Hybrid multi-objective evolutionary computation of constrained downside risk-return efficient sets for credit portfolios

Frank Schlottmann and Detlef Seese
Institute AIFB
Department of Economics
University of Karlsruhe (TH)
D-76128 Karlsruhe, Germany
e-mail {schlottmann | seese}@aifb.uni-karlsruhe.de

First version: December 2001
This version: May 2002

Abstract

In contemporary credit portfolio management, the portfolio risk-return analysis of financial instruments using certain downside credit risk measures requires the computation of a set of Pareto-efficient portfolio structures in a non-linear, non-convex setting. For real-world problems, additional constraints, e. g. supervisory capital limits, have to be respected. Particularly for formerly non-traded instruments, e. g. corporate loans, a discrete set of decision alternatives has to be considered for each financial instrument. The main result of this paper is a new, fast and flexible framework for solving the above issues using a hybrid heuristic method that combines multi-objective evolutionary and problem-specific local search methods in a unique way. We explicitly incorporate computational complexity in some of our considerations and consider proper genetic modelling of portfolio credit risk related problems. Also, we analyse empirical results from a study based on our implementation of the proposed hybrid method in a specific portfolio credit risk model context. These results show that this method is superior in convergence speed to a non-hybrid evolutionary approach and that our implementation finds risk-return efficient sets within reasonable time.

Key words: portfolio credit risk, downside risk, Credit-Value-at-Risk, constrained discrete portfolio optimisation, portfolio selection, hybrid multi-objective evolutionary algorithm

JEL Classification: G11, G20, G31, G33

Introduction

The intensive development of quantitative portfolio credit risk models since the late 1990s and the increasing trade in financial instruments for transferring credit risk like credit default swaps, asset backed transactions etc. are major reasons for a growing importance of credit portfolio risk-return analysis and optimisation. Beyond that, there will possibly be more demand for credit portfolio optimisation as soon as the supervisory capital requirements for banks will be changed due to proposals of the Basle Committee, e. g. by setting new capital weights on some credit risk exposure types, providing supervisory capital relief for risk mitigation and establishing additional regulatory capital requirements for operational risk (cf. Basel Committee for Banking Supervision (2001) and subsequent publications from the Bank of International Settlements).

In this paper, we will focus on an algorithmic framework for the calculation of discrete risk-return efficient sets for credit portfolios with respect to constraints, e. g. imposed by changes of supervisory capital regulations or internal reallocation of risk capital. This kind of portfolio management is of great importance especially for, but not limited to, many German and European banks since the typical largest exposures to credit risk for small and medium size universal banks are loans given to companies not having direct access to the capital market.

In contrast to the methods for the computation of the efficient frontier for a given set of alternative stock market investments based on the portfolio's variance and related measures, usually a non-linear, non-convex downside risk measure like the Credit-Value-at-Risk is preferred for portfolio credit risk-return analysis, therefore requiring a different method of computation. Moreover, this computational problem often cannot be modelled using real-valued variables, since typically neither the decision alternatives allow an arbitrary amount of each credit risk exposure to be traded nor it is possible to obtain a short position providing a hedge for each arbitrarily chosen exposure from a given portfolio. In addition to that, e. g. the capital requirements for credit risk exposures imposed by the banking supervision authorities are an important constraint to be considered in the computation of efficient credit portfolio structures, and these capital requirements are going to be non-linear in the future according to recent proposals of the Basel Committee.

For our considerations, the concept of Pareto-optimality is essential, i. e. efficient structures are Pareto-optimal concerning the two distinct (and in many cases contrary) objective functions specifying the aggregated risk and the aggregated return of each potential credit portfolio structure for a given set of alternatives. Therefore, we are interested in multiple, feasible non-dominated solutions to the constrained portfolio credit risk-return optimisation problem that are comparable to the efficient frontier in stock portfolio investment analysis, but in a discrete search space having many local optima and particularly using multiple target functions not required to be linear, quadratic or convex. In this context, a feasible non-dominated solution is a portfolio structure that does not violate the constraints, and for which we cannot find any other feasible solution being better in all two target function values.

We introduce a novel approach to such problems that combines recent constrained multi-objective evolutionary computation methodology and problem specific knowledge to create a hybrid algorithm providing rapid discovery of a set of efficient credit portfolio structures with respect to constraints for a given instance of the above problem. We derive a general result concerning the appropriate modelling of portfolio credit risk problems for certain genetic variation operators that is not restricted to our problem. Using this concept, we create a proper genetic representation of our decision variables that reflects the problem structure. Furthermore, we derive a local search variation operator that modifies selected solutions in addition to the common genetic operators during the evolution process to improve the convergence speed of the algorithm. Beyond that, this local search variation operator particularly uses portfolio credit risk model features to achieve a high computing speed.

For an empirical evaluation of our concepts, we present the results of a study based on our implementation of the hybrid algorithm. In this study, we calculate sets of efficient portfolio structures for different test problems on a standard Personal Computer. Using a suitably small test portfolio, we show by a complete enumeration of the search space that the hybrid algorithm finds a well-distributed set of different efficient solutions within few minutes whereas the enumeration requires more than an hour of computing time. Beyond that, our large test cases show that the algorithm can still be run on a standard PC using significantly larger portfolio sizes, and since the hybrid algorithm is well-suited for massively parallel implementation there are good prospects even for very large real-world portfolios to be processed. Finally, by comparing the results of a pure evolutionary approach to the performance of our hybrid algorithm on the same test cases, we observe that the latter approach shows a higher convergence speed.

The paper is organised as follows: In the first section, we describe our portfolio credit risk optimisation problem and analyse its structure from a computational perspective. After a short look to traditional methods for multi-objective optimisation, we give a short introduction to multi-objective evolutionary algorithms. Then we derive a proper genetic modelling for portfolio credit risk problems before giving an overview and discussing some elements of our hybrid evolutionary algorithm framework for the computation of constrained risk-return efficient credit portfolio structures. The next section introduces some specific credit portfolio model features which are exploited by our implementation of the proposed framework. We provide details of our implementation before describing the parameters and the results of an empirical study where we compare our hybrid implementation with a non-hybrid approach on several test problems.

1 Structure and complexity of the discrete portfolio credit risk optimisation problem

1.1 Some basic definitions and results

In this section, we will analyse the structure and the complexity of the constrained discrete credit portfolio optimisation problem which is to be solved by our hybrid method. The following definitions and results are necessary to understand the structure and the computational complexity of the problem.

Definition 1. Given are $m > 1$ investment alternatives and a time horizon T.
Each investment alternative (obligor) $i \in \{1,...,m\}$ incorporates the risk of default and it is characterised by the following data which is considered to be constant within the time period $(0, T)$:

- net exposure e_i (loss given default of investment i),
- expected rate of return r_i based on e_i (net of cost),
- expected cumulative default probability p_i within the time horizon $(0, T)$,
- and a capital requirement percentage w_i based on e_i.

There is a dependence structure between joint defaults of investment alternatives $i,j \in \{1,\dots,m\}$ that can be modelled by an undirected graph $G = (V, E)$ and a function h: $E \rightarrow \Re$, where $V = \{1,\dots,m\}$ is the vertex set of investment alternatives, $E = V \times V$ is the complete edge set of potential default dependencies between investment alternatives (i, j) and h: $E \rightarrow \Re$ is the function expressing the strength of the dependency between each pair of two investment alternatives (i, j). The function h can be a correlation based function or more general, a copula based function, see e. g. Frey & McNeil (2001) for details about mathematical modelling of default dependencies.

In the following text, we will abbreviate the respective set of scalar variables e_i , r_i , p_i, w_i of all obligors by vectors $e:=(e_1,...,e_m)^T$, $r:=(r_1,...,r_m)^T$, $p:=(p_1,...,p_m)^T$, $w:=(w_1,...,w_m)^T$.

The investor that has to decide about holding a subset of the investment alternatives in her portfolio can e. g. be a bank that wants to optimise its loan portfolio containing m different obligors. According to the next definition, there is a fixed risk capital budget for the investments that can e. g. be given by the bank's maximum supervisory capital which is required to be provided by the bank due to the supervisory regulations (cf. e. g. Basel Committee for Banking Supervision (2001)).

Definition 2. A capital budget of the investor is given by $K > 0$.

We need the following definition to describe possible portfolio structures that can be constructed for the given investment alternatives.

Definition 3. A portfolio structure is given by a vector $x = (x_1, x_2, \dots, x_m)^T$, where $x_i \in \{0, e_i\}$.

Since every x_i can only take the values 0 or e_i, the investor has to decide whether to hold the whole net exposure e_i in her portfolio. In many real world portfolio optimisation problems the decision is e. g. either keeping the obligor i in the credit portfolio or selling the entire net exposure of obligor i to a risk buyer. This is particularly true for formerly non-traded instruments like corporate loans in a bank's credit portfolio. Even if there are more than two decision alternatives for each potential investment i, the decision variables will still consist of a finite, discrete number of choices.

Facing these decision alternatives, an investor has to consider two conflicting objective functions: the aggregated return and the aggregated risk from her portfolio. Usually, there is a trade-off between both objectives since any rational investor will ask for a premium (additional return) to take risk.[1]

Definition 4. The aggregated expected return from a portfolio structure x is calculated by

$$ret(x,p,r)=\sum_{i=1}^{m} r_i x_i-\sum_{i=1}^{m} p_i x_i=\sum_{i=1}^{m}\left(r_i-p_i\right)x_i. \quad (1.1)$$

This is a common net risk adjusted return calculation since the aggregated expected losses are subtracted from the portfolio's aggregated net return.

Definition 5. The aggregated downside risk $risk(x, p; h)$ from the portfolio structure x for the investor is calculated using an appropriate algorithm A such that $risk(x, p; h)$ satisfies the following condition:

$$\forall i\in\{1,\ldots,m\}: p_i\equiv 0 \qquad \forall x,h: risk(x,p;h)\equiv 0 \quad (1.2)$$

The condition specified in Definition 5 expresses the natural property of the risk measure that there is no credit risk for the investor if the default probabilities of all investment alternatives are equal to zero. Of course, the algorithm A and the downside risk measure $risk(\ldots)$ are required to satisfy additional properties to be a downside risk measure. Artzner et al. (1999) provide definitions and a discussion of different risk measures and their properties. For our empirical study described later, we will use the Credit-Value-at-Risk downside risk measure that satisfies condition (1.2). It is a very common measure of credit risk in contemporary credit risk management.

Definition 6. For a given portfolio structure x the Credit-Value-at-Risk (CVaR) at the arbitrary, but fixed confidence level $\alpha\in(0,1)$ is obtained by calculating

$$risk(x,p;h):=q_{pf}^{\alpha}(x,p;h)-\mu_{pf}(x,p) \quad (1.3)$$

where $q^{\alpha}_{pf}(x,p;h)$ is the α-percentile of the cumulative distribution function of aggregated losses calculated from the portfolio for the given parameters x,p and the dependency structure specified by h. Moreover, $\mu_{pf}(x, p)$ is the expected loss calculated by $\mu_{pf}(x,p)=\sum_{i=1}^{m} x_i p_i$.

For our theoretical considerations in this section, we do not need a specification of the calculation procedure for the cumulative distribution function of aggregated losses or q^{α}_{pf}. We will return to these details in the third section describing the portfolio credit risk model used by our implementation of the proposed hybrid algorithm.

Definition 7. The required capital of a given portfolio structure x is $cap(x,w)=\sum_{i=1}^{m} x_i w_i$.

Definition 8. A portfolio structure x is feasible if and only if $cap(x, w)\le K$.

---

[1] Since there is currently no perfect and transparent capital market for trading illiquid financial instruments like loans we do not distinguish explicitly between systematic and idiosyncratic (obligor specific) risk here. In a capital market equilibrium, there is only a premium for taking systematic risk according to the CAPM (see e. g. Sharpe (1964)).

The following definition is essential for the concept of Pareto-optimality.

Definition 9. Given are two distinct feasible portfolio structures x and y.
x dominates y if and only if one of the following cases is true:

(1) $ret(x, p, r) > ret(y, p, r)$ and $risk(x, p; h) \leq risk(y, p; h)$
(2) $ret(x, p, r) \geq ret(y, p, r)$ and $risk(x, p; h) < risk(y, p; h)$.

If x dominates y, we will denote this relationship by $x >_d y$.

This means that a feasible portfolio structure x is better than a feasible portfolio structure y if and only if x is better in at least one of the two criteria and not worse in the other criterion than y. It is obvious that a rational investor will prefer x over y if $x >_d y$.

Definition 10. Given is the set S of all possible portfolio structures for the specified data from Definition 1 and the subset $S' \subseteq S$ of all feasible structures in S. A solution $x \in S'$ is a feasible global non-dominated portfolio structure if and only if it satisfies the following condition:

$$\forall y \in S' : \neg\left(y >_d x\right). \quad (1.4)$$

This means that there is no $y \in S'$: $y >_d x$.

To choose between the best investment alternatives using her preferences or utility function, a rational investor is interested in finding the set of non-dominated portfolio structures that has maximum cardinality. This set is the Pareto-optimal set which is comparable to the efficient frontier of Markowitz (1952), but in a discrete decision space.

Problem 1. The problem of finding the set of feasible Pareto-efficient portfolio structures having maximum cardinality for the set of investment alternatives S can be formulated as: Calculate the set

$$PE^* := \arg\max_{PE \subseteq S'} \left\{ \left|PE\right| \right\} \quad (1.5)$$

where

$$PE := \left\{ x \in S' : \forall y \in S' : \neg\left(y >_d x\right) \right\}. \quad (1.6)$$

Now we want to focus on the computational complexity of Problem 1. Therefore, we consider the usual computing model of Turing machines (see e. g. Papadimitriou (1994) for details), so we temporarily assume all variables to be rational numbers. For definitions and an overview of the P-NP theory see Garey & Johnson (1979). Other computational complexity results concerning credit risk related problems can be found in Seese & Schlottmann (2002b).

Lemma 1. Assuming all scalar variables to be rational numbers, the corresponding decision problem for Problem 1 is NP-hard.

Proof. See appendix.

Corollary 1. Unless P = NP, there is no exact algorithm that calculates PE^* within polynomial computing time (measured by the size of the input m).

Therefore, we need an approximation algorithm for PE^*. The next section will describe approaches for the approximation of PE^* for portfolio data given according to Definition 1.

2 A framework for hybrid multi-objective evolutionary computation of Pareto-efficient credit portfolio structures

2.1 Traditional methods for solving multi-objective optimisation problems

A well-known traditional method of solving multi-objective optimisation problems like Problem 1 is weighted sum scalarisation. In the case of our optimisation problem this e. g. implies that the two objective functions for the credit portfolio's aggregated return and aggregated risk would have to be transformed into a single objective function *g* by adding the two original objective functions using appropriate weights $c_1, c_2 \in \Re$ (see e. g. Ehrgott (2000), pp. 55-76 for details):

$$g(x,p,r;h) = c_1 ret(x,p,r) + c_2 risk(x,p;h) \quad (2.1)$$

These and other traditional methods (see e. g. Ehrgott (2000), pp. 77ff. for an overview) of solving multi-objective problems and handling constraints work well on linear or convex problems in a continuous setting, i. e. problems consisting of linear or at least convex, continuous objective functions, and constraints also satisfying these properties. However, the portfolio optimisation problems based on downside risk measures usually inhibit non-linear, non-convex objective functions, e. g. if the concept of Value-at-Risk or Credit-Value-at-Risk is used – see e. g. Pflug (2000) for the mathematical properties of such downside risk measures. It is an obvious fact that the above mentioned methods relying on convexity in the objective function space produce sub-optimal results at least for some instances of such problems (cf. Ehrgott (2000), p. 77). Moreover, we have to deal with a discrete optimisation problem consisting of a fixed number of distinct choices. So we apply an alternative approach to our discrete optimisation problem. In the following subsection we will give a short introduction to evolutionary approaches to such problems.

2.2 Evolutionary approaches to multi-objective optimisation

Since the first reported implementation and test of a multi-objective evolutionary approach, the Vector Evaluated Genetic Algorithm (VEGA) by Schaffer (1984), this special branch of Evolutionary Algorithms (EAs) has attracted many researchers dealing with non-linear and non-convex multi-objective optimisation problems. After the introduction of VEGA, many different EAs have been proposed for multi-objective optimisation problems, e. g. the Multi-Objective Genetic Algorithm (MOGA) by Fonseca & Fleming (1993), the Niched Pareto Genetic Algorithm (NPGA) by Horn et al. (1994), the Non-dominated Sorting Genetic Algorithm (NSGA) by Srinivas & Deb (1994) which was refined in Deb et al. (2000), the Distance-based Pareto Genetic Algorithm (DPGA) by Osyczka & Kundu (1995), the Strength Pareto Evolutionary Algorithm by Zitzler & Thiele (1998) which was updated in Zitzler et al. (2001), and the Pareto Archived Evolution Strategy (PAES) by Knowles & Corne (1999). Many of the existing MOEAs were primarily designed for problems having continuous variables, i. e. the decision variables can take values from a subset of $\Re$. A comparison between different approaches is e. g. given in Zitzler et al. (2000) and Deb (2001). Theoretical considerations about the convergence of EAs can be found e. g. in Vose (1999), Van Veldhuizen (1999), Rudolph (1998, 2001), Rudolph & Agapie (2000) and Laumanns et al. (2001).
In general, a Multi-Objective Evolutionary Algorithm (MOEA) is a randomised heuristic search algorithm reflecting the Darwinian 'survival of the fittest principle' that can be observed in many natural evolution processes, cf. e. g. Holland (1975). At each discrete time

step t, a MOEA works on a set of solutions $P(t)$ called population or generation. A single solution $x \in P(t)$ is an individual. To apply a MOEA to a certain problem the decision variables have to be transformed into genes, i. e. the representation of possible solutions by contents of the decision variables has to be transformed into a string of characters from an alphabet Σ. The original representation of a solution is called phenotype, the genetic counterpart is called genotype.

For evaluation of each genotype in a population the MOEA requires a quality measure (fitness function) for evaluation of every possible solution (not necessarily feasible if the problem is constrained) that is usually based on the quality of the corresponding phenotype. The individuals from the population $P(t)$ are selected for survival into the next generation after application of variation operators (see below) according to their fitness values. The fitness function and the selection scheme of most MOEAs differ substantially from the fitness functions and selection procedures of single-objective EAs by incorporating special mechanisms for preserving diversity of solutions in the search space (since one is interested in finding a representative Pareto-efficient set containing different solutions) and for selection of solutions that cannot be directly compared using a total order in a multi-dimensional fitness space.

The selected individuals from the current population $P(t)$ are modified using genetic variation operators (see e. g. Fogel & Michalewicz (2000) for an overview). A standard variation operator for discrete decision variables is the one point crossover, i. e. the gene strings of two selected individuals are cut at a randomly chosen position and the resulting tail parts are exchanged with each other to produce two new offspring. This operation is performed with crossover probability p_{cross} on individuals selected for reproduction. The main goal of crossover is to move the population through the space of possible solutions.

In analogy to natural mutation, the second standard variation operator in most MOEAs changes the genes of selected individuals randomly with probability p_{mut} (mutation rate) per gene to allow the invention of new, previously undiscovered solutions in the population. Its second task is the prevention of the MOEA stalling in local optima as there is always a positive probability to leave a local optimum if the mutation rate is greater than zero.

After this short introduction of MOEAs we will now describe the structure and the details of our Hybrid Multi-Objective Evolutionary Algorithm (HMOEA) which incorporates the above general features of MOEAs. Moreover, since our focus is on the development of a flexible framework for discrete credit portfolio optimisation problems, which should not be restricted to a certain downside risk measure and which should support non-linear, non-convex constraints as well we propose a hybrid method that particularly uses ideas that can be found in different MOEA schemes as well as an additional problem-specific local search operator and a problem-specific preprocessing stage.

2.3 Genetic modelling of portfolio credit risk related problems

The first question when applying an EA to a problem is to choose a proper genetic representation of the decision variables. For portfolio credit risk optimisation problems like Problem 1, we assume that the decision variables x_i will be connected to obtain gene strings representing potential solutions. The resulting genotypes consist of real-valued genes which are connected to strings and take either value 0 or e_i depending on the absence or presence of investment alternative i in the current solution. So we obtain strings of length m that represent some of the 2^m combinations of possible (but neither necessarily feasible nor necessarily optimal) portfolio structures.

Since the one point crossover is the first standard variation operator in many EAs (and we also use it in our approach), we should keep some important issues of portfolio credit risk modelling in mind to choose a well adapted genetic representation for the phenotypes.
The one point crossover cuts two gene strings at a random position and crosses the tails of the strings to produce two offspring with crossover probability p_{cross}. The probability of two genes i, j (these variables represent the index of the genes associated to investment alternative i and j, respectively) from one individual being cut by the crossover increases proportional to the distance $|i-j|$ between the two genes in the gene string as the cut position is determined by a draw from a uniform distribution over m-1 cut possibilities:

$$p(\text{crossover cuts gene } i \text{ and } j) = \frac{1}{m\text{-}1}|i \text{ - } j| \quad (2.2)$$

For better results of the crossover operator we must ensure that there is a high probability of good partial solutions being recombined with other solutions and not being destroyed by the crossover's cut operation. More formally, we search for a permutation $\pi(i)$ of the portfolio data represented by our genes ensuring a high probability of success for crossover. Therefore, we have to remember that the degree of dependence between two different investment alternatives i, j plays the central role in aggregated portfolio credit risk calculations according to Definition 5.
In our Problem 1, the dependence structure has no influence on the other objective function given in Definition 4, the aggregated return, so it is sufficient to concentrate solely on the risk objective function when considering the possible influence between different gene positions. According to Definition 1, the dependence structure between investment alternatives is determined by the function $h(i, j)$. Without any specific assumptions on the structure of the dependencies, we cannot provide a general algorithm for finding an optimal permutation. However, we can determine the maximum strength of the dependency of an investment alternative from all others and build a permutation based on this measure. This greedy algorithm ensures that the genes of the more dependent investment alternatives are located closely to each other, and it has a very low computational complexity compared to combinatorial problems arising from the question of finding the best of $m!$ possible permutations.

Definition 11. The maximum strength of the dependency of investment alternative i from all other investment alternatives $j \neq i$ is given by $s(i) := \max_{j \neq i} \{h(i, j)\}$.

We can build our requested permutation $\pi(i)$ by calculating and sorting these strength values.

Lemma 2. For the given graph $G = (V, E)$ and the function h from Definition 1, we can calculate a permutation $\pi(i)$ of the portfolio data based on the strength from Definition 11 in $O(m^2 H)$ computations where H is the number of necessary steps to compute $h(i, j)$.

Proof. We construct a greedy algorithm *Perm* that takes each vertex $i \in V$ once and computes the (m-1) values of $h(i, j)$ for each $j \in V, j \neq i$ to find the maximum $s(i)$ for the given i. Since G has m vertices, the computational complexity of this operation is $O(m^2 H)$.
Afterwards, algorithm *Perm* sorts the m number pairs $(s(i),i)$ in ascending order using the $s(i)$ values as primary sorting criterion. This operation requires $O(m \log m)$ computational steps. The sorted array of number pairs $(s(i), i)$ represents the permutation. If k is the index of $(s(i), i)$ after sorting, the permutation of i is $\pi(i) := k$. The overall complexity of the algorithm *Perm* is $O(m^2 H)$.

If our algorithm is provided with all values of $h(i, j)$ at its start, e. g. this is the case if pairwise correlations between investment alternatives are specified, then the calculation of the permutation requires only m^2 computational steps.

Of course, the above considerations are also applicable to portfolio credit risk problems having continuous decision variables, and they can be adapted to other choices of crossover operators. After discussing the genetic modelling of portfolio credit risk problems, we will now give an overview of our hybrid approach for solving Problem 1 that also incorporates the algorithm from the proof of Lemma 2 as a preprocessing stage.

2.4 Overview of our Hybrid Multi-Objective Algorithm (HMOEA)

Since many of the general MOEA concepts in the literature were designed and tested for optimisation problems having continuous decision variables and do not respect structural properties of our Problem 1, we have designed a problem-specific algorithm that provides a framework for finding constrained Pareto-efficient credit portfolio structures using non-linear, non-convex downside risk measures and discrete decision variables. Figure 1 shows an overview of our Hybrid Multi-Objective Evolutionary Algorithm (HMOEA).

HMOEA

Input: e, p, r, h, w, K

```
Define gene position of each investment alternative $i$ according to permutation $\pi(i)$ based on dependency structure
$t := 0$
Generate initial population $P(t)$
Initialise elite population $Q(t) := \emptyset$
Evaluate $P(t)$
Repeat
    Select individuals from $P(t)$
    Recombine selected individuals (variation operator 1)
    Mutate recombined individuals (variation operator 2)
    Apply local search to mutated individuals (variation operator 3)
    Create offspring population $P'(t)$ (individuals modified by variation operators)
    Evaluate joint population $J(t) := P(t) \cup P'(t)$
    Update elite population $Q(t)$ from $J(t)$
    Generate $P(t+1)$ from $J(t)$
    $t := t + 1$
Until $Q(t) = Q(\max\{0, t - t_{diff}\}) \vee t > t_{max}$
```

Output: $Q(t)$

Figure 1. HMOEA scheme

The first operation in our HMOEA scheme is the permutation of the given investment alternatives according to our considerations from section 2.3 to obtain an adequate genetic representation of our decision variables. After that, the initial population $P(0)$ will be generated by random initialisation of every individual be obtain a diverse population in the search space of potential solutions.

We propose the use of an elite population $Q(t)$ in our algorithm that contains the feasible, non-dominated solutions found so far at each population step t. It is described in more detail in a dedicated section below. At the start of the algorithm, it is empty.

The evaluation of $P(t)$ in line 5 and $J(t)$ in line 12 is based on the non-domination concept proposed in Goldberg (1989), p. 201 and explicitly formulated for constrained problems e. g. in Deb (2001). In our context, it leads to the following type of domination check (cf. Deb (2001), p. 288) which extends Definition 9 by the cases (3) and (4) below.

Definition 12. Given are two distinct portfolio structures x and y. x constraint-dominates y if and only if one of the following cases is true:

(1) $cap(x,w) \leq K$ and $cap(y,w) \leq K$ and $ret(x,p,r) > ret(y,p,r)$ and $risk(x,p;h) \leq risk(y,p;h)$
(2) $cap(x,w) \leq K$ and $cap(y,w) \leq K$ and $ret(x,p,r) \geq ret(y,p,r)$ and $risk(x,p;h) < risk(y,p;h)$
(3) $cap(x,w) \leq K$ and $cap(y,w) > K$
(4) $cap(x,w) > K$ and $cap(y,w) > K$ and $cap(x,w) < cap(y,w)$.

If x constraint-dominates y, we will denote this relationship by $x >_c y$.

The first two cases in Definition 12 refer to the cases from Definition 9 where only feasible solutions were considered. Case (3) expresses a preference for feasible over infeasible solutions and case (4) prefers the solution that has lower constraint violation.

The non-dominated sorting procedure in our HMOEA uses the dominance criterion from Definition 12 to classify the solutions in a given population, e. g. $P(t)$, into different levels of constraint-domination. The best solutions which are not constraint-dominated by any other solution in the population, obtain fitness value 1 (best rank). After that, only the remaining solutions are checked for constraint-domination, and the non-constraint-dominated solutions among these obtain fitness value 2 (second best rank). This process is repeated until each solution has obtained an associated fitness rank.

In line 7 from figure 1, the selection operator is performed using a binary tournament based on Definition 12. Two individuals x and y are randomly drawn from the current population $P(t)$, using uniform probability of $p_{sel} := \frac{1}{|P(t)|}$ for each individual. These individuals are checked for constraint-domination according to Definition 12 and if, without loss of generality, $x >_c y$ then x wins the tournament and is considered for reproduction. If none of the two solutions dominates the other, they cannot be compared using the constraint-domination criterion, and the winning solution is finally determined using a draw from an uniform distribution over both possibilities.

The first two variation operators are the standard one point crossover and the standard mutation operator as described in section 2.2. Our third variation operator in line 10 of figure 1 represents a problem-specific local search procedure that is applied with probability p_{local} to each selected solution x after crossover and mutation. This local search procedure can exploit the structure of a given solution x to perform an additional local optimisation of x towards the global, feasible Pareto-efficient set, e. g. by using a so-called hill climbing algorithm that changes x according to local information about our objective functions in the region around x. We consider this to be a significant improvement compared a standard, non-hybrid MOEA since the randomised search process of the MOEA can be guided a bit more towards the global, feasible Pareto-efficient set and therefore, such a local search operator can improve the convergence speed of the overall algorithm towards the desired solutions. This is particularly important for real-world applications, where speed matters when large portfolios are to be considered. In addition to these arguments, some portfolio credit risk models provide additional local structure information for a current solution x beyond the objective function values that can be exploited very efficiently from the perspective of computational complexity. An example underlining this fact will be provided in the third section of this article.

By applying the variation operators to the selected individuals we obtain an offspring population $P'(t)$. The members of the joint population $J(t)$ containing all parent solutions from $P(t)$ and all offspring solutions from $P'(t)$ are evaluated using the non-dominated sorting procedure described above.

In the next step, the elite population $Q(t)$ is updated according to our algorithm described in section 2.5 below.

Before finishing the population step t and setting $t \rightarrow t+1$ the members of the new parent population $P(t+1)$ have to be selected from $J(t)$ since $|J(t)| > |P(t+1)|$ by definition of $J(t) := P(t) \cup P'(t)$. Since elitist EAs, which preserve the best solutions from both parents and offspring, usually show better convergence properties, we also use this mechanism in our algorithm. Besides elitism, we also need a diversity preserving concept to achieve a good distribution of the solutions in the whole objective space. We incorporate the concept of crowding-sort proposed in Deb (2001), p. 236. This diversity-preserving mechanism is favourable over other proposals, e. g. niche counting based on Euclidean ε-regions in the decision variable space or the objective function space since the crowding-sort does not require an additional parameter e which is difficult to estimate particularly for our discrete, non-linear and non-convex Problem 1, and in our case the crowding sort has a smaller computational complexity of $O(|J(t)| \log |J(t)|)$ compared to the quadratic complexity which is required by other mechanisms (cf. Deb (2001), p. 237).

The algorithm is terminated if $Q(t)$ has not been improved for a certain number t_{diff} of population steps or if a maximum number of t_{max} population steps has been performed.

2.5 Some considerations concerning the elite population

The elite population $Q(t)$ is updated each time after performing the genetic variation operators and evaluation of the joint population $J(t)$. It is external since the solutions stored in $Q(t)$ do not influence the solutions in the other populations which are modified by the EA. In other proposed MOEA schemes, the members of the elite population are also considered for reproduction into $P(t+1)$. Since our Problem 1 has many local optima we have to ensure that the selection pressure of our combined elitism and diversity-preserving mechanism which is applied when selecting individuals from $J(t)$ for survival into $\mathrm{P}(t+1)$ is not too strong. In that case, the algorithm could get stuck in local optima and might not find a diverse global Pareto-efficient set. Therefore, we avoid such an influence from the members of $Q(t)$ by not considering them for reproduction into $P'(t)$.

In our algorithm, the maintenance of such an elite population has many advantages, e. g. with respect to the structure of Problem 1 and some real-world optimisation requirements:

- Rudolph & Agapie (2000) have shown that the existence of such an elite population ensures convergence of an EA to the global Pareto-efficient set (in their work the elite population it is called ‘archive population’).
- The number of individuals in the population $P(t)$ that has to be chosen a priori before running the algorithm, is limiting the number of best solutions to be kept by the algorithm during time. If we maintain an elite population, the choice of the size of $P(t)$ is not crucial concerning the number of solutions to be kept during the evolution process.
- The algorithm can be terminated at any time by the user without losing the best feasible solutions found so far. This enables real-time optimisation tasks. Remember that we consider constrained optimisation problems so current members of the population $P(t)$ might not be feasible at an arbitrary interruption time.

- Since we have to deal with a discrete non-linear, non-convex problem, which can have many local optima that are not uniformly distributed in the two-dimensional objective function space, it is difficult to satisfy both the requirement of finding a well-distributed set of diverse solutions, and the requirement of approximating the largest possible set of feasible, global Pareto-efficient portfolio structures if we use only one population. We will present an example distribution of the maximum Pareto-efficient set in our empirical study described later to visualize these facts.
- An external elite population provides a good basis for deciding about termination of the algorithm since we can terminate the algorithm if no change of the elite population has occurred for a certain number of population steps.
- In case of the absence of an elite population, the population size of $P(t)$ has to be large enough to ensure that the best solutions found so far survive into the next population. The computational cost of updating the elite population are lower compared to maintaining larger population size for the populations $P(t)$ or $P'(t)$ (and therefore, also for $J(t)$), see below.

We will now regard the computational cost of maintaining the elite population.

Lemma 3. If $k_1 := |Q(t)|$ is the size of the elite population and $k_2 := |J^*(t)|$ is the size of the subset of feasible, non-dominated individuals from the joint population $J^*(t) \subseteq J(t)$ at time step t, the elite population can be updated in $O(k_1\, k_2)$ operations.

Proof. Consider the update procedure shown in figure 2.

```
Update procedure for Q(t)
Input: $Q(t)$, $J^*(t)$
For each $x \in J^*(t)$
    Dominated := False
    For each $y \in Q(t)$
        If $x >_c y$ Then
            $Q(t) := Q(t) \setminus \{y\}$
        End If
        If $y >_c x$ Then
            Dominated := True
        End If
    End For
    If Dominated = False Then
        $Q(t) := Q(t) \cup \{x\}$
    End If
End For
Output: $Q(t)$
```

Figure 2. Update procedure for $Q(t)$

Each $x \in J^*(t)$ is processed once (k_2 loop iterations) and the second loop is run for each $y \in Q(t)$ (k_1 loop iterations). Therefore, we require $O(k_1\, k_2)$ computational steps to update $Q(t)$.

At this point, we want to point out that non-dominated sorting of $k_3 := |J(t)|$ elements of the joint population $J(t)$ requires $O(k_3^2)$ operations in the worst case (cf. the computational complexity for non-dominated sorting derived in Deb (2001), p. 206). In the absence of an elite population, k_3 has to be chosen large enough to keep the non-dominated, feasible individuals in all population steps: $k_3 \geq k_1$.

Of course, we also need the non-dominated sorting procedure if we have an elite population, but we can choose $k_3 << k_1$ without losing the best solutions and the desired convergence to the global feasible Pareto-efficient set so the overall computational complexity of each population step $t \rightarrow t+1$ in our algorithm is lower.

We will now provide some references to related research in the area of portfolio optimisation problems, particularly using downside risk measures.

2.6. Previous approaches to similar problems

In the existing literature, the main focus of portfolio selection and optimisation has been in the area of stock portfolio investments where Markowitz (1952) created the standard framework for calculating efficient frontiers of investment alternatives. Based on his mean-variance approach, many different calculation procedures have been suggested, see e. g. Elton & Gruber (1995) for an overview.

For downside risk measures like the Value-at-Risk which is similar to the Credit-Value-at-Risk from Definition 6, different approaches have been proposed even in a single objective function setting where the expected return from a portfolio is to be maximised with respect to a fixed level of downside risk. This is due to the mathematical properties of such percentile-based downside risk measures, cf. our remarks and references in section 2.1. For example, Gilli & Kellezi (2000) used the Threshold Accepting heuristic to approximate the efficient set of a stock portfolio in a Value-at-Risk based setting. A comparison of different heuristic approaches to constrained stock portfolio optimisation problems was e. g. performed by Chang et al. (2000).

Concerning credit portfolios, Andersson & Uryasev (1999) proposed the use of simplex algorithms under a tail conditional expectation risk measure (Conditional Value-at-Risk) in a simulation model framework. Lehrbass (1999) proposed the use of Kuhn-Tucker optimality constraints for a credit portfolio optimisation problem having real-valued variables.

In the next section we will describe more details of our implementation of the hybrid evolutionary framework and present results of an empirical study where the performance of the implementation is investigated.

The first work proposing the use of Evolutionary Algorithms for solving credit portfolio optimisation problems related to Problem 1 was Schlottmann & Seese (2001a). In that work, a hybrid EA was introduced to solve a constrained optimisation problem that was build upon a single objective function combining the aggregated return and the aggregated risk of a credit portfolio.

3 An implementation and empirical test of the HMOEA framework using CreditRisk+

3.1 Portfolio credit risk models and overview of CreditRisk+

When implementing an algorithm for risk-return optimisation of credit portfolios, the first question is how to model the dependencies between joint changes of the credit quality of different investment alternatives, so we have to choose a model that provides us a function *h* according to Definition 1 and beyond that, a calculation or approximation procedure for the cumulative distribution of aggregated losses from given portfolio data.

In the literature, there are different alternatives for modelling the dependencies between obligors and for calculating portfolio credit risk measures. Among these alternatives, CreditMetrics (see Gupton et al. (1997)), CreditRisk+ (see CreditSuisse Financial Products (1997)) Wilson's model (see Wilson (1997a, 1997b) and the KMV option based approach (see Kealhofer (1998)) are intensively discussed in many academic and application-oriented publications. Since we will set our focus on the default risk of loan portfolios in our empirical study described later, we will concentrate on CreditRisk+. However, our hybrid framework is compatible with any other portfolio credit risk model providing a loss (in case of a default mode model) or a profit-loss distribution (in case of a mark-to-market model).

In the following paragraphs, we will give a brief description of the CreditRisk+ General Sector Analysis model for a one year horizon here that concentrates on the main issues concerning our algorithm (see CreditSuisse Financial Products (1997), pp. 32-57 for a more detailed derivation of the model). It is an actuarial approach that uses an intensity based modelling of defaults, i. e. the default of each obligor in the portfolio is considered to be a stopping time of a hazard rate process expressed by a Poisson-like process. In case of a default event, the amount of credit exposure (net exposure) lent to the defaulting obligor will be entirely lost.

Given is the data from Definition 1 of *m* different obligors in the portfolio. Particularly, each obligor has a net exposure e_i, an associated annual mean default rate p_i (typically, p_i is small: $0 < p_i < 0.1$) and an annual default rate volatility $\sigma_i \geq 0$. Furthermore, there is a total of *n* independent sectors as common risk factors, where the first sector ($k = 1$) is obligor specific, i. e. in this sector there is no implicit default correlation between obligors ($k = 1,..,n$ below unless otherwise noted). The obligors are allocated to the sectors according to sector weights $\Theta_{ik} \in [0,1], \forall i: \sum_{k=1}^{n} \Theta_{ik} = 1$.

The probability generating function (abbreviated PGF) for the losses from the entire portfolio is defined by

$$G(z) := \sum_{i=0}^{\infty} prob(\text{aggregated losses} = i \cdot L) \cdot z^i \quad (3.1)$$

where *L* is a constant defining net exposure bands of constant width and *prob*(...) represents the probability of losing *i* times the value of *L* from the whole portfolio.

Since the sectors are independent this can be decomposed to

$$G(z) = \prod_{k=1}^{n} G_k(z) \quad (3.2)$$

where $G_k(z)$ is the PGF for the losses from the portfolio in sector *k*.

To obtain the approximated cumulative loss distribution function for the portfolio a recurrence relation, the recursion by Panjer (1981), can be applied to evaluate the coefficients

of the PGF (for a more detailed background see Panjer & Willmot (1992)). After that, risk figures, e. g. the 99th percentile, can be calculated.
An interesting feature of the model concerning the portfolio optimisation task are the marginal risk contributions of obligor *i* to the standard deviation of portfolio credit risk:

$$RC_i^{\sigma} := e_i \frac{\partial \sigma_{pf}}{\partial e_i} = \frac{e_i p_i}{\sigma_{pf}} \left(e_i + \sum_{k=1}^{n} \left(\frac{\sigma_k}{\mu_k} \right)^2 e_i p_i \Theta_{ik} \right) \quad (3.3)$$

where σ_{pf} is the portfolio standard deviation derived from the PGF of the portfolio losses, μ_k, σ_k are sector specific parameters calculated directly from the input parameters $e_i, p_i, \sigma_i, \Theta_{ik}$ using formula (3.4) below (note that $\sigma_1 = 0$ by definition of sector 1).

$$\forall k : \mu_k := \sum_{i=1}^{m} \Theta_{ik} p_i, \sigma_1 := 0, \forall k > 1 : \sigma_k := \sum_{i=1}^{m} \Theta_{ik} \sigma_i \quad (3.4)$$

Alternatively, by setting $\sigma_k := \omega_k \mu_k$ for $k > 1$ using parameters (variation coefficients) $\omega_k, k = 2,...,n$ only μ_k has to be calculated according to (3.4) and in this case, no obligor-specific default rate volatilities σ_i are required to calculate the sector specific parameters.
To calculate an approximation for the risk contribution e. g. to the 99th percentile, a scaling factor is defined in the following manner:

$$\xi_{pf} := \frac{q_{pf}^{0.99} - \mu_{pf}}{\sigma_{pf}} \quad (3.5)$$

where $\mu_{pf}, \sigma_{pf}, q_{pf}^{0.99}$ are the expectation, standard deviation and 99th percentile of the portfolio loss distribution, respectively.
The figures calculated by applying formula (3.3) can be used as a basis for the approximate risk contribution to the 99th percentile by scaling the risk contribution obtained from (3.3) according to ξ_{pf} and adding it to the obligor specific expected loss:

$$RC_i^{0.99} := e_i p_i + \xi_{pf} RC_i^{\sigma} \quad (3.6)$$

We use these figures to ensure a computationally efficient calculation within our local search variation operator in the HMOEA implementation described in the next subsection. The calculation of (3.6) for all investment alternatives $i \in \{1,...,m\}$ requires only $O(m\ n)$ additional operations after the calculation of the coefficients of the PGF from formula (3.2) which is mandatory to evaluate the *risk*() target function for each individual. Note that the number of sectors *n* is constant in a given problem instance and usually small ($n < 10$), so the computation of (3.6) requires only linear computing time measured by the number of investment alternatives *m*.

3.2 Further implementation details referring to the HMOEA scheme

In the CreditRisk+ model, the volatilities of the obligors' default probabilities in conjunction with the common risk factors of all obligors replace a direct modelling of the default correlation $\rho(i, j)$ for two investment alternatives *i*, *j*. Therefore, for the calculations according to this portfolio credit risk model, no explicit default correlations are required. However, in CreditSuisse Financial Products (1997), p. 56ff. the following implicit default correlation formula is provided:

$$\rho(i, j) \approx \sqrt{p_i p_j} \sum_{k=1}^{n} \Theta_{ik} \Theta_{jk} \left(\frac{\sigma_k}{\mu_k} \right)^2 \quad (3.7)$$

By setting $h(i, j) := \rho(i, j)$ according to (3.7), we obtain the complete dependence structure required for our Definition 1 and the subsequent results from our theoretical considerations in the first section of this article. Moreover, this explicit definition of the dependence structure can be exploited for the calculation of the permutation $\pi(i)$ according to Lemma 2 that provides us an adequate genetic modelling of the decision variables for the given portfolio data. Since one calculation of $\rho(i, j)$ for given, fixed values of i and j requires $O(n)$ computational steps and n is bounded, the computational cost of calculating $\pi(i)$ is $O(m^2\, n)$ according to Lemma 2.

To create a local search operator required by an implementation of the HMOEA scheme, we use the following local search target function that uses the quotient between aggregated net return and aggregated risk to evaluate a given portfolio structure x:

$$f\left(x, p, r; h\right) := \frac{ret(x, p, r)}{risk(x, p; h)} \quad (3.8)$$

Considering Definitions 4 and 6 as well as the CreditRisk+ calculation method for the 99th percentile $q_{pf}^{0.99}\left(x, p, \sigma, \Theta\right)$ of the cumulative distribution of aggregated losses from the portfolio structure x under the given data p, σ, Θ, r the function f can be written as:

$$f(x, p, \sigma, \Theta, r) := \frac{\sum_{i=1}^{m} x_i (r_i - p_i)}{q_{pf}^{0.99}\left(x, p, \sigma, \Theta\right) - \sum_{i=1}^{m} x_i p_i} \quad (3.9)$$

If we maximise this function f we will implicitly maximise $ret(x, p, r)$ and minimise $risk(x, p; h)$, and this will drive the portfolio structure x towards the set of global Pareto-efficient portfolio structures (cf. the domination criteria specified in Definition 9). In addition to that, we have to consider our constraints to ensure the local search variation operator keeps the portfolio structure x feasible or moves an infeasible portfolio structure x back into the feasible region. An overview of our local search operator scheme based on these considerations is shown in figure 3.

The partial derivative d_j for obligor j required in line 12 of figure 3 can be calculated using the following formula (a proof is provided in the appendix):

$$d_j := \frac{x_j \left(r_i - p_i\right)\left(\xi_{pf}\sigma_{pf}\right) - \sum_{i=1}^{m} \left(r_i - p_i\right) x_i \left(\xi_{pf} RC_j^{\sigma}\right)}{x_j \left(\xi_{pf}\sigma_{pf}\right)^2} \quad (3.10)$$

If the current solution x from $P(t)$ to be optimised with probability p_{local} is infeasible because the capital restriction is violated (cf. line 2 in figure 3), the algorithm will remove the investment alternative having the minimum gradient component value from the portfolio (lines 14 and 15). This condition drives the hybrid search algorithm towards feasible solutions. In case of a feasible solution that is to be optimised, the direction of search for a better solution is determined by a draw of a uniformly distributed (0,1)-random variable (cf. lines 5 and 6). This stochastic behaviour helps preventing the local search variation operator from stalling into the same local optima. The local search algorithm terminates if the current solution cannot be modified further, if it is already included in the populations $P(t)$ or $Q(t)$ or if no improvement considering the violation of constraints or the target function can be made. Remembering the fact that the risk contributions, and therefore, the partial derivatives d_j can be calculated in linear time for an individual which has already a valid fitness evaluation this yields a very fast variation operator.

Local search operator

```
Input: $e, p, \sigma, \Theta, r, w, K, P(t)$
For each $x \in P(t)$ apply the following instruction block with probability $p_{local}$
    If $cap(x, w) > K$ Then
        $D := -1$
    End If
    If $cap(x, w) \leq K$ Then
        Choose between $D := 1$ or $D := -1$ with uniform probability 0.5
    End If
    Initialisation $\forall i : x_i := x_i$
    Do
        Copy $\forall i : x_i := x_i$
        Calculate $ret_{old} := \sum_{i=1}^{m} x_i(r_i - p_i)$ and $risk_{old} := q_{pf}^{0.99}\left(x, p, \sigma, \Theta\right) - \sum_{i=1}^{m} x_i p_i$
        For each $x_j$ calculate the partial derivatives $d_j := \frac{\partial}{\partial x_j} f\left(x, p, \sigma, \Theta, r\right)$
        If $D = -1$ Then
            Choose the minimal gradient component $i := \arg\min_j \left\{ d_j \middle| x_j > 0 \right\}$ of
            exposures currently remaining in the portfolio
            Remove this exposure from portfolio: $x_i := 0$
        Else
            Choose the maximal gradient component $i := \arg\max_j \left\{ d_j \middle| x_j = 0 \right\}$ of
            exposures currently removed from the portfolio
            Add this exposure to portfolio: $x_i := e_i$
        End If
        Calculate $ret_{new} := \sum_{i=1}^{m} \hat{x}_i(r_i - p_i)$ and $risk_{new} := q_{pf}^{0.99}\left(\hat{x}, p, \sigma, \Theta\right) - \sum_{i=1}^{m} \hat{x}_i p_i$
    While $\left(\exists i : \hat{x}_i > 0\right) \wedge \left(\exists j : \hat{x}_j = 0\right) \wedge \hat{x} \notin P(t) \wedge \hat{x} \notin Q(t) \wedge$
    $\left(\left(D = -1 \wedge cap(\hat{x}, w) > K\right) \vee \left(D = 1 \wedge cap(\hat{x}, w) \leq K \wedge \left(ret_{new} > ret_{old} \vee risk_{new} < risk_{old}\right)\right)\right)$
    Replace $x$ in $P(t)$ by its optimised version
End For
Output: $P(t)$
```

Figure 3. Local search operator scheme

The next subsection contains the parameters and test cases for an empirical test of the implemented hybrid framework for credit portfolio risk-return analysis and optimisation.

3.3 Specification of test cases, parameters and performance criteria

Besides our Hybrid Multi-Objective Evolutionary Algorithm (HMOEA), we have also implemented a simple enumeration algorithm that investigates all possible portfolio structures to determine the feasible global Pareto-efficient set *PE** having maximum cardinality for small instances of our Problem 1, i. e. the latter algorithm serves as a proof for the globally optimal portfolio structures that should be discovered by the other search algorithms. For all instances considered in this article, we compared the results of the HMOEA to the respective results of a non-hybrid MOEA that incorporates all features of the HMOEA except for the local search operator which is disabled in the non-hybrid algorithm. Particularly, the MOEA also benefits from all problem specific algorithmic features that we have proposed for the HMOEA in the previous sections, e. g. the presence of the elite population and the preprocessing algorithm. All tests of the above implementations were carried out on a standard desktop PC (800 MHz single CPU). For all evolutionary algorithms, we performed 20 independent runs of the same algorithm on the respective test problem using different pseudorandom number generator seeds.

Although more tests cases were examined during development of the system (e. g. for estimating the parameters like $|P(t)|$, p_{cross} etc.) we focus on the following sample loan portfolios in this paper. The structure of these portfolios is analogous to real world data.[2]

Our first test data set is named portfolio m20n2. It consists of $m = 20$ investment alternatives which are allocated to $n = 2$ sectors. The capital budget restriction is assumed to be 50% of the maximum capital requirement that will be required if all investment alternatives are held in the portfolio. The detailed structure of portfolio m20n2 is provided in the appendix.

The medium size portfolio m45n2 contains $m = 45$ investment alternatives allocated to $n = 2$ sectors. The capital restriction is $K := 80000$ which is about 71% of the sum of all investment alternatives' capital requirements.

The largest problem instance named portfolio m100n3 contains $m = 100$ investment alternatives allocated to $n = 3$ sectors. A capital restriction is set to about 56% of the sum of all investment alternatives' capital requirements.

In all test cases, we chose a quite common parameter setting of $p_{cross} := 0.95$ and $p_{mut} := \frac{1}{m}$, which is reported to work well in many other EA studies, and this was also supported by test results during our development of the HMOEA and the non-hybrid MOEA.

The choice of p_{local} can be made by the respective user of the HMOEA depending on his or her preferences: If one is interested in finding better solutions in earlier populations, the probability shall be set higher, and in this case more computational effort is spent by the algorithm on the local improvement of the solutions. However, the local search optimisation pressure should not be too high since one is usually also interested in finding a diverse set of solutions. Therefore, a choice of $0 < p_{local} \leq 0.1$ appears to be adequate, and this is supported by our tests.

For the portfolio m20n2 data set, we chose $|P(t)| := 30$ individuals per population, and $p_{local} := 0.005$. The evolutionary process was stopped after $t_{max} := 1000$ population steps. For the non-hybrid MOEA all these parameters were set equally except for $p_{local} := 0$ which means that there is no third variation operator in the non-hybrid MOEA.

In the portfolio m45n2 test, we set $|P(t)| := 40$ individuals per population and the probability for the third variation operator in the HMOEA was set higher to $p_{local} := 0.05$ due to the larger search space. The other parameters were set analogously to the portfolio m20n2 test case. In addition to the investigation of the detailed results for $p_{local} = 0.05$, we will compare the average results of our chosen performance metrics for different settings of p_{local} to show the influence of the third variation operator on the results in the HMOEA.

In the portfolio m100n3 test, we set $|P(t)| := 50$ individuals per population and the probability for the third variation operator in the HMOEA was set to $p_{local} := 0.1$ to reveal the significant differences between the hybrid and the non-hybrid approach. Again, the other parameters were set according to the portfolio m20n2 test case.

---

[2] All test portfolios can be retrieved via http://www.aifb.uni-karlsruhe.de/CoM/HMOEA/tests.html.

Particularly, to achieve a better comparison between the evolutionary algorithms we used the same initial population for both the HMOEA and the non-hybrid MOEA given a specific pseudorandom generator seed. This means we used the same 20 (randomly determined) initial populations for both algorithms on a test data set to obtain a fair basis for the comparison of the results.

For all test cases, we have calculated performance measures of the algorithms based on the set coverage metric from Zitzler (1999). In our context, the set coverage metric is defined as follows:

Definition 13. Given are two sets of portfolio structures PE_1, PE_2 which are approximations for PE^* defined in Problem 1. The pair of set coverage metric values $C_{1,2} := (C_1, C_2)$ is calculated by

$$C_1 := C(PE_2, PE_1) = \frac{\left|\{x \in PE_1 \mid \exists y \in PE_2 : y >_c x\}\right|}{|PE_1|} \quad (3.11)$$

$$C_2 := C(PE_1, PE_2) = \frac{\left|\{y \in PE_2 \mid \exists x \in PE_1 : x >_c y\}\right|}{|PE_2|} \quad (3.12)$$

This metric provides us a criterion for comparing two different sets of solutions produced by different algorithms. We have chosen this metric since it allows the comparison of approximation sets having different cardinalities, and particularly in our larger test cases, we do not need PE^* for the evaluation of the results. An algorithm 2 calculating PE_2 is considered to be better in convergence to PE^* than an algorithm 1 that computes PE_1 if $C_1 > C_2$, i. e. if the fraction of solutions in PE_2 which are dominated by solutions from PE_1 is smaller than the fraction of solutions in PE_1 that are dominated by solutions from PE_2. To be more transparent, we investigate both the nominator and the denominator of (3.11) and (3.12) separately. Therefore, two important goals of multi-objective approximation algorithms are evaluated: Finding an approximation set whose elements are very close to corresponding members of PE^* and which also has a high cardinality. So we can compare both the quantity and the quality of two alternative approximations for PE^*.

In addition to the evaluation of these goals, we compare the maximum spread (cf. Zitzler (1999)) for each calculated approximation set for PE^* according to the next definition.

Definition 14. Given is an approximation set of portfolio structures PE_1 for PE^* defined in Problem 1. The maximum spread value $\delta(PE_1)$ is obtained by evaluation of

$$\delta(PE_1) := \sqrt{\left(\max_{x \in PE_1}\left(ret(x,p,r)\right) - \min_{x \in PE_1}\left(ret(x,p,r)\right)\right)^2 + \left(\max_{x \in PE_1}\left(risk(x,p;h)\right) - \min_{x \in PE_1}\left(risk(x,p;h)\right)\right)^2} \quad (3.13)$$

The maximum spread allows a comparison between different approximation sets based on the largest Euclidean distance between two solutions in the two-dimensional objective function space. We have chosen this additional metric because the set coverage metric does not cover the largest spread between the found solutions which is also a goal in multi-objective optimisation. A larger spread is preferable, i. e. an approximation set PE_1 is better than another set PE_2 concerning this criterion if $\delta(PE_1) > \delta(PE_2)$.

Of course, we calculate the set coverage metric and the maximum spread from the members of the elite population $Q(t)$ for a fixed value t after running the respective algorithms. We will present the results in the next subsection.

3.4 Empirical results

In all test cases, the approximation set calculated by the non-hybrid MOEA is denoted by PE_1, the approximation set from running the HMOEA is denoted by PE_2.

First of all, for the portfolio m20n2 test data set, we compare the result PE_2 of one HMOEA run to PE^* which was obtained by a complete enumeration of the search space that required approximately 72 minutes. In contrast to this, each run of the HMOEA (as well as a run of the non-hybrid MOEA) required about 3 minutes for the computation of an approximation set PE_2. Figure 4 shows both results.

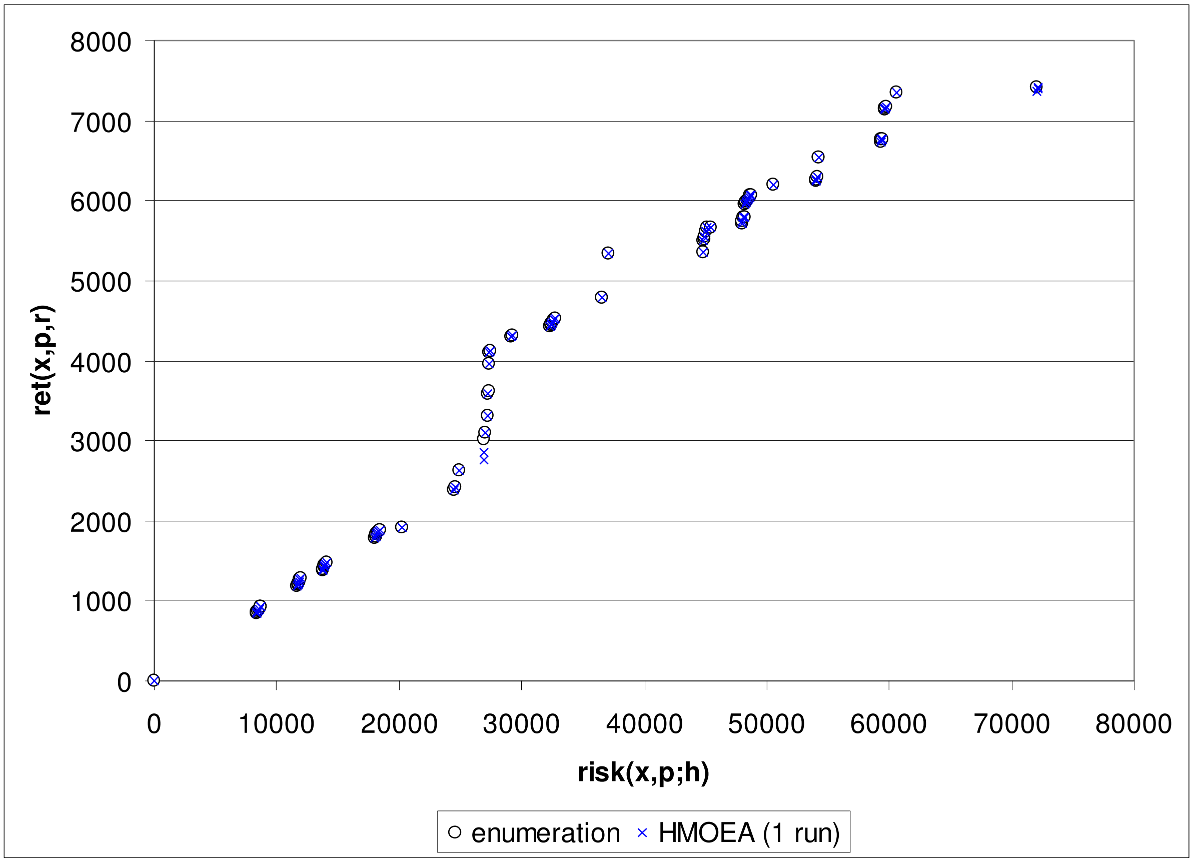

Figure 4. Comparison of PE^* and PE_2 for portfolio m20n2

It is easy to check by visual inspection that PE_2 is a good approximation set for PE^* since all points of PE^* (indicated by circles) are approximated by mostly identical or at least very close points of PE_2 which are marked by a respective 'x' in figure 4.

The table 1 on the following page shows the detailed results of the HMOEA and the non-hybrid MOEA for this small portfolio. The results indicate that both algorithms find quite similar solutions as we expect it when considering the very small local search variation operator probability $p_{local} = 0.005$ for the HMOEA in this case. However, the quality of the solutions found by the HMOEA concerning the set coverage metric is on average a bit better than the quality of the solutions by the non-hybrid algorithm which is indicated by the smaller number of dominated solutions in PE_2 (column 3) compared to PE_1 (column 2). Furthermore, the number of runs where less solutions from PE_2 are dominated by solutions from PE_1 (10 runs) is higher than vice versa (6 runs). In addition to this slightly better performance, the HMOEA found more solutions on average and in more runs (8 runs) than the other algorithm (7 runs), therefore the average value of the set coverage metric is about two times better for the HMOEA. Concerning the maximum spread values, both algorithms yielded quite the same results, but the non-hybrid approach is slightly better in this criterion due to the higher number of runs (6 versus 4) where it had higher spreads than the HMOEA.

no. of run	**nom. of C_1**	**nom. of C_2**	**denom. of C_1**	**denom. of C_2**	C_1	C_2	$\delta(PE_1)$	$\delta(PE_2)$
1	**0**	2	**81**	80	**0.0000**	0.0250	51337.58	51337.58
2	2	**1**	75	**79**	0.0267	**0.0127**	51337.58	51337.58
3	1	1	77	**81**	0.0130	**0.0123**	51337.58	51337.58
4	**1**	3	**79**	78	**0.0127**	0.0385	51191.22	**51337.58**
5	**0**	1	76	76	**0.0000**	0.0132	51337.58	51337.58
6	3	3	77	**79**	0.0390	**0.0380**	51191.22	**51337.58**
7	6	**1**	**81**	77	0.0741	**0.0130**	**51337.58**	51191.22
8	3	**0**	80	**81**	0.0375	**0.0000**	51337.58	51337.58
9	5	**1**	**80**	76	0.0625	**0.0132**	**51337.58**	51191.22
10	**1**	3	76	**78**	**0.0132**	0.0385	51191.22	**51337.58**
11	**1**	2	**81**	77	**0.0123**	0.0260	51337.58	51337.58
12	6	**0**	75	**79**	0.0800	**0.0000**	**51337.58**	51191.22
13	1	**0**	77	77	0.0130	**0.0000**	51191.22	51191.22
14	1	1	80	80	0.0125	0.0125	51337.58	51337.58
15	2	**0**	72	**80**	0.0278	**0.0000**	51285.10	**51337.58**
16	4	**1**	**80**	76	0.0500	**0.0132**	**51337.58**	51191.22
17	4	**0**	**79**	76	0.0506	**0.0000**	**51337.58**	51191.22
18	**0**	1	77	77	**0.0000**	0.0130	51337.58	51337.58
19	2	2	79	**83**	0.0253	**0.0241**	51337.58	51337.58
20	1	**0**	77	**78**	0.0130	**0.0000**	51191.22	51191.22
average	2.20	**1.15**	77.95	**78.40**	0.0282	**0.0146**	**51298.37**	51286.35

Table 1. Comparison of PE_1 and PE_2 for portfolio m20n2 (better values are in bold face)

Summarising the results in the small constrained test case, both algorithms found a good approximation set for PE^* within a few minutes. The HMOEA shows a higher convergence speed but this is at the cost of a slightly reduced maximum spread of the approximation set compared to the non-hybrid algorithm in our test runs. This is mainly due to the fact that the intended higher convergence pressure towards feasible, global non-dominated solutions caused by the local search operator leads to early discovery of isolated Pareto-optimal solutions which might strongly dominate the population in the relatively small search space. However, this is not a general disadvantage of the hybrid algorithm, since we have to remind at this point that there is a trade-off between the two goals of finding globally optimal solutions very fast and discovering a diverse set of solutions, and this conflict is to be faced by any algorithm that solves instances of Problem 1. We have put more weight on the first criterion in conjunction with the discovery of feasible solutions during development of the hybrid algorithm, and the maximum spread of the HMOEA is very close to the globally optimal maximum spread of PE^*, so the slightly lower maximum spread of PE_2 is not critical.

Beyond that, we will show now that in the other test cases, which have larger search spaces that grow exponentially in the number of investment alternatives m, the hybrid approach can exploit its advantages more significantly and usually yields both a better set coverage metric value and a higher maximum spread. To underline this claim, table 2 shows the results in the portfolio m45n2 test case. A run of the non-hybrid MOEA required about 7 min. 30 sec. to compute the approximation set PE_1 in this setting, and the HMOEA required approximately 8 minutes to calculate PE_2.

no. of run	**nom. of C_1**	**nom. of C_2**	**denom. of C_1**	**denom. of C_2**	C_1	C_2	$\delta(PE_1)$	$\delta(PE_2)$
1	118	**67**	428	**446**	0.2757	**0.1502**	65566.46	**70334.59**
2	157	**73**	**450**	449	0.3489	**0.1626**	**73449.55**	73330.93
3	**80**	88	**450**	445	**0.1778**	0.1978	66534.15	**70936.96**
4	160	**67**	450	450	0.3556	**0.1489**	67954.26	**73449.55**
5	115	**62**	417	**448**	0.2758	**0.1384**	65866.56	**73330.93**
6	129	**48**	441	**450**	0.2925	**0.1067**	68265.32	**70334.59**
7	120	**72**	450	450	0.2667	**0.1600**	66537.58	**70936.96**
8	151	**76**	450	450	0.3356	**0.1689**	68940.85	**70936.96**
9	137	**61**	**450**	447	0.3044	**0.1365**	67126.12	**73330.93**
10	136	**57**	449	**450**	0.3029	**0.1267**	65764.79	**72536.10**
11	130	**72**	**450**	442	0.2889	**0.1629**	66349.65	**70936.96**
12	119	**79**	437	**450**	0.2723	**0.1756**	66236.72	**70334.59**
13	130	**84**	**450**	430	0.2889	**0.1953**	68427.92	**70334.59**
14	170	**58**	**450**	447	0.3778	**0.1298**	68010.23	**73475.61**
15	121	**65**	**450**	446	0.2689	**0.1457**	69228.15	**72732.18**
16	139	**71**	436	**449**	0.3188	**0.1581**	69674.10	**70936.96**
17	140	**96**	448	**450**	0.3125	**0.2133**	69223.61	**70334.59**
18	171	**62**	450	450	0.3800	**0.1378**	68316.28	**73112.27**
19	163	**62**	**442**	441	0.3688	**0.1406**	72742.12	**73465.07**
20	110	**85**	442	**450**	0.2489	**0.1889**	66938.45	**70334.59**
average	134.8	**70.25**	444.5	**447**	0.3031	**0.1572**	68057.64	**71772.80**

Table 2. Comparison of PE_1 and PE_2 for portfolio m45n2 (better values are in bold face)

The hybrid approach is better in all averages of the performance metrics for our medium size test case. Except for one of the 20 independent runs, the HMOEA always found remarkably better solutions than the other algorithm (cf. the second and the third column). Moreover, the hybrid algorithm found quite the same number of solutions in all runs like the non-hybrid MOEA, therefore the set coverage metric value is significantly better for the HMOEA due to the better quality of the found solutions. In contrast to the results presented above for the smaller portfolio, the maximum spread values of PE_2 are also much better than the respective values of PE_1 except for one run so the hybrid approach is favourable concerning both performance criteria defined in subsection 3.3.

The influence of the local search variation operator on the results in our medium size test case is indicated in figures 5 and 6 where we have plotted the average performance metric values depending on different settings of p_{local}. We have plotted ordinary least squares (OLS) regression lines in each figure to estimate the linear trend of the performance metric values depending on the choice of of p_{local}.

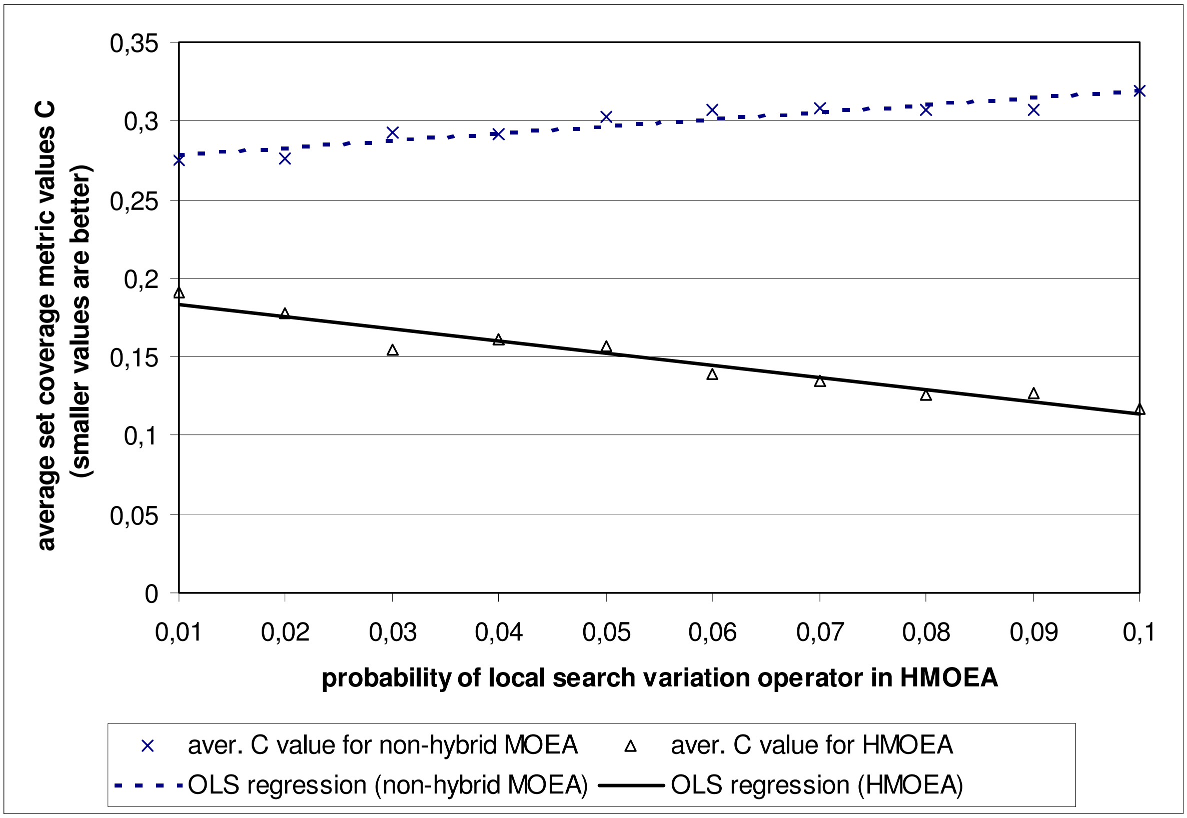

Figure 5. Average set coverage metric values depending on p_{local} for portfolio m45n2

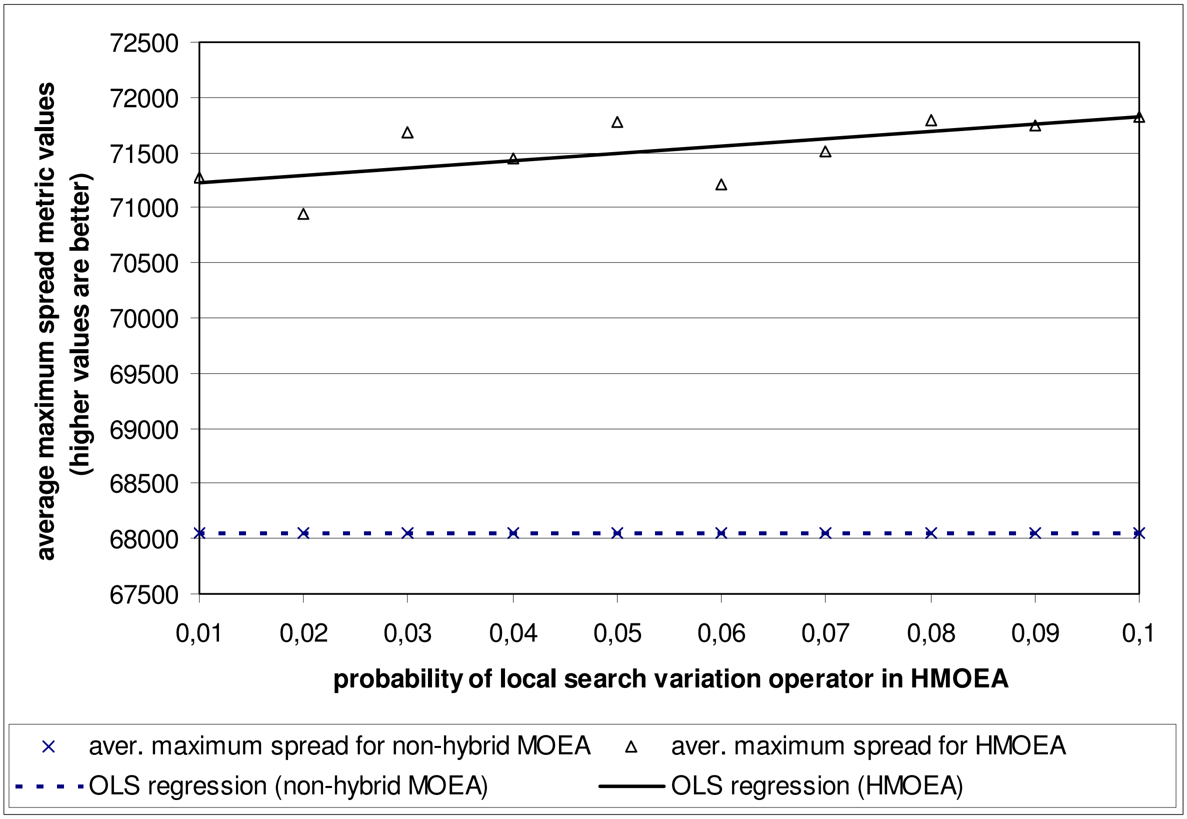

Figure 6. Average maximum spread values depending on p_{local} for portfolio m45n2

Since both algorithms rely on probabilistic variation operators, we cannot expect perfect monotony of the performance metrics depending on the variation of p_{local}. However, the linear regression lines in both figures clearly indicate the influence of the local search variation operator. For the set coverage metric, a higher value of p_{local} typically leads to a higher quality of the solutions discovered by the HMOEA compared to the solutions discovered by the non-hybrid MOEA. Remembering the fact that smaller set coverage metric values are preferable, this is indicated by both the negative slope of the regression line for the set coverage metric values of the HMOEA and the positive slope of the other regression line for the non-hybrid algorithm in figure 6. Of course, different settings of p_{local} do not influence the maximum spread of the non-hybrid MOEA whereas the hybrid algorithm benefits from higher p_{local} values since the slope of the regression line is positive.

Beyond this analysis of the influence of p_{local} on the results, we can see in the portfolio m45n2 test case that the hybrid approach is preferable if the convergence speed, the quality of the found solutions and the maximum spread in the objective function space matters. In addition to our above discussion of the detailed results for $p_{local} = 0.05$, this is underlined by the fact that for each tested value of p_{local} all average values of the performance metrics shown in figure 6 were better in the hybrid case.

For the largest portfolio of our test data sets the results of the HMOEA and the non-hybrid MOEA after $t_{max} = 1000$ population steps are displayed in table 3. A single run of the HMOEA required about 16 minutes for the calculation of PE_2 due to the high value of $p_{local} = 0.1$ which was chosen to reveal the differences between both algorithms, whereas the non-hybrid algorithm terminated within 11 minutes.

no. of run	**nom. of C_1**	**nom. of C_2**	**denom. of C_1**	**denom. of C_2**	C_1	C_2	$\delta(PE_1)$	$\delta(PE_2)$
1	287	**82**	477	**550**	0.6017	**0.1491**	62852.17	**72033.78**
2	267	**100**	497	**536**	0.5372	**0.1866**	69200.63	**71744.75**
3	273	**104**	471	**563**	0.5796	**0.1847**	62622.74	**70284.81**
4	235	**111**	432	**543**	0.5440	**0.2044**	65324.67	**72033.78**
5	249	**108**	487	**554**	0.5113	**0.1949**	61872.88	**70284.81**
6	268	**87**	468	**526**	0.5726	**0.1654**	61469.33	**70284.81**
7	192	**101**	430	**576**	0.4465	**0.1753**	55946.92	**73594.82**
8	235	**118**	447	**570**	0.5257	**0.2070**	61485.23	**70284.81**
9	260	**94**	466	**572**	0.5579	**0.1643**	64166.60	**72033.78**
10	245	**88**	458	**532**	0.5349	**0.1654**	57477.45	**70284.81**
11	236	**115**	468	**549**	0.5043	**0.2095**	67783.41	**73594.82**
12	275	**97**	447	**549**	0.6152	**0.1767**	63082.70	**73594.82**
13	215	**123**	478	**542**	0.4498	**0.2269**	60424.41	**70284.81**
14	339	**56**	456	**579**	0.7434	**0.0967**	**66232.23**	65885.00
15	294	**109**	462	**547**	0.6364	**0.1993**	65921.33	**66057.19**
16	275	**92**	464	**576**	0.5927	**0.1597**	62888.72	**70284.81**
17	308	**94**	496	**566**	0.6210	**0.1661**	61514.92	**73594.82**
18	273	**77**	470	**551**	0.5809	**0.1397**	66370.62	**72033.78**
19	285	**85**	434	**579**	0.6567	**0.1468**	63651.26	**70284.81**
20	269	**113**	512	**516**	0.5254	**0.2190**	69896.12	**72033.78**
average	264	**97.7**	466	**553.8**	0.5669	**0.1769**	63509.22	**71025.68**

Table 3. Comparison of PE_1 and PE_2 for portfolio m100n3 (better values are in bold face)

In the portfolio m100n3 test case, the HMOEA clearly outperforms the non-hybrid approach in every average of the performance criteria. Beyond that, even in each single run the hybrid approach found more solutions than the non-hybrid algorithm (compare the fifth to the fourth column), and the quality of the found solutions is also better concerning the number of dominated solutions (see third versus second column). Therefore, $C_1 > C_2$ in each test run. Obviously, this leads to a significant difference between the average set coverage metric values where the HMOEA is more than three times better than the non-hybrid approach. Concerning the maximum spread, the HMOEA is better in the average over all test cases, and there is only one case where the hybrid approach is slightly worse than the non-hybrid approach whereas in all other cases, the hybrid approach produces an approximation set PE_2 that has a larger maximum spread value than the other approximation set PE_1.

As a consequence, the results of all tests, and particularly the medium and large test cases, support our claim that the hybridisation of the MOEA improves the convergence properties of the algorithm. Especially when dealing with very large search spaces, the exploitation of local information around a solution is valuable in the evolutionary process since it drives the evolutionary process faster towards the most promising solutions. On the other hand, the other variation operators are also very important when using such local information since a strong local search process can stall into a small number of local optima which are only a few points compared to a large feasible, Pareto-optimal set. So a hybrid approach is preferable.

In addition to the results presented above, we also tested the performance of the HMOEA and the non-hybrid MOEA without a capital budget restriction for the respective portfolios. This means, we considered the unconstrained cases, too. We do not discuss them in detail here since the comparison of the HMOEA and the non-hybrid algorithm revealed similar results for all portfolio sizes: The average set coverage metric values of the hybrid approach were always better and even the average maximum spread values of the HMOEA were always equal or higher than these performance metric values of the non-hybrid approach. In all unconstrained cases, the HMOEA benefits strongly from its local search variation operator that enforces a higher quality of the discovered solutions and beyond that, leads to the discovery of the extreme solutions at the margins of the objective function spaces, which are not restricted in the unconstrained cases. Thus, the hybrid approach is also favourable in this problem setting.

Conclusion and Outlook

In this article we have formally defined a constrained multi-objective portfolio selection problem based on investment alternatives which incorporate credit risk. This problem consists of two conflicting objective functions, the aggregated net return from a portfolio and the aggregated downside risk, and an additional capital budget restriction. We have analysed the structure of the problem from a computational perspective and proved the NP-hardness of its associated decision problem.

For the approximation of a large set of feasible, global non-dominated solutions from the feasible, global Pareto-efficient set of solutions to our portfolio problem, we have proposed a hybrid multi-objective evolutionary algorithm framework that combines concepts from different multi-objective evolutionary algorithm schemes with a problem specific local search operator. The framework is not restricted to linear or convex objective functions and also flexible concerning the constraints. A proper genetic modelling of portfolio credit risk problems has been derived in general, and a fast greedy algorithm as a preprocessing stage to support evolutionary algorithms for portfolio credit risk problems has been developed. Further aspects of the algorithm have been considered, particularly with respect to computational complexity.

We have described the CreditRisk+ portfolio credit risk model and derived a local search operator that exploits specific model features. This basis has been used for an implementation of our hybrid algorithm framework, and we have presented empirical results of a test using different portfolios. The results have indicated that our genetic modelling proposed for portfolio credit risk problems is successful since even a non-hybrid MOEA that used our preprocessing algorithm yielded good results for different problem instances. Moreover, the empirical results of different test portfolios showed that the quality of the MOEA could be improved significantly concerning the convergence speed towards the feasible, global Pareto-efficient set by applying the additional local search variation operator that has been developed in this article. Particularly for the medium and larger cases that we have considered, the hybridisation of the MOEA and the local search algorithm has yielded a better quality of the solutions found at a defined population step as well as a higher spread of the solutions in the objective function space both on average and in the majority of the single, independent algorithm runs. The additional computational cost of the local search variation operator are low compared to the advantages, and the user can decide about the amount of additional computational cost to be invested in favour of a higher convergence speed by setting a single parameter, the probability p_{local} for the application of the local search variation operator to each individual. To support this decision, we have carried out a sample analysis of the influence of this parameter on the performance of the algorithm for one of our test data sets.

Although our implementations of the non-hybrid MOEA and the HMOEA have been running on a single standard desktop PC, the algorithms have found approximations of many feasible, Pareto-optimal solutions in different problem instances within a few minutes. Remembering the fact that EAs are well suited for parallel implementation (see e. g. Schmeck et al. (2001)) there are good perspectives for improving the speed of future implementations of our framework by using more than one CPU at least for some parts of the algorithms in each population step.

Further research from the viewpoint of risk modelling can e. g. extend the framework presented here by exploiting the latest developments in the CreditRisk+ context published in Buergisser et al. (2001) to include severity variations concerning the net exposures or use an alternative way of calculating the loss percentiles as proposed by Gordy (2001). Of course, the system can be extended to other credit risk exposure types, e. g. by embedding it into a mark-to-market model context. Due to the flexibility of our framework, many further constraints of practical interest can be easily integrated into our framework, e. g. the simultaneous use of different capital budgets or Credit-Value-at-Risk based limits per rating category and/or industry in the optimisation process. Even more sophisticated restrictions can be handled, e. g. a minimum overall quality of the parts of a portfolio to be sold in an Assed Backed Security transaction which is itself calculated using a non-linear pricing model.

Finally, the system presented in this paper can be integrated into a larger decision support system for risk-return optimisation in a financial institution that supports human portfolio risk-return managers and traders using software agent technology as proposed in Schlottmann & Seese (2001b).

Acknowledgement

The authors like to thank Tobias Dietrich, Michael Lesko, Thomas Stümpert and Stephan Vorgrimler for interesting discussions. Furthermore, support from GILLARDON AG financial software, Germany is hereby gratefully acknowledged.

Appendix

Proof of Lemma 1.

The decision problem that corresponds to Problem 1 has the following form:

Problem 2. Are there numbers $x_i \in \{0, e_i\}$, $x = (x_i)_{i=1,...,m}$ for given nonnegative rational numbers e_i, r_i, p_i, w_i, K, R, Z and a rational function h such that the following three inequalities are satisfied:

$$ret(x,p,r) \geq R \quad (4.1)$$
$$risk(x,p;h) \leq Z \quad (4.2)$$
$$cap(x,w) \leq K \quad (4.3)$$

Now consider an instance of the following decision problem which is known to be NP-complete:

Problem [*0/1 KNAPSACK*]. Given are a finite set U, a rational size $s(u) > 0$, a rational value $v(u) > 0$ for each element $u \in U$ and positive rational numbers V, W. Is there an assignment of an integer value $c(u) \in \{0,1\}$ to each $u \in U$ such that the following two conditions are satisfied:

$$\sum_{u \in U} c(u)v(u) \geq V \quad (4.4)$$
$$\sum_{u \in U} c(u)s(u) \leq W \quad (4.5)$$

We can construct an equivalent instance of Problem 2 for a given [0/1 KNAPSACK] problem instance by using a polynomial time calculable 1-1 function f: $U \rightarrow N$ that assigns a subsequent natural number to each element $u \in U$ starting from $f(u) := 1$ for the first element in U and by setting $m := |U|$, $R := V$, $K := W$ and $\forall i \in \{1,...,m\}$: $e_i := 1$, $r_i := v(f^{1}(i))$, $w_i := s(f^{1}(i))$.

Furthermore, we can set $\forall i \in \{1,...,m\}$: $p_i \equiv 0$ for the given instance of [0/1 KNAPSACK] in our instance of Problem 2 so that the inequality (4.2) is not binding for any given positive rational number Z since $risk(x, p; h) = 0$ according to Definition 5 in this case. Thus, we have to consider only inequalities (4.1) and (4.3) in our construction of the equivalent Problem 2 instance.

For $p_i \equiv 0$ the inequality (4.1) simplifies to

$$ret(x,p,r) = \sum_{i=1}^{m} x_i(r_i - p_i) \overset{p_i \equiv 0}{=} \sum_{i=1}^{m} x_i r_i \geq R \,. \quad (4.6)$$

So a solution to our constructed instance of Problem 2 has to satisfy the following conditions:

$$\sum_{i=1}^{m} x_i r_i \geq R \quad (4.7)$$
$$\sum_{i=1}^{m} x_i w_i \leq K \quad (4.8)$$

By construction of the variables e_i, r_i, w_i, R and K a solution $x = (x_i)_{i=1,...,m}$ is a solution to this instance of Problem 2 if and only if $\forall i \in \{1,..., m\}$:c$(f^{1}(i)) := x_i$ is a solution to the given instance of [0/1 KNAPSACK].

So we have found a polynomial time reduction from [0/1 KNAPSACK] to our Problem 2. Since [0/1 KNAPSACK] is known to be NP-complete, our Problem 2 is NP-hard.

Proof of formula (3.10).

Given is a portfolio specified by the vectors x, p, σ, Θ, r.
The function *f* is defined as follows:

$$f\left(x, p, \sigma, \Theta, r\right) := \frac{\sum_{i=1}^{m} x_i\left(r_i - p_i\right)}{q_{pf}^{0.99}\left(x, p, \sigma, \Theta\right) - \sum_{i=1}^{m} x_i p_i} \quad (4.9)$$

If we calculate a constant multiplier for the given portfolio data

$$\xi_{pf} := \frac{q_{pf}^{0.99}\left(x, p, \sigma, \Theta\right) - \mu_{pf}\left(x, p\right)}{\sigma_{pf}\left(x, p, \sigma, \Theta\right)} \quad (4.10)$$

which can be abbreviated by

$$\xi_{pf} := \frac{q_{pf}^{0.99} - \mu_{pf}}{\sigma_{pf}} \quad (4.11)$$

in analogy to CreditSuisse Financial Products (1997), p. 63, the 99th percentile function can be reformulated by

$$q_{pf}^{0.99} = \mu_{pf} + \xi_{pf}\sigma_{pf} \quad (4.12)$$

By substituting the 99th percentile function in formula (4.9) according to (4.12) we obtain:

$$\frac{\sum_{i=1}^{m} x_i\left(r_i - p_i\right)}{\mu_{pf} + \xi_{pf}\sigma_{pf} - \sum_{i=1}^{m} x_i p_i} \quad (4.13)$$

Taking into account that

$$\mu_{pf} := \sum_{i=1}^{m} x_i p_i \quad (4.14)$$

formula (4.13) can be simplified to

$$\frac{\sum_{i=1}^{m} x_i\left(r_i - p_i\right)}{\xi_{pf}\sigma_{pf}} \quad (4.15)$$

The partial derivative of *f* is calculated by deriving (4.15) using quotient rule:

$$d_j := \frac{\partial}{\partial x_j} f\left(x, p, \sigma, \Theta, r\right)$$

$$= \frac{\left(r_j - p_j\right)\left(\xi_{pf}\sigma_{pf}\right) - \sum_{i=1}^{m} x_i\left(r_i - p_i\right) \quad \frac{\partial}{\partial x_j}\left(\xi_{pf}\sigma_{pf}\right)}{\left(\xi_{pf}\sigma_{pf}\right)^2} \quad (4.16)$$

For $x_j \neq 0$ formula (4.16) is equivalent to

$$\frac{x_j\left(r_j - p_j\right)\left(\xi_{pf}\sigma_{pf}\right) - x_j \sum_{i=1}^{m} x_i\left(r_i - p_i\right) \frac{\partial}{\partial x_j}\left(\xi_{pf}\sigma_{pf}\right)}{x_j\left(\xi_{pf}\sigma_{pf}\right)^2}$$

$$= \frac{x_j\left(r_j - p_j\right)\left(\xi_{pf}\sigma_{pf}\right) - \sum_{i=1}^{m} x_i\left(r_i - p_i\right) x_j \frac{\partial}{\partial x_j}\left(\xi_{pf}\sigma_{pf}\right)}{x_j\left(\xi_{pf}\sigma_{pf}\right)^2} \quad (4.17)$$

Recalling the assumption that ξ_{pf} is considered a constant calculated from the portfolio mean and standard deviation, the previous formula can be transformed into

$$\frac{x_j\left(r_j - p_j\right)\left(\xi_{pf}\sigma_{pf}\right) - \sum_{i=1}^{m} x_i\left(r_i - p_i\right) \xi_{pf} x_j \frac{\partial}{\partial x_j}\sigma_{pf}}{x_j\left(\xi_{pf}\sigma_{pf}\right)^2} \quad (4.18)$$

Finally, remembering that

$$RC_j^{\sigma} := x_j \frac{\partial \sigma_{pf}}{\partial\, x_j}$$

the substitution of the partial derivative yields

$$d_j = \frac{x_j\left(r_j - p_j\right)\left(\xi_{pf}\sigma_{pf}\right) - \sum_{i=1}^{m} x_i\left(r_i - p_i\right) \left(\xi_{pf} RC_j^{\sigma}\right)}{x_j\left(\xi_{pf}\sigma_{pf}\right)^2} \quad (4.19)$$

Specification of portfolio m20n2.

i	e_i	Θ_{i1}	Θ_{i2}	p_i	r_i	w_i
1	12700	89%	11%	2.0%	4.72%	12.60%
2	15000	73%	27%	2.0%	3.33%	10.00%
3	3500	71%	29%	4.0%	2.86%	8.57%
4	19800	54%	46%	3.0%	5.05%	12.63%
5	30100	29%	71%	2.0%	8.31%	12.96%
6	30600	75%	25%	6.0%	7.52%	12.09%
7	43000	37%	63%	3.0%	4.19%	9.30%
8	22800	68%	32%	6.0%	7.02%	14.04%
9	23500	53%	47%	5.0%	5.11%	8.51%
10	9200	39%	61%	4.0%	14.13%	15.22%
11	40800	32%	68%	4.0%	6.13%	9.07%
12	26200	58%	42%	7.0%	4.20%	10.69%
13	42100	24%	76%	4.0%	5.46%	8.79%
14	27200	39%	61%	5.0%	7.72%	11.40%
15	1900	44%	56%	6.0%	5.26%	10.53%
16	34700	27%	73%	5.0%	4.03%	8.65%
17	40900	22%	78%	5.0%	8.80%	9.29%
18	28000	14%	86%	5.0%	6.43%	8.93%
19	32200	8%	92%	5.0%	2.80%	8.70%
20	4800	7%	93%	5.0%	4.17%	8.33%

Note that the variables p_i, r_i and w_i are calculated on a basis of e_i. The variation coefficient for the second sector was set to $\omega_2 := 0.75$ in analogy to real-world variation coefficients of default rates.

References

[1] Andersson, F. & Uryasev, S. 1999, Credit risk optimization with conditional value-at-risk criterion, working paper, University of Florida, Gainesville.

[2] Artzner, P., Delbaen, F., Eber. J. & Heath, D. 1999, Coherent measures of risk, Mathematical Finance 9 (1999) 3, pp. 203-228.

[3] Basel Committee for Banking Supervision 2001, The New Basel Capital Accord, available via www.bis.org/publ/bcbsca03.pdf.

[4] Buergisser, P., Kurth, A. & Wagner, A. 2001, Incorporating severity variations into credit risk, Journal of Risk 3 (2001) 4, pp. 5-31.

[5] Chang, T., Meade, N., Beasley, J. & Sharaiha, Y. 2000, Heuristics for cardinality constrained portfolio optimisation, Computers & Operations Research 27 (2000), pp. 1271-1302.

[6] CreditSuisse Financial Products (1997), CreditRisk+TM - a credit risk management framework, available via http://www.csfp.co.uk/creditrisk/assets/creditrisk.pdf.

[7] Deb, K., Agrawal, S., Pratap, A. & Meyarivan, T. 2000, A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, in: Schoenauer, M., Deb, K., Rudolph, G., Yao. X., Lutton, E., Merelo, J., Schwefel, H. (eds.), Parallel problem solving from nature, LNCS 1917, Springer, Berlin, pp. 849-858.

[8] Deb, K. 2001, Multi-objective optimisation using evolutionary algorithms, John Wiley & Sons, Chichester.

[9] Duffie, D. & Singleton, K 1999, Modelling term structures of defaultable bonds, Review of Financial Studies 12 (1999) 4, pp. 687-720.

[10] Ehrgott, M. 2000, Multicriteria optimisation, Lecture notes in economics and mathematical systems 491, Springer, Heidelberg.

[11] Elton, E. & Gruber, M. 1995, Modern portfolio theory and investment analysis, John Wiley & Sons, Chichester.

[12] Fogel, D. & Michaelwicz, Z. 2000, How to solve it: Modern heuristics, Springer, Heidelberg.

[13] Fonseca, C. & Fleming, P. 1993, Genetic Algorithms for multiobjective optimization: formulation, discussion and generalization, in: Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 416-423.

[14] Frey, R. & McNeil, A. 2001, Modelling dependent defaults, RiskLab working paper, ETH Zuerich, available via www.math.ethz.ch/~frey/credit-paper.pdf.

[15] Garey, M. & Johnson, D. 1979, Computers and Intractability, W. H. Freeman and Company, New York.

[16] Gilli, M. & Kellezi, E. 2000, Portfolio selection, tail risk and heuristic optimisation, working paper, University of Geneva.

[17] Goldberg, D. 1989, Genetic Algorithms for search, optimisation and machine learning, Addison-Wesley, Reading.

[18] Gordy, M. 2001, Calculation of higher moments in CreditRisk+ with applications, Board of Governors of the Federal Reserve System, Washington.

[19] Holland, J. 1975, Adaptation in natural and artificial systems, Michigan University Press, Ann Arbor.

[20] Horn, J., Nafpliotis, N. & Goldberg, D. 1994, A niched pareto genetic algorithm for multiobjective optimization, in: Forrest, S. (ed.), Proceedings of the 5th International Conference on Genetic Algorithms, Morgan Kauffman, San Mateo, pp. 416-423.

[21] Knowles, J. & Corne, D. 1999, The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation, in: Proceedings of the 1999 Congress on Evolutionary Computation, Vol. 1, IEEE Press, Piscataway, pp. 98-105.

[22] Knowles, J. & Corne, D. 2000, M-PAES: A memetic algorithm for multiobjective optimization, in: Zalzala, A. et al. (eds.), Proceedings of the 2000 Congress on Evolutionary Computation, IEEE Press, Piscataway, pp. 325-332.

[23] Lando, D. 1998, On Cox processes and credit risky securities, Review of Derivatives Research (1999) 2, pp. 99-120.

[24] Laumanns, M., Thiele, L., Deb, K. & Zitzler, E. 2001, On the convergence and diversity-preservation properties of multi-objective evolutionary algorithms, TIK Report 108, Eidgenoessisch-Technische Hochschule (ETH), Zuerich.

[25] Lehrbass, F. 1999, Rethinking risk-adjusted returns, in: Credit Risk Special Report, Risk (1999) 4, pp. 35-40.

[26] Markowitz, H. 1952, Portfolio selection, Journal of Finance 7 (1952), pp. 77ff.

[27] Osyczka, A. & Kundu, S. 1995, A new method to solve generalized multicriteria optimization problems using the simple Genetic Algorithm, Structural optimization 10 (1995) 2, pp. 94-99.

[28] Papamiditriou, C. 1994, Computational Complexity, Addison-Wesley, Reading.

[29] Panjer, H. 1981, Recursive evaluation of a family of compound distributions, ASTIN Bulletin 12 (1981), pp. 22-26.

[30] Panjer, H. & Willmot, G. 1992, Insurance risk models, Society of Actuaries, Schaumburg.

[31] Pflug, G. 2000, Some remarks on the Value-at-Risk and the Conditional Value-at-Risk, in: Uryasev, S. (ed.), Probabilistic constrained optimization, Kluwer, Dordrecht, pp. 272-281.

[32] Rudolph, G. 1998, Finite Markov chain results in evolutionary computation: A tour d'horizon, Fundamenta Informaticae 35 (1998) 1-4, pp. 67-89.

[33] Rudolph, G. & Agapie, A. 2000, Convergence properties of some multi-objective evolutionary algorithms, in: Zalzala, A. et al. (eds.), Proceedings of the 2000 Congress on Evolutionary Computation, Vol. 2, IEEE Press, Piscataway, pp. 1010-1016.

[34] Rudolph, G. 2001, Evolutionary search under partially ordered finite sets, in: Sebaaly, M. (ed.), Proceedings of the International NAISO Congress on Information Science Innovations 2001, ICSC Academic Press, Millet, pp. 818-822.

[35] Schaffer, J. 1984, Some experiments in machine learning using vector evaluated genetic algorithms, PhD Thesis, Vanderbilt University, Nashville.

[36] Schlottmann, F. & Seese, D. 2001, A hybrid genetic-quantitative method for risk-return optimisation of credit portfolios, Quantitative Methods in Finance 2001 conference abstracts, University of Technology, Sydney, Australia, p. 55, full paper available via ww.business.uts.edu.au/finance/resources/qmf2001/Schlottmann_F.pdf.

[37] Schlottmann, F., Seese. D. 2001, An agent-based architecture for the management and trading of portfolio credit risk, working paper, University of Karlsruhe, 2001 (in German).

[38] Schmeck, H., Branke, J., Kohlmorgen, U. 2001, Parallel implementations of evolutionary algorithms, in: Zomaya, A., Ercal, F., Olariu, S. (eds.): Solutions to parallel and distributed computing problems, John Wiley & Sons, New York, pp. 47-68.

[39] Seese, D. & Schlottmann, F. 2001, Structural reasons for high complexity: A survey on results and problems, working paper, University of Karlsruhe.

[40] Seese, D. & Schlottmann, F. 2002a, Large grids and local information flow as a reason for high complexity, in: Frizelle, G. & Richards, H., Tackling industrial complexity: the ideas that make a difference, Proceedings of the 2002 conference of the Manufacturing Complexity Network, University of Cambridge, pp. 193-207.

[41] Seese, D. & Schlottmann, F. 2002b, The building blocks of complexity: a unified criterion and selected problems in economics and finance, presented at the Sydney Financial Mathematics Workshop 2002, available via http://www.qgroup.org.au/SFMW/

[42] Sharpe, W. 1964, Capital Asset Prices. A theory of market equilibrium under conditions of risk, Journal of Finance 19 (1964), pp. 425-442.

[43] Van Veldhuizen, D. 1999, Multiobjective Evolutionary Algorithms: classifications, analyses and new innovations, PhD thesis, Air University, Dayton.

[44] Wilson, T. 1997a, Portfolio Credit Risk (I), in: Risk (1997) 9, pp. 111-119.

[45] Wilson, T. 1997b, Portfolio Credit Risk (II), in: Risk (1997) 10, pp. 56-61.

[46] Zitzler, E. & Thiele, L. 1998, An Evolutionary Algorithm for multiobjective optimization: the strength pareto approach, TIK-Report 43, Eidgenoessisch-Technische Hochschule (ETH), Zuerich.

[47] Zitzler, E. 1999, Evolutionary Algorithms for multiobjective optimization: methods and applications, PhD thesis, Eidgenoessisch-Technische Hochschule (ETH), Zuerich.

[48] Zitzler, E., Deb, K. & Thiele, 2000, Comparison of multiobjective evolutionary algorithms: empirical results, Evolutionary Computation 8 (2000) 2, pp. 173-195.

[49] Zitzler, E., Laumanns, M. & Thiele, L. 2001, SPEA2: Improving the Strength Pareto Evolutionary Algorithm, TIK-Report 103, Eidgenoessisch-Technische Hochschule (ETH), Zuerich.