the academic literature to implementation and to algorithmic problems arising in the process of index tracking is relatively small compared to the numerous articles dedicated to the classical problem of portfolio risk and return optimization.

The problem is often formulated in such a way that classical techniques like quadratic or linear programming can be directly applied. The formulation of the problem involves two main choices. First of all, one must choose an objective function that is an appropriate function of the tracking error. This is usually specified as a measure of the closeness of the solution returns to the returns of the index. Another important choice is the set of constraints imposed on the solution. A realistic formulation of the problem should include restrictions on the positions on each asset, the number of assets in the portfolio, the size of transactions costs and minimum transaction lots, as well as liquidity and exposure constraints. The only way of handling them in the context of a realistic problem size is to use heuristic algorithms that provide good approximations of the optimal solution (see e.g. work by Beasley *et al.* (1999), Mansini and Speranza (1999), Chang *et al.* (2000), Gilli and K llezi (2001), Lobo *et al.* (2000), Bertsimas *et al.* (1999), Keber and Maringer (2001)).

Different approaches have been proposed in the literature. In the majority of the work related to index tracking (Toy and Zurack, 1989; Franks, 1992; Roll, 1992; Dahl *et al.*, 1993; Clarke *et al.*, 1994; Connor and Leland, 1995; Jacobs and Levy, 1996; Jobst *et al.*, 2000; Larsen Jr. and Resnick, 1998; Rohweder, 1998; Lobo *et al.*, 2000), the tracking error is defined as the variance of the difference between tracking portfolio return and index return.

Another approach is used by Worzel *et al.* (1994), Consiglio and Zenios (2001) and Rudolf *et al.* (1998), more recently by Rockafellar and Uryasev (2001) and Konno and Wijayanayake (2001) who formulate the problem as a linear

program. The index tracking problem is formulated as a model that minimizes the absolute deviations instead of the squared deviations of the tracking portfolio return from the index as is the case for traditional optimization models.

Finally, another direction of research has been undertaken by Beasley *et al.* (1999), who propose heuristic optimization techniques like population heuristics for the solution of the index tracking problem. The advantage of these techniques is that virtually no restriction has to be imposed on the shape of the objective function or the constraints.

In this paper we propose another heuristic optimization algorithm, called threshold accepting (TA). It has been successfully applied for the portfolio risk and return optimization problem, first by Dueck and Winker (1992) and later by Gilli and K llezi (2001) and, according to results presented in this paper, appears to be an efficient technique for index tracking. One of the appealing features of the method is that it is very efficient for big problem instances where the number of stocks we choose from is high. Another strength is the possibility to easily handle a variety of objective function formulations and constraints, including the use of discrete or integer variables.

In section two of the paper the index tracking problem is defined as an optimization problem. Section three describes the implementation of the TA algorithm for the index tracking problem, section four presents computational results and section five concludes.

2 The optimization problem

The index tracking problem consists in finding an optimal method of reproducing the performance of a given index. The desired solution is a portfolio composed of a relatively small subset of the stocks in the market that behaves like the index. This section describes the problem. We start by giving the market framework and then present the optimization problem.

2.1 Framework

We suppose that there are $n_A + 1$ assets in the market from which a tracking portfolio should be found. Let p_{it} be the price at time t of the asset i , $i = 0, \dots, n_A$. We assume that asset 0 is cash (in a banking account), paying out the constant risk free rate of return r on each period of time.

Let I_t be the value of the index observed at time t . The return on the index over the period $[t - 1, t]$ is defined as

$$r_t^I = \ln \left(\frac{I_t}{I_{t-1}} \right).$$

In the sequel we make no assumption about the composition of the index. The values of I_t are effectively exogenous.

Let x_{it} be the quantity of the i th asset in the tracking portfolio at time t . We call P_t the composition of the tracking portfolio at time t ,

$$P_t = \{x_{it} \mid i = 0, 1, \dots, n_A\},$$

and introduce an index set

$$J_t = \{i \mid x_{it} \neq 0\}$$

as the set of indices of assets appearing in the portfolio P_t . The nominal value of the tracking portfolio at time t is v_t ,

$$v_t = \sum_{i=0}^{n_A} x_{it} p_{it} = \sum_{i \in J_t} x_{it} p_{it}.$$

We suppose that returns to the portfolio are measured in terms of the market value of the constituents, rather than in terms of cash realizable on the sale of the portfolio.

An alternative way to characterize the tracking portfolio is by defining the value v_t of the portfolio at time t and the weights w_{it} of each of the assets in the portfolio,

$$w_{it} = \frac{x_{it} p_{it}}{v_t}.$$

We shall shortly introduce transaction costs and slightly alter the definition of the portfolio value.

We suppose that at each time t the constituents of the tracking portfolio may be altered and that rebalancing takes place at the start of a period. Write v_{t-} for the value of the tracking portfolio at time t^- just before rebalancing,

$$v_{t-} = \sum_{i=0}^{n_A} x_{i,t-1} p_{it}.$$

In the absence of transactions costs, the nominal return on the tracking portfolio in period $[t-1, t]$ is r_t^P ,

$$r_t^P = \ln \left(\frac{v_{t-}}{v_{t-1}} \right) = \ln \left(\frac{\sum_{i=0}^{n_A} x_{i,t-1} p_{it}}{\sum_{i=0}^{n_A} x_{i,t-1} p_{i,t-1}} \right). \quad (1)$$

We suppose that the transaction costs C_t incurred in rebalancing the portfolio at time t are a function of the amounts transferred from one asset to another (including cash) or vice versa. For example, in the sequel we assume that they are proportional to absolute changes in values invested in each stock:

$$C_t = c \sum_{i=0}^{n_A} p_{i,t} |x_{it} - x_{i,t-1}|, \quad (2)$$

where c is a positive coefficient. Other functional forms for the transaction costs can be easily handled by the TA algorithm.

In the absence of transaction costs, we have $v_{t-} = v_t$. When transaction costs are present, we suppose that they may be treated in one of two ways:

1. They are deducted from the value of the tracking portfolio. Then $v_t = v_{t-} - C_t$.
2. They are taken from another account.

In the first case, period returns become

$$r_t^P = \ln \left(\frac{v_{t-}}{v_{t-1}} \right) = \ln \left(\frac{v_{t-}}{v_{(t-1)-} - C_{t-1}} \right). \quad (3)$$

The second case can be considered as similar to the first one if we assume that cash in the separate account is part of the tracking portfolio. This allows to account for the cost of keeping cash immobilized in a separate account and equation (3) remains valid.

However, considering the amount of cash in the separate account as exogenous to our problem is possible and constraints may still be imposed upon C_t .

2.2 Objective function

We have adopted the formulation given by Beasley *et al.* (1999), which will make possible the comparison of the performance of the threshold accepting algorithm with the heuristics used in their paper.

We suppose that we have observed the market prices p_{it} and I_t for periods t_1, \dots, t_2 in the past. We attempt to find the composition of a portfolio which would have tracked in an optimal way the index over the period $[t_1, t_2]$. In other words, we attempt to find quantities x_{it} , $i = 1, \dots, n_A$, and an optimization criteria F_{t_1, t_2} such that the portfolio with constant quantities $x_{it} = x_{i, t_1}$, $t = t_1, \dots, t_2$ minimizes F_{t_1, t_2} . F will be a function of the tracking error of the tracking portfolio against the index value I_t .

The measure used for the tracking error in returns over the period of time between t_1 and t_2 is E_{t_1, t_2} defined as

$$E_{t_1, t_2} = \frac{\left(\sum_{t=t_1}^{t_2} |r_t^P - r_t^I|^\alpha \right)^{\frac{1}{\alpha}}}{t_2 - t_1}, \quad (4)$$

where $\alpha > 0$.²

Let R_{t_1, t_2} be the average of the deviations of the tracking portfolio returns from the index returns over the period of time $[t_1, t_2]$,

$$R_{t_1, t_2} = \frac{\sum_{t=t_1}^{t_2} (r_t^P - r_t^I)}{t_2 - t_1}.$$

Positive deviations from the index may be desirable. One way to account for this is to define the function to be minimized F_{t_1, t_2} as a weighted difference of our measure of the tracking error, E_{t_1, t_2} , and of R_{t_1, t_2} :

$$F_{t_1, t_2} = \lambda E_{t_1, t_2} - (1 - \lambda) R_{t_1, t_2} \quad (5)$$

²This is the α -norm of the vector of deviations of the tracking portfolio returns from the index returns divided by the number of observations.

for some $\lambda \in [0, 1]$.

However, alternative ways of defining the objective function can be imagined. For example, one could require that certain quantiles or partial moments of the distribution of the returns on the index are matched by the distribution of the returns on the solution portfolio. This would be an even more general way of defining the tracking error, as it would account for asymmetries or fat-tailedness and infinite moments in the return distributions. We do not adopt this approach, although it would be easily handled by the TA algorithm. For some alternative ways of defining the objective function in the problem of benchmarking the portfolio returns see (Browne, 1999).

2.3 Constraints

We impose several additional constraints on the problem solution. These are on the size of the position on each asset, the number of assets, the size of transaction costs and roundlots. Other constraints that we do not consider here, although they can be handled by the TA algorithm, are class (or exposure) constraints, constraints on liquidity and on shortfall risk.

Size constraints

We assume that short positions are not allowed, $x_{it} \geq 0$, $i = 0, \dots, n_A$. This is a realistic assumption in many practical portfolio applications.

In order to avoid small trades, we may require that, if an asset i is included in the portfolio, its proportion in the overall portfolio value is not smaller than a minimum level ε_i , called also a buy-in threshold. Notice that buy-in thresholds provide an upper bound on the number of assets in the portfolio.

Also we often want to limit the fraction of the total amount held in each of the individual assets in the portfolio P_t to an upper bound δ_i . This guarantees a certain degree of diversity of the tracking portfolio, providing a lower bound on the number of assets in P_t .

Both constraints can be written as

$$\varepsilon_i \leq \frac{x_{it}p_{it}}{\sum_{i \in J_t} x_{it}p_{it}} \leq \delta_i \quad i \in J_t \quad (6)$$

with $0 \leq \varepsilon_i < \delta_i \leq 1$.

Cardinality constraints

As taking positions in all securities in the index may be expensive and even impossible, portfolio managers are usually constrained to invest in a much smaller number of stocks than the hundreds or even thousands of stocks that might constitute an index. In our problem formulation we constrain the number of assets in the tracking portfolio to a maximum of K ,

$$\#\{J_t\} \leq K. \quad (7)$$

An exact solution of the problem with cardinality constraints would involve the enumeration of all possible combinations of assets in each of the subsets of dimension K from the $n_A + 1$ assets in the market. Hence the problem is a difficult combinatorial one, justifying the use of heuristic methods able to provide approximations to the optimal solution.

Cardinality constraints in the mean-variance framework are discussed in detail by Chang *et al.* (2000). They compare genetic algorithms, tabu search and simulated annealing heuristics for the construction of cardinality constrained mean-variance efficient frontiers.

Constraints on transaction costs

Transaction costs can be used to model a number of costs, such as brokerage fees, bid-ask spreads, taxes, fund loads, etc. In our problem we shall consider transactions costs C_t incurred in rebalancing the portfolio at time t . We constrain them to a small proportion of the value v_{t-} of the portfolio at time t just before the rebalancing. Let γ be the proportion of the portfolio value that we allow to spend in transaction costs. The constraint imposed to the solution is

$$C_t \leq \gamma v_{t-}. \quad (8)$$

If the transaction costs are a linear function of the traded amounts of assets, the constraint on the transaction costs is convex. The problem becomes harder if fixed transaction costs are also taken into account and it can no longer be handled by convex optimization (see e.g. (Lobo *et al.*, 2000)). Realistic transaction cost functions should also account for possible discontinuities and nonconvexity (like e.g. breakpoints over which the transaction costs per share decrease), bringing the need for global optimization algorithms.

Minimum roundlots

In reality, assets are traded as multiples of minimum transaction lots, called also rounds or roundlots. Consequently, the number x_{it} of asset i invested in the portfolio, as well as the fraction w_{it} of the portfolio value invested in each security, are not perfectly fractionable and they cannot be represented by real variables.

Let the integer s_i be the minimum transaction lot for the asset i . Then the number x_{it} of asset i held in the tracking portfolio must be a multiple y_{it} of the minimum transaction lot s_i ,

$$x_{it} = y_{it} s_i. \quad (9)$$

In that case the portfolio composition P_t can be uniquely defined by the set of integers $\{y_{it} \mid i \in J_t\}$.

Mansini and Speranza (1999) solve the mixed integer linear program (MILP) of a portfolio selection problem with a linear risk function and minimum transaction lots with MILP-based heuristics.

2.4 Problem formulation

Suppose that at time t_2 one is endowed with the portfolio $P_{t_2}^- = \{x_{i,t_2}^-\}$ of value $v_{t_2}^-$. We wish to optimally rebalance this portfolio to track the index for the period $[t_2, t_3]$ by calibrating to tracking errors in period $[t_1, t_2]$. To this end suppose at time t_1^- we have the portfolio $P_{t_1}^- = \{x_{i,t_1}^-\}$ of value $v_{t_1}^-$ and weights $w_{i,t_1}^- = w_{i,t_2}^-$, $i = 0, \dots, n_A$.

Our problem consists in finding a new portfolio $P_{t_1} = \{x_{i,t_1}\}$ which, if we hold it unchanged during the period $[t_1, t_2]$, minimizes the objective function F_{t_1,t_2} under the constraints that we mentioned above:

$$\begin{aligned} \min_{P_{t_1}} \quad & F_{t_1,t_2} = \lambda E_{t_1,t_2} - (1 - \lambda) R_{t_1,t_2} \\ & C_{t_1} \leq \gamma v_{t_1}^- \\ & \sum_{i \in J_{t_1}} p_{i,t_1} x_{i,t_1} + C_{t_1} = v_{t_1}^- \\ & \varepsilon_i \leq \frac{p_{i,t_1} x_{i,t_1}}{\sum_{i \in J_{t_1}} p_{i,t_1} x_{i,t_1}} \leq \delta_i \quad i \in J_{t_1} \\ & \#\{J_{t_1}\} \leq K \end{aligned} \tag{10}$$

where C_{t_1} is the rebalancing cost at the time t_1 .

By solving problem (10), we find the number of each asset i that we should have held during all the period $[t_1, t_2]$ in the past in order to optimally reproduce the performance of the index during the same period. Assuming that the same portfolio will also optimally track the index during period $[t_2, t_3]$ in the future, at time t_2 we use the weights $w_{i,t_2} = w_{i,t_1}$ to construct a portfolio P_{t_2} in which the weight invested in asset i is the same as the corresponding weight in the portfolio P_{t_1} . Notice that each of these weights may correspond to a different number of assets x_{i,t_2} as a result of price changes during the period $[t_1, t_2]$.

The out-of-sample measures of the tracking error and the average of deviations in returns will be defined as

$$E_{t_2,t_3} = \frac{\left(\sum_{t=t_2}^{t_3} |r_t^P - r_t^I|^\alpha \right)^{\frac{1}{\alpha}}}{t_3 - t_2}, \tag{11}$$

where $\alpha > 0$ and

$$R_{t_2,t_3} = \frac{\sum_{t=t_2}^{t_3} (r_t^P - r_t^I)}{t_3 - t_2}.$$

3 Implementation of the TA algorithm for the index tracking problem

The discrete optimization problem as formulated in (10) cannot be solved with classical optimization techniques. This is a situation where heuristic optimiza-

tion techniques like simulated annealing (Kirkpatrick *et al.*, 1983) and genetic algorithms (Holland, 1975) can often be used with success.³

We suggest the use of the TA algorithm, introduced by Dueck and Scheuer (1990) as a deterministic analog to simulated annealing.⁴ It is a refined local search procedure which escapes local minima by accepting solutions which are not worse by more than a given threshold. The algorithm is deterministic as it does not depend on some probability. The number of steps where we explore the neighborhood for improving the solution is fixed. The threshold is decreased successively and reaches the value of zero after a given number of steps.

The TA algorithm has an easy parameterization, it is robust to changes in problem characteristics and works well for many problem instances. It has already been applied successfully to portfolio optimization with integer variables and down-side risk constraints.⁵ An extensive introduction to TA is given in Winker (2000).

We can formalize our optimization problem as follows. Let \mathcal{P} be the discrete set of all feasible tracking portfolios and $f : \mathcal{P} \rightarrow \mathbb{R}$ the objective function to be minimized. Define f_{opt} as the minimum of f over the set of feasible portfolios \mathcal{P} ,

$$f_{\text{opt}} = \min_{P \in \mathcal{P}} f(P). \quad (12)$$

We may have more than one optimal solution defined by the set \mathcal{P}_{min} ,

$$\mathcal{P}_{\text{min}} = \{P \in \mathcal{P} \mid f(P) = f_{\text{opt}}\}. \quad (13)$$

The TA heuristic is described in algorithm 1. After completion, it provides a solution $P \in \mathcal{P}_{\text{min}}$ or a solution close to an element in \mathcal{P}_{min} .

Algorithm 1 Pseudo-code for the threshold accepting algorithm.

- 1: Initialize threshold sequence $\tau_i, i = 1, \dots, n_S$
 - 2: Give starting portfolio $P^0 \in \mathcal{P}$
 - 3: **for** $i = 1$ to n_S **do**
 - 4: Generate $P^1 \in \mathcal{N}_{P^0}$ (neighbor of P^0)
 - 5: **if** $f(P^1) < f(P^0) + \tau_i$ **then**
 - 6: $P^0 = P^1$
 - 7: **end if**
 - 8: **end for**
-

The control parameters of the algorithm are the number of steps n_S and the sequence of thresholds $\tau_i, i = 1, \dots, n_S$. We start with the definition of the objective function, construct a mapping \mathcal{N} which defines for each $P \in \mathcal{P}$ a neighborhood $\mathcal{N}(P) \subset \mathcal{P}$ and define the sequence of thresholds. The complexity of the algorithm is $\mathcal{O}(n_S)$. Hereafter we briefly comment our implementation.⁶

³For a survey on global optimization techniques see (Pardalos and Rosen, 1987) and (Horst *et al.*, 2000).

⁴? compare the performance of simulated annealing and threshold accepting.

⁵See (Dueck and Winker, 1992) and (Gilli and K llezi, 2001).

⁶For a more detailed description of the implementation of the Threshold Accepting algorithm see (Winker, 2000) or (Gilli and K llezi, 2001).

Objective function

In (10) we defined the objective function we want to minimize. For the TA algorithm we add a penalty to the objective function

$$f(P) = F_{t_1, t_2} + \max(C_{t_1} - \gamma v_{t_1}^-, 0)$$

in order to take into account the constraint on the transaction costs. We allow the algorithm to accept solutions for which the transactions costs constraints is not satisfied.

Definition of neighborhood

To generate a portfolio P^1 in the neighborhood of P^0 we select at random an asset $a \in J_{P^0}$ and sell a fixed amount of a converted into number of assets. Next we select at random an asset b to buy. If the portfolio P^0 respects the size constraints we chose $b \in J_{P^0}$, else we chose $b \in J_{P^0} \cup J_A$, where J_A is the set of indices of all assets under consideration.

When selling and buying assets a and b we have to check whether our constraints defined in (10) remain satisfied. If not we adjust the amount of the transaction correspondingly.

Definition of thresholds

The sequence of thresholds τ_i , $i = 1, \dots, n_S$ can be defined in various ways. We used a stair function the value of which decreases to zero in a number n_R of rounds. The shape of the threshold function is given in Figure 1. The integers i_k , $k = 1, \dots, n_R - 1$, verifying $1 < i_1 < \dots < i_k < \dots < i_{n_R-1} < i_{n_R} = n_S$ define the $n_R - 1$ points where the threshold decreases. The steps from i_{k-1} to i_k for $k = 1, \dots, n_R$, are called round k .

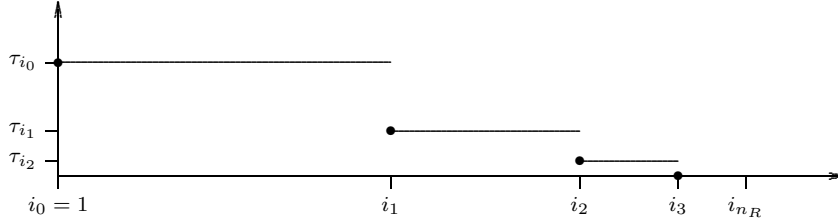


Figure 1: Threshold function.

To fix the k different values of the threshold we compute the empirical distribution of the distance of a number of randomly chosen portfolios from their respective neighbors. The distance is measured by the difference in the value of the objective function. We then take k quantiles in decreasing order for the values of the stair function.

4 Computational results

The computational results given in this section fall into two categories. First we benchmark the performance of the Threshold Accepting algorithm in a situation where we know the exact solution. Second we show some results on the out-of-sample performance of the tracking portfolios.

All computations have been executed on a PC Pentium III running at 800 MHz in a Matlab 5.x environment. The amount of memory is not an issue for the Threshold Accepting algorithm.

4.1 Benchmarking the TA algorithm

We test the TA algorithm using the data set of Beasley *et al.* (1999) and which is publicly available at <http://mscmga.ms.ic.ac.uk/jeb/orlib/indtrackinfo.html>. This data set includes prices of stocks composing the Hang Seng (31 assets), the DAX 100 (85 assets), the FTSE 100 (89 assets), the S&P 100 (98 assets) and the Nikkei (225 assets) indices, as well as the values of the indices observed in a weekly basis. We constructed an additional data set by putting all the previous stock prices together and thus forming a set of 528 assets. This allows us to test our algorithm on a larger problem.

To test the performance of our heuristic we track an artificial index constructed by selecting randomly K assets, the weights of which are also chosen at random such that they satisfy some given holding constraints. Starting from an arbitrary initial portfolio the TA algorithm tries to find the portfolio which best matches this artificial index. Setting transactions costs to zero and $\lambda = 1$ in (10) we minimize the tracking error in the objective function and therefore the optimal tracking portfolio should be identical to the portfolio used to compute the artificial index with a tracking error of zero.

Given the following constraints on the minimum and maximum holding size, $\epsilon_i = 0.01$ and $\delta_i = 1$, $i = 1, \dots, n_A$, setting $\alpha = 1$ and $K = 10$ we computed for all data sets the optimal tracking portfolio of the artificial index. We choose $n_R = 3$ and set i_1 , i_2 and i_3 to $\lceil 0.16 \sqrt{n_A} \rceil \times \lceil 8000 \ 10500 \ 12000 \rceil$ where the coefficient $\lceil 0.16 \sqrt{n_A} \rceil$ adjusts the number of steps to the problem size. The corresponding thresholds are $10^{-3} \times \lceil 0.204 \ 0.024 \ 0 \rceil$.

The results obtained with our largest data set of 528 assets are given hereafter. The composition of the portfolio used to construct the artificial index and the tracking portfolio are illustrated in Figure 2. We observe that the tracking portfolio follows the artificial index almost exactly.

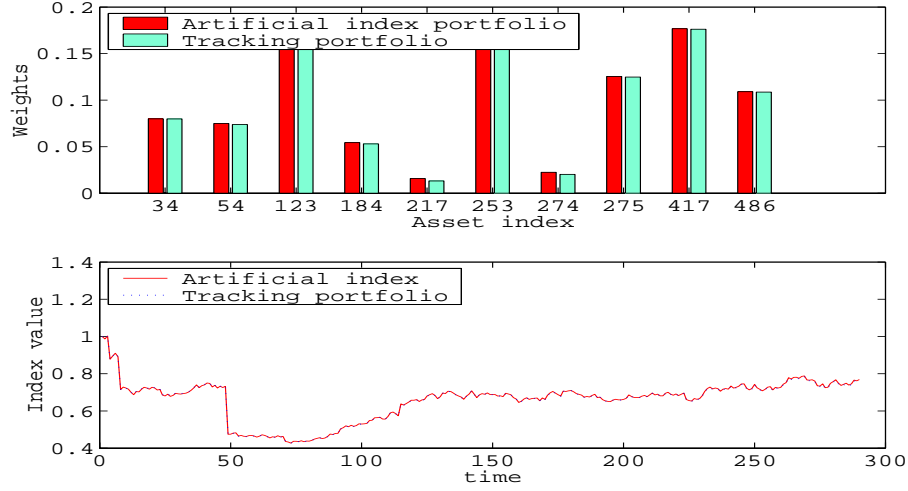


Figure 2: Tracking portfolio and artificial index for the set with 528 assets.

Figure 3 illustrates how the TA algorithm searches its way to the solution. In the upper panel the dots plot the expected return against the tracking error of each of the accepted portfolios in the TA search procedure. The lower panel displays the successive values of the objective function, which in this case is simply the tracking error. In the first 30'000 steps, the objective function does not decrease very much and we can observe that it is even allowed to increase. In the second and the third round the algorithm evolves toward the minimum.

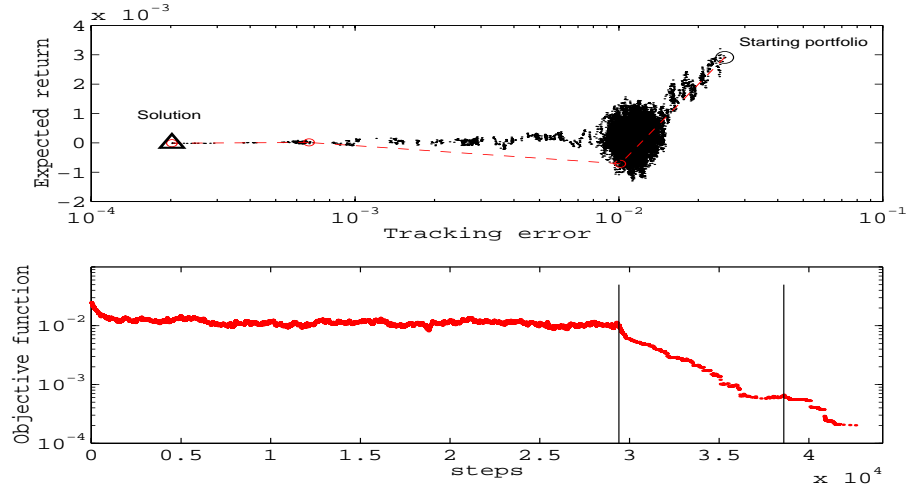


Figure 3: Working of the TA algorithm for the set with 528 assets.

In Table 1 we report the tracking errors and the execution times needed to find the tracking portfolios of the artificial indices in each of the markets. We obtain very small tracking errors.

Table 1: Tracking error and execution time for the tracking portfolios.

Index	Number of assets	Tracking error	Time (sec)
Hang Seng	31	1.80×10^{-5}	5
DAX	85	4.65×10^{-5}	6
FETSE	89	3.11×10^{-5}	7
S&P	98	4.85×10^{-5}	7
Nikkei	225	1.80×10^{-4}	13
All markets	528	2.02×10^{-4}	22

In order to get some insight about how reliable the algorithm is, we repeated for each data set 1000 times the computation of the artificial index with the corresponding tracking portfolio. Figure 4 shows the empirical distribution of the computed tracking errors. We also counted the number of times the algorithm was able to identify a tracking portfolio with the same assets as those in the portfolio which generated the artificial index. These numbers are given in the respective graphs.

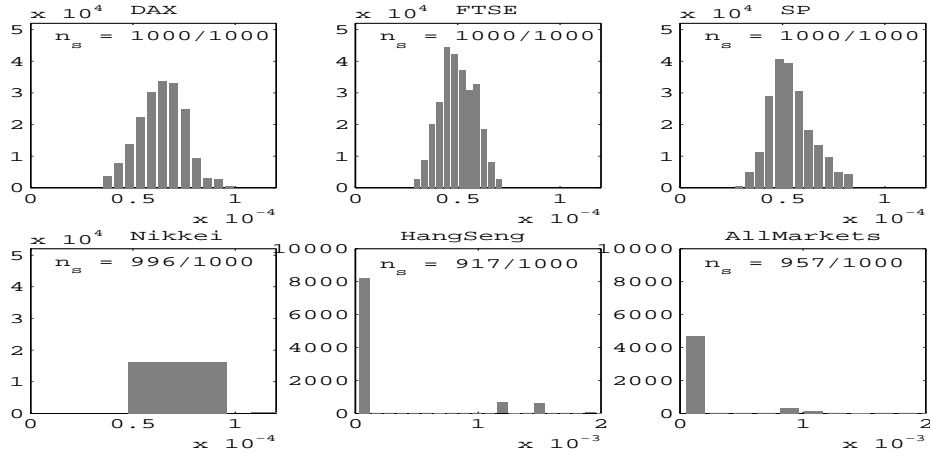


Figure 4: Empirical distribution of the tracking error for 1000 simulations.

We can achieve better results for the large data set by increasing the number of steps. This is illustrated in Figure 5 where we represent the empirical distribution of the tracking error after doubling the number of steps from 44 117 to 88 234. In this case the algorithm finds 998 times a tracking portfolio which

has the same constituents as the artificial portfolio. The mean of the tracking error is 6.45×10^{-5} with a standard deviation of 2.1×10^{-5} . Computing time increases linearly with the number of steps.

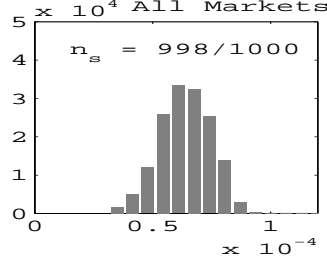


Figure 5: Empirical distribution of the tracking error for 1000 simulations for the large data set after doubling the number of steps.

4.2 Out-of-sample performance

We suppose that at time $t_2 = 245$ one is endowed with an initial portfolio with weights given in the upper panel of Figure 6. We want to optimally rebalance this portfolio in order to track the index for the period $[t_2, t_3]$, $t_3 = 289$, by calibrating to tracking errors in period $[t_1, t_2]$, $t_1 = 100$.

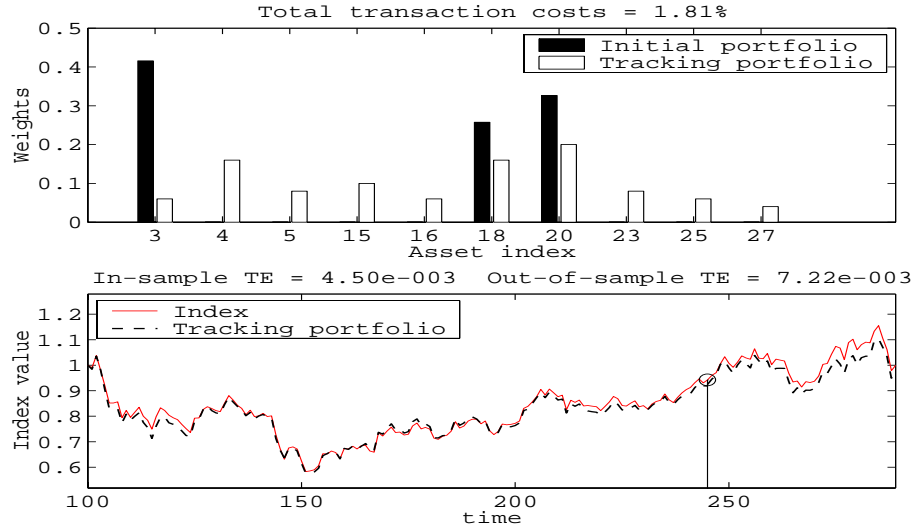


Figure 6: Out-of-sample performance of the tracking portfolio (HangSeng).

In the previous subsection we have reported in-sample tracking errors. In

reality, we are more interested on the out-of-sample performance of the tracking portfolios. In the lower panel of Figure 6 we can compare the performance in- and out-of-sample of the tracking portfolio. At time $t = 245$ the initial portfolio is rebalanced and the transaction costs that are incurred in the rebalancing reduce the value of the portfolio. This is shown in detail in Figure 7.

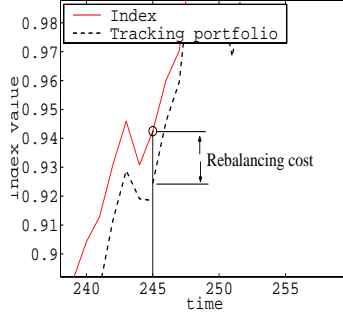


Figure 7: Rebalancing cost.

These results are obtained for $K = 10$ and no constraint on transaction costs. If we assume that transaction costs ($c = 0.01$) are incurred during the process of rebalancing, we expect that a constraint on the total amount of transaction costs would reduce the performance of the tracking portfolio by increasing the tracking error. On the other hand, if we allow a bigger number of assets in the tracking portfolio, we expect to obtain smaller tracking errors. This is confirmed by the results shown in Table 2.

Table 2: In- and out-of-sample tracking errors as a function of total transaction costs and cardinality of the tracking portfolio.

K	TCmax = 2.0%		TCmax = 0.4%	
	TEis	TEos	TEis	TEos
4	8.00×10^{-3}	1.16×10^{-2}	8.00×10^{-3}	1.25×10^{-2}
10	4.50×10^{-3}	7.20×10^{-3}	5.92×10^{-3}	9.00×10^{-3}
20	1.23×10^{-3}	2.02×10^{-3}	4.59×10^{-3}	5.95×10^{-3}

5 Conclusions

In this paper we present the index tracking problem with realistic portfolio constraints and show that this leads to a complex nonconvex optimization problem. We use the threshold accepting algorithm to efficiently solve it.

We test the algorithm in a situation where the exact solution is known and perform various experiments with different constraints. Our conclusion is that the TA algorithm is an efficient method of solving the index tracking and benchmarking problems. A variety of objective functions and constraints can be handled by the algorithm.

In order to assess the practical usefulness of our approach, more extensive simulations with systematic revisions of the tracking portfolio and other alternative objective functions are needed.

References

- Beasley, J. E., N. Meade and T.-J. Chang (1999). Index tracking. Mimeo, <http://mscmga.ms.ic.ac.uk/jeb/track.html>.
- Bertsimas, D., C. Darnell and R. Soucy (1999). Portfolio construction through mixed-integer programming at Grantham, Mayo, Van Otterloo and Company. *Interfaces* **29**(1), 49–66.
- Browne, S. (1999). Beating a moving target: Optimal portfolio strategies for outperforming a stochastic benchmark. *Finance and Stochastics* **3**(3), 275–294.
- Chang, T.-J., N. Meade, J. E. Beasley and Y. M. Sharaiha (2000). Heuristics for cardinality constrained portfolio optimization. *Computers & Operations Research* **27**, 1271–1302.
- Clarke, R. C., S. Krase and M. Statman (1994). Tracking errors, regret, and tactical asset allocation. *The Journal of Portfolio Management* **20**, 16–24.
- Connor, G. and H. Leland (1995). Cash management for index tracking. *Financial Analysts Journal* **51**(6), 75–80.
- Consiglio, A. and S. A. Zenios (2001). Integrated simulation and optimization models for tracking international fixed-income indices. *Mathematical Programming Ser. B* **89**, 311–339.
- Dahl, H., A. Meeraus and S. A. Zenios (1993). Some financial optimization models: I risk management.
- Dueck, G. and P. Winker (1992). New concepts and algorithms for portfolio choice. *Applied Stochastic Models and Data Analysis* **8**, 159–178.
- Dueck, G. and T. Scheuer (1990). Threshold accepting: A general purpose algorithm appearing superior to simulated annealing. *Journal of Computational Physics* **20**, 161–175.
- Franks, E. C. (1992). Targeting excess-of-benchmark returns. *The Journal of Portfolio Management* **18**(4), 6–12.

- Gilli, M. and E. K llezi (2001). A global optimization heuristic for portfolio choice with VaR and expected shortfall. In: *Computational Methods in Decision-making, Economics and Finance* (P. Pardalos and D.W. Hearn, Ed.). Applied Optimization Series. Kluwer. (Forthcoming).
- Holland, J. H. (1975). *Adaptation in natural and artificial systems*. University of Michigan Press.
- Horst, R., P. M. Pardalos and N. V. Thoai (2000). *Introduction to Global Optimization*. Vol. 48 of *Nonconvex optimization and its applications*. Kluwer Academic Publishers.
- Jacobs and Levy (1996). Residual risk: How much is too much. *The Journal of Portfolio Management* **22**(3), 10–15.
- Jobst, N. J., M. D. Horniman, C. A. Lucas and G. Mitra (2000). Computational aspects of alternative portfolio selection models in the presence of discrete asset choice constraints. Mimeo.
- Keber, C. and D. Maringer (2001). On Genes, Insects, and Crystals: Determining Marginal Diversification Effects With Nature Based Algorithms. Mimeo, University of Vienna.
- Kirkpatrick, S., C. Gelatt and M. Vecchi (1983). Optimization by simulated annealing. *Science* (220), 671–680.
- Konno, H. and A. Wijayanayake (2001). Minimal cost index tracking under non-linear transaction costs and minimal transaction unit constraints. *Journal of Theoretical and Applied Finance*. Forthcoming.
- Larsen Jr., G. A. and B. G. Resnick (1998). Empirical insights on indexing. *The Journal of Portfolio Management* **25**(1), 51–60.
- Lobo, M. S., M. Fazel and S. Boyd (2000). Portfolio optimization with linear and fixed transaction costs and bounds on risk. Mimeo. Information Systems Laboratory, Stanford University.
- Mansini, R. and M.-G. Speranza (1999). Heuristic algorithms for the portfolio selection problem with minimum transaction lots. *European Journal of Operational Research* **114**, 219–233.
- Pardalos, P. M. and J. B. Rosen (1987). *Constrained Global Optimization: Algorithms and Applications*. Vol. 268. Springer-Verlag, Berlin.
- Rockafellar, R. T. and S. Uryasev (2001). Conditional value-at-risk for general loss distributions. Mimeo, University of Florida.
- Rohweder, H. C. (1998). Implementing stock selection ideas: does tracking error optimization do any good ?. *The Journal of Portfolio Management* **24**(3), 49–59.

- Roll, R. (1992). A mean/variance analysis of tracking error. *The Journal of Portfolio Management* **18**(4), 13–22.
- Rudolf, M., H.-J. Wolter and H. Zimmerman (1998). A linear model for tracking error minimization. *Journal of Banking and Finance* (23), 85–103.
- Toy, W. M. and M. A. Zurack (1989). Tracking the Euro-Pac index. *The Journal of Portfolio management* **15**(2), 55–58.
- Winker, P. (2000). *Optimization Heuristics in Econometrics: Applications of Threshold Accepting*. Wiley, New York.
- Worzel, K. J., C. Vassiadou-Zeniou and S. A. Zenios (1994). Integrated simulation and optimization models for tracking indices of fixed-income securities. *Operations Research* **42**, 223–232.