Multi-Objective Evolutionary Methods for Time-Changing Portfolio Optimization Problems

by

Iason Hatzakis

Master of Science in Ocean Engineering
Massachusetts Institute of Technology, 2004

Diploma in Naval Architecture and Marine Engineering
National Technical University of Athens, 2000

SUBMITTED TO THE DEPARTMENT OF MECHANICAL ENGINEERING
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN OCEAN SYSTEMS MANAGEMENT

AT THE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

SEPTEMBER 2007

Author:..
Department of Mechanical Engineering
July 6, 2007

Certified by:..
Henry S. Marcus
Professor of Marine Systems
Thesis Supervisor

Accepted by:..
Lallit Anand
Professor of Mechanical Engineering
Chairman, Departmental Committee on Graduate Studies

Multi-Objective Evolutionary Methods for Time-Changing Portfolio Optimization Problems

by

Iason Hatzakis

Submitted to the Department of Mechanical Engineering on July 6, 2007 in Partial Fulfillment of the Requirements for the Degree of Master of Science in Ocean Systems Management

Abstract

This thesis is focused on the discovery of efficient asset allocations with the use of evolutionary algorithms. The portfolio optimization problem is a multi-objective optimization problem for the conflicting criteria of risk and expected return. Furthermore the nonstationary nature of the market makes it a time-changing problem in which the optimal solution is likely to change as time advances. Hence the portfolio optimization problem naturally lends itself to an exploration with multi-objective evolutionary algorithms for time-changing environments.

Two different risk objectives are treated in this work: the established measure of standard deviation, and the Value-at-Risk. While standard deviation is convex as an objective function, historical Value-at-Risk is non-convex and often discontinuous, making it difficult to approach with most conventional optimization techniques. The value of evolutionary algorithms is demonstrated in this case by their ability to handle the Value-at-Risk objective, since they do not have any convexity or differentiability requirements.

The D-QMOO time-changing evolutionary algorithm is applied to the portfolio optimization problem. Part of the philosophy behind D-QMOO is the exploitation of predictability in the optimal solution's motion. This problem however is characterized by minimal or non-existent predictability, since asset prices are hard to forecast. This encourages the development of new time-changing optimization heuristics for the efficient solution of this problem.

Both the static and time-changing forms of the problem are treated and characteristic results are presented. The methodologies proposed are verified through comparison with established methods and through the performance of the produced portfolios as compared to the overall market. In general, this work demonstrates the potential for the use of evolutionary algorithms in time-changing portfolio optimization as a tool for portfolio managers and financial engineers.

Thesis Supervisor: Henry S. Marcus
Title: Professor of Marine Systems

Acknowledgements

Many thanks are due to my supervisor, Hank Marcus, who was always there to provide his insightful advice on matters both in and outside research and school.

I owe a lot to Frank Schlottmann, whose work on credit risk portfolio optimization with evolutionary algorithms largely inspired this thesis, and who collaborated with me in the course of this work.

Many thanks are owed to my doctoral thesis supervisor, David Wallace, whose role was instrumental in the development of the D-QMOO algorithm and who has been a unique inspiration to me over my years in MIT.

I would like to thank Geoff Leyland, who created the base version of D-QMOO (QMOO) during his doctoral work and who gave a lot of his time to help and guide me during the initial stages of D-QMOO's development.

I owe a lot to Elina, for standing next to me from so far and for being my muse.

I also owe a lot to my mother and to the rest of my family and friends, both in Cambridge and in Athens, without whom I would not have been able to finish this work.

Contents

List of Figures

List of Tables

Abbreviations and Symbols

Abbreviations

AR	Autoregressive Model
AP	Anticipatory Population
CTI	Closest-to-Ideal
D-QMOO	Dynamic Queuing Multi-Objective Optimizer
DJIA	Dow Jones Industrial Average
EA	Evolutionary Algorithm
EP	Evolutionary Programming
EPS	Earnings Per Share
ES	Evolution Strategy
FPS	Feed-forward Prediction Strategy
GA	Genetic Algorithm
POF	Pareto Optimal Front
POS	Pareto Optimal Set
QMOO	Queuing Multi-Objective Optimizer
SP600	Standard and Poor's 600 Small Cap Index
SQP	Sequential Quadratic Programming
VAR	Vector Autoregressive Model
VaR	Value-at-Risk

Symbols

$\mathbf{x}$	design vector
n	design vector dimension
$\mathbf{f}$	objective vector
m	objective vector dimension
g	inequality constraint function

q	number of inequality constraints
h	equality constraint function
l	total number of constraints
l_i	lower bound for design variable i
u_i	upper bound for design variable i
t	time
$\mathbf{x}^*_t$	optimal solution at time t
X	search space
q_α	α-quantile
μ	mean
σ	standard deviation
$\mathbf{Q}$	covariance matrix
$r_i(k)$	return of asset i at time k
$R(t)$	asset returns history up to time t

1 Introduction

One of the most crucial decisions that investors and portfolio managers need to make is how to allocate the available capital among different investment opportunities. A successful portfolio maximizes the investor's profit, which forms the ultimate objective of this decision problem. However the resulting profit from a specific portfolio only becomes known after the portfolio has been selected and deployed in the market. Hence, a conventional optimization approach where the objective value is known for each candidate design cannot be followed in this case. As a result, portfolio optimization takes the form of a two objective problem. The first objective is the portfolio's *expected return*, which we seek to maximize. The second objective is the portfolio's *risk*, which we seek to minimize.

The portfolio optimization problem is far from simple to solve. The first source of its complexity is the fact that the pool of available assets is large – it may be in the order of hundreds or even thousands of different bonds, stocks and other kinds of investments. Hence simply processing the data that leads to the estimation of the objectives (risk and expected return) requires some form of computational tool. Furthermore, the large number of available assets also enlarges the sample space of the problem – there is a multitude of different alternative solutions to the portfolio optimization problem, each one with its own unique asset allocation. At the same time, the measures used for the estimation of risk and return may be complex, non-convex objective functions. Finally, the market data used for the calculation of risk and return constantly changes in time, making the design of financial portfolios a time-changing problem.

This complexity and nonstationarity make the portfolio optimization problem a natural application for multi-objective time-changing optimization. This work is focused on the application of multi-objective time-changing evolutionary algorithms to this problem.

1.1 Outline

In chapter 2, a discussion on the portfolio optimization problem is given. The problem is defined and the various measures of risk and return used in this work are described. Two different risk measures are used, the sample standard deviation of returns and the Value-at-Risk. The sample average of returns is used as a return measure. The general statement for the optimization problem solved in this thesis is given, followed by a description of the data sets used in the numerical experiments of the following chapters.

In chapter 3 the computational tool used to solve the optimization problem is described. A general introduction to Evolutionary Algorithms as heuristic optimization methods is given first, followed by a description of the Dynamic Queuing Multi-Objective Optimizer (D-QMOO) which is the algorithm used in this work. The chapter concludes with some useful details on the solution process of the portfolio optimization problem with D-QMOO, such as the performance measures used.

In chapter 4 the static version of the problem is treated. Here, the fact that market conditions change in time is momentarily ignored and the problem of discovering a set of efficient portfolios corresponding to a fixed instance of the market is explored. Since, in this work, the time-changing form of the problem is seen as a sequence of discrete instances, the static problem constitutes a building block for the nonstationary problem. Both resulting optimization problems, (with standard deviation or Value-at-Risk as risk measures) are solved and the algorithm's behavior is discussed.

In chapter 5 the time-changing portfolio optimization problem is solved. The D-QMOO algorithm is applied in its nonstationary form, as developed by this author during his doctoral research. One of the most interesting attributes of the portfolio optimization problem is the lack of predictability in the way the optimal solution moves in the decision space. This lack of predictability inspired the development of new heuristics for the efficient solution of the problem which are described in this chapter.

The thesis closes with conclusions and recommendations for future work in chapter 6.

2 Portfolio Optimization

A portfolio can be defined as a *collection of investments*. Investors have at their disposal an amount of capital and a range of assets on which to invest. These assets may include bonds, stocks, derivatives or real estate; in the context of shipping, they may for example include vessels or freight-forward agreements. The investor's goal is to create the maximum amount of return on their capital with the minimum amount of risk. The problem that arises is how to allocate the capital among the different assets in order to accomplish this goal. Solving this *portfolio optimization problem* provides the investor with an optimal decision for the allocation of her assets.

The two underlying objectives in this problem are the *expected return* and the *risk* that characterize investments and portfolios. Formulating and solving the portfolio optimization problem is far from simple given that in a global market there are hundreds or thousands of assets to select from, and that there exist numerous options for the definition and estimation of risk and expected return. The portfolio design process consists of two stages:

- A quantitative definition for the measures of risk and expected return, which expresses the investors' beliefs about the future performance of the various assets.
- Given the quantitative measures for the two objectives, an optimization process producing a set of portfolios that have a maximum expected return with a minimum amount of risk.

A seminal approach to the portfolio optimization problem was *mean-variance analysis* by Harry Markowitz (Markowitz 1952, Markowitz 2000), where the previous staging of the portfolio selection process is maintained.

In terms of the first stage, a major categorization of risk and expected return measures is between statistical methods and methods which also rely on the investor's personal views about the future performance of assets. According to the first class of methods, the past history of asset returns provides an estimate of their future performance. This history is usually cast into the form of a time series, and its statistics are calculated in order to provide risk and return measures. Markowitz's initial approach (Markowitz 1952) which has been followed in various forms by several researchers (Hirschberger, Qi & Steuer 2004) is such an example. However, a sample of past returns does not necessarily allow us to predict the future. For example, economic fundamentals might have changed or external developments affecting asset performance might be expected. For this reason researchers have developed methods which also take in account the investor's views regarding future performance. The approach by Black and Litterman (Black, Litterman 1992, Litterman 2003), where the equilibrium returns for securities provided by the Capital Asset Pricing Model are used as starting points to which the investor's views are added, is such an example. Other cases of such hybrid methods can also be found in literature (see for example Ehrgott, Klamroth & Schwelm 2004).

The majority of the present work is concerned with the second part of the process – the creation of methods that allow the optimal selection of portfolios given a set of risk and expected return measures. However, we briefly dwell on the first stage when the use of two different risk measures, the *variance* and the *value-at-risk*, is discussed. The approach we follow regarding the quantitative measures of risk and return is statistical; the time history of asset returns is the only source of information that determines risk and expected return for each asset. However, the techniques described in this thesis aim to provide the investor with a helpful computational tool that can potentially be extended to include investor views as well.

Next we discuss the risk and return measures as applied in this work, and then provide the portfolio optimization problem statement.

2.2 Statistical Measures for Risk and Return.

The statistics of asset returns are the sole determinants of the risk and return measures. These statistics are calculated using past samples of data, as we can see in the sketch of Figure 1. In this paragraph the exact definitions for each measure are given, as implemented in the numerical experiments of chapters 4 and 5.

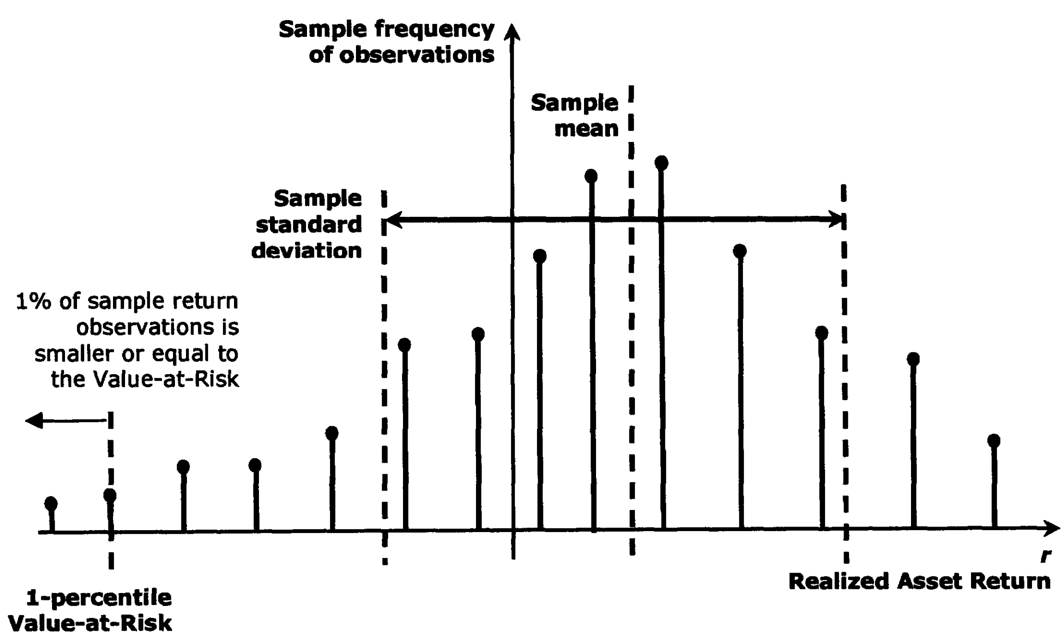


Figure 1. Sample statistics are used for risk and return estimation.

2.2.1 Expected Return.

The expected return of a portfolio is derived from the weighted expected returns of the assets it includes. The measure of expected return for an asset is the un-weighted arithmetic average of a sample of past returns.

Let us assume there is a total of n assets on which to invest. The investor makes a decision about the allocation of her wealth at the t point in time. At time $t > 1$, the history of asset returns up to t is:

$$R(t) = (r_i(k)),\ i = 1,\dots,n,\ k = 1,\dots,t-1\ . \tag{2.1}$$

A portfolio at time t describes an allocation of the investor's capital to the investment opportunities. It is denoted by a real vector:

$$\mathbf{x}(t) = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix},\ \text{where } \forall i \in \{1,\dots,n\}: x_i \geq 0 \tag{2.2}$$

where x_i is the fraction of the total capital allocated to asset i.

Then, the expected return of a portfolio $\mathbf{x}$ is:

$$return(x, R(t), t) = E[R(t)x(t)] = \frac{1}{t-1}\sum_{k=1}^{t-1}\sum_{i=1}^{n} r_i(k)x_i(t) \tag{2.3}$$

In practice, the expected return is calculated using a rolling one-year historical sample (250 trading days) behind the current time t, as shown in Figure 2. In this work, the timestep t denotes one trading day. At the end of each day, the closing market data is gathered. Then a portfolio is designed, and deployed on the market the next day.

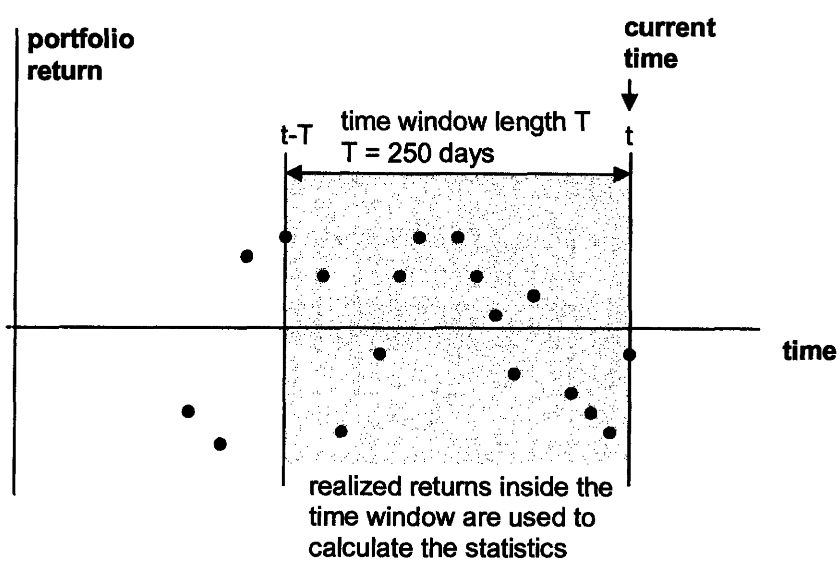


Figure 2. A rolling time window of 250 trading days behind the current timestep is used as a sample for the asset performance estimation.

2.2.2 Standard Deviation of Return (first risk measure).

The first risk measure used in this work is the standard deviation of the portfolio return. Following the conventions in the previous paragraph, this can be defined:

$$\begin{aligned} risk(x,R(t),t) &= \sqrt{\sigma_{\mathbf{x}}^2} \\ &= \sqrt{\mathbf{x}^T \mathbf{Q} \mathbf{x}} \\ &= \sqrt{\frac{1}{t-1}\sum_{k=1}^{t-1}\left(\sum_{i=1}^{n} r_i(k)x_i(t)\right)^2 - \left(\frac{1}{t-1}\sum_{k=1}^{t-1}\sum_{i=1}^{n} r_i(k)x_i(t)\right)^2} \end{aligned} \tag{2.4}$$

where σ_x is the portfolio standard deviation and $\mathbf{Q}$ is the sample covariance matrix of asset returns. Portfolio standard deviation and variance are well established risk measures[1]. Markowitz treated the portfolio optimization problem using variance as the risk measure. Variance, as an objective, has an important computational advantage since it is a quadratic function. This provides a continuous, smooth and convex optimization problem for the solution of which many well-established gradient algorithms exist today, such as Sequential Quadratic Programming (SQP - see for example Papalambros, Wilde 2000).

An objection against using standard deviation as a risk measure lies with its ability to model the actual risk of losing money. Standard deviation expresses an asset's volatility in both directions, hence it only implicitly relates to loss. However from the investor's point of view, risk quantifies the probability of a small or negative return – the probability of losing money.

2.2.3 Value-at-Risk (second risk measure)

The second risk measure used is the Value-at-Risk (VaR). The Value-at-Risk is defined as a quantile of the distribution of past returns:

$$risk(x,R(t),t) = VaR\big((t)x(r)\big) = q_\alpha\big(R(t)x(t)\big),\ \ \alpha \in (0,1) \tag{2.5}$$

where q_a is the α-quantile of the past returns' distribution.

As a risk measure the VaR expresses the maximum loss a portfolio will suffer with a probability of (1-α), as shown on the sketch of Figure 3. This makes the Value-at-Risk a potentially more direct risk measure than standard deviation since it directly relates to portfolio loss.

There exist several methods for the estimation of VaR (Beder 1995, Hendricks 1996, Stambaugh 1996). VaR can be estimated either:

- By assuming a parametric probability distribution for the portfolio return and calculating the VaR analytically, with the estimated portfolio return and variance as parameters. For example in the case of a normal distribution, the minimum VaR problem has the same solution as the minimum variance problem under these assumptions, since the Value-at-Risk for the 1-percentile is:

$$VaR = \mu - 2.32\sigma \tag{2.6}$$

  where μ is the mean and σ is the standard deviation. (2.6) has been derived using the error function, since a normal distribution is assumed.

---

[1] Since the first is the square root of the second, standard deviation and variance can be considered as equivalent measures from an optimization point of view. A decision vector (portfolio) which minimizes one will also minimize the other.

- By a direct historical simulation. For a specific portfolio, this translates to calculating its past realizations and selecting the $\lceil \alpha t \rceil$ worse return value as the VaR, where q_α is the desired VaR percentile. In this case the VaR problem has, generally, a different solution from the variance problem. The interesting point here is that a 'bad' day affects the VaR of a portfolio directly, while with a parametric approach an extreme event has a smoother and more indirect effect.

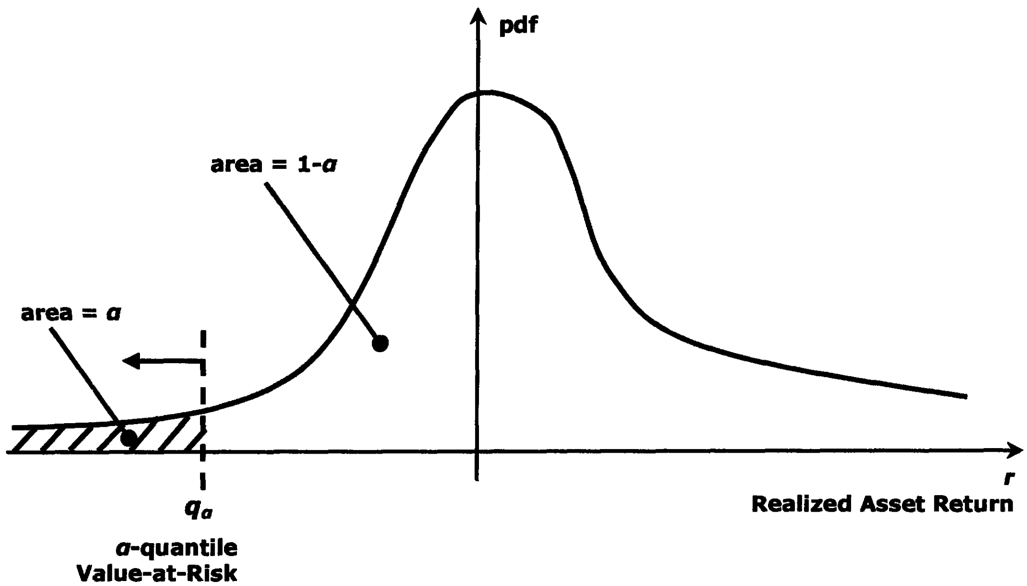


Figure 3. Value-at-Risk expresses the maximum amount of loss a portfolio will suffer with a (1-α) probability. Note that although a probability density function is shown in the sketch, in this work the VaR is not calculated by assuming that asset returns follow a specific distribution, but by direct historical simulation.

In this work the second method, direct historical simulation, is employed. Specifically a 250-day historical sample and the 1-percentile are used. Hence the $\lceil 0.01 \cdot 250 \rceil = \lceil 2.5 \rceil = 3^{rd}$ worse portfolio return inside the rolling time window is the portfolio VaR (see Figure 4). This approach is justified on the grounds of accuracy, since it is the most direct method with the least number of assumptions – there is no need to assume that asset returns follow a specific distribution, which would lead to additional errors. It is also a standard procedure for risk estimation in financial institutions like banks (Jorion 1997), and accepted by international banking supervision authorities (Basel Committee on Banking Supervision 2001).

A basic characteristic of the VaR risk measure as used in this work[2] is that it is an empirical, 'black-box' non-convex and non-differentiable objective (or Pflug 2000 for the non-convexity of VaR, see for example Gaivoronski, Pflug 2005). A representative illustration of the non-convexity of the VaR measure can also be seen in Figure 5 where the VaR of a portfolio of two assets has been plotted. It is also obvious that the combination of the two assets reduces the resulting VaR.

[2] I.e. using a direct, historical simulation.

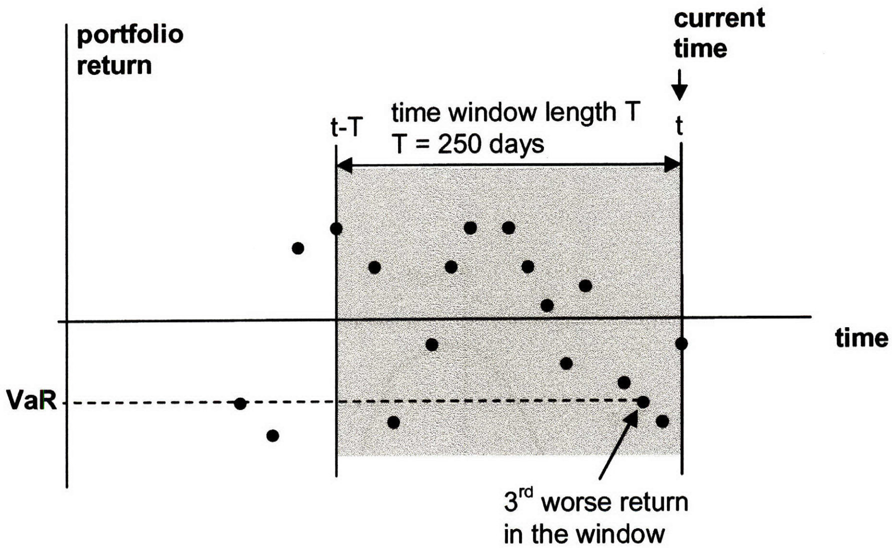


Figure 4. VaR calculation.

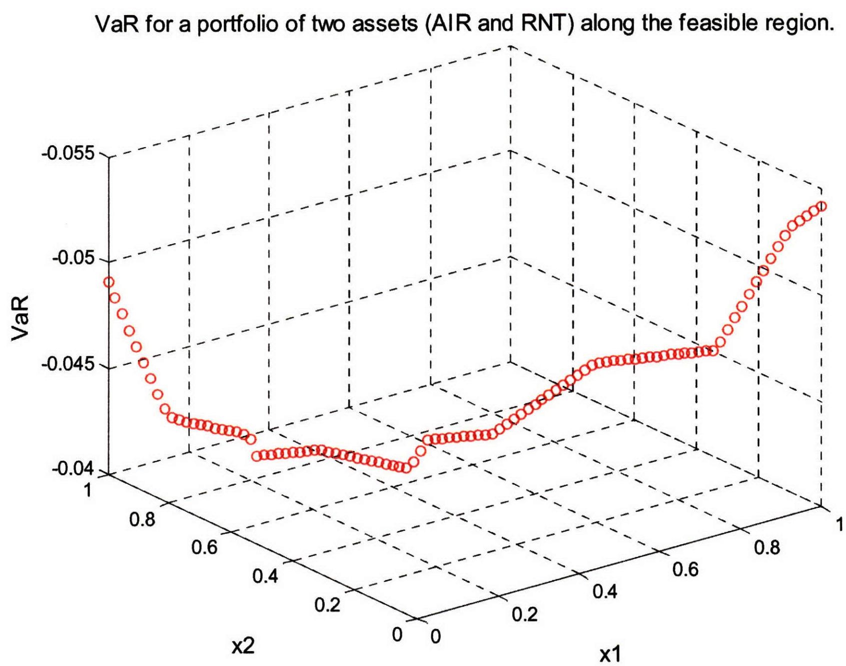


Figure 5. VaR for a combination of two assets x_1 and x_2. The problem's non-convexity is evident.

Hence it is not straightforward to solve the portfolio optimization problem with this risk measure using conventional gradient optimization algorithms. This makes the VaR problem an ideal candidate for evolutionary algorithms.

In this work portfolio optimization is explored under both risk measures. However we concentrate our efforts on the VaR as a risk measure, since it is a direct risk measure accepted in practice, and at the same time difficult to approach with conventional algorithms.

2.3 Problem statement.

Since we have two risk measures, two separate portfolio optimization problems emerge: one for the expected return and the standard deviation, and one for the expected return and the Value-at-Risk. Both are two-objective optimization problems; their solution is a set of portfolios called the *efficient frontier*. If a portfolio is on the efficient frontier, any portfolio with higher (better) return also has higher (worse) risk[3], as we can see in the sketch of Figure 6.

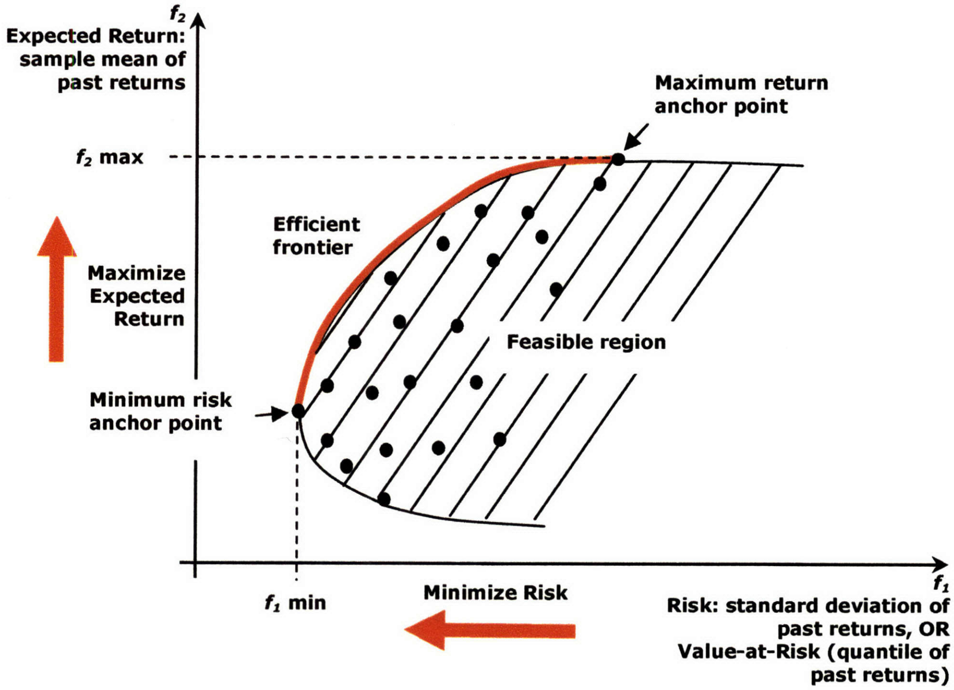


Figure 6. The general portfolio optimization problem as solved in this work.

The definitions of the two problems as implemented in the algorithm can be seen in Table 1. The solutions are subject to the capital availability constraint

$$\sum_{1}^{n} x_i = 1 \tag{2.7}$$

Note that the capital allocations x_i which make up the decision vector may or may not be negative, depending on the l_i bound. In the case when l_i is negative, *short sales* are allowed.

[3] This is the general definition of *Pareto optimality* for multi-objective optimization problems (see for example Deb 2001).

When l_i is zero no short selling is allowed. The allowance of short sales does have a fundamental effect on the problem, since the investor can short securities with a large expected loss and this way greatly increase the portfolio's expected return. We will have a chance to see that in the numerical experiments of chapter 4.

Table 1. Problem statements.

EXPECTED RETURN (MEAN) – STANDARD DEVIATION PROBLEM

Return measure: expected return from the un-weighted sample average

Risk measure: standard deviation from the un-weighted sample average

minimize $\mathbf{f}(x,R(t),t)=\left[risk(x,R(t),t),-return(x,R(t),t)\right]^T$

where:

$$return(x,R(t),t):=E[R(t)x(t)]=\frac{1}{t-1}\sum_{k=1}^{t-1}\sum_{i=1}^{n}r_i(k)x_i(t)$$

$$risk(x,R(t),t):=\sqrt{\frac{1}{t-1}\sum_{k=1}^{t-1}\left(\sum_{i=1}^{n}r_i(k)x_i(t)\right)^2-\left(\frac{1}{t-1}\sum_{k=1}^{t-1}\sum_{i=1}^{n}r_i(k)x_i(t)\right)^2}$$

subject to:

$l_B\le x_i\le u_B$, for $i=1,\dots,n$ (if $l_B=0$ and $u_B=1$, no short sales are allowed).

$$\sum_{1}^{n}x_i=1$$

EXPECTED RETURN (MEAN) – VALUE -AT-RISK PROBLEM

Return measure: expected return from the un-weighted sample average

Risk measure: Value-at-Risk, 1-percentile, from historical simulation

minimize $\mathbf{f}(x,R(t),t)=\left[risk(x,R(t),t),-return(x,R(t),t)\right]^T$

where:

$$return(x,R(t),t):=E[R(t)x(t)]=\frac{1}{t-1}\sum_{k=1}^{t-1}\sum_{i=1}^{n}r_i(k)x_i(t)$$

$$risk(x,R(t),t):=VaR(x,R(t),t)=\begin{bmatrix}3^{rd}\text{ worse return of portfolio }x\\ \text{in the last 250 days (1 percentile)}\end{bmatrix}$$

subject to:

$l_B\le x_i\le u_B$, for $i=1,\dots,n$ (if $l_B=0$ and $u_B=1$, no short sales are allowed).

$$\sum_{1}^{n}x_i=1$$

The complete optimization problem is time changing, as we can see in the problem statements. Each day new market data arrives, changing the statistical risk and return measures for each asset and this way changing the optimal asset allocations. The algorithm is called to discover an approximation to these optimal asset allocations and to the efficient frontier at each time step.

We must note here that although we have a time-changing formulation with several timesteps, we are solving a 'portfolio optimization problem' in the sequential single-period sense, as discussed by Markowitz in the third chapter of his book (Markowitz 2000), not in the utility-maximization dynamic programming sense. Numerous treatments of the latter exist in literature (see for example Merton 1969, Pang 2004, Barro, Canestrelli 2005). The goal of this work is to equip the investors with a tool that provides them with daily knowledge of the efficient frontier in a computationally efficient way. Essentially, we are aiming to create a tool that, using a history of asset returns up to the present time period (for example the current day, if the trading/allocation happens daily), can approximate the optimal allocation front as closely as possible given the available computational resources.

The performance of the portfolio – its actual return when deployed in the market – depends on two factors:

- How good the derived optimal solution is, given the performance measures used for the decision. I.e., how close the solution is to the efficient frontier of risk/return combinations.
- How accurate the performance measures that are used as objectives are. I.e., how accurate the arithmetic average of past returns is in predicting the next period's return for each asset.

As we noted before in this work we are mainly concerned with the first issue. The solution of problems such as the ones in Table 1 is computationally expensive, partly due to the fact that the decision space is very large if in practice a portfolio manager wishes to find optimal allocations among hundreds or thousands of assets. The use of a multi-objective evolutionary algorithm is justified by:

- The inherent ability of EAs to handle multi-objective problems and discover non-dominated fronts in a single run (Deb 2001).
- The problem size. EAs are known to be able to handle problems of very large size due to their implicit parallel processing ability.
- The nature of the VaR risk measure as an objective function (non-linear, non-convex and in cases discontinuous).

2.4 Data sets used for the numerical experiments

Throughout the experiments presented in this thesis we have used two sets of data, as shown in Table 2. These are the collections of assets from which the investor can select in order to create their portfolio.

The time period is two years (2004-2006). Since a rolling window of 250 days is used and one year of data is needed behind each optimization timestep, the optimization process begins halfway through the dataset . Hence the first calculation is for the 250th trading day, aiming for the 251st.

The static optimization results we present in chapter 4 are for the 250th day (first day is considered 'zero' day, where the returns calculation begins), i.e. the data up to and including the

30th of December 2004 is used to create an efficient frontier for a portfolio with a horizon of one day to the 31st of December 2004.

Table 2. Data sets.

Data set	# stocks	Start date	End date
The 30 Dow Jones Industrial Average (DJIA) index members daily adjusted. **name: DJIA_daily_adjusted**	30	Jan 2 2004	Dec 30 2005
From the Standard and Poor's 600 Small Cap index members, the 10 first stocks with expected EPS growth > 10%, daily adjusted. **name: 10_from_SP600_daily_adjusted**	10	Jan 2 2004	Dec 30 2005

The time-changing results in the sixth chapter begin with that day (30th December 2004) and continue as long as the data lasts (to the 30th December 2005), using a rolling window of 250 days behind the current trading day.

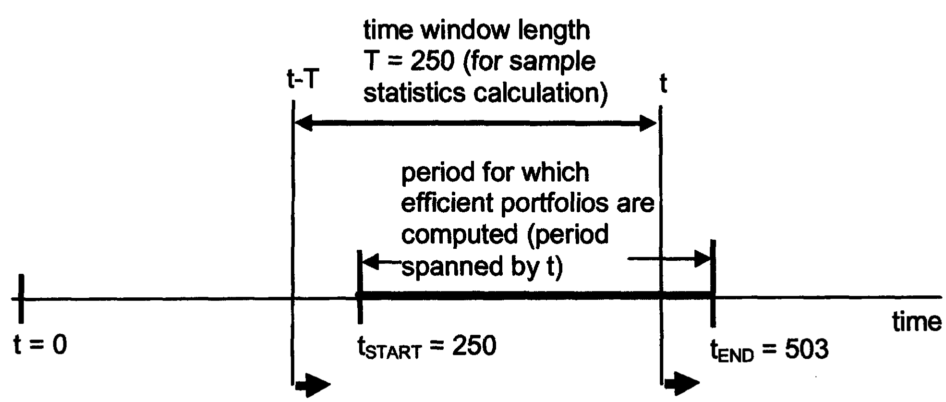


Figure 7. Data time window.

In terms of the source data, the first set is the 30 DJIA index members. The second data set is picked in order to use stocks with different behavior. We randomly (alphabetically) selected 10 stocks from the SP600 Small Cap index, under the constraint that the underlying corporation has a positive profit margin and more than 10% estimated one-year earnings per share (EPS) growth.

2.5 Conclusion

The portfolio optimization problem as explored in this work was presented in this chapter. The problem is treated in its time-changing form as a multi-objective problem under the conflicting criteria of risk and expected return. Two different risk measures are used: standard deviation and Value-at-Risk. Value-at-Risk is calculated using a direct historical simulation, without assuming any parametric probability distribution for the portfolio returns. While standard deviation is a convex objective, Value-at-Risk is a non-convex and often discontinuous function that naturally lends itself to an evolutionary optimization approach. For this reason, this thesis is slightly more focused on the Value-at-Risk problem. The statements of the two problems, with either standard deviation or Value-at-Risk as risk measures, are given and the chapter concludes with the introduction of the two data sets used for the numerical experiments of chapters 4 and 5.

Having just described the optimization problem, the computational tool used for its solution will be presented in the next chapter.

3 Evolutionary Algorithms and the Dynamic Queuing Multi-Objective Optimizer

The Dynamic Queuing Multi-Objective Optimizer (D-QMOO), the optimization tool used to solve the asset allocation problem, is presented in this chapter. First, a brief introduction to evolutionary algorithms is given, followed by some definitions from multi-objective optimization which will be useful later. D-QMOO is described next along with a discussion on its application to the portfolio optimization problem and the performance measures used.

3.1 Evolutionary algorithms

Evolutionary algorithms (EAs) are heuristic optimization tools. Their goal is to discover globally optimal solutions in single- or multi-objective problems. Their general principle of operation is inspired by the selection and evolution process encountered in natural species.

In contrast with most conventional optimization methods (such linear programming), evolutionary algorithms do not work with a single solution. A set of solutions called the *population* is used by EAs and several solutions are being processed simultaneously at any given time during the course of the optimization. In an abstract way, this population resembles the population of a natural species. Each solution is referred to as an *individual.* During each iteration of the algorithm, new solutions (usually called *children*) are derived from the existing individuals using a set of assignment operators. There are two main classes of operators: *crossover* and *mutation.* Crossover operators combine the characteristics of two or more existing individuals (the *parents*) in order to create a new solution. Mutation operators alter the characteristics of an individual by a random amount. Since an optimization problem is being solved, the individual's characteristics that are defined and altered by the assignment operators are elements of its design vector.

Each individual is characterized by its *fitness* measure which is derived from the objective function(s) of the optimization problem. A solution with good objective value has a good fitness measure. Continuing the analogy with the evolution of the species, a selection process exists in the algorithm which favors fit individuals and helps them survive and procreate, much like the survival of the fittest process one encounters in nature. The goal of this process is to start from a random population of solutions and gradually create better individuals, finally converging on the optimization problem's global optimum.

Evolutionary algorithms incorporate some randomness in their operators (e.g. in the mutation operators). This randomness, the way their operators function, and their population-based nature offers EAs some significant advantages over many conventional optimization algorithms. EAs are global optimization tools in the sense that they explore the whole design space and have the potential to discover a global optimum instead of being trapped in local optima – even if one part of the population is converging to a local extremum, individuals might be exploring other areas of the design space. This population-based nature provides the decision maker with a more comprehensive overview of the design space and allows the discovery of several local optima along with the problem's global solution, which might prove to be useful information. Another vital attribute of EAs is that they are not restricted in terms of the nature of the objective function – for example it may be discontinuous, multimodal or non-differentiable (Branke 2002), in contrast to other approaches such as gradient-based methods.

3.2 Multi-objective optimization problems

Multi-objective optimization problems are characterized by the need to simultaneously optimize several conflicting objectives. Portfolio optimization is such an example – minimizing risk usually reduces the expected return as well (recall the problem definition in paragraph 2.3).

A general statement of a multi-objective optimization problem is:

$$\begin{aligned} &\text{Find } \mathbf{x}^*_t \in \mathrm{X} \subseteq R^n \quad \text{which minimizes} \quad \mathbf{f}(\mathbf{x},t) = \left[f_1(\mathbf{x},t),\dots,f_m(\mathbf{x},t)\right] \\ &\qquad\qquad\qquad\qquad\quad \text{subject to} \qquad\quad l_i \le x_i \le u_i, \qquad i = 1,\dots,n \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\; g_j(\mathbf{x},t) \le 0, \qquad j = 1,\dots,q \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\; h_j(\mathbf{x},t) = 0, \qquad j = q+1,\dots,l \end{aligned} \tag{3.1}$$

where $\mathbf{x}$ is an n-dimensional design vector defined on a set X and $\mathbf{f}$ is an m-dimensional objective function. The g and h functions express a total of l inequality and equality constraints. Variable t represents a temporal dimension that advances in a continuous or discrete manner – it may represent actual time or simply different stages of a problem. In the discrete case, time advances through a series of time steps {…, t-2, t-1, t, t+1, …}.

In order to compare solutions of multi-objective problems, the concept of *dominance* is used:

$$\begin{aligned} &\text{A solution } \mathbf{x} \in \mathrm{X} \subseteq R^n \text{ dominates another solution } \mathbf{y} \in \mathrm{X} \subseteq R^n \\ &\text{if} \quad f_i(\mathbf{x}) \le f_i(\mathbf{y}) \text{ for all } i \in \{1,\dots,m\} \\ &\text{and } f_i(\mathbf{x}) < f_i(\mathbf{y}) \text{ for at least one } i \in \{1,\dots,m\} \end{aligned} \tag{3.2}$$

So a solution dominates another if it is better in at least one objective, and not worse in all the other objectives. And we can define *Pareto optimality*:

$$\begin{aligned} &\text{A solution } \mathbf{x} \in \mathrm{X} \subseteq R^n \text{ is Pareto optimal if it is} \\ &\text{not dominated by any other solution.} \end{aligned} \tag{3.3}$$

The *optimal solution of a multi-objective problem is the set of Pareto optimal solutions (non-dominated solutions).*

In the multi-objective case ($m > 1$), the optimal solution $\mathbf{x}^*_t$ at time t belongs to the set of Pareto-optimal solutions in variable space, which will be called Pareto optimal set (POS - see for example Coello Coello, Lamont 2004). The POS maps onto the Pareto optimal front (POF) of non-dominated points in the objective space. Hence, while in a single objective problem the global optimum is a single solution (see Figure 8), in a multi-objective problem the optimum is composed of a set of solutions, the Pareto-optimal set, as shown in Figure 9.

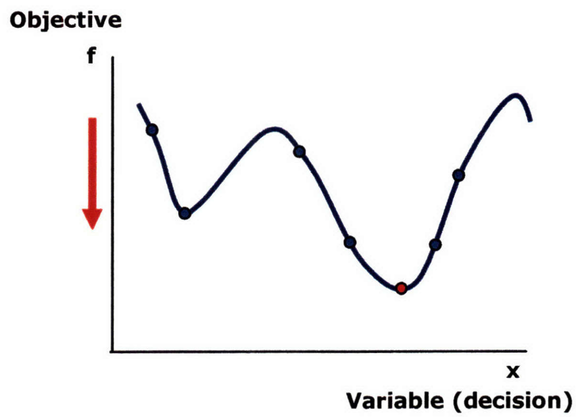


Figure 8. Single-objective problem. The global optimum is marked in red.

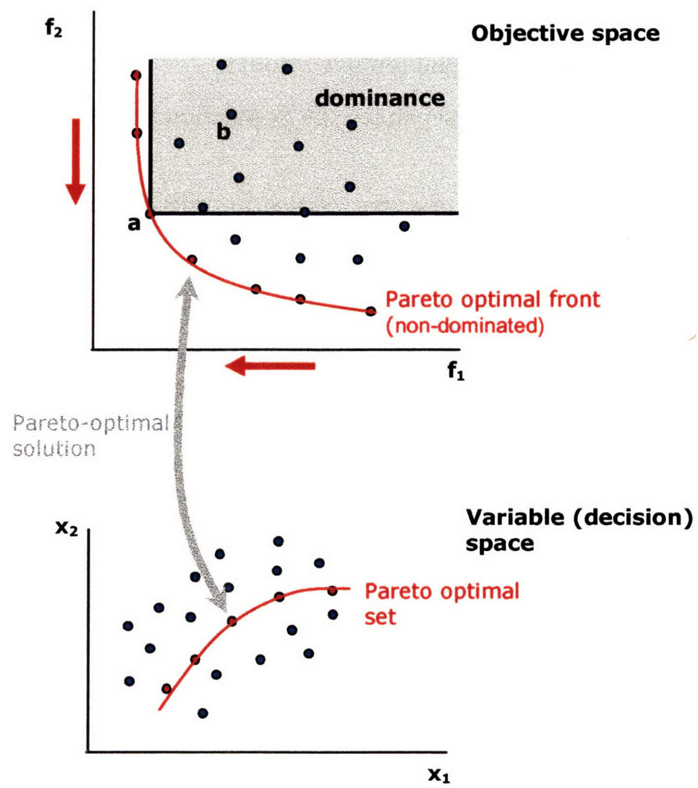


Figure 9. Characteristics of multi-objective problems. In this two-objective problem where the minimization of f_1 and f_2 are sought, solution a dominates solution b and all the other solutions in the gray-shaded area. Solutions are shown both in the objective and the variable space.

The concept of Pareto optimality was first introduced by the Italian economist Vilfredo Pareto (Pareto 1896). Being on the Pareto-optimal front essentially means that one cannot make one of the objectives of the problem better, without making another one worse.

3.3 The Dynamic Queuing Multi-Objective Optimizer (D-QMOO)

The Dynamic Queuing Multi-Objective Optimizer (D-QMOO) is the evolutionary algorithm used in this thesis. D-QMOO is based on the QMOO algorithm (Leyland 2002) developed by Dr Geoff Leyland and other researchers at the Laboratory for Industrial Energy Systems (EPFL-LENI) in Lausanne, as an evolutionary algorithm initially focused on the solution of computationally intensive industrial energy problems. However, QMOO has proven to be a robust and well-performing optimization algorithm, and as a result it has been implemented as a search tool in the Distributed Object-based Modeling Environment (Wronski 2005). QMOO's robustness and wide applicability have also been verified in further work by the author.

QMOO has been developed into Dynamic-QMOO (D-QMOO) in the course of the author's doctoral research at MIT CADLab (Hatzakis 2007). D-QMOO has the ability to handle constrained and time-changing optimization problems. A comprehensive description of the algorithm can be found in the author's doctoral thesis. Here, some key characteristics of D-QMOO are briefly listed:

- D-QMOO is an *elitist* algorithm. A Pareto-optimal (good) solution will never be eliminated unless it is replaced by a better solution.
- D-QMOO is a *steady-state* algorithm. Its population is not characterized by generations which are completely replaced by new ones. The population is in a constant state and solutions are incrementally added and removed from it.
- D-QMOO performs *grouping*. It separates the population into groups which independently explore the design space.
- D-QMOO performs *evolutionary operator choice* in order to select crossover and mutation operators. Four different option for crossover (sbx, blend, linear, uniform) and three different options for mutation (uniform, normal, global) operators are available; the algorithm adapts to each optimization problem by favoring the most successful operators.

A block diagram of the algorithm's basic iteration is shown in Figure 10.

3.3.1 Solving time-changing optimization problems with D-QMOO

Time-changing problems present a special challenge to evolutionary algorithms. Let us have a look at the sketch of Figure 11. It is a two-objective problem, and the design vector has two variables, x_1 and x_2. If this was a static problem, the EA would only need discover the first optimal solution: the first Pareto optimal set, shown with a black line as POS_t. However at t+1 the objective landscape changes and the optimal solution moves to POS_{t+1}. The EA now has to discover the new optimal solution, after having converged on the previous one. In the next timestep the optimal solution moves to POS_{t+2}, and so on.

D-QMOO employs an algorithmic architecture for the solution of time-changing problem that aims to provide it with a well-performing and at the same time robust solving ability. This architecture is based on the simultaneous presence of two elements:

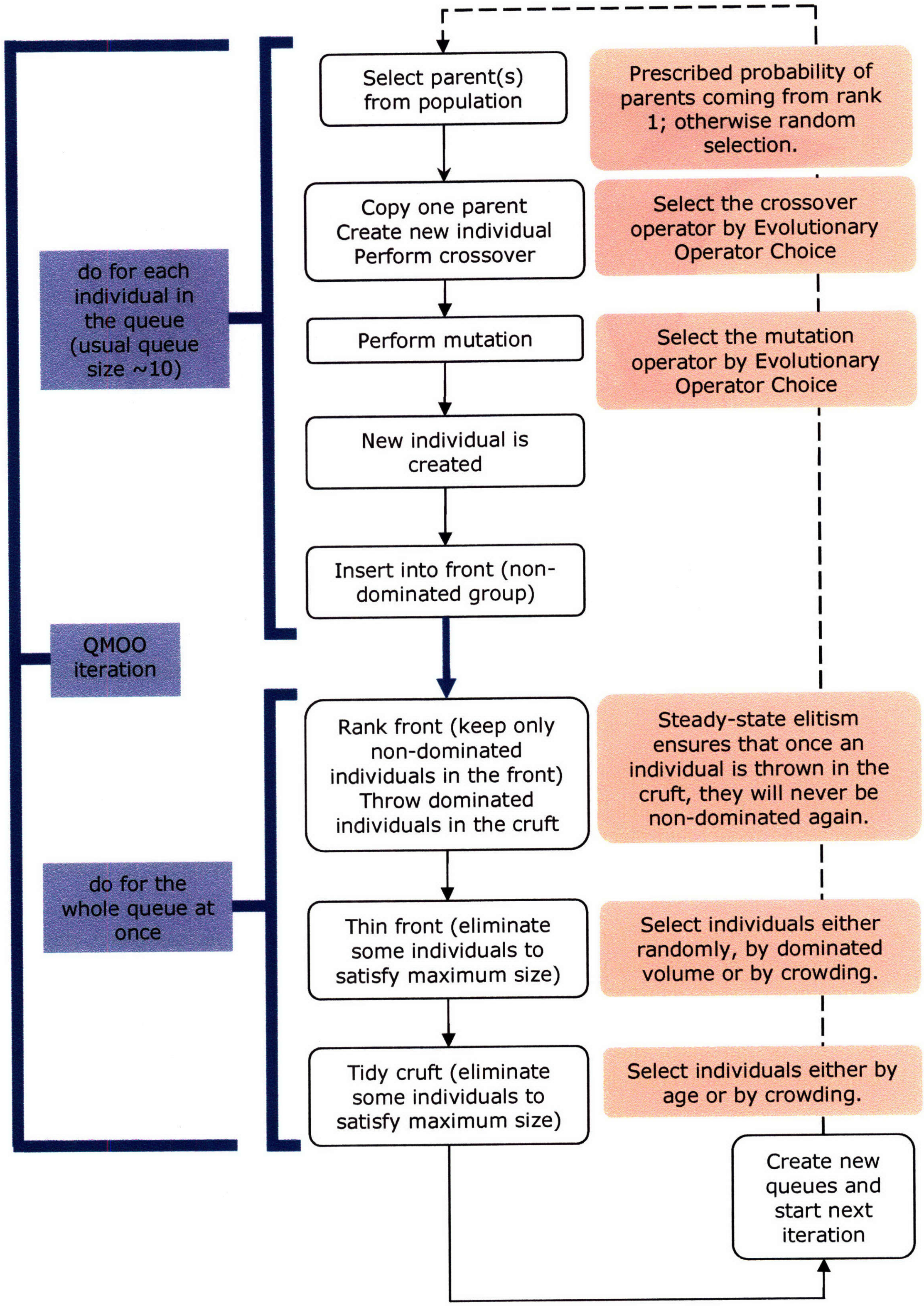


Figure 10. How QMOO and D-QMOO work (Hatzakis 2007).

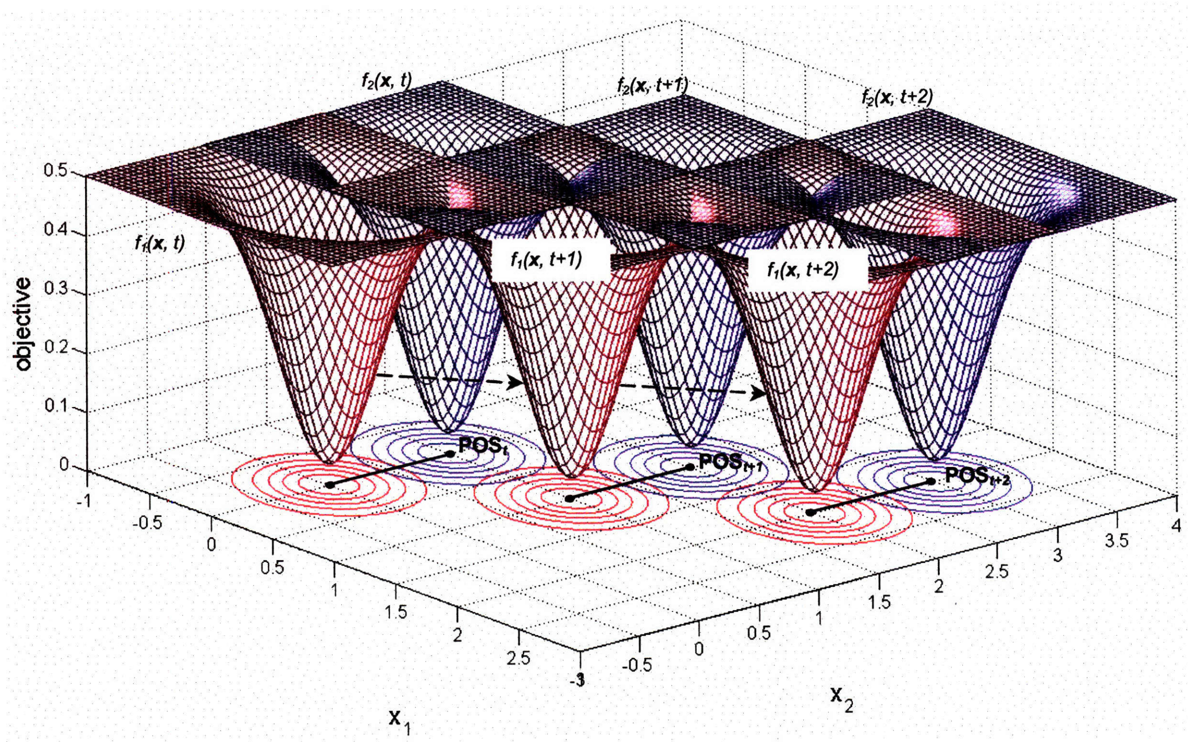


Figure 11. Time-changing two-objective problem. The minimization of f_1 (in red) and f_2 (in blue) is sought.

- An *anticipatory population* which helps the algorithm discover the new optimum when the objective changes in time. This is done by estimating the optimal solution's motion with a forecasting model and placing anticipatory individuals at its estimated future location. As a result the algorithm uses fewer function evaluations and its performance is increased. This method is called *Feed-forward Prediction Strategy (FPS)*.
- A *balance between population convergence and diversity in the design space*, through the retention of an exploratory group of individuals. This means that apart from the non-dominated individuals which converge on the current optimal solution, an additional group of dominated individuals (called the *cruft*) exists. This group preserves diversity – it is scattered over the design space and explores for new (better) solutions. Specifically, the balance of convergence and diversity is provided by:
  - The preservation of this dominated group of individuals whose goal is exploration of the design space.
  - The selection between different criteria for the elimination of individuals. These criteria are age (oldest individuals are eliminated) and crowding (individuals in more densely populated areas are eliminated). The crowding criterion promotes diversity in the population.

This balance assists in the discovery of the new solution, even if the objective moves in an unpredictable way and the anticipatory population cannot be accurately placed near the next optima.

Hence, the anticipatory population increases the algorithm's performance by discovering the new solutions faster. The preservation of diversity on the other hand ensures that the new solutions *will* eventually be discovered, even if the anticipatory population is not successful (also see Figure 12).

The anticipatory population is created with the help of a forecasting model. The optimal solution's past locations are cast into the form of a time series and used as input into a forecasting method such as an Autoregressive model. The forecasting model produces an estimate for the next location of the optimum, and this estimate is used to place individuals there in order to discover the new optimum faster. The function and effect of the anticipatory population is extensively described in various publications by the author (Hatzakis, Wallace 2006b, Hatzakis, Wallace 2006a).

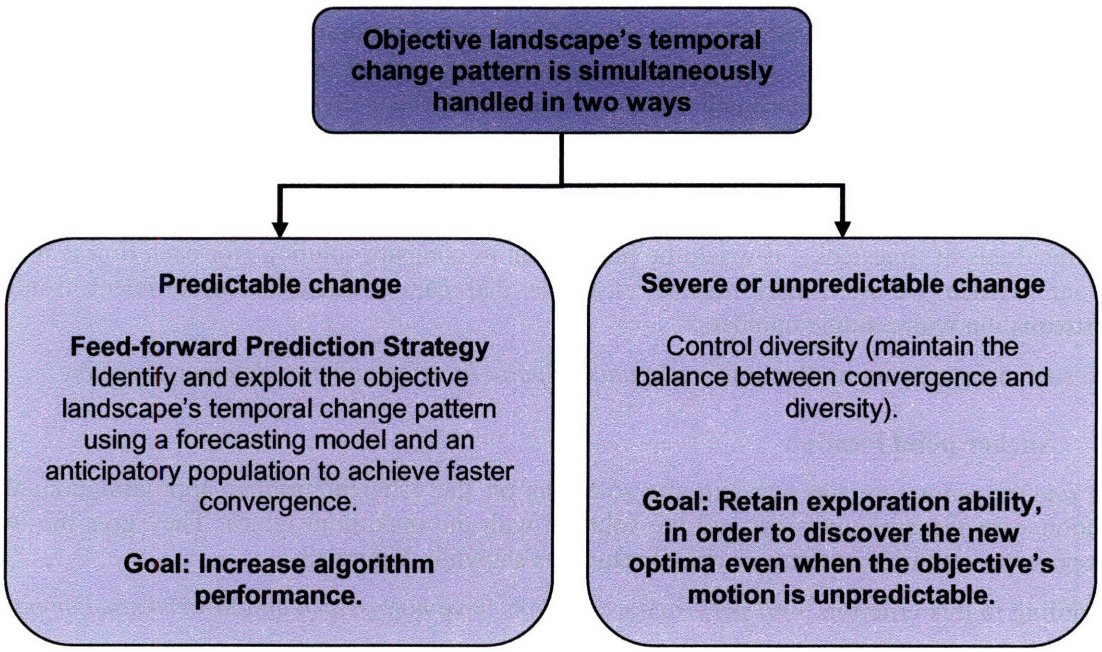


Figure 12. D-QMOO's concept for the solution of time-changing problems (Hatzakis 2007).

3.4 Solving the portfolio optimization problem with D-QMOO

The algorithm's goal when solving the portfolio optimization problem is to discover an accurate approximation to the Pareto front – essentially, to place individuals as close to the efficient frontier curve as possible (recall Figure 6).

Financial portfolio optimization has some characteristics that make it a very challenging multi-objective time-changing problem. The most important one is the lack of predictability in the objective's motion. Indeed, as we will discuss more thoroughly in chapter 5, the optimal solution to the portfolio problem depends on the asset prices' motion. Asset prices are themselves impossible or very hard to forecast, making the optimal asset allocation very hard to forecast as well.

The basic contribution of this work is studying how to solve this problem with an algorithm such as D-QMOO, and developing specialized heuristics for the creation of anticipatory populations since the standard forecasting approach does not work. These issues will be discussed in the next two chapters.

Portfolio optimization with EAs has been studied in the past. The work by Frank Schlottmann (Schlottmann, Seese 2004, Schlottmann, Seese 2005) in credit portfolio optimization has inspired much of the present work, and the author and Dr Schlottmann have collaborated extensively on it.

Here we discuss the performance measures used in the next two chapters in order to compare optimization results.

3.4.1 Performance measures

The first criterion by which we judge the merit of the various frontiers produced by the algorithm is a visual inspection. Indeed, especially in a two-objective problem such as ours where solutions can be plotted in a two-dimensional objective space, it is relatively straightforward to compare two solutions with each other.

Apart from a visual inspection however we need some more 'macroscopic' measures by which to judge and compare solutions. This is mainly because we are solving a time-changing problem with hundreds of timesteps – it would be impractical to compare solutions for each timestep one by one. Hence we need performance measures that can characterize an efficient frontier approximation with a single number.

Two such measures are used in this work: *anchor point location* and *non-dominated volume*.

Anchor point location

The *anchor points* are the two extreme solutions on the efficient frontier approximation: the solution with the minimum risk and the solution with the maximum return. They give the two best possible scenarios in terms of risk or return, as shown in Figure 13.

According to this criterion, *well performing solutions* have *high return* and *low risk anchor point locations*.

Non-dominated volume

The *normalized non-dominated volume* is a comparative performance metric (Zitzler, Laumanns & Thiele 2001, Zitzler, Thiele 1999). This metric is shown graphically in Figure 14. The actual volume dominated by a Pareto front approximation is the un-hatched part of the control volume defined by the utopia and nadir points. The metric's value is the non-dominated hatched area, normalized by the control volume. Hence between two solutions the one with a smaller metric value is better, since it dominates a larger portion of the control volume. This is an appropriate scalar metric since it incorporates both the distance of the Pareto front approximation from some utopian trade-off surface, and its spread.

According to this criterion, *well performing solutions* have a *low non-dominated volume value*.

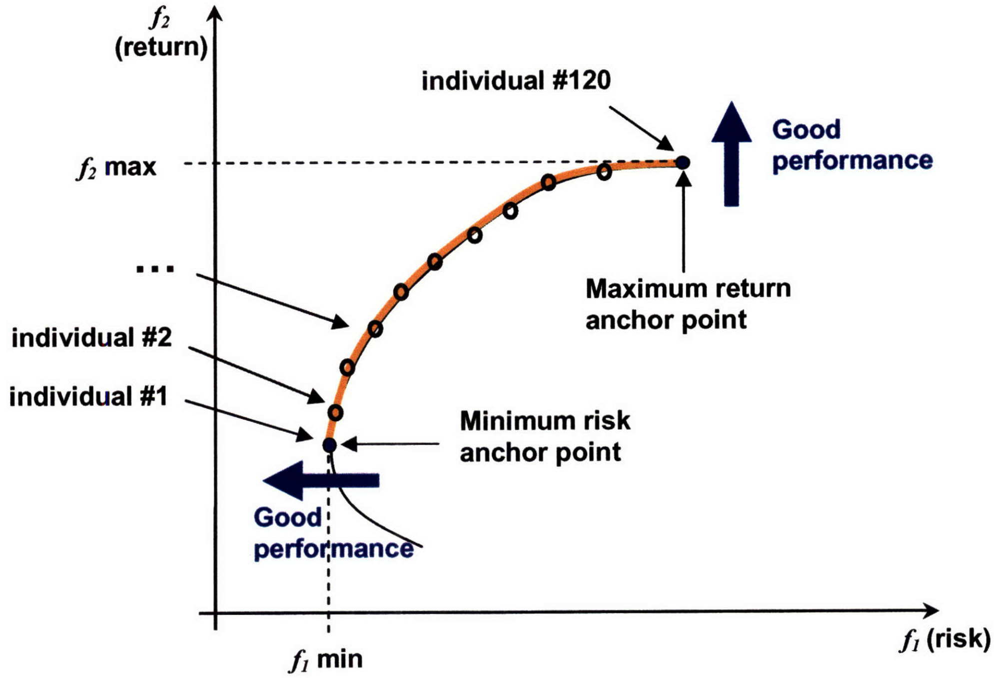


Figure 13. Anchor points and numbering of individuals along the front, for an efficient frontier approximation with 120 individuals.

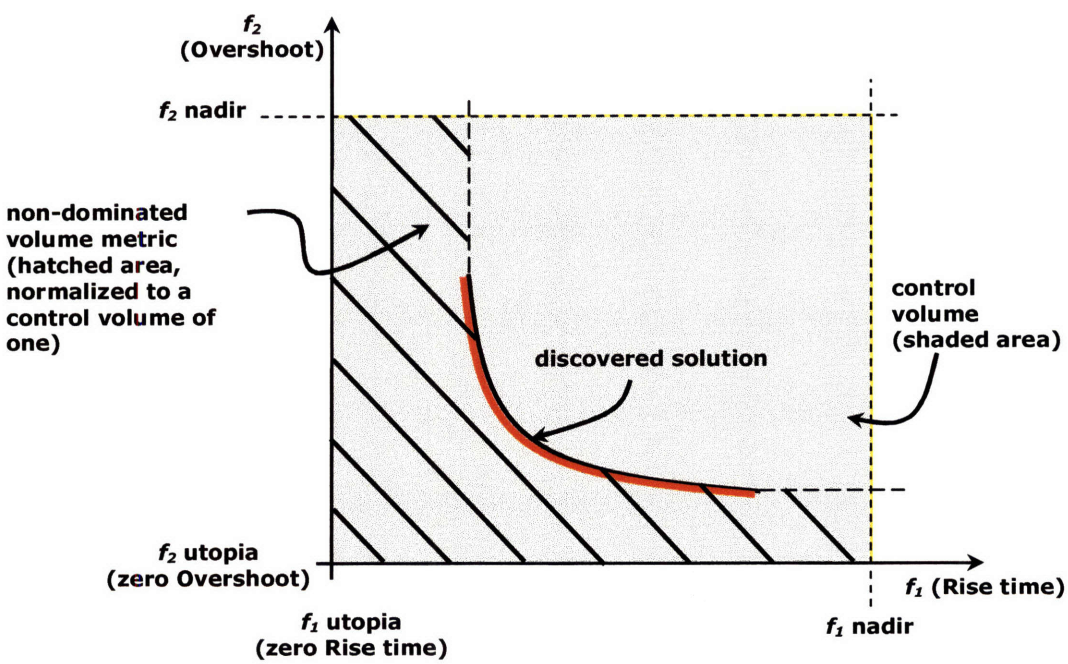


Figure 14. Non-dominated volume performance metric. The hatched area expresses the metric's value. Among two solutions the one with a smaller value is better.

3.5 Conclusion

D-QMOO, the computational tool used for the solution of the portfolio optimization problem, was described in this chapter. D-QMOO uses a combination of two algorithmic elements in order to tackle time-changing multi-objective problems, an anticipatory population and a balance between convergence and diversity. The anticipatory population is created with the help of a forecasting model and increases algorithm performance by helping it discover the successive locations of a moving optimum using less computational time. The balance between convergence and diversity ensures that the moving optimum will be discovered even if the anticipation is not successful.

Some intricacies of the portfolio optimization problem from the algorithmic point of view are discussed next, followed by the performance measures used in this work. These performance measures are the anchor point location and the non-dominated volume, which provide a way to characterize efficient frontier approximations with a single number.

In the following chapter the solution of the static portfolio selection problem is presented.

4 Solving the Static Portfolio Optimization Problem

4.1 General.

In this chapter some sample results from the static portfolio optimization problem are presented, before we discuss the time-changing problem in the next chapter. The static problem deals with the solution of the multi-objective problem for a single timestep[1], given the history of stock returns up to the previous timestep. The goal is to find the efficient frontier of portfolios in the risk-return space for that instance.

Both versions of the optimization problem are solved, with either standard deviation or Value-at-Risk as risk measures, in the following two sections.

4.2 Mean – standard deviation problem.

The mean-standard deviation problem (recall the definition in Table 1) is first solved without allowing any short sales. The results can be seen in Table 3. The amount of computational time used is 65 million function evaluations. If we look at the best obtained values for the risk and return (the anchor points of the Pareto front) at the bottom right figure of the table, we can see that the risk improves rapidly in the beginning (until 0.5 million evaluations, barely visible in the graph) and then continues improving very slowly as the solution converges.

In order to obtain a validity check, the minimum risk portfolio was calculated as a solution to a quadratic optimization problem. The minimum risk portfolio in the mean – standard deviation problem is the portfolio with the minimum standard deviation (minimum variance). Hence it can be discovered by solving the quadratic single-objective minimization problem of Table 4.

This is a common minimization problem, and there is a large number of established tools available for its solution. We used Matlab's *fmincon* function, which applies the Sequential Quadratic Programming (SQP) algorithm. The solution obtained was compared to the minimum risk solution from D-QMOO, which is the top anchor point on the Pareto front (see Figure 13). The two solutions can be seen in Figure 15. The two solutions are not identical but very close, and their risk measure differs only by 0.000146 (around 3%). This comparison serves as a practical validation that the D-QMOO evolutionary algorithm actually solves the 'correct' problem in this region of the Pareto front.

[1] For example, a single trading day.

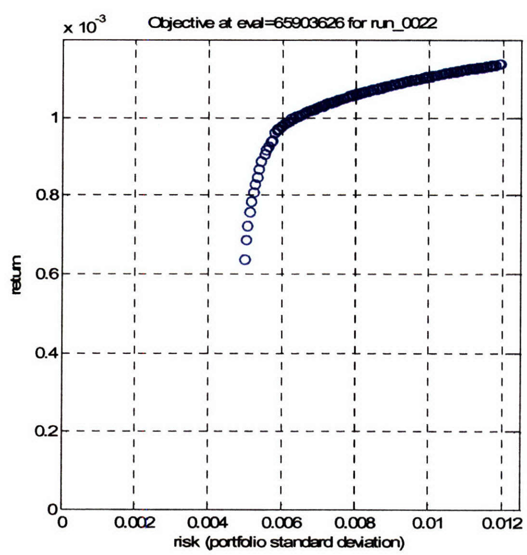
Table 3. Mean - standard deviation solution. No short sales.

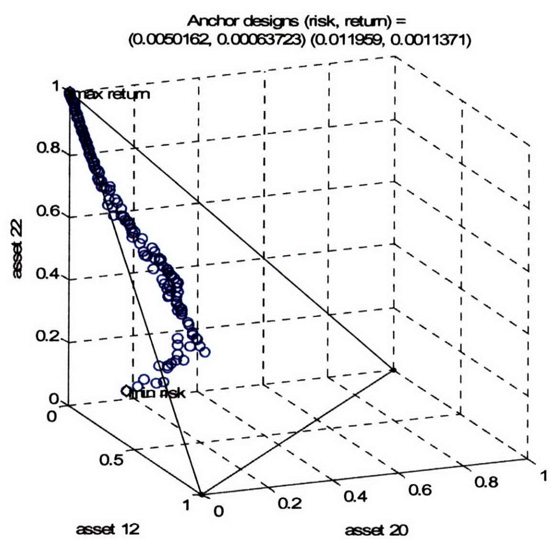
Risk measure	**standard deviation**
Run length:	**65 million evaluations**
Time-changing:	**no (static)**
Short sales:	**no short sales**
Data set:	**DJIA_daily_adjusted**

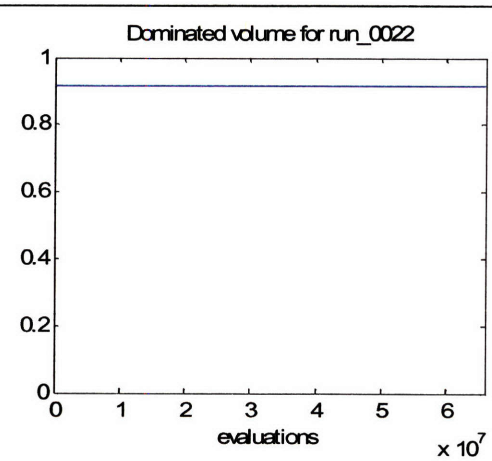
expected returns vector:

0.000009	-0.000471	0.000785	0.000691	0.000038	0.000072	0.000978
0.000742	0.000059	-0.000617	0.000437	0.000980	0.000800	-0.000966
-0.000276	0.000899	0.000403	-0.001156	0.000304	0.000919	0.000420
0.001136	-0.001177	0.000296	-0.001025	0.000526	0.000461	0.000663
0.000146	0.000726					

Final solution obtained







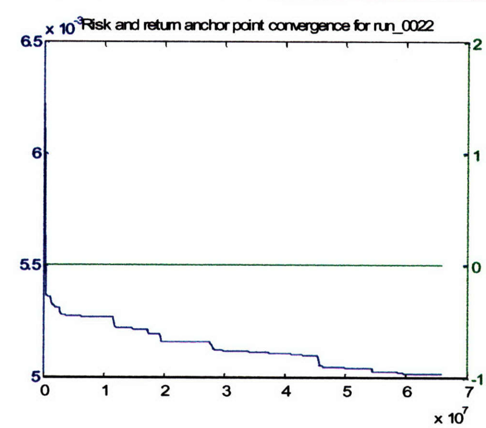


Table 4. Minimum risk problem.

Minimum variance (minimum risk) problem

minimize $f(x,R(t),t) = risk(x,R(t),t)^T$

where:

$$risk(x,R(t),t) := \sqrt{\frac{1}{t-1}\sum_{k=1}^{t-1}\left(\sum_{i=1}^{n} r_i(k)x_i(t)\right)^2 - \left(\frac{1}{t-1}\sum_{k=1}^{t-1}\sum_{i=1}^{n} r_i(k)x_i(t)\right)^2}$$

subject to:

$l_B \le x_i \le u_B$, for $i = 1,\ldots,n$ (if $l_B = 0$ and $u_B = 1$, no short sales are allowed).

$$\sum_{1}^{n} x_i = 1$$

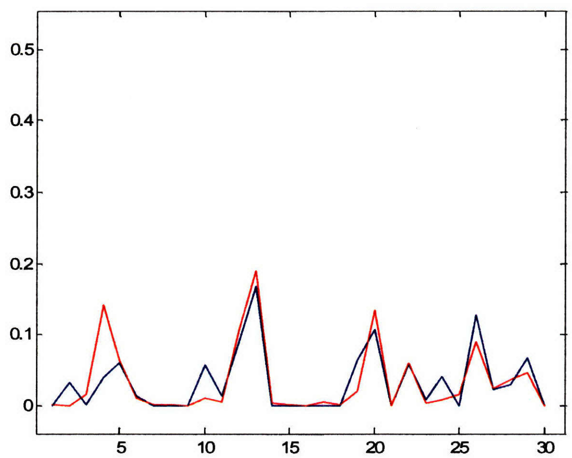


Figure 15. Comparison of the minimum risk solution obtained with Matlab's SQP solver (blue) and the evolutionary algorithm D-QMOO (red). Variables lie on the x-axis (each variable denotes an asset), and the variables' value is on the y-axis (denoting the capital allocation for each asset). The standard deviations obtained are $\sigma_{p\ min\ MATLAB} = 0.00485$, $\sigma_{p\ min\ D\text{-}QMOO} = 0.00501$.

In the second experiment, shown in Table 5, an amount of short-sales is allowed. These results are shown here in order to demonstrate that this solution technique can handle short sales, in the form of negative capital allocations.

The interesting thing to note here is that the inclusion of short sales allows the design of portfolios with much larger expected return. Compare the efficient frontier of Table 5 with the one on Table 3: The highest expected return without short sales is a little more than $1\text{x}10^{-3}$ while with short sales it is around $2\text{x}10^{-2}$, more than 10 times higher. This is because the inclusion of short sales allows the algorithm to short assets with a large expected loss, and allocate more than 100% of the available capital to assets with large expected gains (since other assets can be shorted and hence the total capital constraint satisfied). This way the portfolio expected return can be much higher than the one of the asset with the highest expected return. On the contrary, when short sales are not allowed, the asset with the highest expected return places a limit to the portfolio's highest expected return.

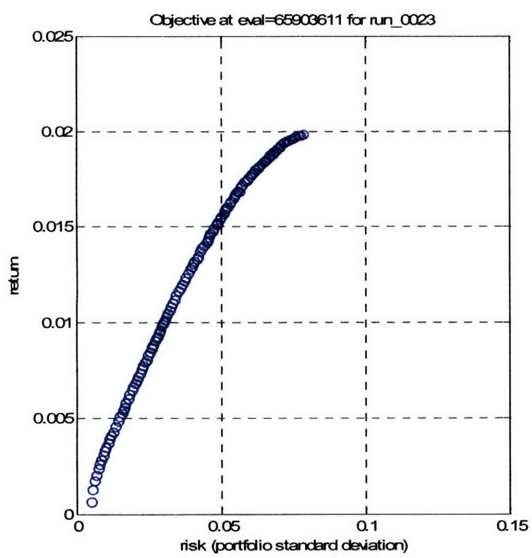
Table 5. Mean - standard deviation solution. Some short sales allowed. Note on the efficient frontier that the expected return is now an order of magnitude larger than when no short sales are allowed.

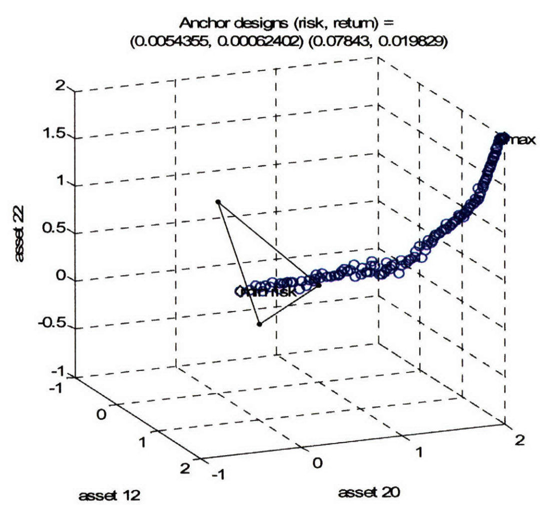
Risk measure	**standard deviation**
Run length:	**66 million evaluations**
Time-changing:	**static**
Short sales:	**Short sales allowed:** $x \in [-1.0, 2.0]$
Data set:	**DJIA_daily_adjusted**

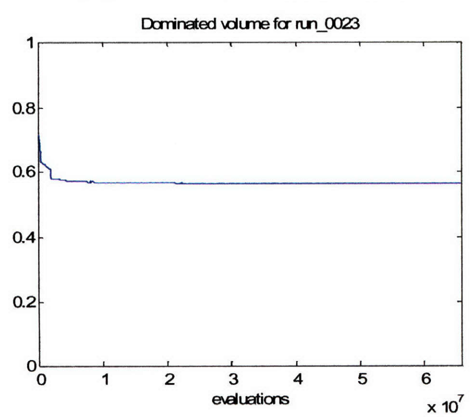
expected returns vector:

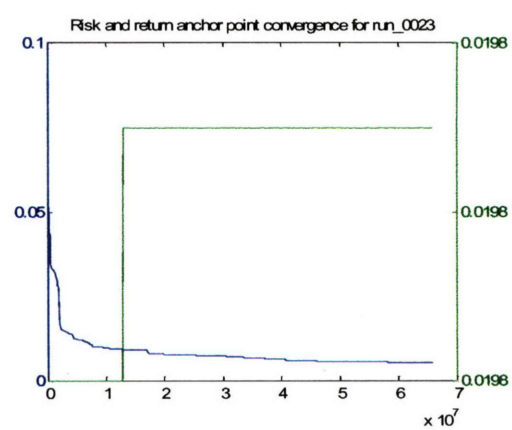
0.000009	-0.000471	0.000785	0.000691	0.000038	0.000072	0.000978
0.000742	0.000059	-0.000617	0.000437	0.000980	0.000800	-0.000966
-0.000276	0.000899	0.000403	-0.001156	0.000304	0.000919	0.000420
0.001136	-0.001177	0.000296	-0.001025	0.000526	0.000461	0.000663
0.000146	0.000726					

Final solution obtained









4.3 Mean – Value-at-Risk Problem.

The mean – VaR problem (recall the definition in Table 1) is explored in its static form in this section. No short sales allowed in the experiments shown here.

Compare the final solution obtained for the mean-VaR problem shown in Table 6, with the solution for the mean-standard deviation problem in Table 3. First, while the same asset provides the maximum return solution (asset 22, MCD), different assets provide the minimum risk solution. Second, it is interesting to see how the mean-VaR solution is more fragmented and discontinuous both in the variable and the objective space, since Value-at-Risk is a non-convex risk measure.

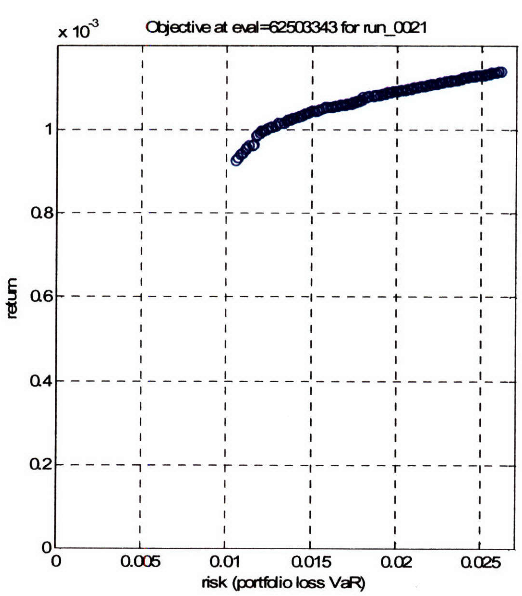
Table 6. Mean-VaR solution.

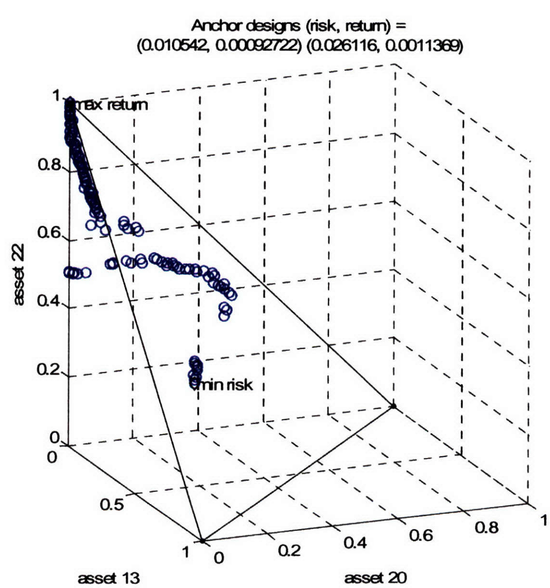
Risk measure	**Value-at-Risk, 1-percentile**
Run length:	**62.5 million evaluations**
Time-changing:	**static**
Short sales:	**no short sales**
Data set:	**DJIA_daily_adjusted**

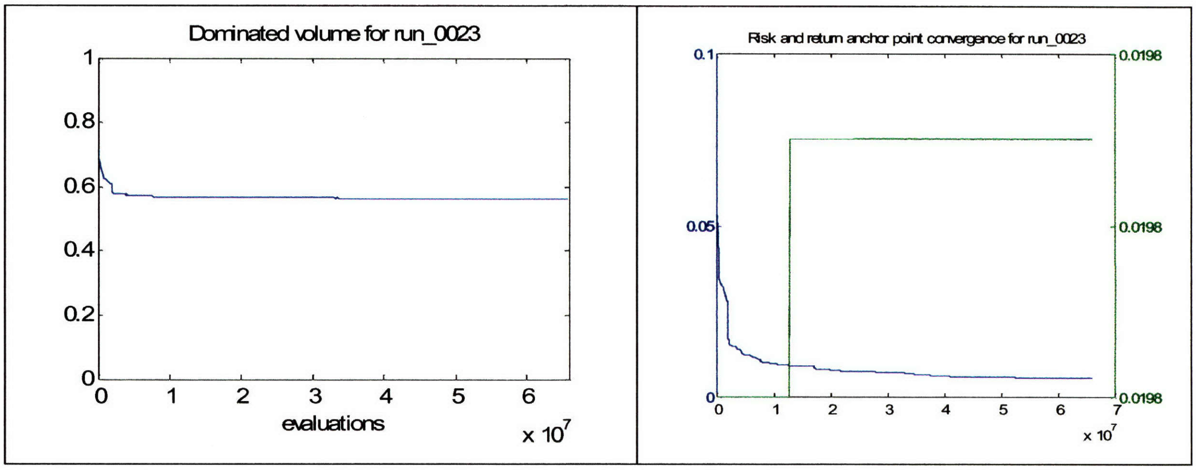
expected returns vector:

0.000009	-0.000471	0.000785	0.000691	0.000038	0.000072	0.000978
0.000742	0.000059	-0.000617	0.000437	0.000980	0.000800	-0.000966
-0.000276	0.000899	0.000403	-0.001156	0.000304	0.000919	0.000420
0.001136	-0.001177	0.000296	-0.001025	0.000526	0.000461	0.000663
0.000146	0.000726					

Final solution obtained







4.4 Conclusion.

Some sample solutions of the static portfolio optimization problem were presented in this chapter. Examples from both the mean-standard deviation and the mean-Value-at-Risk problems are shown. The effect of allowing short sales is discussed, as it dramatically increases the optimal portfolios' expected return. The difference between the two problems (std and VaR) is also seen, partly caused by the non-convexity of VaR as a risk measure.

In the next chapter the time-changing portfolio optimization problem will be studied.

5 Solving the Time-Changing Portfolio Optimization Problem

The time-changing portfolio selection problem is addressed in this chapter. Recalling the problem definitions in Table 1, which are instances of the general problem (3.1), the temporal parameter t is allowed to change expressing the nonstationarity of the problem as trading days advance.

D-QMOO, the evolutionary algorithm described in chapter 3, is used as a solver. Furthermore new heuristic techniques are developed and implemented here in order to solve the problem more efficiently and address the intricacies of portfolio optimization as a time-changing multi-objective problem, which mainly stem from the lack of predictability in the optimal solution's motion.

Recollecting the discussion of section 3.3.1, D-QMOO has two basic mechanisms for the solution of time-changing problems (see for example Hatzakis, Wallace 2006a):

- Anticipatory individuals, placed where the optimal solution is expected to go during the following timesteps. This method is called Feed-forward Prediction Strategy (FPS). The forecasting method used to estimate the location of the next time step's solution is a very important and problem-specific issue. This is especially true in this application, and will be addressed later.
- A mechanism of balancing convergence and diversity in the population, in order to have the exploration ability to discover and track the solution even it moves in an unstructured manner or appears at a new location. This mechanism consists of two elements:
    - The preservation of a dominated group of individuals (the *cruft*) whose goal is exploration of the design space.
    - The selection between different criteria for the elimination of individuals. These criteria are age (oldest individuals are eliminated) and crowding (individuals in more densely populated areas are eliminated). The crowding criterion promotes diversity in the population.

In this chapter we start by solving the time-changing portfolio optimization problem with the 'pure' version of the D-QMOO algorithm, that uses only the convergence/diversity balance mechanism. Subsequently various approaches and heuristics are evaluated in order to improve the performance, i.e. to find better (non-dominated) efficient frontiers in every timestep using the same number of function evaluations.

5.1 Anticipatory populations for the portfolio optimization problem and predictability of asset returns

At this point we need to make an important note regarding the creation of anticipatory populations for the portfolio optimization problem. Specifically, we must make a distinction between *forecasting the asset returns* and *forecasting the location of the optimal solution in the decision space.*

Although of varying accuracy, there is a number of methods which attempt to estimate the price or the return of an asset, one or more time steps into the future. Remember that we are using the one-step-ahead forecast as the objective function of our problem (i.e. we are optimizing a portfolio for this forecast). Then, why not use the two-step-ahead forecast in order to create an anticipatory population? The answer is that simply obtaining a two-step ahead forecast for the asset returns and solving for it is just as expensive computationally as solving the current (one-step-ahead) problem. The key here is that the forecast is for the *asset returns*; however, the feed-forward prediction strategy needs a forecast for *the location of the optimal solution.*

Using a two-step-ahead forecast for the asset returns is, essentially, solving tomorrow's problem with today's data. The goal of FPS is to achieve computational efficiency by directly forecasting the motion of the optimum in the solution space. In this case, this optimum is the design vector of the allocation for the optimal (efficient, non-dominated) portfolios. Hence, we need to find a way to forecast the motion of the portfolios themselves in order to use the FPS effectively.

Here we can distinguish a clear obstacle in using a conventional forecasting algorithm in order to apply the Feed-forward Prediction Strategy with the portfolio optimization problem: Even though some forecasting methods exist, asset returns are in general very hard or impossible to predict. Hence the location of the optimal solution in the decision space, which is a direct product of asset returns, is also very hard to predict. This precludes the use of forecasting methods such as Autoregressive models, which have been used with the FPS on other problems.

5.2 Heuristics for the creation of anticipatory populations

However an anticipatory population for a time-changing algorithm need not necessarily be created using a forecasting model. During the course of this work, 'non-forecasting' heuristics for the creation of anticipatory populations for the portfolio optimization problem were devised and implemented, some of which are shown to be successful. These techniques are summarized in Table 7, and they will also be discussed in more detail later.

Table 7. Algorithmic versions of D-QMOO and heuristics for the time-changing problem. The various heuristics are described in more detail later.

Algorithm versions and anticipation heuristics investigated	
D-QMOO	Base version. No anticipatory population. Only convergence/diversity balance used as a time-changing optimization method.

Maximum return solution seeding.	**Anticipatory population:** At each timestep, the maximum return solution is used as an anticipatory individual, to 'seed' the rest of the population. The location of the maximum return solution is obtained through the asset with the best expected return. Risk is disregarded, and the solution with the maximum expected return is found from a single-objective linear problem: $\mathbf{x}_{\mathbf{maxE}} = [0 \ldots x_i = U \ldots x_j = L \ldots\ldots]$ where $\boldsymbol{\mu} = [\mu_1 \ldots \mu_N]$ and $i = index(\max(\mu)), j = index(\min(\mu))$ and U, L are determined by: $if(x_{\max} > 1 - x_{\min})$, then $U = 1 - x_{\min}$ and $L = x_{\min}$ $elseif(x_{\max} < 1 - x_{\min})$, then $U = x_{\max}$ and $L = 1 - x_{\max}$ $elseif(x_{\max} = 1 - x_{\min})$, then $U = x_{\max}$ and $L = x_{\min}$ under the constraints $\sum_{i=1}^{N} x_i = 1$ and $x_{\min} \le x_i \le x_{\max}$ This technique improved performance.
Minimum variance solution seeding.	**Anticipatory population:** At each timestep, the minimum variance solution is used as an anticipatory individual, to 'seed' the rest of the population. In the mean-std problem, this seed is exactly the minimum risk solution. In the mean-VaR problem, this seed attempts to be near the minimum risk solution. The two solutions do not coincide because the VaR is calculated using empirical historical simulation (recall the discussion in section 2.2.3). The minimum variance solution is found by solving the single-objective quadratic problem of Table 4. This technique improved performance.
Autoregressive forecast seeding.	**Anticipatory population:** Autoregressive models are used to estimate the location of the minimum risk and maximum return solutions, as was done in the introduction of the Feed-forward Prediction Strategy (Hatzakis, Wallace 2006b). Due to the nature of the solution's motion in time (as discussed earlier, and as it will be seen in results presented later), this forecasting method fails to create a prediction that is accurate enough to be useful.
Linear anticipatory population.	**Anticipatory population:** A linear anticipatory population is used. Individuals are placed on a linear segment connecting the estimated anchor points (seeds), under the rationale that in the Markowitz problem, the solution takes the form of linear segment(s). This technique was not very successful either, no matter which kind of forecasting we used for the seeds, mainly due to strong inflexion points which drove the solution away from the 'chord' segment.
Local search for the anticipatory individuals.	The anticipatory individuals, no matter how they have been created, are forced to execute a few steps of local search (see for example the search strategy in Schlottmann & Seese (2004)). Here a 1+1 Evolution Strategy is used as a local search mechanism. The rationale is that there are only two anticipatory individuals, in a total population of 100 or more. Hence they might benefit more from searching on their own around their neighborhood for better solutions than mating with distant unrelated solutions belonging to the previous time step's front. This approach improved performance.

5.3 A note on the two problems (mean-standard deviation and mean-Value-at-Risk)

We must note here that the mean-standard deviation problem is quite straightforward, and has also been treated numerous times in the past literature since it is a quadratic problem that can be solved with existing rigorous methods (e.g. SQP).

For this reason, and recalling the discussion in section 2.2.3, in a large part of this work our efforts are focused on the mean-VaR problem. Historical simulation Value-at-Risk, being an empirical black-box non-convex and non-differentiable function (recall Figure 5 and see Figure 3(Pflug 2000)), lends itself naturally to an evolutionary approach.

5.4 Solving the time-changing problem with the base algorithm

In this paragraph sample results from the solution of the time-changing problem are shown, using the basic form of the D-QMOO algorithm. Only one performance-improving heuristic is used: the maximum return location seed (as described in Table 7), since it is straightforward to implement and was found to increase performance in most cases. No other form of anticipation or performance-enhancement technique is applied. The goal is to study the behavior and performance of the basic D-QMOO on the mean-standard deviation and the mean-Value-at-Risk problems.

The effect of the objective change frequency is specifically studied in this set of numerical experiments. The change frequency is defined as the number of function evaluations per time step. It expresses the computational time available to calculate the efficient frontier during each trading day. Hence it is a vital parameter – if we had infinite evaluations available we would be able to discover the efficient frontier by a crude random search, and a sophisticated algorithm would not be required. The better performing an algorithm is, the fewer evaluations it needs to discover the non-dominated solution. A convergence check for the algorithm is to decrease the frequency (increase the evaluations per time step) until the obtained solution converges to a constant value.

Since it is impractical to compare the obtained non-dominated fronts visually or timestep–by–timestep, we use the anchor points and the non-dominated volume (as defined in section 3.4.1) as macroscopic performance measures and occasionally do a visual comparison at specific time steps.

For these results the DJIA_daily_adjusted data set is used. The time history of the asset prices and the expected returns (arithmetic mean of the past 250 returns) can be seen in Figure 16 and Figure 17.

Sample results from the time-changing mean-standard deviation problem are shown in Table 8 and Figures 18 through 24. These experiments were done as a convergence study. The problem was solved starting from a high objective change frequency (with few function evaluations available per timestep), and this frequency was gradually decreased. As the frequency is decreased the algorithm has more time available per timestep to converge to the solution, and the accuracy is improved. At the *convergence frequency* the obtained solutions start to be similar to each other.

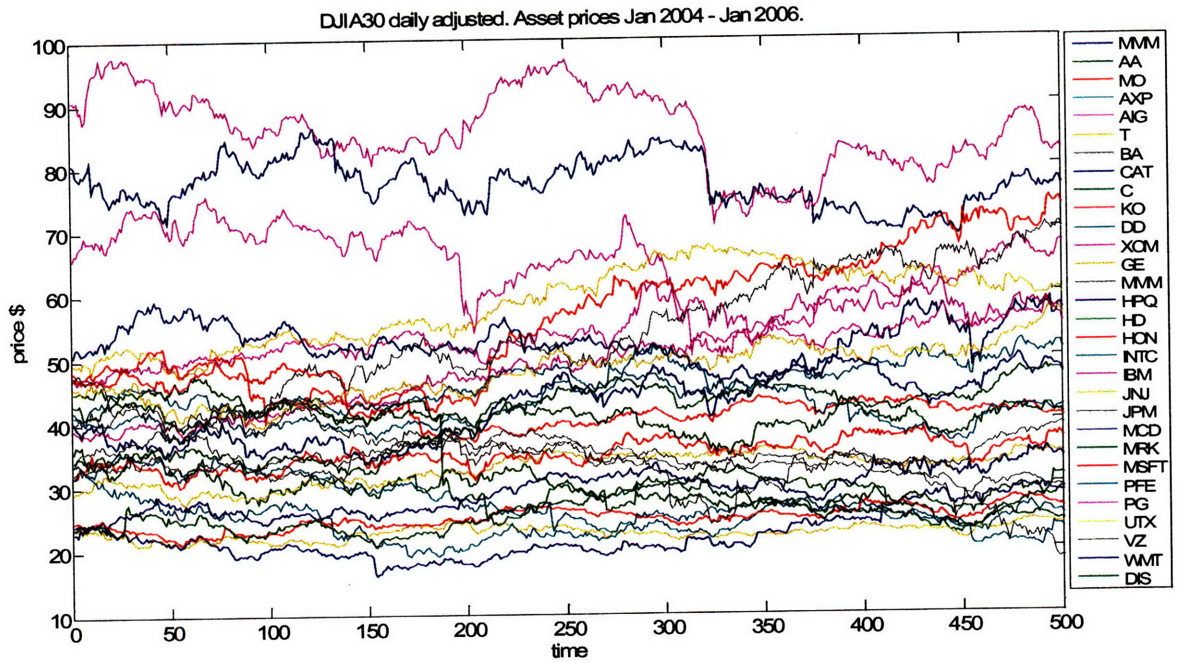


Figure 16. Asset prices time history (DJIA 30).

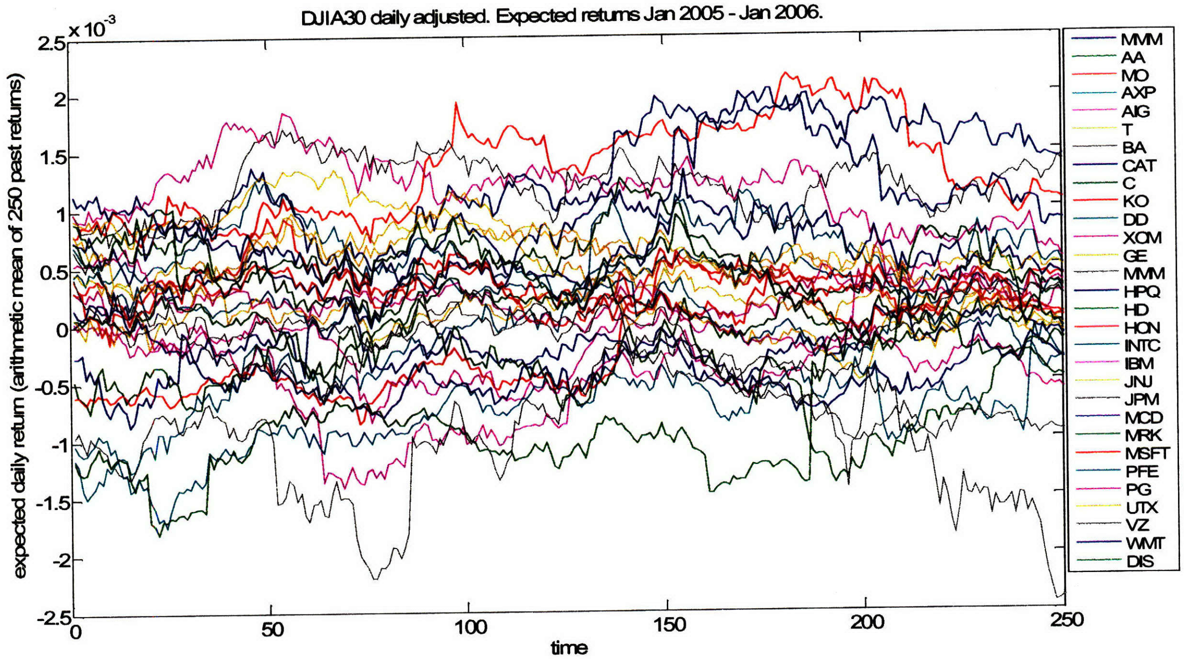


Figure 17. Adjusted expected daily returns time history.

In the first graphs (Figure 18 and Figure 19) the dominated volume metric time history is shown for various objective change frequencies[1]. The period of 100k evaluations per time step is found to be a satisfactory convergence period. Subsequently additional results from that period are shown (asset allocation in Figure 20 and Figure 21, efficient frontier at different time steps in Figure 22, time history of the maximum return and minimum risk solutions in Figure 23 and Figure 24). It is interesting to observe how difficult it would be to forecast the motion of the minimum risk solution in time, as shown in the last figure.

Results from the mean-VaR problem are shown in Table 9 and Figures 25 through 31. Convergence happens between 100k and 500k evaluations per time step.

Table 8. Mean-standard deviation time-changing problem.

Risk measure	**Standard deviation**
Run length:	**254 timesteps (or less).**
Time-changing:	**time-changing, various frequencies**
Short sales:	**no short sales**
Data set:	**DJIA_daily_adjusted**
Anticipation:	**Seeding with maximum return solution at each timestep (this was easy and straightforward to do and has been used at every run in this report).**

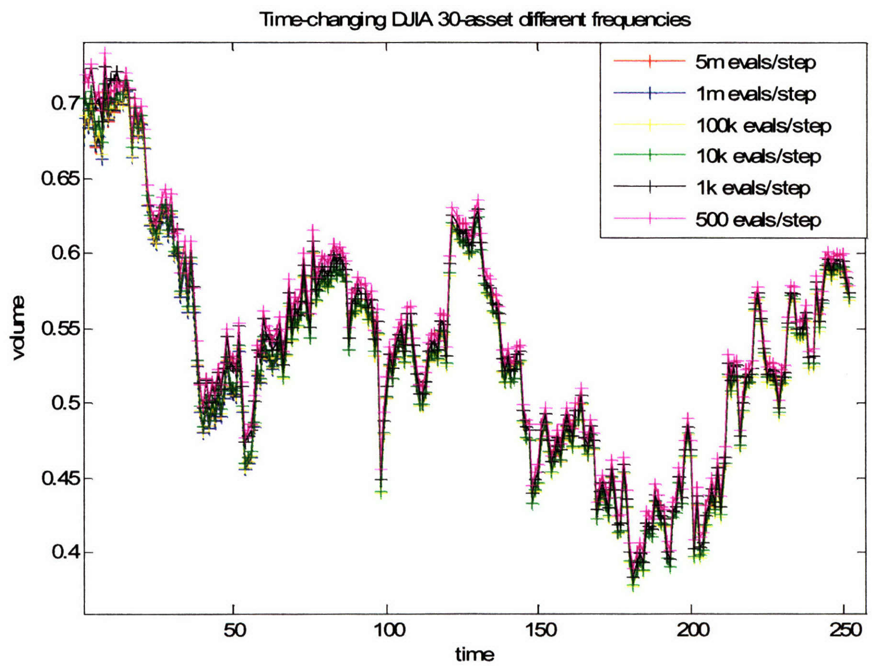


Figure 18. Non-dominated volume. We can see that convergence happens near the period of 100k evaluations/timestep, since the solution accuracy as expressed by the dominated volume remains almost the same up to that period and starts deteriorating above it. The next plots are from results obtained at this period.

[1] Note that we measure 'frequency' by *evaluations per timestep* which is strictly an *objective change period* since less evaluations mean a faster rate of change and hence a higher frequency.

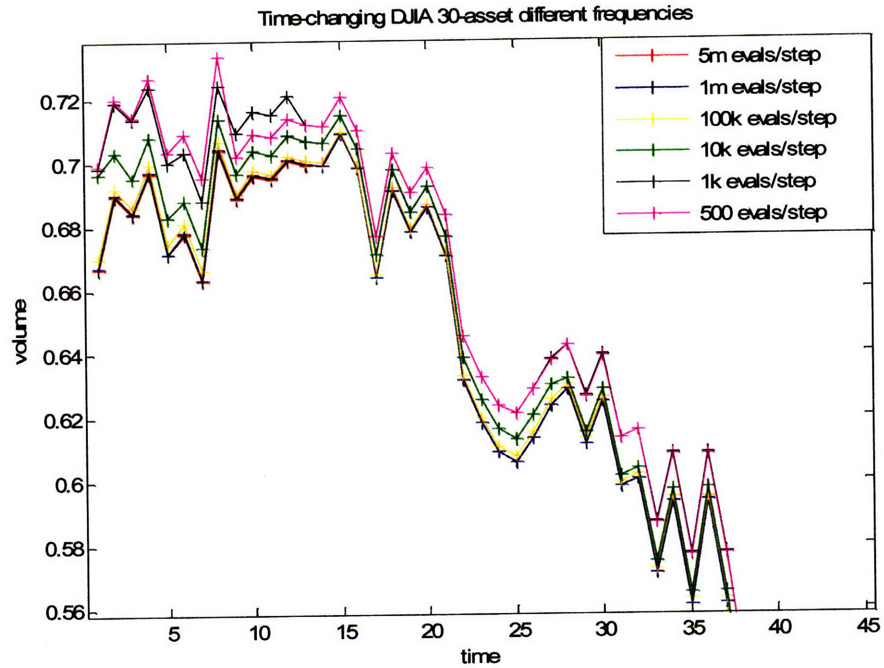


Figure 19. Non-dominated volume (zoom plot).

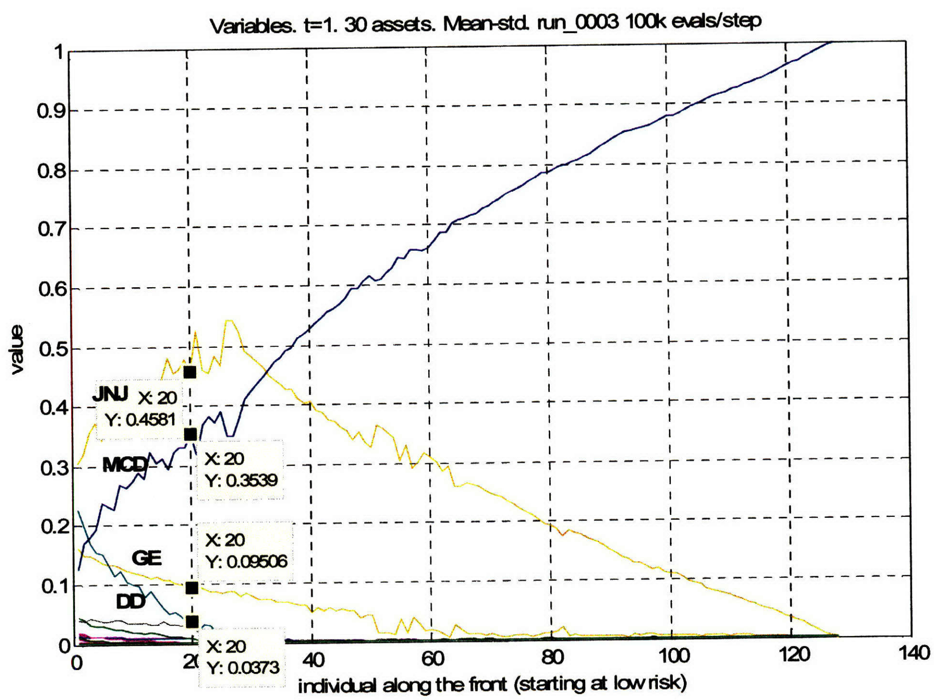


Figure 20. Contribution of various assets along the non-dominated front at the first timestep. The horizontal axis denotes the various solutions along the efficient front. Solution #1 is the minimum risk solution (at the lower left end of the Pareto front) and solution #125 is the maximum return solution. The 20th individual's asset allocations are noted.

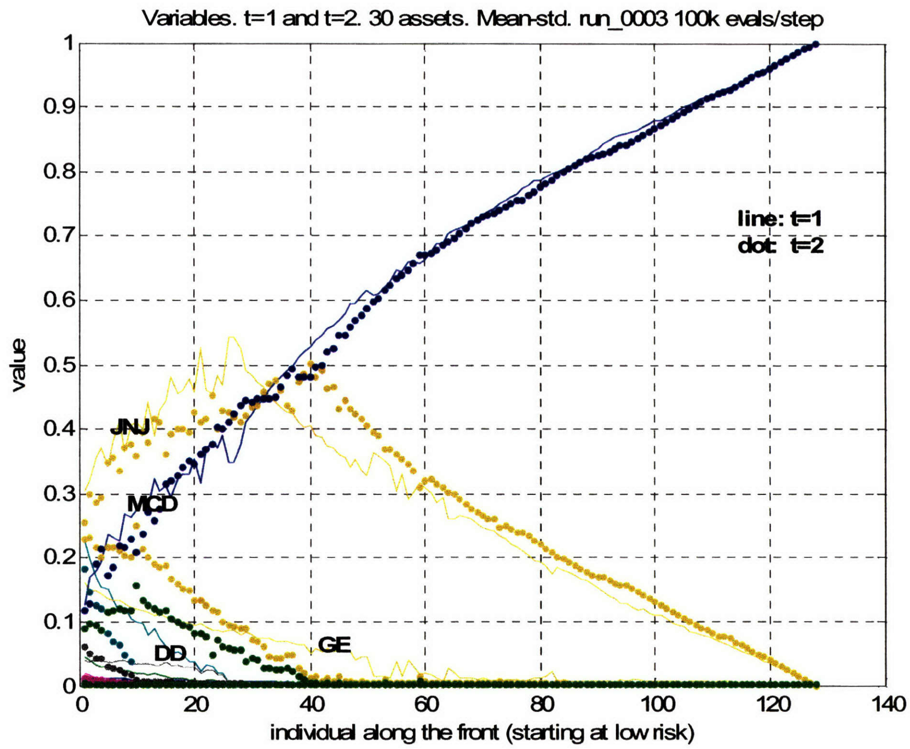


Figure 21. Contribution of various assets along the non-dominated front at the first and second timestep. The second time step is plotted as well, to show the change. Note that, in general, four stocks dominate and the rest have a very small contribution.

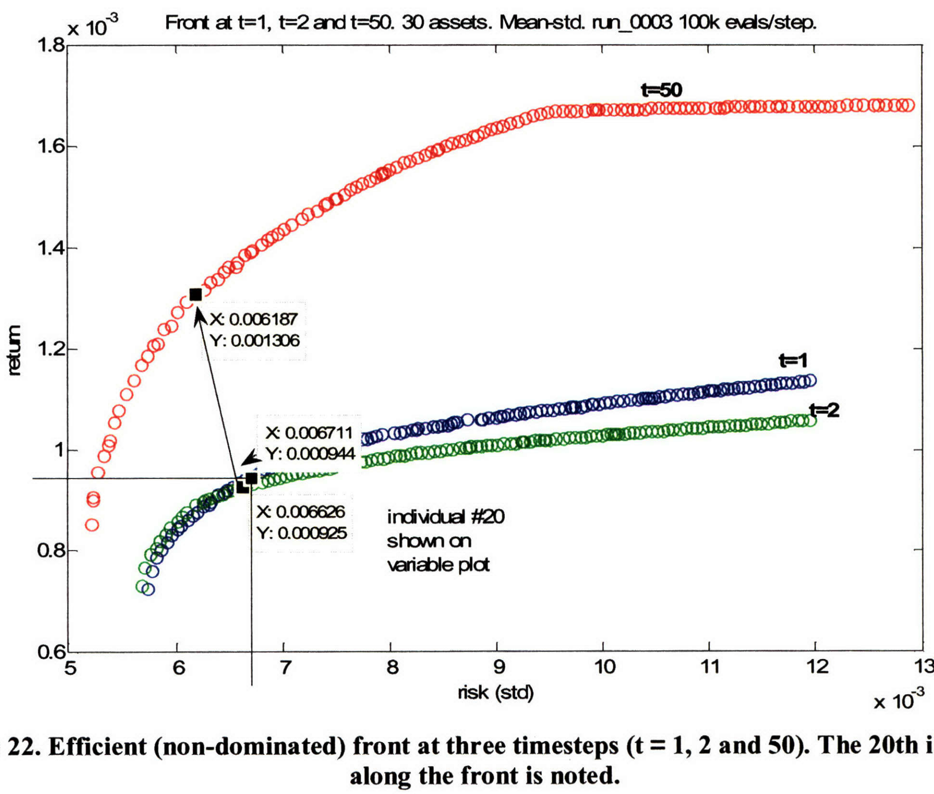


Figure 22. Efficient (non-dominated) front at three timesteps (t = 1, 2 and 50). The 20th individual along the front is noted.

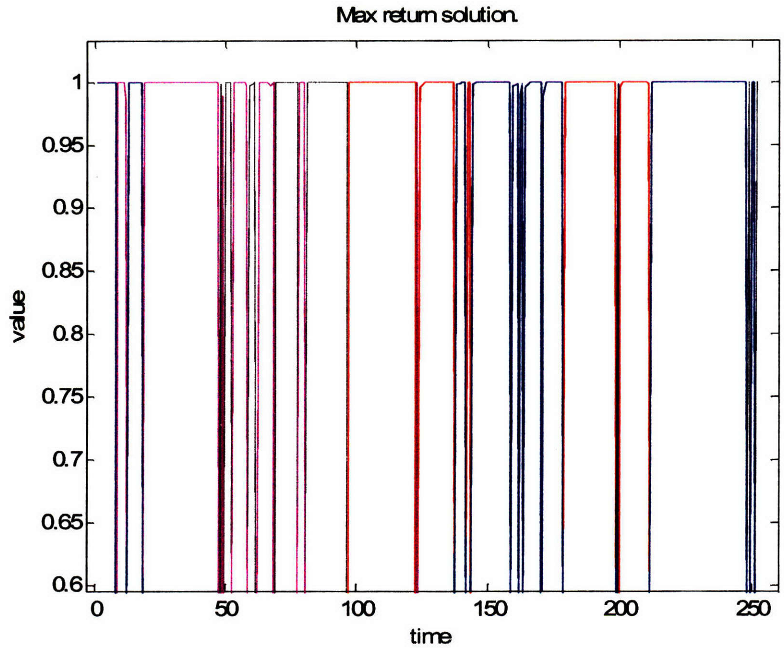


Figure 23. Maximum return solution. A single asset, the one with the highest expected return, comprises this solution. As time advances and new market data arrives, different assets take on this role.

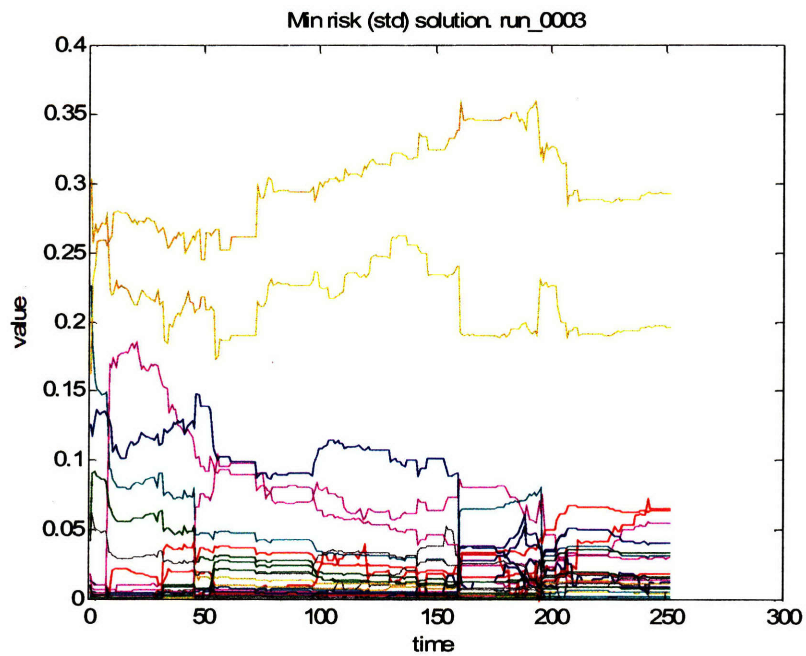


Figure 24. Minimum risk solution (each color indicates a different asset). It becomes obvious even from a simple visual inspection that it is not straightforward to forecast this motion.

Table 9. Mean-Value-at-Risk time-changing problem.

Risk measure	Value-at-Risk (1-percentile).
Run length:	254 timesteps (or less).
Time-changing:	time-changing, various frequencies
Short sales:	no short sales
Data set:	DJIA_daily_adjusted
Anticipation:	Seeding with maximum return solution at each timestep.

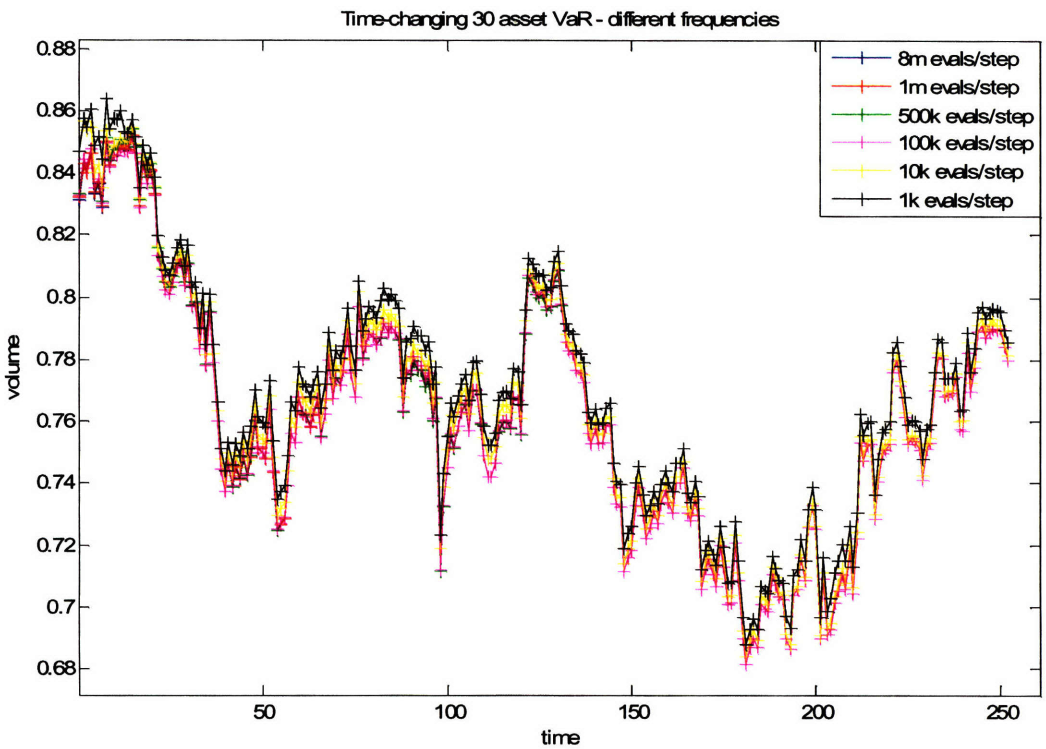


Figure 25. Non-dominated volume. Convergence happens after 100k evaluations, at around 500k evaluations per timestep.

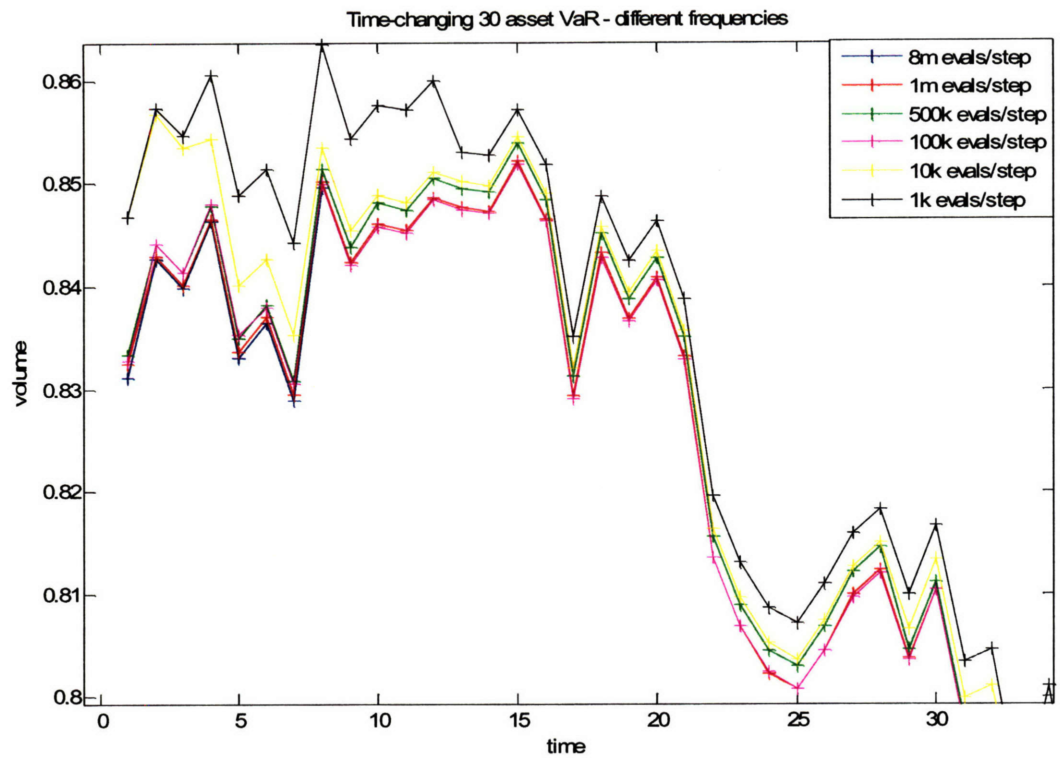


Figure 26. Non-dominated volume (zoom plot).

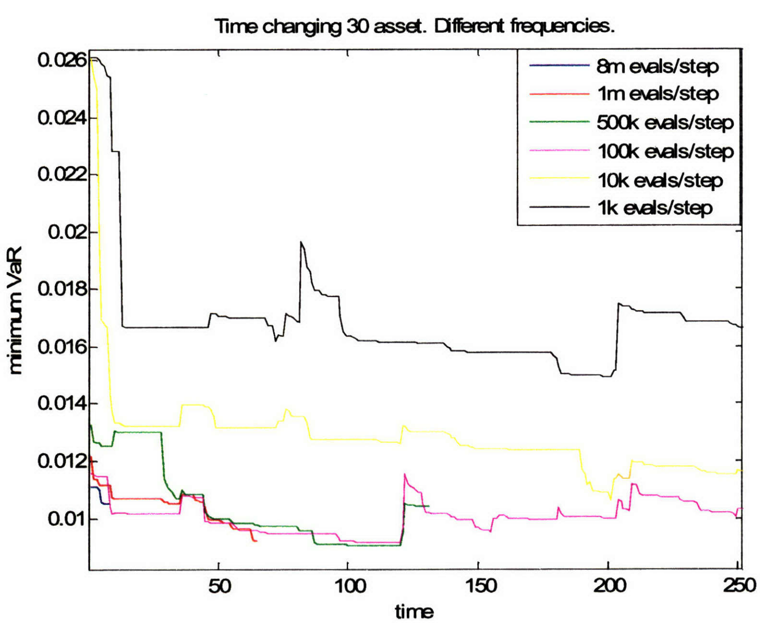


Figure 27. Minimum Value-at-Risk. Here we can see the minimum VaR (the objective value of the minimum risk solution) in time, for various frequencies. It is apparent how much the solution deteriorates for high frequencies (e.g. at 1k evals/timestep).

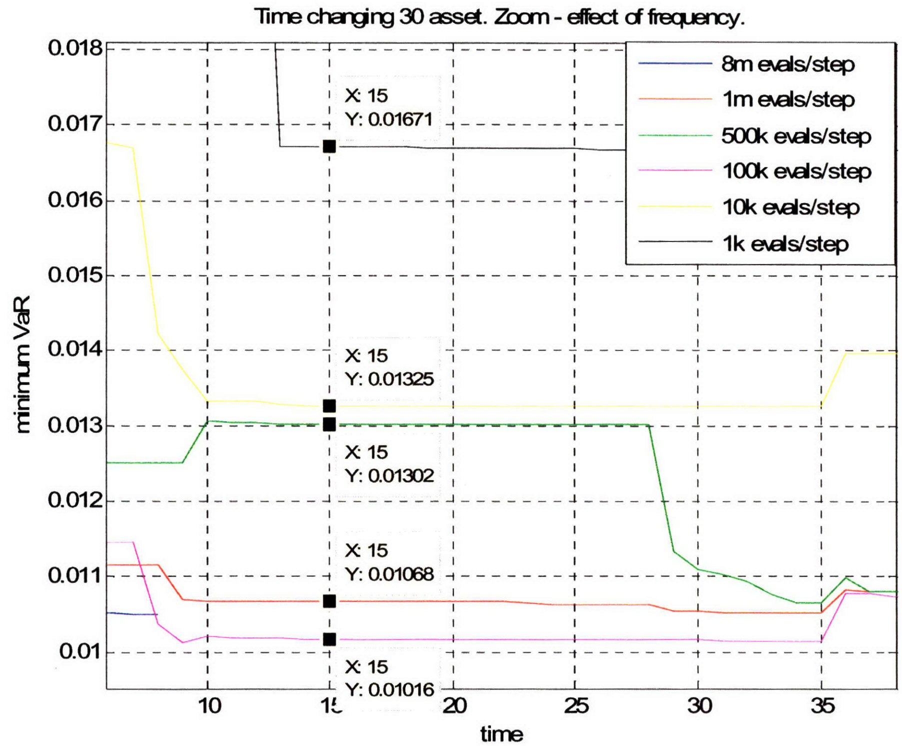


Figure 28. Minimum Value-at-Risk (zoom plot).

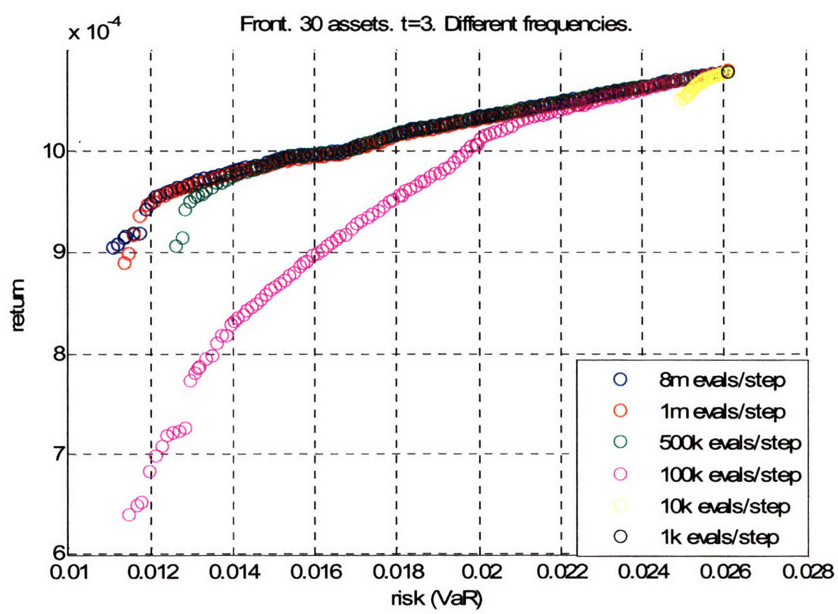


Figure 29. Pareto front (efficient frontier) at *t*=3. The effect of frequency can be seen (notice how much better the obtained front is for 500k evals than, say, 10k evals).

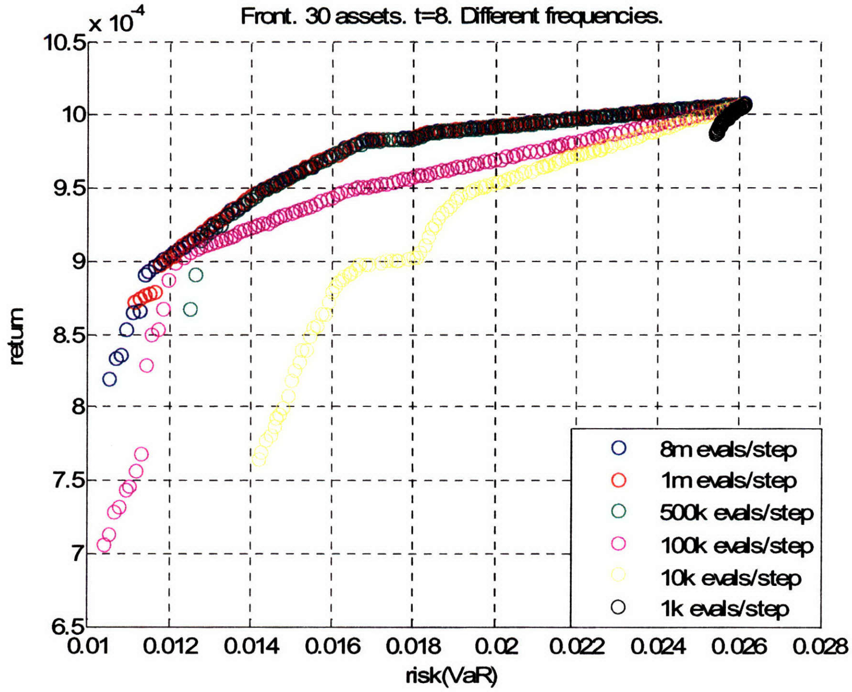


Figure 30. Pareto front (efficient frontier) at *t*=8. The effect of frequency is shown.

These results serve as a demonstration of the ability of D-QMOO's basic version to solve each of the two problems. Sections 5.6 and 5.7 of this chapter deal with trying to find a way of using anticipation (through the Feed-forward Prediction Strategy) and other heuristics in order to improve solution performance.

5.5 Market deployment of the designed portfolios – a real-world verification.

In order to verify whether the portfolios discovered by the algorithm would actually perform well in the real world, we simulated their actual deployment in the market. Each trading day *t*, a set of portfolios are designed. In this section some of these portfolios are deployed in the market the next day (*t*+1) and their return is measured.

The results presented here are from portfolios derived from the solution of the mean-VaR problem (with VaR as the risk measure). Specifically, we use the purple-colored run from Figure 25 at 100k evaluations per time step. The actual, realized returns of the two anchor-point portfolios are measured (minimum risk and maximum return portfolios), together with their expected returns. The time history of these returns can be seen in Figure 32 and Figure 33. We note the following:

- The maximum return design (red) is more volatile. The minimum risk design (blue) is much less volatile as expected. The VaR quantile function usually does not react significantly upon the update of a single asset return if this return is not extremely negative, while the sample mean reacts immediately to asset return updates.
- The actual returns are more volatile than the expected returns of the designed portfolios. This is natural, since the expected returns on which the portfolios are designed are smoothed statistics compared to the actual returns which are single day observations.

- Both portfolios outperform the Dow Jones Index (black line), which contains exactly the same stocks, but with the market allocation. The average daily returns of the minimum risk and the maximum return portfolios are 1.2E-4 and 2.3E-4 respectively, while the DJIA average daily return is -9.5E-6.

The last point serves as a practical demonstration that the portfolios designed using the evolutionary algorithm perform better than the market, since they provide a higher average return than the market portfolio. Also, the trade-off relation between risk and return is confirmed since the maximum return portfolio is also more volatile.

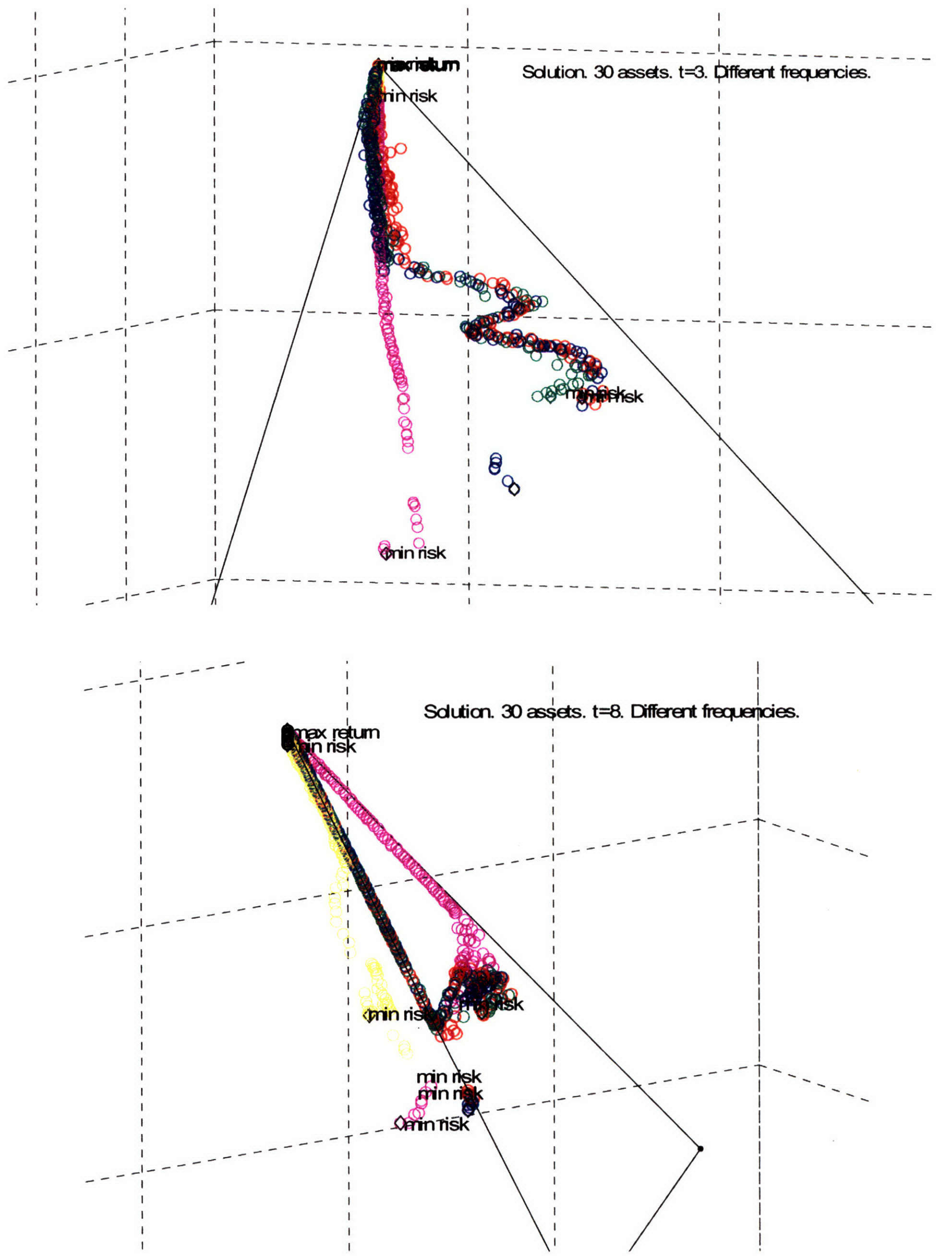


Figure 31. Variable (decision) space plots. The effect of frequency is shown (same color coding as the Pareto plots).

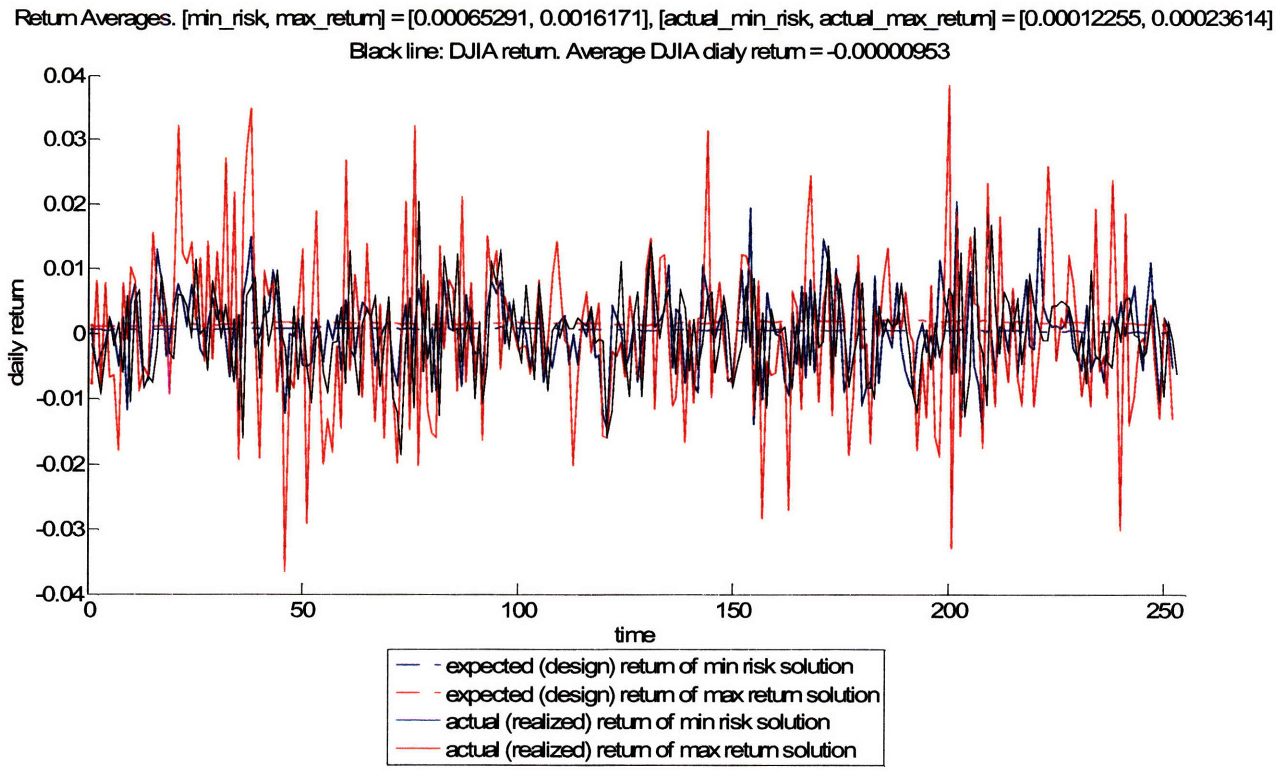


Figure 32. Comparison of the actual performance of the two anchor portfolios (minimum risk and maximum return) discovered by the algorithm, with the performance of the DJIA market portfolio.

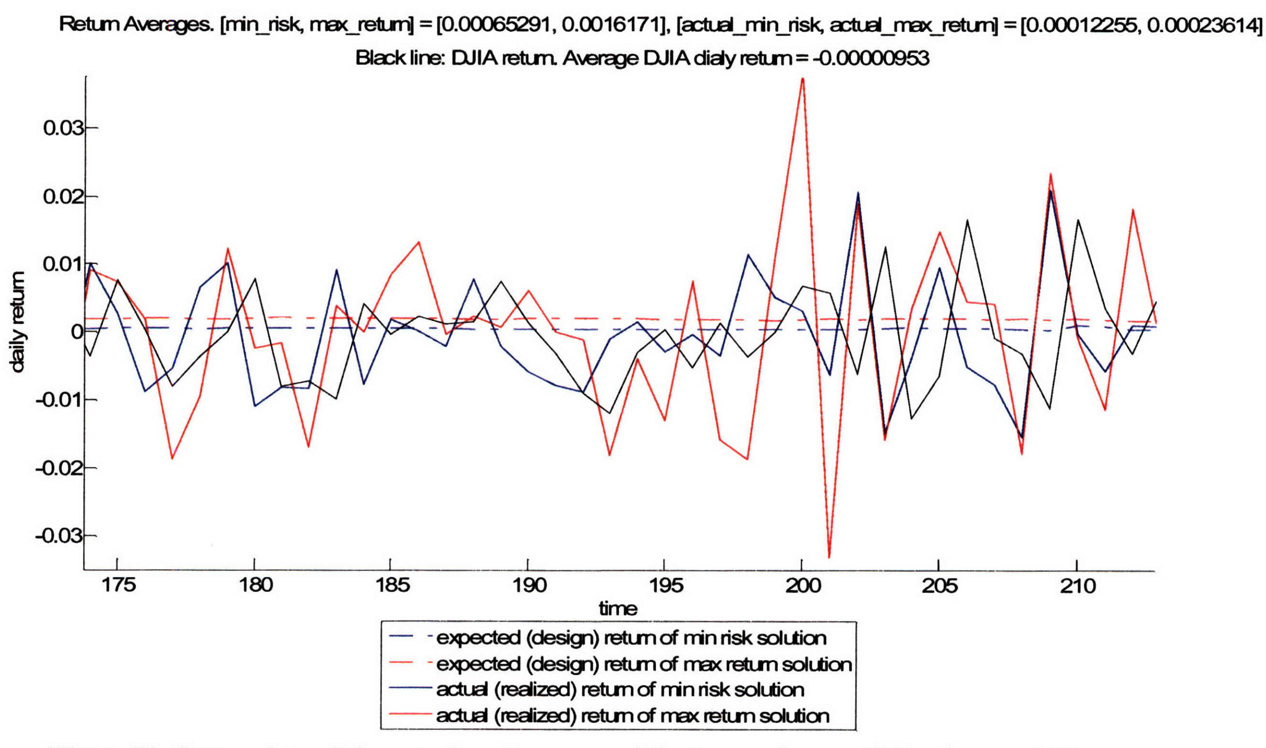


Figure 33. Comparison of the actual performance of the two anchor portfolios (zoom plot).

5.6 Forecasting the location of the minimum Value-at-Risk solution.

In the mean-standard deviation problem, the minimum risk anchor point can be found by solving the quadratic problem using a rigorous method like SQP as it was discussed in section 4.2. This is a straightforward process since quadratic minimization for a single point is not very time consuming and can be done at the beginning of the timestep, as soon as the new market data arrives. This is performed in lieu of an actual forecast (which would have been done in the previous timestep), in order to create an anticipatory individual.

However in the mean-VaR case it is not straightforward to discover the minimum risk solution. Only the evolutionary algorithm itself can be used to globally minimize the non-convex Value-at-Risk objective. Hence finding a suitable estimate for the minimum VaR location is a challenging issue. If an estimate is somehow found however, it can be used to create an anticipatory population and improve performance.

We have identified two approximation concepts for the estimation of the minimum VaR location. Assume that the new time step has just arrived, and the fresh market data (returns for each asset) is just in. Then, the *minimum variance portfolio* can be used as an approximation for *minimum VaR portfolio* (this is one of the heuristics proposed in Table 7).

If we were using a parametric approximation for the portfolio returns (which we are not), the minimum VaR solution would coincide with the minimum variance solution. For example in the case of a normal distribution the 1-percentile VaR is

$$VaR = \mu - 2.32\sigma$$

Hence, finding the solution for minimum σ would also give the solution for minimum VaR in this case.

In our case a direct historical simulation is used to calculate the VaR, and the minimum VaR solution does not coincide with the minimum standard deviation solution[2]. It does, however, provide an approximation. The question here is how good this approximation is. Our initial experiments show that this depends on the case. In general the minimum variance point will not move as abruptly as the minimum VaR point, since variance is an averaged statistic depending on the whole history while VaR is wholly defined by a few single extreme events. Results from the use of this method will be shown later.

It must be repeated here that we go through the trouble of trying to estimate the minimum VaR location because forecasting models that use this location's past time history are hard to use for its prediction. The nature of this point's motion is such that forecasting methods such as autoregressive models have small chances of success. A simple visual inspection of Figure 34 can give us an idea. We can see there that typically an asset stays in roughly the same level for a period of time and then abruptly jumps to a different level, where again it remains for a while, and so on. This sort of motion makes it hard to fit an autoregressive model. Initial experiments using the Feed-forward Prediction Strategy with AR models verified this issue, as will be seen in section 5.6.3; the FPS did not help performance because forecasting was so bad. Hence, heuristic methods such as the minimum variance approximation described above have a bigger chance of success in this problem.

5.6.1 Convergence results for the SP600 data set

The SP600 data set will be used later to study the minimum variance approximation for the

[2] This is also evident in practice; see for example the difference in the variable space plot of the solutions in Table 3 and Table 6 of the previous chapter.

minimum VaR solution location, as described previously, as well as other anticipation methods. Here we present convergence results for this data set, before we continue onto numerical experiments for the approximation heuristics.

In Figure 35 and Figure 36 we can see the time history of asset prices and returns for the SP600 data set. Convergence results for the mean-standard deviation problem are presented in Table 10 and Figures 37 and 38. Convergence results for the mean-standard deviation problem are presented in Table 11 and Figures 39, 40 and 41.

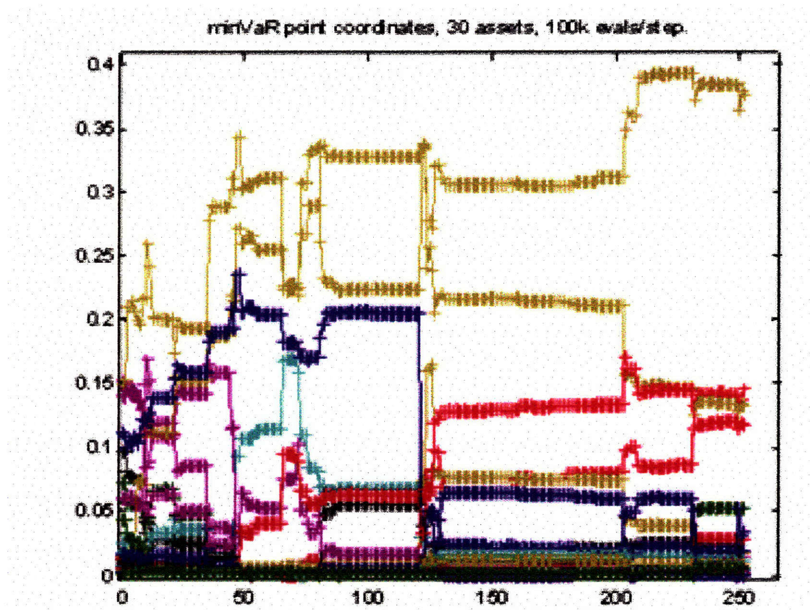


Figure 34. Time history of the coordinates of the minimum VaR point. Each asset's allocation is denoted with a different color.

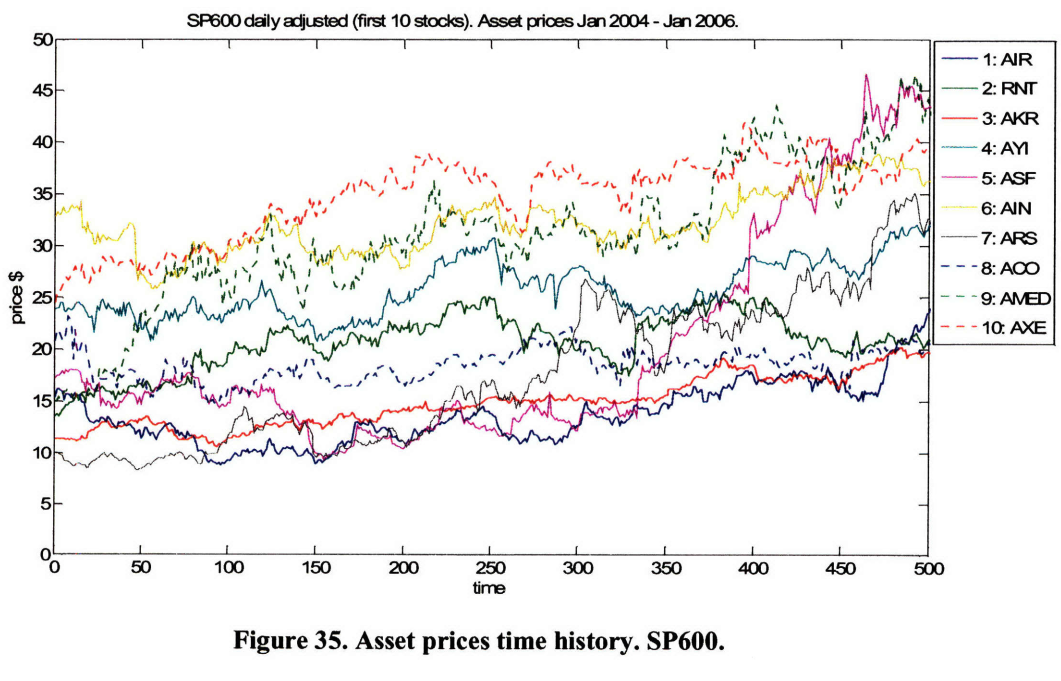


Figure 35. Asset prices time history. SP600.

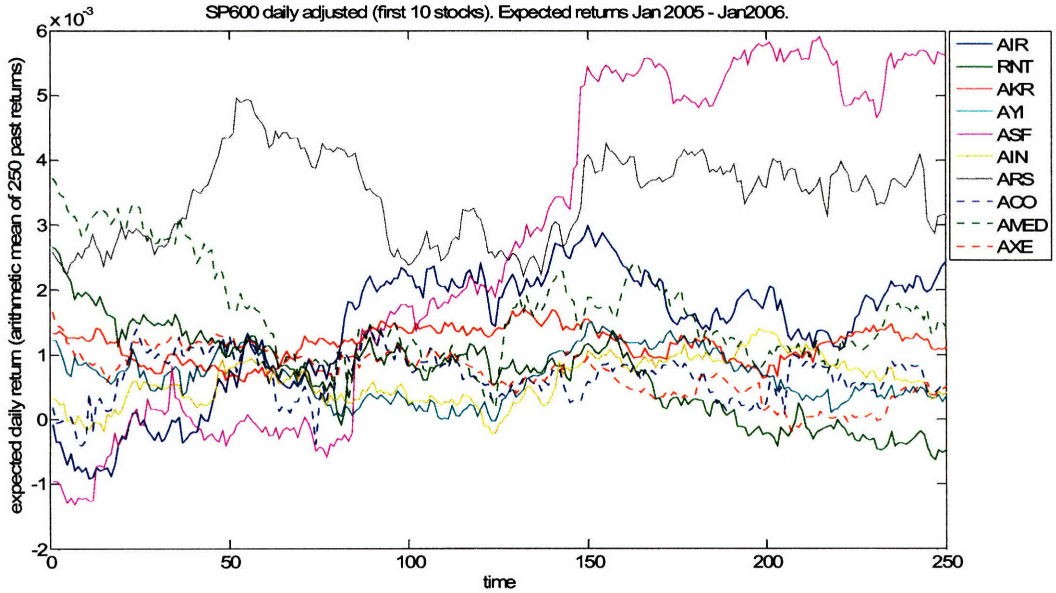


Figure 36. Expected asset returns time history. SP600.

Table 10. Convergence experiment parameters for the mean-standard deviation problem with the SP600 data set.

Risk measure	**Standard deviation.**			
Run length:	**Different lengths.**			
Time-changing:	**Static.**			
Short sales:	**No short sales**			
Data set:	**10_from_SP600_daily_adjusted**			
Anticipation:	**Seeding with maximum return solution at each timestep.**			
expected returns vector:				
-0.000068	0.002662	0.001338	0.001232	-0.000963
0.000322	0.002595	0.000182	0.003721	0.001664

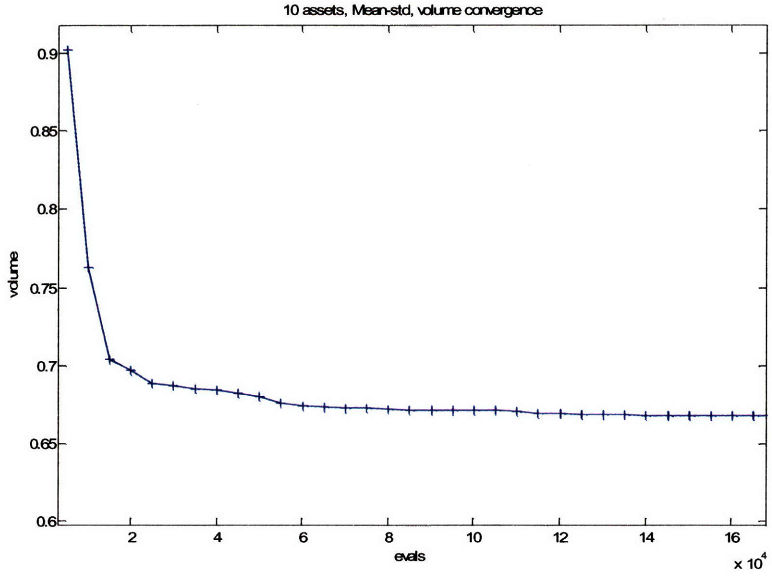


Figure 37. Convergence – non- dominated volume. The algorithm converges at roughly 65k evaluations.

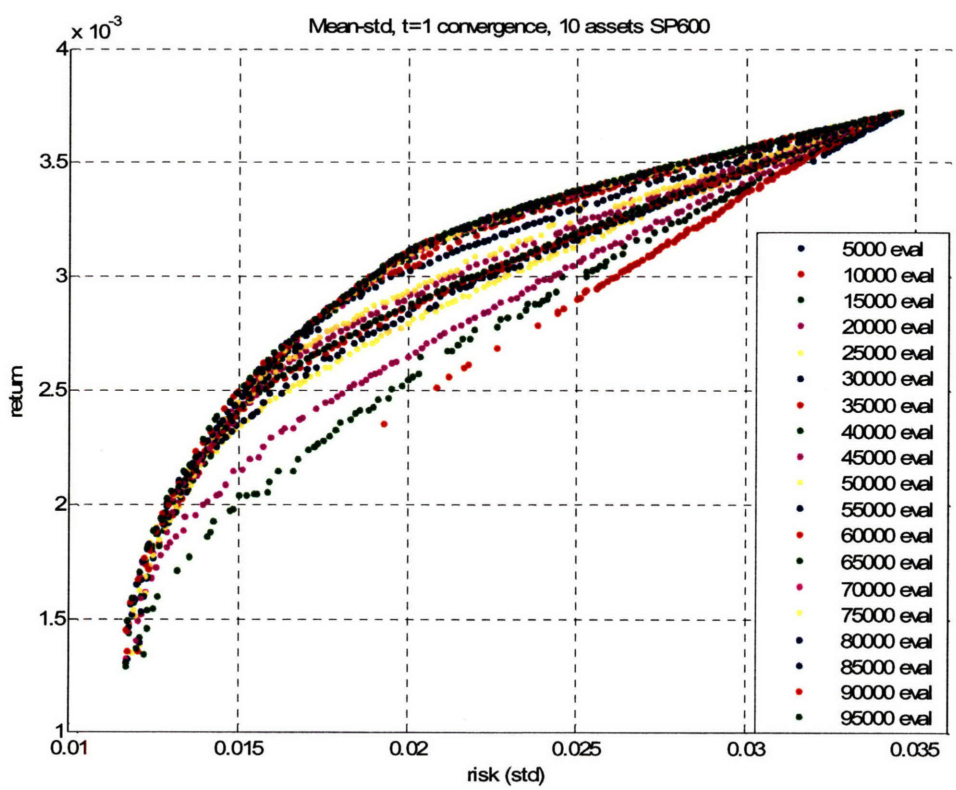


Figure 38. Convergence – non-dominated front.

Table 11. Convergence experiment parameters for the mean-Value-at-Risk problem with the SP600 data set.

Risk measure		**Value-at-Risk (1-percentile).**		
Run length:		**Different lengths.**		
Time-changing:		**Static.**		
Short sales:		**No short sales**		
Data set:		**10_from_SP600_daily_adjusted**		
Anticipation:		**Seeding with maximum return solution at each timestep.**		
expected returns vector:				
-0.000068	0.002662	0.001338	0.001232	-0.000963
0.000322	0.002595	0.000182	0.003721	0.001664

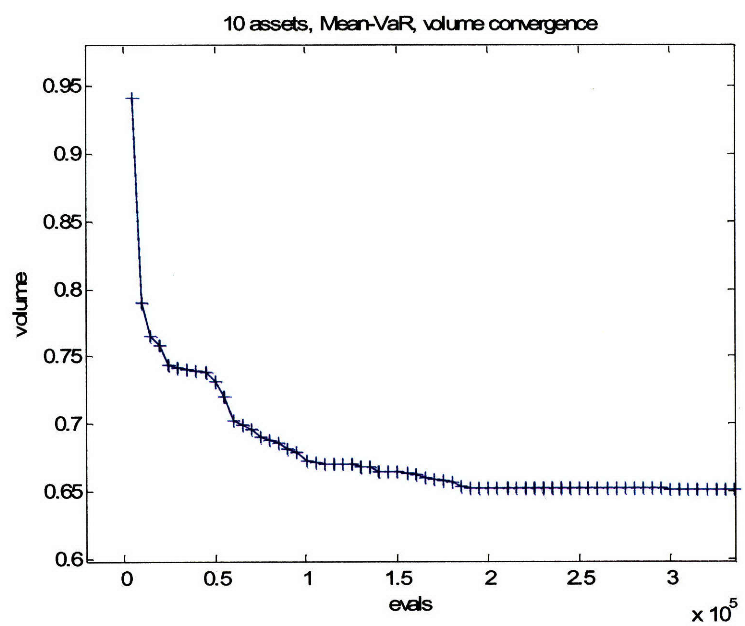


Figure 39. Convergence – dominated volume. In this case convergence happens at about 200k evaluations.

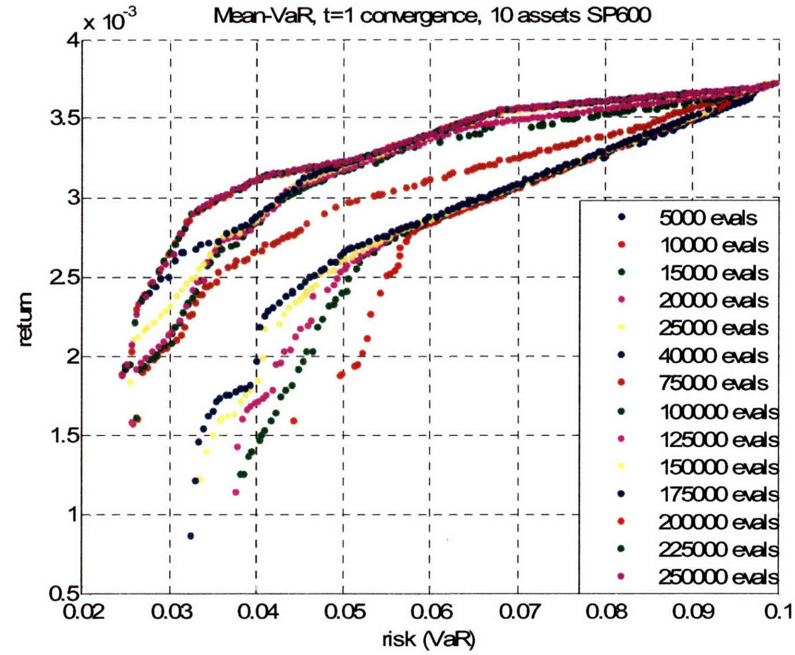


Figure 40. Convergence - non-dominated front.

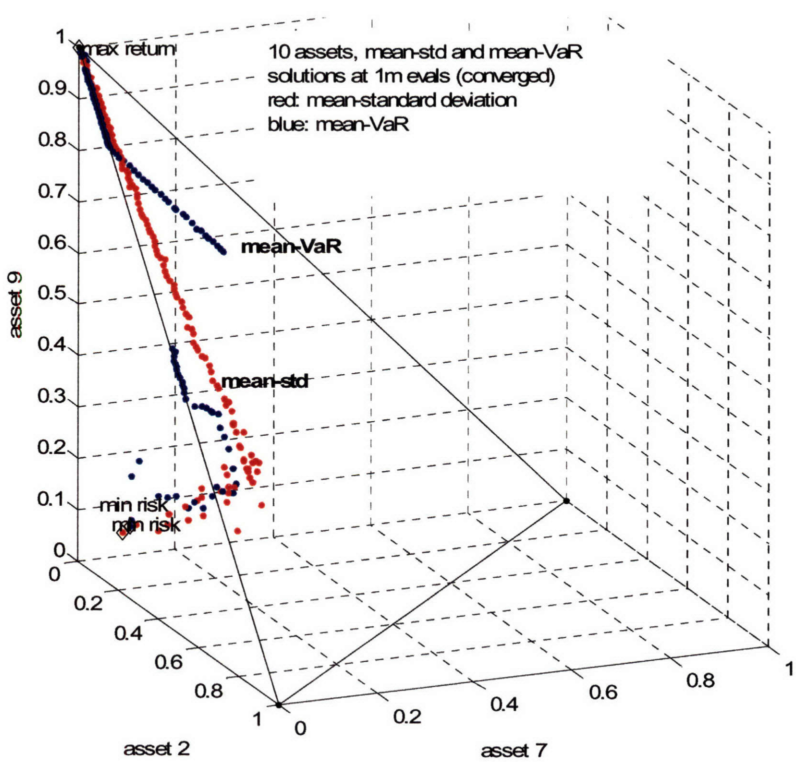


Figure 41. Solution (plotted for three assets [2 7 9]) for both the mean-std and the mean-VaR problems. Asset 9 (AMED) has the highest expected return (see exp. returns time history plot for t=0) and hence the maximum return solution is centered on it.

5.6.2 Results from using the minimum variance solution as a predictor for the minimum VaR solution.

In order to see if the minimum variance solution provides a helpful predictor for the minimum VaR solution within the context of the Feed-forward Prediction Strategy, both static and time-changing experiments were carried out.

In the static optimization problem, using the minimum variance solution as a predictor for the minimum VaR location had a positive effect on the solution performance. In this case performance is expressed by how quickly the solution is discovered and the algorithm converges. This can be seen on the non-dominated volume plot of Figure 42. We can see that the main effect the minimum variance seed has is to place the solution immediately in the neighborhood of the minimum VaR point, accelerating the initial exploration phase.

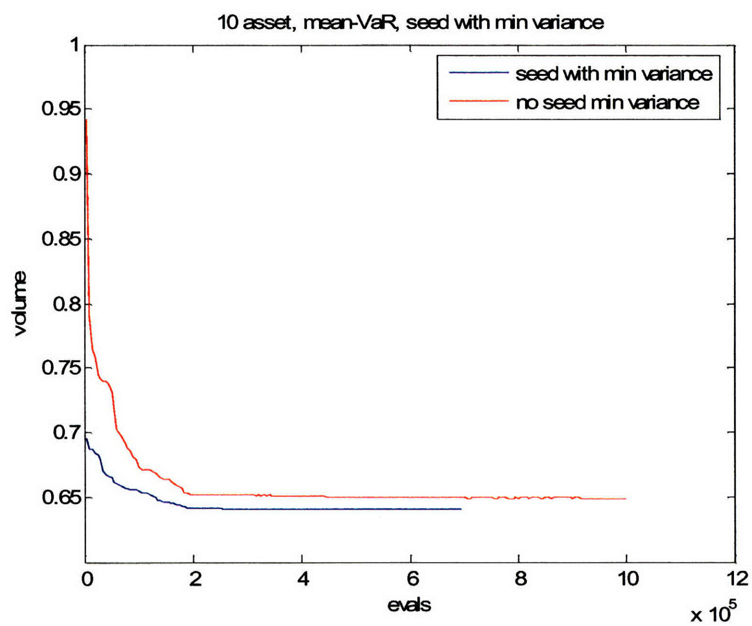


Figure 42. Non-dominated volume for the static mean-VaR optimization problem. It is evident that using the minimum variance approximation (blue curve) helps the algorithm converge faster.

In the time-changing problem, the effect of using the minimum variance point as a predictor in the time-changing mean-VaR problem is evident; it is not, however, very strong. The dominated volume plots in Figure 43 illustrate this effect for an increasing frequency of change. As expected, the FPS benefits performance more in the high frequency case of 1k evaluations per timestep.

In Figure 44 we can see instances of the efficient frontier (Pareto front), with and without the use of the FPS. The effect of the anticipatory population is evident at the minimum risk end of the frontier (bottom left), where in general it produces solutions that dominate the ones without the FPS.

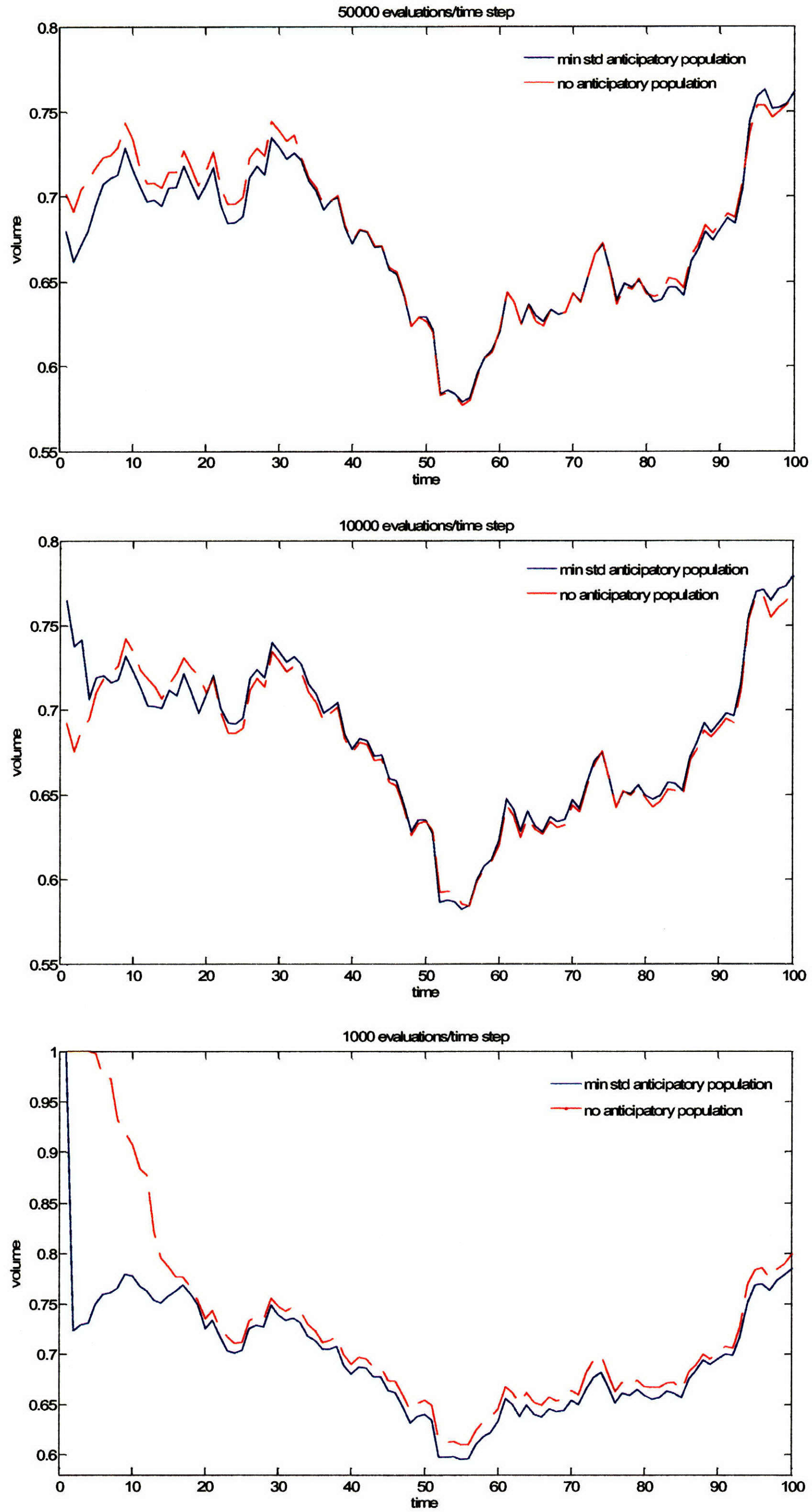


Figure 43. Dominated volume time history. Solution of the time-changing portfolio optimization problem with VaR as risk measure, with and without anticipatory populations. The positive effect of anticipation is evident in the higher frequency of 1000 evaluations per time step.

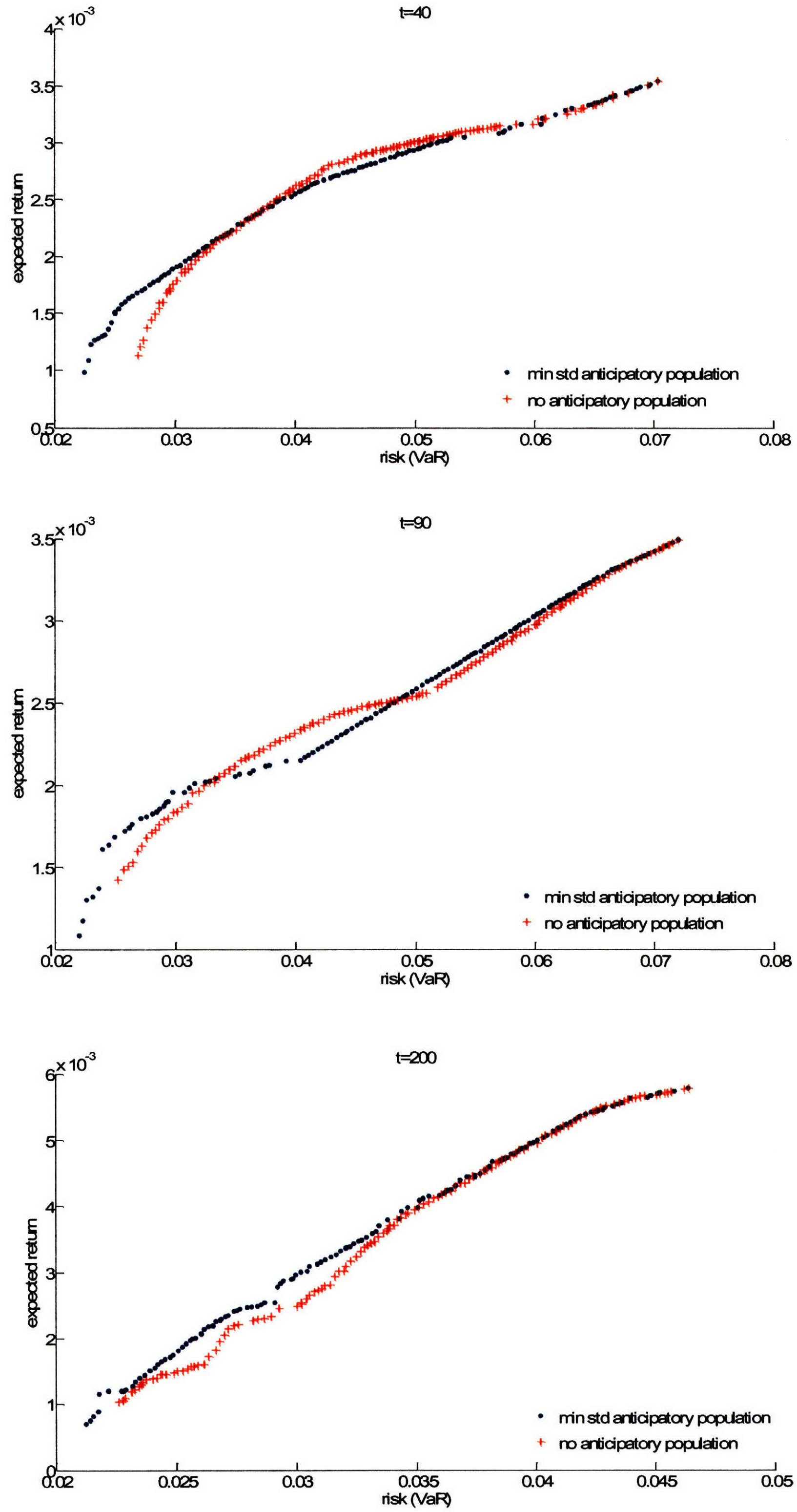


Figure 44. Efficient (Pareto) frontier approximations at various time steps, with and without minimum risk anticipatory populations. The effect of the minimum risk anchor point anticipatory population is evident.

5.6.3 Autoregressive models as predictors of the optimal solution's motion

As noted earlier the solutions to the portfolio optimization problem move in time in a way that makes it hard to successfully use a forecasting model. The use of autoregressive models with the FPS did not manage to significantly improve performance. In the numerical experiments carried out, the forecast for a solution's location was so bad that it practically followed the solution instead of leading it. As an example, in Figure 45 we can see the best discovered solution coordinates (asset allocations) and their forecasted motion, for the closest-to-ideal (CTI) point. The CTI point is an intermediate point along the Pareto front that is sometimes used with the FPS (Hatzakis, Wallace 2006b). Only assets 2, 7 and 9 are shown for clarity's sake. It is obvious that the forecast does not predict changes, it rather just follows the solution's motion.

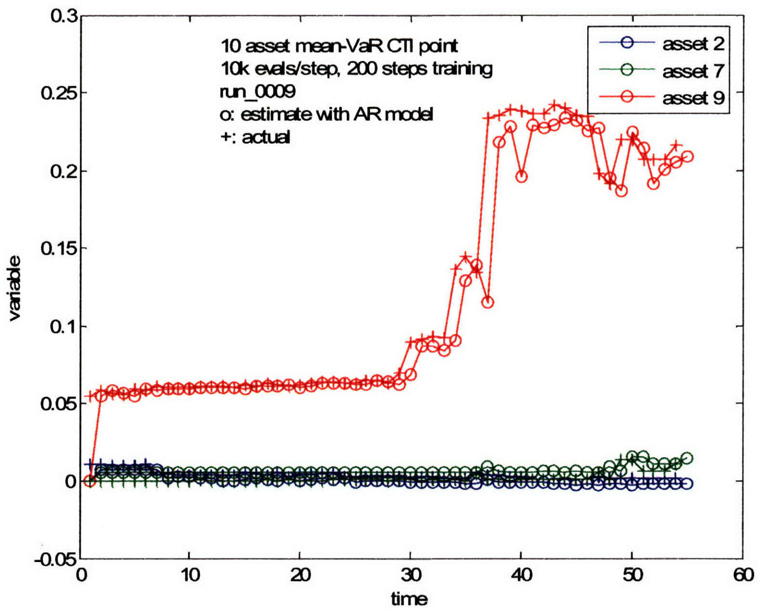


Figure 45. AR forecast and actual motion. The forecast is not successful – it only follows the solution's motion.

5.7 Local search.

Anticipatory individuals, no matter with which method they have been created, are often close to the actual solution for the next time step. In this paragraph a local search (local hill climbing) technique is applied on the anticipatory individuals.

The rationale is shown in the sketch of Figure 46. Anticipatory individuals are usually one, two or three. Often they are at a large distance in the design space from the existing population. Hence they might produce a better result by searching on their own around their neighborhood for better solutions, rather than being mated with distant unrelated solutions belonging to the previous time step's front and trying to produce better individuals by crossover (recall section 3.1 for a discussion on the function of crossover operators). The technique proposed here is to create the anticipatory individuals and then have them perform a number of local hill climbing steps, before inserting them into the population and allowing the evolutionary algorithm to continue solving.

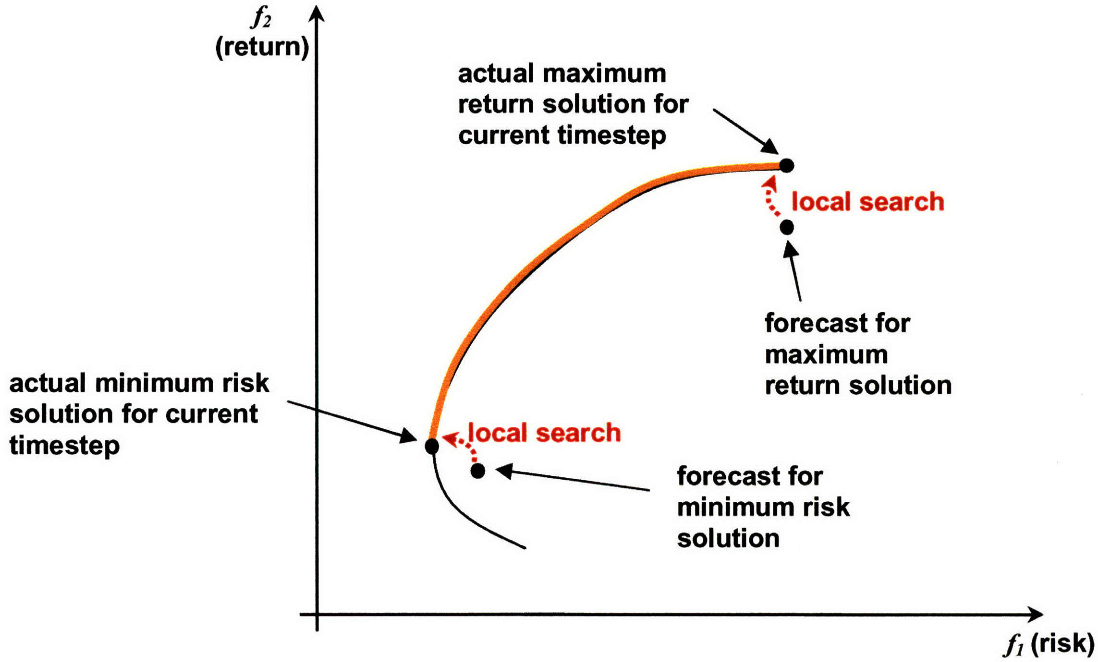


Figure 46. Local search concept.

A 1+1 Evolution Strategy (ES) is used as a local search mechanism[3]. Various local search options arise since the Evolution Strategy has a number of parameters to be controlled. The most basic ones are the number of ES steps to be performed and the local search radius. The minimum variance seeds, as described in the previous paragraph, are used as anticipatory individuals for the minimum risk point.

In the non-dominated volume plots of Figure 47 and Figure 48 we can see the performance of various local search options. In general the local search seems to have a positive effect on performance, compared to using no prediction (FPS) or using prediction without local search. A local search radius of 0.02 had the best performance. Also, increasing the number of steps taken by the local search to 30 had a slightly positive effect, compared to the 10 steps performed normally.

[3] Other, more sophisticated local search techniques can also be used, which are tailored to the specific problem – see for example (Schlottmann, Seese 2004).

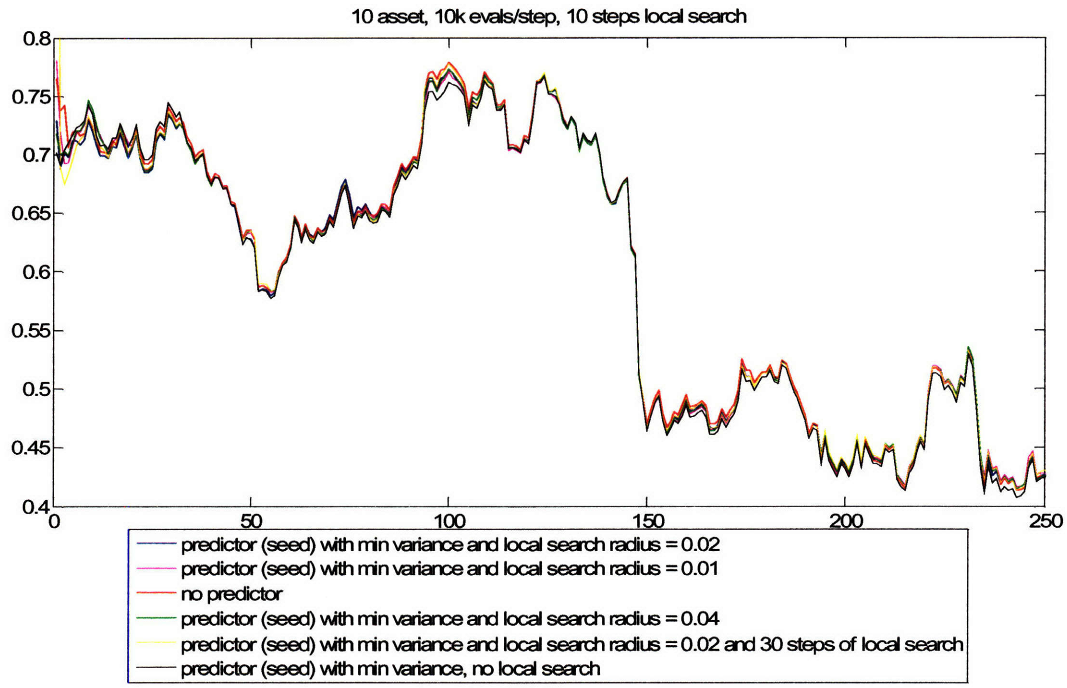


Figure 47. Non-dominated volume time history for various algorithmic versions (with and without local search).

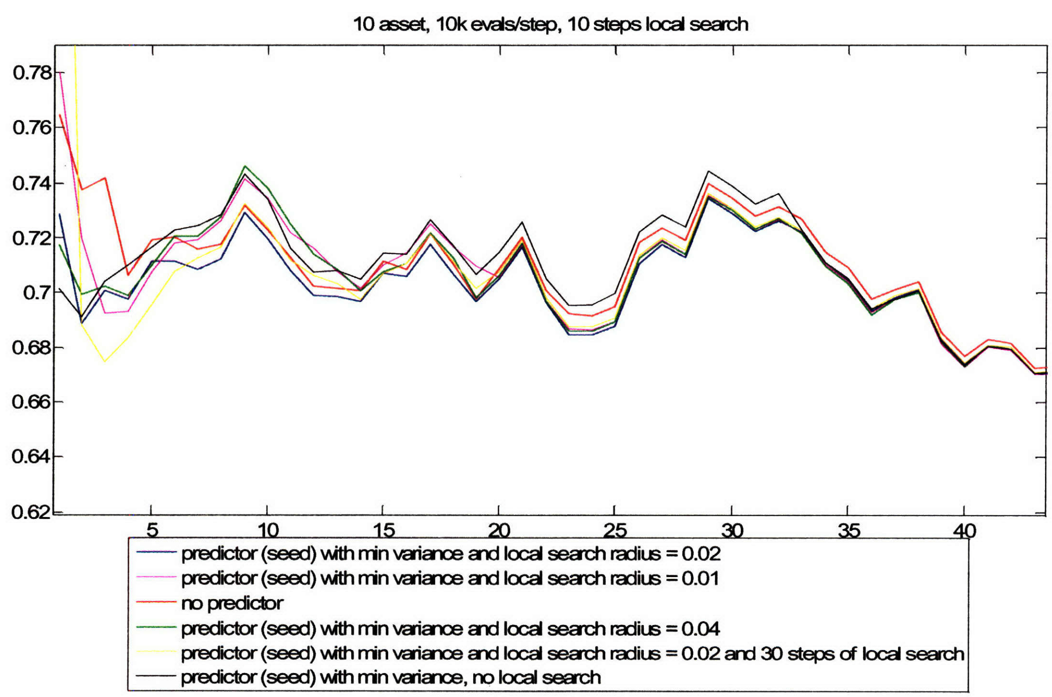


Figure 48. Non-dominated volume time history (zoom plot).

5.8 Conclusion

The initial numerical experiments of this chapter demonstrate the ability of D-QMOO's basic form to solve the time-changing portfolio optimization problem. Subsequently, the lack of predictability that characterizes portfolio optimization inspires the development of new heuristics for the creation of anticipatory populations. While conventional forecasting with autoregressive models does not help performance, estimating the location of the minimum VaR solution with the minimum variance solution and using local search for the anticipatory individuals produce more promising results and increase the algorithm's efficiency.

6 Conclusions and Future Directions

6.1 Conclusions

Overall, the potential for solving the multi-objective time-changing asset allocation problem using evolutionary algorithms has been demonstrated. The main advantages of using evolutionary optimization for this application lie with the inherent ability of EAs to solve, first, multi-objective, and second, discontinuous, non-convex and non-differentiable problems. An initial verification of D-QMOO's performance is given by creating a series of portfolios for a one-year time period that outperform the average market return.

Solving the Value-at-Risk problem with EAs

The mean-Value-at-Risk problem in its sampling form[1] has such a non-convex and often discontinuous nature (Pflug 2000, Gaivoronski, Pflug 2005). It naturally lends itself to an evolutionary optimization approach. At the same time VaR is an established risk measure which might be preferred over other risk metrics such as the standard deviation. This demonstrates the applicability of EAs to the mean-VaR problem alongside other proposed methods such as the one in (Gaivoronski, Pflug 2005).

Nature of the time-changing portfolio optimization problem and heuristics developed

As a time-changing problem, the discovery of efficient portfolios is characterized by a relative unpredictability in the optimal solution's motion in the variable space. This creates a challenge for evolutionary time-changing optimization methodologies such as the Feed-forward Prediction Strategy (FPS - Hatzakis 2007) which exploit predictability. This challenge was addressed in this work through the development of heuristics for the anticipatory population that do not require the use of forecasting. These heuristics are based on the use of fast, single-objective optimization methods for the discovery of the non-dominated front's extreme solutions utilizing data from the current timestep. Other useful techniques that were developed in the context of this work involved the use of a local search (in the form of a 1+1 Evolution Strategy) for the anticipatory

[1] As opposed to its parametric form.

individuals. Such methods proved to perform well in the mean-VaR time-changing problem, demonstrating that in the context of the FPS, *anticipation* is a stronger concept than *forecasting.*

6.2 Future directions

Comparative testing with actual portfolios and practical application

A natural next step for the verification and assessment of the methodologies proposed in this work would be to compare them against the performance of professionally created portfolios (such as mutual funds).

It is important to keep in mind that the methodologies proposed here are not intended to be completely self-sufficient, 'automatic' portfolio selection techniques. They are meant to provide what is hopefully a useful tool to the portfolio manager, to be used alongside other methods and human skills. From this viewpoint, an interesting task would be to use the portfolio optimization version of D-QMOO in order to assist experts in a real-world setting. The feedback derived from such an experiment would be invaluable in improving the algorithm.

Forecasting the optimal solution's motion for the mean-VaR problem

The location of the mean-VaR optimal solutions is very hard to forecast, as discussed earlier in the text. However, an opportunity may exist to derive accurate enough one-step-ahead forecasts in order to create anticipatory populations, instead of using heuristics such as the ones developed in this work. Such forecasting could be based on the nature of the historical sample VaR as a risk measure. Specifically, it is interesting to note that VaR depends on the $\lceil \alpha t \rceil^{th}$ worse return in the portfolio's history. Hence when the new market data arrives, it is relatively straightforward to deduce whether this data will affect the VaR (comparing with the $\lceil \alpha t \rceil$ – usually 3 – worse returns). What remains is to develop a way of inferring some information on the expected motion of the minimum VaR solution.

Including switching/trading cost as a third objective

In most practical optimization problems changing from one solution to a different one involves some kind of cost. In the portfolio optimization problem this cost relates, for example, to trading fees. An integrated portfolio selection environment could include a way of taking this cost into consideration, for example by including it as a third objective.

Including fundamental information and investors' views in the portfolio selection process

The statistical measures of risk and return used in this work are far from self-sufficient. Simply speaking, there is no guarantee that the past history of an asset is a reliable index of its future performance. For this reason, a rounded portfolio selection process may also take in account external information, fundamental economic principles and even the investor's instinct on different assets' performance. It follows that the selection tool presented in this work can be enhanced by methodologies which take these factors in account and accordingly affect the asset allocation.

7 Bibliography

Barro, D. & Canestrelli, E. 2005, "Dynamic Portfolio Optimization: Time decomposition using the Maximum Principle with a scenario approach", *European Journal of Operational Research,* vol. 163, pp. 217-229.

Basel Committee on Banking Supervision 2001, , *The New Basel Capital Accord* [Homepage of Bank for International Settlements], [Online]. Available: http://www.bis.org/.

Beder, T.S. 1995, "VAR: Seductive but Dangerous", *Financial Analysts Journal,* vol. 51, no. 5, pp. 12-24.

Black, F. & Litterman, R. 1992, "Global Portfolio Optimization", *Financial Analysts Journal,* vol. 48, no. 5.

Branke, J. 2002, *Evolutionary Optimization in Dynamic Environments,* 1st edn, Kluwer Academic Publishers, Boston, MA.

Coello Coello, C. & Lamont, G. 2004, "Applications of Multi-Objective Evolutionary Algorithms" in World Scientific, pp. 3-5.

Deb, K. 2001, *Multi-Objective Optimization using Evolutionary Algorithms,* Wiley.

Ehrgott, M., Klamroth, K. & Schwelm, C. 2004, "An MCDM approach to portfolio optimization", *European Journal of Operational Research,* vol. 155, pp. 752-770.

Gaivoronski, A. & Pflug, G. 2005, "Value-at-Risk in Portfolio Optimization: Properties and Computational Approach", *Journal of Risk,* vol. 7, no. 2, pp. 1-31.

Hatzakis, I. 2007, *Multi-Objective Evolutionary Optimization in Time-Changing Environments*, PhD Thesis, MIT.

Hatzakis, I. & Wallace, D. 2006a, "Dynamic Multi-Objective Optimization with Evolutionary Algorithms: A Forward-Looking Approach", *2006 Conference on Genetic and Evolutionary Computation*, Seattle, WA.

Hatzakis, I. & Wallace, D. 2006b, "Topology of Anticipatory Populations for Evolutionary Dynamic Multi-Objective Optimization", *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Portsmouth, VA.

Hendricks, D. 1996, "Evaluation of Value-at-Risk Models Using Historical Data", *FRBNY Economic Policy Review,* no. April, pp. 39-70.

Hirschberger, M., Qi, Y. & Steuer, R. 2004, *Quadratic Parametric Programming for Portfolio Selection with Random Problem Generation and Computational Experience*, Department of Banking and Finance, University of Georgia, Athens.

Jorion, P. 1997, *Value at Risk,* 1st edn, Academic Press, San Diego.

Leyland, G. 2002, *Multi-Objective Optimization Applied to Industrial Energy Problems*, EPFL.

Litterman, R. 2003, *Modern Investment Management,* Wiley.

Markowitz, H. 2000, *Mean-Variance Analysis in Portfolio Choice and Capital Markets,* Fabozzi, New Hope, PA.

Markowitz, H. 1952, "Portfolio Selection", *Journal of Finance,* vol. 7, no. 1, pp. 77-91.

Merton, R.C. 1969, "Lifetime Portfolio Selection Under Uncertainty: The Continuous Time Case", *The Review of Economics and Statistics,* , pp. 247-257.

Pang, T. 2004, "Portfolio Optimization Models on Infinite Time Horizon", *Journal of Optimization Theory and Applications,* vol. 122, no. 3, pp. 573-597.

Papalambros, P. & Wilde, D. 2000, *Principles of Optimal Design,* 2nd edn, Cambridge University Press.

Pareto, V. 1896, *Cours d' Economie Politique,* F. Rouge, Lausanne.

Pflug, G. 2000, "Some remarks on the Value-at-Risk and the Conditional Value-at-Risk" in *Probabilistic constrained optimization*, ed. S. Uryasev, Kluwer, Dordrecht, pp. 272-281.

Schlottmann, F. & Seese, D. 2005, "Financial applications of multi-objective evolutionary algorithms: Recent developments and future research" in *Handbook on Applications of Multi-Objective Evolutionary Algorithms*, eds. Coello Coello, C. A. & G. Lamont, World Scientific, Singapore.

Schlottmann, F. & Seese, D. 2004, "A hybrid heuristic approach to discrete portfolio optimization" in *Computational Statistics and Data Analysis 47*, pp. 379-399.

Stambaugh, F. 1996, "Risk and Value-at-Risk", *European Management Journal,* vol. 14, no. 6, pp. 612-621.

Wronski, J. 2005, *A design tool architecture for the rapid evaluation of product design tradeoffs in an Internet-based system modeling environment*, SM Thesis, MIT.

Zitzler, E., Laumanns, M. & Thiele, L. 2001, *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*, Swiss Federal Institute of Technology (ETH), Zurich.

Zitzler, E. & Thiele, L. 1999, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach", *IEEE Transactions on Evolutionary Computation,* vol. 3, no. 4, pp. 257-271.