Index Tracking using a Hybrid Genetic Algorithm

Roland Jeurissen and Jan van den Berg
Faculty of Economics, Erasmus University Rotterdam
P.O. Box 1738, 3000 DR Rotterdam, The Netherlands
Email: rjeurissen@gmail.com, jvandenberg@few.eur.nl

Abstract—Assuming the market is efficient, an obvious portfolio management strategy is passive where the challenge is to track a certain benchmark like a stock index such that equal returns and risks are achieved. In this paper, we investigate an approach for tracking the Dutch AEX index where an optimal tracking portfolio (consisting of a weighted subset of stock funds) is determined. The optimal weights of a portfolio are found by minimizing the tracking error for a set of historical returns and covariances. The overall optimal portfolio is found using a hybrid genetic algorithm where the fitness function of each chromosome (possible subset of stocks) equals the minimal tracking error achievable. We show the experimental setup and the simulation results, including the out-of-sample performance of the optimal tracking portfolio found.

I. INTRODUCTION

A. Portfolio management strategies

Portfolio management approaches can be divided in three categories [2] namely active management, passive management, or a mixed strategy of the two. Active management relies on the belief that skillful investors are able to outperform the aggregate market return. An active manager tries to pick stocks that will outperform other stocks and attempts other active activities such as market timing (optimal timing their buy/sell decisions). Passive management on the other hand is adopted by investors who believe that financial markets are efficient. Consequently, passive managers believe that it is impossible to consistently beat the market. Their main activity is endeavoring to achieve the same returns and risk of a certain benchmark. Attempting to reproduce the performance of an index, such as the S & P 500, is often referred as index tracking. A passively managed fund whose objective is index tracking is called an index fund, or tracker fund.

The debate about active versus passive management is still continuing. Sharpe [3] asserts that on average active managers can not beat passive strategies. Active investing is a zero sum game where some investors will win and some will lose relative to the return of the market or a market segment. Consequently, after costs, the return on the average actively managed dollar will be less than the return on the average passively managed dollar. Empirical researches by Malkiel [4], Sorenson [5] and Frino [6] have shown that passive strategies are found to outperform active strategies on average. So it is not a surprise that passive strategies are increasingly popular. Frino [6] estimates that in total $1.5 trillion is passively invested in index funds in the USA alone.

Matching the performance of an index can be performed in two ways. The first is full (or complete) replication, a method in which all the shares making up an index are held in their respective market weights. This method perfectly reproduces the index, but incurs high transaction costs. E.g., imagine buying all the stocks of the Wilshire 5000 composed of more than 6000 companies. Moreover, when stock weights comprising the index change (as a result of a merger, stock split etc.), high transaction costs will occur again. Unfortunately, composition change of indices happens a lot. The process of adjusting a tracking portfolio due to changes of an index is referred as rebalancing.

For the above-given reasons, many index fund managers hold only a subset of stocks from the index. This method is called partial replication and involves lower transaction and management costs. However, this strategy introduces a tracking error, the measure of the deviation of the chosen portfolio from the index [7]. Obviously, index fund managers try to minimize this tracking error. Consequently the index tracking problem consists of minimizing both tracking error and transaction costs.

B. Existing literature on index tracking

Let us start to summarize some empirical evidence results. Performance deviations of the tracking portfolio arise through several factors. The main factors identified by Frino and Galagher [6] are transaction costs, the way dividends are treated by the index, the volatility of the index, timing effects, front-running by arbitrageurs and index composition changes. They also noticed seasonality of the tracking error: the tracking error is significantly higher in January and during months when stocks go ex-dividend. Liquid indices such as the S & P 500 tend to have low transaction costs compared to non-liquid ones, but investors have to pay a substantial price premium for stocks in these markets as identified by Petajisto [8]. When a new stock is added to the index, this generally goes up. This results into extra costs for index managers because they have to buy the stock at the new price. Larsen-Jr. and Resnick [9] have shown that tracking portfolios for value weighted indices (based on market capitalization) have a less tracking error and lower standard deviations of the tracking error than they have for equal weighted indices.

Looking at traditional methods of index tracking, we observe that there exist several models for reproducing the performance of an index using only a subset of stocks. Meade and Beasley [10] and Larsen-Jr. and Resnick [9] used
a single factor model in order to minimize tracking error given a subset of shares. Meade and Salkin [11], [12] discuss some assumptions related to the tracking error in order to solve the problem by using quadratic programming. They also considered the effect of industry stratification within a tracking portfolio. However, this strategy did not boost the performance in their research. On the other hand, Larsen-Jr. and Resnick [9] found out that stratification does work, especially for high capitalization indices.

Tabata and Takeda [13] considered the traditional asset allocation of the Markowitz type [14], [15] and used an efficient approach to find a local optimal solution. They also pointed out that the problem is a bi-criteria optimization problem: firstly the zero-one integer problem deciding which stocks to include in the portfolio must be solved and afterwards an algorithm solving the weights of the chosen stocks by minimizing tracking error must be applied. However, this problem has no tractable analytic solution. To see why consider the problem of tracking the S & P 500 with 200 stocks. The number of possible portfolios equals 500!/(200! × 300!) ≈ 5, 055, 101.44. So, the solution space is very large because of its combinatorial explosion. That is why many researchers proposed heuristic approaches for this problem, either of stochastic or of deterministic nature.

Gilli and Kleezi [16] used a Threshold Accepting Algorithm which 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 constructed portfolios performed rather well for different data sets. In their objective function they also considered transaction costs. Another heuristic search approach is simulated annealing [17]. This procedure is implemented for the tracking error problem by Derichs and Nickel [18]. Besides the creation of a successful tracking portfolio, they also considered the revision - including the transaction costs involved - of an existing tracking portfolio over time. Next, Dose and Cincotti [19] formed a tracking portfolio based on complete-link hierarchical clustering. They cluster homogeneous groups based on similarity of returns and portfolio based on complete-link hierarchical clustering over time. Next, Dose and Cincotti [19] formed a tracking portfolio, they also considered the revision - including the transaction costs and the revision - including the transaction costs involved - of an existing tracking portfolio over time. Dose and Cincotti [19] formed a tracking portfolio based on complete-link hierarchical clustering. They cluster homogeneous groups based on similarity of returns and then take one stock from each cluster to form a subset. After the selection they perform weight optimization of the subset. Their results are not as good as found by the previous methods. Last but not least is the use of genetic algorithms in order to search the solution space for a subset of shares that performs well in tracking an index. When this subset is determined, optimization techniques such as quadratic programming are used to solve the optimal proportions to hold the shares. Shapcott [7] is, to our knowledge, the first who successfully employed this strategy. He tracked the FTSE 100 index (UK) effectively with a portfolio consisting of 20 stocks. He also proved the superiority of the algorithm compared to random search algorithms. Eddelbüttler [20] used a similar strategy when tracking the German Xetra DAX 30 index. The research included also a section describing the fact that the genetic algorithm was able to converge to the global minimum tracking error portfolio. Beasley et al. finally [2] extended these approaches by considering transaction costs and the revision of an existing tracking portfolio. They tested the approach on the S & P 500 (USA), the FTSE 100 (UK), Hang Seng (Hong Kong), Xetra DAX 100 (Germany) and the Nikkei 225 (Japan). The results are quite impressive. They also conducted reduction tests in order to reduce the size of the search space and hence enabling the algorithm to be more effective. Because of the completeness of their research, this is the leading paper for index tracking using genetic algorithms.

Inspired by the outstanding results of the genetic algorithm, we decided to use this approach for tracking the Dutch AEX index. The findings of this research are presented in this paper. The rest of this paper is structured as follows. In the next section II, we sketch our approach of tackling the index tracking problem, in section III, we discuss the main results obtained and in section IV we make some conclusions and present an outlook.

II. RESEARCH APPROACH

A. Formulating the problem

Let \( r_B = h'_B r \) represent the expected return of the benchmark (index) portfolio \( B \) where \( h'_B = (h_1, h_2, \ldots, h_n, \ldots) \) is the transposed vector of \( h_B \) defining the weights \( h_B \) of the stocks in this benchmark portfolio and \( r \) is the vector of expected returns. The variance \( \sigma_B^2 \) of portfolio \( B \) can be expressed as \( \sigma_B^2 = h'_B \Sigma h_B \) where \( \Sigma \) is the symmetric matrix of covariances [20].

As has been mentioned above, partial replication involves the inclusion of a subset of stocks in the passive portfolio that tracks the benchmark. We shall assume that the number of stocks in this subset is fixed and given beforehand. Let \( r_P = h'_P r \) represent the expected return of the tracking portfolio \( P \) where \( h_P \) is the vector defining the weights \( h_P \) of the stocks in the tracking portfolio. The variance of portfolio \( P \) can be expressed as \( \sigma_P^2 = h'_P \Sigma h_P \). We further assume that for both portfolios the weights of their stocks sum up to one, i.e., \( \sum_P h_P = 1 \) and \( \sum_P h_P = 1 \).

The objective function to be minimized is the tracking error of the passive portfolio \( P \) with respect to its benchmark portfolio \( B \). There exist several ways to define this tracking error [6]. In this paper, we will use the standard test used in industry called the ‘volatility of the tracking error’ defined by the expected variance of the difference between the returns of the passive portfolio and those of the benchmark. The standard deviation of this variance can be written as [20]

\[
\sigma_{r_P - r_B} = \sqrt{(h_p - h_b)' \Sigma (h_p - h_b)} = \sqrt{x' \Sigma x}. \tag{1}
\]

Note that the tracking error defined by equation (1) just focusses on minimizing the variance/standard deviation in the return differences. So the natural requirement (constraint) of selecting a passive portfolio having the same expected tracking error as the benchmark is simply discarded [20].

Having chosen the above-given approach, the problem of finding an optimal tracking portfolio consists of two parts. The first part consists of selecting the right subset of stocks,
the second part of minimizing equation (1) by changing the weights \( w_p \) of the selected stocks in the tracking portfolio. As weights \( w_b \) of the benchmark portfolio, we simply take their current weights in the index. Minimizing (1) involves a \textit{quadratic programming} problem, i.e., it concerns the minimization of a quadratic function subject to linear constraints. More precisely, we can formulate this quadratic programming problem as

\[
\text{Minimize } \sqrt{x'\Sigma x}
\]

\[
\text{Subject to } 1 \cdot x = 0
\]

\[
\text{with } -h_b \leq x \leq -h_b + 1.
\]

(2)

Because the weights \( w_b \) of the index and the weights \( w_p \) of our portfolio sum to 1, it follows that the sum of their differences \( x = w_p - w_b \) must be equal to 0, or, \( 1 \cdot x = 0 \) where 1 is a vector containing only ones. The inequality in (2) follows directly from the constraint that \( 0 \leq w_p \leq 1 \) by subtracting \( w_b \) from all the terms present in this inequality.

\[2. \text{ Finding the optimal portfolio}\]

A so-called \textit{hybrid genetic algorithm} [20] is used to solve the above-formulated problem of finding the optimal tracking portfolio. A genetic algorithm is used to search the solution space for possible subsets of stocks. The genetic algorithm (GA) is a randomized search algorithm based on mechanics of natural selection and genetics [21]. The evolution starts from a population of randomly chosen individuals represented by chromosomes. Here, in each generation, the fitness of all population members is determined using the solution of the quadratic programming problem (2). Multiple individuals are selected from the current population based on their fitness value found. Next, the selected individuals are modified using ‘crossover’ and ‘mutation’ operators to form a new population [22]. Each chromosome represents a possible solution, here, a subset of shares from the benchmark portfolio. Typically, binary strings are used as an encoding. Since the GA approach is combined with a quadratic programming approach, the GA is termed ‘hybrid’ [20].

In order to find the minimum expected tracking error as defined by (2), we need to know the benchmarks weights \( w_b \) and the covariance matrix \( \Sigma \). Since we apply the index tracking approach to the Dutch AEX-index, we do not need to calculate the benchmark weights since they are freely available. In order to calculate the covariance matrix \( \Sigma \) several approaches can be used (for a discussion on this, we refer to [1]). We decided to use an estimation based on historical data. This method does not apply any statistical assumptions but, instead, relies history using raw historical data. It requires a large data set and is easy implemented.

\[3. \text{ Experimental setup}\]

In this subsection we present a lot of the implementation details related to the simulations we performed.

1) Data set: The AEX-index is the best known index of Euronext Amsterdam and is made up of 25 high capitalized stocks. The weights of the stocks are based on market capitalization and are adjusted real time. The index provides a fair representation of the Dutch economy (more details about this index are available at http://www.euronext.com). Our goal is to reproduce the performance of this index using only a subset consisting of 10 index shares.

The composition of the AEX-index appears to change often. For example, during the time between 01/01/2001-04/05/2004, the index has changed 17 times. We used the data between 02/01/2001-02/03/2004 as \textit{training data} for finding the optimal tracking portfolio. We tested the tracking portfolio found \textit{out of sample} using the data from the period 03/03/2004-03/03/2005.

We used the composition of the index on 02/03/2004 to determine the benchmark weights (for precise information on the benchmark weights used, we again refer to [1]). The covariance matrix is calculated using daily stock quotes between 02/01/2001-02/03/2004 (calculation details are given below). All quotes are obtained by using DATASTREAM (http://www.datastream.net/) and are adjusted for stock splits, dividends etc.

2) Implementation of genetic operators: Since we need to select 10 stocks from an index having 25 stocks, each chromosome can be represented by a binary string consisting of 25 bits, 10 of which are equal to ‘1’. The fitness function evaluates the strength of each chromosome. A more fit individual has a higher probability of reproduction over a less fit one. In our research, the fitness function equals the calculated tracking error of each chromosome. The lower the better, because our objective is to minimize tracking error defined by equation (1). In order to calculate the fitness of each chromosome, we do need \( x_b \) (the vector defining the weights of the stocks for each chromosome). At this stage, the genetic algorithm is hybridized with the quadratic programming routine. The routine solves the problem for each chromosome and delivers the corresponding tracking error to the fitness function of the genetic algorithm. This approach combines the search procedure of the genetic algorithm in the solution space with the local convergence properties of the quadratic programming solver.

Selection is the stage of a genetic algorithm in which individuals are chosen from a population for later breeding. In this thesis we use deterministic tournament selection. Tournament selection runs a tournament among a few individuals and selects the winner (the one with the best fitness). Tournament selection has several benefits: it is efficient to code, works on parallel architectures, and allows the selection pressure to be easily adjusted [23]. Associated with the selection step is the ‘elitism’ strategy. Elitism is a method that guarantees that a number of best solutions are placed directly into the next generation. In the approach presented in this paper, elitism is used, because that way, the search for a good solution never goes backwards.

Crossover operators (which take two parent chromosomes and combine them to produce a child) need to be carefully
The most common crossover operator is ‘one point crossover’. This crossover operator picks a random point within the chromosomes, and then switches the genes of the two chromosomes at this point to produce two new offspring. If an offspring takes the best parts from each of its parents, the result will likely be a better solution. However, when using this operator the number of ones and zero’s in the string of the parents compared to the children can change. This is undesirable because we don’t want that the number of stocks included in the portfolio can change. As a result of this constraint, a two point ‘order based’ crossover is used in this paper. The idea behind the order based crossover is to swap the genes in the order found at the other parent. The number of ones and zeros will remain the same.

| Parent 1 | 11111000100 |
| Parent 2 | 0100101011 |
| Cut Point | 5 |
| Child 1 | 11111010111 |
| Child 2 | 01001001100 |

**TABLE I**
EXAMPLE OF A SINGLE-POINT CROSSOVER

*Mutation* is necessary to prevent areas of the search space being discarded. On the other hand, a too high mutation rate will prevent the desired convergence of the learning process. The standard mutation operator chooses a single bit at random and swaps its value. Again, we don’t allow changing the number of ones and zero’s. Therefore, we applied mutation inversion: rather than selecting a single bit to mutate, inversion mutation finds two random characters in the string and reverses them.

| Parent 1 | 10000101011 |
| Parent 2 | 0110110010 |
| Cut Point | 3 and 7 |
| Child 1 | 01100110011 |
| Child 2 | 01110010110 |

**TABLE II**
EXAMPLE OF A TWO POINT ORDER BASED CROSSOVER. THE BOLD PART IS SWAPPED IN THE ORDER FOUND AT THE OTHER PARENT. THE UNDERLINED PART REMAINS THE SAME.

3) **Parameters of the Genetic Algorithm:** One of the difficulties of genetic algorithms is finding the best internal parameters in order to optimize speed and convergence. These parameters are

- Size of the initial population
- Number of generations
- Crossover probability
- Mutation probability
- Rate of elitism

After extensive trial and error, we found adequate parameters for our problem. These parameters are presented in Table III:

| Size of the initial population | 20 |
| Number of generations | 100 |
| Crossover probability | 0.1 |
| Mutation probability | 1.0 |
| Rate of elitism | 2 |

**TABLE III**
INTERNAL PARAMETERS OF THE GENETIC ALGORITHM

Returns have been calculated ‘continuously’ conform

\[ r_n(t) = \ln\left(\frac{p_n(t)}{p_n(t-1)}\right), \]

where \( p_n(t) \) equals the price of stock \( n \) at time \( t \). The covariance matrix \( V \) has been estimated by calculating the elements of matrix \( (r_n, r_m) \) according to

\[ \text{cov}(r_n, r_m) = \frac{\sum_{t=0}^{T-1} (r_n(T-t) - \bar{r}_n)(r_m(T-t) - \bar{r}_m)}{T-1}, \]

where \( \bar{r}_n \) is the arithmetic mean of historical returns \( r_n(t) \).

5) **Validation:** As mentioned above, we applied out of sample testing in order to evaluate the performance of the optimal tracking portfolio. To do so, we calculated two tracking errors based on the data samples from the period 03/03/2004-03/03/2005. The first tracking error \( E_1 \) is directly related to the volatility of the tracking error and is based on an estimation of standard deviation (1). It is written as

\[ E_1 = \sqrt{\frac{1}{T-1} \sum_{t=1}^{T} (r_P(t) - \bar{r}_P - (r_B(t) - \bar{r}_B))^2}, \]

where \( \bar{r}_P \) and \( \bar{r}_B \) are the arithmetic mean of the returns of tracking portfolio and the benchmark portfolio respectively.

The second out-of-sample tracking error \( E_2 \) is defined as the average of absolute differences in the returns \( r_P(t) \) of the index portfolio and the returns \( r_B(t) \) of the benchmark. According to [24], \( E_2 \) can be stated as.

\[ E_2 = \frac{\sum_{t=1}^{T} \sqrt{(r_P(t) - r_B(t))^2}}{T}. \]

6) **Test environment:** The returns and covariance matrix are calculated using Microsoft Excel 2003. This program is also used to evaluate the performance of constructed portfolios and to make the graphs that are presented in the results section. MATLAB 6.5 is used to code the genetic algorithm and it performs the quadratic programming. All computations were performed on an AMD Athlon™ 64 processor, 1.80 Ghz, 1 GB Ram, personal computer running Windows XP Professional.

III. RESULTS

We did several experiments and while using the above-given parameters, the hybrid genetic algorithm found the solution presented in Table IV. Both the companies and their weights in the tracking portfolio are given. We observe that the stocks selected for our optimal tracking portfolio coincide with the larger stocks of the AEX-index and that, in many but not all
cases, the weights are relatively the same (for more details on this, we refer to [1]). In addition we note that the minimum weight in the tracking portfolio (that of AHOLD shares) is around 4.7% and the maximum weight (that of Unilever stocks) somewhat less than 14.5%.

In order to estimate the quality of this tracking portfolio found, we performed several other experiments. First of all, we measured the above-described out-of-sample performance. Figure 1 shows the out-of-sample performance compared to the performance of the true AEX-index. We observe that initially, both performances are almost equal while gradually small differences emerge (due to, among other things, changes in the weights of the shares composing the AEX-index). We further note that the up and down movements of both portfolios are quite similar.

To better assess the quality of the tracking portfolio found, we also constructed other portfolios:

- 10 randomly selected stocks, equal proportions: figure 2;
- selection of the 10 highest capitalized stocks, equal proportions: figure 3;
- selection of the 10 lowest capitalized stocks, equal proportions: figure 3;
- Full Replication: selection of all the shares making up the AEX are held in their respective market weights: (figure not shown here)

As might be expected from previous observations, we note that the tracking performance of the high capitalized portfolio is similar to but also somewhat less than the performance of the optimal portfolio found. Randomly chosen portfolios and portfolios consisting of low capitalized stocks show even lower performances.

We tried to further underpin these observations. In the first row of table V, we show the estimation of the out-of-sample tracking error as defined by equation (5). The tracking errors $E_2$ defined by (6) are shown in the second row of table V. We observe that generally the tracking errors (6) are slightly larger than those of equation (5). We also conclude that, except from the performance of the Full Replication approach (which, due to transaction costs, can only be executed by large investment companies offering trading in Exchange Traded Funds (ETFs)), the performance of the optimal ‘GA tracker’

### Table IV

<table>
<thead>
<tr>
<th>Company</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABN AMRO</td>
<td>0.126271</td>
</tr>
<tr>
<td>Aegon NV</td>
<td>0.075173</td>
</tr>
<tr>
<td>Ahold</td>
<td>0.04673</td>
</tr>
<tr>
<td>Akzo Nobel</td>
<td>0.078453</td>
</tr>
<tr>
<td>Fortis</td>
<td>0.087774</td>
</tr>
<tr>
<td>ING</td>
<td>0.084059</td>
</tr>
<tr>
<td>KPN</td>
<td>0.075942</td>
</tr>
<tr>
<td>Philips</td>
<td>0.141958</td>
</tr>
<tr>
<td>Royal Dutch</td>
<td>0.139015</td>
</tr>
<tr>
<td>Unilever</td>
<td>0.144623</td>
</tr>
<tr>
<td>SUM</td>
<td>1</td>
</tr>
</tbody>
</table>

**The portfolio constructed by the genetic algorithm**

![Fig. 1. Tracking performance of a portfolio (10 stocks) constructed by the Hybrid Genetic Algorithm.](image1.png)

![Fig. 2. Tracking performance of a typical random drawn portfolio (10 stocks).](image2.png)

![Fig. 3. Tracking performance of respective high and low capitalized portfolios (10 stocks).](image3.png)

### Table V

<table>
<thead>
<tr>
<th></th>
<th>GA Tracker</th>
<th>Random</th>
<th>High Cap.</th>
<th>Low Cap.</th>
<th>Full Replication</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.001170</td>
<td>0.005197</td>
<td>0.001611</td>
<td>0.005449</td>
<td>0.000122</td>
</tr>
<tr>
<td></td>
<td>0.001534</td>
<td>0.005229</td>
<td>0.001905</td>
<td>0.006034</td>
<td>0.000089</td>
</tr>
</tbody>
</table>

**Out of sample tracking errors of different portfolios. Row 1 shows the tracking errors defined by equation (5), row 2 those defined by equation (6).**
IV. CONCLUSIONS AND OUTLOOK

In this paper, we presented a GA approach for finding an optimal tracking portfolio of the Dutch AEX stock index. The overall optimal portfolio is found using a hybrid genetic algorithm where the fitness function of each chromosome (possible subset of stocks) equals the minimal tracking error achievable. This minimal tracking error is determined by solving the corresponding quadratic programming problem. We have shown the experimental setup of the experiments as well as the simulation results.

We conclude that the performance of the tracking portfolio found is much better than that of randomly selected portfolios and that of low capitalized portfolios which can be derived from the AEX-index. In addition, it is shown that the performance of high capitalized portfolios is similar although still slightly worse than the optimal GA tracking portfolio. This research shows that even a small stock index like the AEX-index can be tracked quite well by a relatively subset of its composing stocks.

Further research might show whether portfolios consisting of subsets with a number of stocks different from, but close to 10 yield similar results. In addition it is interesting to analyze the frequency needed for re-balancing the optimal tracking portfolio in an attempt to further improve the performance in the long run of the GA tracker.

REFERENCES