GILLARDON

Auszug Publikationen 2004

- Risk-Return Analysis of Credit Portfolios

Wir denken nach, um vorzudenken

Risk-Return Analysis of Credit Portfolios

Dr. Frank Schlottmann / Prof. Dr. Detlef Seese / Dr. Michael Lesko / Stephan Vorgrimler

Gundlach, M. / Lehrbass, F. (Eds.): CreditRisk+™ in the Banking Industry

Summary

We consider a problem of real-world risk-return analysis of credit portfolios in a multi-objective function setting with respect to additional constraints. For the approximation of a set of feasible, risk-return-efficient portfolio structures in this setting we discuss a flexible approach that incorporates multi-objective evolutionary and local search methods as well as specific features of the CreditRisk+ model. We apply the hybrid approach to a sample loan portfolio to illustrate its working principle.

Introduction

The intensive development of quantitative portfolio credit risk models like CreditRisk+ and the increasing trade in financial instruments for transferring credit risk like credit default swaps, collateralized loan obligations etc. are major reasons for a growing importance of credit portfolio risk-return analysis and optimization. Beyond that, there will possibly be more demand for credit portfolio optimization, as soon the supervisory capital requirements for banks will be changed due to proposals of the Basel Committee, e.g. by setting new capital weights on some credit risk exposure types and providing supervisory capital relief for risk mitigation (cf. e.g. Basel Committee for Banking Supervision [1]).

In the following sections, we will focus on the approximation of risk-return efficient sets[1] 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 or private households not having direct access to the capital market. Such exposures are well-suited for risk measurement within the CreditRisk+ model framework.

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, e.g. Lehrbass [6] has pointed out that 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 is it 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.

For our considerations, the concept of Pareto-optimality is essential, i.e. efficient structures are Pareto-optimal concerning the two distinct (and usually 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 a constrained portfolio credit risk-return optimization problem that are comparable to the efficient frontier in stock portfolio investment analysis. However, in our case we deal with 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.

The remainder of this chapter is organized as follows: In the first section, we specify our portfolio credit risk optimization problem. Afterwards, we compare this problem to a related problem considered by Lehrbass [6]. Then we give an overview of our hybrid evolutionary algorithm framework for the portfolio credit risk optimization problem. The next section shows how the CreditRisk+ model is integrated into our framework. In a sample application we illustrate the working principle of the hybrid algorithm.

1. Notation and Problem Definition

In this section, we will present the basic terminology of the constrained discrete credit portfolio risk-return analysis problem to be solved.

Definition 1. *Given is a credit portfolio containing $K > 1$ obligors. Each investment alternative (obligor) $A \in \{1, \ldots, K\}$ incorporates the risk of default and is characterized by the following data which is considered to be constant within the time period $[0,t_1]$ where t_1 is the chosen risk horizon (usually $t_1 = 1$ in practical applications):*

- *net exposure $v_A \geq 0$ (loss in monetary units if obligor A defaults)*
- *expected default probability $p_A \geq 0$*
- *standard deviation of default probability[2] $\sigma_A \geq 0$*

- *sector weights* w_{Ak}, $k \in \{0, \dots, N\} \geq 0$

- *expected rate of return* η_A *in relation to* v_A
  (net of cost, particularly after cost of funding but before credit risk)

- *supervisory capital requirement percentage* $c_A \geq 0$ *in relation to* v_A.

The above variables are mainly the basic inputs for the CreditRisk+ model as described in Chapter 2. Moreover, the expected return and the capital requirement for each obligor have been added. Both are expressed in relation to v_A, which usually requires a simple rescaling step of real-world parameters like supervisory capital weights based on gross exposure but this avoids unnecessary additional variables in our problem specification. In the following text, we will abbreviate the respective set of scalar variables v_A, p_A, σ_A, η_A, c_A of all obligors by vectors $v := (v_1, \dots, v_K)^T$ and analogously η, p, σ and c. The sector weights are abbreviated by a matrix $w := (w_{Ak})_{A \in \{1, \dots, K\}, k \in \{0, \dots, N\}}$.

We assume that the investor has to decide about holding a subset of the obligors in her portfolio, consider e.g. a bank that wants to optimize its loan portfolio containing K different obligors. We fix the following notations according to the assumptions that a supervisory capital budget for the investments is fixed and given by the bank's maximum supervisory capital.

- A supervisory capital budget of the investor is given by $B > 0$.

- A portfolio structure is given by a vector

$$x = (x_1, x_2, \dots, x_K)^T, x_A \in \{0, \nu_A\}.$$

Since every x_A can only take the values 0 or ν_A, the investor has to decide whether to hold the whole net exposure ν_A in her portfolio. In many real-world portfolio optimization problems the decision is e.g. either keeping the obligor A in the credit portfolio or selling the entire net exposure of obligor A to a risk buyer. Or, alternatively stated, whether adding a new exposure to the credit portfolio. 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 in obligor A, the decision variables will often 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 tradeoff between both objectives, since any rational investor will ask for a premium (additional return) to take risk.

Definition 2. *The aggregated expected return from a portfolio structure* x *is calculated by*

$$ret(x, p, \eta) := \sum_A \eta_A x_A - \sum_A p_A x_A = \sum_A (\eta_A - p_A) x_A .$$

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

Definition 3. *The aggregated downside risk from the portfolio structure* x *for the investor is calculated by*

$$risk(x, p, \sigma, \omega) := UL_\varepsilon(x, p, \sigma, \omega)$$

where $UL_\varepsilon(x, p, \sigma, \omega)$ *denotes the resulting Credit-Value-at-Risk to the chosen confidence level* ε *for the given portfolio structure represented by the vector* x.

The above choice of the risk measure is a common choice for measuring credit risk or economic capital in banks, cf. the remarks by Gundlach in Chapter 2. For our considerations in this section, we do not need to choose the calculation procedure for the cumulative distribution function of aggregated losses or the approximation procedure for the risk measure. A summary of Panjer's recursion algorithm can be found in the appendix to Chapter 2 and more recent approaches are discussed in Part II of this volume.

Definition 4. *The required supervisory capital of a given portfolio structure* x *is*

$$cap(x, c) := \sum_A x_A c_A$$

Definition 5. *A portfolio structure* x *is feasible if and only if*

$$cap(x, c) \leq B.$$

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

Definition 6. *Given two distinct feasible portfolio structures* x *and* y, x *dominates* y *if and only if one of the following cases is true:*

$$ret(x, p, \eta) > ret(y, p, \eta) \wedge risk(x, p, \sigma, \omega) \leq risk(y, p, \sigma, \omega)$$

$$ret(x, p, \eta) \geq ret(y, p, \eta) \wedge risk(x, p, \sigma, \omega) < risk(y, p, \sigma, \omega)$$

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 than y in the other criterion. It is obvious that a rational investor will prefer x over y if $x >_d y$.

Definition 7. *Let 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 *be given. A solution* $x \in S'$ *is a feasible Pareto-optimal portfolio structure if and only if it satisfies the following condition:*

$$\forall y \in S' : \neg (y >_d x).$$

To choose between the best combinations of obligors using her preferences or utility function, a rational investor is interested in finding the set of feasible Pareto-optimal portfolio structures that has maximum cardinality. This set is comparable to the efficient frontier of Markowitz [7], but in a constrained, discrete decision space.

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

$$PE^* := \{x \in S' : \forall y \in S' : \neg (y >_d x)\}.$$

Problem 1 is a computationally hard problem in the sense of **NP**-hardness – see Seese and Schlottmann [12] for a formal analysis of such and further portfolio credit risk problems.[3] This means that unless the computational complexity classes **P** and **NP** satisfy the condition **P** = **NP**, which is still an open problem besides many attempts to prove this relationship, there is no deterministic algorithm that calculates PE^* within polynomial computing time (measured by the number of the obligors K). For practical implementations, these results on computational complexity imply that the constrained search space for feasible risk-return efficient portfolio structures grows exponentially with the numbers of obligors, i.e. the number of potential solutions is 2^K. Given limited computational resources (e.g. a fixed number of executable operations per second) and a bounded runtime, Problem 1 cannot be solved exactly by simple enumeration of all possible solutions even when dealing with rather small problem dimensions.

2. The Approach by Lehrbass using Real-Valued Decision Variables

In his article on constrained credit portfolio risk-return optimization, Lehrbass [6] maximized a single target function using real-valued variables, i.e. he considered a relaxation of our Problem 1 in a single objective function setting. Using our notation, the following optimization problem is an example for a two-obligor problem discussed by Lehrbass:

Problem 2. Given is an arbitrary, but fixed bound $B > 0$ (capital budget) and an arbitrary, fixed bound $V > 0$ on the downside risk (in [6] : risk-adjusted capital), solve the following optimization problem:

$$\max_{x_{A1}, x_{A2}} \{\eta_{A1}\, x_{A1} + \eta_{A2}\, x_{A2}\}, \tag{1}$$

$$c_{A1}\, x_{A1} + c_{A2}\, x_{A2} \le B, \tag{2}$$

$$risk(x, p, \sigma, w) \le V, \tag{3}$$

$$x_{A1} \le \nu_{A1},\ x_{A2} \le \nu_{A2}, \tag{4}$$

$$x_{A1}, x_{A2} \in \mathbb{R}^0_+. \tag{5}$$

Note that in contrast to the formulation in [6] the above problem has upper bounds on the decision variables to keep consistence with the decision alternatives considered so far, but this does not change the main idea. In his study, Lehrbass applied standard methods for solving convex optimization problems to Problem 2, see [6] for more details.

Assuming that there is a solution satisfying (2), (3), (4) and (5), one obtains a single optimal solution x^*_{A1}, x^*_{A2} by solving an instance of Problem 2 once for the given parameters. This optimal solution has the maximum return for the given upper bound V on the risk and for the given upper bound B on the required capital.

By solving Problem 2 several times using different upper bounds V in the *risk* constraint (3) one obtains a set of Pareto-optimal solutions that represent the tradeoff between portfolio risk and return. Since Problem 2 has real-valued decision variables, there are usually infinitely many solutions, hence this set of Pareto-optimal solutions is an approximation for the tradeoff between portfolio risk and return.

In contrast to the above formulation of Problem 2, when returning to our Problem 1 we have to deal with a discrete optimization problem consisting of a fixed number of distinct choices. This is due to real-world restrictions like integral constraints and market frictions, e.g. transaction cost or round lots. Moreover, portfolio optimization problems based on downside risk measures incorporate non-linear objective functions. If the Credit-Value-at-Risk is used in the *risk* objective function as in Problem 1, this objective function becomes even non-convex.[4] Thus, the necessary conditions for the application of standard algorithms for convex optimization problems are not satisfied for Problem 1. Therefore, we apply an algorithm to this problem that searches for discrete solutions and does not assume linearity or convexity of the objective functions or constraints.

3. A Multi-Objective Evolutionary Approach Combined with Local Search

3.1. Description of the algorithmic framework

Since the first reported implementation and test of a multi-objective evolutionary approach, the vector-evaluated genetic algorithm (VEGA) by Schaffer [9] in 1984, this special branch of evolutionary computation has attracted many researchers dealing with non-linear and non-convex multi-objective optimization problems. After the introduction of VEGA, many different evolutionary algorithms (EAs) have been proposed for multi-objective optimization problems, see e.g. Deb [3] for an overview and a detailed introduction.

In general, a multi-objective evolutionary algorithm (MOEA) is a randomized heuristic search algorithm reflecting the Darwinian 'survival of the fittest' principle that can be observed in many natural evolution processes, cf. e.g. Holland [5]. 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 our portfolio credit risk optimization problem (Problem 1), we assume that the decision variables x_A will be arranged in a vector to obtain gene strings representing potential solutions. The resulting genotypes consist of real-valued genes that are connected to strings and take either value 0 or v_A depending on the absence or presence of obligor A in the current solution. So we obtain strings of length K that represent some of the 2^K combinations of possible (but neither necessarily feasible nor necessarily optimal) portfolio structures. Examples showing such gene strings are given later in Figure 1 and Figure 2.

The following hybrid multi-objective evolutionary algorithm (HMOEA) combining MOEA concepts and local search methods computes an approximation of PE^* mainly by modifying individuals using so-called variation operators, which change the contents of genes, by evaluating individuals based on the given objective functions and the constraints, and by preferring individuals that have a better evaluation than other individuals in $P(t)$. More details on these subjects and the other elements of the HMOEA are given below.

Algorithm 1. *HMOEA basic algorithm scheme*

1: $t := 0$
2: *General initial population* $P(t)$
3: *Initialize elite population* $Q(t) := \varnothing$
4: *Evaluate* $P(t)$
5: **Repeat**
6: *Select individuals from* $P(t)$
7: *Recombine selected individuals (variation operator 1)*
8: *Mutate recombined individuals (variation operator 2)*
9: *Apply local search to mutate individuals (variation operator 3)*
10: *Create offspring population* $P'(t)$
11: *Evaluate joint population* $J(t) := P(t) \cup P'(t)$
12: *Update elite population* $Q(t)$ *from* $J(t)$
13: *Generate* $P(t+1)$ *from* $J(t)$
14: $t := t + 1$
15: **Until** $Q(t) = Q(\max\{0, t - t_{diff}\}) \vee t > t_{max}$
16: **Output:** $Q(t)$

At the start of the algorithm, the initial population $P(0)$ will be generated by random initialization of every individual to obtain a diverse population in the search space of potential solutions.

We use an elite population $Q(t)$ in our algorithm that keeps the best feasible solutions found so far at each time step t. Rudolph & Agapie [8] have shown under weak conditions that in a MOEA having such an elite population, the members of $Q(t)$ converge to elements of PE^* with probability 1 for $t \to \infty$ which is a desirable convergence property. Furthermore, the algorithm can be terminated at any time by the user without losing the best feasible solutions found so far. This is particularly important for real-world applications. At the start of the algorithm, $Q(t)$ is empty. When the algorithm terminates, $Q(t)$ contains the approximation for PE^*.

The evaluation of $P(t)$ in line 4 and $J(t)$ in line 11 is based on the non-domination concept proposed by Goldberg [4] and explicitly formulated for constrained problems, e.g. in Deb [3, p. 288]. In our context, it leads to the following type of domination check, which extends Definition 6.

Definition 8. *Given two distinct portfolio structures x and y, x constraint-dominates y if and only if one of the following cases is true:*

$$cap(x, c) \le B \wedge cap(y, c) \le B \wedge ret(x, p, \eta) > ret(y, p\, \eta) \wedge risk(x, p, \sigma, w) \le risk(y, p, \sigma, w) \quad (6)$$

$$cap(x, c) \le B \wedge cap(y, c) \le B \wedge ret(x, p, \eta) \ge ret(y, p, \eta) \wedge risk(x, p, \sigma\, w) < risk(y, p, \sigma\, w) \quad (7)$$

$$cap(x, c) \le B \wedge cap(y, c) > B \quad (8)$$

$$cap(x, c) > B \wedge cap(y, c) > B \wedge cap(x, c) < cap(y, c) \quad (9)$$

If x constraint-dominates y, we will denote this relationship by $x >_{cd} Y$.

The first two cases in Definition 8 refer to the cases from Definition 6 where only feasible solutions were considered. Case (8) expresses a preference for feasible over infeasible solutions and case (9) prefers the solution that has lower constraint violation.

The non-dominated sorting procedure in our HMOEA uses the dominance criterion from Definition 8 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 domination level 0 (best rank). After that, only the remaining solutions are checked for constraint-domination, and the non-constraint-dominated solutions among these obtain domination level 1 (second best rank). This process is repeated until each solution has obtained an associated domination level.

In line 6 of Algorithm 1, the selection operation is performed using a binary tournament based on the domination level. 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. The domination levels of these individuals are compared, which implicitly yields a comparison between the individuals' dominance relation according to Definition 8. If, without loss of generality, $x >_{cd} y$ then x wins the tournament and is considered

Figure 1: One-point crossover variation operator

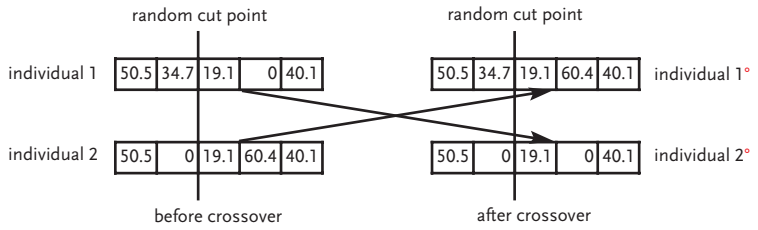
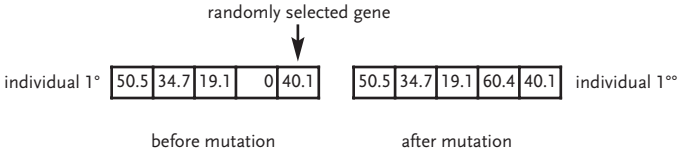


Figure 2: Mutation variation operator



for reproduction. If they cannot be compared using the constraint-domination criterion (i.e. they have the same domination level) the winning individual is finally determined using a draw from a uniform distribution over both possibilities.

The first variation operator is the standard one-point crossover for discrete decision variables, 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 this variation operator is to move the population through the space of possible solutions. In the example displayed in Figure 1, two individuals are shown on the left, each representing a possible portfolio structure for the given investment alternatives. Each individual has five genes, which code the decision variables. The randomly chosen cut position is between the second and the third gene, such that the contents of the third, fourth and fifth gene are exchanged between the two individuals to obtain two offspring.

In analogy to natural mutation, the second variation operator changes the genes of the obtained offspring 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 HMOEA stalling in local optima as there is always a positive probability leaving a local optimum if the mutation rate is greater than zero. Figure 2 gives an example for this variation operator where the fifth gene of the offspring individual 1' from Figure 1 is mutated.

Our third variation operator in line 9 of Algorithm 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 optimization of x towards elements of PE^*, e.g. by using an 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 with a standard, non-hybrid MOEA, since the randomized search process of the MOEA can be guided a bit more towards the elements of PE^* 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, portfolio credit risk models like CreditRisk+ provide additional local structure information for a current solution x beyond the objective function values that can be exploited very efficiently from a computational complexity's perspective. This is described in more detail in the next subsection below.

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. After that, the elite population $Q(t)$ is updated by comparing the best feasible solutions in $J(t)$ (i.e. having domination level 0) to the solutions in $Q(t)$: if a solution $x \in J(t)$ constraint-dominates a solution $y \in Q(t)$, the solution y is removed from $Q(t)$. Furthermore, if $x \in J(t)$ is not constraint-dominated by any solution from $Q(t)$, x is added to $Q(t)$. After this update procedure, $Q(t)$ contains the best feasible solutions found so far.

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 have better convergence properties, we also use this mechanism in our algorithm. Hence, the parents of the new parent population $P(t+1)$ are selected from $J(t)$ according to their domination level. This means the solutions from $J(t)$ having domination level 0 are copied to $P(t+1)$, the remaining capacity of $P(t+1)$ is filled with solutions from $J(t)$ having domination level 1 and so on, until not all solutions from $J(t)$ having a certain domination level k can be copied to $P(t+1)$ due to the bounded capacity of $P(t+1)$. Of course, if the number of solutions from $J(t)$ which have domination level 0 already exceeds the capacity of $P(t+1)$ then $k = 0$ in the above example.

Let $J'(t) \subseteq J(t)$ denote the subset of solutions having domination level k which cannot be copied entirely to $P(t+1)$. To solve the selection problem and to obtain a good distribution of the solutions in the two-dimensional objective function space, an additional criterion is used to choose the solutions from $J'(t)$ to be copied to $P(t+1)$: we incorporate the concept of crowding-sort described e.g. in Deb [3], which selects the solutions from $J'(t)$ by considering the distance between the objective function values of these solutions. Here, the perimeter of the largest possible rectangle that can be drawn around a single solution $x \in J'(t)$ in the objective function space without adding a second solution $y \in J'(t)$, $y \neq x$, to the rectangle's interior serves as a distance measure. The solution $x \in J'(t)$ that has the largest value of this distance measure is copied to $P(t+1)$, and afterwards x is removed from $J'(t)$. This process is repeated until $P(t+1)$ is filled up. As a consequence, at this step the algorithm prefers solutions that belong to less crowded regions of the objective function space.

Finally, the algorithm terminates 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.

3.2. Incorporating CreditRisk+ into the algorithmic framework

In the remaining sections of this chapter, we assume the bank has implemented the standard CreditRisk+ model. For the corresponding calculations no explicit default correlations are required, since 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 $p(A_1, A_2)$ for two obligors A_1, A_2. However, in [2, 56 ff.], the following implicit default correlation formula is provided (below we use the assumption that all random scalar factors S_k are normalized to an expectation $\mu_k = 1$):

$$p(A_1, A_2) \approx \sqrt{p_{A1}\, p_{A2}} \sum_{k=1}^{N} w_{A1k}\, w_{A2k}\, \sigma_k$$

where σ_k is the variance of the random scalar factor for sector k.

The dependence structure between obligors can be exploited to provide an adequate genetic modelling of the decision variables for the given portfolio data by sorting the obligors in ascending order according to the level of default dependency. This can be performed using a simple sorting procedure that exploits the structure of the sector weights and the variation coefficients of the sector random variables, see Schlottmann and Seese [10, 10 - 11] for a detailed example and further explanations.

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

$$f(x, p, \sigma, w, \eta) := \frac{ret(x, p, \eta)}{risk(x, p, \sigma, w)}$$

This is a common definition of a risk-adjusted performance measure like RAROC and similar concepts. Considering the specification of the return function in Definition 2 as well as the chosen risk measure UL_ε we obtain:

$$f(x, p, \sigma, w, \eta) := \frac{\sum_{A=1}^{K} x_A(\eta_A - p_A)}{UL_\varepsilon(x, p, \sigma, w)} . \qquad (10)$$

If we maximize this function f, we will implicitly maximize the net return and minimize UL_ε, and this will drive the portfolio structure x towards the set of global Pareto-optimal portfolio structures (cf. the domination criteria specified in Definition 6). The local search variation operator shown in Algorithm 2 exploits this property and besides that, it also respects the given supervisory capital constraint.

Algorithm 2. *Local search variation operator*

1: **For each** $x \in P(t)$ *execute the following lines with probability* p_{local}
2: **If** $cap(x, c) > B$ **Then**
3: $D := -1$
4: **End If**
5: **If** $cap(x, c) \leq B$ **Then**
6: *Choose* D *between* $D := 1$ or $D := -1$ *with uniform probability 0.5*
7: **End If**
8: $\forall A : \widehat{x_A} := x_A, Step := 0$
9: **Do**
10: $\forall A : x_A := \widehat{x_A}, Step := Step + 1$
11: $ret_{old} := \sum_{A=1}^{K} x_A\,(\eta_A - p_A)$ *and* $risk_{old} := UL_\varepsilon(x, p, \sigma, w)$
12: **For each** x_J *calculate partial derivative* $d_J := \frac{\partial}{\partial x_J}(x, p, \sigma, w, r)$
13: **If** $D = -1$ **Then**
14: *Choose* $\hat{A} := \arg\min_J \{d_J \mid x_J > 0\}$
15: *Remove this exposure from portfolio :* $\widehat{x_{\hat{A}}} := 0$
16: **Else**
17: *Choose* $\hat{A} := \arg\max_J \{d_J \mid x_J = 0\}$
18: *Add this exposure to portfolio:* $\widehat{x_{\hat{A}}} := v_{\hat{A}}$
19: **End If**

20: $ret_{new} := \sum_{A=1}^{K} \hat{x}_A (\eta_A - p_A)$ *and* $risk_{new} := UL_\varepsilon(\hat{x}, p, \sigma, w)$

21: **While** $Step \le Step_{max} \wedge (\exists A : \hat{x}_A > 0) \wedge (\exists J : \hat{x}_J = 0) \wedge \hat{x} \notin P(t) \wedge \hat{x} \notin Q(t) \wedge ((D = -1 \wedge cap(\hat{x}, c) > B) \vee (D = 1 \wedge cap(\hat{x}, c) \le B \wedge (ret_{new} > ret_{old} \vee risk_{new} < risk_{old})))$

22: *Replace x in P(t) by its optimized version*

23: **End For**

24: **Output:** *P(t)*

If the current solution x from $P(t)$ to be optimized with probability p_{local} (which is a parameter of the algorithm) is infeasible because the capital restriction is violated (cf. line 2 in Algorithm 2), the algorithm will remove the obligor having the minimum gradient component value from the portfolio (lines 14 and 15). This condition drives the hybrid search algorithm towards feasible solutions. In the case of a feasible solution that is to be optimized, 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.

Table 1: Overview of data and parameters

Parameter	Value
K (# obligors)	20
N (# systematic risk factors)	1
p_A (probability of default)	2% to 7%
Variation coefficient of default probabilitty	0,75
$\frac{(\sum_{A=1}^{K} \nu_A)^2}{\sum_{A=1}^{K} \nu_A^2}$ (granularity of portfolio)	15,7
L_0 (loss unit)	$\frac{max_{A \in \{1, \ldots, K\}} \{\nu_A\}}{100}$
ε (confidence level)	0,99
$\frac{B}{\sum_{A=1}^{K} c_A \nu_A}$ (constraint level)	2/3
$\lvert P(t) \rvert$ (population size)	30
p_{cross} (crossover probability)	0,95
p_{mut} (mutatio probability)	1/20
p_{local} (local search probability)	0,10
$Step_{max}$ (max. local search iterations)	4
t_{diff} (termination condition)	100

The partial derivative d_J for obligor J required in line 12 of Algorithm 2 can be obtained by evaluation of the following formula (a proof is provided in the appendix):

$$d_J = \frac{x_J(\eta_J - p_J)\, UL_\varepsilon(x, \ldots) - \sum_{A=1}^{K} x_A (\eta_J - p_J) r_J(UL_\varepsilon(x, \ldots))}{x_J\, (UL_\varepsilon(x, \ldots))^2} \qquad (11)$$

where we abbreviated $UL_\varepsilon(x,\ldots) := UL_\varepsilon(x,p,\sigma,w)$ and $r_J(UL_\varepsilon(x,\ldots))$ is the marginal risk contribution for obligor J to the unexpected loss, cf. the corresponding remarks in Chapter 2.

Remembering the fact that the risk contributions, and therefore, the partial derivatives d_J can be calculated efficiently (e.g. in KN computational steps for K obligors and N sectors if the approximation suggested by [2] is used[5]) for an individual that has already a valid fitness evaluation, this yields a very fast variation operator.

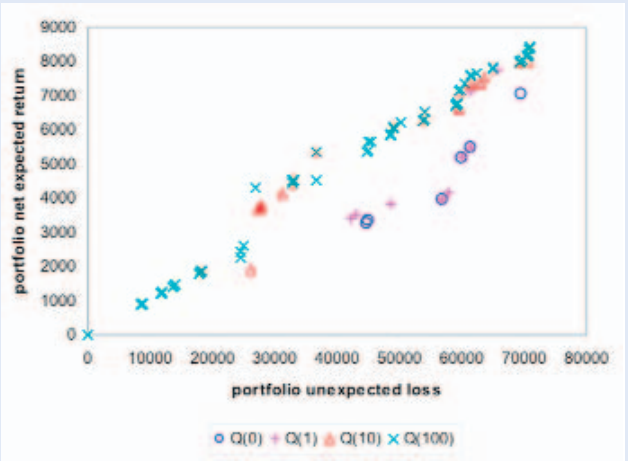
Besides the proposed use of the risk contributions suggested by CSFP, the above algorithm can easily be adapted to other types of risk contributions or calculation schemes, see Chapter 3 by Tasche for more details.

The local search algorithm terminates if at most $Step_{max}$ iterations have been performed (parameter $0 < Step_{max} \le K$), 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.

3.3. Application to a sample loan portfolio

We now focus on a small sample portfolio containing middle-market loans to illustrate the proposed method of risk-return analysis in the CreditRisk+ model framework. Table 1 gives a short overview of the characteristics of the data and the chosen parameters.[6]

Figure 3: Evolution of individuals within Q(t)

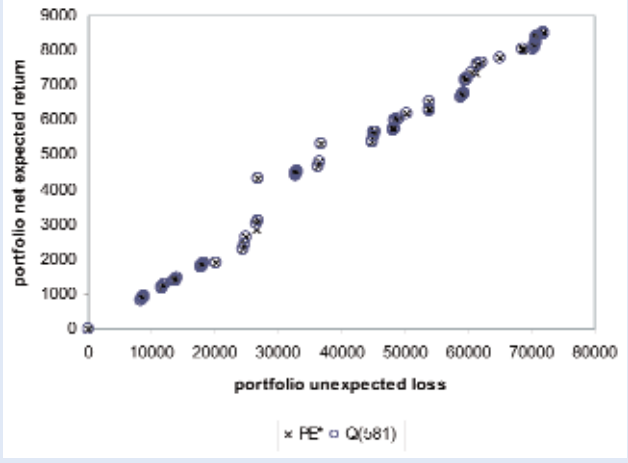


In our example, we used $|P(t)| = 30$ individuals in each population step t, which is a compromise between developing a diverse set of solutions in $P(t)$ and the corresponding computational effort spent per step t. Due to the elite population and the local search variation operator in the HMOEA, the choice of this parameter is not very crucial for this algorithm. In other tests using different portfolio sizes, we chose $20 \leq |P(t)| \leq 100$, and this parameter range is a common choice in other Evolutionary Algorithm applications, too. The common parameter setting of $p_{cross} := 0.95$ and $p_{mut} := 1/K$ is reported to work well in many other studies using elitist Evolutionary Algorithms, and this was also supported by test results during development of the HMOEA.

The choice of p_{local} and $Step_{max.}$ can be made by the respective user of the HMOEA depending on her preferences: if one is interested in finding better solutions in earlier populations, then both the probability and the number of local search iterations given an application of the respective variation operator will 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 optimization pressure should not be too high, since one is usually also interested in finding a diverse set of solutions. A choice of $0 < p_{local} \leq 0.1$ and $Step_{max} << K$ (which means the parameter is significantly lower than K) is also preferable concerning the additional computational effort to be spent by the local search variation operator. In our tests, we use a parameter set of $p_{local} := 0.1$ and $Step_{max.} := 4$ for the HMOEA which also yielded promising results in other test cases consisting of significantly different loan portfolio data sets.

Figure 3 illustrates the evolution of the individuals within $Q(t)$ during a run of the HMOEA until $t = 100$. The objective function values of the six individuals within the first elite population $Q(0)$ are indicated by circles in Figure 3. The elite population $Q(1)$ obtained after the following iteration of lines 5 to 15 of Algorithm 1 contains the individuals that have the objective function values marked by a plus sign. For instance, the two individuals from $Q(0)$ whose objective function values are approximately (45000,3500) are dominated by the two individuals from $Q(1)$ that are close to them but have higher portfolio return and/or less portfolio unexpected loss values. Hence, the former two members of $Q(0)$ are no longer members of $Q(1)$. On the other hand, three elements from $Q(0)$ are not dominated within the next iteration of the evolution process, hence they remain in $Q(1)$. This is indicated by the three circles that contain a plus sign.

Figure 4: Comparison of PE* to Q(581)



Until $t = 10$ the individuals in the elite population have moved significantly in the relevant direction towards the upper left corner of the displayed risk-return space, and the cardinality of the elite population has been raised to 33. Several members of $Q(10)$ even cannot be improved until $t = 100$,which can be seen in the upper right corner of Figure 3. However, the members of $Q(100)$ have a larger spread over the two-dimensional objective function space due to individuals which have been found in the lower left area of this figure. The cardinality of $Q(100)$ is 68.

Summarizing the elite populations shown, Figure 3 displays typical convergence properties during a run of the hybrid algorithm: In early iterations, the algorithm mainly improves the domination-related objective function values. Over time, more solutions are discovered and the diversity of the individuals in the objective function space is raised due to the crowding-sort procedure described in Section 3.1.

Our test data set is small enough to allow a complete enumeration of the search space, i.e. the feasible Pareto-optimal set PE^* can be computed within reasonable computing time to verify the solution quality of the approximation set computed by the HMOEA after a total of $t = 581$ population steps.

Figure 4 displays a comparision between the objective function values of the 100 elements within PE^* and those of the 97 members of the final elite population $Q(581)$. The hybrid approach required 43 seconds for the computation of the approximation set $Q(581)$ on a standard PC (2 GHz single CPU), whereas the enumeration took 1318 seconds (approx. 22 minutes) of computation time for PE^*. A visual inspection reveals that $Q(581)$ is a good approximation set for PE^* since the points of PE^* (indicated by "x") are approximated by mostly identical or at least very close points of $Q(581)$ which are marked by a respective circle in Figure 4.

In many MOEA applications, actually a single run of the algorithm is performed to obtain an approximation of the feasible Pareto-optimal solutions as shown in the above example. We performed a total of 50 independent runs of the HMOEA on the test problem using different pseudorandom number sequences to obtain 50 approximations of PE^*. The average runtime over 50 independent runs of the HMOEA was only 33 seconds, and no single run required more than 54 seconds. In each independent run, the HMOEA found the two boundary solutions, which is a desirable result for the stochastic search algorithm. The cardinality of the approximation set was 96.3 on average and ≥ 91 in all

runs, and the spread of the approximation solutions in the objective function space was similar to Figure 4 in each single run.

Remembering that adding one obligor to the portfolio doubles the size of the search space, which means that the enumeration effort increases exponentially with the number of obligors, we expect that the gap between the runtime of the enumeration and the runtime for obtaining an approximation using the HMOEA increases significantly. For a suitably large portfolio, a complete enumeration cannot be performed within reasonable time given a bounded runtime on a specific computer. Hence, the hybrid approach is particularly useful to compute approximations for PE^* in these cases. For instance, we have computed an approximation set for a non-homogeneous portfolio containing 386 corporate obligors of a German bank on the specified PC within approximately 1 hour.

Concerning larger portfolio dimensions, a few issues have to be kept in mind. First, the computational effort to calculate or approximate UL_ε strongly influences the runtime of the HMOEA. Thus, the choice of the calculation procedure for the loss distribution and the corresponding percentile has to be made with respect to the tradeoff between accuracy of the results and the required computation time. In our case, the extent of loss discretization in the standard CreditRisk+ model (i.e. choice of L_0) strongly influences the computational effort to be spent on the calculation of the risk objective function. A finer discretization yields more precise results but at the cost of a higher runtime, whereas less precise UL_ε results can be achieved in less runtime. We have to point out that this issue is not specific to our algorithm but to all algorithms that require a large number of risk objective function evaluations during the search for risk-return efficient solutions. Alternative methods of approximating UL_ε offer potentially faster approximations that might be interesting for larger portfolios, see Chapters 5 to 8 for more details.

Secondly, the usual convergence results for Evolutionary Algorithms assume $t \to \infty$, whereas in practical applications our hybrid algorithm has to be stopped after a finite number of population steps t. Therefore, it is important to remember that $Q(t)$ is an approximation and not necessarily identical to PE^*, particularly for large portfolios having a huge search space.

Thirdly, since evolutionary algorithms are well-suited for distributed computation or parallel implementation (see e.g. [11]) there is a good prospect of improving the speed of our approach by using more than one CPU at least for some tasks to process large-dimensional portfolios.

4. Conclusion and Outlook

We formally defined a constrained multi-objective problem of risk-return analysis for a credit portfolio based on binary decision variables. The aggregated expected net return from a potential portfolio structure and the corresponding aggregated downside risk measured by unexpected loss were considered as objective functions of a bank also having an additional capital budget restriction imposed e.g. by supervisory capital.

For the approximation of the set of feasible Pareto-optimal solutions to our problem of portfolio risk-return analysis, we discussed a hybrid approach that combines concepts from different multi-objective evolutionary algorithm schemes and a problem-specific local search operator based on a risk-adjusted performance measure. This hybrid approach is not restricted to linear or convex objective functions and is also flexible concerning the constraints. We implemented the CreditRisk+ portfolio credit risk model into the algorithm and derived a local search operator that exploits model-specific features. Although our implementation was executed on a single standard desktop PC, the hybrid algorithm found approximations of almost all feasible, Pareto-optimal solutions within seconds for our small test portfolio. Moreover, we have pointed out important issues for applications of our proposed hybrid method to larger problem instances.

Besides the integration of another variant of the CreditRisk+ (cf. the chapters in the first three parts of this volume), further research from the viewpoint of risk measurement could focus on alternative objective functions like expected shortfall, tail conditional expectation and related measures (see e.g. Chapter 3), and in this case an appropriate local search operator can easily be implemented. Moreover, the multi-objective evolutionary approach supports the incorporation of more than two objective functions without significant changes, which might e.g. be very interesting for hold-to-maturity calculations as proposed in the CreditRisk+ manual [2].

Due to the flexibility of our algorithm, many further constraints of practical interest can be considered, for instance the simultaneous use of different capital budgets or unexpected loss limits on subsets of obligors (e.g. depending on obligor-specific criteria like country or industry) in the risk-return analysis. Even more sophisticated restrictions can be handled, e.g. restrictions on the structure of the parts of a portfolio to be sold in an asset-backed security transaction that is itself calculated using a non-linear pricing model as described in the Chapters 18 by Kluge and Lehrbass and 19 by Hellmich and Steinkamp.

5. Appendix: Proof of Formula (11)

Given a portfolio structure specified by $x, p, \sigma, \omega, \eta$, the partial derivative of the function f given by (10) is calculated using the quotient rule and the abbreviation $UL_\varepsilon(x, \ldots) := UL_\varepsilon(x, p, \sigma, \omega)$:

$$d_J := \frac{\partial}{\partial x_J} f(x, p, \sigma, w, \eta)$$

$$= \frac{(\eta_J - p_J)\ UL_\varepsilon(x, \ldots) - (\sum_{A=1}^{K} x_A\ (\eta_A - p_A)) \frac{\partial}{\partial x_J} (UL_\varepsilon(x, \ldots))}{(UL_\varepsilon(x, \ldots))^2}$$

Assuming $x_J \neq 0$ yields

$$d_J = \frac{x_J(\eta_J - p_J)\ UL_\varepsilon(x, \ldots) - (\sum_{A=1}^{K} x_A\ (\eta_A - p_A))x_J \frac{\partial UL_\varepsilon(x, \ldots)}{\partial x_J}}{x_J\ (UL_\varepsilon(x, \ldots))^2}$$

Using the relationship

$$r_J(UL_\varepsilon(x, p, \sigma, \omega)) \equiv x_J \frac{\partial}{\partial x_J} (UL_\varepsilon(x, p, \sigma, \omega))$$

(cf. Chapters 2 and 3) we finally obtain

$$d_J = \frac{x_J(\eta_J - p_J)\ UL_\varepsilon(x, \ldots) - (\sum_{A=1}^{K} x_A\ (\eta_A - p_A))r_J\ (UL_\varepsilon(x, \ldots))}{x_J\ (UL_\varepsilon(x, \ldots))^2}$$

Autoren:

Dipl. Wi.-Ing. Dr. Frank Schlottmann
Studium und Promotion an der Universität Karlsruhe (TH). Parallel dazu unternehmerische Tätigkeit im Bereich Informationstechnologie-Consulting und -Training. Seit 1994 bei GILLARDON tätig in den Bereichen Entwicklung, Beratung und Projekte mit aktuellem Schwerpunkt Kreditrisiko und Research. Zahlreiche internationale Publikationen und Vorträge im Bereich des finanziellen Risikomanagements.

Prof. Dr. Detlef Seese
Leiter der Forschungsgruppe Komplexitätsmanagement am Institut für Angewandte Informatik und Formale Beschreibungsverfahren der Universität Karlsruhe (TH).

Dipl. Math. Dipl.-Wirtsch.-Inform. Stephan Vorgrimler
Studium der Mathematik und Wirtschaftsinformatik an der Universität Mannheim. Seit 1996 bei GILLARDON tätig mit den Schwerpunkten Kreditrisiko und Research.

References:

[1] Basel Committee for Banking Supervision. The new basel capital accord. 2003. Bank for International Settlements, Basel, http://www.bis.org/bcbs/cp3full.pdf.

[2] CreditSuisse Financial Products. CreditRisk+™ - a credit risk management framework. 1997. http://www.csfb.com/institutional/research/credit_risk.shtml.

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

[4] David Goldberg. Genetic algorithms for search, optimisation and machine learning. Addison-Wesley, Reading, 1989.

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

[6] Frank Lehrbass. Rethinking risk-adjusted returns. Credit Risk Special Report, Risk, (4):35–40, 1999.

[7] Harry Markowitz. Portfolio selection. Journal of Finance, 7:77ff., 1952.

[8] Guenter Rudolph and Alexandru Agapie. Convergence properties of some multi-objective evolutionary algorithms. In A. Zalzala, editor, Proceedings of the 2000 Congress on Evolutionary Computation, pages 1010–1016. IEEE Press, Piscataway, 2000.

[9] Jonathan Schaffer. Some experiments in machine learning using vector evaluated genetic algorithms. PhD thesis, Vanderbilt University, Nashville, 1984.

[10] Frank Schlottmann and Detlef Seese. A hybrid genetic-quantitative method for risk-return optimisation of credit portfolios. In Carl Chiarella and Eckhard Platen, editors, Quantitative Methods in Finance 2001 Conference abstracts, page 55. University of Technology, Sydney, 2001. Full paper: http://www.defaultrisk.com/pp_model_31.htm.

[11] Hartmut Schmeck, Juergen Branke, and Udo Kohlmorgen. Parallel implementations of evolutionary algorithms. In A. Zomaya, F. Ercal, and S. Olariu, editors, Solutions to parallel and distributed computing problems, pages 47–68. John Wiley & Sons, Chichester, 2001.

[12] Detlef Seese and Frank Schlottmann. The building blocks of complexity: a unified criterion and selected problems in economics and finance. 2002. Presented at the Sydney Financial MathematicsWorkshop 2002, http://www.qgroup.org.au/SFMW/8may02.html.

1 Risk-return efficient sets consist of Pareto-optimal solution with respect to the objective functions portfolio risk and portfolio return. A formal definition is given later.

2 Alternativly, the variation coefficient of the gamma-distributed random scalar factors in the CreditRisk+ model can be given, see the remarks in Chapter 2. In this case, no obligor-specific standard deviation of default probability is required.

3 The computational complexity class P contains all problems which can be solved by a deterministic algorithm in polynomial time (measured by the size of the input). NP contains all problems for which a deterministic algorithm can verify in polynomial time whether a given solution candidate actually is a solution. A problem is NP-hard if all problems in the class NP can be reduced to it.

4 This is due to the non-convexity of the Value-at-Risk downside risk measure.

5 Using the common *O*-notation the computational effort is *O(KN)*, hence the required number of operations grows linearly in the number ofobligors.

GILLARDON – innovative Lösungen für die Finanzwirtschaft

Die Lösungen

Unsere Kernkompetenzen umfassen die Bereiche Kundenberatung, Produktkalkulation und Gesamtbanksteuerung.

Kundenberatung

evenit™ ist das themenorientierte Beratungssystem für alle Vertriebskanäle für die Themen Altersvorsorge, Baufinanzierung, Vermögensanalyse und Financial Planning.

Produktkalkulation

MARZIPAN™ ist die Lösung zur Produktberatung und -kalkulation von Aktiv- und Passivgeschäften auf Basis der Marktzins- und Barwertmethode.

FinanceFactory™ ist das regelbasierte Kalkulationssystem für die Absatzfinanzierung, das alle Darlehensvarianten der Absatzfinanzierung inklusive Restkreditversicherung und Subventionsrechnung abdeckt.

Gesamtbanksteuerung

THINC™ ist die integrierte Softwarelösung zur wertorientierten Gesamtbanksteuerung und deckt die Themen Markt- und Vertriebssteuerung, Bilanzstrukturmanagement, Risikocontrolling, Treasury, Adressrisikosteuerung, Basel II und IAS / IFRS ab. THINC unterstützt Sie bei der Erfüllung der Anforderungen aus den MaRisk.

GILLARDON ist Branchenspezialist für Softwarelösungen, Consulting und Seminare in den Themenbereichen Kundenberatung, Produktkalkulation und Gesamtbanksteuerung.