

Chapter 8

ALGORITHMS FOR OPTIMIZATION OF VALUE-AT-RISK*

Nicklas Larsen

*Royal Institute of Technology (KTH),
Division of Optimization and Systems Theory,
Dept. of Mathematics,
Stockholm, Sweden
f94-nla@f.kth.se*

Helmut Mausser

*Algorithmics, Inc.,
185 Spadina Avenue,
Toronto, Ontario M5T 2C6, Canada*

Stanislav Uryasev¹

*University of Florida,
Dept. of Industrial and Systems Engineering,
PO Box 116595, 303 Weil Hall,
Gainesville, FL 32611-6595
URL: <http://www.ise.ufl.edu/uryasev>
uryasev@ufl.edu*

Abstract This paper suggests two new heuristic algorithms for optimization of Value-at-Risk (VaR). By definition, VaR is an estimate of the maximum portfolio loss during a standardized period with some confidence level. The optimization algorithms are based on the minimization of the closely related risk measure Condi-

*This paper is based on the research project conducted by Nicklas Larsen during his visit to the University of Florida in Spring 2000. This research project has not been conducted in the framework of any legal agreement between the University of Florida and Algorithmic, Inc.

¹The correspondence author

tional Value-at-Risk (CVaR). For continuous distributions, CVaR is the expected loss exceeding VaR, and is also known as Mean Excess Loss or Expected Shortfall. For discrete distributions, CVaR is the weighted average of VaR and losses exceeding VaR. CVaR is an upper bound for VaR, therefore, minimization of CVaR also reduces VaR. The algorithms are tested by minimizing the credit risk of a portfolio of emerging market bonds. Numerical experiments showed that the algorithms are efficient and can handle a large number of instruments and scenarios. However, calculations identified a deficiency of VaR risk measure, compared to CVaR. Minimization of VaR leads to an undesirable stretch of the tail of the distribution exceeding VaR. For portfolios with skewed distributions, such as credit risk, minimization of VaR may result in a significant increase of high losses exceeding VaR. For the credit risk problem studied in this paper, VaR minimization leads to about 16% increase of the average loss for the worst 1% scenarios (compared to the worst 1% scenarios in CVaR minimum solution). 1% includes 200 of 20000 scenarios, which were used for estimating credit risk in this case study.

1. Introduction

Traditionally used tools for assessing and optimizing market risk assume that the portfolio return is normally distributed. In this way, the two statistical measures, mean and standard deviation, can be used to balance return and risk. However, often, as in the case of credit losses, the distributions of losses are far from normal; they are heavily skewed, with a long fat tail. In the case of credit losses, the distribution is a result of the fact that an obligor rarely defaults or changes credit rating, but when default occurs losses are generally substantial.

Value-at-Risk (VaR) is by far the most popular and most accepted risk measure among financial institutions. VaR is an estimate of the maximum potential loss with a certain confidence level, which a dealer or an end-user of financial instruments would experience during a standardized period (e.g. day, week, or year). In other words, with a certain probability, losses will not exceed VaR. Various issues related to risk management using VaR are studied in [4, 5, 6, 11, 12, 14, 15, 16, 17, 20, 22, 24, 27, 28, 30, 31, 32]. However, this list of references is far from complete. Other publications related to VaR can be found at <http://www.gloriamundi.org>.

Although popular, mathematically VaR has some serious limitations, such as lack of sub-additivity [5]. In the case of a finite number of scenarios, VaR is a nonsmooth, nonconvex, and multiextreme function with respect to positions [24], making it difficult to control and optimize. Therefore, in spite of significant research efforts [4, 6, 12, 15, 16, 17, 20, 30, 32], efficient algorithms for optimization of VaR for reasonable dimensions (over one hundred instruments and one thousand scenarios) are still not available. This fact stimulated our development of the new optimization algorithms presented in this paper.

Papers [35, 36] has considered an alternative risk measure called *Conditional Value-at-Risk* (CVaR). This measure, for continuous distributions, is also known as Mean Excess Loss, Expected Shortfall, or Tail VaR. However, for discrete distributions, CVaR may differ from Expected Shortfall. By definition, for continuous distributions, α -CVaR is the expected loss exceeding α -VaR, i.e., it is the mean value of the worst $(1-\alpha)*100\%$ losses. For instance at $\alpha=0.99$, CVaR is the average of the 1% worst losses. For general loss distributions (including discrete distributions) CVaR is defined as the weighted average of VaR and conditional expectation of losses strictly exceeding VaR. However, recently, Acerbi et al. [1, 2] redefined Expected Shortfall risk measure in a manner consistent with definition of CVaR; also they have proved several important properties of CVaR including asymptotical convergence of statical estimates to CVaR.

There are several reasons why CVaR is a preferred risk measure to VaR. To begin with, CVaR is a sub-additive measure of risk compared with VaR which is not sub-additive [35, 36, 27]. Sub-additivity means that diversification of a portfolio reduces CVaR but may increase VaR. Also, VaR does not provide any information about the amount of loss exceeding VaR. Moreover, conducted in this paper numerical experiments showed that minimization of VaR leads to an undesirable stretch of the tail of the distribution exceeding VaR. For portfolios with skewed distributions, risk management using VaR may result in a significant increase in high losses. Similar effect also was observed in paper [38]. CVaR quantifies losses exceeding VaR and acts as an upper bound for VaR. Therefore portfolios with a low CVaR, also have a low VaR. Under quite general conditions, CVaR is a convex function with respect to positions [35]. Moreover, CVaR is a *coherent* measure of risk in the sense of [5] (coherency of CVaR was first proved in [27], see also [36, 1, 2]). It was shown in [35], that CVaR can be minimized using *linear programming techniques*, allowing handling of portfolios with a large number of instruments and scenarios.

However, considering VaR's popularity and the fact that government regulations demand that financial institutions control VaR, there is a need for algorithms that minimize VaR. Utilizing the fact that CVaR minimization can be done very efficiently, this paper examines two heuristic algorithms for minimization of VaR. Although the considered algorithms are quite general, numerical experiments were concentrated on optimization of credit risk. Compared to other algorithms that have been applied to the same test portfolio as we have used in this paper, we achieve the lowest VaR so far.

Credit risk is the risk of a trading partner not fulfilling his obligations in full on due date or at any time thereafter. The loss can result from counter party default, but also from a decline in market value stemming from a credit quality migration of an issuer or counter party. The efficient management of credit risk can save economic capital and protect an institution from undesirable high

levels of risk, see for instance [21, 33, 34]. Credit risk management includes not only monitoring risk, but also efficient restructuring of portfolios to reduce risks and maximize returns. Several approaches are available for estimating credit risk [9, 10]. Probably, the most influential contribution in this field has been CreditMetrics methodology [9]. Bucay and Rosen [8] conducted a case study and applied the CreditMetrics methodology to a portfolio of bonds issued in emerging markets. Mausser and Rosen [25] applied the regret optimization framework to minimize the credit risk of this portfolio. Using the CreditMetrics methodology, a large number of scenarios is generated based on credit events such as default and credit migration. The portfolio is revalued under each scenario resulting in a portfolio loss distribution. We used the credit risk model for the portfolio of bonds [8] to test the new algorithms.

Concepts similar to VaR and CVaR have earlier been studied in the stochastic programming literature, although not in a financial mathematics context [7, 13, 18, 19, 29]. The reader interested in other applications of optimization techniques within the finance area can find relevant papers in [37].

The rest of this paper is organized as follows. The second chapter defines the VaR and CVaR risk measures. In this chapter, we also describe the two algorithms for optimization of VaR. In the next chapter we apply the algorithms to the minimization of credit risk for a portfolio of emerging market bonds. The forth chapter presents the analysis of the results and compare our findings with previous studies made with the same test portfolio. We then present our conclusions. The appendix contains a short description of the Credit Metrics framework and the portfolio of emerging market bonds.

2. Problem Statement and Algorithms

2.1. Value-at-Risk (VaR): Problem Statement

Let $f : \mathbf{R}^n \times \mathbf{R}^m \rightarrow \mathbf{R}$ be a loss function which depends on the control vector \mathbf{x} belonging to the feasible set $X \subset \mathbf{R}^n$ and a random vector $\mathbf{y} \in \mathbf{R}^m$. We suppose the random vector \mathbf{y} is governed by a probability measure P that is independent of \mathbf{x} . For each x , we denote by $\Psi(\mathbf{x}, \cdot)$ on \mathbb{R} the resulting distribution function for the loss $\mathbf{z} = f(\mathbf{x}, \mathbf{y})$, i.e.,

$$\Psi(\mathbf{x}, \zeta) = P\{\mathbf{y} \mid f(\mathbf{x}, \mathbf{y}) \leq \zeta\}. \quad (1)$$

By definition, (1) is the probability that the loss function $f(\mathbf{x}, \mathbf{y})$ does not exceed the threshold ζ . The VaR function, $\zeta_\alpha(\mathbf{x})$, which is a quantile of the loss distribution is defined in the following way

$$\zeta_\alpha(\mathbf{x}) = \min\{\zeta \in \mathbf{R} : \Psi(\mathbf{x}, \zeta) \geq \alpha\}. \quad (2)$$

In this paper, we assume that losses are discretely distributed. This discrete loss distribution is obtained by drawing J samples, \mathbf{y}_j , $j = 1, \dots, J$ from

some underlying continuous distribution. Sample points, \mathbf{y}_j , $j = 1, \dots, J$ may coincide, therefore probabilities of discrete points are equal to multiples of $1/J$. Given a sample of size J , let the losses $L_j = f(\mathbf{x}, \mathbf{y}_j)$, $j = 1, \dots, J$, for a given decision vector \mathbf{x} be ordered $L_1 \leq L_2 \leq L_3 \leq \dots \leq L_J$. For instance, let $\alpha = k/J$ (e.g., if $J = 100$ and $\alpha = 0.95$, $k = 95$). In this case, according to definition (2), VaR equals¹ L_k .

Here, we study the following problem of minimizing VaR

$$\min_{\mathbf{x} \in \mathbf{X} \subset \mathbf{R}^n} \zeta_\alpha(\mathbf{x}). \quad (3)$$

For discrete distributions, VaR is a nonsmooth, nonconvex, and multiextreme function with respect to the vector \mathbf{x} , which makes it difficult to control and optimize (see, for instance, [24]). Also, VaR has some other undesirable properties, such as lack of sub-additivity [5]. Figure 8.1 gives an example of how VaR for a portfolio depends on a single position. Because of the multiextremum structure of the VaR function, it is difficult to optimize it with standardly available methods. In this paper we use the closely related risk measure, CVaR, to construct the optimization algorithms for VaR.

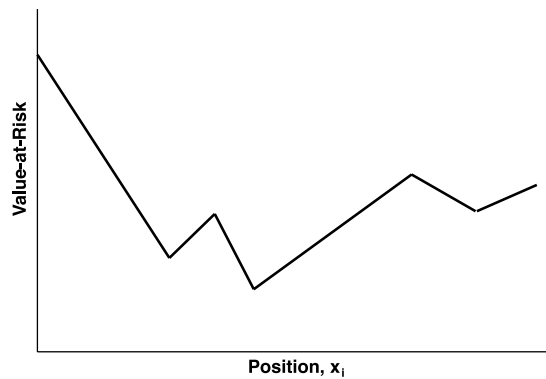


Figure 8.1. Simulation-based Value-at-Risk profile generated by sampling from continuous distribution.

2.2. Conditional Value-at-Risk (CVaR)

For continuous distributions, CVaR is defined as the conditional expected value of losses under condition that they exceed VaR, which is denoted by

¹in some other papers, including [25, 26], VaR is defined to be equal L_{k+1} .

$\zeta_\alpha(\mathbf{x})$. We can write the CVaR function, $\phi_\alpha(\mathbf{x})$, as follows

$$\phi_\alpha(\mathbf{x}) = (1 - \alpha)^{-1} \int_{f(\mathbf{x}, \mathbf{y}) \geq \zeta_\alpha(\mathbf{x})} f(\mathbf{x}, \mathbf{y}) p(\mathbf{y}) d\mathbf{y} , \quad (4)$$

where $p(\mathbf{y})$ is a density function. For general distributions, including discrete distributions, CVaR is defined as the weighted average of VaR and losses strictly exceeding VaR. Denote by $\phi_\alpha(x)^+$ the conditional expectation of losses² strictly exceeding VaR. For general distributions, CVaR is defined as follows

$$\phi_\alpha(x) = \lambda \zeta_\alpha(\mathbf{x}) + [1 - \lambda] \phi_\alpha(x)^+ ,$$

where $\lambda = [\Psi(x, \zeta_\alpha(\mathbf{x})) - \alpha] / [1 - \alpha] \in [0, 1]$. CVaR is a coherent percentile risk measure³ having nice mathematical properties (see detail explanation of CVaR properties in [35, 36, 27]). In particular, CVaR is convex, which makes possible to construct efficient algorithms for controlling CVaR.

Minimizing CVaR. It is shown in [35, 36] that the minimization of the CVaR function, $\phi_\alpha(\mathbf{x})$, on the feasible set \mathbf{X} can be reduced to the minimization of the function,

$$F_\alpha(\mathbf{x}, \zeta) = \zeta + \frac{1}{1 - \alpha} E \left\{ [f(\mathbf{x}, \mathbf{y}) - \zeta]^+ \right\} , \quad (5)$$

on the set $X \times \mathbf{R}$, where $a^+ = \max\{0, a\}$. It can be verified that the function $F_\alpha(\mathbf{x}, \zeta)$, is convex with respect to (w.r.t.) ζ . Also, the function $F_\alpha(\mathbf{x}, \zeta)$ is convex w.r.t. \mathbf{x} , if the function $f(\mathbf{x}, \mathbf{y})$ is convex w.r.t. \mathbf{x} . Minimizing of function $F_\alpha(\mathbf{x}, \zeta)$ simultaneously finds VaR and the minimal CVaR value. The minimum of $F_\alpha(\mathbf{x}, \zeta)$ equals to the minimum of CVaR, optimal \mathbf{x} equals to the optimal decision vector, and smallest of optimal ζ equals to VaR.

To calculate the integral function $F_\alpha(\mathbf{x}, \zeta)$ we can use various approaches. If in formula (5) the expectation can be calculated or approximated analytically, then to optimize the function $F_\alpha(\mathbf{x}, \zeta)$ we can use nonlinear programming techniques. However, in this paper, we suppose that the expectation in the function $F_\alpha(\mathbf{x}, \zeta)$ is calculated (or approximated) using equally probable scenarios, $\mathbf{y}_j, j = 1, \dots, J$, i.e.,

$$E \left\{ [f(\mathbf{x}, \mathbf{y}) - \zeta]^+ \right\} = J^{-1} \sum_{j=1}^J [f(\mathbf{x}, \mathbf{y}_j) - \zeta]^+ . \quad (6)$$

²it is supposed that there exist losses strictly exceeding VaR.

³as it is defined in [5].

If both the loss function $f(\mathbf{x}, \mathbf{y}_j)$ and the feasible set X are convex, then to minimize CVaR we can solve the following *convex* optimization problem

$$\min_{\mathbf{x} \in X, \zeta \in \mathbf{R}} F_\alpha(\mathbf{x}, \zeta), \quad (7)$$

where

$$F_\alpha(\mathbf{x}, \zeta) = \zeta + \nu \sum_{j=1}^J [f(\mathbf{x}, \mathbf{y}_j) - \zeta]^+, \quad (8)$$

and the constant ν equals $((1-\alpha)J)^{-1}$. Moreover, if the loss function $f(\mathbf{x}, \mathbf{y}_j)$ is linear w.r.t. \mathbf{x} , and the set X is given by linear (in)equalities, then we can reduce the optimization problem (7) to the linear programming problem.

$$\min_{\mathbf{x} \in \mathbf{R}^n, \mathbf{z} \in \mathbf{R}^J, \zeta \in \mathbf{R}} \zeta + \nu \sum_{j=1}^J z_j \quad (9)$$

subject to

$$\mathbf{x} \in \mathbf{X}, \quad (10)$$

$$z_j \geq f(\mathbf{x}, \mathbf{y}_j) - \zeta, \quad z_j \geq 0, \quad j = 1, \dots, J, \quad (11)$$

where $z_j, j = 1, \dots, J$ are dummy variables.

2.3. Optimization Algorithms for VaR

In this section, we present two algorithms for minimization of VaR. We consider the case where all the assumptions which were stated for problem (9)-(11) are valid.

Informal Description of the Algorithms. By definition, α -VaR is a smallest value such that probability that loss will be less or equal to this values is bigger or equal to α . Let us consider a portfolio that has been simulated over J scenarios (in our case J equals 20,000). Based on the simulation, the portfolio's α -VaR is estimated as the loss L_k in a scenario k , where the total probability of all scenarios with losses smaller or being equal to L_k is at least α . Keeping track of this "tail" while optimizing α -VaR, requires integer variables. Therefore, VaR minimization, is a complicated mixed integer programming problem.

The general line of thought behind the considered heuristic algorithms is rather simple. Starting with the optimal portfolio obtained when applying the minimum CVaR approach as described in [3], we then systematically reduce VaR of the portfolio by solving a series of CVaR problems using linear programming techniques. These CVaR problems are obtained by constraining and "discarding" scenarios that show large losses. However, the two considered

algorithms, further denoted by Algorithms A1 and A2, differ in the procedure of generating the sequence of CVaR problems.

The approach in Algorithm A1 is to construct upper bounds for VaR and then to minimize these bounds. The first upper bound for α -VaR is the α -CVaR, which we minimize. Then, we split the scenarios whose losses exceed α -VaR, and “discard” the upper portion of these scenarios (see Figure 8.2). The number of scenarios that are discarded is determined by the parameter ξ (e.g., if ξ is equal to 0.5, then the upper half is discarded). Figure 8.2 shows the first step of the approach, when we discard scenarios with large losses and exclude them (make them “inactive”). Further, we calculate a new α_1 in such a way that CVaR with this new α_1 is an upper bound for VaR of the original problem. This α_1 -CVaR is the mean loss of the active scenarios with losses exceeding α -VaR, i.e., the scenarios between α -VaR and the dotted line in Figure 8.2. Then, we minimize this upper bound, and so on. To summarize, we construct a series of upper bounds and minimize them until we do not have anymore scenarios to discard. In the final step of this process we are coming to the heuristic procedure considered in [26]. The last step minimizes the loss L_k , while ensuring that losses in the scenarios exceeding L_k is kept above L_k . The approach requires solving a series of linear programming problems.

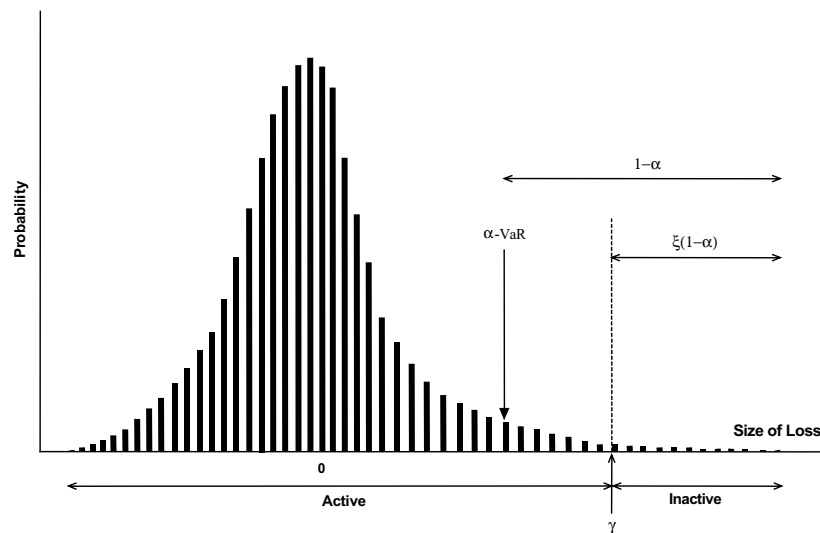


Figure 8.2. Algorithm A1. In the second step of Algorithm A1, we constrain and “discard” $\xi(1 - \alpha)J$ scenarios that show large losses (make them “inactive”). The new CVaR is generated in such a way that this CVaR is an upper bound to VaR.

Algorithm A2 is based on the idea that low values of VaR can be obtained by minimizing CVaR with some new α defined so that the values of the two measures coincide (or almost coincide). The first iteration after optimizing the

original CVaR problem will have a α_1 defined according to Figure 8.3. This α_1 is defined so that α -VaR and α_1 -CVaR coincide (or get as close as possible). Thus, compared to Algorithm A1, instead of constructing the upper bounds for VaR, we construct a series of CVaR problems closely approximating VaR.

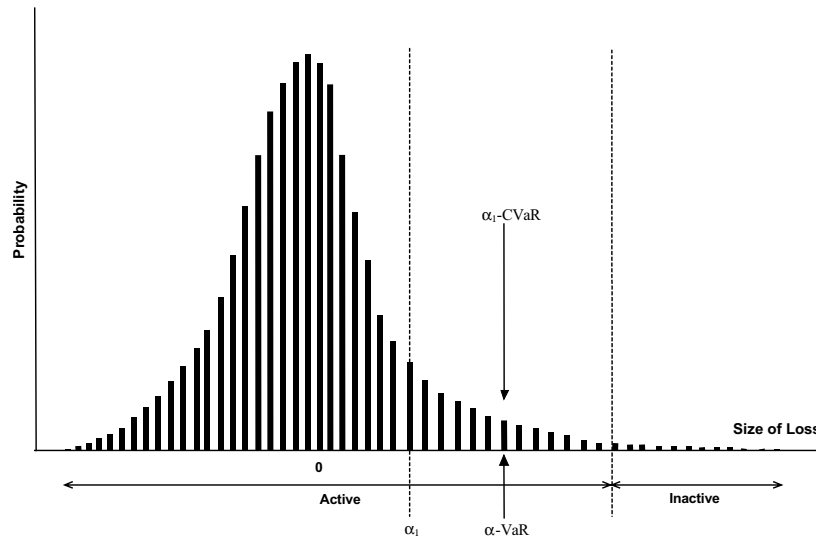


Figure 8.3. Algorithm A2. Construction of the CVaR problem in the second iteration. The α_1 is defined so that α -VaR and α_1 -CVaR coincide (or get as close as possible).

It may also be worth noting that, in both algorithms, once a scenario is rendered “inactive” its loss is no longer taken into consideration when solving the CVaR minimization problem. Hence, the actual α -CVaR, which still is the mean value of the $(1 - \alpha)J$ largest losses, will probably increase, as we further reduce α -VaR.

In the following sections, these two approaches will be explained in greater detail.

Algorithm A1. This section gives a formal description of Algorithm A1. Explanations are included after the formal description.

Step 0. Initialization

- i) Set $\alpha_0 = \alpha$, $i = 0$, $H_0 = \{j : j = 1, \dots, J\}$.
- ii) Assign a value for the constant ξ , $0 < \xi < 1$.

Step 1. Optimization subproblem

i) Minimize α_i -CVaR

$$\min_{\mathbf{x}, \mathbf{z}, \zeta, \gamma} \quad \zeta + \nu_i \sum_{j \in H_i} z_j \quad (12)$$

s.t.

$$\mathbf{x} \in \mathbf{X}, \quad (13)$$

$$z_j \geq f(\mathbf{x}, \mathbf{y}_j) - \zeta, \quad z_j \geq 0, \quad j \in H_i, \quad (14)$$

$$f(\mathbf{x}, \mathbf{y}_j) \leq \gamma \quad j \in H_i, \quad (15)$$

$$f(\mathbf{x}, \mathbf{y}_j) \geq \gamma \quad j \notin H_i, \quad (16)$$

where

$$\nu_i = ((1 - \alpha_i)J)^{-1}. \quad (17)$$

Denote the solution of this optimization problem by \mathbf{x}_i^* .

ii) With respect to the value of the loss function $f(\mathbf{x}_i^*, \mathbf{y}_j)$, order the scenarios, $j = 1, \dots, J$, in ascending order. Denote the ordered scenarios by j_ℓ , $\ell = 1 \dots, J$.

Step 2. Estimating VaR

Calculate VaR estimate, $\zeta_i = f(\mathbf{x}_i^*, \mathbf{y}_{\ell(\alpha)})$, where $\ell(\alpha) = \min\{\ell : \ell/J \geq \alpha\}$.

Step 3. Stopping the algorithm

If $H_i = H_{i-1}$ then stop the algorithm, and \mathbf{x}_i^* is the estimate of the optimal portfolio and VaR equals ζ_i .

Step 4. Reinitialization

- i) $i = i + 1$.
- ii) $b_i = \alpha + (1 - \alpha)(1 - \xi)^i$ and $\alpha_i = \alpha/b_i$.
- iii) $H_i = \{j_\ell \in H_{i-1} : \ell/J \leq b_i\}$.
- iv) Go to Step 1.

Further, we clarify some of the steps in Algorithm A1.

Step 0 initializes the algorithm by defining α_0 and setting the iteration counter to zero.

We say that scenarios included in the CVaR optimization subproblem (12) are *active*. Initially, all scenarios are active (the set is denoted by H_0). In the following steps, when solving the CVaR optimization subproblem, only the set of active scenarios, H_i , is considered. The so called *inactive* scenarios are the scenarios that has been excluded in the previous iterations. The parameter ξ

defines the portion of scenarios in the tail that will be excluded in each iteration. In the calculations presented later in the paper, ξ equals 1, 0.75, 0.5, 0.25 and 0.01. For instance, if $\xi = 0.5$, half of the tail is excluded in each iteration. Note that setting ξ equal to 1 would result in the heuristic optimization described in [26].

Step 1 solves the optimization subproblem of reducing α_i -CVaR, which is an upper bound for α -VaR. The variable γ is a free variable that ensures that losses in inactive scenarios exceed those in active scenarios. Note that dropping constraints (15) and (16) may in fact lead to an improved solution, although the algorithm must then be modified accordingly (i.e., an inactive scenario may become active if its loss declines sufficiently as a result solving the subproblem).

In Step 2, VaR is estimated by the loss in the scenario where the aggregated probability of the scenarios with losses lower or equal than this scenario, exceed or equal α .

In Step 3, the algorithm is stopped when an optimization of the subproblem has been made over just one active scenario, i.e., when we have minimized the loss in the scenario that is the α -VaR estimate.

In Step 4, α_i is defined in such a way that the α_i -CVaR, which is calculated only for the active scenarios, is an upper bound of the original α -VaR. Minimizing α_i -CVaR over the active scenarios, results in a minimization of the mean of the active tail *exceeding* α -VaR, as shown in Figure 8.2.

In Step 4, iii) the upper part of the active scenarios exceeding α -VaR is excluded from the set of active scenarios, H_i (i.e., made inactive). As illustrated in Figure 8.2, in the first iteration, the tail is split in two parts. The upper part of the tail is made inactive, and the lower part remains in the active scenario set H_1 .

Algorithm A2. In this section we present a formal description of Algorithm A2.

Step 0. Initialization

- i) Set $\alpha_0 = \alpha$, $i = 0$, $H_0 = \{j : j = 1, \dots, J\}$.
- ii) Assign a value for the constant ξ , $0 < \xi < 1$.

Step 1. Optimization subproblem

- i) Minimize α_i -CVaR

$$\min_{\mathbf{x}, z, \zeta, \gamma} \quad \zeta + \nu_i \sum_{j \in H_i} z_j \quad (18)$$

s.t.

$$\mathbf{x} \in \mathbf{X}, \quad (19)$$

$$z_j \geq f(\mathbf{x}, \mathbf{y}_j) - \zeta, \quad z_j \geq 0, \quad j \in H_i, \quad (20)$$

$$f(\mathbf{x}, \mathbf{y}_j) \leq \gamma \quad j \in H_i, \quad (21)$$

$$f(\mathbf{x}, \mathbf{y}_j) \geq \gamma \quad j \notin H_i, \quad (22)$$

where

$$\nu_i = ((1 - \alpha_i)J)^{-1}. \quad (23)$$

Denote solution of this optimization problem by \mathbf{x}_i^* .

- ii) With respect to the value of the loss function $f(\mathbf{x}_i^*, \mathbf{y}_j)$, order the scenarios, $j = 1, \dots, J$, in ascending order. Denote the ordered scenarios by j_ℓ , $\ell = 1 \dots, J$.

Step 2. Estimating VaR

- i) Calculate VaR estimate, $\zeta_i = f(\mathbf{x}_i^*, \mathbf{y}_{\ell(\alpha)})$, where $\ell(\alpha) = \min\{\ell : \ell/J \geq \alpha\}$.
- ii) For $i > 0$, calculate $\zeta_i = \min\{\zeta_{i-1}, \zeta_i\}$.

Step 3. Stopping the algorithm

If $H_i = H_{i-1}$ then Stop the algorithm.

If $\zeta_i < \zeta_{i-1}$ then \mathbf{x}_i^* is the estimate of the optimal portfolio and VaR equals ζ_i , otherwise, \mathbf{x}_{i-1}^* is the estimate of the optimal portfolio and VaR equals ζ_{i-1} .

Step 4. Reinitialization

- i) $i = i + 1$.
- ii) $b_i = \alpha + (1 - \alpha)(1 - \xi)^i$.
- iii) $H_i = \{j_\ell \in H_{i-1} : \ell/J \leq b_i\}$.
- iv) Calculate α_i so that α_i -CVaR for the active scenarios coincides with (or gets as close as possible to) α -VaR. This can be done as follows:
- (a) calculate number of active scenarios, $|H_i|$;
- (b) calculate

$$\ell_i = \min\left\{\ell : \frac{1}{|H_i| - \ell + 1} \sum_{\ell \leq t \leq |H_i|} f(\mathbf{x}_{i-1}^*, \mathbf{y}_{j_t}) \geq f(\mathbf{x}_{i-1}^*, \mathbf{y}_{\ell(\alpha)})\right\}$$

(c)

$$\alpha_i = \frac{\ell_i}{|H_i|}.$$

- v) Go to Step 1.

The algorithms A1 and A2 are quite similar. However, there are some differences. In Algorithm A1, we define α_i so that the CVaR minimization sub-problem is solved for the active scenarios exceeding α -VaR. In Algorithm A2, α_i is defined such that α_i -CVaR coincides with (or gets as close as possible to) α -VaR. In Algorithm A1, α -VaR can never increase while going through the different iterations. However, in Algorithm A2, this is not always the case. After each iteration, we must, therefore, check if α -VaR has been improved or not and choose the portfolio that has the lowest α -VaR.

3. Application of the VaR-minimization Algorithms to Credit Risk

This chapter applies the two algorithms discussed in the previous chapter to the problem of minimizing the Credit Risk of a portfolio of bonds. The portfolio is described in Appendix. First, we define the problem of minimizing VaR for the portfolio. Then, we describe how we have applied Algorithms A1 and A2 to this problem.

3.1. Minimum VaR model

Let $\mathbf{x} = (x_1, \dots, x_N)$ be the obligor weights expressed as multiples of current holdings and $\mathbf{b} = (b_1, \dots, b_N)$ be the future values of each instrument with no credit migration (benchmark scenario). Also, let $y_j, j = 1, \dots, J$, be the future, scenario-dependent, values with credit migration. In this case, J equals 20000. The loss function, i.e. the loss of the entire portfolio in scenario j , given obligor weights \mathbf{x} , is

$$f(\mathbf{x}, \mathbf{y}_j) = \sum_{i=1}^n (b_i - y_{ji}) x_i . \quad (24)$$

The loss function is linear w.r.t. the obligor weights, \mathbf{x} . The current market-to-market obligor exposures is denoted by \mathbf{c} , and the expected return for the obligors in the absence of credit migration is \mathbf{r} . The problem of minimizing α -VaR (see definition (2)) for a portfolio of bonds can then be expressed as follows:

$$\min_{\mathbf{x} \in \mathbf{R}^n} \zeta_\alpha(x) \quad (25)$$

s.t.

$$\sum_{i=1}^N b_i x_i = \sum_{i=1}^N b_i , \quad (26)$$

$$\sum_{i=1}^N c_i (r_i - R) x_i \geq 0 , \quad (27)$$

$$l_i \leq x_i \leq u_i \quad i = 1, \dots, N, \quad (28)$$

$$c_i x_i \leq 0.20 \sum_{j=1}^N c_j \quad i = 1, \dots, N. \quad (29)$$

Constraint (26) maintains the future portfolio value. In order to achieve an expected portfolio return, R , we can write the constraint for the expected portfolio return with no credit migration as

$$\frac{\sum_{i=1}^N (c_i x_i) r_i}{\sum_{i=1}^N c_i x_i} \geq R, \quad (30)$$

or equivalently as (27). To avoid unrealistic long and short positions in any of the holdings, we also impose constraints on the change in obligor weights. In constraint (28), the lower trading limit is denoted l_i , and u_i is the upper limit, both expressed as multiples of current weighting. In the calculation results presented in the next section, we have assumed no short positions and the only restriction on the upper limit is given by constraint (29), i.e. $l_i = 0$ and $u_i = \infty$. Constraint (29) implies that the value of each long individual position cannot exceed 20% of the current portfolio value.

3.2. Application of the Algorithms

In this section we describe how to apply Algorithms A1 and A2 to minimize VaR for a credit portfolio of emerging market bonds.

Algorithm A1. We start by minimizing CVaR for the portfolio.

$$\min \quad \zeta + \nu \sum_{j=1}^J z_j, \quad (31)$$

subject to

$$z_j \geq \sum_{i=1}^N ((b_i - y_{ji}) x_i) - \zeta, \quad j = 1, \dots, J, \quad (32)$$

$$\sum_{i=1}^N b_i x_i = \sum_{i=1}^N b_i, \quad (33)$$

$$\sum_{i=1}^N c_i (r_i - R) x_i \geq 0, \quad (34)$$

$$l_i \leq x_i \leq u_i, \quad i = 1, \dots, N, \quad (35)$$

$$c_i x_i \leq 0.20 \sum_{j=1}^N c_j \quad i = 1, \dots, N, \quad (36)$$

$$z_j \geq 0, \quad j = 1, \dots, J, \quad (37)$$

where $\nu = ((1 - \alpha)J)^{-1}$. The probability of each scenario is assumed to be equal.

After obtaining a solution, we order the losses from the smallest to the largest and calculate the cumulative probability. We obtain α -VaR as the loss in the scenario for which the cumulative probability first meets or exceeds α .

The next step is to define a α to be used in the minimum CVaR approach that allows us to minimize the “active” losses above VaR. This α_1 , with the index indicating the iteration count, is defined as follows

$$\alpha_1 = \frac{\alpha_0}{1 - \xi(1 - \alpha_0)}, \quad (38)$$

where α_0 is the original α .

Then we solve the first optimization subproblem, which is formulated as follows

$$\min \quad \zeta + \nu_1 \sum_{j=1}^{J_1} z_j \quad (39)$$

subject to

$$z_j \geq \sum_{i=1}^N ((b_i - y_{ji})x_i) - \zeta \quad j = 1, \dots, J_1, \quad (40)$$

$$\sum_{i=1}^N b_i x_i = \sum_{i=1}^N b_i, \quad (41)$$

$$l_i \leq x_i \leq u_i \quad i = 1, \dots, N, \quad (42)$$

$$c_i x_i \leq 0.20 \sum_{j=1}^N c_j \quad i = 1, \dots, N, \quad (43)$$

$$z_j \geq 0 \quad j = 1, \dots, J_1, \quad (44)$$

$$\sum_{i=1}^N ((b_i - y_{ji})x_i) \leq \gamma \quad j = 1, \dots, J_1, \quad (45)$$

$$\sum_{i=1}^N ((b_i - y_{ji})x_i) \geq \gamma \quad j = J_1 + 1, \dots, J - 1, J, \quad (46)$$

where $\nu_1 = ((1 - \alpha_1)J)^{-1}$, and J_1 is the integer part of $((\alpha_0/\alpha_1))J$. The variable γ is a free variable that separates the active scenarios from the inactive ones.

After solving (39)-(46) we need to check how many scenarios are active, i.e., how many scenarios are in the set H_i . If the number of active scenarios equals one, then we stop the algorithm. If the number of active scenarios exceed one, we split the active scenarios exceeding VaR into two groups and make the scenarios in the upper group inactive. The α for the k -th iteration is defined as

$$\alpha_k = \frac{\alpha_0}{\alpha_0 + (1 - \alpha_0)(1 - \xi)^i}, \quad (47)$$

and hence the integer J_k is defined as

$$J_k = \left[\frac{\alpha_0}{\alpha_k} J \right], \quad (48)$$

and

$$\nu_k = ((1 - \alpha_k)J)^{-1}. \quad (49)$$

The optimization problem solved in the k -th iteration is

$$\min \quad \zeta + \nu_k \sum_{j=1}^{J_k} z_j \quad (50)$$

subject to

$$z_j \geq \sum_{i=1}^N ((b_i - y_{ji})x_i) - \zeta \quad j = 1, \dots, J_k, \quad (51)$$

$$\sum_{i=1}^N b_i x_i = \sum_{i=1}^N b_i, \quad (52)$$

$$l_i \leq x_i \leq u_i \quad i = 1, \dots, N, \quad (53)$$

$$c_i x_i \leq 0.20 \sum_{j=1}^N c_j \quad i = 1, \dots, N, \quad (54)$$

$$z_j \geq 0 \quad j = 1, \dots, J_k, \quad (55)$$

$$\sum_{i=1}^N ((b_i - y_{ji})x_i) \leq \gamma \quad j = 1, \dots, J_k, \quad (56)$$

$$\sum_{i=1}^N ((b_i - y_{ji})x_i) \geq \gamma \quad j = J_k + 1, \dots, J - 1, J. \quad (57)$$

Algorithm A2. Algorithm A2 has a lot in common with Algorithm A1. However, in Algorithm A1 we define α in such a way that we optimize over the active scenarios that have larger losses than VaR, equation (47). In Algorithm A2, we instead define α in a way so that CVaR coincides with VaR (or almost coincides). Then, since CVaR is close to VaR, minimizing CVaR, results in an efficient reduction of VaR for the portfolio. In all other aspects, including the discarding of the upper part of the tail, the two algorithms are identical.

4. Analysis

In this section, we briefly present the results from algorithms that previously have been applied to the same test portfolio as we have used in this paper. We also present our results and then we discuss and compare them with the previous results.

4.1. Portfolio of Bonds and Previous Results

A short description of the portfolio of bonds considered in this study is included in Appendix. The values of VaR and CVaR, for this portfolio, are given in Table 8.1. All reported VaR and CVaR values correspond to an expected portfolio return of 7.26%.

Table 8.1. VaR and CVaR for the one-year loss distribution of the portfolio of bonds (MUSD).

α	VaR	CVaR
0.95	518	824
0.99	1026	1320

Previous work by Mausser and Rosen [26], Andersson, Mausser, Rosen, and Uryasev [3], for the same portfolio has generated the results presented in Tables 8.2 and 8.3, accordingly.

One should take into account that a different initial portfolio has been used for these algorithms. In the case of the heuristic algorithm in [26], the initial

Table 8.2. 95%-VaR and 95%-CVaR for the one-year loss distribution (MUSD).

Method	VaR(95%)	CVaR(95%)
Mausser/Rosen	121	204
Andersson et al.	–	–

Table 8.3. 99%-VaR and 99%-CVaR for the one-year loss distribution (MUSD).

Method	VaR(99%)	CVaR(99%)
Mausser/Rosen	179	296
Andersson et al.	210	263

portfolio⁴ is taken from the so called efficient frontier. This frontier is derived by minimizing expected regret for different values of threshold. For each threshold, an optimal portfolio and corresponding α -VaR are calculated. Then, from these optimal portfolios, the portfolio with minimum α -VaR is selected. For the specific case of a return of 7.26%, this portfolio has a VaR of 184 MUSD which is lower than the VaR that we started with. This portfolio was then used as an initial point for the Mausser and Rosen [26] heuristics (one linear programming problem). To compare different algorithms (the heuristic procedure in Mausser and Rosen [26] and the heuristic procedure in this paper), one should start with the same initial portfolio.

4.2. Calculation Results

In this section, we present the calculation results for Algorithms A1 and A2 when applied to the bond portfolio. First, we have optimized the portfolio using a ξ of 0.5 in both algorithms, i.e., half of the active scenarios was made inactive in each iteration. Calculation results for these runs are presented in Tables 8.4 and 8.5, respectively.

Figures 8.4 and 8.5 show VaR after each iteration of Algorithms A1 and A2 with $\alpha = 0.95$, $\xi = 0.5$ and $\alpha = 0.99$, $\xi = 0.5$, respectively. The first iteration gives the solution of the α -CVaR minimization problem. This step of the algorithms was implemented in [3]. VaR for the original portfolio is not included in the graph (to avoid problems with scaling).

As seen in Figures 8.4 and 8.5, Algorithm A2 outperforms Algorithm A1, i.e., it results in a lower VaR for both values of the parameter α .

⁴although [26] calculates VaR as the loss in the next (higher) scenario, it has no impact on the results since losses were identical in both scenarios.

Table 8.4. VaR and CVaR (MUSD), number of linear programming problems solved, approximate calculation time (Sun Ultra-1 workstation) for the portfolio of bonds optimized using Algorithm A1, $\xi = 0.5$.

α	VaR	CVaR	# of solved LP problems	calculation time (min)
0.95	143	186	10	55
0.99	194	272	8	16

Table 8.5. VaR and CVaR (MUSD), number of linear programming problems solved, approximate calculation time (Sun Ultra-1 workstation) for the portfolio of bonds optimized using Algorithm A2, $\xi = 0.5$.

α	VaR	CVaR	# of solved LP problems	calculation time (min)
0.95	142	187	10	55
0.99	190	273	8	16

Also, we have made runs with Algorithms A1 and A2 with the parameter ξ other than 0.5. In particular, we ran Algorithm A2 with $\alpha = 0.99$, and ξ of 0.1, 0.25, 0.75 and 1. The results from these calculations are summarized in Table 8.6.

Table 8.6. Parameter ξ , VaR and CVaR (MUSD), number of linear programming problems solved, approximate calculation time (Sun Ultra-1 workstation) for the portfolio of bonds optimized using Algorithm A2.

ξ	VaR	CVaR	# of solved LP problems	calculation time (min)
1	207	265	1	2
0.75	197	268	4	8
0.5	190	273	8	16
0.25	185	281	19	37
0.1	176	306	52	102

For instance, as expected, we got a lower VaR (176 MUSD) with a ξ of 0.1, compared to a VaR of 190 MUSD for the case with a ξ of 0.5. However, this leads to a higher CVaR (306 MUSD and 273 MUSD respectively).

As is obvious from Figures 8.6 and 8.7, Algorithm A2 lowered VaR at the expense of an increase in CVaR. Actually, this is an undesirable feature of VaR optimization. The purpose of VaR minimization is to reduce extreme losses.

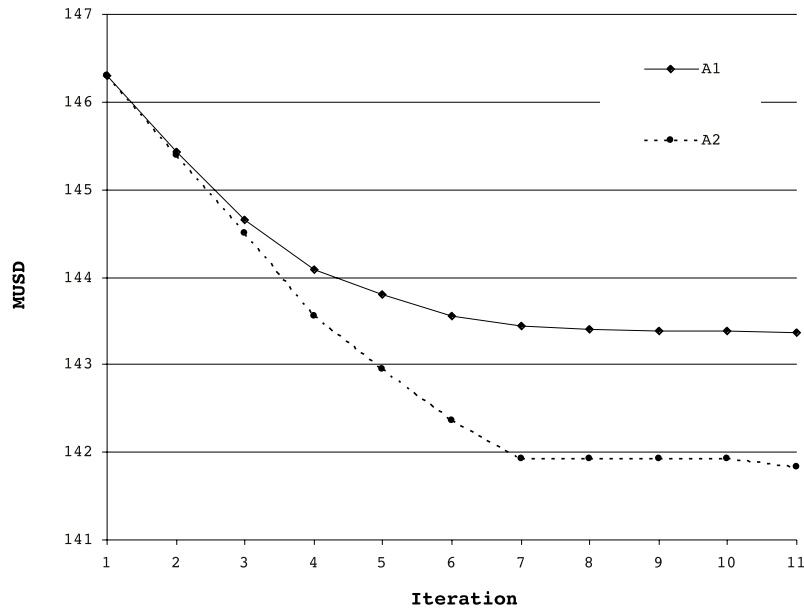


Figure 8.4. VaR at each iteration for Algorithms A1 and A2, with parameters $\alpha = 0.95$, $\xi = 0.5$.

However, VaR minimization leads to an increase in the losses exceeding VaR, i.e., an increase in the extreme losses which we try to control. Therefore, minimization of VaR, starting from the portfolio optimal from a CVaR point of view, actually leads to an undesirable increase in the portfolio risks. CVaR controls extreme $(1-\alpha)\%$ losses, while VaR controls the highest loss among the *lowest* $\alpha\%$ losses. These numerical results are consistent with theoretical results showing that CVaR is a coherent risk measure while VaR is not a coherent measure, see [27, 36, 5].

Our numerical experiments indicate that for low values of the parameter ξ , Algorithm A2 outperforms the heuristic procedure used in [26]. With Algorithm A2, when $\alpha = 0.99$ and $\xi = 0.1$, we obtained VaR = 176 MUSD. This should be compared to a VaR = 179 MUSD obtained in [26]. Nevertheless, judging from the numbers presented in Tables 8.2 - 8.5, it looks like the one step heuristic algorithm suggested in [26] outperforms Algorithms A1 and A2 with ξ equal to 0.5 and other parameter values. However, this is not really the case. In [26], first an efficient frontier has been generated, and then the one linear programming problem has been solved. The minimum VaR portfolio on the efficient frontier has a much lower VaR than the VaR obtained with the minimum CVaR approach.

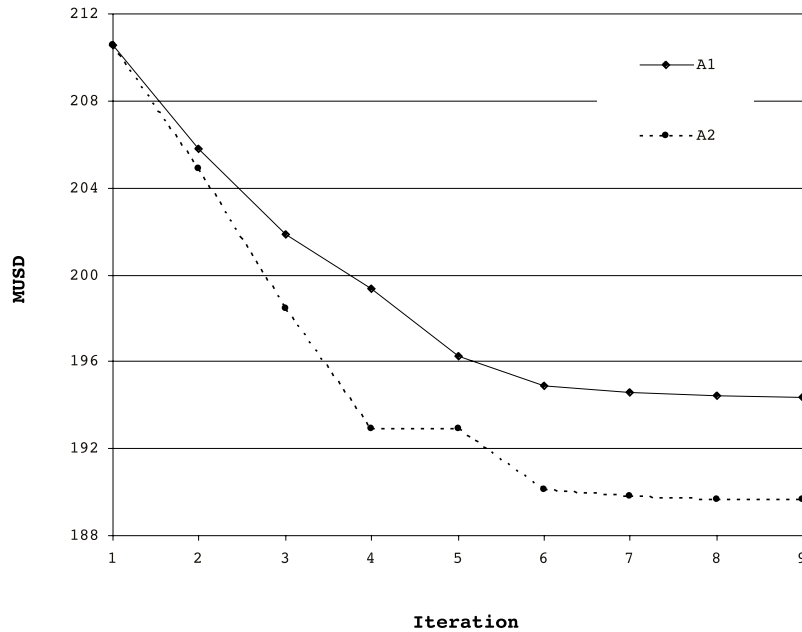


Figure 8.5. VaR at each iteration for Algorithms A1 and A2, with parameters $\alpha = 0.99$, $\xi = 0.5$.

To compare the one step algorithm in [26] with Algorithms A1 and A2, we made numerical experiments starting with the same minimum CVaR portfolio. As we explained earlier, the one step algorithm in [26] corresponds to the special case of the Algorithm A1 and A2 with $\xi = 1$. The results, when we started with the same portfolio, are presented in Tables 8.7 and 8.8. They show that Algorithms A1 and A2 outperform the heuristic method described in [26].

Table 8.7. Optimization results for the one step method (one LP) and Algorithms A1 and A2, when all algorithms start from the same Minimum CVaR Portfolio, $\xi = 0.5$, $\alpha=0.95$, (MUSD).

Method	VaR(95%)	CVaR(95%)
One step	146	185
Algorithm A1	143	186
Algorithm A2	142	187

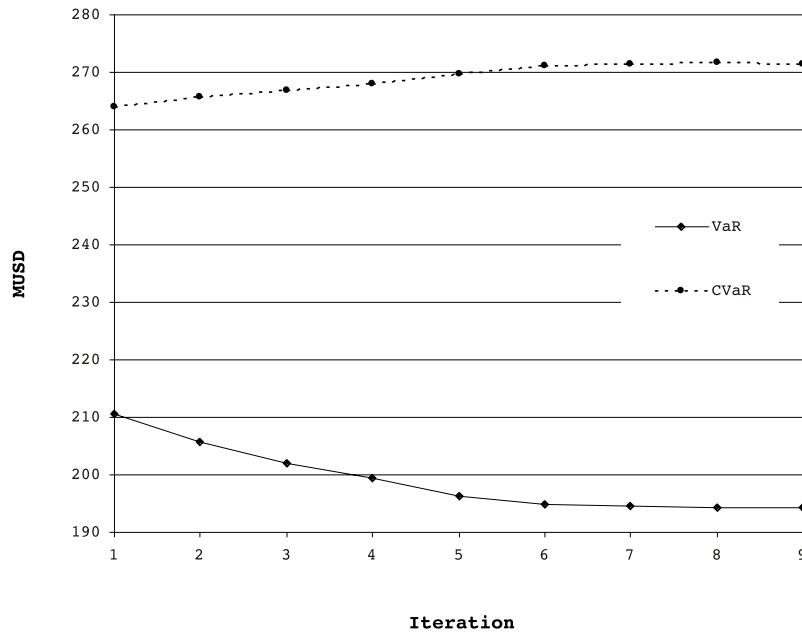


Figure 8.6. Algorithm A2. VaR and CVaR for $\alpha = 0.99$, $\xi = 0.5$.

Table 8.8. Optimization results for the one step method (one LP) and Algorithms A1 and A2, when all algorithms start from the same Minimum CVaR Portfolio, $\xi = 0.5$, $\alpha=0.99$, (MUSD).

Method	VaR(99%)	CVaR(99%)
One step	207	265
Algorithm A1	194	272
Algorithm A2	190	273

5. Conclusion

We have suggested and numerically tested two new algorithms for the optimization of VaR. Both algorithms use the efficient CVaR minimization procedure in each iteration. Numerical tests were conducted on the credit risk model for a portfolio of emerging market bonds. Both Algorithms A1 and A2 showed good performance when compared to other numerical algorithms which were tested with the same portfolio. Furthermore, Algorithm A2 demonstrated a better performance than Algorithm A1. The numerical experiments also demonstrated that reducing the parameter ξ leads to an improvement in the quality of the solution (from VaR point of view). On the other hand, it

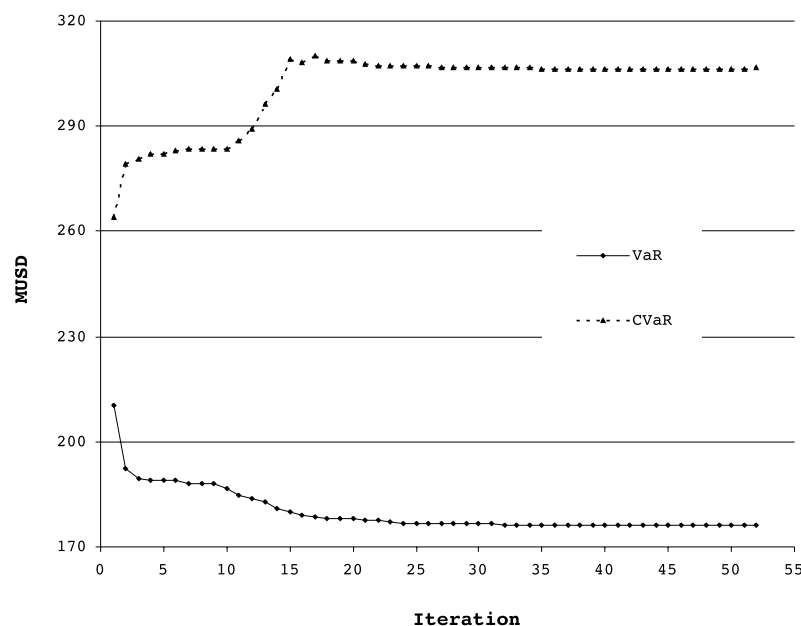


Figure 8.7. Algorithm A2. VaR and CVaR for $\alpha = 0.99$, $\xi = 0.1$.

also leads to a larger number of iterations. Also, the numerical experiments revealed some undesirable characteristics of VaR minimization. The purpose of VaR minimization is to reduce extreme losses. However, VaR minimization leads to an increase of the losses exceeding VaR, i.e., an increase in the extreme losses which we try to control. Therefore, minimization of VaR, starting from the portfolio that is optimal from a CVaR point of view, actually leads to an undesirable increase in the portfolio risks. The numerical results obtained in this paper are consistent with theoretical results showing that CVaR is a coherent risk measure while VaR is not a coherent measure, see [27, 36, 5].

Acknowledgments. We would like to acknowledge fruitful discussions related to implementation of the algorithms with Dr. Pavlo Krokmal at the University of Florida. Also, we want to thank Prof. Ulf Brännlund at the Royal Institute of Technology (KTH) for his help with conducting numerical experiments.

6. Appendix: Bond Portfolio

This appendix gives a short overview of the Bond Portfolio, following the papers [8, 24]. Also, we briefly describe the CreditMetrics methodology [9], which was used to construct the credit risk model for the portfolio.

6.1. Portfolio Description

The test portfolio used in this paper has been compiled by a group of financial institutions to assess the state-of-the-art of portfolio credit risk models. It consists of 197 bonds in emerging markets, issued by 86 obligors in 29 countries. The mark-to-market value of the portfolio is 8.8 billion US dollars (USD). Most of the instruments are denominated in USD, but 11 fixed rate bonds are denominated in seven other currencies; DEM, GDP, ITL, JPY, TRL, XEU, and ZAR. Bond maturities range from a few months to 98 years and the portfolio duration is approximately five years. The date of the analysis is October 13, 1998.

6.2. CreditMetrics

CreditMetrics is a tool for assessing portfolio risks due to defaults and changes in the obligors' credit quality, such as upgrades and downgrades in credit ratings. In this case study, we consider eight credit rating categories, including default, and the modeling time horizon is one year.

The first step in computing portfolio losses due to credit events involves calculating exposures, or the forward mark-to-market values for each obligor under each possible credit state. The exposures are calculated using the forward rates implied by today's term structures in each of the seven non-default states. In the case of default, the value is based on an appropriate recovery rate. The probabilities of changing credit rating are obtained from a credit transition matrix, namely the July 1998 matrix provided by Standard & Poor's in this case. Once the possible exposures have been tabulated, Monte Carlo simulation can be used to obtain scenarios of credit losses.

A Monte Carlo simulation samples a large number of scenarios on the joint credit states of each obligor at the horizon. Joint default and migration correlations are driven by the correlations of the asset values of the obligors. Since the asset values are not observable, equity correlations of traded firms are used as a proxy for the asset correlations. More specifically, CreditMetrics maps each obligor to a country, region, or sector indices that is more likely to affect its performance, and to a risk component that captures the firm-specific volatility.

In each scenario, the portfolio mark-to-market value is obtained by summing up the exposure corresponding to each scenario and obligor. The credit loss distribution is then simply calculated by subtracting the portfolio mark-to-market values in each credit stage from the forward value of the portfolio if no credit migration occurs.

Figure 8.8 illustrates the original portfolio loss distribution. It is skewed and has a long fat right tail. Some statistics from the one-year credit loss distribution are presented in Tables 8.9 and 8.10.

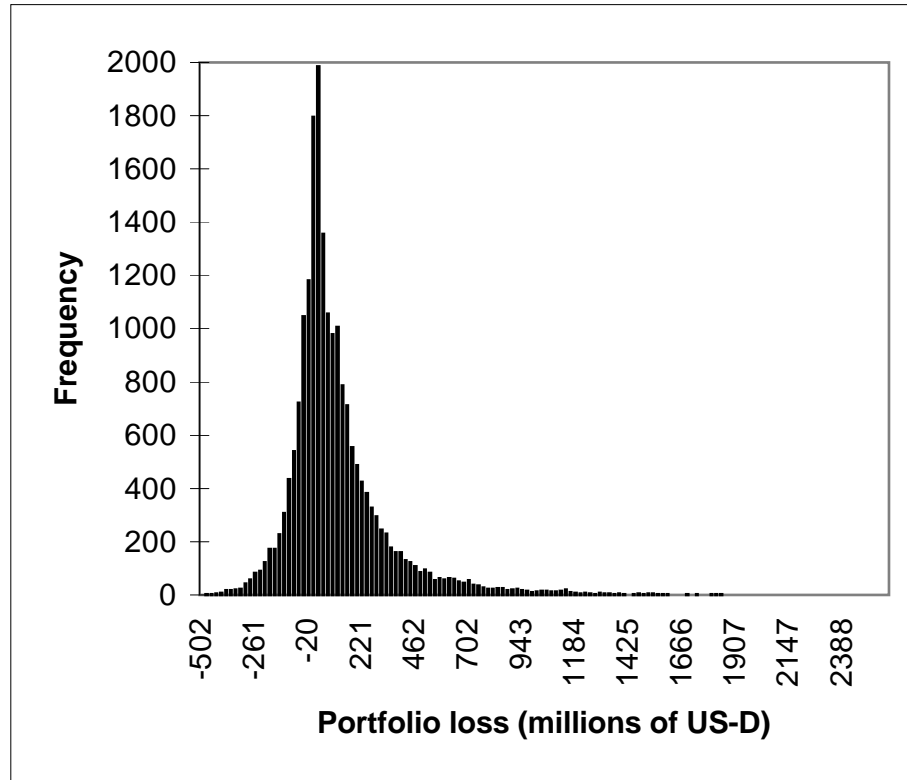


Figure 8.8. One year credit loss distribution (millions of USD).

Table 8.9. Mean and standard deviation for one-year loss distribution (MUSD).

Expected loss	95
Standard deviation	232

Table 8.10. VaR and CVaR for the one-year loss distribution (MUSD).

α	VaR	CVaR
0.90	341	621
0.95	518	824
0.99	1026	1320
0.999	1782	1998

References

- [1] Acerbi, C., Nordio, C. and C. Sirtori, *Expected shortfall as a tool for financial risk management*. Working paper, (2001), (Can be downloaded from <http://www.gloriamundi.org>).
- [2] Acerbi, C., and D. Tasche, *On the coherence of expected shortfall*. Working paper, (2001), (Can be downloaded from <http://www.gloriamundi.org>).
- [3] Andersson, F., Mausser, H., Rosen, D. and S. Uryasev (2001): Credit Risk Optimization with Conditional Value-at-Risk Criterion. *Mathematical Programming*, Series B, December, 2000, relevant Research Report 99-9 can be downloaded from www.ise.ufl.edu/uryasev/pubs.html#t.
- [4] Andersen, J.V., and D. Sornette (1999): *Have Your Cake and Eat It Too: Increasing Returns While Lowering Large Risks*. Working Paper, University of Los Angeles, can be downloaded from <http://www.gloriamundi.org>.
- [5] Artzner, P., Delbaen F., Eber, J.M., and D. Heath (1999) Coherent Measures of Risk. *Mathematical Finance*, 9, 203-228.
- [6] Basak, S. and A. Shapiro (1998): *Value-at-Risk Based Management: Optimal Policies and Asset Prices*. Working Paper, Wharton School, University of Pennsylvania, can be downloaded from <http://www.gloriamundi.org>.
- [7] Birge J.R. and F. Louveaux (1997): *Introduction to Stochastic Programming*. Springer, New York.
- [8] Bucay, N. and D. Rosen (1999): Credit Risk of an International Bond Portfolio: a Case Study. *ALGO Research Quarterly*. Vol.2, No. 1, 9–29.
- [9] *CreditMetrics: The Benchmark for Understanding Credit Risk* (1997): Technical Document, J.P.Morgan Inc., New York, NY.
- [10] *CreditRisk+: A Credit Risk Management Framework* (1997): Credit Suisse Financial Products, New York, NY.

- [11] Duffie, D. and J. Pan (1997): An Overview of Value-at-Risk. *Journal of Derivatives*. 4, 7–49.
- [12] Emmer S., Kluppelberg, C., and R. Korn. (2000): Optimal Portfolios with Bounded Capital at Risks. Working paper, Munich University of Technology, June.
- [13] Ermoliev, Yu. and R. J-B Wets (Eds.) (1988): *Numerical Techniques for Stochastic Optimization*, Springer Series in Computational Mathematics, 10.
- [14] Jorion, Ph. (1996): *Value at Risk : A New Benchmark for Measuring Derivatives Risk*. Irwin Professional Pub.
- [15] Gaivoronski, A.A. and G. Pflug (2000): *Value at Risk in portfolio optimization: properties and computational approach*, NTNU, Department of Industrial Economics and Technology Management, Working paper, July.
- [16] Gouriéroux, C., Laurent, J.P. and O. Scaillet (2000): *Sensitivity Analysis of Values-at-Risk.*, Working paper, Université de Louvain, January, can be downloaded from <http://www.gloriamundi.org>.
- [17] Grootweld H. and W.G. Hallerbach (2000): *Upgrading VaR from Diagnostic Metric to Decision Variable: A Wise Thing to Do?*, Report 2003 Erasmus Center for Financial Research, June.
- [18] Kall, P. and S.W. Wallace (1994) *Stochastic Programming*. Wiley, Chichester.
- [19] Kan, Y.S., and Kibzun, A.I. (1996). *Stochastic Programming Problems with Probability and Quantile Functions*, John Wiley & Sons, 316.
- [20] Kast, R., Luciano, E., and L. Peccati (1998): *VaR and Optimization*. 2nd International Workshop on Preferences and Decisions, Trento, July 1-3 1998.
- [21] Kealhofer, S. (1997): *Portfolio Management of Default Risk*, KMV Corporation, Document # 999-0000-033, Revision 2.1.
- [22] Lucas, A., and Klaassen, P. (1998): Extreme Returns, Downside Risk, and Optimal Asset Allocation. *Journal of Portfolio Management*, Vol. 25, No. 1, 71-79.
- [23] Markowitz, H.M. (1952): Portfolio Selection. *Journal of finance*. Vol.7, 1, 77–91.
- [24] Mausser, H. and D. Rosen (1998): Beyond VaR: From Measuring Risk to Managing Risk, *Algo Research Quarterly*, Vol. 1, No. 2, 5–20.

REFERENCES

157

- [25] Mausser, H. and D. Rosen (1999): Applying Scenario Optimization to Portfolio Credit Risk, *Algo Research Quarterly*, Vol. 2, No. 2, 19–33.
- [26] Mausser, H and D. Rosen (1999): Efficient Risk/Return Frontiers for Credit Risk, *Algo Research Quarterly*, Vol. 2, No. 4, 35–48.
- [27] Pflug, G.Ch. (2000): *Some Remarks on the Value-at-Risk and the Conditional Value-at-Risk*. In "Probabilistic Constrained Optimization: Methodology and Applications", Ed. S. Uryasev, Kluwer Academic Publishers, 2000.
- [28] Pritsker, M. (1997): Evaluating Value at Risk Methodologies, *Journal of Financial Services Research*, 12:2/3, 201-242.
- [29] Prekopa, A. (1995) *Stochastic Programming*, Kluwer Academic Publishers.
- [30] Puelz, A. (1999): *Value-at-Risk Based Portfolio Optimization*. Working paper, Southern Methodist University, November.
- [31] RiskMetricsTM. (1996): *Technical Document*, 4-th Edition, New York, NY, J.P.Morgan Inc., December.
- [32] Tasche, D. (1999): *Risk contributions and performance measurement*. Working paper, Munich University of Technology (July).
- [33] Wilson, T. (1997a): Portfolio Credit Risk I, *Risk*, No. 10(9), 111–117.
- [34] Wilson, T. (1997b): Portfolio Credit Risk II, *Risk*, No. 10(10), 56–61.
- [35] Rockafellar, R.T. and S.Uryasev. (2000): Optimization of Conditional Value-At-Risk. *The Journal of Risk*, Vol. 2, No. 3, 2000, 21-41.
- [36] Rockafellar R.T. and S. Uryasev. (2001): *Conditional Value-at-Risk for General Loss Distributions*. Research Report 2001-5. ISE Dept., University of Florida, April (2001) (Can be downloaded: www.ise.ufl.edu/uryasev/cvar2.pdf)
- [37] Ziemba, T.W. and M.J. Mulvey Eds. (1998): *Worldwide Asset and Liability Modeling*, Cambridge Press, Publications of the Newton Institute.
- [38] Yamai, Y. and T. Yoshihara. (2001): *On the Validity of Value-at-Risk: Comparative Analyses with Expected Shortfall*. Institute for Monetary and Economic Studies. Bank of Japan. IMES Discussion Paper 2001-E-4. (Can be downloaded: www.imes.boj.or.jp/english/publication/edps/fedps2001_index.html)