A Hybrid Genetic Algorithm for Passive Management

Dirk Eddelbüttel
GREQAM–EHESS
Marseilles, France

Second Conference *Computing in Economics and Finance*
Society of Computational Economics
Geneva, Switzerland, 26-28 June 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. This problem is computationally hard as it belongs to the class of NP complete problems as pointed out by Tabata and Takeda (1995) who proposed a heuristic solution.

In Eddelbüttel (1996), we used a canonical genetic algorithm with standard binary encoding techniques and genetic algorithm operators for the passive management portfolio construction problem with fewer stocks. In this paper, we extend that work by using a hybrid genetic algorithm for the construction of index funds. In each of the iterative steps, the genetic algorithm selects the tracking portfolio stocks and a quadratic program is then solved for optimal weights for these stocks in order to minimize tracking error variance. Not only does the genetic algorithm efficiently search the solution space, but the quadratic program also guarantees local convergence for a given selection of stocks. This new method is tested empirically on the German stock market index DAX, and a comparison with the earlier approach is provided. The results show that the hybrid genetic algorithm finds the global minimum tracking error variance portfolio. Moreover, the out-of-sample performance of portfolios constructed using the hybrid genetic algorithm clearly dominates the performance of those constructed with the canonical genetic algorithm, at least for the more relevant portfolios of medium to large size. Extensions of this application are discussed.

JEL Classification: C15, C49, C61, G11.

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

1 Introduction

Passive investors strive to obtain the same returns 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.

However, 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. Eddelbüttel (1996) applies a canonical genetic algorithm to the passive management portfolio construction problem where the genetic algorithms selects both the stocks in the index tracking portfolio and their investment weights.

This paper extends Eddelbüttel (1996) by proposing hybrid approach: for each selection of stocks that is identified by the genetic algorithm, we solve the corresponding variance minimization problem with a quadratic programming routine. This approach combines the global search properties of the genetic algorithm with the local convergence of the quadratic programming solver. 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 and also describes the hybridization. The DAX index and data are described in section 4. An empirical illustration is presented in section 5. Section 6 concludes the paper.

2 The Optimization Problem

The optimization problem in this paper follows that of Eddelbüttel (1996). 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.

As in Eddelbüttel (1996), we extends the work by Roll (1992). Using the index vector $\boldsymbol{j}$ where $j_i \in \{0,1\}\ \forall\ i = 1,\ldots,N$, we can state the model of minimizing the tracking error variance using only up to k stocks as

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

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. The first constraint of a tracking error value of ρ is not imposed in this application but left for future research.

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 passive management 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 and Hybridization

This section provides a brief discussion of genetic algorithms in the context of function optimization, and describes how we hybridize the genetic algorithm in this application. Standard optimization methods, given *one* initial starting point, exploit problem-specific knowledge iteratively in order to improve the current solution. Genetic algorithms, on the other hand, are population-based and 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 function 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.

Eddelbüttel (1996) used binary strings of length $15k$ to represent portfolios of k shares in order to construct index tracking portfolios with a canonical genetic algorithm (CGA). The CGA determined both which shares were selected, and how much weight each selected share received. In this paper, the genetic algorithm is hybridized by combining it with a quadratic programming routine. The genetic algorithm is used as a global search routine to identify combinations of shares. That is, we use the genetic algorithm to find solutions to the integer programming problem of selecting stocks for the tracking portfolio. For each of these portfolios, the quadratic programming routine is then solved numerically using a modified version of the Powell (1983) solver to find weights such that the tracking error variance is minimized.[1] This hybrid approach combines the power of the genetic algorithm to efficiently search a vast parameter space with the local convergence properties of the quadratic program solver. Moreover, because the genetic algorithm is used only to identify shares, but not their

[1]Klaus Schittkowski of the University of Bayreuth kindly provided this routine.

weights as in Eddelbüttel (1996), we can use a simpler data representation. A binary vector of length $5k$ is used as we need, for each of the k shares, 5 bits to encode an integer value between 1 and 30 to indentify a selected share. Therefore, the dimension of the search space is only 2^{150} for a thirty share portfolio, and not 2^{450} as in Eddelbüttel (1996).

4 Data

For compararison with Eddelbüttel (1996), we also use one year of German stock market data covering the period from September 3, 1990, until August 30, 1991. Returns are expressed in percentages and 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 as $100(P_t - P_{t-1})/P_{t_1}$. 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 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. Of course, actual index funds face the problem of tracking changing investment weights, but as this paper is concerned with optimizational aspects of passive management, we prefer to track an index with fixed weights. The benefit is that we know the minimum tracking error variance for a portfolio of thirty shares: it is zero, and not some residual variance as in the case of changing index weights. So the 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 the methodology 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 Empirical Application

This section reports the results obtained by using the hybrid genetic algorithm outlined at the end of section 3 to construct DAX tracking portfolios. Formally, the problem is to find a solution to the minimization problem (3). We are not imposing a given tracking error ρ at this stage in order to obtain 'baseline' results; the question of minimizing tracking error variance under the constraint of posive tracking error, or for a specified tracking error, is left for future research. So the minimization problem (3) simplifies to

$$\begin{aligned}
\underset{\boldsymbol{p},\boldsymbol{j}}{\text{Minimise}} \quad & (\boldsymbol{p}-\boldsymbol{m})'\boldsymbol{V}(\boldsymbol{p}-\boldsymbol{m}) \\
\text{subject to} \quad & p_i \geq 0 \quad \text{if and only if} \quad j_i = 1 \\
& p_i = 0 \quad \text{if and only if} \quad j_i = 0 \\
& \boldsymbol{j}'\boldsymbol{i} \leq k \\
& \boldsymbol{p}'\boldsymbol{i} = 1
\end{aligned} \tag{4}$$

where the genetic algorithm selects the composition of the index vector $\boldsymbol{j}$ by using the quadratic programming solver to find a vector $\boldsymbol{p}$ such that the tracking error variance is minimized. As the first constraint imposes non-negative weights, short sales are excluded.

As genetic algorithms are a randomized search method, we analyse the performance of this hybrid genetic algorithm in a Monte Carlo experiment. Simulation allows us to quantify how this procedure performs over repeated trials, and furthermore allows us to assess the robustness of performance. As in Eddelbüttel (1996), we address this optimization problem for each portfolio size k between 4 and 30 shares. Table 2 reports descriptive statistics for the sample obtained from a Monte Carlo experiment of size $N = 250$, and provides a comparison to the results in Eddelbüttel (1996) where a canonical genetic algorithm (CGA) was used. Interestingly, for $k \leq 9$, the hybrid genetic algorithm (HGA) does not perform as well as the simpler approach. For $k = 10$ and $k = 11$, the means of the tracking error standard deviation, obtained from the Monte Carlo samples using the HGA, are equal to the means from the canonical genetic algorithm (CGA) used in the earlier paper. But for higher k, the HGA finds solutions with smaller tracking error standard deviation. And the more k increases, the more the HGA outperforms the CGA. For both Monte Carlo populations, the sample mean is only slightly higher than the minimum, but clearly lower than the maximum which shows that the sample distribution is non-symmetric. It is worth noting that the sample dispersion is much lower in the HGA population. Both the standard deviation of the sample and the range, that is the difference between the sample minimum and maximum, are lower for a given size k in the HGA-generated sample. For portfolio sizes greater or equal to 19, we see that the HGA consistently outperforms the CGA —- the sample maxima, drawn from 250 replications of the HGA, are lower than the sample means for the 100 replications of the CGA. For portfolios of size 24 and bigger, the worst result for the HGA is still better than the best result of the CGA.

While the descriptive statistics presented in table 2 shed some light on the empirical distribution of tracking error variance, we are unable to assess the convergence properties of the HGA from this table. Because of the discrete nature of integer programming, we see that the parameter space in which the genetic algorithm searches for vectors $\boldsymbol{j}$ is in fact finite. This in turn allows us to enumerate all possible vectors $\boldsymbol{j}_i$ where the index i ranges from 1 to $b_k \equiv \binom{30}{k}$. We use a non-recursive routine tailored after the algorithm in Gentleman (1975). Feeding each of these vectors into the quadratic programming solver then forms an exhaustive, albeit very expensive, way to search for the index tracking portfolio that *globally* minimizes the tracking error variance. Because of the combinatorial explosion, we did not undertake computations in order to find the globally best portfolio for every portfolio size k — in the most expensive case, $k = 15$, we would have had to evaluate $\binom{30}{15} = 155\,117\,520$ portfolios. As we can evaluate approximately 75 combinations per second on a Pentium 90, this would have taken almost twentyfour days. Nevertheless, table 3 shows the Monte Carlo means and minima, along with the global minima, for the values of k greater than or equal to 19 for which we searched the parameter space exhaustively. This allows us also to show the percentages of Monte Carlo runs in which the HGA obtained the global minimum. Note that in this paper, we have opted for a higher number of Monte Carlo replications, rather than a larger number of iterations used for each Monte Carlo replication which would without doubt have increased this convergence ratio. We see that the convergence percentages decrease with decreasing k — but this is hardly suprising as the parameter space increases when k decreases from its maximal value of 30 to $k = 15$ which corresponds to the biggest possible parameter space. For both $k = 19$ and $k = 20$, this trend is broken and the convergence percentage is higher than it is for $k = 21$ and $k = 22$. This might be due to the fact that we optimized the parameters used for the genetic algorithm search in a preliminary experiment where we held k fixed at 20. While we have to acknowledge that all the sample means are statistically significantly higher than the global minima found for that size k, it is worth noting that the absolute difference between them is always less than a basispoint (which is commonly defined as one hundreds of a percentage point).

An important verification of the validity of these results can be obtained by comparing 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. It is of course important to see how these portfolios perform on different, unused samples of data. From the remaining four columns of table 3, we can assess the usefulness of the genetic algorithm by comparing the tracking error standard deviation from the in-sample period with the out-of-sample values. 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. As we would expect, the sample mean of tracking error standard deviation decreases with increases portfolio size for all four out-of-sample periods, as it does in the in-sample period. But perhaps most important is the comparison with the CGA results: for all k greater or equal to 14, the sample mean of tracking error standard deviation is lower for the HGA than for the CGA.

We also computed test statistics for the null hypothesis of equal sample means, these are shown in table 4. We see that while the HGA is always outperforming the CGA in-sample, the results are different out of sample. In the first, third and fourth out-of-sample period, small portfolios constructed using the HGA outperform the CGA. However, the sample mean of tracking error standard deviation from the HGA-constructed portfolios in the second out-of-sample period, and from all portfolios greater or equal to $k = 21$ in the fourth (and longest) out-of-sample period, are in fact significantly lower than those constructed using the simpler CGA. Some caution might be advised for interpreting these test statistics as both sample populations are clearly non-normal. But as noted by Tassi (1992), if the sum of both sample sizes is bigger than 100, the distributions of the samples does not matter. This point clearly applies in our case where we compare two samples of sizes 100 and 250, respectively.

Finally, 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 almost always positive. On the other hand, 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, in our case is $\sqrt{250} \approx 15.81$, equals the usual t-statistic for the significance of a sample mean, we can verify that the tracking errors for the smaller portfolios are statistically significant at conventional levels. Focussing again on the fourth and largest out-of-sample period, and portolios of size greater or equal to 21, we see that the tracking error is not statistically significant in these cases.

Last but not least, figure 3 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 portfolios of size $k = 20$ over the data period calculated using the canonical genetic algorithm and the hybrid genetic algorithm. We observe that there is virtually no difference between the index and the tracking portfolios during the in-sample period, and only a small difference 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 a hybrid genetic algorithm 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 hybrid genetic algorithm can be applied successfully to this problem. The analysis shows that minimizing the variance of tracking error does not come at the expense of statistically significant tracking error, at least for the more relevant portfolio sizes with at least two thirds of the index stocks included. More importantly, we have seen that the hybrid genetic algorithm produces results, that at these portfolio sizes, are highly significantly improved over the canonical genetic algorithm used in Eddelbüttel (1996). Using exhaustive search of the finite parameter space, we showed that the hybrid genetic algorithm converges to the global minimum tracking error variance portfolio for the portfolio sizes. Examining the impact of imposing a restriction of a tracking error greater or equal to a given threshold ρ is left for future research.

Acknowledgements

Financial support from the Ministère de l'Éducation, Paris, and the Apel Fellowship, Braunschweig, is gratefully acknowledged. Deutsche Bank Research made the data available. Klaus Schittkowski kindly provided his modified version of the Powell (1983) routine for solving quadratic programming problems subject to parameter bounds and restrictions. Remaining errors are the sole responsibility of the author.

References

Eddelbüttel, Dirk (1996). A genetic algorithm for passive management: Creating index funds with fewer stocks. Proceedings of the Third International Conference *Forecasting Financial Markets: New Advances for Exchange Rates, Interest Rates and Asset Management* sponsored by Chemical Bank and Imperial College, London.

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

Gentleman, Jane F. (1975). Algorithm AS 88: Generation of all $_NC_R$ combinations by simulating nested Fortran `DO` loops. *Applied Statistics 24(3)*, 374–376.

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

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.

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

Powell, Michael J. D. (1983). ZQPCVX: A Fortran subroutine for convex programming. Report DAMTP/1983/NA17, University of Cambridge, Cambridge, UK.

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.

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.

Tassi, Philippe (1992). *Méthodes Statistiques* (Second ed.). Collection Économie et Statistiques Avancées. Paris: Economica.

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

	Canonical GA				Hybrid GA			
k	mean	stdev.	min.	max.	mean	stdev.	min.	max.
4	0.267	0.005	0.264	0.280	0.270	0.008	0.264	0.316
5	0.219	0.008	0.214	0.238	0.227	0.009	0.214	0.257
6	0.193	0.005	0.187	0.205	0.199	0.008	0.187	0.242
7	0.169	0.006	0.161	0.180	0.175	0.006	0.161	0.194
8	0.153	0.005	0.147	0.164	0.157	0.006	0.148	0.182
9	0.140	0.004	0.136	0.149	0.141	0.005	0.136	0.155
10	0.126	0.004	0.120	0.139	0.126	0.005	0.120	0.146
11	0.114	0.004	0.109	0.129	0.114	0.005	0.109	0.134
12	0.104	0.005	0.099	0.124	0.103	0.004	0.099	0.121
13	0.095	0.005	0.089	0.110	0.093	0.004	0.089	0.107
14	0.088	0.005	0.082	0.103	0.085	0.004	0.082	0.100
15	0.082	0.005	0.074	0.093	0.077	0.003	0.074	0.092
16	0.077	0.005	0.068	0.092	0.071	0.003	0.068	0.092
17	0.072	0.005	0.063	0.086	0.065	0.003	0.062	0.078
18	0.069	0.005	0.059	0.084	0.060	0.003	0.058	0.072
19	0.065	0.005	0.054	0.079	0.056	0.002	0.054	0.063
20	0.061	0.005	0.050	0.080	0.051	0.002	0.049	0.058
21	0.058	0.005	0.046	0.070	0.046	0.001	0.045	0.053
22	0.056	0.004	0.046	0.068	0.042	0.002	0.040	0.051
23	0.053	0.005	0.041	0.071	0.037	0.001	0.036	0.045
24	0.050	0.004	0.039	0.060	0.033	0.001	0.032	0.037
25	0.048	0.004	0.039	0.060	0.028	0.001	0.028	0.037
26	0.045	0.004	0.036	0.056	0.024	0.001	0.023	0.032
27	0.042	0.004	0.030	0.051	0.018	0.001	0.018	0.023
28	0.040	0.005	0.031	0.052	0.013	0.002	0.012	0.022
29	0.038	0.005	0.021	0.048	0.007	0.002	0.006	0.021
30	0.035	0.004	0.027	0.046	0.001	0.003	0.000	0.015

Note: The columns report the following values: mean, standard deviation, minimum and maximum of tracking error standard deviation computed from two Monte Carlo populations. For each value of k, we conducted 100 Monte Carlo experiments with 50 000 iterations each for the canonical genetic algorithm. For the hybrid genetic algorithm, 250 Monte Carlo experiments with 5 000 iterations each were conducted for each value of k, and for each of the 5 000 elements in each of the 250 Monte Carlo replications, the corresponding quadratic program was solved numerically.

Table 3: In-sample and out-of-sample comparisons

	In-sample				Out-of-sample Means			
k	Mean	Min.	Optim.	Cvg.	p1	p2	p3	p4
4	0.2695	0.2644			0.3547	0.1860	0.3016	0.2918
5	0.2270	0.2136			0.3025	0.1727	0.2758	0.2583
6	0.1990	0.1870			0.2662	0.1589	0.2509	0.2329
7	0.1754	0.1609			0.2450	0.1462	0.2152	0.2082
8	0.1573	0.1475			0.2264	0.1396	0.1961	0.1933
9	0.1410	0.1356			0.2109	0.1246	0.1741	0.1750
10	0.1260	0.1198			0.1770	0.1135	0.1540	0.1516
11	0.1143	0.1088			0.1609	0.1049	0.1385	0.1375
12	0.1029	0.0994			0.1440	0.0961	0.1240	0.1232
13	0.0935	0.0886			0.1326	0.0839	0.1178	0.1132
14	0.0854	0.0815			0.1216	0.0768	0.1076	0.1036
15	0.0775	0.0739			0.1114	0.0678	0.0921	0.0919
16	0.0712	0.0682			0.1059	0.0623	0.0841	0.0857
17	0.0652	0.0619			0.0984	0.0568	0.0791	0.0797
18	0.0602	0.0581			0.0828	0.0557	0.0699	0.0704
19	0.0556	0.0536	0.0536	19.2%	0.0772	0.0525	0.0637	0.0655
20	0.0506	0.0489	0.0489	17.6%	0.0636	0.0465	0.0557	0.0561
21	0.0463	0.0454	0.0454	10.8%	0.0548	0.0439	0.0480	0.0496
22	0.0418	0.0400	0.0400	16.0%	0.0505	0.0407	0.0436	0.0457
23	0.0374	0.0362	0.0362	22.8%	0.0439	0.0351	0.0380	0.0396
24	0.0332	0.0323	0.0323	26.4%	0.0353	0.0304	0.0336	0.0335
25	0.0285	0.0276	0.0276	36.4%	0.0292	0.0254	0.0286	0.0280
26	0.0235	0.0225	0.0225	37.2%	0.0237	0.0194	0.0238	0.0226
27	0.0182	0.0175	0.0175	47.6%	0.0171	0.0165	0.0206	0.0183
28	0.0131	0.0121	0.0121	60.4%	0.0128	0.0109	0.0161	0.0136
29	0.0070	0.0059	0.0059	80.8%	0.0063	0.0047	0.0084	0.0067
30	0.0012	0.0000	0.0000	81.2%	0.0011	0.0008	0.0015	0.0012

Note: The columns contain the sample means and minima of in-sample tracking error standard deviation as in table 2), the global minimal tracking error standard deviations found by exhaustive search of all corresponding quadratic programming problems, the percentage of Monte Carlo replications that obtained this global minimum, and the sample means of tracking error standard deviation for the four out-of-sample periods. The quadratic programs were solved using a modified version of Powell (1983).

Table 4: Testing for equal means between CGA and HGA results

		Out-of-sample			
k	In-sample	p1	p2	p3	p4
4	27.981	-43.266	15.666	-16.876	-37.175
5	17.382	-43.918	21.325	-10.150	-44.355
6	15.469	-39.208	11.523	-18.201	-62.150
7	18.754	-35.427	9.355	-14.163	-53.922
8	20.463	-36.205	8.957	-8.851	-55.332
9	11.943	-34.064	9.726	-7.889	-63.929
10	11.847	-35.630	9.166	-7.997	-52.428
11	10.206	-29.408	8.825	-8.122	-36.005
12	11.375	-29.367	9.269	-7.018	-31.529
13	12.526	-34.183	10.181	-5.534	-32.701
14	15.326	-35.366	9.740	-7.482	-31.086
15	15.948	-34.220	10.643	-6.248	-27.262
16	14.549	-33.590	10.208	-1.910	-19.221
17	15.899	-30.221	13.000	0.072	-19.202
18	14.717	-25.884	12.841	1.215	-14.592
19	11.163	-21.295	10.290	1.854	-7.298
20	13.022	-17.514	10.761	4.384	-5.507
21	11.884	-11.418	11.925	6.825	2.607
22	11.786	-5.743	13.626	11.209	9.570
23	12.583	-3.878	12.761	12.412	9.413
24	12.210	0.965	13.276	14.109	18.915
25	12.827	10.074	18.818	18.987	26.079
26	12.815	14.624	16.858	17.694	31.952
27	14.261	23.457	21.558	21.110	34.164
28	16.533	30.239	23.670	23.093	37.145
29	20.425	33.895	28.672	28.012	43.907
30	26.258	36.731	32.658	34.177	49.454

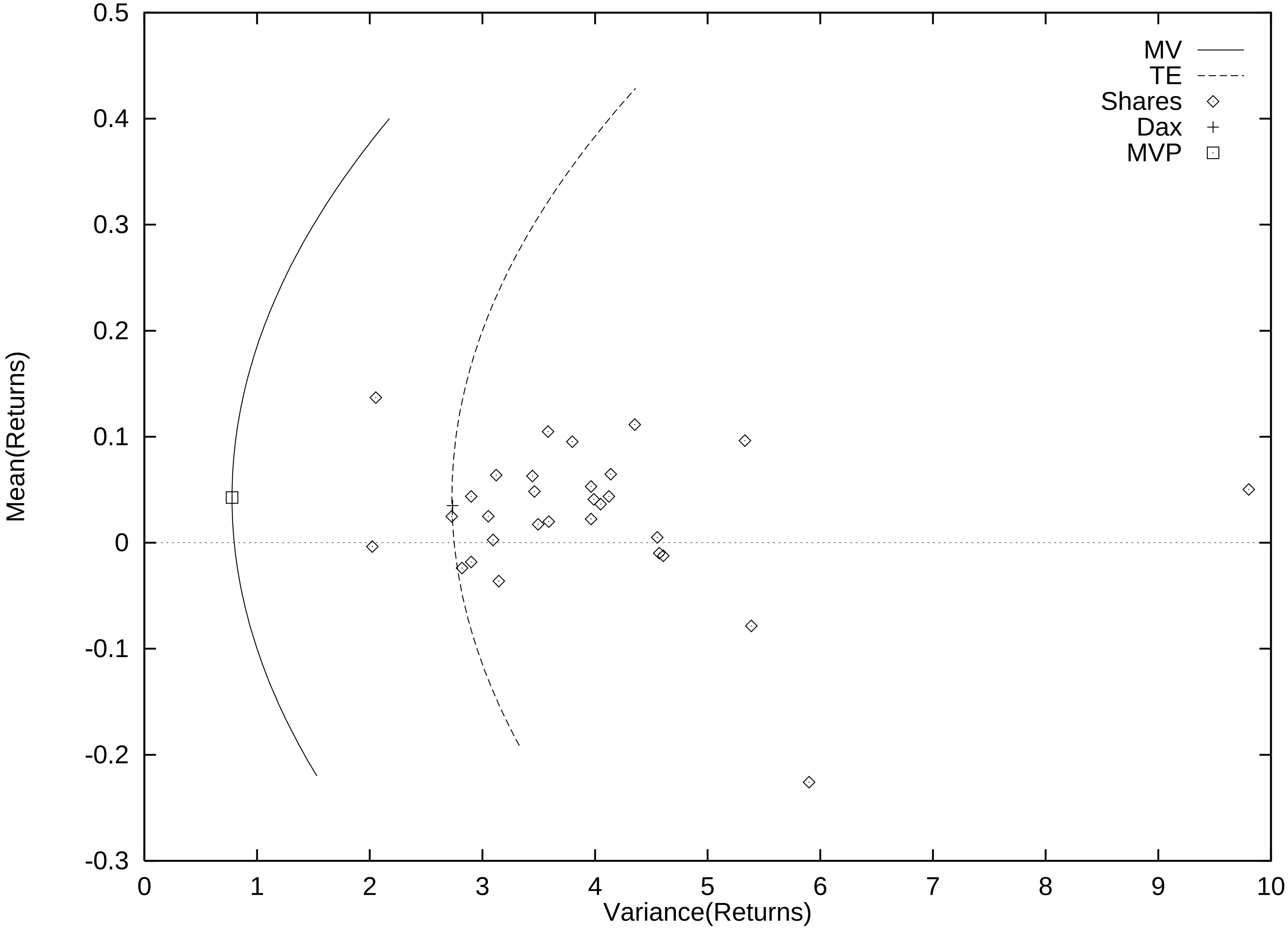
Note: This table shows the t-statistics for the null hypothesis of that the sample means of the tracking error standard deviations are equal, for a given portfolio size k, between the canonical and the hybrid genetic algorithm. The columns correspond to the in-sample period and the four out-of-sample periods. The statistics are calculated using the standard formula $(m_{\text{CGA}} - m_{\text{HGA}}) \cdot (n_{\text{CGA}}^{-1} s_{\text{CGA}}^2 + n_{\text{HGA}}^{-1} s_{\text{HGA}}^2)^{-0.5}$ where m, s and n represent sample mean, standard deviation and sample size and which is asymptotically distributed as $N(0,1)$.

Table 5: Sharpe Ratio and Tracking Error

	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	0.019	0.000	-0.002	0.067	0.401	-0.016	-0.054	0.019	0.063
5	0.005	0.021	0.002	0.005	0.040	0.251	-0.026	-0.095	0.006	0.023
6	0.005	0.026	-0.003	-0.012	0.019	0.139	-0.028	-0.116	-0.003	-0.016
7	0.005	0.029	-0.006	-0.025	0.014	0.114	-0.034	-0.163	-0.008	-0.037
8	0.005	0.032	-0.007	-0.031	0.009	0.071	-0.033	-0.171	-0.010	-0.050
9	0.003	0.023	-0.003	-0.016	0.002	0.017	-0.033	-0.199	-0.011	-0.064
10	0.001	0.011	-0.004	-0.024	-0.003	-0.024	-0.033	-0.217	-0.013	-0.087
11	0.000	0.003	-0.009	-0.059	-0.007	-0.067	-0.028	-0.201	-0.014	-0.107
12	0.000	0.004	-0.008	-0.059	-0.011	-0.112	-0.024	-0.196	-0.014	-0.118
13	0.001	0.016	-0.010	-0.080	-0.011	-0.128	-0.021	-0.178	-0.014	-0.124
14	0.001	0.018	-0.010	-0.089	-0.011	-0.149	-0.018	-0.167	-0.013	-0.129
15	0.002	0.023	-0.006	-0.056	-0.010	-0.149	-0.013	-0.142	-0.010	-0.107
16	0.002	0.025	-0.006	-0.055	-0.010	-0.158	-0.011	-0.142	-0.009	-0.108
17	0.002	0.027	-0.005	-0.057	-0.009	-0.155	-0.009	-0.120	-0.007	-0.099
18	0.001	0.024	-0.004	-0.054	-0.007	-0.126	-0.008	-0.122	-0.006	-0.095
19	0.002	0.031	-0.004	-0.068	-0.003	-0.070	-0.007	-0.132	-0.005	-0.084
20	0.002	0.032	-0.007	-0.143	-0.003	-0.074	-0.007	-0.133	-0.006	-0.109
21	0.002	0.041	-0.008	-0.180	-0.003	-0.056	-0.006	-0.119	-0.005	-0.109
22	0.001	0.035	-0.007	-0.157	-0.001	-0.024	-0.005	-0.115	-0.004	-0.088
23	0.001	0.030	-0.008	-0.189	0.001	0.020	-0.001	-0.023	-0.002	-0.062
24	0.001	0.026	-0.006	-0.174	-0.001	-0.026	-0.002	-0.037	-0.003	-0.078
25	0.000	0.010	-0.005	-0.156	-0.000	-0.018	0.001	0.039	-0.001	-0.042
26	-0.001	-0.040	-0.003	-0.114	0.000	0.005	0.002	0.101	-0.000	0.001
27	-0.001	-0.076	-0.002	-0.103	0.002	0.109	0.002	0.103	0.001	0.044
28	-0.001	-0.049	-0.001	-0.100	0.002	0.157	0.001	0.060	0.000	0.033
29	-0.000	-0.054	-0.001	-0.227	0.001	0.297	0.000	-0.010	0.000	0.001
30	-0.000	-0.028	-0.000	-0.047	0.000	-0.072	-0.000	0.047	0.000	-0.017

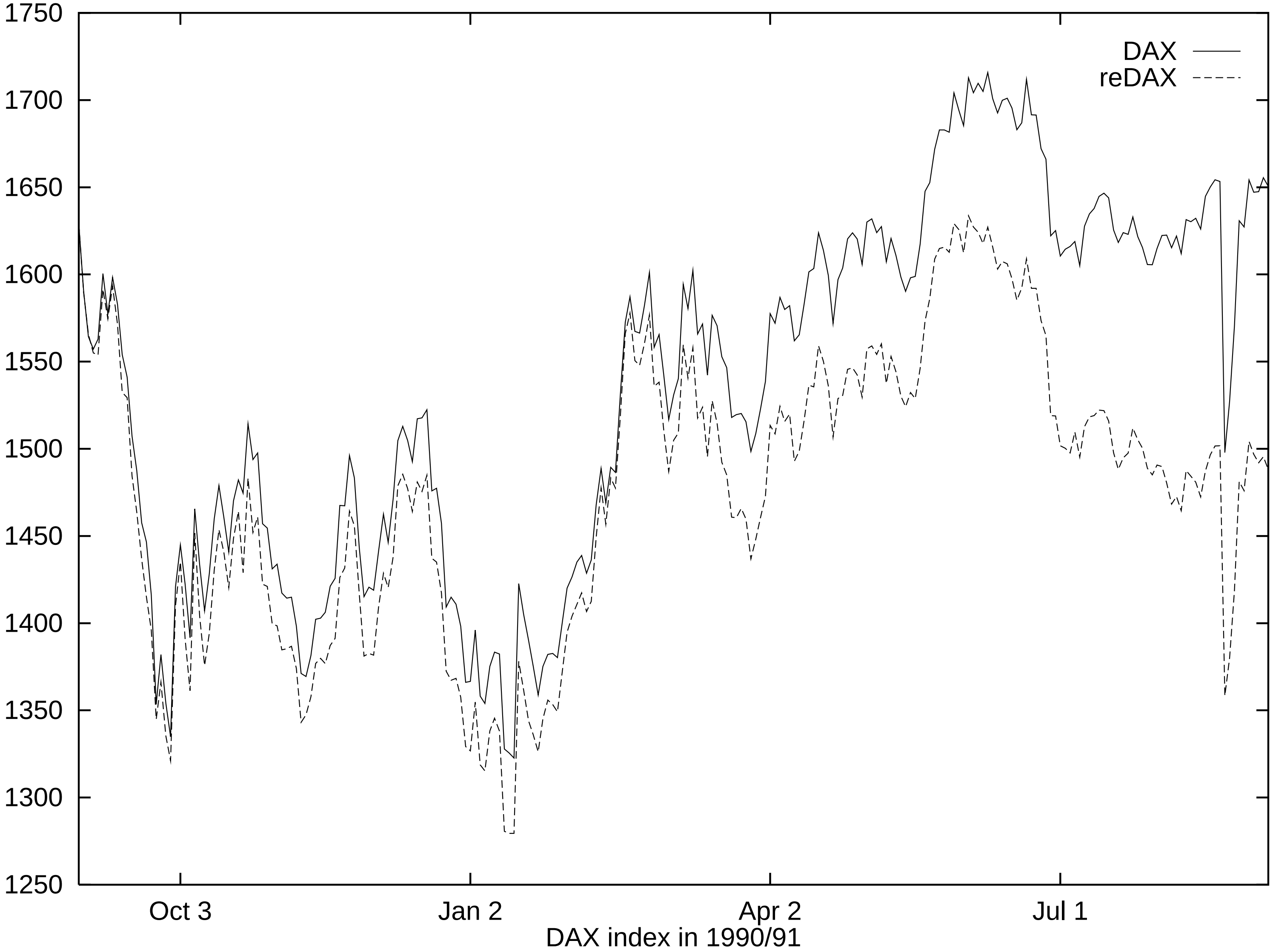
Note: The Sharpe ratio (Sharpe 1994) is defined as the mean of return differences over a benchmark investment divided by the sample standard deviation of the return differences. In this application, it is computed as the sample mean of tracking error over the sample standard deviation of tracking error. Multiplying the Sharpe Ratio by the square root of the sample size, in this case 15.81, yields the usual t-statistic for the significance of a sample mean. As we here have 249 degrees of freedom, this can clearly be compared against the critical values of the $N(0,1)$ distribution.

Figure 1: Mean/Variance and Tracking Error Efficient Portfolios



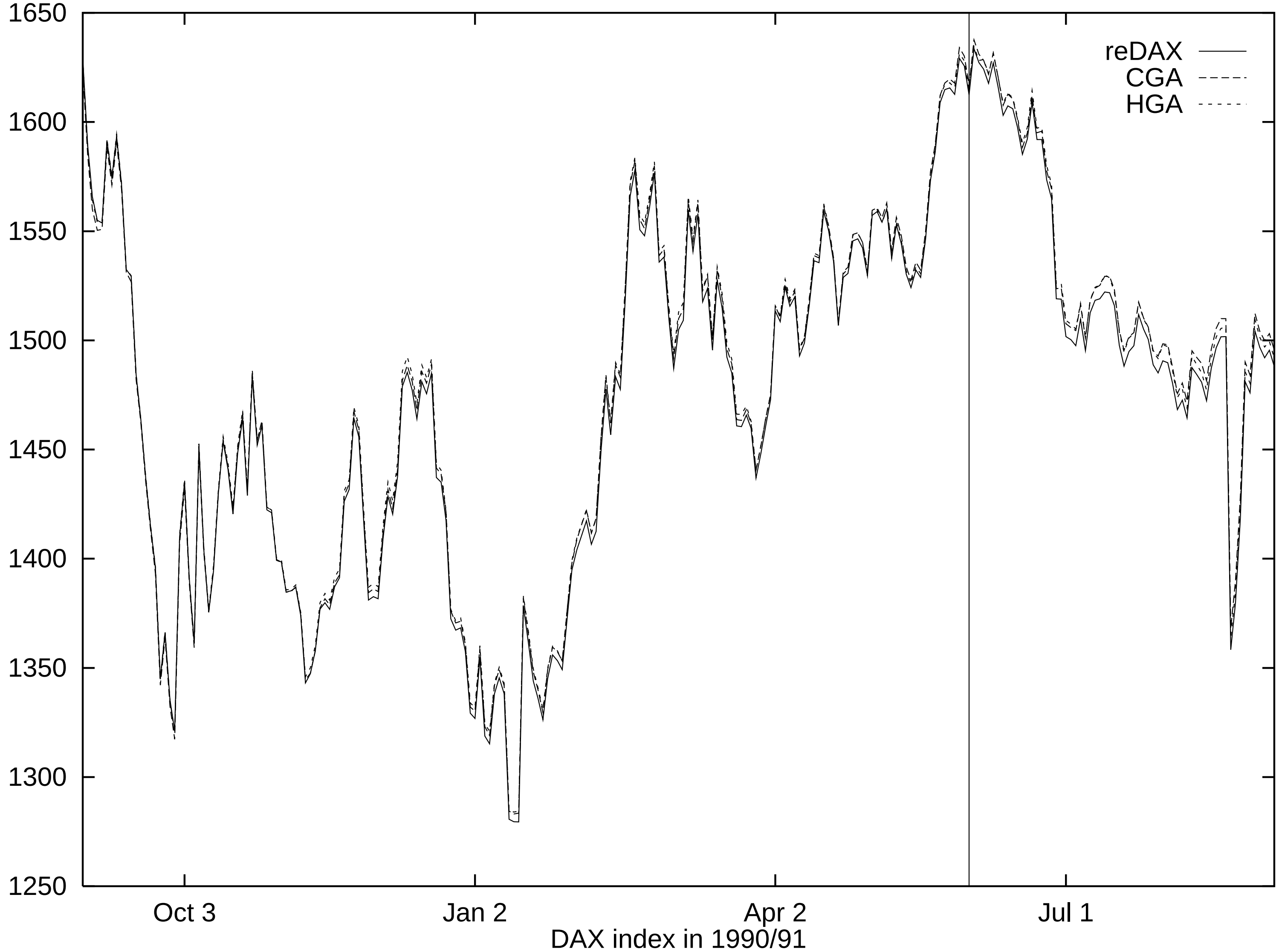
Note: The efficient frontier and the tracking error frontier were calculated following the procedures in Huang and Litzenberger (1988) and Roll (1992), respectively, with the data described in section 4. As these data comprise only a year, and a year in which the market index moved mostly sideways, the 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.

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 definition of the reDAX.

Figure 3: Index Tracking Portfolio and reDAX



Note: The series 'reDAX' shows the recalculated DAX as described in section 4. The series 'CGA' and 'HGA' show the best 20 share tracking portfolio constructed by the canonical and hybrid genetic algorithms, respectively. 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.