A Genetic Algorithm for Passive Management: Creating Index Funds with Fewer Stocks

Dirk Eddelbüttel
GREQAM-EHESS
Marseilles, France
edd@ehess.cnrs-mrs.fr

Third International Conference
Forecasting Financial Markets:
Advances for Exchange Rates,
Interest Rates and Asset Management
London, 27-29 March 1996

Abstract

Passive management is an asset-allocation methodology that attempts to reproduce the behaviour of a chosen financial market index. It poses an interesting optimization problem: how well can we reproduce the market index using only a subset of the stocks that comprise the index. This problem has no tractable analytic solution as it is a combination of two distinct operations research problems. First, we must solve the zero-one integer problem of deciding which stocks to include into the tracking portfolio. Second, conditional on this first problem, we must solve a quadratic minimization problem in order to find the proportions for each of the chosen stocks that minimize the tracking error.

We propose to use a genetic algorithm for this passive management portfolio construction problem. Genetic algorithms appear as a natural choice for this type of optimization problem, given their capacity for identifying possible solutions from a vast parameter space. We use standard encoding techniques and genetic algorithm operators in an empirical application to the German stock market that uses daily price data for thirty German stocks and the DAX index of the stock market in Frankfurt.

We first verify whether a genetic algorithm is capable of finding the index weights under the restriction that all stocks comprised in the market index must be included in the tracking portfolio. Next, we lift the restriction of pre-selected stocks and allow the genetic algorithm to determine both which stocks are to be included in the tracking portfolio and the proportions with which these stocks are represented. We show that the genetic algorithm selects portfolios which compare favourably with the tracking errors from portfolios that were constructed using a quadratic program. Extensions of this application are discussed.

JEL Classification: C15, C49, C61, G11.

Keywords: Index fund, passive management, portfolio selection, quadratic programming, genetic algorithm.

Acknowledgements: Financial support from the Ministère de l'Éducation, Paris, and the Apel Fellowship, Braunschweig, is gratefully acknowledged. Deutsche Bank Research provided the data. This paper has benefited from comments by Russell Davidson and Lisa Powell. Remaining errors are the sole responsibility of the author.

1 Introduction

Institutional investors can be characterized as either *active* or *passive*. The former aim to achieve superior results by *actively* searching for promising investments. Examples of active management strategies include 'stock picking', that is the attempt to identify companies that appear to be fundamentally undervalued, and 'market timing' which tries to identify the direction of the next move of the markets. Empirical evidence suggests that active management has difficulties consistently outperforming a benchmark such as a broad market index that represents a stock market (Jeffrey and Arnott 1993). More than 500 billion US dollars in assets are now passively managed by the leading index funds (Schramm 1995).

Passive investors have a more modest goal: to obtain returns that are as good as a pre-chosen benchmark, for example a stock market index. The objective of passive management therefore is to replicate the time series behaviour of a selected market index. The commonly chosen metric that is minimized in order to achieve this goal is the standard deviation of the differences in daily returns between the market index and its tracking portfolio. This problem can be transformed into a quadratic programming problem under linear constraints for which analytical solutions can be obtained.

A more interesting problem arises if the tracking portfolio is to be constructed with fewer stocks. The passive investor is now faced with two problems: first, stocks for the index tracking portfolio have to be selected and, second, weights for each of the selected stocks must be chosen. The first task is a zero-one, or integer, problem which creates a set of stocks which form the tracking portfolio. The second task is to select the weights of each stock given this set. As before, for a given set of stocks, a quadratic program finds the optimal weights that minimize the standard deviation. But because these two tasks are linked for constructing passive portfolios, no efficient solution is available. The problem actually is NP-complete, as pointed out by Tabata and Takeda (1995) who propose a heuristic method for a compromise solution. NP-completeness means that while the problem is soluble in principle, it is not tractable in practice: for increasing portfolio size N, the necessary computing time increases at a faster than polynomial rate.

This paper extends earlier work (Eddelbüttel 1992) and proposes to apply a genetic algorithm to the passive management portfolio construction problem. Genetic algorithms, a global search method, appear as a natural choice for this type of optimization problem given their capacity for efficiently identifying possible solutions from a vast parameter space. By using genetic algorithms, we provide a new approach to the intractable portfolio construction problem of passive management with fewer stocks. The proposed methodology is tested in an empirical application of creating portfolios that track the Deutsche

Aktienindex (DAX), the index of the German stock market in Frankfurt. We first verify whether the genetic algorithm can identify the known optimal solution for the quadratic program. Next, we examine the ability of the genetic algorithm to find tracking portfolios with fewer stocks.

The rest of the paper is organized as follows. The optimization problem is formalized in section 2. Section 3 provides a brief discussion of genetic algorithms in the context of optimization. The DAX index and data are described in section 4. Empirical applications of genetic algorithms are found in section 5. Section 6 concludes the paper and presents possible extensions.

2 The Optimization Problem

We assume that the investor chooses among N stocks. We denote the random vector of daily returns on stock i as $\tilde{\boldsymbol{r}}_i$ and use $r_i = \mathrm{E}(\tilde{\boldsymbol{r}}_i)$ for the expected return on stock i. The N vector $\boldsymbol{r} = [r_1 \ \ldots \ r_N]'$ is the vector of expected returns on these N stocks. We assume that the distribution of returns is stationary.

A portfolio is denoted by a vector of weights. We impose the normalizing restriction that the sum of portfolios weights equals 1. Formally, if $\boldsymbol{p}$ denotes the passive portfolio and $\boldsymbol{i}$ denotes the N vector $[1 \ \ldots \ 1]'$, we have $\boldsymbol{p}'\boldsymbol{i} = 1$. Note that this does not impose non-negativity on the weights, so short sales are allowed. The market portfolio is denoted by the weight vector $\boldsymbol{m}$. We will use $\boldsymbol{x} = \boldsymbol{p} - \boldsymbol{m}$ for the vector of differences in the portfolio weights between the market portfolio and the passive portfolio $\boldsymbol{p}$. Thus, the expected return on the portfolio $\boldsymbol{p}$ can be written as $\boldsymbol{p}'\boldsymbol{r}$ and the portfolio variance is $\boldsymbol{p}'\boldsymbol{V}\boldsymbol{p}$ where $\boldsymbol{V}$ is the symmetric matrix of sample covariances. We assume that the random return vectors are linearly independent so that $\boldsymbol{V}$ can be inverted.

Portfolio selection is traditionally concerned with finding the minimum variance portfolio among portfolios with the same expected rate of portfolio return ϕ. The set of portfolios satisfying this criterion is called *frontier portfolios*. These portfolios are the solution to the quadratic program

$$\begin{aligned} \underset{\boldsymbol{p}}{\text{Minimise}} \quad & \frac{1}{2}\boldsymbol{p}'\boldsymbol{V}\boldsymbol{p} \\ \text{subject to} \quad & \boldsymbol{p}'\boldsymbol{i} = 1 \\ & \boldsymbol{p}'\boldsymbol{r} = \phi. \end{aligned} \tag{1}$$

See Huang and Litzenberger (1988) for the properties of the set of efficient frontier portfolios. For passive management, the expected return difference between the market portfolio

and the passive portfolio is seen to be

$$\boldsymbol{p}'\boldsymbol{r} - \boldsymbol{m}'\boldsymbol{r} = (\boldsymbol{p} - \boldsymbol{m})'\boldsymbol{r} = \boldsymbol{x}'\boldsymbol{r} \equiv \rho.$$

Roll (1992) defines this as the *expected tracking error* which is, in this paper, denoted by ρ. Equivalently, what he calls the *volatility of tracking error*, the expected variance is seen to be

$$(\boldsymbol{p} - \boldsymbol{m})'\boldsymbol{V}(\boldsymbol{p} - \boldsymbol{m}) = \boldsymbol{x}'\boldsymbol{V}\boldsymbol{x}.$$

Roll (1992) notes:

> Thus, current professional investment practice follows a two-dimensional approach to performance. Beating the benchmark on average is tantamount to having a positive expected tracking error. Reducing the volatility of tracking error is tantamount to minimizing the variance of the *difference* between managed portfolio returns and benchmark returns. This is mean/variance analysis but instead of finding the portfolio with the smallest *total return* volatility for a given expected total return, a Markovitz [1959] EV efficient portfolio, money managers are obliged to find the portfolio with minimum tracking error variance for a given expected performance relative to the benchmark.

Therefore, the optimization problem for passive management can be stated as follows: find the portfolio weights that yield a specified target expected tracking error ρ, that is an excess return of ρ between the tracking portfolio and the market portfolio, with the smallest possible tracking error variance:

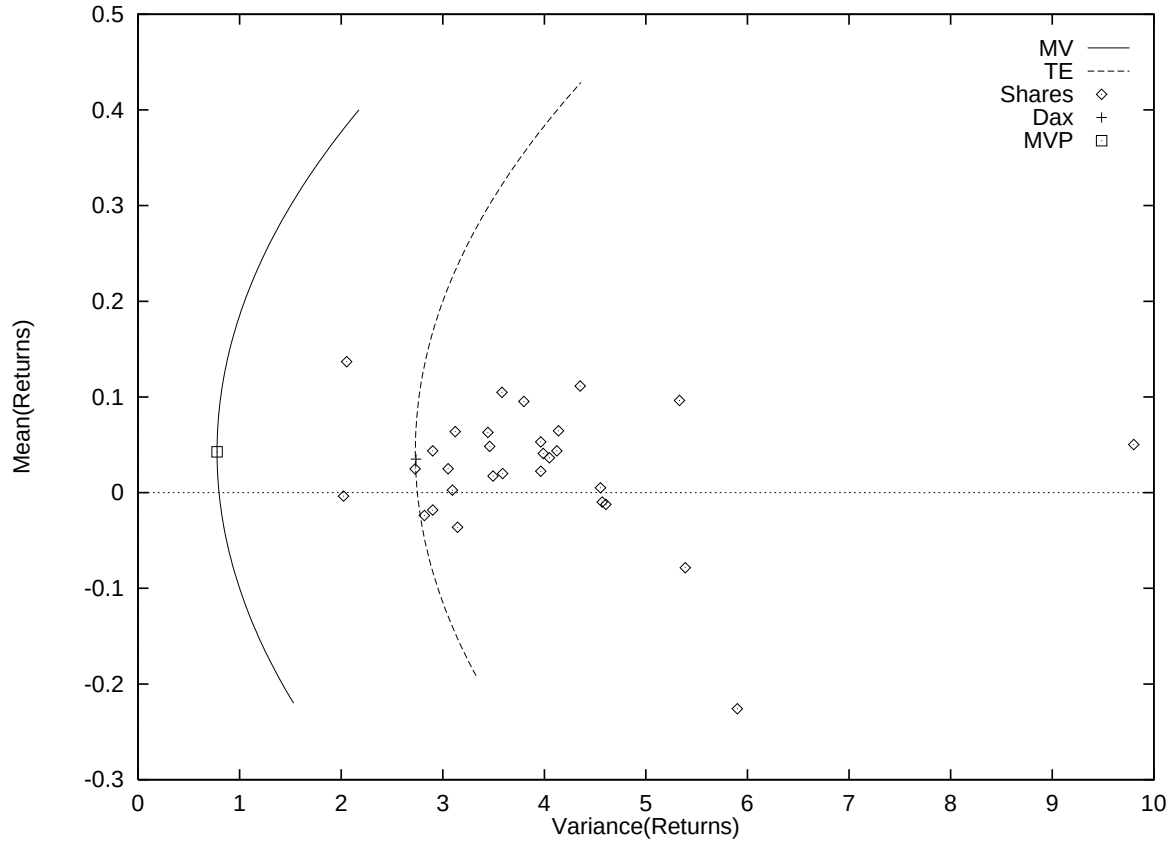
$$\begin{aligned} \underset{\boldsymbol{x}}{\text{Minimise}} \quad & \frac{1}{2}\boldsymbol{x}'\boldsymbol{V}\boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{x}'\boldsymbol{i} = 0 \\ & \boldsymbol{x}'\boldsymbol{r} = \rho. \end{aligned} \tag{2}$$

The first constraint is equal to 0 because both $\boldsymbol{m}'\boldsymbol{i}$ and $\boldsymbol{p}'\boldsymbol{i}$ are constrained to 1. Roll (1992) derives the solution to this problem using a simple Lagrangian. The solution vector $\boldsymbol{x}$ is a set of portfolio alterations that are self-financing (as $\boldsymbol{x}'\boldsymbol{i} = 0$).

Figure 1 displays both the mean-variance efficient frontier, 'MV', and the tracking error efficient frontier 'TE', calculated for the data set described in the next section. Also displayed is the locus of the minimum variance portfolio, 'MVP', the locus of the market index, 'DAX', and the thirty shares that comprise the DAX.

This paper extends the work by Roll (1992). Using the index vector $\boldsymbol{j}$ where $j_i \in \{0, 1\}$

Figure 1: Mean/Variance and Tracking Error Efficient Portfolios



Note: The efficient frontier and the tracking error frontier were calculated following the procedures in Huang and Litzenberger (1988) and Roll (1992), respectively, with the data described in section 4. As these data comprise only a year, and a year in which the market index moved mostly sideways, the vertex of the two parabolas is lower than usually seen in textbooks where several years of data from an upward moving market might have been used.

$\forall\ i = 1, \ldots, N$, we can state the model of minimizing the tracking error variance using only up to k stocks as

$$\begin{aligned} \underset{\boldsymbol{p},\boldsymbol{j}}{\text{Minimise}} \quad & (\boldsymbol{p}-\boldsymbol{m})'\boldsymbol{V}(\boldsymbol{p}-\boldsymbol{m}) \\ \text{subject to} \quad & (\boldsymbol{p}-\boldsymbol{m})'\boldsymbol{r} = \rho \\ & p_i \geq 0 \quad \text{if and if} \quad j_i = 1 \\ & p_i = 0 \quad \text{if and if} \quad j_i = 0 \\ & \boldsymbol{j}'\boldsymbol{i} \leq k \\ & \boldsymbol{p}'\boldsymbol{i} = 1 \end{aligned} \tag{3}$$

This problem differs from problem 2. It is now formulated in terms of $\boldsymbol{p}$, an actual portfolio, rather than $\boldsymbol{x}$, a set of portfolio alterations. We impose constraints such that

all elements of $\boldsymbol{p}$ are non-negative and that, at the most, k elements are positive — in other words that the portfolio comprises holdings of up to k shares where short sales are excluded. We also drop the factor $1/2$ as we are not writing out first derivatives.

This problem is computationally *hard* because of its combinatorial explosion. Consider the German stock market index DAX, which is comprised of 30 shares. Therefore, we can build

$$\binom{30}{20} = 30\,045\,015$$

different portfolios of size 20 using the 30 different DAX shares. In order to find the global minimum variance portfolio, we must solve a quadratic program for the optimal weights for *each* portfolio. While this might still be considered feasible, the number of, say, 300 share portfolios out of the Standard & Poors SP500 index is already seen to be

$$\begin{aligned}\binom{500}{300} &= 5\,054\,949\,849\,935\,532\,221\,313\,990\,781\,909\,778\,483\,343\,803\,009\,698\,868 \\ &\qquad 187\,076\,680\,591\,556\,272\,248\,008\,055\,135\,883\,642\,183\,491\,723\,760 \\ &\qquad 259\,953\,174\,958\,042\,846\,026\,025\,943\,757\,858\,951\,462\,375\,963\,470 \\ &= 5 \cdot 10^{147} \approx 5 \cdot 2^{490}\end{aligned}$$

where we use the standard approximation that $2^{10} \approx 10^3$. This illustrates the fact that this optimization problem is NP-complete. In practice, analytical solutions are not available. This motivates the application of genetic algorithms to the portfolio construction problem of passive management with fewer stocks.

3 Genetic Algorithms

This section provides a brief discussion of canonical genetic algorithms in the context of function optimization. Standard optimization methods, given *one* initial starting point, exploit problem-specific knowledge iteratively in order to improve the current solution. For example, the conjugate gradient and quasi-Newton methods evaluate the derivative of the objective function at the current solution and then choose a new solution using this derivative information. These steps are repeated until a stopping criterion is met.

Genetic algorithms, on the other hand, are a population-based method that use selection and recombination to proceed. Genetic algorithms have been applied to a wide range of problems, see Fogel (1994) and Goldberg (1994) for recent surveys. A genetic algorithm always starts from an initial *population* of n elements. Each population element is a prototype solution. The *representation* of a prototype solution is important, as

is the *objective function*. Typically, binary strings are used as an encoding in order to sample a maximum number of schemata (Holland 1992). It is worth stressing that the representation and the objective function are the only problem-specific parts of a genetic algorithm application. No problem-specific knowledge is required for the operations of the genetic algorithm itself.

For each iteration, all the elements in the population are evaluated through the objective function and assigned an objective value. This allows sorting and ranking of the current generation of prospective solutions before *selection*. Here, an intermediate generation of size n is formed by randomly choosing elements of the population with replacement. In the Darwinian spirit of the 'survival of the fittest', the better the objective function value of a population element, the higher is the probability of it being selected for inclusion in the intermediate generation. The next generation is created from the intermediate generation using *crossover* and *mutation* until n new elements have been formed. For crossover, two 'parents' are selected at random from the intermediary generation. Crossover is applied with a high probability p_c, typically chosen between 0.6 and 0.95. When crossover is applied, two crossover points are selected randomly and the binary information between the two crossover points is swapped. As an example, consider two binary strings of length ten: 1111111111 and 0000000000 and suppose that 2 and 7 were the crossover points. Then two new elements 1100001111 and 0011110000 are created by swapping the binary coding between the two crossover points. With probability $1 - p_c$, the new elements are simply a copy of the parents. A second operator, mutation, inverts the bit it is applied to. Mutation is executed with a low probability p_m of between 0.001 and 0.01 on every bit of the newly created generation. Thus a new generation has been formed. This iterative process is repeated until a stopping criterion, either excess of computing time or sufficient convergence, is met.

Genetic algorithms are hence seen to be a global, randomized search method. What can be said about their convergence properties? Unfortunately not that much. Performance depends not only on the problem chosen, but also on its representation. Variations for the selection method, the crossover and mutation operators, as well as for the different parameters such as population size and operator probabilities, have been proposed to trade off between *premature convergence*, that is steep initial progress that ends in a local, rather than global, optimum, and slow convergence speed. Rudolph (1994) has shown that the canonical genetic algorithm with mutation, crossover and proportional reproduction *never* converges to the global optimum of a static optimization problem. Yet, when an *elitist* selection strategy, which preserves the best elements in the population is used, the canonical genetic algorithm is shown to converge to a global optimum, albeit with unknown convergence speed.

Empirically, genetic algorithms have been successfully applied to a wide range of problems. They do not require derivatives and can hence be used for non-differentiable or non-continuous objective functions. They do not require problem specific knowledge and can thus be applied with relative ease to different problem domains. They have been seen to be very useful for problems with vast, unstructured parameter spaces. Finally, they are also easy to hybridize, that is to combine with a fast local method that complements their capacity of identifying promising regions in the parameter space.

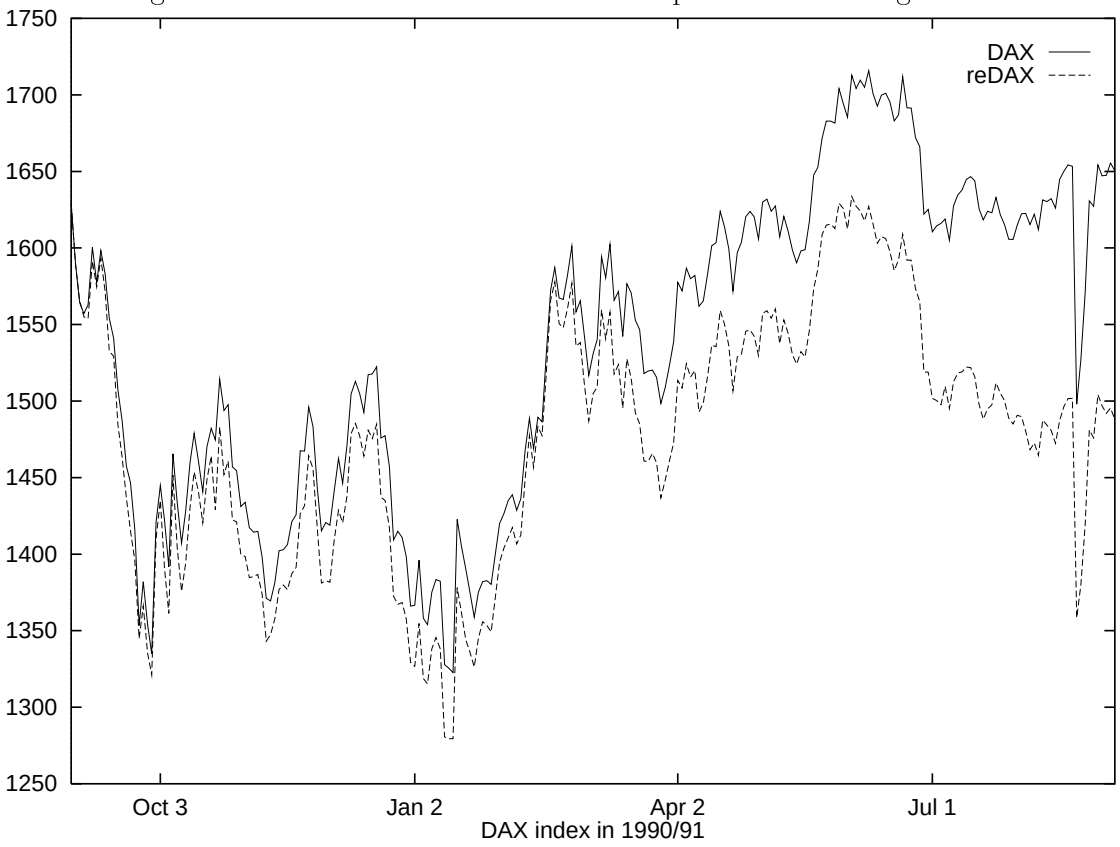
In this paper, we use binary strings of length $15k$ to represent portfolios of k shares. Each string is interpreted as a $2k$ vector. The first k elements are 'selectors' that can take on integer values between 1 and 30 where a value of j is interpreted as selecting shares of company j. We use a binary string of length 5 to represent 30 different values out of the possible 32 by taking the decimal value of the bitstring *modulo* 30. The second half of the $2k$ vector is used for the investment weight of each of the k selected shares. A binary string of length 10 allows for 1024 different values from an equally-spaced grid between 0 and 1. Note that this imposes a constraint of no short sales which we could relax by mapping onto the $[-1, 1]$ interval. A normalization function adds the k different weights and divides by their sum to give investment percentages. As for the parameter values, preliminary runs have shown good success with a 'micro' genetic algorithm: low population size of $n = 25$, high crossover probability $p_c = 0.9$ and low mutation probability $p_m = 0.009$. We use 2-point crossover and an elitist, unbiased selection strategy. As genetic algorithms are a randomized search method, we repeat each experiment $n = 100$ times and compute statistics for the sample of size n of reported results. We use the `GENESIS` genetic algorithm program library (Grefenstette, Davis, and Cerys 1991). Note that standard genetic algorithm programs do not allow for the formal imposition of constraints, as for example linear or quadratic programming libraries do. Constraints are imposed indirectly through the chosen representation.

4 Data

This study uses one year of German stock market data covering the period from September 3, 1990, until August 30, 1991. Returns are calculated from the daily closing prices for the thirty shares that are included in the Deutsche Aktienindex (DAX) as well as the DAX series itself using the standard formula $100(P_t - P_{t-1})/P_{t_1}$ that maps into percentages. Table 1 shows the composition of the DAX at the beginning of the data period.

The DAX is a real-time, capitalization-weighted, performance index — every cash-flow is completely reinvested in the company that generated the cash-flow. Therefore, the relative weights of the DAX are not constant over time. This creates a residual tracking

Figure 2: Deutsche Aktienindex from Sep. 1990 until Aug. 1991



Note: DAX data was provided by Deutsche Bank Research AG. See section 4 for the reDAX.

error when compared to a passive management portfolio because the latter is restricted to have fixed investment weights for each share. In order to simplify the analysis, we recalculate the DAX for every day of the period with the fixed index weights from the last column of table 1. This recalculated DAX, denoted reDAX, represents the evolution of the German stock index under the simplifying assumption that no company pays dividends, splits shares or issues new shares during the data period. As all of the thirty companies actually generate an extra cash-flow that is present in the DAX but not in our reDAX, the latter trails the former in performance. This is irrelevant for the present analysis as we are concerned only with methods of minimizing tracking error variance. Using the reDAX instead of the DAX simplifies the analysis because the global optimum for a portfolio of size thirty is now easily identifiable — it is the vector of weights used for the reDAX which yields a tracking error variance of zero. Figure 2 shows the difference in the evolution over the data period of the DAX and the artificial reDAX.

5 Applications

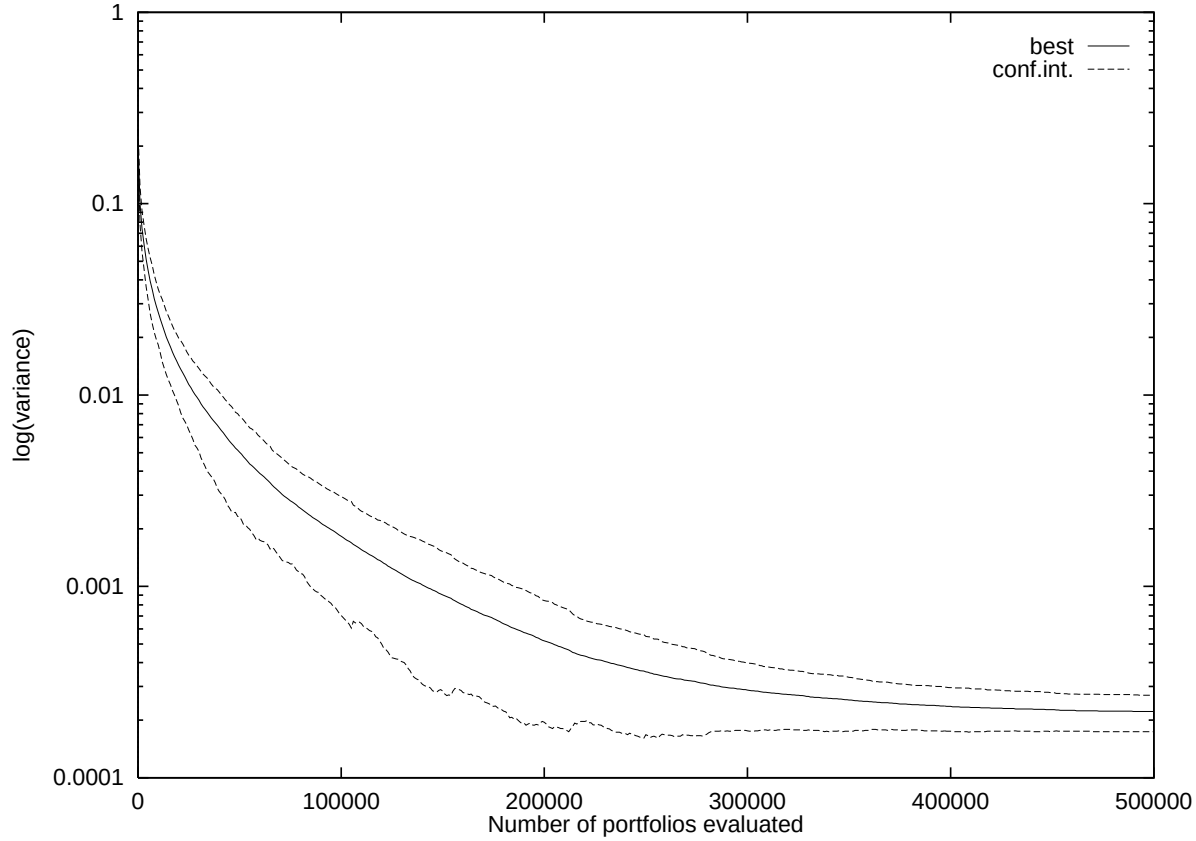
The first empirical illustration investigates how the genetic algorithm manages to find a solution to the minimization problem (2), i.e. to find a portfolio with minimum tracking error variance for a given tracking error ρ. If we set the expected tracking error ρ equal to zero, and as before drop the term 1/2, the minimization problem (2) simplifies to

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{Minimise}} \quad & \boldsymbol{x}'\boldsymbol{V}\boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{x}'\boldsymbol{i} = 0 \\ & \boldsymbol{x}'\boldsymbol{r} = 0. \end{aligned} \tag{4}$$

It is easy to see that the composition of the minimum tracking error variance portfolio, for an expected return equal to that of the market portfolio, is trivially equal to the composition of the index: $\boldsymbol{p}^\star = \boldsymbol{m}$ which is equal to $\boldsymbol{x}^\star = \boldsymbol{0}$. This solution attains the global minimum of the objective function, zero. This problem provides a good benchmark for assessing the convergence of the genetic algorithm. Through the binary representation of the solution vector, we impose the restriction that every DAX index stock must be included in the portfolio and we let the genetic algorithm search for the DAX index weights. Note that we are not actually imposing $\boldsymbol{x}'\boldsymbol{r} = \rho = 0$ as the `GENESIS` library used in this paper does not allow for constraints. Subsequent research will investigate the potential of genetic algorithms with linear and non-linear constraints (Michalewicz 1994). Figure 3 shows the evolution, plotted onto a logarithmic scale, over the number of portfolio evaluations. The graphs and tables in the remainder of the paper display the standard deviations of the tracking errors and not the variances. Taking the square root of the variances is a strictly monotonic transformation which actually increases the values since all the tracking error variances that we report are smaller than one.

From figure 3, we can see that the genetic algorithm approaches the optimal solution. The speed at which portfolio variance is minimized diminishes over the course of the optimization. The standard deviation of tracking error of the best portfolio in the randomly chosen initial population is seen to be approximately 0.21. From this value, the standard deviation declines rapidly to values of about 0.01 after 30 000 evaluations, but then takes 141 000 evaluations to reach 0.001. The experiment is terminated after 500 000 evaluations with the best portfolio having a tracking error standard deviation of $2.22 \cdot 10^{-4}$, or a tracking error variance of $4.84 \cdot 10^{-8}$. Keeping in mind that the discrete representation of portfolio weights in binary strings of length ten used by the genetic algorithms results in some residual tracking error variance, we conclude that the genetic algorithm does indeed obtain the solution to this problem.

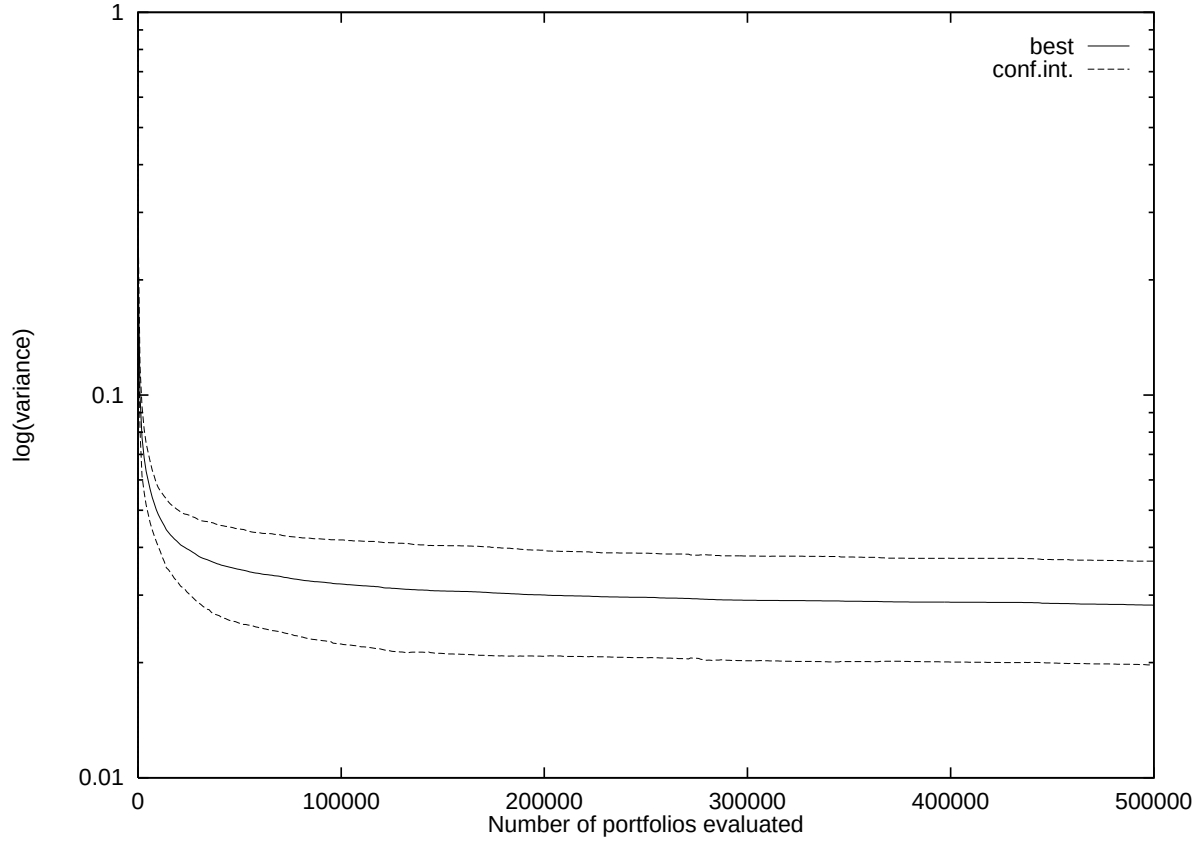
Figure 3: Convergence of Genetic Algorithm — First Application



Note: The genetic algorithm was parametrized as outlined at the end of section 3. 'Best' shows the sample mean over the 100 experiments of the square root of minimum portfolio tracking error variance after the given number of evaluations. The confidence interval is of width 1.96 sample standard deviations, also calculated over these 100 experiments .

Next, we consider a slightly more complex problem. The first empirical application imposed the inclusion of all thirty DAX shares in the tracking portfolio through the binary representation: a binary string used by the genetic algorithm was interpreted as a $30 \cdot 1$ vector where the j-th element was taken as the investment proportion for share j. We now remove this implicit interpretation and use the representation outlined in the last paragraph of section 3 which is used in the remainder of this paper. Here, the binary string contains $2k$ elements: thirty selectors followed by thirty weights. The j-th weight is now used for the share indicated by the j-th selector. For this second application, we set $k = 30$. While this imposes a portfolio that comprises thirty shares, our representation does not impose the selection of thirty *different* shares. The genetic algorithm is free to select any of the thirty shares that are available more than once, of course, at the expense of unchosen shares. The size of the parameter space is now increased substantially when compared to the first empirical application — the genetic algorithm now has to select

Figure 4: Convergence of Genetic Algorithm — Second Application



Note: The genetic algorithm was parametrized as outlined at the end of section 3. 'Best' shows the sample mean over the 100 experiments of the square root of minimum portfolio tracking error variance after the given number of evaluations. The confidence interval is of width 1.96 sample standard deviations, also calculated over these 100 experiments .

which shares to use, and also the weight of each of the thirty shares — making this second optimization problem more complex.

Figure 4 shows the evolution of the solutions that the genetic algorithm finds for this second application. The solution for the first application, which the genetic algorithm has attained, was to replicate the weights found in the market index. Combining these weights with the selection of every index stock, where the j-th selector actually selects share j, would also provide a solution to our second application (as would all the different orderings of such a selection vector). But we have to acknowledge that the genetic algorithm, as seen from figure 4, does not perform as well as before. The sample mean of the standard deviation of tracking error over 100 experiments, taken after 500 000 evaluations, is $2.83 \cdot 10^{-2}$, which corresponds to a variance of $7.98 \cdot 10^{-4}$. While this a small value, it is some orders of magnitude behind the result that we obtained in the previous experiment. A possible explanation is that the representation which concatenated

the selection vector and a weight vector is not as appropriate as a representation with weights only. Michalewicz (1994) claims that non-binary representations are better suited to problem domains such as this. An empirical investigation for this problem is left for future research. Another possible explanation is that more diversity in the population, obtained through a bigger population size, could help.

The third application is the most comprehensive one. We expand the analysis to a series of experiments where we vary k, the number of shares that can be selected for a portfolio, from 4 to 30. That is, we let the genetic algorithm solve problem (3) for different values of k that we impose. Thereafter, we analyze the minimum tracking error variance, or rather its square root, that was found as a function of k. Table 2 presents descriptive statistics for the solutions that the genetic algorithm obtains for this problem. As expected, the tracking error standard deviation decreases monotonically with increasing portfolio size. The variation of the tracking error standard deviation, calculated as the sample standard deviation over the sample of 100 experiments and reported in the second column of table 2, is remarkably stable. We see from the third column that the skewness coefficient is positive for all but five of the twenty-seven values and in four instances has a value that is greater than one, corresponding to a mass of probability of 0.99998 — we clearly reject normality for these four experiments.[1] As for the excess kurtosis, we see values close to zero and values as big as 2.45. The signs of the excess kurtosis coefficient change frequently; no obvious pattern can be detected. Column six in table 2 reports the marginal probability values for a test statistic for normality that uses both the skewness and the excess kurtosis.[2] At the 5% level, we reject the null hypothesis of normality for thirteen of the twenty-seven values. But we also fail to reject the null hypothesis thirteen times at a level of 15%. Clearly, the sample of size 100 is not big enough to draw firm conclusions. As for the minimum and maximum values of the tracking error standard deviation that are reported in the last two columns of table 2, we see that these values decrease with increasing portfolio size, albeit not monotonically.

An important verification of the validity of these results can be obtained by comparing

[1] We calculated the skewness and excess kurtosis as

$$\text{skew} = \frac{1}{n}\sum_{1}^{n}\left(\frac{x_i-\mu}{\sigma}\right)^3 \qquad \text{and} \qquad \text{kurt} = \frac{1}{n}\sum_{1}^{n}\left(\frac{x_i-\mu}{\sigma}\right)^4 - 3.0$$

where μ and σ denote the sample mean and the sample standard deviation. Asymptotically, both $(n/6)^{1/2}\cdot\text{skew}$ and $(n/24)^{1/2}\cdot\text{kurt}$ are distributed as $N(0,1)$.

[2] This test statistic is computed as

$$w = n\left(\frac{(\text{skew})^2}{6} + \frac{(\text{kurt})^2}{24}\right)$$

and is distributed as a χ^2 with two degrees of freedom under the null hypothesis that the sample is drawn from a normal distribution. The marginal probability value is then calculated as $1-\chi^2(2,w)$.

the outcome with hitherto unused data. From the outset, we have split the twelve months of data into an *in-sample* period and several *out-of-sample* periods. The in-sample period covers the first nine months of the data set. The remaining three months of our data set are first used for three out-of-sample periods with a length of one month each. The increasing distance of these separate out-of-sample periods to the in-sample period, which is used to generate the tracking portfolios, provides insight on how an unupdated portfolio would perform. Finally, we group the three months into a single longer out-of sample period. From table 3, we can assess the usefulness of the genetic algorithm by comparing the tracking error standard deviation from the in-sample period with both the optimal value and the out-of-sample values. The optimal value is calculated for each solution that the genetic algorithm reported at the end of each of the 50 000 iterations. The quadratic programming problem is solved numerically, given the stocks in a reported solution, using the CFSQP solver by Lawrence, Zhou, and Tits (1994). This provides a benchmark for how the tracking error standard deviation can improve by choosing better weights for the given selection of stocks. Both the in-sample and the optimal tracking error standard deviation are calculated over the initial nine months of the data set.

Comparing the tracking error standard deviation obtained after only 50 000 evaluations with the benchmark values from solving the corresponding quadratic program is illuminating. We can quantify the difference between the sample means for both approaches using a t-test. The marginal probability values for this test are shown in the fourth column of table 3. For portfolios of size smaller than $k = 22$, we fail to reject the null hypothesis that the sample means of the tracking error standard deviations from the genetic algorithm are equal to those of the quadratic program, even at the 20% level. For $k = 22$ and $k = 23$, the marginal probabilities of rejecting this null hypothesis are roughly 10% and 5%, respectively. Only for larger portfolio sizes is the difference between the two methods highly significant. So the genetic algorithm does locate portfolios that contain fewer shares than the market index, and these portfolios have tracking error standard deviations that, for medium size portfolios, are not statistically different from those obtained using the quadratic program.

It is of course important to see how these portfolios perform on different, unused samples of data. The remaining four columns of table 3 permit us to quantify this behaviour. For the lowest portfolio sizes, especially the shortest term performance where we compare against only one month of out-of-sample data, the tracking error standard deviation can be both considerably higher — or lower — than in the in-sample period. Interestingly enough, the first and the third months give higher tracking error standard deviations whereas for the second month of the out-of-sample period, the tracking error standard deviation decreases. When the out-of-sample data are grouped into a single three

month period, the tracking error is only slightly higher than in the in-sample period.

The sample means reported in table 3 give an impression of the sampling distribution of the tracking errors for the different data periods. The non-parametric Spearman rank correlation test allows us to quantify the correlation between the results from the in-sample period and the different out-of-sample periods. Moreover, it allows us to test the null hypothesis of no correlation between two respective periods. In table 4, the test values for this non-parametric rank correlation r_s and the marginal p-value of the null hypothesis of no correlation are shown. Several conclusions can be drawn. The correlation between the in-sample tracking error standard deviation and the tracking error standard deviation obtained from solving the quadratic program is very high, but declines with increasing portfolio size. As can be seen from the illustration that we provided at the end of section 2, the solution space increases with the portfolio size. In this setup, we do not increase the number of evaluations with the portfolio size k. For every value of k, we stop each experiment after 50 000 evaluations. For larger k a smaller portion of the solution space, which increases with k, is sampled, even though the genetic algorithm evaluates the same absolute number of portfolios. Presumably, increasing the number of evaluations as a function of k would also increase the correlation of the tracking error standard deviations between the two methods. For portfolios of size smaller than ten, twelve out of the twenty-four rank correlation test statistics obtained from comparing against the out-of-sample periods are negative. High in-sample tracking errors can correspond to low out-of-sample tracking errors. Ten out of these twelve negative correlations are significant at a test level of 5%. Focusing now, and for the rest of the analysis, on portfolio sizes greater than or equal to ten, we see that all correlations are positive. Generally, the correlation increases with the portfolio size. We observe the smallest correlations between the in-sample period and the first out-of-sample period. From the fifth column in table 4, we see that eight out of twenty-one test statistics for portfolio sizes greater than or equal to ten are not significant at the 10% level. On the other hand, we see more correlation between the second out-of-sample period and the in-sample period. All of the test statistics for the second out-of-sample period reject the null hypothesis of no correlation even at the 2% level. For the third period, two test statistics are highly insignificant, but the null hypothesis of no correlation with the in-sample period can be rejected at the 3% level in the other nineteen cases. Finally, and most importantly, when we compare the in-sample results with the single three-month out-of-sample period as reported in the last two columns of table 4, we see that all test statistics reject the null hypothesis of no correlation at the significance level of 0.5%. Clearly, good in-sample results correlate with good out-of-sample results.

Table 5 shows the actual tracking errors ρ, that is the values of the excess returns over

the index or $\boldsymbol{x}'\boldsymbol{r} = (\boldsymbol{p} - \boldsymbol{m})'\boldsymbol{r}$. Also displayed is the Sharpe ratio (Sharpe 1994), that is the ratio of the mean of the sample excess return over the sample standard deviation of the excess returns. Interestingly enough, for the in-sample period, the sample mean of the tracking error, computed over the 100 experiments, is always positive. For the out-of-sample periods, most portfolio sizes k yield a small negative tracking error. The absolute size of these returns is very small. Based on the Sharpe ratio which, multiplied by the square root of the sample size, equals the usual t-statistic for the significance of a sample mean, we can easily verify that only 15 out of the 137 sample means for the tracking error are significantly different from zero at the 10% level. Yet, out of these 15, not one sample mean is significantly different from zero at the 1% level. Moreover, for the fourth out-of-sample period that comprises three months of data, not one sample mean is significantly different from zero even at the 20% level. From table 5, we can conclude that the minimization of the tracking error variance does not come at the price of statistically significant negative tracking error.

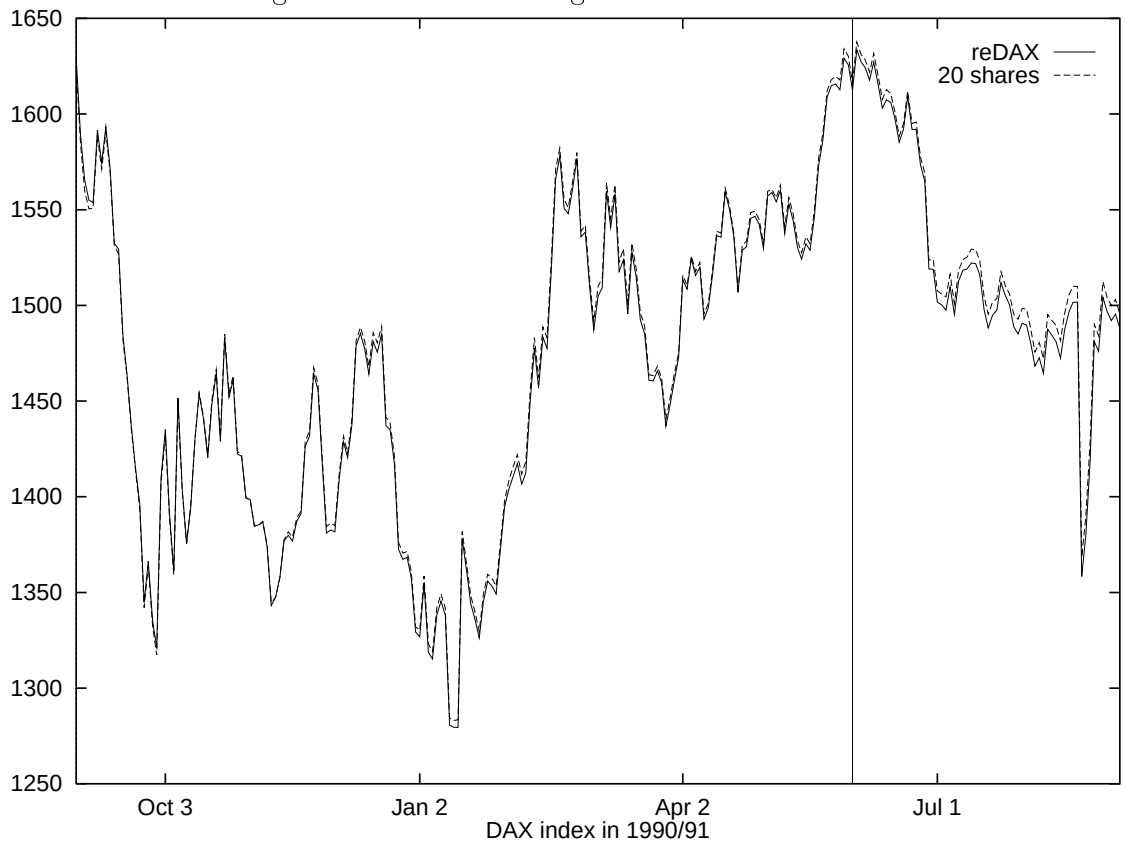
As a final note, figure 5 allows us to visualize the success of the proposed approach for constructing index tracking portfolios from a subset of stocks. The figure shows the benchmark index and the best tracking portfolio of size $k = 20$ over the data period. We observe that there is virtually no difference between the index and the tracking portfolio during the in-sample period, and very little variation during the out-of-sample period that we use for validation.

6 Conclusion

This paper has formalized and discussed the optimization problem faced by passive management investors. We have seen that it is computationally intractable to find the analytic solution to the problem of minimizing tracking error variance using only a subset of shares. In order to overcome this problem, we have offered a new approach using genetic algorithms for stock market index tracking when only a subset of the index stocks is used.

Extended empirical applications in this paper have shown that a canonical genetic algorithm, using standard encoding techniques and operators, can be applied successfully to this problem. Our empirical analysis shows that index tracking portfolios created by the genetic algorithm have an in-sample performance that correlates significantly with their out-of-sample performance. The analysis also shows that minimizing the variance of tracking error does not come at the expense of statistically significant tracking error. Finally, for portfolios of small and medium size, there is no statistically significant difference between the tracking error standard deviations from the genetic algorithm and those from the quadratic program.

Figure 5: Index Tracking Portfolio and reDAX



Note: The series 'reDAX' shows the recalculated DAX as described in section 4. The series '20 shares' shows the best 20 share tracking portfolio. The vertical line marks the division between the in-sample period used for the optimization and the out-of-sample period which is used for validation only.

Several extensions are left for future research. Two different approaches to enhancing the genetic algorithm are available. First, we can hybridize the genetic algorithm by combining it with a constrained quadratic program solver such as CFSQP (Lawrence, Zhou, and Tits 1994) to combine the power in locating promising regions in the solution space of the former with the local convergence properties of the latter. Second, a constrained genetic algorithm (Michalewicz 1994) might identify solutions that are superior, from an investment viewpoint, by restricting search to solutions with positive tracking error. Finally, despite the vast literature on time-varying conditional covariances in financial time series, we have in this first application used only a simple unconditional estimate of the covariance matrix of returns. Integrating the genetic algorithm for passive management with a multivariate model of conditional covariances should lead to superior passive investment results.

Appendix

Table 1: Composition of the DAX index as of Sep. 3, 1990

Company	Symbol	Capital	Price	Percentage
Allianz	ALV	825.0	2570.00	12.47
BASF	BAS	2849.8	235.50	3.95
Bayer	BAY	3200.2	247.00	4.65
Bayr. Hypo. u. W'bank	BHW	827.3	359.50	1.75
Bayr. Motorenwerke	BMW	790.6	484.00	2.25
Bayr. Vereinsbank	BVM	617.3	382.00	1.39
Commerzbank	CBK	1272.7	263.50	1.97
Continental	CON	445.5	300.00	0.79
Daimler-Benz	DAI	2327.4	699.00	9.57
Degussa	DGS	365.0	348.00	0.75
Deutsche Babcock	DBC	350.0	151.00	0.31
Deutsche Bank	DBK	2201.2	692.00	8.96
Dresdner Bank	DRB	1819.7	413.00	4.42
Henkel	HEN3	302.5	595.50	1.06
Hoechst	HFA	2920.0	237.50	4.08
Karstadt	KAR	360.0	689.00	1.46
Kaufhof	KFH	478.6	590.00	1.66
Linde	LIN	291.6	880.00	1.51
Lufthansa	LHA	1520.0	130.00	1.16
MAN	MAN	771.0	359.00	1.63
Mannesmann	MMW	1449.2	274.50	2.34
Metallgesellschaft	MET	385.1	569.90	1.29
Preussag	PRS	709.4	346.00	1.45
RWE	RWE	2215.3	418.00	5.45
Schering	SCH	341.1	652.00	1.31
Siemens	SIE	2534.2	626.00	9.33
Thyssen	THY	1665.0	233.50	2.15
VEBA	VEB	2226.7	354.00	4.64
VIAG	VIA	789.0	354.00	1.64
Volkswagen	VOW	1650.0	475.00	4.61
Sum		38401.0		100.00

Note: Data are from Frankfurter Wertpapierbörse AG. Capital is in millions of DM, price is in DM.

Table 2: Descriptive Statistics of Minimum Tracking Error Standard Deviation

k	mean	stdev.	skew.	kurt.	p-val.	min.	max.
4	0.267	0.005	1.815	1.445	0.000	0.264	0.280
5	0.219	0.008	0.961	-0.948	0.000	0.214	0.238
6	0.193	0.005	0.723	0.172	0.012	0.187	0.205
7	0.169	0.006	-0.393	-1.562	0.002	0.161	0.180
8	0.153	0.005	0.431	-0.957	0.032	0.147	0.164
9	0.140	0.004	0.781	-0.571	0.003	0.136	0.149
10	0.126	0.004	0.336	-0.170	0.368	0.120	0.139
11	0.114	0.004	1.025	0.711	0.000	0.109	0.129
12	0.104	0.005	1.516	2.450	0.000	0.099	0.124
13	0.095	0.005	1.087	0.129	0.000	0.089	0.110
14	0.088	0.005	0.593	-0.383	0.039	0.082	0.103
15	0.082	0.005	0.328	-0.264	0.353	0.074	0.093
16	0.077	0.005	0.732	0.650	0.005	0.068	0.092
17	0.072	0.005	0.232	-0.305	0.527	0.063	0.086
18	0.069	0.005	0.520	0.408	0.074	0.059	0.084
19	0.065	0.005	0.217	-0.255	0.590	0.054	0.079
20	0.061	0.005	0.725	0.952	0.002	0.050	0.080
21	0.058	0.005	-0.067	-0.476	0.600	0.046	0.070
22	0.056	0.004	-0.034	-0.081	0.977	0.046	0.068
23	0.053	0.005	0.596	1.171	0.003	0.041	0.071
24	0.050	0.004	-0.005	-0.005	1.000	0.039	0.060
25	0.048	0.004	0.259	0.144	0.549	0.039	0.060
26	0.045	0.004	0.194	-0.332	0.580	0.036	0.056
27	0.042	0.004	0.024	-0.593	0.478	0.030	0.051
28	0.040	0.005	0.408	-0.379	0.185	0.031	0.052
29	0.038	0.005	-0.308	0.667	0.179	0.021	0.048
30	0.035	0.004	0.191	-0.726	0.246	0.027	0.046

Note: The genetic algorithm was parametrized as outlined at the end of section 3. The columns report the following values: mean, standard deviation, skewness, excess kurtosis, p-value for normality, minimum and maximum. For each value of k, a sample of size 100 was obtained by running 100 experiments and retaining the best tracking error standard deviation (i.e. square root of the value of the objective function) after 50 000 evaluations for each of these experiments. The p-value is for the null hypothesis that the sample values are drawn from a normal distribution. The probability value is the significance level of the test statistic, calculated as $1 - \chi^2(2, w)$ where w denotes the test statistic described in the text. The χ^2 distribution values were calculated using the `chdtr` function from the `cephes` library (Moshier 1989).

Table 3: In-Sample Tracking Error versus Out-Of-Sample and Optimal Values

	In-sample period			Out-of-sample periods			
k	GA	QP	p-val.	GA p1	GA p2	GA p3	GA p4
4	0.267	0.267	1.000	0.379	0.164	0.274	0.287
5	0.219	0.219	0.995	0.329	0.192	0.268	0.270
6	0.193	0.193	0.950	0.268	0.163	0.233	0.227
7	0.169	0.169	0.999	0.239	0.155	0.200	0.202
8	0.153	0.153	0.971	0.227	0.146	0.181	0.189
9	0.140	0.140	0.932	0.198	0.129	0.172	0.170
10	0.126	0.126	0.928	0.179	0.112	0.153	0.152
11	0.114	0.114	0.909	0.157	0.102	0.138	0.136
12	0.104	0.104	0.949	0.142	0.094	0.126	0.123
13	0.095	0.095	0.817	0.130	0.084	0.120	0.113
14	0.088	0.088	0.905	0.124	0.078	0.107	0.105
15	0.082	0.082	0.675	0.118	0.072	0.097	0.098
16	0.077	0.076	0.765	0.107	0.067	0.088	0.089
17	0.072	0.072	0.621	0.102	0.063	0.083	0.084
18	0.069	0.069	0.457	0.098	0.061	0.079	0.081
19	0.065	0.064	0.482	0.086	0.055	0.072	0.073
20	0.061	0.061	0.232	0.085	0.054	0.069	0.071
21	0.058	0.058	0.231	0.077	0.051	0.063	0.065
22	0.056	0.055	0.104	0.074	0.049	0.062	0.063
23	0.053	0.052	0.045	0.068	0.046	0.058	0.059
24	0.050	0.049	0.003	0.062	0.045	0.053	0.055
25	0.048	0.046	0.000	0.057	0.043	0.052	0.052
26	0.045	0.043	0.000	0.052	0.040	0.047	0.047
27	0.042	0.040	0.000	0.049	0.037	0.045	0.045
28	0.040	0.038	0.000	0.045	0.036	0.041	0.042
29	0.038	0.036	0.000	0.043	0.035	0.040	0.040
30	0.035	0.033	0.000	0.038	0.031	0.035	0.036

Note: The genetic algorithm was parametrized as outlined at the end of section 3. The columns contain, respectively, the in-sample tracking error standard deviation (same as in column 2 of table 2), the resulting minimal tracking error standard deviation obtained by solving the corresponding quadratic program, the marginal p-value for the null hypothesis that the in-sample tracking error standard deviation is equal to the one from the quadratic program, and the tracking error standard deviations for the four out-of-sample periods described in the text. The sample resulted from running 100 experiments for each of the 27 optimization problems. Each experiment was stopped after 50 000 portfolio evaluations. The quadratic program was solved using CFSQP (Lawrence, Zhou, and Tits 1994).

Table 4: Spearman Rank Correlation Tests Statistics

	Correlation r_s and marginal p-value between in-sample period and									
	optim. val.		o.o.s. p1		o.o.s. p2		o.o.s. p3		o.o.s. p4	
k	r_s	p	r_s	p	r_s	p	r_s	p	r_s	p
4	1.000	0.000	0.330	0.001	0.714	0.000	-0.289	0.004	0.395	0.000
5	0.989	0.000	0.344	0.000	0.270	0.007	-0.542	0.000	0.048	0.636
6	0.996	0.000	-0.446	0.000	-0.396	0.000	0.372	0.000	-0.202	0.044
7	0.999	0.000	0.200	0.046	-0.499	0.000	0.433	0.000	0.474	0.000
8	0.999	0.000	-0.167	0.096	-0.544	0.000	0.062	0.543	-0.387	0.000
9	0.998	0.000	-0.486	0.000	-0.027	0.787	0.359	0.000	-0.228	0.022
10	0.999	0.000	0.015	0.881	0.385	0.000	0.071	0.485	0.286	0.004
11	0.998	0.000	0.029	0.773	0.467	0.000	0.324	0.001	0.465	0.000
12	0.999	0.000	0.111	0.272	0.408	0.000	0.225	0.024	0.551	0.000
13	0.993	0.000	0.146	0.148	0.529	0.000	0.053	0.598	0.478	0.000
14	0.999	0.000	0.233	0.019	0.510	0.000	0.241	0.016	0.482	0.000
15	0.992	0.000	0.370	0.000	0.577	0.000	0.256	0.010	0.517	0.000
16	0.997	0.000	0.126	0.210	0.368	0.000	0.401	0.000	0.440	0.000
17	0.993	0.000	0.104	0.305	0.489	0.000	0.337	0.001	0.345	0.000
18	0.990	0.000	0.201	0.045	0.447	0.000	0.473	0.000	0.436	0.000
19	0.995	0.000	0.362	0.000	0.431	0.000	0.361	0.000	0.500	0.000
20	0.981	0.000	0.161	0.108	0.258	0.009	0.367	0.000	0.318	0.001
21	0.991	0.000	0.400	0.000	0.243	0.015	0.539	0.000	0.530	0.000
22	0.992	0.000	0.245	0.014	0.441	0.000	0.327	0.001	0.386	0.000
23	0.980	0.000	0.221	0.027	0.249	0.012	0.494	0.000	0.434	0.000
24	0.975	0.000	0.306	0.002	0.407	0.000	0.506	0.000	0.472	0.000
25	0.965	0.000	0.006	0.949	0.387	0.000	0.456	0.000	0.329	0.001
26	0.959	0.000	0.332	0.001	0.284	0.004	0.481	0.000	0.485	0.000
27	0.972	0.000	0.353	0.000	0.344	0.000	0.582	0.000	0.588	0.000
28	0.941	0.000	0.492	0.000	0.475	0.000	0.591	0.000	0.670	0.000
29	0.944	0.000	0.417	0.000	0.469	0.000	0.612	0.000	0.640	0.000
30	0.947	0.000	0.562	0.000	0.556	0.000	0.628	0.000	0.761	0.000

Note: The test values and the corresponding probability values for the null hypothesis of no correlation between the two corresponding samples were calculated using the `spear` function from Press, Teukolsky, Vetterling, and Flannery (1992) using the previously described samples of size 100. 'opt.val.' refers to the optimal values that stem from the quadratic program. 'o.o.s.p' is short for out-of-sample period. As before, the first three out-of-sample periods are of one month length and the fourth is of length three months.

Table 5: Sharpe Ratio and Tracking Error

	Tracking Error ρ and Sharpe Ratio s									
	in-sample		o.o.s.p. 1		o.o.s.p. 2		o.o.s.p. 3		o.o.s.p. 4	
k	ρ	s	ρ	s	ρ	s	ρ	s	ρ	s
4	0.005	0.020	-0.007	-0.019	0.080	0.537	-0.019	-0.068	0.020	0.071
5	0.006	0.029	0.030	0.089	0.034	0.183	-0.027	-0.087	0.013	0.044
6	0.004	0.024	-0.001	-0.006	0.012	0.085	-0.036	-0.163	-0.008	-0.035
7	0.007	0.046	-0.012	-0.047	0.008	0.055	-0.037	-0.185	-0.013	-0.065
8	0.006	0.038	-0.006	-0.027	0.002	0.022	-0.034	-0.189	-0.012	-0.064
9	0.006	0.041	-0.008	-0.042	0.003	0.028	-0.039	-0.231	-0.014	-0.084
10	0.003	0.024	-0.006	-0.038	-0.003	-0.021	-0.030	-0.195	-0.013	-0.084
11	0.001	0.011	-0.012	-0.080	-0.007	-0.061	-0.029	-0.210	-0.016	-0.117
12	0.002	0.015	-0.009	-0.062	-0.008	-0.081	-0.026	-0.209	-0.014	-0.117
13	0.002	0.021	-0.005	-0.047	-0.007	-0.086	-0.021	-0.180	-0.011	-0.103
14	0.002	0.018	-0.008	-0.068	-0.009	-0.118	-0.020	-0.199	-0.012	-0.123
15	0.001	0.015	-0.008	-0.071	-0.008	-0.125	-0.017	-0.183	-0.011	-0.117
16	0.001	0.016	-0.008	-0.084	-0.009	-0.139	-0.016	-0.183	-0.011	-0.127
17	0.001	0.017	-0.007	-0.079	-0.007	-0.106	-0.013	-0.167	-0.009	-0.111
18	0.001	0.019	-0.009	-0.102	-0.006	-0.098	-0.013	-0.167	-0.009	-0.115
19	0.001	0.023	-0.008	-0.127	-0.006	-0.105	-0.011	-0.162	-0.008	-0.121
20	0.001	0.016	-0.008	-0.116	-0.004	-0.093	-0.010	-0.155	-0.007	-0.111
21	0.001	0.018	-0.008	-0.121	-0.004	-0.090	-0.008	-0.138	-0.007	-0.109
22	0.001	0.013	-0.007	-0.125	-0.005	-0.096	-0.009	-0.149	-0.007	-0.115
23	0.001	0.019	-0.009	-0.164	-0.003	-0.070	-0.006	-0.114	-0.006	-0.111
24	0.001	0.016	-0.008	-0.157	-0.002	-0.049	-0.005	-0.098	-0.005	-0.095
25	0.001	0.014	-0.008	-0.172	-0.002	-0.054	-0.005	-0.095	-0.005	-0.098
26	0.001	0.014	-0.006	-0.149	-0.001	-0.018	-0.004	-0.077	-0.003	-0.077
27	0.001	0.029	-0.005	-0.134	-0.001	-0.027	-0.004	-0.101	-0.003	-0.081
28	0.001	0.026	-0.005	-0.131	0.000	0.008	-0.002	-0.058	-0.002	-0.056
29	0.001	0.022	-0.004	-0.122	0.001	0.016	-0.002	-0.067	-0.002	-0.050
30	0.000	0.004	-0.005	-0.135	0.000	-0.002	0.000	0.003	-0.001	-0.043

Note: The Sharpe ratio (Sharpe 1994) is defined as the mean of return differences over a benchmark investment divided by the sample standard deviation of the return differences. In this application, it is computed as the sample mean of tracking error over the sample standard deviation of tracking error. Multiplying the Sharpe Ratio by the square root of the sample size, in this case 10, yields the usual t-statistic for the significance of a sample mean. The t-distribution for 99 degrees of freedom has the critical values 1.290, 1.660, 1.984, 2.365 and 2.626 for two-sided tests of levels 20%, 10%, 5%, 2% and 1%, respectively.

References

Eddelbüttel, Dirk (1992). A Genetic Algorithm for Passive Management. Mémoire de D.E.A., GREQE–EHESS, Marseilles.

Fogel, David B. (1994). An introduction to simulated evolutionary optimization. *IEEE Transactions on Neural Networks 5(1)*, 3–14.

Goldberg, David E. (1994). Genetic and evolutionary algorithms come of age. *Communications of the ACM 37(3)*, 113–119.

Grefenstette, John J., Lawrence Davis, and Daniel Cerys (1991). *GENESIS & OOGA: Two Genetic Algorithm Systems*. Melrose, Mass.: TSP Publications. `GENESIS` is also available at `ftp.aic.nrl.navy.mil/pub/galist/src/ga/genesis.tar.Z`.

Holland, John H. (1992). *Adaptation in Natural and Artificial Systems* (Second ed.). Cambridge, Mass.: MIT Press.

Huang, Chi-Fu and Robert H. Litzenberger (1988). *Foundation for Financial Economics*. New York: North-Holland.

Jeffrey, Robert H. and Robert D. Arnott (1993). Is your alpha big enough to cover its taxes? *Journal of Portfolio Management 19(3)*, 15–25.

Lawrence, Craig, Jian L. Zhou, and André L. Tits (1994). User's Guide for CFSQP Version 2.3: A `C` Code for Solving (Large Scale) Constrained Nonlinear (Minimax) Optimization Package, Generating Iterates Satisfying All Inequality Constraints. Institute for Systems Research TR-94-16r1, University of Maryland, College Park, MD.

Markowitz, Harry M. (1959). *Portfolio Selection: Efficient Diversification of Investments*. New Haven: Yale University Press. Reprinted, New York, John Wiley & Sons, 1970.

Michalewicz, Zbigniew (1994). *Genetic algorithms + Data Structures = Evolution Programs* (Second ed.). New York: Springer Verlag.

Moshier, Stephen L. B. (1989). *Methods and Programs for Mathematical Functions*. Ellis Horwood Series in Mathematics and its Applications. Chichester: Ellis Horwood. The routines from this book are availabe at `ftp://netlib.att.com/cephes/`.

Press, William H., Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery (1992). *Numerical Recipes in C: The Art of Scientific Computing* (Second ed.). Cambridge: Cambridge University Press.

Roll, Richard (1992). A mean/variance analysis of tracking error. *Journal of Portfolio Management 18(4)*, 13–22.

Rudolph, Günter (1994). Convergence analysis of canonical genetic algorithms. *IEEE Transactions on Neural Networks 5(1)*, 96–101.

Schramm, Sabine (1995). Index managers get active. *Pensions & Investments 23(21)*, 3–47.

Sharpe, William H. (1994). The Sharpe ratio. *Journal of Portfolio Management 21(1)*, 49–58.

Tabata, Yoshito and Eiji Takeda (1995). Bicriteria optimization problem of designing an index fund. *Journal of the Operational Research Society 46(8)*, 1023–1032.