Available online at www.sciencedirect.com
SCIENCE DIRECT®
European Journal of Operational Research 155 (2004) 752–770
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/dsw

Decision Aiding

An MCDM approach to portfolio optimization

Matthias Ehrgott [a,*], Kathrin Klamroth [b], Christian Schwehm [c]

[a] Department of Engineering Science, University of Auckland, Private Bag 92029, Auckland 92019, New Zealand
[b] Department of Statistics and Operations Research, Institute of Mathematics, University of Copenhagen, 2100 Copenhagen Ø, Denmark
[c] DWS Investment GmbH, Grünbergweg 113–115, 60612 Frankfurt, am Main, Germany

Received 6 August 2001; accepted 1 November 2002

Abstract

We propose a model for portfolio optimization extending the Markowitz mean–variance model. Based on cooperation with Standard and Poor's we use five specific objectives related to risk and return and allow consideration of individual preferences through the construction of decision-maker specific utility functions and an additive global utility function. Numerical results using customized local search, simulated annealing, tabu search and genetic algorithm heuristics show that problems of practically relevant size can be solved quickly.

Keywords: Portfolio optimization; Multicriteria decision making; Utility function; Metaheuristics

1. Portfolio optimization

The *Markowitz covariance model* (Markowitz, 1952, 1959), the classical approach to portfolio optimization, is based on two conflicting optimization criteria: On one hand, the risk of a portfolio, represented by its variance, is to be minimized, while on the other hand the expected return of the portfolio is to be maximized. This naturally leads to the following bicriteria formulation of the problem:

$$\begin{aligned} \max \quad & \sum_{i=1}^{M} \mu_i x_i, \\ \min \quad & \sum_{i=1}^{M}\sum_{j=1}^{M} \sigma_{ij} x_i x_j, \\ \text{s.t.} \quad & \sum_{i=1}^{M} x_i = 1, \\ & x_i \geqslant 0, \quad i = 1, \ldots, M. \end{aligned} \tag{1}$$

* Corresponding author. Tel.: +64-9-373-7599x82421; fax: +64-9-373-7468.
E-mail addresses: m.ehrgott@auckland.ac.nz (M. Ehrgott), klamroth@math.ku.dk (K. Klamroth), christian.schwehm@dws.de (C. Schwehm).

0377-2217/$ - see front matter
doi:10.1016/S0377-2217(02)00881-0

Here, M denotes the number of available assets; x_i represents the investment portion in asset $i \in \{1,\ldots,M\}$; where $x = [x_1,\ldots,x_M]^T \in \mathbb{R}^M$ is the M-dimensional solution vector; μ_i denotes the expected return of asset $i \in \{1,\ldots,M\}$; σ_{ij} denotes the covariance between the returns of assets $i,j \in \{1,\ldots,M\}$, and $\sigma = (\sigma_{ij})_{i=1,\ldots,M;j=1,\ldots,M}$ denotes the corresponding $M \times M$ covariance matrix.

To simplify further notation, we will often replace the minimization of the variance in (1) by the equivalent maximization of the negative variance, using the equivalence

$$\min x^T\sigma x = -\max(-x^T\sigma x).$$

Note that the first objective, $f_1(x) = \sum_{i=1}^{M} \mu_i x_i$, is a linear function while the second objective, $f_2(x) = -x^T\sigma x$, is a quadratic function of x.

In the context of multicriteria programming, solving (1) is understood as generating its Pareto optimal (efficient) solutions. If $X = \{x \in \mathbb{R}^M : \sum_{i=1}^{M} x_i = 1, x_i \geqslant 0\ \forall i = 1,\ldots,M\}$ denotes the set of feasible solutions of (1), a portfolio $x_e \in X$ is said to be a Pareto optimal solution (an efficient portfolio) if there is no other solution $x \in X$ such that $f(x) < f(x_e)$, i.e. $f_1(x) < f_1(x_e)$ and $f_2(x) \leqslant f_2(x_e)$ or $f_1(x) \leqslant f_1(x_e)$ and $f_2(x) < f_2(x_e)$. In other words, Pareto optimal solutions are those solutions for which none of the criteria can be improved without deterioration of the other criterion.

Let $\mathscr{X}_e$ denote the set of efficient solutions of (1) and let $\mathscr{Y}_e$ denote the image of $\mathscr{X}_e$ in the objective space, that is $\mathscr{Y}_e = f(\mathscr{X}_e)$, where $f = [f_1, f_2]^T$. $\mathscr{Y}_e$ is referred to as the set of efficient points or the efficient frontier of (1).

Fig. 1 illustrates the shape of the efficient frontier of (1) for an example problem based on data for 40 investment funds (Standard and Poor's Funds Services Database (1999), see also Schwehm (2000)). The same data was used in the small real case studied in Section 6. Objective function f_1 represents the expected return and objective function f_2 models the variance of the portfolio.

The tool proposed in Markowitz (1959) to find efficient portfolios is the critical line algorithm. Basically, it solves a parametric quadratic programming reformulation

$$\min x^T\sigma x - \lambda\mu x$$

of (1) using Karush–Kuhn–Tucker optimality conditions. The main drawback of that original method was the time needed to compute the covariance matrix from historical data and the difficulty of solving the large scale quadratic programming problems. Of course, with modern soft- and hardware this is not an issue anymore.

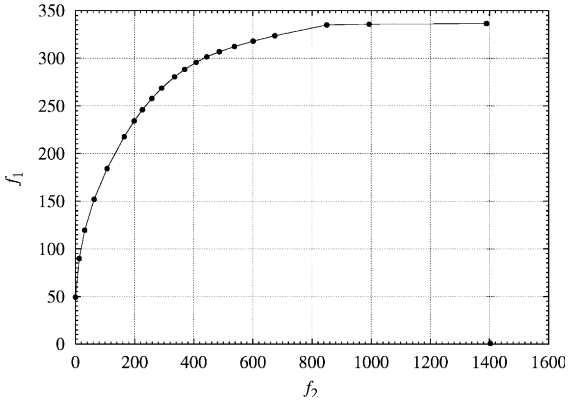


Fig. 1. Approximation of the non-dominated frontier of (1) for data consisting of 40 different investment funds. The piecewise linear approximation was constructed using the norm-based algorithm of Schandl et al. (2001).

Nevertheless, various authors have proposed models which do not imply these problems. Konno (1990) formulated a piecewise risk function to replace the covariance, thus reducing the problem to a linear programming one. He could prove that his model is equivalent to the Markowitz model, when the vector of returns is multivariate normally distributed. Markowitz et al. (1994) later described a method which avoids actual computation of the covariance matrix, and Morita et al. (1989) applied a stochastic linear knapsack model to the portfolio selection model.

In recent years, criticism of the basic model has been increasing because of its disregard for individual investors' preferences. Konno (1990) observed that most investors do not actually buy efficient portfolios, but rather those behind the efficient frontier. Ballestero and Romero (1996) first proposed a compromise programming model for an "average" investor, which was modified to approximate the optimum portfolio of an individual investor (Ballestero, 1998). A different approach was described in Arthur and Ghandforoush (1987), who propose the use of objective and subjective measures for assets. Their idea leads to a simple linear programming model. Hallerbach and Spronk (1997) argue that most models do not incorporate the multidimensional nature of the problem and outline a framework for such a view on portfolio management. For further references on the use of optimization models for portfolio selection the reader is referred to Pardalos et al. (1994).

The rest of the paper is organized as follows. In Section 2 we present an objective hierarchy and formulate a multicriteria optimization model which uses five objective functions. In Section 3 we show how to construct decision maker specific utility functions using interpolation methods. The final model is described in Section 4 as a mixed integer programming model. Section 5 gives an overview of the algorithms implemented. Finally, in Section 6 we present numerical results on some test problems.

2. Multicriteria model and objective hierarchy

As discussed in Section 1 it is often found in portfolio optimization that investors prefer portfolios that lie behind the non-dominated frontier of the Markowitz model (1) even though they are dominated by other portfolios with respect to the two criteria expected return and risk. This observation can be explained by the fact that not all the relevant information for an investment decision can be captured in terms of explicit return and risk (see, for example, Hallerbach and Spronk, 1997). By considering additional and/or alternative decision criteria, a portfolio that is dominated with respect to expected return and risk may make up for the deficit in these two criteria by a very good performance in one or several other criteria and thus be non-dominated in a multicriteria setting. Moreover, investors may differ significantly in their perception of the relative importance of different attributes like dividends, financial stability or future growth expectations.

As a result, a multicriteria model based on more than two objective functions allows for a higher flexibility in modeling the objectives of investors, and, combined with an appropriate utility approach, is likely to lead to better representations of their preferences. Fig. 2 shows an example of an *objective hierarchy* extending the classical Markowitz model in the sense that the two classical criteria risk and return are replaced by five more specific objective functions.

The model was found after several discussions with investors as well as analysts from Standard and Poor's Funds Services GmbH, Germany (see Schwehm, 2000). It was indicated that the expected return as used in the Markowitz model should be broken down into the criteria *12-month performance*, *3-year performance* and *annual dividend* in order to improve the possibilities of the individual investor to articulate subjective preferences. This gain of flexibility seems to outweigh the additional time required for the consideration of further objectives. The fourth objective, the *Standard and Poor's star ranking*, describes to what extent an investment fund follows a specific market index and is applied particularly in the case that a portfolio consists exclusively of investment funds. It evaluates the out- or under-performance divided by the

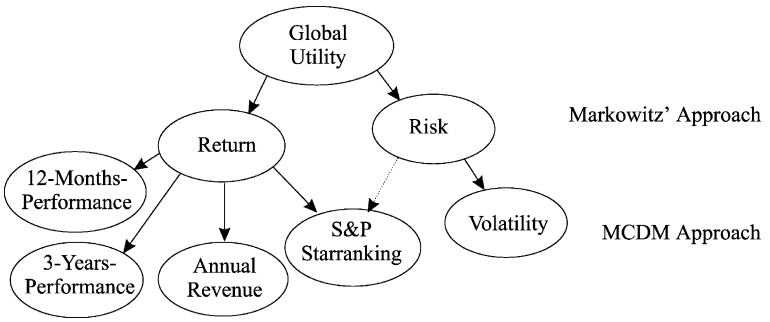


Fig. 2. Example for an objective hierarchy based on the Markowitz model. The arcs indicate relations between the different criteria.

tracking error over three years and rewards funds that closely follow the market index. The fifth attribute, the *12-month volatility*, is used as a measure of the risk of a portfolio.

The resulting model with the five objective functions $f_1, \ldots, f_5$ as specified above can be formulated as follows:

$$\begin{aligned} \max \quad & [f_1(x), \ldots, f_5(x)]^T, \\ \text{s.t.} \quad & \sum_{i=1}^{M} x_i = 1, \\ & x_i \geqslant 0, \quad i = 1, \ldots, M. \end{aligned} \tag{2}$$

Based on the available data, the following implementations of the five objective functions were used (see, for example, Gouriéroux (1997), for a more detailed discussion of alternative implementations).

Let $p_{t,i}$ denote the price (value) of asset i in period t, for $i = 1, \ldots, M$ and $t = 1, \ldots, T$, where T denotes the present.

- **12-month performance**

The 12-month performance r_i^{12} of an individual asset $i \in \{1, \ldots, M\}$ measures the relative change of the price of the asset over the last twelve months (in percent) and is therefore a measure for the short term expected return. In particular,

$$r_i^{12} = \frac{p_{T,i} - p_{T-1,i}}{p_{T-1,i}}.$$

We assume that the returns follow some statistical distribution, so that the expected return (12-month performance of a portfolio) can be obtained as a weighted sum of the expected return of the individual assets in the portfolio. Thus we obtain objective function f_1 as

$$f_1(x) = \sum_{i=1}^{M} r_i^{12} x_i.$$

- **3-year performance**

The long term performance (expected return) r_i^{36} of an asset is calculated, similarly to the 12-month performance, as

$$r_i^{36} = \frac{p_{T,i} - p_{T-3,i}}{p_{T-3,i}}.$$

Consequently, the 3-year performance of a portfolio is given by

$$f_2(x) = \sum_{i=1}^{M} r_i^{36} x_i,$$

again assuming a statistical distribution of the r_i^{36} values.

- **Annual dividend**

Objective function f_3 represents the relative annual dividend of a portfolio. It is calculated as the weighted sum of the relative annual dividends d_i of the individual assets in the portfolio, where the total dividend of an asset is set in relation to its highest sales price during the last twelve months. Alternatively, the lowest price of an asset during the last twelve months or its current value could be used as a reference value. We decided to use the highest price of an asset here since this approach, in general, underestimates the relative annual dividend and is therefore a cautious value. Hence,

$$d_i = \frac{d_i^{\mathrm{a}}}{p_i^{\mathrm{h}}},$$

where d_i^{a} denotes the nominal annual revenue of asset i in the last year and p_i^{h} denotes the highest price (value) of asset i in the last year. The dividend of a portfolio is the weighted sum of dividends of individual assets. Overall, we obtain

$$f_3(x) = \sum_{i=1}^{M} d_i x_i.$$

- **Standard and Poors Star Ranking**

The Standard and Poor's Fund Services GmbH evaluates the performance of most investment funds contained in their data base on an annual basis which results in a performance ranking (star ranking). The ranking is based on the performance of an investment fund in comparison to the sector index and assigns between one star (for a relatively poor performance) and up to five stars (for a very good performance). We will assume in the following that the ranking is additive in the sense that the ranking of a portfolio of investment funds can be obtained as the weighted sum of the rankings of the individual investment funds in the portfolio. Consequently, the fourth objective function can be written as

$$f_4(x) = \sum_{i=1}^{M} sr_i x_i,$$

where sr_i denotes the number of stars assigned to investment fund i.

- **Volatility**

The risk associated to a certain investment is often measured in terms of the standard deviation of the time series reflecting the price of an asset in the past. The volatility of a portfolio can be found as the square root of its variance which is given by $x^T \sigma x$, cf. (1). It relies on the length of the underlying time series and can be calculated in various ways, yielding, for example, a 12-month volatility or a 3-year volatility. In the following, a 12-month volatility is used to formulate the fifth objective function. The resulting value is multiplied by (-1) to fit into the maximization framework of problem (2):

$$f_5(x) = -\sqrt{x^T \sigma x} = -\sqrt{\sum_{i=1}^{M} \sum_{j=1}^{M} \sigma_{ij} x_i x_j}.$$

As in the Markowitz model, σ_{ij} denotes the covariance between the returns of assets $i, j \in \{1, \ldots, M\}$, based on observations over the last twelve months and summarized in the covariance matrix σ.

It should be noted that the simultaneous consideration of highly correlated criteria like the 12-month performance and the 3-year performance of an investment entails the risk of double counting, see, for example, Jenkins and Anderson (2001) and Jenkins and Anderson (2003). This has to be taken into account when deriving utility functions and an overall preference structure which will be discussed in detail in Section 3.

Moreover, many alternative objective hierarchies, incorporating other criteria like, for example, the social responsibility of an investment fund, the number of securities in a portfolio or the mean absolute deviation as an alternative measure of risk, can be formulated. While the models and methods discussed in the following focus on the objective hierarchy specified in Fig. 2, the results directly transfer to other choices of criteria and can be easily adapted to the preferences of the individual decision situation.

3. Utility functions and multiattribute utility theory

In Section 2 we introduced the five attributes by which we measure the performance of a portfolio. We will combine these objectives using DM-specific utility functions for each of the attributes (see Section 3.1) and an additive global utility function to derive our final model (see Section 4).

Keeney and Raiffa (1993) state that the set of attributes in an objective hierarchy should be *complete*, *operational*, *decomposable*, *non-redundant*, and *minimal*. In this regard our hierarchy might be criticized for not being non-redundant (as we have the correlated criteria of 1- and 3-year performance) and not being completely decomposable, as the S&P star ranking is a measure relevant for both risk and return. We felt, however, that this is justified, as the issue was discussed with the project partner and inclusion of all these measures was considered important.

The other issue is the existence of an additive value (or utility function). Theoretically, the condition for its existence is that the attributes are mutually preferentially independent (see Keeney and Raiffa (1993, p. 111), for a formal definition). Again, due to correlation among some of the attributes, this condition is not completely satisfied. Nevertheless, the project partner accepted the assumption of the existence of an additive global utility function.

3.1. Generation of DM-specific utility functions

Since generally neither utility functions for the different decision criteria nor a global utility function are known explicitly, pointwise representations of the utility functions for the individual criteria or of the global utility function are used to generate approximations of the respective utility functions. Utility points can be either specified for the global utility function, thus combining all the individual decision criteria into a single function, or an additive approach could be used in which utility points are given for all individual decision criteria, yielding one utility function per criterion. In the latter case, the resulting utility functions are combined into a global utility based on a weighting of the criteria according to their relative importance for the decision maker, cf. Section 3.

While the knowledge of a global utility function would have many advantages, its generation and, in particular, the specification of an appropriate number of utility points by the decision maker taking into account all criteria and their interrelations, is very difficult in practice (see, for example, Keeney and Raiffa, 1993). Tangian (2001) (see also Tangian and Gruber, 1997) tries to overcome these difficulties by generating a quadratic function representing level sets (i.e., sets of equally preferred solutions) of the global utility function. Even though this leads to a relatively simple representation of the preference structure, utility points interrelating all considered criteria are needed also in this model. Moreover, due to the simple structure of the function (if more data points are given than needed to define the functional coefficients, an approximating quadratic function is constructed using a least squares method) valuable preference information may be lost.

In the following we will focus on additive models since they can be viewed as a compromise between the time required by the decision maker, the achieved accuracy and the simplicity of the model (see, for example, Hallerbach and Spronk, 1997).

Given a finite set of n pairwise different utility points (data points) $D = \{D_1, \ldots, D_n\}$ with coordinates $D_i = (s_i, t_i)$, $i = 1, \ldots, n$, satisfying $-\infty < s_1 < \cdots < s_n < \infty$ for each of the considered objective functions (these points have to be provided by the decision maker), various different interpolation techniques can be used to generate an approximation of the underlying utility functions. We will concentrate on piecewise linear, piecewise quadratic and cubic spline interpolations since they appear to combine several desirable properties like an easy representation and a sufficiently good approximation of the (unknown) correct utility functions.

3.2. Interpolation methods

Perhaps the easiest way to combine a set of utility points D is by determining a piecewise linear function with break points exactly at the utility points. This yields a utility function $u^{\mathrm{pl}} : [s_1, s_n] \to \mathbb{R}$ which can be piecewise defined as

$$u_i^{\mathrm{pl}}(s) = t_i + \frac{t_{i+1} - t_i}{s_{i+1} - s_i}(s - s_i), \quad s \in [s_i, s_{i+1}), \;\; i \in \{1, \ldots, n-1\}.$$

Edwards and Barron (1994) have classified different types of functions encountered in practical applications. In many of these cases piecewise linear utility functions are convenient. However, a relatively high curvature of the utility function and/or a relatively small number of utility points may result in large approximation errors.

To account for the curvature of the approximated utility functions, piecewise quadratic functions may be utilized. This approach can be additionally motivated by a general property of many utility functions, that is, that the marginal utility is decreasing (see, for example, Keeney and Raiffa, 1993).

Let $D_{i-1} = (s_{i-1}, t_{i-1})$, $D_i = (s_i, t_i)$ and $D_{i+1} = (s_{i+1}, t_{i+1})$ be three consecutive utility points. Then a piecewise quadratic interpolating function $u^{\mathrm{pq}} : [s_1, s_n] \to \mathbb{R}$ consists of segments

$$u_i^{\mathrm{pq}}(s) = t_i \frac{(s - s_{i+1})(s - s_{i+2})}{(s_i - s_{i+1})(s_i - s_{i+2})} + t_{i+1} \frac{(s - s_i)(s - s_{i+2})}{(s_{i+1} - s_i)(s_{i+1} - s_{i+2})} + t_{i+2} \frac{(s - s_i)(s - s_{i+1})}{(s_{i+2} - s_i)(s_{i+2} - s_{i+1})}$$

for $s \in [s_i, s_{i+1}), i \in \{1, \ldots, n-2\}$, and $u^{\mathrm{pq}}(s) = u_{n-1}^{\mathrm{pq}}(s)$ for $s \in [s_{n-1}, s_n]$.

One drawback of piecewise quadratic interpolations is that even if the set of utility points represents a monotone (strictly increasing or strictly decreasing) function, the resulting piecewise quadratic approximation may fail to have the same monotonicity property (see, for example, Fig. 4). To overcome this difficulty, non-monotone quadratic segments may be replaced by linear segments to obtain a combined piecewise linear and quadratic, but monotone function.

The interpolation schemes above are relatively simple and allow for an easy implementation. A major drawback, however, is that the approximations are in general not differentiable at the interpolation nodes, whereas utility functions are often assumed to be differentiable. To obtain smooth approximating functions, polynomials of higher order are needed which immediately leads to polynomial spline functions. To keep the model simple, we will concentrate on cubic splines in the following, that is, on spline functions of degree $r = 3$. For a more detailed introduction to spline functions we refer, for example, to Boor (1978) or Shikin and Plis (1995).

Cubic splines $u : [s_1, s_n] \to \mathbb{R}$ consist of cubic polynomials pieced together in such a way that their values and those of their first two derivatives coincide at the knots $s_1, \ldots, s_n$, and that one of the following additional constraints at the boundaries are satisfied:

(a) $u''(s_1) = u''(s_n) = 0$ (u is called natural),
(b) $u^{(k)}(s_1) = u^{(k)}(s_n)$ for $k = 0, 1, 2$ (u is called periodic),
(c) $u'(s_1) = t'_1$, $u'(s_n) = t'_n$ for given numbers t'_1 and t'_n (u has the Hermite property).

Using condition (c) seems to be appropriate in the context of utility functions since it allows for a linear continuation of the approximation outside the interval $[s_1, s_n]$ by setting

$$t'_1 := \frac{t_2 - t_1}{s_2 - s_1} \quad \text{and} \quad t'_n := \frac{t_n - t_{n-1}}{s_n - s_{n-1}}.$$

At this point the cubic spline function approximating the utility function as $u^{cs}(s) := u(s)$ is completely determined and can be piecewise described as

$$u_i^{cs}(s) = a_i(s - s_i)^3 + b_i(s - s_i)^2 + c_i(s - s_i) + d_i, \quad s \in [s_i, s_{i+1}), \ \ i \in \{1, \ldots, n-1\},$$

with real coefficients a_i, b_i, c_i, d_i, $i = 1, \ldots, n-1$. The explicit computation of these coefficients remains a technical task and is described in detail, for example, in Stoer and Bulirsch (1993).

Note that, while the resulting function u^{cs} is differentiable throughout its domain, the same problems as in the case of the piecewise quadratic interpolation may be encountered with respect to the monotonicity of the approximation.

3.3. Comparison of the interpolation methods

We consider the following two sets of utility points to compare the interpolation methods discussed in the previous sections:

	Objective value (s_i)	Utility (t_i)
D_1	0	100
D_2	2	80
D_3	6	70
D_4	12	40
D_5	20	15
D_6	30	10
D_7	50	5

	Objective value (s_i)	Utility (t_i)
D_1	0	100
D_2	2	80
D_3	6	78
D_4	12	40
D_5	20	15
D_6	30	10
D_7	50	5

Note that the two data sets differ only in the coordinates of the point D_3. Even though both data sets are monotone in the sense that a higher objective value always results in a lower utility, the second data set contains two points, namely the points D_2 and D_3, with only slightly different utility values. While all three

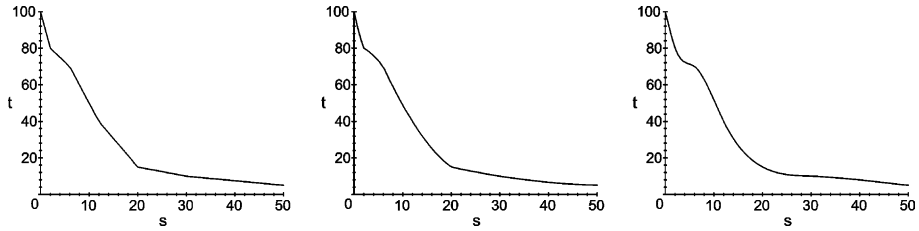


Fig. 3. The different interpolation methods applied to the first data set. The first graph shows a piecewise linear interpolation, the second graph shows the piecewise quadratic interpolation and the third graph shows the cubic spline interpolation.

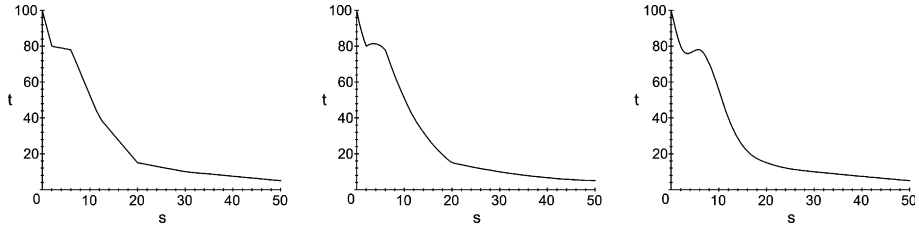


Fig. 4. The three interpolation methods applied to the second data set. The piecewise linear interpolation shown is the first graph is monotone while the piecewise quadratic interpolation and the cubic spline interpolation shown in the second and third graph are non-monotone.

interpolation methods yield monotone functions if applied to the first data set (see Fig. 3), the piecewise quadratic as well as the cubic spline interpolation yield non-monotone functions in case of the second data set which is an unwanted property in most practical applications (see Fig. 4).

These simple examples show that no general preference can be given to any of the described methods, but that an appropriate interpolation method should be selected according to the data set at hand. However, since the cubic spline interpolation has several desirable properties and, in particular, since it yields a continuously differentiable, smooth function, this approach will be used in the following to approximate the utility functions for the portfolio optimization problem. If a non-monotone utility function is obtained with the cubic spline interpolation, this can be viewed as an indication that the selection of the utility points needs further consideration. In any case, non-monotone segments can be replaced by linear segments to obtain a monotone utility function.

4. The MAUT model

Based on the objective hierarchy developed in Section 2 and on the utility theory discussed in Section 3 we use an objective function to maximize the overall (individual) utility of the investor. In this way a single criterion optimization problem is obtained, and a unique optimal objective value is generated according to the preferences specified by the decision maker. The complete decision making phase is thus incorporated into the development of the utility function while the Markowitz model (1) generates an efficient frontier from which a preferred solution yet has to be selected (cf. Fig. 1).

Using positive weights $w_q > 0$ for the five decision maker specific utility functions $u_q(f_q(x))$, where $f_q(x)$ are the five objective functions described in Section 2 we obtain the following MAUT model for portfolio optimization.

$$\begin{aligned} \max \quad & \sum_{q=1}^{5} w_q u_q(f_q(x)), \\ & \sum_{i=1}^{M} x_i = 1, \\ & x_i \geqslant 0 \quad \forall i = 1, \ldots, M. \end{aligned} \tag{3}$$

We now include further constraints in our model which extend the constraints of the Markowitz model (1) in that they allow to specify both the number of assets to be included in the portfolio as well as lower and upper bounds on the percentage of those assets selected to be included in the portfolio. Such constraints were proposed in the context of the standard mean–variance model by Chang et al. (2000) to avoid impractical solutions that contain many assets with a very small percentage of the portfolio, a situation that occurs frequently in the classical mean–variance model.

In order to accommodate the additional constraints, we use the binary variables

$$y_i = \begin{cases} 1, & \text{asset } i \text{ is included in the portfolio,} \\ 0, & \text{otherwise.} \end{cases}$$

We obtain the following non-linear mixed integer programming model for the portfolio optimization problem:

$$\max U(x) = \sum_{j=1}^{5} w_j u_j(x), \tag{4}$$

$$\sum_{i=1}^{M} x_i = 1, \tag{5}$$

$$x_i - l_i y_i \geqslant 0, \quad i = 1, \ldots, M, \tag{6}$$

$$x_i - r_i y_i \leqslant 0, \quad i = 1, \ldots, M, \tag{7}$$

$$\sum_{i=1}^{M} y_i = m, \tag{8}$$

$$x_i \geqslant 0, \quad i = 1, \ldots, M,$$

$$y_i \in \{0, 1\}, \quad i = 1, \ldots, M,$$

where the weights $w_1, \ldots, w_5$ of the global utility function (4) are specified by the decision maker such that $\sum_{j=1}^{5} w_j = 1$. In the case study discussed in Sections 2 and 6.1, these weights were chosen as $w_1 = 0.12$, $w_2 = 0.24$, $w_3 = 0.04$, $w_4 = 0.4$ and $w_5 = 0.2$. Moreover,

$$u_1(x) = u_1^{\mathrm{cs}}(f_1(x)) = u_1^{\mathrm{cs}}\left(\sum_{i=1}^{M} r_i^{12} x_i\right),$$

$$u_2(x) = u_2^{\mathrm{cs}}(f_2(x)) = u_2^{\mathrm{cs}}\left(\sum_{i=1}^{M} r_i^{36} x_i\right),$$

$$u_3(x) = u_3^{\mathrm{cs}}(f_3(x)) = u_3^{\mathrm{cs}}\left(\sum_{i=1}^{M} d_i x_i\right),$$

$$u_4(x) = u_4^{\mathrm{cs}}(f_4(x)) = u_4^{\mathrm{cs}}\left(\sum_{i=1}^{M} sr_i x_i\right),$$

$$u_5(x) = u_5^{\mathrm{cs}}(f_5(x)) = u_5^{\mathrm{cs}}(-\sqrt{x^T \sigma x}).$$

See Sections 2 and 3.2 for the details of the implementation.

Constraint (5) in the above model corresponds to the general knapsack constraint (cf. problem (1)). Constraints (6)–(8) are optional and can be added or dropped according to decision makers preferences. In particular, constraints (6) and (7) can be used to incorporate lower bounds l_i and upper bounds r_i on the investment in individual assets ($0 \leqslant l_i, r_i \leqslant 1$ and $l_i \leqslant r_i$, $i = 1, \ldots, M$). In most applications, all lower bounds (upper bounds, respectively) will be equal, i.e., $l_i = l$ and $r_i = r$, $i = 1, \ldots, M$, with fixed values $0 \leqslant l \leqslant r \leqslant 1$. Implementing these bounds avoids large numbers of very small investments (lower bounds) and at the same time ensures a sufficient diversification of the investment (upper bounds). Additionally, constraint (8) can be used to specify the total number $m \in \mathbb{N}$ of assets in a portfolio. Other constraints may be added as necessary, to ensure, for example, that a certain asset is part of the portfolio, or that certain segments of the market are covered. Note that lower and upper bounds have to be chosen carefully so that feasible solutions exist (e.g. l_i can be positive for at most m assets, $\sum_{i=1}^{M} l_i \leqslant 1$). For a similar discrete model where x_i are integer variables and represent numbers of lots of an asset bought, Mansini and Speranza (1999) proved that finding a feasible solution is $\mathcal{NP}$-hard.

It is also important to remark that the global objective function can be non-convex in general. With the combination of continuous and binary variables the resulting problem will usually be a very difficult non-convex mixed integer programming problem. Even if the utility functions u_q are monotone (as was the case in our case study), which implies $U(x)$ to be quasi-convex, the cardinality constraints (6) and lower and upper bounds (7) and (8) imply that the feasible set of the problem is not connected, and the problem remains a non-convex optimization problem on the boundary of the feasible set.

5. Solution methods

Since the model formulated in Section 4 is a generally very complex, non-convex optimization problem in a large number of variables, exact solution methods seem to be inapplicable. Moreover, due to the fuzziness of input parameters like, for example, the utility points and the approximated utility functions as well as their respective weighting, the notion of an "exact optimum" appears to be inadequate.

Consequently, four different heuristic solution methods, including a local search approach (particularly designed for the given problem context) and three metaheuristics (simulated annealing, tabu search and an implementation of a genetic algorithm) were applied to the problem and compared on different classes of problem instances.

All methods are based on two neighborhood structures of a given portfolio, which we describe in the context of the local search algorithm.

5.1. Two phase local search algorithm

The two phase local search algorithm is based on a local improvement method that searches for the best alternative to a given portfolio x within a certain *neighborhood* of x. We used two neighborhood structures based on the following definitions.

- The *member neighborhood* $N_1(x)$ of a portfolio x including m different assets is the set of all portfolios consisting of the same percentage distribution of assets and differing in at most one asset (one member) from the portfolio x. This is essentially a neighborhood based on the 0–1 variables y_i.

- The *percentage neighborhood* $N_2(x)$ of a portfolio x including m different assets is the set of all those portfolios that consist of the same assets and differ in at most two percentage values from the portfolio x. Here, the differing percentage values may be one unit higher or one unit lower than those in x. This neighborhood is based on changing the x_i variables, whereas y_i variables are kept constant. (Note that this definition leads to a discretization of the solution set.)

Using these neighborhood structures facilitates both the control over lower and upper bounds on the investments in individual assets (constraints (6) and (7)) and over the total number of assets in a portfolio (constraint (8)). Furthermore, due to its simplicity it supports efficient implementations.

The basic steps of the algorithm can be described as follows: Starting from an initial portfolio x, the neighborhood of x is searched for that portfolio yielding the maximal improvement. This process is then repeated until no further improvement is possible. During the iterations, on one hand the set of assets in the portfolio and on the other hand their percentage values are adapted by alternately exploring the neighborhoods $N_1(x)$ and $N_2(x)$ of the current portfolio P. As these changes alternate, we call the algorithm two phase local search.

Since both improvement processes only lead to small changes in the portfolio, the search procedure is restricted to a relatively small neighborhood of the current portfolio. To avoid an early termination of the procedure at a local maximum, an additional searching process is incorporated that allows the exchange of more than one asset at a time. In this process a subset of assets from the current portfolio is randomly selected and removed from the portfolio. Then new assets are successively added in a greedy fashion, and the modified portfolio is accepted if it yields a higher global utility value. Even though this extended search procedure is rather time consuming, it turns out to lead to significant improvements of the achieved solutions.

The initial portfolio may be found either by a deterministic approach, where new assets are successively added to the portfolio following a greedy strategy until the number of assets m is reached, or in a random approach which randomly selects an initial portfolio in accordance with the problem constraints.

5.2. Metaheuristics

In addition to the two phase local search discussed in the previous section, three metaheuristics were implemented and compared in the context of the portfolio optimization problem. We refer for example, to the texts by Reeves (1993), Rayward-Smith et al. (1996), or Aarts and Lenstra (1997) for details on metaheuristics.

For the simulated annealing algorithm as well as the tabu search algorithm, the same neighborhood structures were used as for the two phase local search.

- **Simulated annealing**

The simulated annealing algorithm starts from a randomly generated feasible solution. In each iteration a random member of the portfolio is exchanged. If this change deteriorates the objective value, it is accepted with a certain probability depending on the state of the process, represented by a parameter c_i. In particular, c_i is the expected value of accepted deteriorations and is reduced after a predefined number of n_l iterations following a standard cooling scheme (c.f. Section 6 for the specification of the parameters).

The corresponding parameter for acceptance of deteriorations due to percentage changes is set to $0.1c_i$. If a change is accepted, the percentages of two assets in the portfolio are changed by one percentage point.

- **Tabu search**

The tabu search algorithm uses two tabu lists T_n of length t_n and T_0 of length t_0. These contain the assets most recently added respectively removed from the portfolio. The member neighborhood $N_1(x)$ is searched repeatedly, allowing for a maximum of n_{md} (a predefined number) deteriorations of the objective value during the search procedure. Starting from the best solution found during this search process, the percentage neighborhood $N_2(x)$ is searched as in the local search algorithm. The algorithm iterates by alternately using these two neighborhood structures.

As a way of diversification, the second best solution found in each search of the membership neighborhood is stored, and a new search process is started from there. In this diversification phase n_{md} is set to zero. The initial (feasible) solution is chosen randomly.

- **Genetic algorithm**

The genetic algorithm creates an initial population of g feasible portfolios randomly. In every one of a fixed number of iterations, the $g - r$ best members of the population survive. For the remaining r, one parent is chosen from the total population, the other from among the $g - r$ surviving individuals. Then m assets with their percentages present in the parents are chosen randomly for the offspring. In addition to that, a mutation process slightly perturbing the percentages of the funds in the new solution can be applied: A mutation operator multiplies the percentage of a randomly chosen asset by a fixed value < 1 or > 1 (or 1 if no mutation is desired).

Note that, in order to obtain feasibility of the newly generated portfolios, the percentage values of the individual assets may have to be adapted and normalized.

Metaheuristics have also been used in Chang et al. (2000) to generate the efficient frontier of cardinality constrained portfolio optimization problems (in fact, this study also considered a tabu search, a simulated annealing, and a genetic algorithm implementation). With the constraints of our MAUT model of Section 4 being the same as those in Chang et al. (2000) these could also be adapted for use with our model by including our objective function.

6. Computational results

The performance of the algorithms described in Section 5 was tested on two different problem classes. In the first, we used data from the Standard and Poor's database of 1108 funds of which we selected 190 non-dominated assets. Dominance of assets is understood as dominance of single asset portfolios. A non-dominated asset is therefore not dominated by any other asset with respect to all five criteria, where criterion f_5 in the case of a single asset portfolio is only based on the variance of this asset. Even though an optimal portfolio of (3) may contain dominated assets due to the non-linearity of the volatility (objective f_5), this approach is based on the wish for a portfolio consisting of assets that are also individually strong. For evaluation we used decision maker specific utility functions and the five objective functions described in Section 2.

To compare the performance of the different algorithms on other and more general problem instances, we additionally generated 10,000 assets with uniformly distributed random values for seven (linear) objective values. Of those, the 1416 non-dominated assets were selected. The utility functions used in this case were monotone or random (non-monotone) and equal weights $w_q = 1/7$ were applied to get the global utility function. Because in the randomly generated problem instances we exclusively work with additive (linear) criteria dominated assets can be ignored without changing the optimal solution value in the case of monotone utility functions.

The main attributes of the problem instances are summarized in the table below:

	Partially approximated large real case	Small real case	Large monotone case	Large random case
# of assets	$M = 190$	$M = 40$	$M = 1416$	$M = 1416$
# of criteria	5	5	7	7
Generation method	S&P database (with incomplete covariance matrix)	S&P database (with complete covariance matrix)	Uniformly random in the set $\{0, 1, \ldots, 1000\}$	Uniformly random in the set $\{0, 1, \ldots, 1000\}$
Utility functions	Section 6.1	Section 6.1	Monotone, cubic splines	Random, piecewise linear
Weighting of the criteria	Section 6.1	Section 6.1	$w_q = 1/7$	$w_q = 1/7$
Bounds (constr. (6,7))	$l_i = 4$, $r_i = 30$	$l_i = 4$, $r_i = 30$	$l_i = 4$, $r_i = 30$	$l_i = 4$, $r_i = 30$
# of assets (constr. (8))	$m = 10$	$m = 10$	$m = 10$	$m = 10$

Since some of the internal parameters of the implemented algorithms have a considerable impact on the quality of the generated solutions and at the same time their optimal choice depends on the given problem instance, different program specifications were compared. These include:

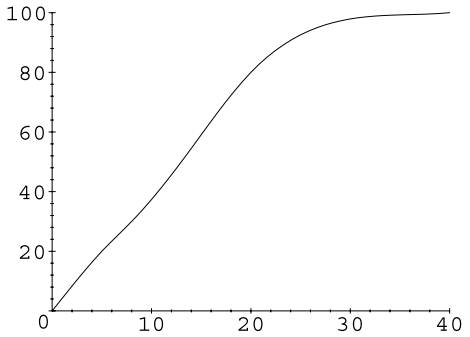
(1) For the two phase local search heuristic:
n_{mc}, the number of consecutive iterations based on the member neighborhood,
n_{pc}, the number of consecutive iterations based on the percentage neighborhood,
n_{smc}, the number of consecutive iterations between application of the extended neighborhood, and the procedure used for the generation of the initial solution.
(2) For the simulated annealing algorithm:
The cooling schedule of the algorithm is controlled by the parameters b_1 and b_3 where b_1 determines the probability of accepting deteriorating objective values and b_3 influences the speed of the cooling process. Moreover, the termination of the algorithm is controlled by the specification of the total number of iterations.
(3) For the tabu search algorithm:
t_n, the length of the tabu list T_n containing assets added to the portfolio in the preceding iterations,
t_0, the length of the tabu list T_0 containing assets removed from the portfolio in the previous iterations, and
n_{md}, the maximal number of consecutive deteriorations of the objective value.
(4) For the genetic algorithm:
The performance of the genetic algorithm depends on the number of genotypes g in the initial population, on the number of replaced genotypes r per iteration, and on the total number of iterations (or generations) gen.

All experiments were run on an Intel Pentium III PC with 600 MHz. In the following sections we summarize the best results obtained by each of the algorithms for the four test problems. Each run was repeated ten times. Here, 2PLS = 2 phase local search, SA = simulated annealing, TS = tabu search, GA = genetic algorithm. Below we report the results of our tests.

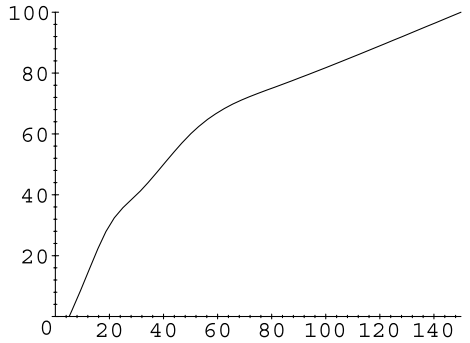
6.1. Results on fund database

Investors were asked to specify utility points for the five criteria 12-month performance, 3-year performance, annual dividend, Standard and Poor's star ranking and volatility. The resulting utility values were normalized to lie in the interval $[0, 100]$. We used the cubic spline interpolation described in Section 3.2. The utility points and corresponding functions are shown below.

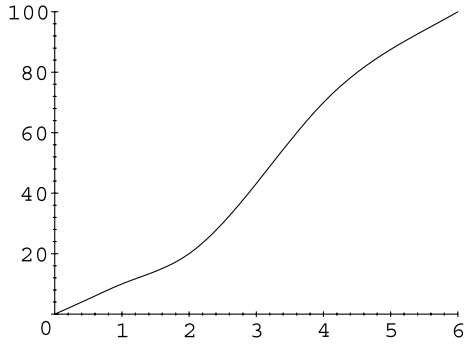
12-month performance	Assigned utility
0	0
5	20
8	30
13	50
20	80
40	100



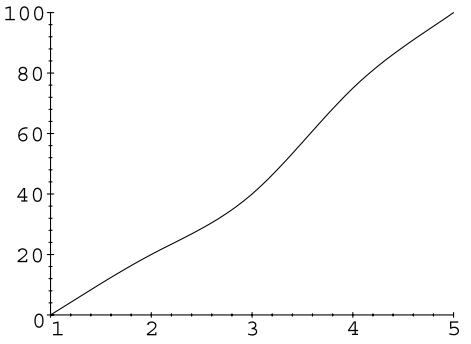
3-year performance	Assigned utility
10	10
20	30
30	40
40	50
50	60
80	75
150	100



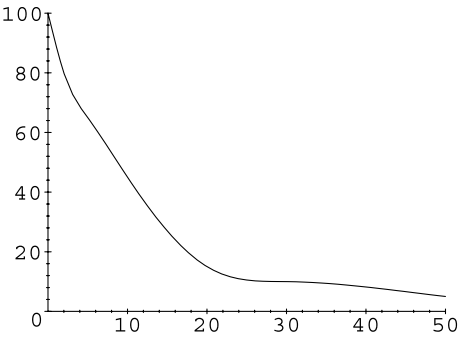
Annual dividend (in %)	Assigned utility
0	0
0.5	5
1	10
2	20
4	70
6	100



S&P star ranking (# of stars)	Assigned utility
1	0
2	20
3	40
4	75
5	100



12-month volatility	Assigned utility
0	100
2	80
5	65
10	45
20	15
30	10
50	5



Note that the first four utility functions are strictly increasing while the last one, modeling the volatility, is strictly decreasing, i.e here we consider objective function $-f_5(x)$ which is to be minimized. All generated functions are monotone.

In addition to the specification of the five utility functions, a relative weighting of the criteria is needed accounting for their relative importance to the decision maker (cf. Section 3). In the case study, the relative weight of the 12-month performance (objective function f_1) was set to $w_1 = 0.12$ while the 3-year performance (objective function f_2) was considered more relevant and was assigned a weight of $w_2 = 0.24$. This yields a total weight of $w_1 + w_2 = 0.36$ for these two closely related criteria. The relative weight for the annual dividend (objective function f_3) was set to $w_3 = 0.04$.

Moreover, the Standard and Poor's star ranking (objective f_4) was assigned a relatively high weight of $w_4 = 0.4$. Since part of the risk associated with a portfolio is already captured by the Standard and Poor's star ranking, the relative weight of the volatility (objective f_5) was set to a comparably low value of $w_5 = 0.2$, yielding an overall weight of $w_4 + w_5 = 0.6$ for these risk-related criteria.

It was not possible to obtain a full covariance matrix for the 190 non-dominated funds in this example due to limited capabilities of the S&P database. We used covariance values randomly generated in the interval $[-0.4, 1]$.

Table 1 shows that the two phase local search heuristic has a good and stable performance, yielding satisfactory solution values. The simulated annealing algorithm produces stable results which are slightly better than those of the two phase local search. The tabu search algorithm appears to be less stable on this problem class than the other three algorithms. A long tabu-list appears to be advantageous particularly for the newly added assets in a portfolio. The genetic algorithm seems to have the least difficulties with this problem class and produces the best results and runs most stable.

Table 1
Results for the partially approximated large real case

Algorithm	$U(x)$	Standard deviation	Function evaluations	CPU time	Parameters
2PLS	91.197	0.127	105,000	1.291	$n_{pc} = 30$, $n_{mc} = 60$, $n_{smc} = 25$, initial solution random
SA	91.413	0.093	282,800	7.226	$b_1 = 0.25$, $b_3 = 0.25$, 100,000 iterations
TS	90.399	1.504	151,100	3.932	$t_n = 8$, $t_0 = 2$, $n_{md} = 3$
GA	92.336	0.002	150,500	7.124	$g = 500$, $r = 300$, gen = 500

Table 2
Results for the small real case

Algorithm	$U(x)$	Standard deviation	Function evaluations	CPU time	Parameters
2PLS	86.870	0.325	20,400	0.363	$n_{pc} = 25$, $n_{mc} = 50$, $n_{smc} = 20$, initial solution random
SA	86.841	0.242	56,500	1.417	$b_1 = 1$, $b_3 = 1$, 20,000 iterations
TS	86.353	0.664	41,900	1.027	$t_n = 8$, $t_0 = 8$, $n_{md} = 3$
GA	86.787	0.102	75,200	3.367	$g = 200$, $r = 150$, gen = 500

For a smaller data set containing only 40 funds, all of which where non-dominated, we had a complete covariance matrix available.

Obviously, the smaller size of the problem leads to a reduction in computational effort, see Table 2. The lower utility values might be partly due to the fact, that for this problem we had a complete covariance matrix and did not have to rely on randomly generated covariances, and partly due to the smaller feasible set.

The two phase local search heuristic and the simulated annealing algorithm lead to similar results that appear to be good and sufficiently stable. Different from the partially approximated real case, tabu search produces satisfying results for the small data set, however, it is still outperformed by the other three methods. The genetic algorithm performs nearly as well as the two phase local search and the simulated annealing method, and the standard deviation of the achieved results is smaller than for the other two procedures.

6.2. Results on randomly generated instances

The six utility points (s_i, t_i) for each objective in the large monotone case were selected to represent a monotone sequence $t_i < t_{i+1}$, with $s_i = 200(i-1), i = 1, \ldots, 6$. We used the cubic spline interpolation described in Section 3.2 to compute the seven utility functions.

The utility points for the large random case were randomly generated (the objective values s_i are uniformly distributed in the set $\{0, 1, \ldots, 1000\}$ while the utility values t_i are uniformly randomly distributed in the set $\{0, 1, \ldots, 100\}$). A piecewise linear interpolation seems most suitable for this data set (cf. Section 3.2). It yields seven non-monotone piecewise linear utility functions.

As might be expected from the more difficult (highly non-convex) structure of this instance, in general computation times, variability of results, and function evaluations increased whereas $U(x)$ values decreased, see Tables 3 and 4.

The two phase local search heuristic performs well also on the randomly generated problem instances. The high standard deviation particularly in the second case indicates multiple restarts from different starting solutions.

Maybe due to the relatively narrow neighborhood structure, the simulated annealing and tabu search algorithms do not produce as good results for this hard problem instance.

Table 3
Results for the large monotone case

Algorithm	$U(x)$	Standard deviation	Function evaluations	CPU time	Parameters
2PLS	84.909	0.001	344,000	5.180	$n_{pc} = 20$, $n_{mc} = 60$, $n_{smc} = 10$, initial solution greedy
SA	84.501	0.808	126,200	2.016	$b_1 = 0.25$, $b_3 = 1$, 50,000 iterations
TS	83.906	0.923	797,000	11.897	$t_n = 8$, $t_0 = 2$, $n_{md} = 2$
GA	84.204	0.895	81,000	5.333	$g = 1000$, $r = 400$, gen = 200

Table 4
Results for the large random case

Algorithm	$U(x)$	Standard deviation	Function evaluations	CPU time	Parameters
2PLS	86.914	1.077	926,600	11.073	$n_{pc} = 30$, $n_{mc} = 60$, $n_{smc} = 30$, initial solution greedy
SA	80.720	2.342	680,200	8.723	$b_1 = 0.5$, $b_3 = 0.5$, 200,000 iterations
TS	80.111	2.771	676,500	7.783	$t_n = 2$, $t_0 = 2$, $n_{md} = 2$
GA	86.083	0.349	241,000	7.904	$g = 1000$, $r = 800$, gen = 300

The strength of the genetic algorithm may be seen in the fact that solutions in very different parts of the solution space are generated in each iteration of the procedure, thus allowing a much broader view of the solution space. This observation is in accordance with results of Deb (2001) who describes very successful applications of evolutionary algorithms to multicriteria optimization problems.

7. Conclusions

In this paper we propose a model for portfolio optimization based on multiattribute utility theory and the classical mean–variance model of Markowitz. We address criticism of the Markowitz model and extend it by formulating a hierarchy of objectives, which decomposes risk and return into five sub-objectives. For each of the five objectives, decision-maker specific utility functions are generated from utility values specified by the user. Various methods for interpolation are used, depending on the nature of the points specified. In this way, we allow individual investors preferences to be taken into account.

Our final model also includes constraints on lower and upper bounds on percentages of assets included in the portfolio as well as a specification of the number of assets. Consequently, we formulate a non-convex mixed-integer programming model. We implemented several heuristics to solve this problem. Numerical results show, that good solutions can be obtained for problem sizes relevant in practical applications in just a few seconds of CPU time.

Acknowledgements

We wish to express our thanks to Standard and Poor's Fund Services GmbH, Hochheim, Germany, for fruitful discussions and information. We would also like to thank two anonymous referees, whose comments on an earlier version of the paper were very helpful.

References

Aarts, E., Lenstra, J. (Eds.), 1997. Local Search in Combinatorial Optmization. John Wiley and Sons, Chichester.

Arthur, L., Ghandforoush, P., 1987. Subjectivity and portfolio optimization. In: Lawrence, K., Guerard, J., Reeves, G. (Eds.), Advances in Mathematical Programming and Financial Planning. JAI Press, Greenwich, CT, pp. 171–186.

Ballestero, E., 1998. Approximating the optimum portfolio for an investor with particular preferences. Journal of the Operational Research Society 49, 998–1000.

Ballestero, E., Romero, C., 1996. Portfolio selection: A compromise programming solution. Journal of the Operational Research Society 47, 1377–1386.

Boor, C.D., 1978. A Practical Guide to Splines. Springer, New York.

Chang, T., Meade, N., Beasley, J., Sharaiha, Y., 2000. Heuristics for cardinality constrained portfolio optimisation. Computers and Operations Research 27, 1271–1302.

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

Edwards, W., Barron, F.H., 1994. SMARTS and SMARTER: Improved simple methods for multiattribute utility measurement. Organizational Behavior and Human Decision Processes 60, 306–325.

Gouriéroux, C., 1997. ARCH Models and Financial Applications. Springer, New York.

Hallerbach, W., Spronk, J., 1997. A multi-dimensional framework for portfolio management. In: Karwan, M.H., Spronk, J., Wallenius, J. (Eds.), Essays in Decision Making. A Volume in Honour of Stanley Zionts. Springer, Berlin, pp. 275–293.

Jenkins, L., Anderson, M., 2001. A statistical approach to eliminating some criteria in a multi-criteria decision analysis with little loss of information. Working paper, presented at the International Conference on Multi-Criteria Decision Making in Cairo, Egypt, 2001.

Jenkins, L., Anderson, M., 2003. A multivariate statistical approach to reducing the number of variables in data envelopment analysis. European Journal of Operational Research 147 (1), 51–61.

Keeney, R.L., Raiffa, H., 1993. Decisions with Multiple Objectives: Preferences and Value Tradeoffs. Cambridge University Press, Cambridge.

Konno, H., 1990. Piecewise linear risk function and portfolio optimization. Journal of the Operational Research Society of Japan 33 (2), 139–156.

Mansini, R., Speranza, M., 1999. Heuristic algorithms for the portfolio selection problem with minimum transaction lots. European Journal of Operational Research 114, 219–233.

Markowitz, H., 1952. Portfolio selection. Journal of Finance 7, 77–91.

Markowitz, H., 1959. Portfolio Selection: Efficient Diversification of Investments. John Wiley and Sons, New York.

Markowitz, H., Todd, P., Xu, G., Yamane, Y., 1994. Fast computation of mean–variance efficient sets using historical covariances. Journal of Financial Engineering 1 (2), 117–132.

Morita, H., Ishi, H., Nishida, T., 1989. Stochastic linear knapsack programming problem and its application to a ortfolio selection problem. European Journal of Operational Research 40, 329–346.

Pardalos, P., Sandström, M., Zopounidis, C., 1994. On the use of optimization models for portfolio selection: A review and some computational results. Computational Economics 7, 227–244.

Rayward-Smith, V.J., Osman, I., Smith, G. (Eds.), 1996. Modern Heuristic Search Methods. Wiley, Chichester.

Reeves, C.R. (Ed.), 1993. Modern Heuristic Techniques for Combinatorial Problems. Blackwell, Oxford.

Schandl, B., Klamroth, K., Wiecek, M.M., 2001. Norm-based approximation in bicriteria programming. Computational Optimization and Applications 20, 23–42.

Schwehm, C., 2000. A Multicriteria Decision Making Approach to Portfolio Theory. Master's thesis, Department of Mathematics, University of Kaiserslautern.

Shikin, E., Plis, A., 1995. Handbook of Splines for the User. CRC Press, Boca Raton, FL.

Standard and Poor's Funds Services Database, 1999. Standard and Poor's Fund Services GmbH, Germany. 40 arbitrarily selected nondominated funds from the database.

Stoer, J., Bulirsch, R., 1993. Introduction to Numerical Analysis. Springer, New York.

Tangian, A., 2001. Constructing a monotonic quadratic objective function in n variables from a few two-dimensional indifferences. European Journal of Operational Research 130, 276–304.

Tangian, A., Gruber, J., 1997. Constructing quadratic and polynomial objective functions. In: Tangian, A., Gruber, J. (Eds.), Constructing Scalar-Valued Objective Functions. Lecture Notes in Economics and Mathematical Systems 453. Springer, Berlin, pp. 166–194.