Computational Methods in Finance Related to Distributions with Known Marginals

by

Amir Memartoluie

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2017

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner: Hao Yu, Professor

Supervisors: Tony Wirjanto, Professor
David Saunders, Associate professor

Internal Member: Yuying Li, Professor
Justin Wan, Associate professor

Internal-external Member: Pierre Chaussé, Assistant professor

Other Members: Scott Leatherdale, Associate professor

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Model uncertainty and the dependence structures of various risk factors are important components of measuring and managing financial risk, such as market, credit and operational risks. In this thesis we provide a systematic investigation into these issues by studying their impacts on Credit Value Adjustment (CVA), Counterparty Credit Risk (CCR), and estimating Value-at-Risk for a portfolio of financial instruments. In particular we address the numerical issues of finding an unknown (worst-case) copula that ties marginal distributions of risk factors together given partial information about them.

Acknowledgements

I would like to express my gratitude to my supervisors, Professor Tony Wirjanto and Professor David Saunders for their support and mentorship in both my masters and PhD studies. Their invaluable guidance assisted my research. I would also like to thank Professor Marius Hofert who has made important contributions to chapter 5 and 6.

I am grateful to my committee members, Professor Yuying Li and Professor Justin Wan, for their guidance, suggestions and support during my graduate studies. Their constructive comments and questions have been invaluable in improving both the content and the presentation of this thesis and I am very fortunate to have them on my committee.

I would like to thank my external examiners, Professor Pierre Chaussé from the Department of Economics at University of Waterloo and Professor Hao Yu from the Department of Statistical and Actuarial Sciences at Western University for being internal-external examiner and external-external examiner in my Ph.D. thesis defense.

I also benefited greatly from academic discussions with my fellow schoolmates at SciCom lab. Especially Stephen Tse, Kai Ma, Ad Tayal, Parsiad Azimzadeh, Edward Cheung, Ryan Goldade, Colleen Kinross, Swathi Amarala who have become good friends.

I would like to thank Sig Walter, my Tai Chi Sifu without whose unwavering support this journey would not have been possible. Many thanks to my uncle, Mojtaba Zarnegar, that never stopped believing in me and motivated and supported me from the beginning of this journey. I am grateful to Malcolm Rutherford for his support and encouragement.

I would like to acknowledge Professor Ken Seng Tan and the Waterloo Research Institute in Insurance, Securities and Quantitative Finance for providing support in the form of the scholarships and research assistantships. This support has made it possible for me to pursue my graduate studies and conduct this research on a full-time basis.

Finally, I would like to thank my parents, Soudabeh Farhang and Abbas Memartoluie, who have always supported and encouraged me to pursue my goals. Last but not least, I would like to thank my wife, Sara, for her love and encouragement.

Dedication

I dedicate this thesis to my parents.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Model Uncertainty and Risk Management

In general there are at least three primary sources of risk for a given portfolio of financial instruments: market risks, credit risks and operational risks. Market risk is defined as the losses accrued to the portfolio due to the changes in its underlying financial instruments and parameters (such as interest rates, exchange rates, equity prices and commodity prices as well as economic indicators such as gross domestic product (GDP) and consumer price index (CPI)). Credit risk takes into account the risk of not receiving a payment from a borrower or counterparty due to a default. Finally, operational risk represents the risk that is attributed to those losses due to insufficient internal processes, ineffective risk mitigation measures or exogenous events. In this thesis we will focus only on market risk factors and credit risk factors.

Credit value adjustment (CVA) of a portfolio of derivatives is the difference between the default-free portfolio value and the portfolio value, when a counterparty is subject to default. Calculating CVA requires non-trivial simulations. Identifying the underlying risk factors, conducting simulations of prices using a specific model and combining this information at the portfolio level are some of the main steps in calculating CVA. Chapter 2 reviews the common practices underlying such calculations and proposes a new framework for calculating bounds on CVA contributions with given marginals.

Counterparty Credit Risk (CCR), defined as the risk of default before the final settlement of a transaction's cash flows, played an important role in the financial crisis of 2008. Modelling stochastic exposures and the number of relevant risk factors is a challenging problem for quantitative modelers. In chapters 3 and 4 we provide an overview of this problem and propose the *worst-case copula methodology* to calculate an upper bound on the *Conditional Value-at-Risk* (CVaR) of a portfolio by optimizing over all joint distributions of the market risk factors and credit risk factors with given marginals.

Value-at-Risk (VaR) is a commonly used tool in Quantitative Risk Management (QRM) for measuring the potential loss of a portfolio of risky assets over a defined period and at a chosen confidence level α. To this end analyzing a one-period ahead vector of losses

$\mathbf{L} = (L_1, \ldots, L_d)^\top$ is required. After providing an overview of the existing literature on estimating the upper bound on VaR, namely $\overline{\text{VaR}}_\alpha(L_1 + \ldots + L_d)$, given prescribed marginals of $(L_1, \ldots, L_d)$, we propose several improvements over the existing algorithm, known as the Rearrangement Algorithm (RA), and discuss the performance of the resulting *Adaptive Rearrangement Algorithm* (ARA) for this problem under different test cases.

In summary, the common thread in the above three problems is that we only know the marginal distributions of risk factors in each case and the (worst-case) copula that binds them together is unknown to us.

1.2 Overview and Contributions

The remainder of the thesis is structured as follows.

1.2.1 Bounds on CVA Contributions with Given Marginals

In chapter 2 we review numerical schemes for computing bounds on CVA using linear programming techniques, originally proposed by Glasserman and Yang [1] by means of a set of numerical examples for both unilateral and bilateral CVAs in sections 2.6.2 and 2.6.4.

We further present the main principles underlying the calculation of CVA and review an existing framework, known as the *Ordered Scenario Copula* (OSC) that is used in practice for such calculations.

Then we present a numerical example on a portfolio comparing the method of the ordered scenario copula presented in Rosen and Saunders [2] to that of Glasserman and Yang [1].

The research contribution of this chapter is in section 2.7 in which we extend the approach of Glasserman and Yang [1] to derive bounds on CVA *contributions in the presence of netting agreements as defined by Pykhtin and Rosen [3].*

1.2.2 Counterparty Credit Risk and Bounds on Conditional Value-at-Risk

The contribution of Chapter 3 lies in the formulation of the problem of computing bounds on CVaR when the underlying risk factors have given marginals as a linear optimization problem, and in the case of a discrete probability space as a linear programming problem. After formalizing this problem in section 3.5 and proposing the *worst-case copula methodology*, we present the linear programming problem for calculating the bounds on CVaR of a portfolio in section 3.7.

Then we provide an application of the worst-case copula methodology for the counterparty credit risk. In section 4.1, we formalize this problem under the Basel Accord framework. A real application is provided in section 4.2, in which we use the methodology proposed in section 4.1 to find the joint distribution of market risk factors and credit risk

factors, given their prescribed set of marginal distributions to calculate an upper bound on CVaR for a real-world portfolio of over-the-counter (OTC) derivatives.

The results presented in chapters 3 and 4 have been published in the Journal of Risk Management in Financial Institutions (Memartoluie et al. [4]).

1.2.3 Rearrangement Algorithm and VaR Analytical Bounds

In chapter 5, we review existing methods for computing an upper bound on VaR, $\overline{\mathrm{VaR}}_\alpha(L^+)$, for homogeneous portfolios with $L^+ = L_1 + \ldots + L_d$ where $L_i \sim F_i$ and $F_1 = \cdots = F_d$ including the *dual bound approach* of Embrechts et al. [5], *Wang's approach* presented in Embrechts et al. [6] and the *Rearrangement Algorithm* (RA) of Puccetti and Rüschendorf [7].

The research contribution of this chapter is in examining numerical challenges inherent in the implementation of these algorithms used for computing an upper bound for the worst VaR in the case of homogenous marginals, and presenting some theoretical results relevant to these algorithms. In particular, we consider:

- **Dual bound approach**: We prove properties of the auxiliary functions that are used in calculating $\overline{\mathrm{VaR}}_\alpha(L^+)$. Specifically, Proposition 5.4.2 investigates convexity and monotonicity of the functions $D(s,t)$ and $\tilde{D}(s)$ respectively. We show uniqueness of the minimum for the auxiliary function $\tilde{D}(s)$, and in example 5.4.3 we provide numerical examples of the challenges presented by this function for the Generalized Pareto distribution.
- **Wang's approach**: Our contribution in this section is stated in Propositions 5.4.4 and 5.4.5. Propositions 5.4.4 studies the uniqueness of the root of the auxiliary function that is used in this approach and Proposition 5.4.5 presents the results of computing the appropriate lower and upper bounds for the root-finding procedure for Pareto marginals. We further demonstrate selected properties of the auxiliary function h. More specifically, Proposition 5.4.6 shows continuity and differentiability of the auxiliary function h. In addition to that, in Proposition 5.4.13 we prove the existence of the root for the auxiliary function h. Finally, we compare the dual bound approach and Wang's approach using Pareto marginals to illustrate the above results in examples 5.4.14 and 5.4.15.

In the case of non-homogenous marginals, after describing the RA of Puccetti and Rüschendorf [7] for approximating the worst VaR in section 5.5, we study the performance of this algorithm. More specifically in section 5.6, we present numerical examples to illustrate the inner working of this method. These include:

- Calculating the upper bound on VaR for portfolios with given marginal loss distributions in 4 different cases using Pareto marginals, more specifically, when the marginals are driven by a heavy-tailed distribution, a moderately heavy-tailed distribution, marginals varying from a heavy-tailed distribution to a not so heavy-tailed distribution and a combination of the previous cases.

- Investigating the impact of the input parameters of the RA (the choice of discretization parameter N and absolute error ε).
- Calculating the run time, the VaR bounds, the number of iterations and the number of oppositely ordered columns (as a proxy for the objective function) at convergence.

This is the first study that attempts to assess the effect of different properties of the input distributions (number, heaviness of tail and degree of homogeneity) on the performance of the Rearrangement Algorithm.

1.2.4 Adaptive Rearrangement Algorithm for Non-homogenous Portfolios

Our contribution in chapter 6 is in presenting an *Adaptive Rearrangement Algorithm* (ARA) and an *Enhanced Adaptive Rearrangement Algorithm* (EARA), to address some of the issues identified in chapter 5.

In the ARA:

1. New convergence criteria are used to choose N adaptively.
2. Two stopping criteria are introduced based on a given vector of relative errors, $\varepsilon = (\varepsilon_{\mathbf{1}}, \varepsilon_{\mathbf{2}})$. Iteration continues until both the relative change of the input matrix X used in computing the lower and upper bounds for $\overline{\mathrm{VaR}}_\alpha(L^+)$ as well as the relative difference of these bounds to $\overline{\mathrm{VaR}}_\alpha(L^+)$ are satisfied as specified by $\varepsilon = (\varepsilon_{\mathbf{1}}, \varepsilon_{\mathbf{2}})$.

The performance of the ARA is further investigated in sections 6.3.2, 6.3.3 and 6.3.4 for some of the most commonly used distributions in quantitative risk management, namely generalized Pareto, Student's t-distribution and log-normal for the four portfolios with different marginal tail behaviors, similar to the studies presented in section 5.6.

Finally, we present the EARA. The main advantage of this algorithm is that it converges faster when the number of risk factors d is large for a given input portfolio. In addition, a detailed example of the use of the ARA and EARA using Operational Risk data is also provided in section 6.5.

The results presented in chapters 5 and 6 are forthcoming in Statistics & Risk Modeling (Hofert et al. [8]).

Chapter 2

Worst-case Credit Valuation Adjustment and CVA Contributions

2.1 Overview

Prior to the global financial crisis of 2008, *Credit Valuation Adjustment (CVA)*, which is defined as the difference between the portfolio value calculated by assuming that the counterparty can not default, and the portfolio value calculated when counterparty defaults are taken into account, was overlooked in the derivatives markets, accounting standards and regulatory frameworks. However, today CVA is one of the main tools that financial institutions use to price counterparty risk (a detailed example on how we can calculate CVA for a simple portfolio, consisting of only one instrument, is presented in section A.1 in Appendix A.).

CVA played an important role in the financial crisis. According to the Basel Committee on Banking Supervision (BCBS) [9] "Mark-to-market losses due to CVA were not directly capitalized. Roughly two-thirds of CCR losses [counterparty credit risk losses] were due to CVA losses and only one-third were due to actual defaults". If the crisis has taught us anything, it is that, including the creditworthiness of the respective counterparties and the credit risks faced by them in the pricing process should become an important part of calculating the market price of over-the-counter derivatives.

There are two main methods to measure CVA, namely the *Unilateral* and *Bilateral* approaches. The main assumption in calculating the unilateral CVA is that the institution that performs the CVA analysis (the bank) is default-free. Unilateral CVA pricing produces the current market value of future losses due to the counterparty's potential default. It would be difficult to agree upon a *fair* trade value if both the bank and the counterparty require a premium for the credit risk they are bearing. Bilateral CVA addresses this issue: if we perform our analysis under the assumption that both the counterparty and the bank can default, we can produce an objective fair value for the trade that both of the parties can agree upon.

Regardless of the approach that we take for calculating CVA, its implementation can be intricate for the following reasons:

- **Pricing models:** Given a large portfolio (of possibly up to several hundred different types of instruments) a variety of models is used in the pricing process, some of which can be quite complex.

- **Scenario generation:** From a computational standpoint, and given the wide range of models that are used for pricing various instruments, running Monte Carlo simulations for computing counterparty exposures, defined as the potential positive future value of the portfolio, at the time of default, is generally the most expensive step in computing CVA. For example, in order to perform *one* CVA calculation for a portfolio of 40,000 positions, over 1,000 scenarios and 250 time steps, 10 billion valuations are required.

- **Model uncertainty:** Even if all of the aforementioned problems are resolved satisfactorily, the inherent *model risk* issues surrounding CVA calculation can still subvert the results of the CVA analysis.

Appendix A illustrates various aspects of model uncertainty, misspecification of parameters and the impact of such issues on CVA calculation using a numerical example for a simple portfolio.

2.2 Outline and Contributions

Section 2.3 provides an overview of the existing literature on CVA calculation and shows how it is related to the contributions of the thesis in this chapter.

The basics of CVA calculation under both the unilateral and bilateral settings are presented in section 2.4. The inner workings of one of the existing frameworks, namely the Monte Carlo approach of the *Ordered Scenario Copula (OSC)* methodology, proposed by Rosen and Saunders [2] for computing CVA in the presence of wrong-way risk, are shown in sections 2.5.2 and 2.6.

In sections 2.6.4 and 2.6.2 the *Worst-case CVA methodology* under both the unilateral and bilateral settings is illustrated. The methodology presented in sections 2.6.2 and 2.6.4 is from Glasserman and Yang [1]. In sections 2.6.3, 2.6.5 and 2.6.6, we implement and compare the performance of the worst-case copula methodology to that of the ordered scenario copula of Rosen and Saunders [2] for calculating the unilateral and bilateral CVAs.

Section 2.7 outlines the issues surrounding the calculation of CVA with netting agreements. In particular, we outline the methodology of Pykhtin and Rosen [3] for calculating CVA contributions in the presence of a netting agreement in sections 2.7.1 and 2.7.2.

The main contribution of this chapter is presented in section 2.7.4. By combining the method of Pykhtin and Rosen [3] with that of Glasserman and Yang [1], presented in section 2.6.2, we calculate bounds on CVA contributions at the position level when the counterparty's credit quality and exposures are correlated. A comparison of the performance of this methodology with that of Pykhtin and Rosen [3] is presented at the end of this section.

2.3 Literature Review

One of the earliest works which attempts to classify issues surrounding *model risk* in the valuation of financial assets is found in Derman [10]. In this work the author investigates various assumptions of the underlying models used for valuing securities in finance and the ensuing risk. Analyzing various kinds of models used in finance (fundamental, phenomenological and statistical) and the advantages and disadvantages associated with each of them in practice is an important aspect of this work. Specifically the author investigates issues such as *inapplicability of a model*, *usage of an incorrect model*, *incorrect solution produced by correct model*, *badly approximated solution*, *software and hardware bugs* and *inappropriate usage of correct model* for valuation purposes.

Several authors have investigated the pricing of derivatives under different assumptions and models, taking into account only the default risk of clients. One of the earliest works in this regard is by Cooper and Mello [11] in which the authors introduce the concept of an unilateral counterparty value adjustment of interest rate swaps and derive equilibrium swap rates. More recent works by Brigo and Masetti [12] and Brigo and Pallavicini [13] take an in-depth look at pricing interest rate swaps and equity derivatives and the counterparty credit risk associated with the portfolios of such instruments. In a related work Brigo et al. [14] investigate the valuation of counterparty risk for commodity derivatives.

Rüschendorf [15] performs a comparison of multivariate risk vectors with respect to supermodular or related orderings in order to identify some function classes which allow him to conclude that positive (negative) dependent random vectors are more (less) risky than independent vectors with respect to these functions. The performance of Fréchet bounds when multivariate marginal distributions are given is investigated as well. Rüschendorf [16] adapt some classical tools such as Fréchet bounds for describing the worst case dependence structures for a portfolio of financial assets.

The approach of Haase et al. [17] does not rely on any specific model for the joint evolution of the underlying risk factors when a bilateral counterparty value adjustment is made. In order to cover various types of derivatives (such as interest rate, commodity and credit default swaps) in their valuation process, they incorporate three main components in the counterparty valuation: the first part describes the loss given default process that is assumed to be constant unless random recoveries are made; the second component corresponds to default indicators of the two counterparties; and the last component consists of exposure-at-default of the over-the-counter derivative; the latter part describes the risk-free present value of any outstanding amount if any of the counterparties defaults. The authors show how any coupling of the aforementioned components leads to a feasible adjustment. After that a series of linear optimization problems is presented. The solutions to these problems provide tight bounds on the adjustments. Despite the simplicity of these methods, it is not clear how close the proposed bounds are to the worst possible counterparty value adjustment.

Compared to other contracts, an analysis of the counterparty credit risk for credit default swaps (CDS) has proven to be far more complex. An accurate valuation of CDS contracts requires that we consider joint defaults of both counterparties and the

reference entity. In addition, in order to account for the fluctuation of the market value of CDS contracts, one has to consider stochastic credit spreads. Brigo et al. [14], Brigo and Chourdakis [18] and Brigo and Pallavicini [13] utilize dynamic stochastic models to investigate this issue. Leung and Kwok [19] model default intensities as deterministic constants. Lipton and Sepp [20] take a different approach and utilize a multi-dimensional jump-diffusion version of a structural default model and use it to compute the credit value adjustment for a credit default swap. Crépey et al. [21] study the counterparty risk on a payer CDS using a Markov chain model of two reference credits, one representing the firm underlying the CDS and the other the protection seller in the CDS. Hull and White [22] use correlated models for valuing CDS contracts, allowing the payoff to be contingent on defaults by multiple reference entities. Finally Walker [23] utilizes a continuous-time Markov approach for pricing CDS.

One of the earliest works that incorporates a bilateral default risk in pricing derivatives is by Sorensen and Bollier [24] on interest rate swaps contracts. The authors develop a model of swap default risk that evaluates a joint probability of the swap counterparty defaulting and the cost of the default for the solvent party. The authors further include a bilateral default risk into this framework and evaluate a replacement cost that affects both parties in case of default. In addition to incorporating different default risks, Duffie and Huang [25] investigate the impact of credit risk asymmetry and netting rules on pricing interest rate derivatives and currency swaps. Brigo and Capponi [26] model default dependence by introducing stochastic intensity and a trivariate copula function for default times. The authors further analyze a portfolio of CDS to illustrate the application of this methodology. Brigo et al. [27] incorporate the correlation between the default times of the investor and the counterparty, as well as the correlation of each of these with the underlying risk factors to analyze a portfolio of interest rate derivatives.

A more detailed treatment of fundamental concepts of counterparty credit risk and credit value adjustment can be found in Gregory [28], Cesari et al. [29] and Brigo et al. [30].

Glasserman and Xu [31] take a more mathematical approach to the issue of model uncertainty in financial risk management. In addition to investigating the impact of imperfect assumptions and parameter estimates in creating model risk, they develop a framework for assessing how such model errors can be quantified. Glasserman and Yang [1] propose a method for bounding wrong-way risk (when there exists an unfavorable correlation between the value of the underlying portfolio and the default of the counterparty) when the marginal distributions of market and credit risk factors are known (this method is reviewed in section 2.5). Similarly, Beiglböck et al. [32] use infinite-dimensional linear programming techniques to calculate model-independent bounds for exotic options.

Calculating position-level CVA contributions to the counterparty CVA is a fairly new problem and has been studied in Pykhtin and Rosen [3]. The authors reduce the problem of calculating position-level CVA contributions to the problem of calculating contributions of individual trades to the counterparty-level expected exposure.

In section 2.7 we apply the approach of Glasserman and Yang [1] to the problem of calculating bounds on the position-level contributions to CVA. A brief summary of the

notation that we use in chapter 2 is shown below:

CVA^U :	the unilateral CVA
CVA^B :	the bilateral CVA
CVA_i^U :	CVA contribution of the i-th trade in the portfolio of the bank with the counterparty C
τ_{C} :	the default time of the counterparty C
R_{C} :	the recovery rate of the counterparty C
τ_{B} :	the default time of the bank B
R_{B} :	the recovery rate of the bank B

2.4 CVA Basics

In practice, there are two main methods for calculating Credit Value Adjustment: *Unilateral* and *Bilateral*. In the following section, we will define these concepts in detail. Although *netting* and *collateral agreements* are an important part of CVA calculation in practice, we are more interested in the inner-dynamics of CVA calculation and we refer the reader to Pallavicini et al. [33], Morini and Prampolini [34] and Burgard and Kjaer [35] for more details on these issues.

Consider a portfolio of K instruments that the bank has with a given counterparty C. Let $v_i(t)$ denote the value of the i-th instrument in the portfolio at time t from the bank's perspective. When there is no netting agrement, the counterparty-level exposure, $E(t)$, for counterparty C is defined as:

$$E(t) = \sum_{i=1}^{K} \max\{v_i(t), 0\} \tag{2.4.1}$$

Alternatively, if there is a single netting agreement in place, the (netted) exposure is defined as:

$$E(t) = \max\{V(t), 0\} \tag{2.4.2}$$

where

$$V(t) = \sum_{i=1}^{N} v_i(t) \tag{2.4.3}$$

For the sake of simplicity, we assume a single netting agreement to be in place. To this end let $V^+(t)$ represent the bank's exposure to counterparty C at the default time t, i.e.,

$$V^+(t) = \max\{V(t), 0\} \tag{2.4.4}$$

Then the bank's loss when counterparty C defaults at the time of default of the counterparty, τ_{C}, discounted to today is written as:

$$L = 1_{\{\tau_{\text{C}} \leqslant T\}}(1 - R_{\text{C}})D(\tau_{\text{C}})V^+(\tau_{\text{C}}) \tag{2.4.5}$$

where R_{C} represents a *recovery rate*, which is a fraction of the exposure that the bank recovers if the counterparty defaults, T is the maturity of the longest contract in the portfolio, $D(\tau_{\mathrm{C}})$ is the discount factor at τ_{C} and $1_{\{\tau_{\mathrm{C}} \leqslant T\}}$ is the default indicator function, defined as:

$$1_{\{\tau_{\mathrm{C}} \leqslant T\}} = \begin{cases} 1 & \text{if counterpary C defaults} \\ 0 & \text{otherwise} \end{cases}$$

Note that in calculating L in (2.4.5) we assume that the bank *has not* defaulted prior to the counterparty's default. Similarly, if we are interested in the counterparty's exposure to the bank's default, we should calculate:

$$V^{-}(t) = \max\{-V(t), 0\} \tag{2.4.6}$$

where $V^{-}(t)$ represents a positive value of the portfolio from the counterparty's perspective.

The *counterparty exposure* $V^{+}(t)$ represents the economic loss due to the counterparty's default. Note that the recovery rate is not taken into account in calculating the exposures and they simply represent the cost of replacing the trades if default occurs. In the presence of a collateral agreement,

$$V^{+}(t) = \max\{V(t) - C(t), 0\}$$

where $V(t)$ and $C(t)$ represent the portfolio value and the available collateral at time t respectively.

Let the distribution of the counterparty's default time, τ_{C}, be given by $F_{\mathrm{C}}(t) = P(\tau_{\mathrm{C}} \leqslant t)$. Then the *unilateral CVA*, denoted by CVA^{U} (and assuming a constant recovery rate R_{C}), is calculated by taking an expectation of equation (2.4.5):

$$\mathrm{CVA}^{U} = E[L] = (1 - R_{\mathrm{C}}) \int_0^T EE^{U+}(t) dF_{\mathrm{C}}(t) \tag{2.4.7}$$

where

$$EE^{U+}(t) = E[D(t)V^{+}(t) | \tau_{\mathrm{C}} = t] \tag{2.4.8}$$

represents a risk-neutral discounted expected exposure (EE^{U+}) given that the counterparty has defaulted at time t. Throughout the following, the discount factors $D(t)$ are computed by using continuous compounding for a given fixed risk-free rate r_0, i.e.,

$$D(t) = \exp(-r_0 t)$$

Note that in calculating CVA using equation (2.4.7), we make no assumption on the dependence of exposures and the counterparty's credit quality. When the counterparty's credit quality and exposure are assumed to be independent of each other, we can replace $EE^{U+}(t)$ by $E[D(t)V^{+}(t)]$ in equation (2.4.7) to calculate CVA.

Similarly, and by taking into account the possibility of the bank's default, the bilateral

CVA, CVA^B, can be calculated by using:

$$\text{CVA}^B = (1-R_{\text{C}})\int_0^T EE^{B+}[t]dF'_{\text{C}}(t) - (1-R_{\text{B}})\int_0^T EE^{B-}[t]dF'_{\text{B}}(t) \quad (2.4.9)$$

where

$$EE^{B+}[t] = E[D(t)V^+(t)|\tau_{\text{C}} = t, \tau_{\text{B}} > t] \quad (2.4.10)$$
$$EE^{B-}[t] = E[D(t)V^-(t)|\tau_{\text{B}} = t, \tau_{\text{C}} > t] \quad (2.4.11)$$

and R_{B} denotes the recovery rate that the counterparty receives in case of the bank's default, τ_B is the default time of the bank, and $V^-(t)$ is given by (2.4.6).

Note that computing EE^{B+} and EE^{B-} for the CVA^B is slightly different from its computation in the CVA^U. The main assumption in calculating the bilateral CVA by using equation (2.4.9) is that the counterparty's expected exposure, defined in equation (2.4.10) (and similarly that of the bank, defined in equation 2.4.11) is calculated conditional on the bank *not* defaulting at or before τ_{C} (at or before τ_{B}). That is in $F'_{\text{C}}(t) = P[\tau_{\text{C}} \leqslant t, \tau_B > \tau_{\text{C}}]$ (and $F'_B(t) = P[\tau_B \leqslant t, \tau_{\text{C}} > \tau_B]$) in equation (2.4.9) we assume that the bank (the counterparty) has not defaulted before the counterparty (the bank). For the sake of simplicity, the probability of both parties defaulting at the same time, $P[\tau_B = \tau_{\text{C}}]$, is assumed to be zero.

2.5 Wrong-way Risk and CVA

As we have pointed out earlier, to devise a comprehensive CVA pricing scheme, we should not ignore the correlation between the counterparty's credit quality and the underlying market movements. The unfavorable correlation between the value of the underlying portfolio and the default of the counterparty is known as a *wrong-way risk*. A wrong-way risk presents itself in increments of exposures when the counterparty's credit quality deteriorates (see Arvanitis and Gregory [36], Cesari et al. [29] and Garcia-Cespedes et al. [37]). Note that we may also have a *right-way risk*, a case in which exposures decrease as their corresponding counterparty's default probability increases. A wrong-way risk is classified into two main categories; a *general wrong-way risk* and a *specific wrong-way risk*:

- **General wrong-way risk**: this risk occurs when systematic factors jointly affect a counterparty's credit quality and the underlying portfolio's value. An example of this can be seen in correlation of some of the counterparties' default times with changes in interest rates or an index.

- **Specific wrong-way risk**: this risk occurs when idiosyncratic factors are correlated with exposures, leading to a high correlation between the counterparty's exposure and its default likelihood. For example, this can happen if a company uses its shares as a collateral. If the company's creditworthiness deteriorates due to an idiosyncratic risk, its share price declines, leading to increased exposure.

In the next section, we briefly describe the existing literature on calculating the unilateral and bilateral CVAs based on the frameworks presented in [2]. Afterward, we present the method of Glasserman and Yang [1] for computing the *worst-case CVA* in the presence of the wrong-way risk.

2.5.1 Worst-Case Unilateral CVA: Dependence of Exposures and Counterparty's Credit Quality

The dependence of the exposures and the counterparty's credit quality affects the CVA calculated in equation (2.4.7).

Assume Y to be the random variable which represents the exposures factors, with the cumulative distribution function $F_Y(y)$. Assume further that the distribution of the default time of the counterparty C is given by $F_{\mathrm{C}}(t)$ as before. To formalize the calculation of the worst-case CVA, assume that we know the distributions of the counterparty's default time, $F_{\mathrm{C}}(t)$, and that of the exposures factors, $F_Y(y)$, but do not know their joint distribution, $F_{\mathrm{C}Y}(t, y)$, i.e., the distributions $F_{\mathrm{C}}(t)$ and $F_Y(y)$ represent the *marginal* distributions of $F_{\mathrm{C}Y}(t, y)$.

We define portfolio losses $L = L(\tau_{\mathrm{C}}, Y)$ and calculate the unilateral CVA (typically done by simulating default events of the respective counterparties across various market scenarios; see sections 2.5.2 and 2.6.2 for a detailed treatment of this approach). Our goal is to determine the joint distribution of (τ_{C}, Y) that maximizes the unilateral CVA:

$$\sup_{F_{\mathrm{C}Y} \in \mathfrak{F}(F_{\mathrm{C}}, F_Y)} \mathrm{CVA}^U \left(L(\tau_{\mathrm{C}}, Y) \right) \tag{2.5.1}$$

where $\mathfrak{F}(F_{\mathrm{C}}, F_Y)$ is the *Fréchet class* of all possible joint distributions of (t, Y) matching the previously defined marginal distributions F_{C} and F_Y.

In the next section we discuss the existing CVA calculation methodologies in practice and how they relate to the worst-case CVA problem.

2.5.2 CVA in Practice

In many practical applications, Monte-Carlo simulation methods are the primary tool for calculating CVA. Assume that we primarily focus on a single counterparty C. The first step in calculating CVA is to simulate counterparty exposures over time. The Monte-Carlo simulation process generates counterparty portfolio values and discount factors for a given set of time points and market scenarios related to the portfolio. Default events are then simulated under each market scenario and equations (2.4.7) and (2.4.9) are used to calculate CVA.

More formally, assume that we are interested in calculating CVA, given a set of M market scenarios y_j, $j = 1, \ldots, M$, each of which has been discretized over a set of $N + 1$

time steps, t_i, $i = 1, \ldots, N+1$. Assume further that the probability of the j-th market scenario is p_j, $j = 1, \ldots, M$.

Let $V(t_i, y_j)$ be the value of the portfolio under the j-th market scenario at time t_i. Then the positive and negative parts of the counterparty value (determining the bank's exposure to the counterparty's default and the counterparty's exposure to the bank's default respectively) are:

$$\begin{aligned} V^+(t_i, y_j) &= \max\{V(t_i, y_j), 0\} \\ V^-(t_i, y_j) &= \max\{-V(t_i, y_j), 0\} \\ & \quad i = 1, \ldots, N+1, \quad j = 1, \ldots, M \end{aligned} \tag{2.5.2}$$

Using (2.4.8) and by averaging the equations in (2.5.2) at a given time point t_i, $i = 1, \ldots, N+1$ and over all scenarios we can derive the discounted expected exposures as:

$$\begin{aligned} EE^{U+}(t_i) &= E(D(t_i)V^+(t_i, y_j)|\tau_{\mathrm{C}} = t_i) = \sum_{j=1}^{M} P(Y = y_j|\tau_{\mathrm{C}} = t_i)D(t_i)V^+(t_i, y_j) \\ & \quad i = 1, \ldots, N+1 \end{aligned} \tag{2.5.3}$$

where $E(\cdot)$ denotes an expectation operator. Assuming the independence of counterparty defaults and exposures, the unilateral CVA is calculated by discretizing the integral in (2.4.7) and by using the expected exposures given in (2.5.3). The discretized unilateral CVA is given by:

$$\mathrm{CVA}^U = (1 - R_{\mathrm{C}}) \sum_{i=1}^{N} \overline{EE}^{U+}(t_i) \cdot (F_{\mathrm{C}}(t_{i+1}) - F_{\mathrm{C}}(t_i)) \tag{2.5.4}$$

where $\overline{EE}^{U+}(t_i)$ is the representative discounted expected exposure over the period (t_i, t_{i+1}) and is defined as:

$$\overline{EE}^{U+}(t_i) = \frac{EE^{U+}(t_i) + EE^{U+}(t_{i+1})}{2}, \quad i = 1, \ldots, N \tag{2.5.5}$$

The discretization scheme used in (2.5.5) is proposed in Basel III (see Basel Committee on Banking Supervision (BCBS) [38] for more details.). Furthermore $(F_{\mathrm{C}}(t+i) - F_{\mathrm{C}}(t_i))$ denotes the probability that the counterparty C defaults at the i-th time interval, $i = 1, \ldots, N$ (note that we have N time intervals corresponding to $(t_1, t_2), \ldots, (t_N, t_{N+1})$).

Similarly, the discretized expression for bilateral CVA is derived based on (2.4.9):

$$\begin{aligned} \mathrm{CVA}^B &= (1 - R_{\mathrm{C}}) \sum_{i=1}^{N} \overline{EE}^{B+}(t_i) \cdot (F'_{\mathrm{C}}(t_{i+1}) - F'_{\mathrm{C}}(t_i)) \\ & \quad -(1 - R_{\mathrm{B}}) \sum_{i=1}^{N} \overline{EE}^{B-}(t_i) \cdot (F'_{\mathrm{B}}(t_{i+1}) - F'_{\mathrm{B}}(t_i)) \end{aligned} \tag{2.5.6}$$

where $\overline{EE}^{B+}(t_i)$ and $\overline{EE}^{B-}(t_i)$ over the i-th time interval, $i = 1, \ldots, N$ are defined by:

$$\overline{EE}^{B+}(t_i) = \frac{EE^{B+}(t_i) + EE^{B+}(t_{i+1})}{2} \tag{2.5.7}$$

$$\overline{EE}^{B-}(t_i) = \frac{EE^{B-}(t_i) + EE^{B-}(t_{i+1})}{2} \tag{2.5.8}$$

for which $EE^{B+}(t_i)$ and $EE^{B-}(t_i)$ for bilateral CVA calculation at each time step $t_i, i = 1, \ldots, N+1$ are given by:

$$EE^{B+}(t_i) = E(D(t_i)V^+(t_i)|\tau_{\mathrm{C}} = t_i, \tau_B > t_i) = \sum_{j=1}^{M} P(Y = y_j|\tau_{\mathrm{C}} = t_i, \tau_B > t_i)D(t_i)V^+(t_i, y_j)$$

$$EE^{B-}(t_i) = E(D(t_i)V^-(t_i)|\tau_{\mathrm{B}} = t_i, \tau_C > t_i) = \sum_{j=1}^{M} P(Y = y_j|\tau_B = t_i, \tau_{\mathrm{C}} > t_i)D(t_i)V^-(t_i, y_j)$$

respectively. In the next sections, we present the *ordered scenario copula* methodology, proposed by Rosen and Saunders [2], for stress testing CVA under a wrong-way risk. This method is designed to capture the wrong-way risk effect when the counterparties credit quality and exposures are correlated. We then present the *worst-case* CVA methodology of Glasserman and Yang [1] for calculating the worst-case unilateral and bilateral CVAs in sections 2.6.2 and 2.6.4.

2.6 Wrong-way Risk and the Ordered Scenario Copula Methodology

2.6.1 Joint Market-Credit Model and OSC

As discussed in section (2.6), estimating the joint market-default probabilities is an important part of capturing the wrong-way risk when calculating CVA. In this section, we take a closer look at the credit risk model, the market risk model and the resulting joint market-credit codependence model which is used in the *ordered scenario copula* methodology.

Credit Risk Model: Using a single-factor copula model for counterparty default times and a Gaussian copula (consistent with the Basel II model; see Basel Committee on Banking Supervision (BCBS) [39] for further details.), we assume that the counterparty's default is driven by its Creditworthiness Index (CWI)[1],

$$\mathrm{CWI}_{\mathrm{C}} \leqslant \Phi^{-1}(F_{\mathrm{C}}(t))$$

that is counterparty C defaults at time t if its creditworthiness index falls below a certain

[1]This notation is commonly used in Basel Accord documentation on credit risk models.

threshold and the creditworthiness index is defined as:

$$\mathrm{CWI_C} = \sqrt{\rho_\mathrm{C}} \cdot Z + \sqrt{1-\rho_\mathrm{C}} \cdot \varepsilon_\mathrm{C}$$

where Z and ε_C are the systematic and idiosyncratic risk factors respectively, following an independent standard normal distribution and ρ_C is the factor loading, giving the sensitivity of counterparty C to the systematic factor Z.

Market Risk Model: As described in section (2.5.2), assume that using a Monte Carlo simulation, M market scenarios (with uniform distribution) over $N+1$ time steps have generated the counterparty exposures, i.e.,

$$P(Y = y_j) = p_j = \frac{1}{M}, \qquad j = 1, \ldots, M$$

This implies a two-dimensional matrix of exposures $V^+(t_i, y_j)$ for all $i = 1, \ldots, N+1, \quad j = 1, \ldots, M$.

Using these simulated exposures, we construct the market risk part of the model as follows. Let s denote a market factor, used to describe exposures and assume that it follows a standard normal distribution. The exposure scenario is determined by using the value of this factor in the following algorithm:

Algorithm 2.6.1: Market Risk Model Algorithm in OSC

---

1. Let Φ be the standard normal cumulative distribution function. Calculate the thresholds H_m, defined as:

$$\mathrm{H}_m = \begin{cases} -\infty, & m = 0 \\ \Phi^{-1}(\overline{P}_m), & m = 1, \ldots, M-1 \\ \infty, & m = M \end{cases} \tag{2.6.1}$$

   where for a fixed m, $\quad m = 1, \ldots, M-1$

$$\overline{P}_m = \sum_{j=1}^{m} p_j \tag{2.6.2}$$

2. Given a random $s \sim N(0,1)$, determine the exposure scenario y_j according to:

$$Y = y_j \iff \mathrm{H}_{j-1} < s \leqslant \mathrm{H}_j, \quad j = 1, \ldots, M \tag{2.6.3}$$

Note that in the step 1 of algorithm 2.6.1 all thresholds H_m are calculated first for all m, $m = 0, \ldots, M$ and then used in step 2 to determine a market scenario y_j.

It is important to distinguish between market scenarios (each of which is denoted by y_j, $j = 1, ..., M$ in algorithm 2.6.1) and the value of the exposures of the bank to the

counterparty's default (or that of the counterparty to the bank's default) under each market scenario y_j and at each time step t_i from which we calculate $V^+(t_i, y_j)$ (and respectively $V^-(t_i, y_j)$). A market scenario y_j, $j = 1, ..., M$, contains the information [2] that is required to calculate $V^+(t_i, y_j)$ and $V^-(t_i, y_j)$ at each time step t_i, $i = 1, \ldots, N+1$. An example of a complete market scenario y_i, depicting the exposure of an ongoing trading relationship between the bank and counterparty for one year, is shown in figure 2.1. A numerical example for how the exposures are simulated under each market scenario is presented in section 2.6.2.

Joint Market-Credit Codependence Model: In this model we assume that the market risk factor Y and the credit risk factor Z are random variables which have a bivariate normal distribution with a given market-credit correlation $\overline{\rho}$, i.e., $(Y, Z) \sim N(\mathbf{0}, \Sigma)$, where:

$$(Y, Z) \sim N(\mathbf{0}, \Sigma), \quad \mathbf{0} = (0, 0), \quad \Sigma = \begin{pmatrix} 1 & \overline{\rho} \\ \overline{\rho} & 1 \end{pmatrix} \tag{2.6.4}$$

In order to specify a particular method that correlates exposures with defaults, we define an *ordering* of the exposure scenarios. For example by using a particular method of *scenario ordering* we can assume that the counterparty portfolio values are directly correlated with its credit quality.

We order the exposure scenarios in an increasing order of the time-averaged total portfolio exposure.[3] Moreover, for a given $\overline{\rho} \in [-1, 1]$, the pre-specified correlation for the systematic credit factor Z and exposure factor Y, we simulate z_0 from their standard bivariate Gaussian distribution (2.6.4), assuming that $Y = s$ is the same as the one that we generated in the step 2 of algorithm 2.6.1. That is, given $Y = s$, the distribution of Z becomes $N(\overline{\rho}, 1 - \overline{\rho}^2)$ and

$$z_0 = s \cdot \overline{\rho} + \sqrt{1 - \overline{\rho}^2} \cdot r$$

where r is a standard normal random variable. Finally, by generating ε_{C} for the idiosyncratic risk factor we can compute

$$\mathrm{CWI}_{\mathrm{C}} = \sqrt{\rho_{\mathrm{C}}} \cdot z_0 + \sqrt{1 - \rho_{\mathrm{C}}} \cdot \varepsilon_{\mathrm{C}}$$

and determine whether or not counterparty C has defaulted. By ordering the exposure scenarios in an increasing order of the time-averaged total portfolio exposure and choosing a positive market-credit correlation $\overline{\rho}$, we will get higher concurrent default rates and

[2] Note that depending on the application, Y could be a random variable, a random vector or a stochastic process.

[3] Assume that T is the longest maturity of the contracts in the counterparty's portfolio with the bank. Then this is simply done by computing $\frac{1}{T}\sum_{i=1}^{N}[\frac{V^+(t_i,y_j)+V^+(t_{i+1},y_j)}{2} \cdot (t_{i+1} - t_i)]$ for all j, $j = 1, \ldots, M$ and sorting them in an increasing order to determine $j_{(1)}, j_{(2)}, \ldots, j_{(M-1)}, j_{(M)}$, the new ordering of the exposure scenarios for which we have:

$$\sum_{i=1}^{N}[\frac{V^+(t_i, y_{j_{(1)}}) + V^+(t_{i+1}, y_{j_{(1)}})}{2} \cdot (t_{i+1} - t_i)] \leqslant \ldots \leqslant \sum_{i=1}^{N}[\frac{V^+(t_i, y_{j_{(M)}}) + V^+(t_{i+1}, y_{j_{(M)}})}{2} \cdot (t_{i+1} - t_i)]$$

exposures. Note that Rosen and Saunders [2] suggest alternative methods (such as *single scenario ordering* and *time-dependent scenario ordering*) to correlate the exposures and defaults.

2.6.2 Worst-Case Unilateral CVA Optimization

To demonstrate the worst-case CVA methodology of Glasserman and Yang [1], we start this section by describing the exposure dynamics of the portfolio that the bank has with a single counterparty. In order to simplify the model, we only consider the aggregated exposure between the counterparty and the bank. Assume that this aggregated exposure is modelled by using an Ornstein-Uhlenbeck process $X(t)$:

$$dX(t) = \kappa(\mu - X(t))dt + \sigma dW(t) \tag{2.6.5}$$

where μ is the level toward which the aggregated exposure reverts, κ is the rate of the mean reversion, σ is the instantaneous standard deviation of $X(t)$ and $W(t)$ is a standard Brownian motion.

As we have shown in figure 2.1, using an O-U process allows us to assume a positive aggregated exposure for either the bank or the counterparty. The exposure dynamics remain stationary overall. The mean-reverting coefficient is $\kappa = 1$, $\mu = 0$, and σ is set as 20% in (2.6.5), and $X(0) = 0$, indicating that neither the bank nor the counterparty has exposure to the other at time zero.

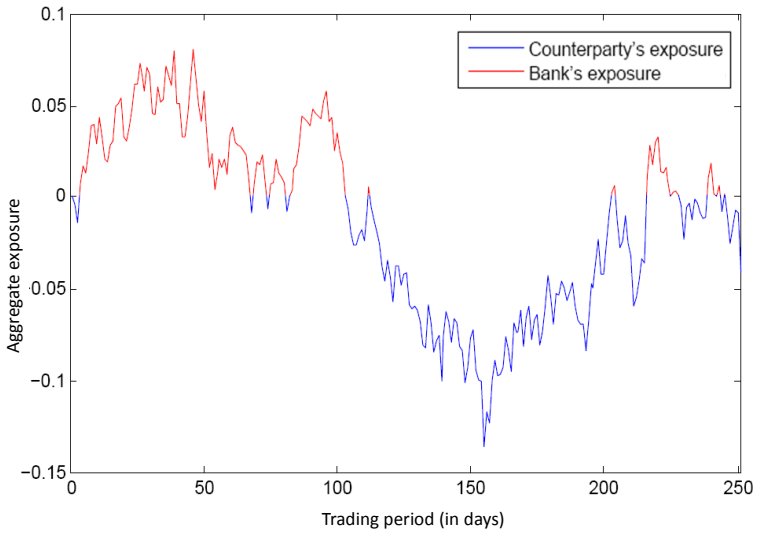


FIGURE 2.1: *An instance of the market scenario y_j depicting the bank's exposure to the counterparty's default (red path) and the counterparty's exposure to the bank's default (blue path) for a period of 250 trading days.*

Assume further that we simulate M market scenarios (each denoted by $y_j, j = 1, 2, \ldots, M$) according to the process (2.6.5). That is, each simulated market scenario

$y_j, j = 1, 2, \ldots, M$ contains the aggregate exposures information for both the bank and counterparty (as shown in figure 2.1) over a predefined time horizon T, the longest maturity of the contract in the counterparty's portfolio with the bank.

Each scenario is equally likely, i.e., they are drawn from a Monte-Carlo simulation and:

$$P(Y = y_j) = p_j = \frac{1}{M}, \qquad j = 1, \ldots, M \tag{2.6.6}$$

Recall from equation (2.4.5) that the first step in calculating CVA is to determine the default status of the counterparty:

$$L = 1_{\{\tau_{\mathrm{C}} \leqslant T\}}(1 - R_{\mathrm{C}})D(\tau_{\mathrm{C}})V^{+}(\tau_{\mathrm{C}}) \tag{2.6.7}$$

Next we define a credit model. Assume that the counterparty default time follows an exponential distribution with a prespecified parameter λ. Under this distribution, counterparty C defaults before time τ_{C} where

$$\tau_{\mathrm{C}} \sim \exp(\lambda) \tag{2.6.8}$$

Our goal is to find the *worst-case unilateral* CVA by solving the following optimization problem:

$$\begin{aligned} &\max \sum_{i=1}^{N+1} \sum_{j=1}^{M} \mathrm{E}_{\vartheta}(L) \\ &\sum_{i=1}^{N+1} \vartheta_{ij} = p_j, \quad j = 1, \ldots, M \\ &\sum_{j=1}^{M} \vartheta_{ij} = q_i, \quad i = 1, \ldots, N+1 \\ &\vartheta_{ij} \geqslant 0, \quad i = 1, \ldots, N+1, \quad j = 1, \ldots, M \end{aligned} \tag{2.6.9}$$

where

$$\mathrm{CVA}^{U}(L) = \mathrm{E}_{\vartheta}(L)$$

and $p_j = P(Y = y_j), j = 1, \ldots, M$ is defined in (2.6.6) and $q_i, i = 1, \ldots, N+1$ is the corresponding discretized default time distribution given in (2.6.8), defined as:

$$\begin{aligned} q_i &= F_{\mathrm{C}}(t_{i+1}; \lambda) - F_{\mathrm{C}}(t_i; \lambda), \qquad i = 1, \ldots, N \\ q_i &= 1 - F_{\mathrm{C}}(t_i; \lambda), \quad i = N+1 \end{aligned} \tag{2.6.10}$$

where $F_{\mathrm{C}}(t; \lambda)$ is the cumulative distribution function of the exponential distribution that governs the default of counterparty C.

Note that we assume that the defaults can occur at or before time step $t_i, i \leqslant N+1$ and there is no default event after $t_{N+1} = T$, the longest maturity of a contract in the portfolio.

$\vartheta_{ij} = P(t_i < \tau_C \leqslant t_{i+1}, Y = y_j)$ denotes the joint probability of market scenario y_j and the default event occurring in the time interval $(t_i, t_{i+1}]$.

Moreover, $P(\tau_C, Y) \in \mathfrak{F}(F_C, F_Y)$, the *Fréchet class* of all possible joint distributions of (τ_C, Y) matching the prescribed discretized marginals F_C and F_Y. It is not necessary to specify the additional constraint that the total sum of the joint distribution is equal to one as the sum of each marginal distribution is equal to one.

2.6.3 Numerical Algorithm for the Worst-case Unilateral CVA: Dependence of Exposures and Counterparty's Credit Quality

Problem (2.6.9) can be recast as the following linear programming problem:

$$\begin{aligned} &\max \sum_{i=1}^{N+1} \sum_{j=1}^{M} l_{ij}\vartheta_{ij} \\ &\sum_{i=1}^{N+1} \vartheta_{ij} = p_j, \quad j = 1, \ldots, M \\ &\sum_{j=1}^{M} \vartheta_{ij} = q_i, \quad i = 1, \ldots, N+1 \\ &\vartheta_{ij}, \geqslant 0 \quad i = 1, \ldots, N+1, \quad j = 1, \ldots, M \end{aligned} \tag{2.6.11}$$

where

$$\begin{aligned} l_{ij} &= \frac{1}{2}(1 - R_C)\left(D(t_i)V^+(t_i, y_j) + D(t_{i+1})V^+(t_{i+1}, y_j)\right), \quad i = 1, \ldots, N, \quad j = 1, \ldots, M \\ l_{ij} &= \frac{1}{2}(1 - R_C)\left(D(t_i)V^+(t_i, y_j)\right), \quad i = N+1, \quad j = 1, \ldots, M \end{aligned} \tag{2.6.12}$$

The first constraint ensures that the paths of the market factors have equal weights and the second ensures that the default time distribution in the joint model has the correct marginal distribution. The last equation in (2.6.12) when computing the coefficients of the linear programming problem (2.6.11) corresponds to the final time step as there are no defaults for any $t > t_{N+1}$ (in all market scenarios $j = 1, \ldots, M$) and $V^+(t, y_j) = 0$ as it is defined in equation (2.5.2). The running example considered by Glasserman and Yang [1] has a structure of a transportation problem for which efficient algorithms (such as a strictly polynomial algorithm) can be used.

Algorithm 2.6.3: Worst-case Unilateral CVA Algorithm

---

1. Simulate M market scenarios to generate aggregated exposures between the counterparty and the bank according to the process (2.6.5):

$$dX(t) = \kappa(\mu - X(t))dt + \sigma dW(t), \quad X(0) = 0$$

2. Construct the discretized marginal distributions of the market scenarios, p_j, and the default time q_i using:

$$\begin{aligned} q_i &= F_{\mathrm{C}}(t_{i+1};\lambda) - F_{\mathrm{C}}(t_i;\lambda), \qquad i = 1,\ldots,N \\ q_i &= 1 - F_{\mathrm{C}}(t_i;\lambda), \qquad i = N+1 \\ p_j &= \frac{1}{M}, \qquad j = 1,\ldots,M \end{aligned}$$

3. For all market scenarios y_j and time steps t_i calculate $V^+(t_i,y_j) = \max\{V(t_i,y_j),0\}$ and define l_{ij} according to

$$l_{ij} = \frac{1}{2}(1-R_{\mathrm{C}})\left(D(t_i)V^+(t_i,y_j) + D(t_{i+1})V^+(t_{i+1},y_j)\right), \quad i = 1,\ldots,N, \quad j = 1,\ldots,M$$

$$l_{ij} = \frac{1}{2}(1-R_{\mathrm{C}})\left(D(t_i)V^+(t_i,y_j)\right), \quad i = N+1, \quad j = 1,\ldots,M$$

4. Find the optimal worst-case distribution ϑ_{ij}, $i = N+1$, $j = 1,\ldots,M$ of the market scenarios and the counterparty's default time by solving (2.6.11)

5. Calculate the worst-case unilateral CVA as:

$$\mathrm{CVA}^U = \sum_{i=1}^{N+1}\sum_{j=1}^{M} l_{ij}\vartheta_{ij} \tag{2.6.13}$$

In order to assess the performance of the *worst-case unilateral CVA* methodology described above, we start by simulating $M = 10{,}000$ market scenarios, generating the aggregated exposure between the counterparty and the bank over a $T = 5$ year time horizon using the Ornstein-Uhlenbeck process in (2.6.5) with the specified configuration of parameters described in section 2.6.2. Each exposure scenario is simulated by using daily time steps, i.e., each scenario consists of the aggregated exposures for 1250 trading days based on an Ornstein-Uhlenbeck process. Then by using a recovery rate of $R_{\mathrm{C}} = 0.3$ and a risk-free rate of $r_0 = 5\%$ (for computing the discount factors at each time step t_i, $i = 1,\ldots,N+1$, using continuous compounding), we define the coefficients of the linear programming problem in algorithm 2.6.3. As noted earlier, the marginal distribution of market scenarios, p_j, is defined as $1/M$ in the linear programming problem (2.6.11).

Next, we discretize the default time distributions. Recall that we have assumed that the default time of counterparty C follows an exponential distribution. This is achieved by defining q_i in the linear programming problem in (2.6.11) to be:

$$q_i = F_{\mathrm{C}}(t_{i+1};\lambda) - F_{\mathrm{C}}(t_i;\lambda), \quad i = 1,\ldots,N \tag{2.6.14}$$

where $N = 1250$ and an equally spaced grid is used in equation (2.6.14). Moreover, for the final time step ($i = N+1$), q_{N+1} reduces to $1 - F_{\mathrm{C}}(T;\lambda)$.

Having defined the discretized marginal distributions of the exposure scenarios and the counterparty default time, we use algorithm (2.6.3) to derive the worst-case coupling of the market scenarios and the counterparty defaults. The worst-case unilateral CVA is

then found by taking the expectation of the l_{nm} defined in (2.6.12) with respect to this new joint distribution.

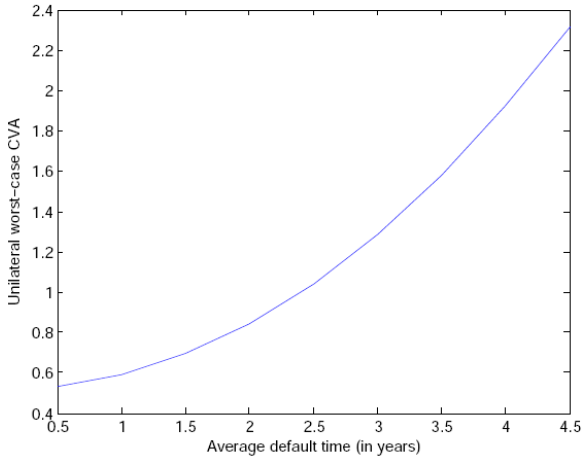


FIGURE 2.2: *Unilateral CVA calculated using the worst-case CVA method.*

The results are shown in figure 2.2. To generate figure 2.2, 9 instances of the linear programming problem (2.6.11) are solved by using different average default rates (parameterized using equation (2.6.14)) for the counterparty C, i.e.,

$$\tau_{\mathrm{C}} \sim \exp(\lambda_i) \quad \lambda_i \in \{0.5, 1, \ldots, 4.5\} \tag{2.6.15}$$

Each instance of the linear programming problems is solved by using IBM ILOG CPLEX Optimization Studio. Appendix E provides an overview of the CPLEX Optimization Studio and a description of various optimization methods provided in CPLEX as well as different solution status codes and their respective meaning. The results shown in table 2.1 were produced on a platform that uses an AMD 3.2 GHz Phenom II X4 955 processor with 16 GB RAM.

λ_{C}	T	M	N+1	run time (in seconds)	sum of primal residuals	sum of dual residuals	duality gap	solution status code
0.5	5	10000	1251	528.9764	1.19e-07	1.071e-07	6.9908e-09	1
1	5	10000	1251	435.4414	4.9836e-07	4.4853e-07	8.909e-09	1
1.5	5	10000	1251	544.6313	9.5974e-07	8.6377e-07	9.5929e-09	1
2	5	10000	1251	550.9365	3.4039e-07	3.0635e-07	5.4722e-09	1
2.5	5	10000	1251	404.5313	5.8527e-07	5.2674e-07	1.3862e-09	1
3	5	10000	1251	452.3975	2.2381e-07	2.0143e-07	1.4929e-09	1
3.5	5	10000	1251	412.0297	7.5127e-07	6.7614e-07	2.5751e-09	1
4	5	10000	1251	545.5098	2.551e-07	2.2959e-07	8.4072e-09	1
4.5	5	10000	1251	463.7388	5.0596e-07	4.5536e-07	2.5428e-09	1

TABLE 2.1: *Run time, sum of primal residuals, sum of dual residuals, duality gap and solution status code for the 9 linear programming problems (2.6.11) associated with different average default rates given in* (2.6.15).

As we have shown in table 2.1, we check the quality of the solution by using various measures. The *solution status code* of 1 indicates that an optimal solution has been found. Other solution status codes, listed in Appendix E, are indicative of the algorithm encountering an issue when solving linear programming problems. We further check the quality of the solution by using the duality gap and the sum of primal and dual residuals for each linear programming problem.

Alternatively, we can calculate the unilateral CVA corresponding to each λ_i in (2.6.15) by using the ordered scenario copula, discussed in section 2.6. The results are shown in figure 2.3. In order to compare these results with the performance of the worst-case

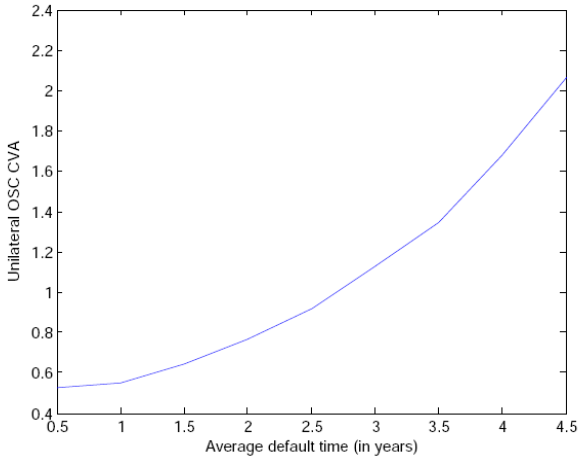


FIGURE 2.3: *Unilateral CVA calculated using ordered scenario copula method.*

unilateral CVA methodology described above, we use the worst-case copula results as a benchmark and calculate the ratios of

$$\frac{\text{CVA}^U_{OSC_i}}{\text{CVA}^U_{WC_i}}, \quad i = 1, \ldots, 9 \tag{2.6.16}$$

for each instance of the corresponding ordered scenario copula and the worst-case CVA (as proposed by Glasserman and Yang [1]) given a fixed average default rate described in (2.6.15). The resulting ratios are plotted against the average default rate in years and are shown in figure 2.4.

As noted earlier in section 2.6.1, the co-dependence of the exposures and the defaults are determined by the market-credit correlation, $\overline{\rho}$, in the ordered scenario copula. The market-credit correlation of $\overline{\rho} = 1$ is chosen in all of the nine instances in (2.6.16). As can be seen in figure 2.4, the worst-case unilateral CVA values are approximately 5 to 11% larger than those calculated by the ordered scenario copula.

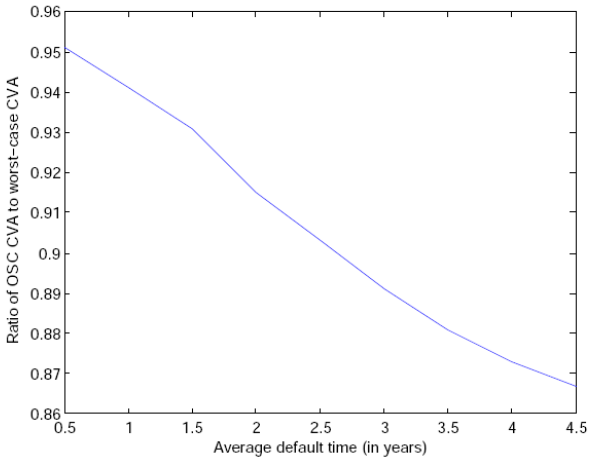


FIGURE 2.4: *Ratio of OSC CVA to worst-case CVA.*

2.6.4 Worst-Case Bilateral CVA: Dependence of Exposures and Counterparties' Credit Quality

An intrinsic aspect of counterparty risk is that it is *bilateral*, in the sense that either the bank or the counterparty can default. Furthermore, the bilateral characteristic of the credit exposure makes the quantification of counterparty risk in the presence of wrong-way risk more challenging. An important difference between the bilateral CVA and the unilateral CVA is that bilateral CVA, seen as the risk premium that either counterparty pays to the other one, changes sign over time due to the changes in the relative riskiness and exposure of the counterparties. To formalize the problem and calculate the worst-case bilateral CVA in the presence of dependence between the exposures and the counterparty's credit quality, we assume that the marginal distribution of the counterparty's default time, $F_{\mathrm{C}}(t)$, the marginal distributions of the bank's default time, $F_{\mathrm{B}}(t)$ and that of the market factors, $F_Y(y)$, are known but their joint distribution, $F_{\mathrm{BCY}}(\tau_{\mathrm{B}}, \tau_{\mathrm{C}}, y)$ is unknown. In this case we can define portfolio losses $L = L(\tau_{\mathrm{B}}, \tau_{\mathrm{C}}, Y)$ and calculate the bilateral CVA as described in sections 2.5.2 and 2.6.5.

To determine the joint distribution of $(\tau_{\mathrm{B}}, \tau_{\mathrm{C}}, Y)$ that maximizes bilateral CVA, we solve the following optimization problem:

$$\sup_{F_{\mathrm{BCY}} \in \mathfrak{F}(F_{\mathrm{B}}, F_{\mathrm{C}}, F_Y)} \mathrm{CVA}^B \left(L(\tau_{\mathrm{B}}, \tau_{\mathrm{C}}, Y) \right) \tag{2.6.17}$$

where $\mathfrak{F}(F_{\mathrm{B}}, F_{\mathrm{C}}, F_Y)$ is the *Fréchet class* of all possible joint distributions of $(t_{\mathrm{B}}, t_{\mathrm{C}}, Y)$ matching the given marginal distributions F_{B}, F_{C} and F_Y.

In the next section, we take a closer look at the default dynamics of the bank and the counterparty when computing the bilateral CVA and present a numerical algorithm for solving the problem stated in (2.6.17).

2.6.5 Worst-Case Bilateral CVA Optimization

We consider three distinct default scenarios for the bank and the counterparty, shown in figure 2.5:

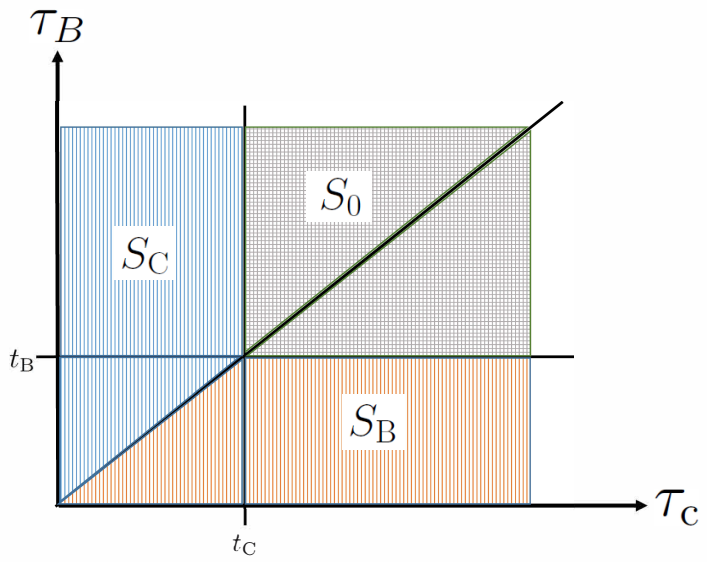


FIGURE 2.5: *Bank and Counterparty's default events decomposition.*

- If $\tau_C < \tau_B$, then the bank experiences losses given that it has a positive exposure to the counterparty. This corresponds to the region S_C in figure 2.5:

$$S_C = \{\tau_C \leqslant T\} \cap \{\tau_C < \tau_B\}$$

- If $\tau_B < \tau_C$, then the counterparty experiences losses given that it has positive exposure to the bank. This corresponds to the region S_B in figure 2.5:

$$S_B = \{\tau_B \leqslant T\} \cap \{\tau_B < \tau_C\}$$

- Finally, neither the counterparty nor the bank defaults. This region is denoted S_0 in figure 2.5 and:

$$S_0 = \{\tau_B > T\} \cap \{\tau_C > T\}$$

Assume further that concurrent default events do not happen, i.e., under the above decomposition of default events, we have:

$$\mathbf{1}_{\mathbf{S}_\mathrm{B}} + \mathbf{1}_{\mathbf{S}_\mathrm{C}} + \mathbf{1}_{\mathbf{S_0}} = 1 \tag{2.6.18}$$

where $\mathbf{1}_{\{.\}}$ is the indicator function of defaults. The exposure dynamics used in this section are similar to what we have described earlier in (2.6.5), which represents the net aggregated exposure of the parties to one another in the case of default. Note that as the underlying market factors vary, the exposure can become negative or positive for either party as is illustrated in figure 2.1.
If the counterparty defaults at time $\tau_\mathrm{C} < \tau_\mathrm{B} < T$, the bank's loss would be:

$$L_\mathrm{B} = 1_{\{\tau_\mathrm{C}<\tau_\mathrm{B}<T\}}(1 - R_\mathrm{C})D(\tau_\mathrm{C})V^+(\tau_\mathrm{C}) \tag{2.6.19}$$

and in the case of a bank default at time $\tau_\mathrm{B} < \tau_\mathrm{C} < T$, the counterparty's loss is calculated as:

$$L_\mathrm{C} = 1_{\{\tau_\mathrm{B}<\tau_\mathrm{C}<T\}}(1 - R_\mathrm{B})D(\tau_\mathrm{B})V^-(\tau_\mathrm{B}) \tag{2.6.20}$$

where $V^+(\tau_\mathrm{C})$ and $V^-(\tau_\mathrm{B})$, as defined in (2.5.2), denote the positive exposure of either party to the other one in the case of default. M simulated market scenarios, each denoted by y_k, $k = 1, 2, \ldots, M$, which are generated by using an Ornstein-Uhlenbeck process X_t, and described in (2.6.5), drive the aggregate exposures.

Each market scenario y_k, $k = 1, 2, \ldots, M$ contains the exposure information of both the bank and the counterparty for calculating $V^+(\tau_\mathrm{C})$ and $V^-(\tau_\mathrm{B})$ at any given set of points in time (t_j, t_j). As before t_i, $i = 1, 2, \ldots, N_1 + 1$ and t_j, $j = 1, 2, \ldots, N_2 + 1$ correspond to the time steps used for discretizing the distribution of the bank's default time and that of the counterparty respectively.

$V^+(\tau_\mathrm{C})$ and are $V^-(\tau_\mathrm{B})$ are calculated based on this over a given time horizon T, which is the longest maturity of the contract in the counterparty's portfolio with the bank.

As before, we assume that the market scenarios have equal probability:

$$P(Y = y_k) = p_k = \frac{1}{M}, \quad k = 1, \ldots, M \tag{2.6.21}$$

Assume further that the default events for the bank and the counterparty are exponentially distributed with given parameters λ_1 and λ_2, i.e., $\tau_B \sim \exp(\lambda_1)$ and $\tau_\mathrm{C} \sim \exp(\lambda_2)$.

2.6.6 Numerical Algorithm for the Worst-case Bilateral CVA: Dependence between Exposures and Counterparties' Credit Quality

The calculation of worst-case bilateral CVA described in section 2.6.5 can be written in the form of a linear programming problem as follows:

$$\begin{aligned}
&\max \sum_{i=1}^{N_1+1}\sum_{j=1}^{N_2+1}\sum_{k=1}^{M} l_{ijk}\vartheta_{ijk} \qquad (2.6.22)\\
&\sum_{i=1}^{N_1+1}\sum_{j=1}^{N_2+1} \vartheta_{ijk} = p_k, \quad k = 1,\dots,M\\
&\sum_{k=1}^{M}\sum_{j=1}^{N_2+1} \vartheta_{ijk} = r_i, \quad i = 1,\dots,N_1+1\\
&\sum_{k=1}^{M}\sum_{i=1}^{N_1+1} \vartheta_{ijk} = q_j, \quad j = 1,\dots,N_2+1\\
&\vartheta_{ijk} \geqslant 0, \quad i = 1,\dots,N_1+1, \quad j = 1,\dots,N_2+1, \quad k = 1,\dots,M
\end{aligned}$$

where $p_k = P(Y = y_k), k = 1,\dots,M$ is defined in (2.6.21) and $r_i, i = 1,\dots,N_1+1$ and $q_j, j = 1,\dots,N_2+1$ represent the corresponding discretized default time distributions of the bank and the counterparty that are $\exp(\lambda_1)$ and $\exp(\lambda_2)$-distributed respectively.

While the calculation of the coefficients l_{ij} in the unilateral worst-case CVA optimization problem (2.6.9) is straightforward, the construction of the corresponding coefficients, l_{ijk}, in the bilateral worst-case CVA problem (2.6.22) is slightly different and a little bit more involved. Recall that the decomposition of default times described in section 2.6.5 implies that concurrent default events do not happen at any time $t < T$. Based on this assumption and given the k-th market scenario ($k = 1,\dots,M$), the bank's loss due to the counterparty's default in the linear programming problem (2.6.22) is calculated as:

$$\begin{aligned}
l_{ijk} &= \frac{1}{2}(1-R_{\mathrm{C}})\left(D(t_j)V^+(t_j,y_k) + D(t_{j+1})V^+(t_{j+1},y_k)\right)\\
&\qquad \forall t_j < t_i, \quad i = 1,\dots,N_1+1, \quad j = 1,\dots,N_2, \quad k = 1,\dots,M\\
l_{ijk} &= \frac{1}{2}(1-R_{\mathrm{C}})\left(D(t_j)V^+(t_j,y_k)\right)\\
&\qquad \forall t_j < t_i, \quad i = 1,\dots,N_1+1, \quad j = N_2+1, \quad k = 1,\dots,M \qquad (2.6.23)
\end{aligned}$$

Similarly, the counterparty's loss to the bank default becomes:

$$\begin{aligned}
l_{ijk} &= -\frac{1}{2}(1-R_{\mathrm{B}})\left(D(t_i)V^-(t_i,y_k) + D(t_{i+1})V^-(t_{i+1},y_k)\right)\\
&\qquad \forall t_i < t_j, \quad i = 1,\dots,N_1, \quad j = 1,\dots,N_2+1, \quad k = 1,\dots,M\\
l_{ijk} &= -\frac{1}{2}(1-R_{\mathrm{B}})\left(D(t_i)V^-(t_i,y_k)\right)\\
&\qquad \forall t_i < t_j, \quad i = N_1+1, \quad j = 1,\dots,N_2+1, \quad k = 1,\dots,M \qquad (2.6.24)
\end{aligned}$$

The condition $t_j < t_i$ in equation (2.6.23) (and $t_i < t_j$ in equation (2.6.24)) indicates that the default of the counterparty has occurred prior to that of the bank (or the default of the bank has occurred prior to that of the counterparty). We summarize these steps in the following algorithm:

Algorithm 2.6.6: Worst-case Bilateral CVA Algorithm for Computing Bilateral CVA

---

1. Simulate M market scenarios to generate aggregated exposures between the counterparty and the bank according to the process (2.6.5):

$$dX(t) = \kappa(\mu - X(t))dt + \sigma dW(t), \quad X(0) = 0,$$

2. Construct the discretized marginal distributions of market scenarios (p_k's), the bank and counterparty default times, (r_i's) and (q_j's) by using:

$$\begin{aligned} r_i &= F_{\mathrm{B}}(t_{i+1};\lambda_1) - F_{\mathrm{B}}(t_i;\lambda_1), \qquad i = 1,\ldots,N_1 \\ r_i &= 1 - F_{\mathrm{B}}(t_i;\lambda_1), \qquad i = N_1+1 \\ q_j &= F_{\mathrm{C}}(t_{j+1};\lambda_2) - F_{\mathrm{C}}(t_j;\lambda_2), \qquad j = 1,\ldots,N_2 \\ q_j &= 1 - F_{\mathrm{C}}(t_j;\lambda_2), \qquad j = N_2+1 \\ p_k &= \frac{1}{M}, \qquad k = 1,\ldots,M \end{aligned} \tag{2.6.25}$$

3. Given a market scenario y_k, banks's default times t_i and counterparty's default times t_j, calculate l_{ijk} by using:

$$\begin{aligned} l_{ijk} &= \frac{1}{2}(1-R_{\mathrm{C}})\left(D(t_j)V^+(t_j,y_k) + D(t_{j+1})V^+(t_{j+1},y_k)\right) \\ &\qquad \forall t_j < t_i, \quad i = 1,\ldots,N_1+1, \quad j = 1,\ldots,N_2, \quad k = 1,\ldots,M \\ l_{ijk} &= \frac{1}{2}(1-R_{\mathrm{C}})\left(D(t_j)V^+(t_j,y_k)\right) \\ &\qquad \forall t_j < t_i, \quad i = 1,\ldots,N_1+1, \quad j = N_2+1, \quad k = 1,\ldots,M \\ l_{ijk} &= -\frac{1}{2}(1-R_{\mathrm{B}})\left(D(t_i)V^-(t_i,y_k) + D(t_{i+1})V^-(t_{i+1},y_k)\right) \\ &\qquad \forall t_i < t_j, \quad i = 1,\ldots,N_1, \quad j = 1,\ldots,N_2+1, \quad k = 1,\ldots,M \\ l_{ijk} &= -\frac{1}{2}(1-R_{\mathrm{B}})\left(D(t_i)V^-(t_i,y_k)\right) \\ &\qquad \forall t_i < t_j, \quad i = N_1+1, \quad j = 1,\ldots,N_2+1, \quad k = 1,\ldots,M \end{aligned}$$

4. Find the optimal worst-case coupling ϑ_{ijk}, $i = 1,\ldots,N_1+1$, $j = 1,\ldots,N_2+1$, $k = 1,\ldots,M$ of default times of the bank, the counterparty and that of the market scenarios by solving the linear programming problem (2.6.22)

5. Calculate the worst-case bilateral CVA as:

$$\mathsf{CVA}^B = \sum_{i=1}^{N_1+1}\sum_{j=1}^{N_2+1}\sum_{k=1}^{M} l_{ijk}\vartheta_{ijk} \tag{2.6.26}$$

Simulation of the market scenarios that contain an exposure of the bank and the counterparty to one another is the first step in implementing the *worst-case bilateral CVA* methodology. Although we keep the number of the simulated market scenarios at $M = 10{,}000$ (similar to what was used in the unilateral CVA optimization numerical example), the time steps used in these simulations are different. Instead of using daily time steps, we consider the exposure scenarios to be simulated on a monthly basis. The reason for choosing a monthly time frame is twofold. First, in many practical examples, exposures are simulated over a longer time horizon T. Choosing a monthly time step enables us to replicate this condition better. Second, if we were to choose a daily time step for exposure scenarios for the linear programming problem (2.6.22), the construction of the coefficient matrix of the linear programming problem would become highly time-intensive and impractical. The coefficient matrix of the constraints in the linear programming problem (2.6.22) has $M + N_1 + N_2 + 2$ rows and $M \times (N_1 + 1) \times (N_2 + 1)$ columns and choosing daily time steps for generating the market scenarios would be highly time-intensive.

As before, an Ornstein-Uhlenbeck process described in section 2.6.2 is assumed to be the main driver of the aggregate exposure and the marginal distributions of market scenarios, (p_k's), is defined as $p_k = 1/M$, $k = 1, \ldots, M$ in the linear programming problem (2.6.22). Using the discretization scheme in (2.6.25), we define the marginal distributions $r_i, i = 1, \ldots, N_1 + 1$ and $q_j, j = 1, \ldots, N_2 + 1$ for the default time of the bank and the counterparty in the linear programming problem (2.6.22). The recovery rate of both the bank and the counterparty is assumed to be 0.3, i.e., $R_{\mathrm{B}} = 0.3$ and $R_{\mathrm{C}} = 0.3$ and a risk-free rate of $r_0 = 5\%$ (for computing the discount factors at each time step by continuous compounding) is used throughout for defining the coefficients of the linear programming problem in algorithm 2.6.6. Finally, we can solve the linear programming problem (2.6.22) by using the worst-case bilateral CVA algorithm described above and calculate the worst-case bilateral CVA.

In this section, we present two sets of results based on contracts maturing at $T = 5$ and $T = 10$ years and construct a surface for the ratios of CVA^Bs calculated by the ordered scenario copula (using market-credit correlation $\overline{\rho} = 1$ as in section 2.6.3) and the worst-case bilateral CVA methodology.

The results are shown in figure 2.6 for which $T = 5$ and

$$\begin{aligned} \tau_{\mathrm{B}} &\sim \exp(\lambda_i), \quad \lambda_i \in \{0.5, 1, \ldots, 4.5\} \\ \tau_{\mathrm{C}} &\sim \exp(\lambda_i), \quad \lambda_i \in \{0.5, 1, \ldots, 4.5\} \end{aligned} \tag{2.6.27}$$

and in figure 2.7 for which $T = 10$ and

$$\begin{aligned} \tau_{\mathrm{B}} &\sim \exp(\lambda_i), \quad \lambda_i \in \{1, 2, \ldots, 9\} \\ \tau_{\mathrm{C}} &\sim \exp(\lambda_i), \quad \lambda_i \in \{1, 2, \ldots, 9\} \end{aligned} \tag{2.6.28}$$

In generating each of these surfaces, the market scenarios y_k, $k = 1, \ldots, M$ have been kept the same. Moreover, each point on these surfaces is found by the following steps:

1. (a) Fix an average default rate for the bank, τ_{B}, and use (2.6.25) to define

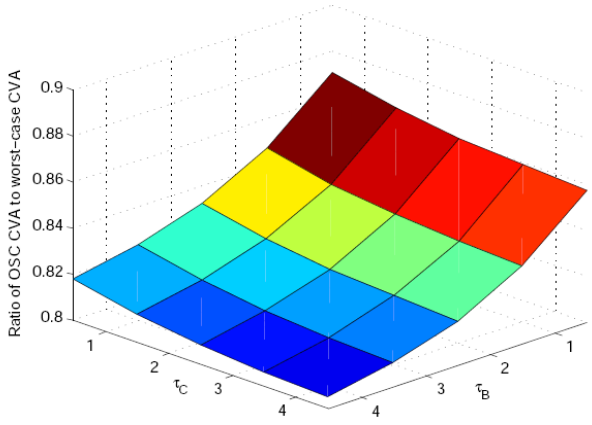


FIGURE 2.6: *Ratio of the bilateral* CVA *computed with the ordered scenario copula to the worst-case* CVA*;* $T = 5$.

$r_i, i = 1, \ldots, N_1 + 1$;

(b) Fix an average default rate for the counterparty, τ_{C}, and use (2.6.25) to define $q_j, j = 1, \ldots, N_2 + 1$;

(c) Solve the linear programming problem (2.6.22).

2. Calculate the bilateral CVA using the ordered scenario copula (described in section 2.6) for the above fixed average default rate for the counterparty, τ_{C} and the average default rate for the bank, τ_{B};

3. Calculate the ratios of

$$\frac{\mathrm{CVA}^{B}_{OSC_i}}{\mathrm{CVA}^{B}_{WC_i}}, \quad i = 1, \ldots, N_1^0 \times N_2^0$$

where N_1^0 and N_2^0 are the numbers of the points that we have used for discretizing the τ_{B} and τ_{C} axis (corresponding to a given average default time rate λ_{B} and λ_{C}) respectively.

In generating both figures 2.6 and 2.7, $N_1^0 = 9$ and $N_2^0 = 9$, corresponding to the equations (2.6.27) and (2.6.28) respectively. As before we have used CPLEX Optimization Studio for solving each instance of the linear programming problem (2.6.22) and the results are shown in tables E.3 and E.4 in section E.0.2 in Appendix E.

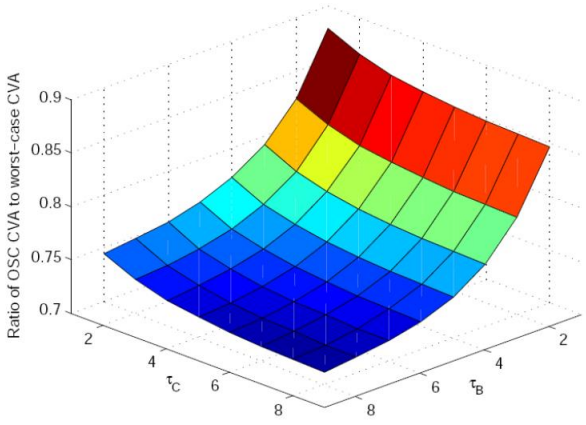


FIGURE 2.7: *Ratio of the bilateral* CVA *computed with the ordered scenario copula to the worst-case* CVA*;* $T = 10$.

Figures 2.6 and 2.7 show similar results to what was found in section 2.6.3 for the unilateral CVA optimization. The worst-case bilateral CVA methodology produces CVA bounds that are higher than CVA computed with the ordered scenario copula in both examples (between 13% to 19% higher in the $T = 5$ case and 14% to 27% higher in the $T = 10$ case).

The marginal distributions of the market risk factors and the credit risk factors used in both the worst-case unilateral and bilateral CVA algorithms described earlier are estimated in practice. Note that Glasserman and Yang [1] prove the convergence of the solution of the linear programming problem (2.6.9) to the true worst-case CVA by using the the results of Villani [40] on the convergence of Wasserstein distance of empirical measures.

2.7 Bounds on CVA Contributions

In sections 2.6.2 and 2.6.5, we demonstrated the application of the worst-case CVA methodology for bounding unilateral and bilateral CVAs. We are also interested in determining the individual contributions of the trades comprising the portfolio to the CVA^U.

In this section, we discuss the computation of CVA^U at the trade level in sections 2.7.1, 2.7.2 and 2.7.3. We then present a new methodology for computing bounds on CVA contributions in section 2.7.4, motivated by the methodology of the Glasserman and Yang [1], and compare the results to those of the ordered scenario copula for computing CVA contributions in section 2.7.5.

2.7.1 CVA at Trade Level

As we described in sections 2.6.2 and 2.6.5, we can calculate the unilateral and bilateral CVAs by using equations (2.5.4) and (2.5.6). It is important to note that the CVA measured based on these approaches is calculated at the counterparty level and we can not determine the contributions of each trade to the calculated CVA directly.

In the absence of netting between trades, the portfolio-level CVA^U is the sum of the individual trade's *stand-alone* CVA^Us where a stand-alone CVA is defined as the CVA^U of each transaction. Let CVA_k^U denote the contribution of the k-th trade, $k = 1, \ldots, K$, in the portfolio. CVA contributions are additive when they can be summed up to the counterparty-level CVA^U, i.e.,

$$\text{CVA}^U = \sum_{k=1}^{K} \text{CVA}_k^U \tag{2.7.1}$$

In practice, under netting and margin agreements, the stand-alone CVA of the individual trades will no longer add up to the total portfolio CVA^U.

Alternatively, one can use the *incremental* CVA *contribution* of a trade. Defined as the difference between the portfolio-level CVA^U with and without the trade, it is a commonly used tool for pricing counterparty risk when new trades with the counterparty are added to the portfolio. Incremental CVA suffers from non-additivity as well. The sum of the individual trade's CVA contributions will not add up to the portfolio's CVA^U.

To address this issue, Pykhtin and Rosen [3] propose the use of *marginal* CVA *contributions* with a given counterparty. This notion allows us to determine how much each trade contributes to the counterparty-level CVA. In the coming sections, we discuss the calculation of the *marginal* CVA *contributions*.

2.7.2 CVA in the Presence of a Netting Agreement and Expected Exposure Contributions

As discussed earlier, a counterparty credit risk, which is the risk that the counterparty defaults before the final settlement of a transaction's cash flows, will result in an economic loss for the bank if the counterparty portfolio has a positive economic value at the time of default. At the core of the calculation of the marginal CVA contributions is a new perspective on how we dissect the exposure of the bank to a counterparty at the time of the default of the counterparty. In what follows we discuss the counterparty exposure calculation and show how it relates to the above problem.

Recall that given a portfolio of K trades that a bank has with a counterparty C, by using equations (2.4.2) and (2.4.3) we defined $V^+(t)$, the bank's netted exposure to counterparty C at the default time $\tau_{\text{C}} = t$ and $V^+(t)$ is written as:

$$V^+(t) = \max\{V(t), 0\} \tag{2.7.2}$$

where $V(t)$ in equation 2.7.2 represents the sum of the value of the individual trades for a portfolio consisting of K trades, (namely $v_k(t)$, $k = 1, \ldots, K$), from the bank's perspective.

If there is no netting among these trades, the bank's exposure to the counterparty C is calculated as:

$$E(t) = \sum_{k=1}^{K} \max\{v_i(t), 0\} \tag{2.7.3}$$

Furthermore given a single netting agreement, the netted exposure for a counterparty portfolio becomes:

$$E(t) = \max\{V(t), 0\} \tag{2.7.4}$$

In the following section, we assume that the counterparty C has posted no collateral.

As mentioned earlier, both the stand-alone CVA and the incremental CVA contributions lack additivity and we are interested in a framework for calculating additive contributions of each trade to the counterparty-level CVA. To address these issues, Pykhtin and Rosen [3] proposed an additive CVA contribution framework which is described in the next section.

2.7.3 CVA Contributions

Let CVA_k^U denote the contribution of the k-th trade ($k = 1, \ldots, K$) to the counterparty level CVA in the portfolio. To calculate each of these CVA contributions we start by calculating contributions of each trade to the portfolio's conditional expected exposure, $EE^{U+}(t)$. Obtaining additive CVA contributions reduces to obtaining conditional expected exposure contributions that are additive, i.e., conditional expected exposure contributions that sum up to the portfolio conditional discounted EE^+. Recall that:

$$\text{CVA}^U = (1 - R_\text{C}) \int_0^T EE^{U+}(t) dF_\text{C}(t)$$

We can consider CVA^U to be written as:

$$\text{CVA}^U = (1 - R_\text{C}) \, \text{E} \left(D(\tau_\text{C}) \max\{\sum_{k=1}^{K} w_i v_i(\tau_\text{C}), 0\} \mathbf{1}\{V^+(\tau_\text{C}) > 0\} \right) \tag{2.7.5}$$

where $w = (w_1, \ldots, w_K)$ represents the weight of the k-th trade in the bank's portfolio with counterparty C, $D(t)$ is the discounting factor at time t and $\mathbf{1}\{V^+(\tau_\text{C}) > 0\}$ indicates that there is a single netting agreement in place in the portfolio of the bank for the counterparty C. It can be easily seen that if we let $\text{CVA}^U = f(w)$, then for $\lambda \geqslant 0$ we have:

$$\begin{aligned} f(\lambda w) &= (1 - R_\text{C}) \, \text{E} \left(D(\tau_\text{C}) \max\{\sum_{k=1}^{K} \lambda w_i v_i(\tau_\text{C}), 0\} \mathbf{1}\{V^+(\tau_\text{C}) > 0\} \right) \\ &= \lambda (1 - R_\text{C}) \, \text{E} \left(D(\tau_\text{C}) \max\{\sum_{k=1}^{K} w_i v_i(\tau_\text{C}), 0\} \mathbf{1}\{V^+(\tau_\text{C}) > 0\} \right) \\ &= \lambda f(w) \end{aligned}$$

implying that $f(\cdot)$ is positively homogenous of degree 1. By applying Euler's theorem to (2.7.5) we have:

$$w_k \frac{\partial f}{\partial w_k} = w_k(1-R_{\mathrm{C}})\,\mathrm{E}\left(D(\tau_{\mathrm{C}})v_k(\tau_{\mathrm{C}})\mathbf{1}\{V^+(\tau_{\mathrm{C}})>0\}\right)$$

Let $w=1$. Then we have:

$$\mathrm{CVA}_k^U = (1-R_{\mathrm{C}})\,\mathrm{E}\left(D(\tau_{\mathrm{C}})v_k(\tau_{\mathrm{C}})\mathbf{1}\{V^+(\tau_{\mathrm{C}})>0\}\right) \tag{2.7.6}$$

or equivalently:

$$\mathrm{CVA}_k^U = (1-R_{\mathrm{C}})\int_0^T EE_k^{U+}(t)dF_{\mathrm{C}}(t), \qquad k=1,\ldots,K \tag{2.7.7}$$

where $EE_k^{U+}(t)$ represents the risk-neutral discounted expected exposure contribution of the k-th trade at time t

$$EE_k^{U+}(t) = E[D(t)v_k^+(t)\mathbf{1}\{V^+(\tau_{\mathrm{C}})>0\}|\tau_{\mathrm{C}}=t], \quad k=1,\ldots,K \tag{2.7.8}$$

and

$$v_k^+(t) = \max\{v_k(t),0\} \tag{2.7.9}$$

As before and given a market scenario y_j, $j=1,\ldots,M$, equation (2.7.9) can be further discretized to calculate the positive exposure at trade level under each market scenario:

$$\begin{aligned} v_k^+(t_i,y_j) &= \max\{v_k(t_i,y_j),0\} \\ &\quad i=1,\ldots,N+1, \quad j=1,\ldots,M. \end{aligned} \tag{2.7.10}$$

Note that in equation (2.7.6) the indicator function, given in $\mathbf{1}\{V^+(\tau_{\mathrm{C}})>0\}$, states that there is a single netting agreement in place for the counterparty C.

2.7.4 Bounds on CVA Contributions with Netting

Our goal is to find the upper bound for CVA_k^U, the k-th trade CVA contribution (k=1,. . . ,K),

$$\sup_{F_{\mathrm{C}Y}\in\mathfrak{F}(F_{\mathrm{C}},F_Y)} \mathrm{CVA}_k^U \tag{2.7.11}$$

and respectively the lower bound for the CVA_k^U

$$\inf_{F_{\mathrm{C}Y}\in\mathfrak{F}(F_{\mathrm{C}},F_Y)} \mathrm{CVA}_k^U \tag{2.7.12}$$

where CVA_k^U is given in equation (2.7.6) and as before in both equations (2.7.11) and (2.7.12) $\mathfrak{F}(F_{\mathrm{C}},F_Y)$ is the *Fréchet class* of all possible joint distributions of (τ_{C},Y) matching the given marginal distributions of the default time F_{C}, and market factors F_Y. For notational simplicity in what follows we will drop the superscript U from CVA_k^U.

Note that the main assumption for finding CVA contributions in section 2.7.2 is that

we have a single netting set. In practice, calculation of EE contributions and allocation of the portfolio-level EE for collateralized counterparties are more complex and have been discussed in detail in Pykhtin and Rosen [3].

2.7.5 Numerical Algorithm for Computing Bounds on CVA Contributions

Problems (2.7.11) and (2.7.12) for computing the upper bound $\overline{\text{CVA}}_k$ and lower bound $\underline{\text{CVA}}_k$ on CVA_k, $k = 1, \ldots, K$ can be recast as the following linear programming problems:

$$\begin{aligned} \max \sum_{i=1}^{N+1} \sum_{j=1}^{M} l_{ij} \overline{\vartheta}_{ij} \qquad & (2.7.13) \\ \sum_{i=1}^{N} \overline{\vartheta}_{ij} = p_j, \quad j = 1, \ldots, M \\ \sum_{j=1}^{M} \overline{\vartheta}_{ij} = q_i, \quad i = 1, \ldots, N+1 \\ \overline{\vartheta}_{ij} \geqslant 0, \quad i = 1, \ldots, N+1, \quad j = 1, \ldots, M \end{aligned}$$

and

$$\begin{aligned} \min \sum_{i=1}^{N+1} \sum_{j=1}^{M} l_{ij} \underline{\vartheta}_{ij} \qquad & (2.7.14) \\ \sum_{i=1}^{N} \underline{\vartheta}_{ij} = p_j, \quad j = 1, \ldots, M \\ \sum_{j=1}^{M} \underline{\vartheta}_{ij} = q_i, \quad i = 1, \ldots, N+1 \\ \underline{\vartheta}_{ij} \geqslant 0, \quad i = 1, \ldots, N+1, \quad j = 1, \ldots, M \end{aligned}$$

where in both linear programming problems (2.7.13) and (2.7.14), l_{ij} is defined as:

$$\begin{aligned} l_{ij} = \frac{1}{2}(1 - R_{\text{C}}) \left(D(t_i) v_k^+(t_i, y_j) + D(t_{i+1}) v_k^+(t_{i+1}, y_j) \right) \cdot \mathbf{1}\{V^+(t_i, y_j) + V^+(t_{i+1}, y_j) > 0\} \\ i = 1, \ldots, N, \quad j = 1, \ldots, M, \quad k = 1, \ldots, K \\ l_{ij} = \frac{1}{2}(1 - R_{\text{C}}) \left(D(t_i) v_k^+(t_i, y_j) \right) \cdot \mathbf{1}\{V^+(t_i, y_j) > 0\} \\ i = N+1, \quad j = 1, \ldots, M, \quad k = 1, \ldots, K \qquad (2.7.15) \end{aligned}$$

Note that when solving the linear programming problems (2.7.13) and (2.7.14), l_{ij} are calculated for a fixed trade in the portfolio (corresponding to a fixed k, $k = 1, \ldots, K$ in equation (2.7.5)). Moreover, the indicators are computed by using $\mathbf{1}\{V^+(t_i, y_j) + V^+(t_{i+1}, y_j) > 0\}$ for $i = 1, \ldots, N, \quad j = 1, \ldots, M$, and $\mathbf{1}\{V^+(t_i, y_j) > 0\}$ for $i = N+1, \quad j = 1, \ldots, M$, stating that there is a single netting agreement in place for

counterparty C. A numerical example on how these indicators are calculated is provided shortly.

To demonstrate the above methodology, consider the exposure dynamics within the portfolio that the bank has with a single counterparty C. Let the position-level exposure of the k-th trade (k=1,. . . ,K) of counterparty C in the bank's portfolio be driven by an Ornstein-Uhlenbeck process $X_k(t)$. The portfolio consists of three trades and the respective parameters of the O-U process for each trade are listed in table 2.2.

$$dX_k(t) = \kappa(\mu_k - X_k(t))dt + \sigma_k dW_k(t), \qquad k = 1, 2, 3 \tag{2.7.16}$$

where $W_k, k = 1, \ldots, K$ are independent Wiener processes.

Parameter	**Trade 1**	**Trade 2**	**Trade 3**
κ	1	1.2	0.9
μ	0	0	0
σ	20%	25%	16%

TABLE 2.2: *Ornstein-Uhlenbeck exposure model parameters for the three trades of the counterparty* C *in the banks's portfolio.*

For the sake of simplicity, we assume that there is a single netting agrement for the three trades for this counterparty and that these are the only trades that the bank has in its portfolio with counterparty C. Assume further that the counterparty C has posted no collateral. An upper (lower) bound on the CVA contribution is found through the following algorithm:

Algorithm 2.7.5: Bounds on Unilateral CVA Contributions Algorithm

---

1. Simulate M market scenarios to generate the exposures between the counterparty and the bank for the k-th trade in the portfolio according to the process:

   $$dX_k(t) = \kappa(\mu_k - X_k(t))dt + \sigma_k dW_k(t), \qquad k = 1, \ldots, K$$

   where $W_k, k = 1, \ldots, K$ are independent Wiener processes.

2. Construct discretized marginal distributions of the market scenarios, p_j and the default time, q_i using:

   $$\begin{aligned} q_i &= F_{\mathrm{C}}(t_{i+1}; \lambda) - F_{\mathrm{C}}(t_i; \lambda), \qquad i = 1, \ldots, N \\ q_i &= 1 - F_{\mathrm{C}}(t_i; \lambda), \qquad i = N + 1 \\ p_j &= \frac{1}{M}, \qquad j = 1, \ldots, M \end{aligned}$$

3. For all market scenarios y_j and a time step t_i calculate $v_k^+(t_i, y_j) = \max\{v_k(t_i, y_j), 0\}$ and determine:

$$\begin{aligned} l_{ij} &= \frac{1}{2}(1 - R_{\mathrm{C}}) \left(D(t_i) v_k^+(t_i, y_j) + D(t_{i+1}) v_k^+(t_{i+1}, y_j) \right) \\ &\qquad \cdot \mathbf{1}\{V^+(t_i, y_j) + V^+(t_{i+1}, y_j) > 0\} \\ &\qquad\qquad i = 1, \ldots, N, \quad j = 1, \ldots, M, \quad k = 1, \ldots, K \\ l_{ij} &= \frac{1}{2}(1 - R_{\mathrm{C}}) \left(D(t_i) v_k^+(t_i, y_j) \right) \cdot \mathbf{1}\{V^+(t_i, y_j) > 0\} \\ &\qquad\qquad i = N + 1, \quad j = 1, \ldots, M, \quad k = 1, \ldots, K \end{aligned} \tag{2.7.17}$$

4. Find the optimal coupling $\overline{\vartheta}_{ij}$ and $\underline{\vartheta}_{ij}$ $(i = 1, \ldots, N+1, \quad j = 1, \ldots, M)$ of the exposure scenarios and default times for computing the upper-bound and lower-bound on CVA_k, $k = 1, \ldots, K$, by solving the linear programming problems (2.7.13) and (2.7.14) respectively,

5. Calculate the upper bound and lower bound for the CVA_k as:

$$\overline{\mathrm{CVA}}_k = \sum_{i=1}^{N+1} \sum_{j=1}^{M} \overline{\vartheta}_{ij} l_{ij}, \qquad \underline{\mathrm{CVA}}_k = \sum_{i=1}^{N+1} \sum_{j=1}^{M} \underline{\vartheta}_{ij} l_{ij}, \quad k = 1, \ldots, K \tag{2.7.18}$$

Assume further that $T = 5$ years is the longest maturity of the contracts in the portfolio and the counterparty's default follows an exponential distribution.

To assess the performance of algorithm 2.7.5, we start by simulating $M = 10{,}000$ market scenarios, generating the exposures for each of the three trades using the O-U process described in (2.7.16). The respective parameters of the O-U process for each trade are listed in table 2.2. A constant $R_{\mathrm{C}} = 0.3$ and risk-free rate of $r_0 = 5\%$ (for generating the discount factors by continuous compounding) are used to calculate the coefficients of the linear programming problems in the algorithm 2.7.5.

The simulation is conducted for a $T = 5$ year time horizon, the longest maturity of a trade in the portfolio. Each exposure scenario for each trade is simulated by using daily time steps, i.e., each scenario consists of the exposures on 1250 trading days based on an Ornstein-Uhlenbeck process. The marginal distribution of the market scenarios, p_j, is defined as $1/M$:

$$P(Y = y_j) = p_j = \frac{1}{M}, \quad j = 1, \ldots, M \tag{2.7.19}$$

Similarly, we define q_i's, which discretize the default time distribution, in both linear programming problems (2.7.13) and (2.7.14) to be:

$$q_i = F_{\mathrm{C}}(t_{i+1}; \lambda) - F_{\mathrm{C}}(t_i; \lambda) \quad i = 1, \ldots, N \tag{2.7.20}$$

where $F_{\mathrm{C}}(t; \lambda)$ represents the cumulative distribution function associated with default times which are $\exp(\lambda)$-distributed. Note that q_{N+1}, which corresponds to a *no default event* becomes $1 - F_{\mathrm{C}}(T; \lambda)$.

trade (k)	λ_C	T	M	N+1	run time (in seconds)	sum of primal residuals	sum of dual residuals	duality gap	solution status code
1	0.5	5	10000	1251	544.5535	6.944e-07	6.2496e-07	6.7144e-09	1
	1	5	10000	1251	467.678	2.5678e-07	2.3111e-07	8.3717e-09	1
	1.5	5	10000	1251	429.5602	9.7586e-09	8.7828e-09	9.715e-09	1
	2	5	10000	1251	514.7261	5.3228e-07	4.7905e-07	5.6933e-10	1
	2.5	5	10000	1251	489.2148	2.7939e-07	2.5145e-07	4.5032e-09	1
	3	5	10000	1251	465.2151	9.4623e-07	8.5161e-07	5.8247e-09	1
	3.5	5	10000	1251	460.183	9.0644e-07	8.158e-07	6.8664e-09	1
	4	5	10000	1251	547.227	3.9268e-07	3.5342e-07	7.1943e-09	1
	4.5	5	10000	1251	436.8489	2.4855e-08	2.237e-08	6.5004e-09	1
2	0.5	5	10000	1251	447.3958	1.1704e-07	1.0533e-07	3.9813e-09	1
	1	5	10000	1251	525.6425	8.1468e-07	7.3321e-07	5.1537e-09	1
	1.5	5	10000	1251	408.6204	3.2486e-07	2.9237e-07	6.5753e-09	1
	2	5	10000	1251	391.5595	2.4623e-07	2.2161e-07	9.5092e-09	1
	2.5	5	10000	1251	552.0856	3.4271e-07	3.0844e-07	7.2235e-09	1
	3	5	10000	1251	481.8613	3.7569e-07	3.3812e-07	4.0008e-09	1
	3.5	5	10000	1251	412.5224	5.4655e-07	4.919e-07	8.3187e-09	1
	4	5	10000	1251	436.1491	5.6192e-07	5.0573e-07	1.3434e-09	1
	4.5	5	10000	1251	523.782	3.9582e-07	3.5624e-07	6.0467e-10	1
3	0.5	5	10000	1251	424.3305	3.8639e-07	3.4775e-07	3.8993e-09	1
	1	5	10000	1251	495.6495	7.7555e-07	6.98e-07	5.909e-09	1
	1.5	5	10000	1251	423.7944	7.3427e-07	6.6084e-07	4.5938e-09	1
	2	5	10000	1251	549.4803	4.3028e-07	3.8725e-07	5.034e-10	1
	2.5	5	10000	1251	437.3544	6.9375e-07	6.2438e-07	2.2869e-09	1
	3	5	10000	1251	396.373	9.4521e-07	8.5069e-07	8.3419e-09	1
	3.5	5	10000	1251	553.1387	7.8423e-07	7.0581e-07	1.5645e-10	1
	4	5	10000	1251	443.9969	7.0557e-07	6.3501e-07	8.6371e-09	1
	4.5	5	10000	1251	525.4212	1.0933e-07	9.8401e-08	7.8069e-10	1

TABLE 2.3: *Run time, sum of primal residuals, sum of dual residuals, duality gap and solution status code for the 9 linear programming problems (2.7.14) associated with different average default rates given in* (2.7.21) *for finding the lower bound on* CVA *contributions.*

A similar approach to solving the linear programming problem (2.6.11) is adopted here and the algorithm for calculating the bounds on the CVA contributions is described in algorithm 2.7.5. While in calculating the worst-case CVA by using algorithm 2.6.3, we focus on the counterparty-level CVA, algorithm 2.7.5 calculates the upper and lower bounds on CVA contributions of the individual trades of counterparty C (namely $\overline{\text{CVA}}_k$ and $\underline{\text{CVA}}_k, k = 1, \ldots, K$) in the bank's portfolio.

$$\tau_C \sim \exp(\lambda_i), \quad \lambda_i \in \{0.5, 1, \ldots, 4.5\} \tag{2.7.21}$$

As before 9 instances of the problems (2.7.13) and (2.7.14), corresponding to average default rates, given in equation (2.7.21), are solved for each trade to calculate the upper and lower bound on their respective CVA contributions. That is, for each trade (with the respective exposure parameters listed in table 2.2), we simulate $M = 10{,}000$ market scenarios and generate the exposures using the O-U process described in (2.7.16); we then compute the netted exposures at portfolio level, $V^+(t_i, y_j)$, $i = 1, \ldots, N+1$, $j = 1, \ldots, M$. These netted exposures are then used in the indicator functions, given in step 3 of algorithm 2.7.5 to compute the coefficients of the linear programming problems (2.7.13) and (2.7.14) (using equations (2.7.17)).

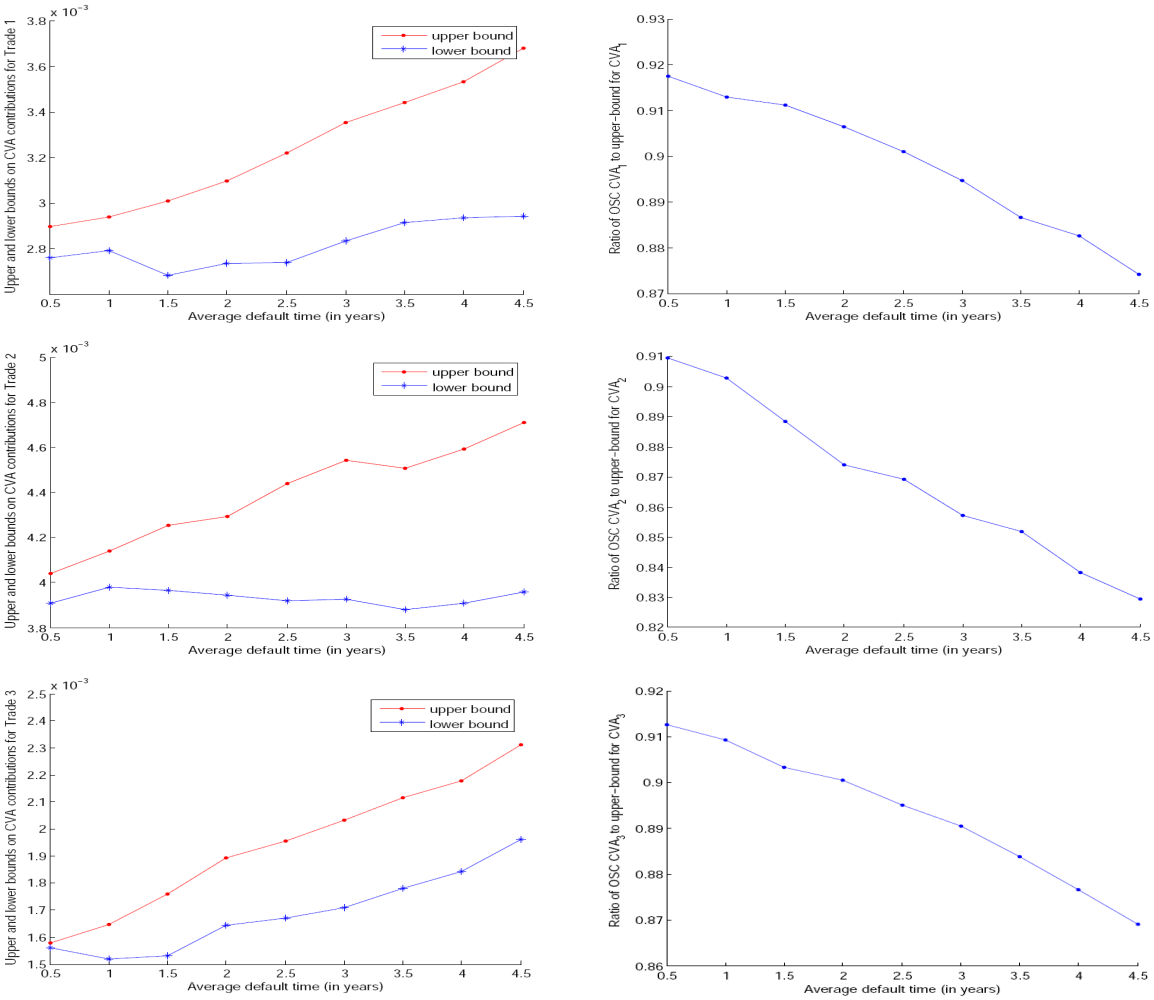


FIGURE 2.8: *Left: upper and lower bounds on* CVA *contributions calculated using algorithm 2.7.5. Right: ratio of the ordered scenario copula CVA contributions to upper-bound of* CVA_k *calculated using algorithm 2.7.5; exposure is modeled by an O-U process for each trade of counterparty* C *in the bank's portfolio; respective parameters are listed in table 2.2.*

trade (k)	λ_C	T	M	N+1	run time (in seconds)	sum of primal residuals	sum of dual residuals	duality gap	solution status code
1	0.5	5	10000	1251	495.2501	1.0802e-07	9.7215e-08	5.0513e-09	1
	1	5	10000	1251	478.0379	5.17e-07	4.653e-07	2.7142e-09	1
	1.5	5	10000	1251	436.4824	1.4316e-07	1.2884e-07	1.0075e-09	1
	2	5	10000	1251	396.7779	5.5937e-07	5.0343e-07	5.0785e-09	1
	2.5	5	10000	1251	541.739	4.5796e-09	4.1217e-09	5.8561e-09	1
	3	5	10000	1251	532.0129	7.6668e-07	6.9001e-07	7.6289e-09	1
	3.5	5	10000	1251	393.6132	8.4871e-07	7.6384e-07	8.2963e-10	1
	4	5	10000	1251	470.2302	9.1682e-07	8.2514e-07	6.616e-09	1
	4.5	5	10000	1251	570.8052	9.8697e-07	8.8827e-07	5.1698e-09	1
2	0.5	5	10000	1251	466.1351	6.8357e-08	6.1521e-08	6.8004e-09	1
	1	5	10000	1251	420.1114	4.3633e-07	3.9269e-07	7.0595e-09	1
	1.5	5	10000	1251	389.8942	1.7385e-07	1.5647e-07	6.4513e-09	1
	2	5	10000	1251	386.062	2.6107e-08	2.3496e-08	5.5231e-09	1
	2.5	5	10000	1251	457.8816	9.5468e-07	8.5921e-07	2.1811e-09	1
	3	5	10000	1251	528.2671	4.306e-07	3.8754e-07	7.7237e-09	1
	3.5	5	10000	1251	517.6011	9.6156e-07	8.654e-07	2.2803e-09	1
	4	5	10000	1251	532.378	7.6241e-07	6.8617e-07	3.7086e-09	1
	4.5	5	10000	1251	506.4873	7.3487e-09	6.6138e-09	8.9093e-09	1
3	0.5	5	10000	1251	435.7602	4.0386e-07	3.6347e-07	9.3566e-09	1
	1	5	10000	1251	570.4355	5.4857e-07	4.9371e-07	8.1871e-09	1
	1.5	5	10000	1251	483.3179	4.8739e-08	4.3865e-08	7.2826e-09	1
	2	5	10000	1251	426.7574	5.5273e-07	4.9746e-07	1.7581e-09	1
	2.5	5	10000	1251	551.6497	2.7481e-07	2.4733e-07	3.6037e-09	1
	3	5	10000	1251	462.0698	2.415e-07	2.1735e-07	1.8879e-09	1
	3.5	5	10000	1251	446.3467	2.4315e-07	2.1883e-07	1.1984e-11	1
	4	5	10000	1251	498.7713	1.5416e-07	1.3874e-07	3.1642e-09	1
	4.5	5	10000	1251	447.8073	9.5642e-07	8.6077e-07	6.9962e-09	1

TABLE 2.4: *Run time, sum of primal residuals, sum of dual residuals, duality gap and solution status code for the 9 linear programming problems (2.7.13) associated with different average default rates given in* (2.7.21) *for finding the upper bound on* CVA *contributions.*

These results are used to generate the three figures on the left in figure 2.8 and show the bounds on the CVA contribution corresponding to each of the three trades of the counterparty C with parameters listed in table 2.2.

Next we can calculate the CVA contributions corresponding to each $\lambda_i \in \{0.5, 1, \ldots, 4.5\}$, using the ordered scenario copula methodology described in section 2.7.2. The conservative market-credit correlation of $\overline{\rho} = 1$ is chosen in the ordered scenario copula and the exposure scenarios for each trade are sorted in order of the increasing time-averaged total portfolio exposure as described in section 2.6.1.

The results generated from the ordered scenario copula methodology are then compared to those that are generated by algorithm 2.7.5 when computing the upper bound on CVA_k, namely $\overline{\text{CVA}_k}$, by solving the linear programming problem (2.7.13). That is, for a fixed average default rate described in (2.7.21) we calculate the ratios

$$\frac{\text{CVA}_{OSC_k}}{\overline{\text{CVA}_k}}, \quad k = 1, \ldots, 3$$

for each of the three trades of the counterparty C with the respective exposure parameters given in table 2.2. The graph of the resulting ratios, plotted against the average default

trade (k)	λ_C	T	M	N+1	$\overline{\rho}$ 0.05	0.15	0.25	0.35	0.45
1	0.5	5	10000	1251	57.58%	60.95%	64.74%	69.11%	73.77%
	1	5	10000	1251	57.15%	60.39%	63.96%	68.20%	72.52%
	1.5	5	10000	1251	56.73%	59.84%	63.20%	67.32%	71.29%
	2	5	10000	1251	56.45%	59.48%	62.75%	66.73%	70.49%
	2.5	5	10000	1251	56.03%	58.94%	61.96%	65.87%	69.29%
	3	5	10000	1251	55.69%	58.49%	61.33%	65.13%	68.28%
	3.5	5	10000	1251	55.24%	57.91%	60.54%	64.21%	67.05%
	4	5	10000	1251	54.82%	57.37%	59.77%	63.32%	65.78%
	4.5	5	10000	1251	54.41%	56.83%	59.03%	62.45%	64.58%
2	0.5	5	10000	1251	56.96%	59.66%	62.81%	66.56%	70.56%
	1	5	10000	1251	56.48%	59.13%	62.04%	65.66%	69.33%
	1.5	5	10000	1251	55.98%	58.46%	61.15%	64.63%	67.89%
	2	5	10000	1251	55.51%	57.84%	60.29%	63.62%	66.51%
	2.5	5	10000	1251	55.34%	57.63%	60.81%	63.28%	66.03%
	3	5	10000	1251	55.17%	57.41%	59.69%	62.93%	65.55%
	3.5	5	10000	1251	54.72%	56.81%	58.87%	61.97%	64.22%
	4	5	10000	1251	54.27%	56.23%	58.06%	61.02%	62.92%
	4.5	5	10000	1251	53.86%	55.71%	57.33%	60.17%	61.74%
3	0.5	5	10000	1251	57.45%	60.71%	64.39%	68.65%	73.19%
	1	5	10000	1251	57.08%	60.24%	63.73%	67.89%	72.13%
	1.5	5	10000	1251	56.64%	59.66%	62.93%	66.95%	70.85%
	2	5	10000	1251	56.24%	59.14%	62.21%	66.11%	69.68%
	2.5	5	10000	1251	55.82%	58.59%	61.45%	65.23%	68.46%
	3	5	10000	1251	55.39%	58.04%	60.68%	64.33%	67.22%
	3.5	5	10000	1251	54.89%	57.39%	59.78%	63.28%	65.77%
	4	5	10000	1251	54.81%	57.27%	59.63%	63.16%	65.52%
	4.5	5	10000	1251	54.34%	56.68%	58.81%	62.13%	64.19%

TABLE 2.5: *The ratio of the* CVA *contributions generated by dividing the* CVA *contributions of the ordered scenario copula methodology, with the market-credit correlation* $\overline{\rho} \in \{0.05, 0.15, 0.25, 0.35, 0.45\}$*, to the* $\overline{\text{CVA}}_k$ *calculated using algorithm 2.7.5 for the three trades of the counterparty* C *with the respective parameters listed in table 2.2.*

rate in years, is shown on the right hand side in figure 2.8.

As can be seen in figure 2.8, the upper-bound for the CVA contributions calculated by algorithm 2.7.5 is consistently higher than the contributions calculated with the ordered scenario copula methodology by using the market-credit correlation $\overline{\rho} = 1$, for various average default times and across all trades. Furthermore it is important to note that as the average default times (λ_C) increase, the difference in the ratios is increasing for all three trades.

It has been noted in Rosen and Saunders [2] that in many practical applications, the market-credit correlation $\overline{\rho}$ is chosen between 0.2 and 0.3. Therefore, while in generating the results shown in figure 2.8 the market-credit correlation of $\overline{\rho} = 1$ is chosen to account for the highest level of the correlation of the exposures and defaults, we conduct an additional analysis by allowing $\overline{\rho} \in \{0.05, 0.15, 0.25, 0.35, 0.45\}$.

For each market-credit correlation $\overline{\rho} \in \{0.05, 0.15, 0.25, 0.35, 0.45\}$ and for each of the three trades of the counterparty C, we have calculated the ratio of the ordered scenario

copula CVA_{OSC_k} to the upper bound on CVA contribution, namely $\overline{\text{CVA}}_k$, generated by algorithm 2.7.5. The results are shown across various levels of market-credit correlation in table 2.5.

Moreover, the ratios of the CVA contributions (shown in table 2.5) calculated by dividing the CVA contributions generated by the ordered scenario copula methodology to the upper bound of the CVA contributions calculated by using algorithm 2.7.5 when the market-credit correlation $\overline{\rho} \in \{0.05, 0.15, 0.25, 0.35, 0.45\}$ indicate that there is a significant difference between the CVA contributions computed using these methodologies. For example this difference for the first trade in the portfolio varies between 45.59% (corresponding to $\lambda_{\text{C}} = 4.5$ and $\overline{\rho} = 0.05$) and 26.23% (corresponding to $\lambda_{\text{C}} = 0.5$ and $\overline{\rho} = 0.45$) and between 46.14% (corresponding to $\lambda_{\text{C}} = 4.5$ and $\overline{\rho} = 0.05$) and 29.44% (corresponding to $\lambda_{\text{C}} = 0.5$ and $\overline{\rho} = 0.45$) for the second trade.

It is important to note that even by choosing $\overline{\rho} = 1$, leading to the highest level of the correlation of the exposures and defaults and thereby largest losses in the ordered scenario copula, the upper bound of the CVA contributions calculated by using algorithm 2.7.5 is significantly higher than that of the ordered scenario copula. Moreover, when we choose the market-credit correlation $\overline{\rho} \in \{0.05, 0.15, 0.25, 0.35, 0.45\}$, the difference between the upper bound of the CVA contributions calculated by using algorithm 2.7.5 and the ordered scenario copula is larger compared to the $\overline{\rho} = 1$.

Chapter 3

Wrong-Way Risk and Counterparty Credit Risk

3.1 Overview

An important problem that the global financial crisis of 2008 revealed relates to how well financial institutions are able to assess counterparty credit risk in their portfolios. In simple terms, Counterparty Credit Risk (CCR) is defined as the risk of default before the final settlement of a transaction's cash flows.

The internal rating-based framework (see Basel Committee on Banking Supervision (BCBS) [41], Basel Committee on Banking Supervision (BCBS) [42] and Basel Committee on Banking Supervision (BCBS) [42]) that banks use to determine the minimum capital requirement for counterparty credit risk of their derivatives portfolios is based on four major inputs: Probability of Default, Exposure at Default, Loss Given Default and finally Maturity. The aforementioned parameters are the main drivers of the complexity of a portfolio's CCR. In order to accurately manage and measure a portfolio's CCR and identify the embedded model risk, we face the challenging task of capturing the stochastic nature of counterparty exposures driven by market factors. In addition the dependence structures between exposures, between counterparty defaults and between defaults and exposures (wrong- or right-way risk) are critical. Hence in addition to both modeling and computational complexity of evaluating CCR capital requirements, determining and validating market-credit correlations is an important part of the quantitative modeling process. Another challenge that the task of estimating market-credit correlations poses is that since these correlations can vary drastically with the portfolio composition and market conditions, assessing their probable impact on the final capital estimate can be a difficult task.

3.2 Outline and Contributions

In chapters 3 and 4 we propose a new approach for computing bounds on Conditional Value at Risk (CVaR) of a portfolio. The remainder of chapter 3 and chapter 4 are organized as follows.

After providing an overview of the existing literature on model uncertainty and counterparty credit risk in section 3.3, we discuss the existing frameworks for calculating the capital charge for counterparty credit risk under the Basel Accord in section 3.4.

One of the main issues that quantitative modelers face in this regard is estimating the joint distribution of credit risk factors and market risk factors. In section 3.5 we formalize this problem and present the worst-case risk measure problem.

Sections 3.6 and 3.7 illustrate the connection of problem (3.5.1) and our choice of the risk measure, CVaR, in detail. We further formulate the problem of computing bounds on CVaR as a linear optimization problem in section 3.7.

In chapter 4 we demonstrate the application of the worst-case copula methodology under the Basel credit model. More specifically after describing the default dynamics of the counterparties, the loss functions and their connection to the Basel credit model in section 4.1, we present the linear programming formulation that is used for finding bounds on CVaR in this framework.

Section 4.2 demonstrates an application of the worst-case copula methodology in counterparty credit risk management and provides numerical examples, comparing the performance of this approach with the *ordered scenario copula* methodology by using a real-world portfolio of OTC derivatives.

3.3 Literature Review

Capturing the risk of exotic options is an interesting topic for market practitioners. Given a set of liquid prices for such options, we can calibrate various parameters that these models depend upon. An undesirable by-product of such an approach is that one gets a wide range of prices for a specific exotic option when we use various models for pricing and calibrate them to the same market data. If the lower and upper bounds that we get from various calibration processes for these models based on the same market data are tight, we can determine if there are any arbitrage opportunities in the market. This problem has been studied for instance in Bertsimas and Popescu [43], Hobson et al. [44], Hobson et al. [45], Laurence and Wang [46], Laurence and Wang [47] and Chen et al. [48] for an exotic option written on multiple-assets $(S_1, \ldots, S_T)$ observed at the same time T. If we focus only on models with fixed marginals, $(\Psi(S_1^T), \ldots, \Psi(S_k^T))$, our main tool for searching lower and upper bounds is an infinite-dimensional linear programming problem. Beiglböck et al. [32] propose using infinite-dimensional linear programming methods. In comparison to previously cited literature, the authors argue that since requiring that the asset price S_t be a discrete time martingale can be more restrictive, achieving tighter bounds becomes more feasible.

The approach closest to the one we take in this chapter is that of Beiglböck et al. [32], in which the marginals $(\Psi(S_1^T), \ldots, \Psi(S_k^T))$ are assumed to be given, and an infinite-dimensional linear programming technique is employed to derive price bounds. There is also a large literature on deriving bounds on VaR for sums of random variables with given marginals. See Makarov [49], Williamson and Downs [50], Denuit et al. [51], Firpo and Ridder [52] and Embrechts et al. [53] for the theoretical treatment of this problem and

Embrechts and Puccetti [54], Puccetti and Rüschendorf [55] and Puccetti and Rüschendorf [7] for the numerical methods developed in solving this problem. As discussed earlier Glasserman and Yang [1] have studied the problem of bounding wrong-way risk, given the marginal distributions of market and credit risk factors in a similar setting.

The problems considered in this chapter can be characterized by three important aspects; (i) we use an alternative risk measure (CVaR), (ii) we are provided with multivariate (non-overlapping) marginal distributions, and (iii) we have losses that are a non-linear function of the underlying risk factors.

Talay and Zhang [56] approach the issue of model uncertainty as a model risk control problem. They adopt a *worst case* stochastic game approach and look at this problem in a trader-versus-market framework. In this framework the trader's goal is to minimize the risk of his position, while the market aims at doing the opposite. The authors point out that while one can never assume that the trader knows the exact model of the market, they can assume that the correct model of the market belongs to a wide class of models. By defining a cost function which describes the risk faced by the trader under such conditions, the trader is able to choose trading strategies from a set of admissible strategies in order to decrease the risk of his/her position. On the other hand, the market is assumed to behave systematically against the interest of the trader and maximizes the risk of his/her position. Hence the model risk control problem can be viewed as a Trader-versus-Market zero-sum stochastic differential game problem.

Another approach to static risk measures with model uncertainty can be seen in the work of Kervarec [57]; model uncertainty in this work is specified by a non-dominated weakly compact set of probability measures. J. and M. [58] studied *regular* convex risk measures on $\mathcal{C}_b(\Omega)$, the set of continuous bounded functions on a Polish space Ω. They point out that *regularity* in their work is equivalent to continuity with respect to a certain capacity c, which, if one considers the completion $\mathcal{L}^1(c)$ of $\mathcal{C}_b(\Omega)$ with respect to this capacity c would be equivalent to studying convex risk measures on the Banach space $L^1(c)$. The main result of such an approach is that for every regular convex risk measure on $\mathcal{C}_b(\Omega)$, there is a unique equivalence class of probability measures characterizing the risk-less non-positive elements of $\mathcal{C}_b(\Omega)$.

The following provides a brief summary of the notation that we have used in chapters 3 and 4:

PD:	Probability of default of a given counterparty
EAD:	Exposure at default matrix, that contains all market scenarios
CVaR:	Conditional Value at Risk
CWI:	Creditworthiness index of a given counterparty

3.4 Wrong-Way Risk and Counterparty Credit Risk

We start this chapter by looking at different aspects of the existing framework under the Basel Accord for calculating capital requirements of counterparty credit risk on a typical

derivatives portfolio. Upon entering an Over The Counter (OTC) contract, the entities signing the contract are exposed to the risk of the default of other parties during the lifetime of the contract.

Banks use an internal rating-based method for calculating capital requirements for the CCR of derivatives portfolios. The risk-weight formula that they use for calculating counterparty capital charges uses four quantitative inputs:

- Probability of Default (PD),
- Exposure at Default (EAD),
- Loss Given Default (LGD) and
- Maturity (MTY).

Potential Future Exposures, or PFEs, are also an important element in the process of calculating counterparty credit risk capital through their influence on EAD. PFEs tell us how exposures evolve during the lifetime of the contract. More specifically, since the value of derivatives can change substantially over time according to the market conditions and the age of the portfolio, PFEs can act as a good indicator of such future changes in the value of the contract.

Banks use internal models for determining probabilities of default and losses given default. Furthermore the Basel accord allows banks to use internal models for determining EADs and Maturity. In order to calculate EAD and Maturity, we should find the expected positive exposure (EPE) (defined as the average of the PFE over time and scenarios; see section 4.1 for more details), the effective EPE, the effective maturity and the alpha multiplier.

In the Basel II formula, the minimum capital requirement for a given obligor j is defined as:

$$\text{Capital(j)} = \text{EAD}_j \cdot \text{LGD}_j \cdot \left[\Phi\left(\frac{\Phi^{-1}(PD_j) + \sqrt{\rho_j}.\Phi^{-1}(0.999)}{\sqrt{1-\rho_j}}\right)\right] \cdot \frac{1 + (\text{MTY}_j - 2.5)b_j}{1 - 1.5b_j}$$

where $\text{EAD}_j, \text{LGD}_j$ and PD_j are calculated by the bank's internal rating system. The time horizon for calculating default probabilities is one year. The asset correlation ρ_j and maturity adjustment b_j are parameters that are specified in Basel accord. The confidence level that is used in the Accord is 99.9%. Although this confidence level may seem to be rather high, according to the Basel Committee on Banking Supervision (BCBS) [39], it aims at protecting against errors introduced in the banks' internal models for estimation of probability of default, loss given default and exposure at default, along with other model uncertainties.

A portfolio's alpha multiplier is defined as:

$$\alpha = \frac{\text{EC}^{\text{Total}}}{\text{EC}^{\text{EPE}}}$$

where $\mathrm{EC}^{\mathrm{Total}}$ denotes the economic capital (defined as the capital that should be invested in the company in order to limit the probability of default to a given confidence level over a given time horizon) for CCR based on a joint simulation of market and credit risk factors and $\mathrm{EC}^{\mathrm{EPE}}$ is the economic capital when counterparty exposures are deterministic and equal to the EPE.

In calculating the numerator of alpha a joint simulation of market and credit risk factors that incorporates the uncertainty of, and correlation between, counterparty exposures, as well as the correlation between exposures and defaults is required. This in turn captures the stochastic nature of the exposures and their correlations, the correlations of exposures and credit events and portfolio granularity. In contrast, in an ideal universe, for a very large infinitely granular portfolio, in which PFEs are constant and independent of each other and of default events, one can assume that exposures are deterministic and they are given by the EPEs. Calculating α tells us how far we are from such an ideal case; in other words, alpha acts as a tool to condition internal EPE estimates on a "bad state" of the economy and to adjust internal EPEs for:

- lack of granularity across portfolios,
- correlation between counterparty exposures and defaults,
- and uncertainty of counterparty exposures and correlation between them as well as the correlation between the counterparty exposures and LGDs.

A closer look at various aspects of the processes that we described for calculating capital requirements for CCR brings the role of market-credit correlation to prominence as it determines the co-dependence between exposures and defaults and plays an integral role in stress testing and sensitivity analysis in many practical applications. The challenge faced by researchers in this remains in having a conservative estimate of these market-credit correlations and a dependable process for validating them. Given the importance of capturing the right co-dependence structure of market risk factors and credit risk factors, answering the question *"What is the worst-case dependence structure?"* can provide us with a solid framework for bounding various risk statistics of a portfolio. The next section is devoted to formalizing this problem for finding the worst-case joint distribution of market risk factors and credit risk factors assuming that the respective marginal distributions of these factors are given. Later on we focus on a nontrivial counterparty credit risk example under this setting.

3.5 The Worst-Case Risk Measure Problem

Let Y and Z be two vectors of random risk factors respectively. We assume that the multi-dimensional marginals of Y and Z, denoted by $F_Y(y)$ and $F_Z(z)$ respectively, are known, but the joint distribution of (Y, Z) is unknown. (Note: in the context of the counterparty credit risk management discussed in the next section, Y and Z will be vectors

of market and systematic credit factors respectively). Portfolio losses are defined to be $L = L(Y, Z)$, where in general this function may be non-linear. We are interested in determining the joint distribution of (Y, Z) that maximizes a given risk measure ρ:

$$\max_{\mathfrak{F}(F_Y, F_Z)} \rho\left(L(Y, Z)\right) \tag{3.5.1}$$

where $\mathfrak{F}(F_Y, F_Z)$ is the *Fréchet class* of all possible joint distributions of (Y, Z) matching the previously defined marginal distributions F_Y and F_Z. More explicitly, for any joint distribution $F_{YZ} \in \mathfrak{F}(F_Y, F_Z)$ we have $\Pi_y\{F_{YZ}\} = F_Y$ and $\Pi_z\{F_{YZ}\} = F_Z$, where $\Pi_\cdot\{.\}$ denote the projections that take the joint distribution to its (multi-variate) marginals. [1]

3.6 Coherent Risk Measures

Assume Ω to be a pre-determined set of scenarios that we need for calculating the future loss of our portfolio. Using Ω we can calculate various uncertain future values of our portfolio; more specifically let $V : \Omega \to \mathbb{R}$ calculate the future value of our portfolio. Our goal is to define a number $\rho(V)$ that quantifies the risk of the portfolio.

The following axioms and definitions are from Artzner et al. [59]. In the following let χ to be the linear space of all functions $V : \Omega \to \mathbb{R}$.

Definition 3.6.1 (Risk Measure)**.** *The mapping $\rho : \chi \to \mathbb{R}$ is called a* risk measure *if it satisfies the following properties:*

- Finiteness: $\rho(0)$ *is finite*
- Monotonicity: *For all $X_1, X_2 \in \chi$, if $X_1 \leqslant X_2$ a.s., then $\rho(X_1) \leqslant \rho(X_2)$ and*
- Translation invariance: *For $m \in \mathbb{R}$ and $X_1 \in \chi$,* $\quad \rho(X_1 + m) = \rho(X_1) + m$

Monotonicity has a clear interpretation: portfolios which always lose more require more risk capital. *Translation invariance*, also called *Cash invariance* ensures that with adding risk-free amount $\$m$ of cash to our portfolio for meeting regulatory obligations, the capital requirement of our position is adjusted accordingly.

Definition 3.6.2 (Convex Risk Measure.)**.** *If a risk measure ρ satisfies*

- $\rho(\lambda X_1 + (1-\lambda)X_2) \leqslant \lambda\rho(X_1) + (1-\lambda)\rho(X_2), \quad$ *for* $\quad 0 \leqslant \lambda \leqslant 1$

then it is called a convex risk measure.

[1] While we are mainly interested in applications to counterparty credit risk, we note that bounds on instrument prices can be derived within the above formulation by taking the risk measure to be the expectation operator. An alternative application of this methodology for calculating worst-case credit value adjustment, proposed by Glasserman and Yang [1], is presented in chapter 2.

The intuitive idea underlying the axiom of convexity is based on the benefits of diversification. Suppose that an investor is given two alternative investment strategies leading to risk profiles X_1 and X_2. Then by investing only λ and $1-\lambda$ of his wealth in the strategies X_1 and X_2 respectively, his total risk should not increase.

Definition 3.6.3 (Coherent Risk Measure.)**.** *A convex risk measure is called a coherent risk measure if it satisfies*

- *Positive homogeneity: If* $\lambda \geqslant 0$*, then* $\rho(\lambda X_1) = \lambda\rho(X_1)$

A direct implication of positive homogeneity for a convex risk measure ρ is that it becomes *sub-additive*, that is

- Sub-additive: $X_1, X_2, \quad X_1 + X_2 \in \chi \Rightarrow \rho(X_1 + X_2) \leqslant \rho(X_1) + \rho(X_2)$

An important application of sub-additivity for practitioners is that it enables them to *decentralize* the total risk of a portfolio. More precisely, if we assign different risk limits for each asset class in our portfolio, then the total risk of this collection does not exceed the sum of the risks for each class.

3.7 VaR, CVaR and Coherence

It is well known that *VaR* is not a coherent risk measure, while its averaged value, *CVaR* complies with axioms of coherence (see Artzner et al. [59]). In this section we provide a further discussion on at VaR and CVaR.

Given a portfolio of risky assets and a fixed time horizon δ, let $F_L(l) = P(L \leqslant l)$ denote the loss distribution function (in simple terms, L is defined as the initial value of the portfolio minus its final value over the time horizon δ). VaR is a quantile of the loss distribution L. More formally, we assume that L is a random variable defined on a probability space $(\Omega, \mathcal{B}, F)$.

Definition 3.7.1 (VaR)**.** *Given a confidence level* α *and a loss random variable* L*, the* VaR at confidence level α *is defined as:*

$$\text{VaR}_\alpha(L) = \inf\{l \in \mathbb{R} : P(L \geqslant l) \leqslant 1-\alpha\} = \inf\{l \in \mathbb{R} : F_L(l) \geqslant \alpha\} \tag{3.7.1}$$

The time horizon of one year and confidence level of $\alpha = 0.999$ are the most common parameters used in practice. In spite of it being a popular risk measure in practice, VaR suffers from the lack of sub-additivity.

Note that while the above definition of $\text{VaR}_\alpha(L)$ provides us with a probabilistic statement, controlling the probability of losses, the size of such losses remains unknown. CVaR, also known as *tail VaR* or *Expected Shortfall* addresses sub-additivity issues, by taking into account what happens *beyond* a fixed confidence level α of the loss distribution. By using CVaR we can learn more about extreme behaviours of the loss distribution. In what follows we give formal definition of this risk measure and connect it with the worst-case copula problem.

Definition 3.7.2 (CVaR)**.** *For the confidence level α and the loss random variable L, the* CVaR at level α *is defined by*

$$\mathrm{CVaR}_\alpha(L) = \frac{1}{1-\alpha}\int_\alpha^1 \mathrm{VaR}_\xi(L)d\xi$$

The following theorem is from Schied [60] and provides an important characterization of CVaR.

Theorem 3.7.3. $\mathrm{CVaR}_\alpha(L)$ *can be represented as*

$$\mathrm{CVaR}_\alpha(L) = \sup_{G\in\mathcal{G}_\alpha} \mathrm{E}_G[L]$$

where $\mathcal{G}_\alpha$ is the set of all probability measures $G \ll F$ whose density dG/dF is F-a.s. bounded by $1/(1-\alpha)$.

In the above theorem, $G \ll F$ means G is absolutely continuous with respect to F, i.e. for any $\mathrm{B} \in \mathcal{B}$ that $F(\mathrm{B}) = 0$ we have $G(\mathrm{B}) = 0$. Applying the above result, with $\rho = \mathrm{CVaR}_\alpha$, the worst-case joint distribution problem stated in (3.5.1) can be conveniently reformulated as:

$$\sup_{F\in\mathfrak{F}(F_Y,F_Z),G\ll F} \mathrm{E}_G[L] \tag{3.7.2}$$

$$\frac{dG}{dF} \leqslant \frac{1}{1-\alpha} \quad a.s. \tag{3.7.3}$$

Note that the final constraint assumes explicitly that the corresponding density exists.

In many practical cases the marginal distributions will be discrete, either due to a modelling choice, or because they arise from the simulation of separate continuous models for Y and Z. In this case, the marginal distributions of the credit risk factors and market risk factors can be represented by $F_Z(Z = z_i) = q_i, i = 1,\ldots,N$ and $F_Y(Y = y_j) = p_j, j = 1,\ldots,M$ respectively. Any joint distribution of (Y, Z) is then specified by the quantities $F_{YZ}(Y = y_j, Z = z_i) = \vartheta_{ij}$, and the worst-case CVaR optimization problem above can be further simplified to:

$$\max_{\vartheta,\mu} \quad \frac{1}{1-\alpha}\sum_{i=1}^N\sum_{j=1}^M l_{ij}\cdot\mu_{ij} \tag{3.7.4}$$

$$\sum_{i=1}^N \vartheta_{ij} = p_j, \quad j = 1,\ldots,M$$

$$\sum_{j=1}^M \vartheta_{ij} = q_i, \quad i = 1,\ldots,N$$

$$\sum_{i=1}^N\sum_{j=1}^M \mu_{ij} = 1-\alpha, \tag{3.7.5}$$

$$0 \leqslant \mu_{ij} \leqslant \vartheta_{ij}, \quad i = 1,\ldots,N, \quad j = 1,\ldots,M \tag{3.7.6}$$

where l_{ij} represents the discretized loss function under the i-th credit scenario, $i = 1, \ldots, N$ and j-th market scenario, $j = 1, \ldots, M$. Section 4.1 provides a detailed description of how $l_{ij}, i = 1, \ldots, N$, $j = 1, \ldots, M$ is calculated when the linear programming problem (3.7.4) is solved.

The constraint (3.7.3) in problem (3.7.2) is (roughly) translated into two constraints (3.7.5) and (3.7.6) in problem (3.7.4); by dividing both sides of the (3.7.5) by $1 - \alpha$, (3.7.5) can be written as:

$$\sum_{i=1}^{N} \sum_{j=1}^{M} \frac{1}{1-\alpha} \mu_{ij} = 1$$

that is $\frac{1}{1-\alpha}\mu$ represents a probability mass function; similarly, (3.7.6) can be written as:

$$\left(\frac{1}{1-\alpha}\mu_{ij}\right)\left(\frac{1}{\vartheta_{ij}}\right) \leqslant \frac{1}{1-\alpha} \quad \text{as long as} \quad \vartheta_{ij} > 0$$

now it can be easily seen that there is a one-to-one correspondence between the latter inequality and (3.7.3).

Chapter 4

Worst-Case Conditional Value-at-Risk

4.1 Worst-Case Joint Distribution in the Basel Credit Model

A nontrivial application of the aforementioned worst-case copula is given in the form of a Counterparty Credit Risk problem. We start this section by describing the default behaviour of a single counterparty; we then proceed to incorporate this behaviour into the portfolio's *loss function*.

In order to calculate the total loss incurred by the portfolio, we have to determine whether the respective counterparties in the portfolio have defaulted or not. In general, assuming that there are K counterparties in the portfolio, the joint counterparty defaults would be driven by a set of S systematic factors $Z_i, i = 1, \ldots, S$ along with K distinct idiosyncratic factors for the respective counterparties, $\varepsilon_k, k = 1, \ldots, K$. For simplicity we assume further that Z_i and ε_k are independent standard random normal variables. The creditworthiness index of each counterparty k is defined as:

$$\mathrm{CWI}_k = \sum_{i=1}^{S} \beta_{ik} Z_i + \sigma_k \varepsilon_k, \qquad k = 1, \ldots, K$$

where β_{ik} represents the sensitivity of the k-th counterparty to the i-th systematic factor, $i = 1, \ldots, S$ and:

$$\sigma_k = \sqrt{1 - \sum_{i=1}^{S} \beta_{ik}^2}, \quad k = 1, \ldots, K$$

Given PD_k, the default probability of counterparty k, the k-th counterparty defaults if its creditworthiness index falls below the threshold $\Phi^{-1}(\mathrm{PD}_k)$, where $\Phi(.)$ is the cumulative normal distribution and Φ^{-1} is its inverse. So the default indicator of counterparty k can be written as

$$D_k = \begin{cases} 1, & \text{if } \mathrm{CWI}_k \leqslant \Phi^{-1}(\mathrm{PD}_k) \\ 0, & \text{otherwise} \end{cases}$$

For the sake of simplicity, in our work we assume that the creditworthiness index of a counterparty is in agreement with the Basel portfolio model and driven by only a single factor Gaussian copula[1] (i.e. we have assumed $S = 1$ in equation (4.1.1) and the systematic risk factor Z_1 is simply referred to as Z henceforth):

$$\mathrm{CWI}_k = \sqrt{\rho_k} \cdot Z + \sqrt{1 - \rho_k} \cdot \varepsilon_k, \qquad k = 1, \ldots, K \tag{4.1.1}$$

where Z and ε_k are independent standard normal random variables and ρ_k is the factor loading giving the sensitivity of counterparty k to the systematic factor Z. The systematic risk factor, Z, represents macroeconomic or industry level events which influence the performance of each counterparty while idiosyncratic risk factors (denoted by ε_k) reflect the risk that is unique to each counterparty. While we use a single factor model in this chapter, one can in principle use a multi-factor model to capture more sophisticated dependence structures. Given PD_k, the default probability of counterparty k, then that counterparty will default if:

$$\mathrm{CWI}_k \leqslant \Phi^{-1}(\mathrm{PD}_k), \qquad k = 1, \ldots, K$$

where $\Phi^{-1}(\cdot)$ represents the inverse of the cumulative distribution function for the standard normal random variable. Assume that we are given a finite number of market scenarios ($M < \infty$), each of which is denoted by $y_j, \quad j = 1, \ldots, M$.

Let $\overline{y}_{kj}$ be the exposure of the portfolio to counterparty k, $k = 1, 2, \ldots, K$ under the j-th market scenario $y_j, \quad j = 1, \ldots, M$. These exposures are given in an EAD matrix, i.e., $\overline{y}_{kj}$, the exposure to the k-th counterparty, $k = 1, \ldots, K$ under the j-th market scenario $j = 1, \ldots, M$, is given in $\mathrm{EAD}(k, j)$. Then the total loss under each market scenario is defined as:

$$\begin{aligned} l_j &= \sum_{k=1}^{K} \overline{y}_{kj} \cdot D_k \\ P(Y = y_j) &= p_j, \qquad j = 1, \ldots, M \end{aligned} \tag{4.1.2}$$

The random variable Y in (4.1.2) is used to denote market scenarios, each of which contains the exposure information of all K counterparties. That is each market scenario $y_j, \quad j = 1, \ldots, M$ contains all of the respective exposure information for the K counterparties in the portfolio, namely $\overline{y}_{kj}, k = 1, \ldots, K$. Moreover the j-th market scenario happens with probability of p_j and:

$$\sum_{j=1}^{M} P(Y = y_j) = \sum_{j=1}^{M} p_j = 1$$

where $P(Y = y_j) = p_j$ represents the marginal distribution of the market scenarios. Note that we have assumed there is no correlation between exposures in each market scenario and the respective idiosyncratic factors, i.e. y_j and ε_k are independent. We will be focusing

[1] We choose the Gaussian copula solely for illustration. It could be easily replaced by other copulas in the simulation algorithm, as long as systematic and idiosyncratic risk can be identified, and conditional probabilities of default given systematic scenarios computed.

on the systematic risk, Z, and formulate the worst-case copula by using systematic losses. We do not focus on name concentration and idiosyncratic risks. Given the systematic risk factor Z, the portfolio's total loss under the j-th market scenario is:

$$l_j(Z) = \sum_{k=1}^{K} \overline{y}_{kj} \Phi \left(\frac{\Phi^{-1}(\mathrm{PD}_k) - \sqrt{\rho_k} \cdot Z}{\sqrt{1-\rho_k}} \right) \tag{4.1.3}$$

In order to formulate the *worst-case joint distribution* problem, we discretize the systematic risk factor Z by using N points and define the credit space for these N states as $(z_1, \ldots, z_N)$. By further discretizing the portfolio losses under each market scenario, $l_j, j = 1, \ldots, M$ and for the i-th credit state, $i = 1, \ldots, N$, we have:

$$\begin{aligned} l_{ij} &= \sum_{k=1}^{K} \overline{y}_{kj} \Phi \left(\frac{\Phi^{-1}(\mathrm{PD}_k) - \sqrt{\rho_k} \cdot z_i}{\sqrt{1-\rho_k}} \right) \\ P(Z = z_i) &= q_i, \quad i = 1, \ldots, N \end{aligned} \tag{4.1.4}$$

where $P(Z = z_i)$ represents the marginal distribution of the credit risk factor Z for which we have

$$\sum_{i=1}^{N} P(Z = z_i) = \sum_{i=1}^{N} q_i = 1$$

and l_{ij} represents the losses under the i-th credit scenario $i = 1, \ldots, N$ and the j-th market scenario $j = 1, \ldots, M$.

Therefore the worst-case joint distribution of market risk factors and credit risk factors is found by solving the following optimization problem:

$$\max_{P(Y,Z) \in \mathfrak{F}(F_Y, F_Z)} \mathrm{CVaR}_\alpha(L) \tag{4.1.5}$$

where $P(.,.) \in$ is a joint distribution of market risk factors and credit risk factors for which $\vartheta_{ij} = P(Y = y_j, Z = z_i)$ denotes the joint probability of the j-th market scenario and the i-th credit scenario ($i = 1, \ldots, N$ and $j = 1, \ldots, M$). $\mathfrak{F}(F_Y, F_Z)$ is the *Fréchet class* of all possible joint distributions of (Y, Z) matching the previously defined marginals F_Y and F_Z. Note that since the sum of each marginal distribution is equal to one, we do not have to include the additional constraint for specifying that the total sum of the joint distribution is equal to one as well.

4.1.1 LP Formulation for the Worst-case Conditional Value-at-Risk

Problem (4.1.5) can be recast as:

$$\begin{aligned}
\max_{\vartheta,\mu} \quad & (1-\alpha)^{-1} \sum_{i=1}^{N}\sum_{j=1}^{M} l_{ij}\mu_{ij} \qquad (4.1.6)\\
& \sum_{i=1}^{N} \vartheta_{ij} = p_j, \quad j = 1,\ldots,M\\
& \sum_{j=1}^{M} \vartheta_{ij} = q_i, \quad i = 1,\ldots,N\\
& \sum_{i=1}^{N}\sum_{j=1}^{M} \mu_{ij} = 1-\alpha,\\
& 0 \leqslant \mu_{ij} \leqslant \vartheta_{ij}, \quad i = 1,\ldots,N, \quad j = 1,\ldots,M
\end{aligned}$$

While we will be concerned with the latter representation (problem (4.1.6)) of the worst-case joint distribution problem and present our numerical example in section 4.2 based on this, an alternative derivation of problem (4.1.6) is also presented in Appendix B for the sake of completeness.

4.2 Application to Counterparty Credit Risk

At a given confidence level α, the worst-case joint distribution of market and credit factors, $\vartheta_{ij}, i = 1,\ldots,N, \quad j = 1,\ldots,M$, can be obtained by solving the linear programming problem stated in (4.1.6).

Having found the discretized worst-case joint distribution, we can simulate from the full (not just systematic) credit loss distribution by using the following algorithm in order to generate portfolio losses:

Algorithm 4.2.1: Total (or Systematic) Portfolio Loss Generation Using the Worst-case Copula

---

1. Given a set of M market scenarios (each of which contains the exposure information for all K counterparties) and an N-point discretization of the systematic risk factors $Z \sim N(0,1)$ as $(z_1,\ldots,z_N)$, define:

$$\begin{aligned}
l_{ij} &= \sum_{k=1}^{K} \overline{y}_{kj}\Phi\left(\frac{\Phi^{-1}(\mathsf{PD}_k) - \sqrt{\rho_k}z_i}{\sqrt{1-\rho_k}}\right)\\
q_i &= \Phi(z_i) - \Phi(z_{i-1}), \qquad i = 2,\ldots,N-1\\
q_1 &= \Phi^{-1}(z_1), \qquad q_N = 1-\Phi^{-1}(z_N)\\
p_j &= 1/M, \qquad j = 1,\ldots,M \qquad (4.2.1)
\end{aligned}$$

and solve the linear programming problem (4.1.6) to determine the worst-case joint distribution of market risk factors and credit risk factors, $\vartheta_{ij}, \quad i = 1, \ldots, N, \quad j = 1, \ldots, M$.

2. Repeat the following steps N_0 times to simulate the total (or systematic) loss vector L_0 as required:

   2.1 Generate a random credit state i_0 and a random market scenario j_0 by using algorithm 4.2.2,

   2.2 Simulate z_0, a standard normal random variable conditioned to be in (z_{i_0-1}, z_{i_0+1}) for the credit state i_0. Generate K i.i.d. random standard normal random variables ε_k and define the creditworthiness index of each counterparty using

$$\mathsf{CWI}_k = \sqrt{\rho_k} \cdot z_0 + \sqrt{1-\rho_k} \cdot \varepsilon_k, \qquad k = 1, \ldots, K$$

   2.3 Determine the default status of all K counterparties by using the creditworthiness indices defined in step 2.2 and calculate their respective default indicators:

$$D_k = \begin{cases} 1, & \text{if } \mathsf{CWI}_k \leqslant \Phi^{-1}(\mathsf{PD}_k) \quad k = 1, \ldots, K \\ 0, & \text{otherwise} \end{cases}$$

   2.4 For the simulated market scenario j_0, given in step 2.1, calculate the total losses by using

$$\sum_{k=1}^{K} \overline{y}_{kj_0} \cdot D_k$$

   or the systematic losses for the market scenario j_0

$$\sum_{k=1}^{K} \overline{y}_{kj_0} \Phi\left(\frac{\Phi^{-1}(\mathsf{PD}_k) - \sqrt{\rho_k} z_0}{\sqrt{1-\rho_k}}\right)$$

   as given in step 2 where $\mathsf{PD}_k, k = 1, ..., K$, is the default probability of the k-th counterparty.

In solving the linear programming problem (4.1.6), the optimal solution is stored in a vector of length MN, namely $\overline{\vartheta}$. In order to generate the random credit scenario $i_0, i_0 = 1, \ldots, N$ and a random market scenario $j_0, j_0 = 1, \ldots, M$ in the step 2.1 of algorithm 4.2.1, we use the following algorithm:

Algorithm 4.2.2: Random Market and Credit Scenario Simulation in the Worst-case Copula Loss Generation

---

1. Generate a uniform$(0, 1)$ random variable, $u \sim \mathcal{U}(0, 1)$,

2. If $u < \overline{\vartheta}(1)$ then let $c_0 = 1$. Otherwise find the *smallest* index c_0 in the vector $\overline{\vartheta}$ for which we have:

$$\sum_{i=1}^{c_0} \overline{\vartheta}(i) \leqslant u$$

3. Using the c_0 found above, calculate the market scenario j_0 as:

$$j_0 = \left\lceil \frac{c_0}{N} \right\rceil$$

and the credit state i_0, by using the above j_0:

$$i_0 = c_0 - N \cdot (j_0 - 1)$$

Once we have calculated a vector L_0 (with length N_0 as described in step 2 of algorithm 4.2.1) for the total (or systematic) losses, we can calculate the worst-case $\text{CVaR}_\alpha(L_0)$. This is done by sorting the vector L_0 in an increasing order and taking an average of the worst N_0' of the simulated losses as prescribed by the confidence level α, i.e.,

$$N_0' = \lfloor N_0(1-\alpha) \rfloor$$

For example if $N_0 = 10{,}000$ in step 2 of algorithm 4.2.1 and we are interested in calculating the worst-case CVaR for $\alpha = 0.95$, we will take an average of the largest $N_0' = 500$ elements of the sorted vector L_0.

To illustrate application of the worst-case copula problem, we use a real-world portfolio of a large financial institution. The portfolio consists of over-the-counter derivatives with a wide range of counterparties, and is sensitive to many risk factors, including interest rates and exchange rates. Results calculated by using the worst-case joint distribution are compared to those obtained by using the stress-testing algorithm correlating the systematic credit factor to total portfolio exposure, as described in Garcia-Cespedes et al. [37] and Rosen and Saunders [61].

More specifically we begin by solving the worst-case joint distribution (4.1.6) for a given, pre-computed set of exposure scenarios, and the discretization of the (systematic) credit factor in the single factor Gaussian copula credit model described above. We then simulate the full model based on the resulting joint distribution, under the assumption of no idiosyncratic wrong-way risk (so that the market factors and the idiosyncratic credit risk factors remain independent). Note that exposures are single-step EPEs based on a multi-step simulation using a model that assumes mean reversion for the underlying stochastic factors. A detailed treatment of the EPE is presented in Rosen and Saunders [61]. The market scenarios are derived from a standard Monte-Carlo simulation of portfolio exposures, so that we have:

$$P(Y = y_j) = p_j = \frac{1}{M}, \qquad j = 1, \ldots, M \tag{4.2.2}$$

Algorithm 4.2.3: Total (Systematic) Portfolio Loss Generation Using Ordered Scenario Copula

---

1. For a given EAD matrix, sort the columns of the matrix, each of which defines the exposure scenarios, in order of the increasing total portfolio exposure, i.e., after sorting we must have:

$$\sum_{i=1}^{K} \overline{y}_{ij_{(1)}} \leqslant \sum_{i=1}^{K} \overline{y}_{ij_{(2)}} \leqslant \cdots \leqslant \sum_{i=1}^{K} \overline{y}_{kj_{(M-1)}} \leqslant \sum_{i=1}^{K} \overline{y}_{kj_{(M)}}$$

where $j_{(1)}, j_{(2)}, \ldots, j_{(M-1)}, j_{(M)}$ represents the new ordering of the columns.

2. Calculate the thresholds H_m, defined as:

$$\mathrm{H}_m = \begin{cases} -\infty & m = 0 \\ \Phi^{-1}(\overline{P}_m) & m = 1, \ldots, M-1 \\ \infty & m = M \end{cases} \tag{4.2.3}$$

where for a fixed m, $\quad m = 1, \ldots, M-1$, $\overline{P}_m = \sum_{j=1}^{m} p_j$

3. Repeat the following steps N_0 times to simulate the total (or systematic) loss vector L_0 as required:

   3.1 Generate a random $s \sim N(0,1)$, determine the exposure scenario y_j according to:

$$Y = y_{j_0} \iff \mathrm{H}_{j_0-1} < s \leqslant \mathrm{H}_{j_0}, \quad j_0 = 1, \ldots, M \tag{4.2.4}$$

   and choose the j_0-th column of the sorted exposure matrix in step 1 as the respective market exposure scenario that is used in computing total (or systematic) losses in step 3.5.

   3.2 For a given $\overline{\rho} \in [-1, 1]$, the pre-specified correlation for the systematic credit factor Z and exposure factor Y, simulate from their standard bivariate Gaussian distribution, i.e., $(Y, Z) \sim N(\mathbf{0}, \Sigma)$,

$$(Y, Z) \sim N(\mathbf{0}, \Sigma), \quad \mathbf{0} = (0, 0), \quad \Sigma = \begin{pmatrix} 1 & \overline{\rho} \\ \overline{\rho} & 1 \end{pmatrix}$$

   assuming that $Y = s$ is given in the step 3.1. In this case

$$z_0 = s \cdot \overline{\rho} + \sqrt{(1 - \overline{\rho}^2)} \cdot r$$

   where r is a standard normal random variable, i.e., the conditional disribution of the systematic credit factor Z given $Y = s$ (used to generate z_0) becomes $N(\overline{\rho}s, 1 - \overline{\rho}^2)$.

   3.3 Generate K indepedently and identically distributed standard normal random variables, $\varepsilon_k, \quad k = 1, \ldots, K$, for the respective idiosyncratic risk factors of each of the K counterparties and define the creditworthiness index of each counterparty (using their respective factor loading ρ_k) as:

$$\mathsf{CWI}_k = \sqrt{\rho_k} \cdot z_0 + \sqrt{1 - \rho_k} \cdot \varepsilon_k, \qquad k = 1, \ldots, K$$

   3.4 Determine the default status of all K counterparties by using the creditworthiness

indices defined in step 3.3 and calculate their respective default indicators:

$$D_k = \begin{cases} 1, & \text{if } \mathsf{CWI}_k \leqslant \Phi^{-1}(\mathsf{PD}_k) \quad k = 1, \ldots, K \\ 0, & \text{otherwise} \end{cases}$$

3.5 For the market scenario j_0, determined in step 3.1, calculate the total losses by using

$$\sum_{k=1}^{K} \overline{y}_{kj_0} \cdot D_k$$

or the systematic losses for the market scenario j_0

$$\sum_{k=1}^{K} \overline{y}_{kj_0} \Phi \left(\frac{\Phi^{-1}(\mathsf{PD}_k) - \sqrt{\rho_k} z_0}{\sqrt{1-\rho_k}} \right)$$

as given in step 3 where PD_k, $k = 1, ..., K$, is the default probability of the k-th counterparty.

Similarly, after simulating a vector of the total (or systematic) losses L_0 using the ordered scenario copula algorithm 4.2.3) we calculate the $\mathrm{CVaR}_\alpha(L_0)$ for a given confidence level α by sorting the vector L_0 in an increasing order and taking an average of the worst N_0' of the simulated losses as given by the confidence level α. Having done that, we can compare the results of the $\mathrm{CVaR}_\alpha(L_0)$ generated by the worst-case copula algorithm 4.2.1 to those that are generated by the ordered scenario copula algorithm 4.2.3.

4.2.1 Portfolio Characteristics

The analysis presented in this section is based on a large portfolio of over-the-counter derivatives including positions in interest rate swaps and credit default swaps with 4794 counterparties. We focus on two cases, the largest 220 and largest 410 counterparties as ranked by exposure (EPE). Each case accounts for more than 95% and 99% of total portfolio exposure respectively.

Figures 4.1 and 4.2 present exposure concentration reports, giving the number of effective counterparties among the largest 220 and 410 counterparties respectively. Let w_n be the n^{th} largest exposure. Then the Herfindahl index of the N largest exposures is defined as:

$$H_N = \frac{\sum_{n=1}^{N} w_n^2}{\left(\sum_{n=1}^{N} w_n\right)^2}$$

The effective number of counterparties among the N largest counterparties with respect to total portfolio exposures is H_N^{-1}. The effective number of counterparties for the entire portfolio is shown in Figure 4.3. As can be seen in these figures the choice of the largest 220 and 410 counterparties is well justified as the number of effective counterparties for the entire portfolio is 31 in each case.

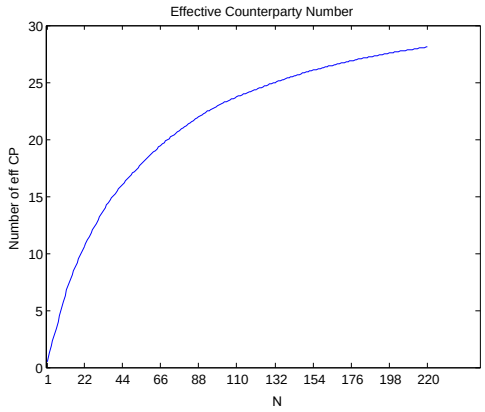


FIGURE 4.1: *Effective number of counterparties for the largest 220 counterparties.*

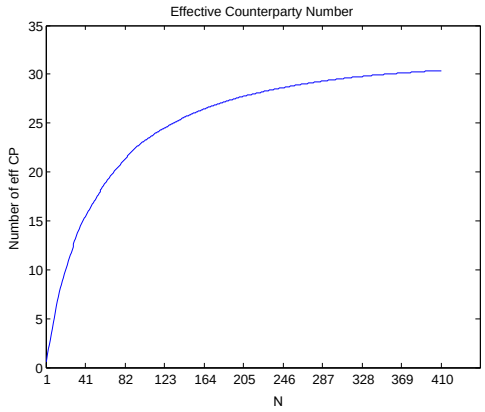


FIGURE 4.2: *Effective number of counterparties for the largest 410 counterparties.*

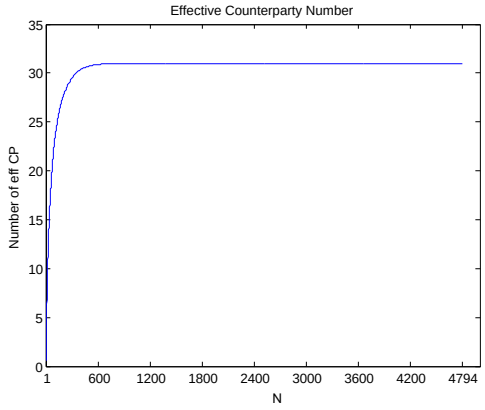


FIGURE 4.3: *Effective number of counterparties for the entire portfolio.*

The exposure simulation uses $M = 1000$ and $M = 2000$ market scenarios, while the systematic credit risk factor is discretized with $N = 1000$, $N = 2000$ and $N = 5000$ as described in algorithm 4.2.1. For CVaR calculations and comparing the results of the total (systematic) losses generated by the worst-case copula algorithm 4.2.1 and the ordered scenario copula algorithm 4.2.3, we employ the 95% and 99% confidence levels. A summary

of the above parameters is presented in Table 4.1.

The ranges of individual counterparty exposures are plotted in Figure 4.4. The 95^{th} and 5^{th} percentiles of the exposure distribution are given as a percentage of the mean exposure for each counterparty. The volatility of the counterparty exposure tends to increase as the mean exposure of the respective counterparties decreases. In other words, counterparties with a higher mean exposure tend to be less volatile compared to counterparties with lower mean exposure. Given the above characteristics, we would expect that the wrong-way risk could have an important impact on the portfolio risk, and that the contribution of idiosyncratic risk will also be significant. The distribution of the total portfolio exposures

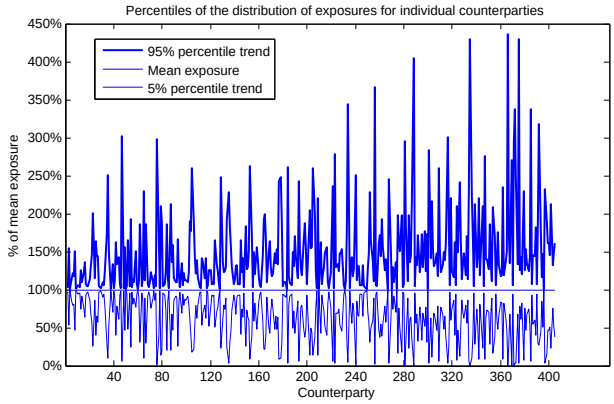


FIGURE 4.4: *5% and 95% percentiles of the exposure distributions of individual counterparties, expressed as a percentage of counterparty mean exposure (counterparties are sorted in order of decreasing mean exposure).*

from the exposure simulation is given in Figure 4.5. The histogram shows that the portfolio exposure distribution is both leptokurtic and highly skewed. The excess kurtosis and the skewness of the exposure are 117.21 and 9.42 respectively, indicating an extreme leptokurtosis and a highly skewed distribution.

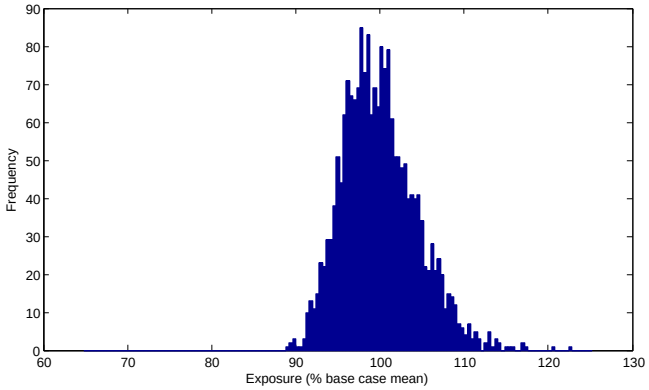


FIGURE 4.5: *Histogram of total portfolio exposures from the exposure simulation.*

Portfolio characteristics	Parameter values
Number of counterparties	4794
Market scenarios	2000
Excess kurtosis	117.21
Skewness	9.42

TABLE 4.1: *Summary of the portfolio characteristics.*

4.2.2 Numerical Results

We compare the CVaR calculated for the losses that are generated by the worst-case copula algorithm 4.2.1 for solving the linear programming problem (4.1.6), to those computed when the losses are generated by the ordered scenario copula, described in algorithm 4.2.3 of Rosen and Saunders [61]. For each level of the correlation of the systematic risk factor Z and market risk factors Y, $\overline{\rho}$, in algorithm 4.2.3 we calculate the ratio of the CVaR (measured at the confidence levels $\alpha = 95\%$ and $\alpha = 99\%$) generated by the ordered scenario copula to the one that is generated from the worst-case copula algorithm 4.2.1. In conducting simulations when both of the algorithms 4.2.1 and 4.2.3 are used, $N_0 = 100{,}000$ is chosen and the range of $\overline{\rho}$ that is used for the correlation of the systematic risk factor Z and market risk factor Y is:

$$\overline{\rho} \in \{-1, -0.8, \ldots, 0.8, 1\} \tag{4.2.5}$$

We present the results based on the three sets of discretizations of the worst-case joint distribution of market risk factors and credit risk factors. Case I employs $M = 1{,}000$ market scenarios and $N = 1{,}000$ credit scenarios; Case II doubles the number of market and credit scenarios. Lastly in Case III we use 2,000 market scenarios and 5,000 credit scenarios. Note that Case I and Case II yield a discretized worst-case distribution that has 10^6 and 4×10^6 elements while Case III's output has 10^7 elements. In each of these cases, we have solved the linear programming problem 4.1.6 at the confidence levels $\alpha = 0.95$ and $\alpha = 0.99$ respectively.

We have used CPLEX Optimization Studio to solve various instances of the linear programming problem described above and the results of the solution quality as well as the computational performance for each case are shown in table 4.2. All *solution status codes* indicate that the optimization has reached its optimal solution. This can be further confirmed by the reported *duality gap* and *sum of primal residuals* and *sum of dual residuals*.

Test I		
$MN = 10^6$	$M = 1000$ market scenarios	$N = 1000$ credit scenarios
α	$\min(\text{CVaR}_{\text{sys}}/\text{CVaR}_{\text{wcc}})$	$\max(\text{CVaR}_{\text{sys}}/\text{CVaR}_{\text{wcc}})$
0.95	52.2%	95.1%
0.99	51.8%	95.9%
Test II		
$MN = 4 \times 10^6$	$M = 2000$ market scenarios	$N = 2000$ credit scenarios
α	$\min(\text{CVaR}_{\text{sys}}/\text{CVaR}_{\text{wcc}})$	$\max(\text{CVaR}_{\text{sys}}/\text{CVaR}_{\text{wcc}})$
0.95	50.3%	96.9%
0.99	49.6%	96.6%
Test III		
$MN = 10^7$	$M = 2000$ market scenarios	$N = 5000$ credit scenarios
α	$\min(\text{CVaR}_{\text{sys}}/\text{CVaR}_{\text{wcc}})$	$\max(\text{CVaR}_{\text{sys}}/\text{CVaR}_{\text{wcc}})$
0.95	44.8%	96.4%
0.99	44.1%	97.1%

TABLE 4.3: *Minimum and maximum of the ratio of the systematic CVaR generated using the ordered scenario copula algorithm 4.2.3 to the systematic CVaR generated using the worst-case copula algorithm 4.2.1 for the largest 220 counterparties at the 95% and 99% confidence level.*

α	K	M	N	run time (in seconds)	sum of primal residuals	sum of dual residuals	duality gap	solution status code
0.95	220	1000	1000	451.1216	8.0101e-07	7.2091e-07	5.4681e-09	1
0.95	220	2000	2000	712.1227	6.9875e-07	6.2887e-07	8.5944e-09	1
0.95	220	2000	5000	1021.2694	8.1815e-07	7.3633e-07	4.538e-09	1
0.99	410	1000	1000	515.163	9.8305e-07	8.8475e-07	9.9908e-09	1
0.99	410	2000	2000	819.9888	7.1836e-07	6.4652e-07	2.6647e-09	1
0.99	410	2000	5000	1070.9282	6.6201e-07	5.9581e-07	2.6478e-09	1

TABLE 4.2: *Run time, sum of primal residuals, sum of dual residuals, duality gap and solution status code for the 6 instances of the linear programming problems (4.1.6) at the confidence levels* $\alpha = 0.95$ *and* $\alpha = 0.99$ *for the largest 220 and 410 counterparties.*

Figure 4.6 shows the graph of the ratio of the systematic CVaR calculated using the losses generated by the ordered scenario copula algorithm 4.2.3 to the systematic CVaR calculated using the losses generated by the worst-case copula algorithm 4.2.1 for the largest 220 counterparties at the 95% confidence level. The ratios of the CVaR of the systematic portfolio loss to the CVaR calculated using the worst-case joint distribution across various levels of market-credit correlation (given in (4.2.5)) and discretization scenarios indicates the worst-case joint distribution has a higher CVaR compared to the previous simulation methods for the largest 220 counterparties by 4.9%, 3.1% and 3.6% at $\alpha = 0.95$ when the systematic risk factor and the market risk factor are perfectly correlated ($\overline{\rho} = 1$).

Test I		
$MN = 10^6$	$M = 1000$ market scenarios	$N = 1000$ credit scenarios
α	$\min(\text{CVaR}_{\text{tot}}/\text{CVaR}_{\text{wcc}})$	$\max(\text{CVaR}_{\text{tot}}/\text{CVaR}_{\text{wcc}})$
0.95	52.2%	96.8%
0.99	51.6%	96.5%
Test II		
$MN = 4 \times 10^6$	$M = 2000$ market scenarios	$N = 2000$ credit scenarios
α	$\min(\text{CVaR}_{\text{tot}}/\text{CVaR}_{\text{wcc}})$	$\max(\text{CVaR}_{\text{tot}}/\text{CVaR}_{\text{wcc}})$
0.95	49.6%	97.2%
0.99	48.9%	97.4%
Test III		
$MN = 10^7$	$M = 2000$ market scenarios	$N = 5000$ credit scenarios
α	$\min(\text{CVaR}_{\text{tot}}/\text{CVaR}_{\text{wcc}})$	$\max(\text{CVaR}_{\text{tot}}/\text{CVaR}_{\text{wcc}})$
0.95	45.8%	98.1%
0.99	44.7%	98.2%

TABLE 4.4: *Minimum and maximum of the ratio of the CVaR for the total losses generated using the ordered scenario copula algorithm 4.2.3 to the CVaR for the total losses generated using the worst-case copula algorithm 4.2.1 for the largest 410 counterparties at the 95% and 99% confidence level.*

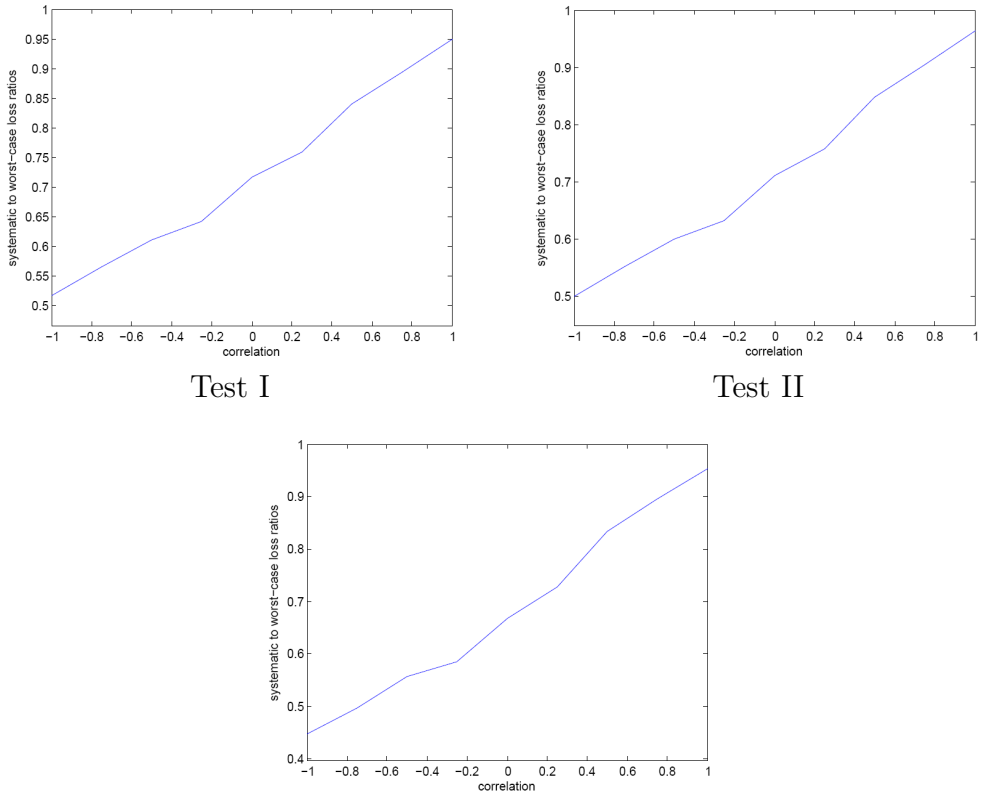


FIGURE 4.6: *Ratio of the systematic CVaR generated using the ordered scenario copula algorithm 4.2.3 to the systematic CVaR generated using the worst-case copula algorithm 4.2.1 for the largest 220 counterparties at the 95% confidence level.*

Moreover, this difference is larger for the lower levels of market-credit correlation when losses are generated by algorithm 4.2.3. It is important to note that while by using the highest level of market-credit correlation ($\overline{\rho} = 1$) in algorithm 4.2.3 the relative difference between the CVaRs generated using the ordered scenario copula algorithm 4.2.3 and the worst-case copula algorithm 4.2.1 does not exceed 4.9% (see the last column, associated with $\overline{\rho} = 1$, in table 4.5), in many practical applications the market-credit correlation is chosen in $0.2 \leqslant \overline{\rho} \leqslant 0.3$ (see Rosen and Saunders [61]) which can drastically change this difference. This relative difference for the top 220 counterparties is shown in table 4.5 (for example, see the column corresponding to $\overline{\rho} = 0.2$). In addition to the results presented in figure 4.6, table 4.3 shows the minimum and maximum of the systematic CVaR calculated using algorithm 4.2.3 to the CVaR calculated from the worst-case copula algorithm 4.2.1 at the 99% confidence level.

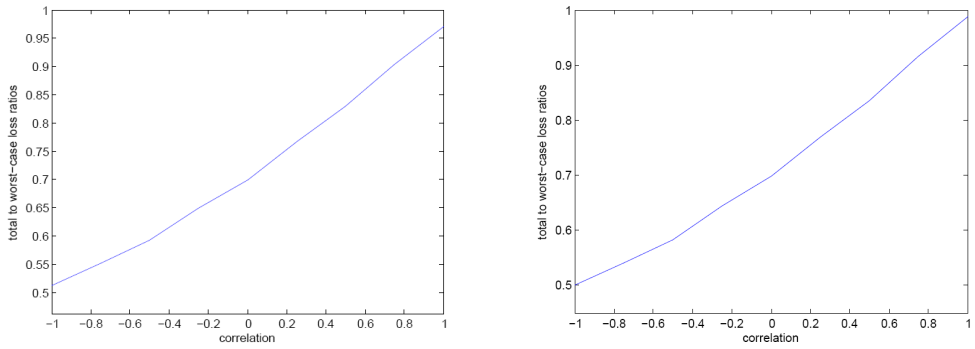
α	K	M	N	$\overline{\rho}$										
				-1	-0.8	-0.6	-0.4	-0.2	0	0.2	0.4	0.6	0.8	1
0.95	220	1000	1000	47.8%	43.0%	38.7%	34.5%	30.3%	26.0%	21.8%	17.6%	13.3%	9.1%	4.9%
0.95	220	2000	2000	49.7%	44.5%	39.9%	35.3%	30.7%	26.1%	21.5%	16.9%	12.3%	7.7%	3.1%
0.95	220	2000	5000	55.2%	49.4%	44.3%	39.2%	34.1%	29.0%	23.9%	18.9%	13.8%	8.7%	3.6%
0.99	410	1000	1000	48.2%	43.3%	38.9%	34.6%	30.2%	25.8%	21.5%	17.1%	12.8%	8.4%	4.1%
0.99	410	2000	2000	50.4%	45.2%	40.5%	35.9%	31.2%	26.6%	21.9%	17.3%	12.6%	8.0%	3.4%
0.99	410	2000	5000	55.9%	50.0%	44.8%	39.5%	34.3%	29.0%	23.8%	18.6%	13.3%	8.1%	2.9%

TABLE 4.5: *The relative difference between the systematic CVaR generated using the ordered scenario copula algorithm 4.2.3 and the systematic CVaR generated using the worst-case copula algorithm 4.2.1 for the largest 220 counterparties at the 95% and 99% confidence level.*

α	K	M	N	$\overline{\rho}$										
				-1	-0.8	-0.6	-0.4	-0.2	0	0.2	0.4	0.6	0.8	1
0.95	220	1000	1000	47.8%	42.8%	38.4%	34.0%	29.6%	25.2%	20.8%	16.4%	12.0%	7.6%	3.2%
0.95	220	2000	2000	50.4%	45.1%	40.4%	35.7%	31.0%	26.3%	21.6%	16.9%	12.2%	7.5%	2.8%
0.95	220	2000	5000	54.2%	48.4%	43.2%	38.0%	32.9%	27.7%	22.5%	17.4%	12.2%	7.0%	1.9%
0.99	410	1000	1000	48.4%	43.4%	38.9%	34.5%	30.1%	25.6%	21.2%	16.8%	12.3%	7.9%	3.5%
0.99	410	2000	2000	51.1%	45.7%	40.9%	36.1%	31.3%	26.5%	21.7%	16.9%	12.2%	7.4%	2.6%
0.99	410	2000	5000	55.3%	49.4%	44.1%	38.8%	33.5%	28.2%	22.9%	17.6%	12.4%	7.1%	1.8%

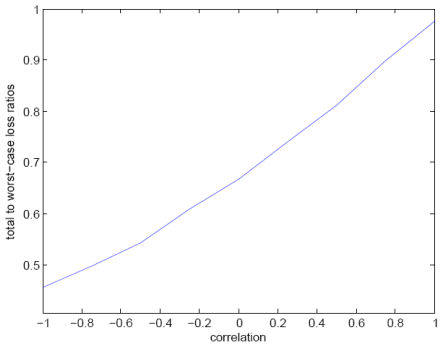
TABLE 4.6: *The relative difference between the total CVaR generated using the ordered scenario copula algorithm 4.2.3 and the systematic CVaR generated using the worst-case copula algorithm 4.2.1 for the largest 410 counterparties at the 95% and 99% confidence level.*

Alternatively we can calculate the CVaR ratios of the total losses when the losses are generated by algorithms 4.2.1 and 4.2.3 for the largest 410 counterparties (constituting more than 99.6% of the total portfolio exposure) in the portfolio. Figure 4.7 shows the



Test I

Test II



Test III

FIGURE 4.7: *Ratio of the CVaR for the total losses generated using the ordered scenario copula algorithm 4.2.3 to the CVaR for total losses generated using the worst-case copula algorithm 4.2.1 for the largest 410 counterparties at the 99% confidence level.*

CVaR ratios, computed at $\alpha = 99\%$ for the total losses and table 4.6 provides a summary of the relative difference between the total CVaR calculated at both $\alpha = 0.95$ and $\alpha = 0.99$. Similar to the results observed earlier, table 4.6 indicates that while the relative difference between the CVaRs generated using the ordered scenario copula algorithm 4.2.3 and the worst-case copula algorithm 4.2.1 does not exceed 3.5% when the market-credit correlation is perfect, this difference is larger when a lower market-credit correlation is used.

Chapter 5

Rearrangement Algorithm and VaR Analytical Bounds

5.1 Overview

An important aspect of Quantitative Risk Management (QRM) is to analyze a one-period ahead vector of losses $\mathbf{L} = (L_1, \dots, L_d)^\top$, where L_j represents the loss (which is treated as a random variable) associated with a given business line or risk type j, $j \in \{1, \dots, d\}$ over a fixed time horizon. Financial institutions often consider *aggregated loss*

$$L^+ = \sum_{j=1}^{d} L_j,$$

of particular interest. Under Pillar I of the Basel Accord (Basel Committee on Banking Supervision [62]), financial institutions are required to set capital to manage market, credit and operational risks. To this end, a risk measure $\rho(\cdot)$ is used to map the aggregate position L^+ to $\rho(L^+) \in \mathbb{R}$ to obtain the amount of capital required to account for the losses for a predetermined time horizon. Note that, in practice, capital for different risk types covers different time periods, typically based on VaR_α measured over different time horizons. As a risk measure, VaR has been widely adopted by the financial industry since the mid-nineties. It is defined as the α-quantile of the distribution function F_{L^+} of L^+, i.e.,

$$\mathrm{VaR}_\alpha(L^+) = F^-_{L^+}(\alpha) = \inf\{x \in \mathbb{R} : F_{L^+}(x) \geqslant \alpha\},$$

where $F^-_{L^+}$ denotes the quantile function of F_{L^+}; see Embrechts and Hofert [63] for more details. A well known drawback of $\mathrm{VaR}_\alpha(L^+)$ as a risk measure is that $\mathrm{VaR}_\alpha(L^+)$ is not necessarily sub-additive, unless $\mathbf{L}$ follows an elliptical distribution; see, e.g., Embrechts et al. [64], McNeil et al. [65], Embrechts et al. [5] and Hofert and McNeil [66].

There are various methods for estimating the marginal loss distributions $F_1, \dots, F_d$ of $L_1, \dots, L_d$, respectively, but the d-variate dependence structure of $\mathbf{L}$ is often more difficult to capture as typically not much is known about the underlying copula C.

In this chapter, we focus on the case where C is unknown. The case of partial information about C is studied by Bernard et al. [67] and Bernard et al. [68]. In our case, we only know that $\mathrm{VaR}_\alpha(L^+) \in [\underline{\mathrm{VaR}}_\alpha(L^+), \overline{\mathrm{VaR}}_\alpha(L^+)]$. Note that an analytical solution for $\mathrm{VaR}_\alpha(L^+)$ (or the best or worst VaR_α) is in general not available if $d \geqslant 3$.

A brief summary of the notation used in chapter 5 is given below:

$\mathbf{L}$:	the loss vector associated with a given set of d risk types
L^+ :	aggregated loss
$\mathrm{VaR}_\alpha(L^+)$:	Value-at-Risk at confidence level α
$\underline{\mathrm{VaR}}_\alpha(L^+)$:	smallest $\mathrm{VaR}_\alpha(L^+)$ over all distributions of $\mathbf{L}$ with marginals $F_1, \ldots, F_d$
$\overline{\mathrm{VaR}}_\alpha(L^+)$:	largest $\mathrm{VaR}_\alpha(L^+)$ over all distributions of $\mathbf{L}$ with marginals $F_1, \ldots, F_d$

5.2 Outline and Contributions

In section 5.3 we provide an overview of the literature on estimating the upper bound on Value-at-Risk, $\overline{\mathrm{VaR}}(L_1 + \ldots + L_d)$, when we are given the marginals of $(L_1, \ldots, L_d)$. In chapter 6, we propose improvements over the existing algorithm, the Rearrangement Algorithm (RA), discuss the performance of a modified *Adaptive Rearrangement Algorithm* (ARA) and the *Enhanced Adaptive Rearrangement Algorithm* (EARA) for this problem under different test cases.

The remaining parts of chapters 5 and 6 are organized as follows.

Section 5.3 provides a historical overview of the methodologies for computing an upper bound on VaR, $\overline{\mathrm{VaR}}_\alpha(L^+)$, for homogeneous portfolios with $L^+ = L_1 + \ldots + L_d$ where $L_i \sim F_i$ and $F_1 = \cdots = F_d$.

In section 5.4, we discuss the methods for computing an upper bound on VaR, $\overline{\mathrm{VaR}}_\alpha(L^+)$, for homogeneous portfolios, including the *dual bound approach* of Embrechts et al. [5], *Wang's approach* presented in Embrechts et al. [6] and the RA of Puccetti and Rüschendorf [7]. Then we examine numerical challenges inherent in their implementation, and present theoretical results relevant to each algorithm. More specifically:

In section 5.4.2, we prove properties of the auxiliary functions that are used in calculating $\overline{\mathrm{VaR}}_\alpha(L^+)$ in the approach of Embrechts et al. [5]. In Proposition 5.4.2, we investigate convexity and monotonicity of these functions. We further show uniqueness of the minimum for one auxiliary function, and provide example 5.4.3 to illustrate challenges presented by these functions for the case of a Generalized Pareto distribution.

In section 5.4.3, we study the uniqueness of the root of the auxiliary function used in Wang's approach, and compute the appropriate lower and upper bounds for the root-finding procedure for Pareto marginals in Propositions 5.4.4 and 5.4.5.

We prove the existence of the root for the auxiliary function h, used in Wang's approach in Proposition 5.4.13. This is done by focusing on selected properties of the auxiliary function h such as continuity, monotonicity, and differentiability in Proposition 5.4.6 and Lemmas 5.4.9, 5.4.11 and 5.4.12.

We compare the dual bound approach and the Wang approach in computing an upper bound on VaR for homogeneous portfolios using Pareto marginals. This is illustrated by the results in examples 5.4.14 and 5.4.15.

We present the RA of Puccetti and Rüschendorf [7] for approximating the worst VaR in section 5.5 and study the performance of this algorithm.

In section 5.6, we assess the effect of different properties of the input distributions such as number, heaviness of tail and degree of homogeneity on the performance of the RA and present numerical examples to illustrate the inner working of this method. More specifically, we calculate the upper bound on VaR for portfolios with given marginal loss distributions in four different portfolios using Pareto marginals (when all of the marginals are driven by a heavy-tailed distribution, a moderately heavy-tailed distribution, marginals varying from a heavy-tailed distribution to a not so heavy-tailed distribution and a mixture of the previous portfolios). We then investigate the impact of the input parameters of the RA (the choice of discretization parameter N and absolute error ε) and calculate the run time, VaR bounds, the number of iterations and the number of oppositely ordered columns (as a proxy for the objective function) at convergence.

In chapter 6, we present the *Adaptive Rearrangement Algorithm* (ARA) and the *Enhanced Adaptive Rearrangement Algorithm* (EARA), and address some of the issues identified in chapter 5. These include: (i) an introduction of new convergence criteria to choose N adaptively and (ii) an introduction of two stopping criteria, based on a given vector of relative errors, $\varepsilon = (\varepsilon_{\mathbf{1}}, \varepsilon_{\mathbf{2}})$. In the newly proposed ARA, the algorithm adaptively chooses the number of discretization points N by using new stopping criteria that computes both the relative change of the input matrix X used in computing the lower and upper bounds for $\overline{\mathrm{VaR}}_\alpha(L^+)$ as well as the relative difference of these bounds, for computing the $\overline{\mathrm{VaR}}_\alpha(L^+)$ bounds, as specified by $\varepsilon = (\varepsilon_{\mathbf{1}}, \varepsilon_{\mathbf{2}})$.

We further investigate the performance of the ARA in sections 6.3.2, 6.3.3 and 6.3.4 for generalized Pareto, Student's t-distribution and log-normal marginals for the same four portfolios studied in section 5.6 and calculate the run time, VaR bounds, the number of iterations and the number of oppositely ordered columns.

In section 6.4, we present the EARA which converges faster when the number of risk factors d is large for a given input portfolio. Finally, we provide a detailed example of the use of the ARA and the EARA based on Operational Risk data in section 6.5.

5.3 Literature Review

One of the early works on finding analytical bounds on the α-quantile of the distribution of the sum of random variables can be found in Makarov [49], who solves this problem for $d = 2$. Later on, Firpo and Ridder [52] proved these results using copula theory, introducing dependence structures into the above framework and extending Makarov's results to include an arbitrary increasing continuous aggregation function. But the authors do not prove the sharpness of the bounds. In addition they also point out that the bounds found in Makarov [49] are not the best bounds that can be attained.

Williamson and Downs [50] develop new methods for calculating convolutions and dependency bounds for the distributions of non-decreasing, continuous functions of random variables. They use lower and upper discrete approximations to the desired distribution, and provide bounds on the errors when estimating these bounds. An important contribution of the authors is in obtaining the error of the lower and upper approximations, allowing the user to know that the true distribution is contained within these bounds. Despite many interesting features of this framework, we should note that it only works for $d = 2$.

Denuit et al. [51] extend the above two-dimensional frameworks and show how to compute analytical bounds on the distribution function of $L^+ = L_1 + \ldots + L_d$ for $d \geqslant 3$. Using the notion of "stochastic dominance" (proposed in Lehmann [69]), the authors provide analytical bounds which do not use any assumption on the dependence of the underlying random variables. In a related work, Cossette et al. [70] develop results about (L_1, L_2) by assuming additional information about the correlation structure of (L_1, L_2). They further extend their results to the general multivariate case and propose analytical bounds for continuous and componentwise monotone functions in $L_1, \ldots, L_d$, assuming that the only available information on L_j is its distribution function F_j, $j \in \{1, \ldots, d\}$.

By relaxing some of the continuity assumptions with respect to the aggregation function, Embrechts et al. [71] provide a generalization of these results using copula theory. Their copula-based approach aims at addressing several issues, including the construction of risk measures for functions of dependent risks. This works aims at clarifying and identifying some of the basic issues in risk management and the tools that can help to solve these problems. Embrechts et al. [53] show that without any prior information on the dependence structure, only *some bounds* for the distribution function of the sum of the risks can be found and the problem of the *sharpness* of these bounds remained an open question when $d \geqslant 2$.

Embrechts and Puccetti [72] provide better analytical bounds on F_{L^+} based on the duality result of Rüschendorf [73] and derive the aforementioned bounds for $\sum_{j=1}^{d} L_j$ in the *homogeneous case* $F_1 = \cdots = F_d = F$ for a continuous distribution function F. Furthermore, the authors find the *best possible* bounds for the $d = 2$ case. Note that the homogeneity assumption is rather restrictive, especially for large d.

To address this problem, Embrechts and Puccetti [54] extend the dual bound approach to general portfolios and describe a numerical procedure to compute these bounds for a non-homogeneous portfolio of risks. The shortcoming of this method is that it requires the application of global optimization algorithms for which there is no guarantee for convergence to a global optimum in a reasonable and predictable amount of time. More importantly, the quality of performance of many of these optimization procedures is not well understood and, in general, the performance of such algorithms tends to deteriorate as d increases. For these reasons, the application of this method for $d \geqslant 50$ becomes intractable in some cases.

As noted earlier, the complexity of the optimization procedures required for finding the dual bounds, given arbitrary marginal distributions, along with a high running time, is the main drawback of using the dual bounds approach for obtaining a lower bound and an upper bound for $\mathrm{VaR}_\alpha(L^+)$. Puccetti and Rüschendorf [7] propose the RA to tackle these

issues. The initial idea underlying the RA and the numerical approximation introduced in Puccetti and Rüschendorf [7] for calculating $\underline{\mathrm{VaR}}_\alpha(L^+)$ and $\overline{\mathrm{VaR}}_\alpha(L^+)$ can be traced back to Rüschendorf [74] and Rüschendorf [75], respectively. The higher accuracy of the RA (which theoretically is still an open question) along with its simple implementation compared to the previous methods makes the RA an attractive alternative approach for obtaining $\underline{\mathrm{VaR}}_\alpha(L^+)$ and $\overline{\mathrm{VaR}}_\alpha(L^+)$, when (only) the marginal loss distributions $F_1, \dots, F_d$ are known. In the following section, we look at how the RA works and analyze its performance using various test cases.

More recently, Jakobsons et al. [76] utilize the concept of a convex order for inhomogeneous marginals and calculate the lower VaR bounds in this setting. They also show that by assuming only a "decreasing density" for all marginals, sharp bounds can be found. An important finding of the authors is that comonotonicity is not necessarily the worst dependence structure that one can impose, but a combination of joint mixability and mutual exclusivity. Alternatively, Bernard and McLeish [77] propose combining MCMC techniques with the Rearrangement Algorithm in designing an algorithm which converges to the global optimum. This is done by utilizing a newly defined multivariate dependence measure which assesses the convergence of the rearrangement algorithm and acts as a stopping rule.

5.4 Known Optimal Solutions in the Homogenous Case

In order to assess the quality of algorithms such as the RA, we need to know (at least some) optimal solutions with which we can compare their results. Embrechts et al. [5, Proposition 4] and Embrechts et al. [6, Proposition 1] present mathematical formulas for obtaining the worst $\mathrm{VaR}_\alpha(L^+)$ in the homogeneous case. In this section, we address numerical aspects and algorithmic improvements for computing $\overline{\mathrm{VaR}}_\alpha(L^+)$ in the homogeneous case. We assume $d \geqslant 3$ throughout this subsection. An explicit solution to this problem for $d = 2$ is given in Embrechts et al. [5, Proposition 2].

5.4.1 Crude Bounds for any $\mathrm{VaR}_\alpha(L^+)$

The following lemma provides (crude) bounds for $\mathrm{VaR}_\alpha(L^+)$. Such bounds are useful for computing initial intervals and conducting sanity checks. We do not make any (moment or other) assumptions on the involved marginal loss distributions and the bounds do not depend on the underlying unknown copula. This lemma is presented in Puccetti and Rüschendorf [55, Theorem 2.7] as well.

Lemma 5.4.1. *Let $L_j \sim F_j$, $j \in \{1, \dots, d\}$. For any $\alpha \in (0, 1)$,*

$$d \min_j F_j^-(\alpha/d) \leqslant \mathrm{VaR}_\alpha(L^+) \leqslant d \max_j F_j^-\Big(\frac{d-1+\alpha}{d}\Big), \tag{5.4.1}$$

where F_j^- denotes the quantile function of F_j, i.e., $F_j^-(u) = \inf\{x \in \mathbb{R} : F_j(x) \geqslant \alpha\}$.

Proof. Consider the lower bound for $\mathrm{VaR}_\alpha(L^+)$. By De Morgan's Law and Boole's inequality, the distribution function F_{L^+} of L^+ satisfies

$$\begin{aligned} F_{L^+}(x) &= P\Big(\sum_{j=1}^d L_j \leqslant x\Big) \leqslant P(\min_j L_j \leqslant x/d) = P\Big(\bigcup_{j=1}^d \{L_j \leqslant x/d\}\Big) \leqslant \sum_{j=1}^d P(L_j \leqslant x/d) \\ &\leqslant d \max_j F_j(x/d). \end{aligned}$$

Now, $d\max_j F_j(x/d) \leqslant \alpha$ if and only if $x \leqslant d\min_j F_j^-(\alpha/d)$ and, thus, $\mathrm{VaR}_\alpha(L^+) = F_{L^+}^-(\alpha) \geqslant d\min_j F_j^-(\alpha/d)$.

Similarly, for the upper bound for $\mathrm{VaR}_\alpha(L^+)$, we have that

$$\begin{aligned} F_{L^+}(x) &= P\Big(\sum_{j=1}^d L_j \leqslant x\Big) \geqslant P(\max_j L_j \leqslant x/d) = P(L_1 \leqslant x/d, \dots, L_d \leqslant x/d) \\ &= 1 - P\Big(\bigcup_{j=1}^d \{L_j > x/d\}\Big) \geqslant \max\Big\{1 - \sum_{j=1}^d P(L_j > x/d), 0\Big\} \\ &= \max\Big\{\sum_{j=1}^d F_j(x/d) - d + 1, 0\Big\} \geqslant \max\{d\min_j F_j(x/d) - d + 1, 0\}. \end{aligned}$$

So, $d\min_j F_j(x/d) - d + 1 \geqslant \alpha$ if and only if $x \geqslant d\max_j F_j^-((d-1+\alpha)/d)$ and, therefore $\mathrm{VaR}_\alpha(L^+) = F_{L^+}^-(\alpha) \leqslant d\max_j F_j^-((d-1+\alpha)/d)$. □

5.4.2 The Dual Bound Approach for Computing $\overline{\mathrm{VaR}}_\alpha(L^+)$

This approach for computing $\overline{\mathrm{VaR}}_\alpha(L^+)$ in the homogeneous case with margin(s) F is presented in Embrechts et al. [5, Proposition 4] and termed the *dual bound approach* in what follows. In the remaining part of this section, we assume that $F(0) = 0$, $F(x) < 1$ for all $x \in [0, \infty)$ and that F is absolutely continuous with ultimately decreasing density. Let

$$D(s,t) = \frac{d}{s - dt} \int_t^{s-(d-1)t} \bar{F}(x)\, dx \quad \text{and} \quad \tilde{D}(s) = \min_{t \in [0, s/d]} D(s,t), \tag{5.4.2}$$

where $\bar{F}(x) = 1 - F(x)$. In comparison to Embrechts et al. [5, Proposition 4], the *dual bound* D here uses a compact interval for t (and thus $\min\{\cdot\}$) by our requirement $F(0) = 0$ and since $\lim_{t \uparrow s/d} D(s,t) = d\bar{F}(s/d)$ by l'Hospital's Rule. The procedure for computing $\overline{\mathrm{VaR}}_\alpha(L^+)$ (for which we have $\mathrm{VaR}_\alpha(L^+) \leqslant \overline{\mathrm{VaR}}_\alpha(L^+)$) according to Embrechts et al. [5, Proposition 4] can now be given as follows.

Algorithm 5.4.2: Dual Bound Approach for Computing $\overline{\mathrm{VaR}}_\alpha(L^+)$

---

1. At a chosen confidence level α, specify initial intervals $[s_l, s_u]$ and $[t_l, t_u]$ by using:

$$t_l = 0, \quad t_u = \frac{s}{d}$$
$$s_l = d\,\overline{\mathrm{VaR}}_\alpha(L^+), \quad s_u = \max\left\{s_l + 1, \quad d \max_j F_j^-\left(1 - \frac{1-\alpha}{d}\right)\right\}$$

2. Inner root-finding in t: For each fixed $s \in [s_l, s_u]$, compute $\tilde{D}(s)$ by iterating over $t \in [t_l, t_u]$ until a t^* is found for which $h(s, t^*) = 0$ (adjusted by $-h(s, 0)$ as described below for for finding a root below s/d), where

$$h(s,t) = D(s,t) - (\bar{F}(t) + (d-1)\bar{F}(s-(d-1)t)).$$

   Then $\tilde{D}(s) = \bar{F}(t^*) + (d-1)\bar{F}(s-(d-1)t^*)$.

3. Outer root-finding in s: Iterate Step 2 over $s \in [s_l, s_u]$ until an s^* is found for which $\tilde{D}(s^*) = 1 - \alpha$.

4. Return

$$\overline{\mathrm{VaR}}_\alpha(L^+) = s^*. \tag{5.4.3}$$

Note that Algorithm 5.4.2 requires a specification of the two initial intervals $[s_l, s_u]$ and $[t_l, t_u]$ and Embrechts et al. [5] give no particular practical advice on how to choose them.

First, consider $[t_l, t_u]$. Using the definition of $D(s)$ in equation (5.4.2) and since $F(0) = 0$, we choose $t_l = 0$. For t_u, one would like to choose s/d. However, care has to be taken as $h(s, s/d) = 0$ for any s and, thus, the inner root-finding procedure will stop immediately when a root is found at $t_u = s/d$. therefore, in order to detect a root below $t_u = s/d$, the inner root-finding algorithm fixes the upper bound to a small number of an opposite sign to $h(s, 0)$. Note that this is an adjustment in the function value, and not in the root-finding interval $[t_l, t_u]$.

Now, consider $[s_l, s_u]$, in particular, s_l. According to Embrechts et al. [5, Proposition 4] it has to be chosen "sufficiently large". If s_l is chosen too small, the inner root-finding procedure in Step 2 of Algorithm 5.4.2 will not be able to locate a root; see also the left-hand side of Figure 5.1. There is currently no (good) solution on how to automatically determine a sufficiently large s_l. Given s_l, one can choose s_u as the maximum of $s_l + 1$ and the upper bound on VaR_α as given in (5.4.1), for example.

We now show a few selected properties of $D(s, t)$ and $\tilde{D}(s)$.

Proposition 5.4.2.

1. *$\tilde{D}(s)$ is decreasing.*
2. *If $\bar{F}$ is convex, so is $D(s, t)$.*

Proof.

1. Let $s \geqslant s'$ and $t' \in [0, \frac{s'}{d}]$ such that $D(s', t') = \tilde{D}(s')$. Define

$$t = \frac{s - (s' - t'd)}{d} = \frac{s - s'}{d} + t'$$

so that $0 \leqslant t' \leqslant t \leqslant \frac{s}{d}$. Let $\kappa = s' - t'd = s - td$. If $\kappa > 0$, noting that $\bar{F}$ is decreasing and that $t' \leqslant t$, we obtain

$$D(s', t') - D(s, t) = \frac{d}{\kappa}\Big(\int_{t'}^{t'+\kappa} \bar{F}(x)dx - \int_{t}^{t+\kappa} \bar{F}(x)dx \Big) \geqslant 0,$$

so that $\tilde{D}(s) \leqslant D(s, t) \leqslant D(s', t') = \tilde{D}(s')$. If $\kappa = 0$ then $\tilde{D}(s') = D(s', \frac{s'}{d}) = d\bar{F}(\frac{s'}{d}) \geqslant d\bar{F}(\frac{s}{d}) \geqslant \tilde{D}(s)$.

2. Recall that $D(s, t) = \frac{d}{s-td} \int_t^{t+(s-td)} \bar{F}(x)\, dx$. Using the transformation $z = (x - t)/(s - td)$, we have

$$D(s, t) = d \int_0^1 \bar{F}(sz + t(1 - zd))\, dz$$

Define $C = \{(s, t) \,|\, 0 \leqslant s < \infty,\ 0 \leqslant t \leqslant \frac{s}{d}\}$, and note that C is convex. Furthermore, if $\bar{F}$ is convex, then $D(s, t)$ is jointly convex in s and t on C since for $\lambda \in (0, 1)$,

$$\begin{aligned}
&D(\lambda s_1 + (1-\lambda)s_2, \lambda t_1 + (1-\lambda)t_2) \\
&= d \int_0^1 \bar{F}((\lambda s_1 + (1-\lambda)s_2)z + (\lambda t_1 + (1-\lambda)t_2)(1 - zd))\, dz \\
&= d \int_0^1 \bar{F}(\lambda(s_1 z + t_1(1 - zd)) + (1-\lambda)(s_2 z + t_2(1 - zd)))\, dz \\
&\leqslant \int_0^1 \lambda \bar{F}(((s_1 + t_1(1 - zd))) + (1-\lambda)\bar{F}((s_2 + t_2(1 - zd))))\, dz \\
&= \lambda D(s_1, t_1) + (1-\lambda)D(s_2, t_2)
\end{aligned}$$

□

Proposition 5.4.2 part 2 shows that if $\bar{F}$ is strictly convex, so is $D(s, \cdot)$ for fixed s. This gives the uniqueness of the minimum when computing $\tilde{D}(s)$ as in (5.4.2). A standard result on convexity of marginal value functions then implies that $\tilde{D}(s)$ is convex; see Rockafellar and Wets [78, Proposition 2.22].

Example 5.4.3. *As an example, consider $d = 8$ $\mathrm{Par}(\theta)$ risks, $\theta > 0$. The graph on the top in Figure 5.1 illustrates $t \mapsto h(s, t)$ for $\theta = 2$ and various s. Note that $h(s, s/d)$ is indeed 0 and for (too) small s, $h(s, t)$ does not have a root for $t \in [0, s/d)$ as mentioned above. The graph on the bottom of Figure 5.1 shows the decreasing dual bound $\tilde{D}(s)$ for various parameters θ.*

5.4.3 Wang's Approach for Computing $\overline{\mathrm{VaR}}_\alpha(L^+)$

The approach mentioned in Embrechts et al. [6, Proposition 1] is termed *Wang's approach* here. It originates from Wang et al. [79, Corollary 3.7] and, thus, historically precedes *the*

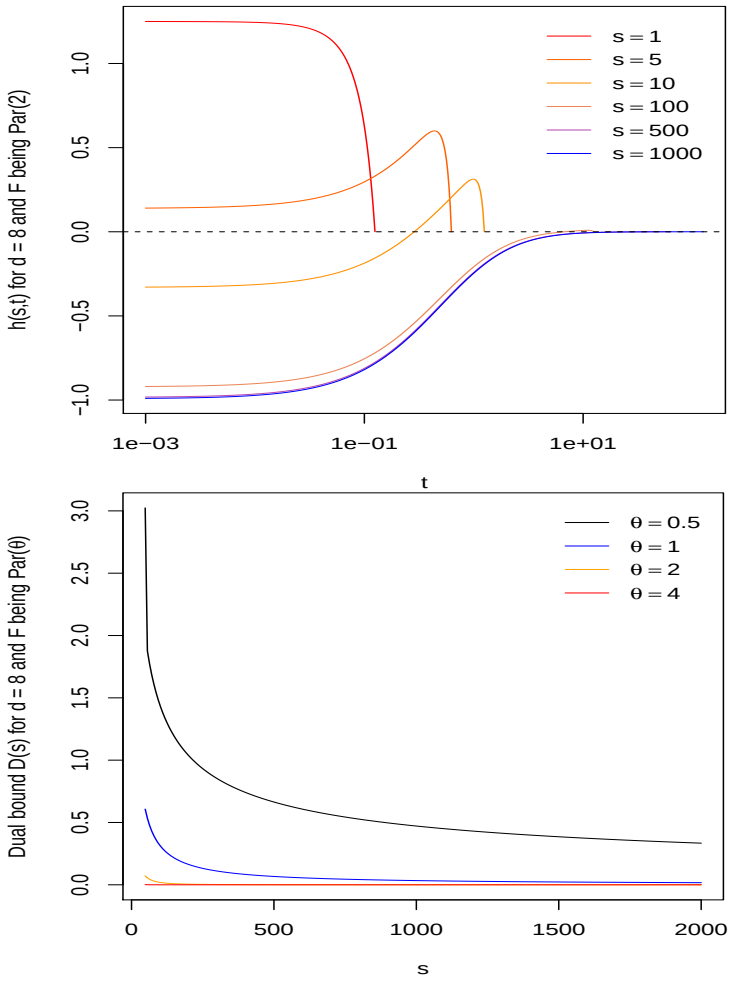


FIGURE 5.1: $t \mapsto h(s,t)$ *for various* s, $d = 8$ *and* F *being* Par(2) *(top graph). The dual bound* $\tilde{D}(s)$ *for* $d = 8$ *and* F *being* Par(θ) *for various parameters* θ *(bottom graph).*

dual bound approach of Embrechts et al. [5, Proposition 4][1]. It is conceptually simpler and numerically more stable than the dual bound approach. For notational simplicity, let

$$a_c = \alpha + (d-1)c, \quad b_c = 1 - c, \tag{5.4.4}$$

for $c \in [0, (1-\alpha)/d]$ (so that $a_c \in [\alpha, 1-(1-\alpha)/d]$ and $b_c \in [1-(1-\alpha)/d, 1]$). Assume that F admits a density which is positive and decreasing on $[\beta, \infty)$ for some $\beta \leqslant F^-(\alpha)$. Then, for $L \sim F$,

$$\overline{\mathrm{VaR}}_\alpha(L^+) = d\,\mathbb{E}[L \,|\, L \in [F^-(a_c), F^-(b_c)]], \quad \alpha \in [F(\beta), 1), \tag{5.4.5}$$

[1]Its generalization to inhomogeneous margins for convex order bounds, which implies VaR bounds is given in Jakobsons et al. [76, Corollaries 4.6 and 4.7]. The connection between the two concepts can be found in Bernard et al. [80]. This latest result is also collected in Embrechts et al. [6, Proposition 3]. In addition Bernard et al. [80] also investigate VaR bounds using complete mixability by relaxing the severity of the assumption of a "decreasing density".

where c is the smallest number in $(0, (1-\alpha)/d]$[2], such that

$$\bar{I}(c) := \frac{1}{b_c - a_c} \int_{a_c}^{b_c} F^-(y)\, dy \geqslant \frac{d-1}{d} F^-(a_c) + \frac{1}{d} F^-(b_c). \tag{5.4.6}$$

Note that c typically depends on d and α. Furthermore, in the case of a $\mathrm{Par}(\theta)$ distribution, $\bar{I}$ is given by

$$\bar{I}(c) = \begin{cases} \frac{1}{b_c - a_c} \frac{\theta}{1-\theta} ((1-b_c)^{1-1/\theta} - (1-a_c)^{1-1/\theta}) - 1, & \text{if } \theta \neq 1, \\ \frac{1}{b_c - a_c} \log\left(\frac{1-a_c}{1-b_c}\right) - 1, & \text{if } \theta = 1. \end{cases} \tag{5.4.7}$$

The conditional distribution function $F_{L|L\in[F^-(a_c),F^-(b_c)]}$ of $L \,|\, L \in [F^-(a_c), F^-(b_c)]$ is given by

$$F_{L|L\in[F^-(a_c),F^-(b_c)]}(x) = \frac{F(x) - a_c}{b_c - a_c} \quad x \in [F^-(a_c), F^-(b_c)] \tag{5.4.8}$$

Using this fact and by means of a substitution, we obtain that, for $\alpha \in [F(\beta), 1)$, (5.4.5) becomes

$$\overline{\mathrm{VaR}}_\alpha(L^+) = d \int_{F^-(a_c)}^{F^-(b_c)} x\, dF_{L|L\in[F^-(a_c),F^-(b_c)]}(x) = d\, \frac{\int_{F^-(a_c)}^{F^-(b_c)} x\, dF(x)}{b_c - a_c} = d\bar{I}(c). \tag{5.4.9}$$

Equation (5.4.9) has the advantage of having the integration in $\bar{I}(c)$ over a compact interval. Furthermore, finding the smallest c such that (5.4.6) holds also involves $\bar{I}(c)$. This procedure, thus, only requires the quantile function F^- to compute $\overline{\mathrm{VaR}}_\alpha(L^+)$. This leads to the following algorithm.

Algorithm 5.4.3: Wang's Approach for Computing $\overline{\mathrm{VaR}}_\alpha(L^+)$

---

1. Specify an initial interval $[c_l, c_u]$ with $0 \leqslant c_l < c_u < (1-\alpha)/d$.
2. Root-finding in c: Iterate over $c \in [c_l, c_u]$ until a c^* is found for which $h(c^*) = 0$, where

$$h(c) := \bar{I}(c) - \Big(\frac{d-1}{d} F^-(a_c) + \frac{1}{d} F^-(b_c)\Big). \tag{5.4.10}$$

3. Return

$$\overline{\mathrm{VaR}}_\alpha(L^+) = (d-1)F^-(a_{c^*}) + F^-(b_{c^*}). \tag{5.4.11}$$

The following proposition shows that the root-finding problem in Step 2 of Algorithm 5.4.3 is well-defined in the case of Pareto margins for all $\theta > 0$, including the infinite-mean case

[2] In contrast to what is given in Embrechts et al. [6], note that this interval has to exclude 0 since otherwise for $\mathrm{Par}(\theta)$ margins with $\theta \in (0, 1]$, c equals 0 and thus leads erroneously to the result that $\overline{\mathrm{VaR}}_\alpha(L^+) = d\bar{I}(0) = \infty$.

(note: other distributions under more restrictive assumptions have been investigated in Bernard et al. [80]).

Proposition 5.4.4. *Let $F(x) = 1 - (1+x)^{-\theta}$, $\theta > 0$, be the distribution function of the* $\mathrm{Par}(\theta)$ *distribution. Then h in Step 2 of Algorithm 5.4.3 has a unique root on $(0, (1-\alpha)/d)$, for all $\alpha \in (0,1)$ and $d > 2$.*

Proof. First consider $\theta \neq 1$. Using (5.4.7), we can rewrite $h(c)$ as

$$h(c) = \frac{c^{-1/\theta+1}\frac{\theta}{1-\theta}(1 - (\frac{1-\alpha}{c} - (d-1))^{-1/\theta+1})}{1-\alpha-dc} - \frac{(d-1)(\frac{1-\alpha}{c} - (d-1)^{-1/\theta} + 1)}{c^{1/\theta}d}.$$

Multiplying with $c^{1/\theta}d$ and rewriting the expression, we can see that $h(c) = 0$ is equivalent to $h_1(x_c) = 0$, where

$$x_c = \frac{1-\alpha}{c} - (d-1) \tag{5.4.12}$$

(which is in $(1,\infty)$ for $c \in (0,(1-\alpha)/d)$) where

$$h_1(x) = d\frac{\theta}{1-\theta}\frac{1-x^{-1/\theta+1}}{x-1} - ((d-1)x^{-1/\theta} + 1)$$

It is easy to see that $h_1(x) = 0$ if and only if $h_2(x) = 0$, where

$$h_2(x) = (d/(1-\theta) - 1)x^{-1/\theta+1} - (d-1)x^{-1/\theta} + x - (d\theta/(1-\theta) + 1), \quad x \in (1,\infty). \tag{5.4.13}$$

The proof is complete for $\theta \neq 1$ if we show that h_2 has a unique root on $(1,\infty)$. To this end, note that $h_2(1) = 0$ and $\lim_{x\uparrow\infty} h_2(x) = \infty$. Furthermore,

$$\begin{aligned} h_2'(x) &= (1 - 1/\theta)(d/(1-\theta) - 1)x^{-1/\theta} + (d-1)/\theta x^{-1/\theta-1} + 1, \\ h_2''(x) &= (d+\theta-1)/\theta^2 x^{-1/\theta-1} - (1/\theta+1)(d-1)/\theta x^{-1/\theta-2} \end{aligned}$$

It is not difficult to check that $h_2''(x) = 0$ if and only if $x = \frac{(d-1)(1+\theta)}{d+\theta-1}$, i.e.,

$$x = \frac{(d-1)(1+\theta)}{d+\theta-1} = 1 + \frac{\theta(d-2)}{d+\theta-1}, \quad d > 2, \quad \theta > 0, \quad \theta \neq 1$$

which is greater than 1 for $d > 2$. Moreover $h_2''(1) = (\theta(2-2)/\theta^2) < 0$ and $\lim_{x\downarrow 1} h_2''(x) = -(d-2)/\theta < 0$ for $d > 2$, indicating that $h_2'(x)$ is strictly negative on $(1, 1+\delta]$ for some $\delta > 0$ (as $\lim_{x\downarrow 1} h_2'(x) = 0$ and $h_2'(x)$ is strictly decreasing on $(1, 1+\delta]$). Therefore, since $h_2(1) = 0$, $h_2(x)$ is strictly negative on $(1, 1+\delta]$. Since $\lim_{x\uparrow\infty} h_2(x) = \infty$, h_2 is positive at some point $x > 1$. By the intermediate value theorem, it follows that there is *at least one* root in $(1,\infty)$. Suppose there is more than one root in $(1,\infty)$. Then, it follows that h_2 admits at least two inflection points in $(1,\infty)$ as it is continuously differentiable, a contradiction.

The proof for $\theta = 1$ works similarly. In this case, h_2 is given by

$$h_2(x) = x^2 + x(-d\log(x) + d - 2) - (d-1), \quad x \in (1, \infty), \tag{5.4.14}$$

and

$$h_2'(x) = 2x - d\log(x) - 2$$
$$h_2''(x) = 2 - \frac{d}{x}$$

implying that h_2 has the unique point of inflection at $x = d/2$. □

Let us now focus on the case of $\mathrm{Par}(\theta)$ margins and, in particular, how to choose the initial interval $[c_l, c_u]$ in Algorithm 5.4.3. We first consider c_l. $\bar{I}$ satisfies $\bar{I}(0) = \frac{1}{1-\alpha}\int_\alpha^1 F^-(y)\,dy = \mathrm{ES}_\alpha(L)$, i.e., $\bar{I}(0)$ is the *expected shortfall* of $L \sim F$ at confidence level α. If L has a finite first moment, then $\bar{I}(0)$ is finite. Therefore, $h(0)$ is finite (if $F^-(1) < \infty$) or $-\infty$ (if $F^-(1) = \infty$). Either way, one can take $c_l = 0$. However, if $L \sim F$ has an infinite first moment (see, e.g., Hofert and Wüthrich [81] or Chavez-Demoulin et al. [82] for situations in which this can happen), then $\bar{I}(0) = \infty$ and $F^-(1) = \infty$, so $h(0)$ is not defined; this happens, e.g., if F is $\mathrm{Par}(\theta)$ with $\theta \in (0, 1]$. In such a case, we are forced to choose $c_l \in (0, (1-\alpha)/d)$; see the following proposition for how this can be done theoretically.

Concerning c_u, note that l'Hospital's Rule implies that $\bar{I}(c_u) = F^-(1 - (1-\alpha)/d)$ and thus that $h((1-\alpha)/d) = 0$. We thus have a similar problem (a root at the upper endpoint of the initial interval) as for computing the dual bound. However, here we can construct a suitable $c_u < (1-\alpha)/d$. The following proposition summarizes these findings:

Proposition 5.4.5. *Let F be the distribution function of a $\mathrm{Par}(\theta)$ distribution, $\theta > 0$. Set c_l and c_u to be:*

$$c_l = \begin{cases} \frac{(1-\theta)(1-\alpha)}{d}, & \text{if } \theta \in (0,1), \\ \frac{1-\alpha}{(d+1)^{\frac{e}{e-1}}+d-1}, & \text{if } \theta = 1, \\ \frac{1-\alpha}{(d/(\theta-1)+1)^\theta+d-1}, & \text{if } \theta \in (1,\infty), \end{cases} \qquad c_u = \begin{cases} \frac{(1-\alpha)(d-1+\theta)}{(d-1)(2\theta+d)}, & \text{if } \theta \neq 1, \\ \frac{1-\alpha}{3d/2-1}, & \text{if } \theta = 1. \end{cases}$$

Then $h(c)$ defined in equation (5.4.10) *in Algorithm 5.4.3 has a unique root in $[c_l, c_u]$.*

Proof. First consider c_l and $\theta \neq 1$. Instead of h, equations (5.4.12) and (5.4.13) allow us to study

$$h_2(x) = (d/(1-\theta) - 1)x^{-1/\theta+1} - (d-1)x^{-1/\theta} + x - (d\theta/(1-\theta) + 1), \quad x \in [1, \infty).$$

Consider the two cases $\theta \in (0,1)$ and $\theta \in (1,\infty)$ separately. If $\theta \in (0,1)$, then $d/(1-\theta) - 1 > d - 1 \geqslant 0$ and $x^{-1/\theta+1} \geqslant x^{-1/\theta}$ for all $x \geqslant 1$, so $h_2(x) \geqslant ((d/(1-\theta)-1) - (d-1))x^{-1/\theta} + x - (d\theta/(1-\theta)+1) \geqslant x - (d\theta/(1-\theta)+1)$ which is 0 if and only if $x = d\theta/(1-\theta) + 1$.

Using equation (5.4.12) and solving for c we have:

$$x = d\theta/(1-\theta)+1 \quad \text{and} \quad x_c = \frac{1-\alpha}{c} - (d-1) \Rightarrow c = \frac{(1-\theta)(1-\alpha)}{d}, \quad \theta \in (0,1)$$

If $\theta \in (1,\infty)$, then using $x^{-1/\theta} \leqslant 1$ leads to $h_2(x) \geqslant (d/(1-\theta)-1)x^{-1/\theta+1} + x$ which is 0 for $x \geqslant 1$ if and only if $x = (d/(\theta-1)+1)^{\theta}$, i.e.,

$$x = (d/(\theta-1)+1)^{\theta} \quad \text{and} \quad x_c = \frac{1-\alpha}{c} - (d-1) \Rightarrow c = \frac{1-\alpha}{(d/(\theta-1)+1)^{\theta}+d-1}, \quad \theta = 1$$

Now, consider $\theta = 1$. As before, we can consider (5.4.12) and (5.4.14). By using the result that $\log x \leqslant x^{1/e}$ and $x \geqslant -x^{1+1/e}$ for $x \in [1,\infty)$, we obtain $h_2(x) \geqslant x^2 + x(-dx^{1/e} + d - 2) - (d-1) \geqslant x^2 - (d+1)x^{1+1/e}$ which is 0 if and only if $x = (d+1)^{e/(e-1)}$, and

$$x = (d+1)^{e/(e-1)} \quad \text{and} \quad x_c = \frac{1-\alpha}{c} - (d-1) \Rightarrow c = \frac{1-\alpha}{(d/(\theta-1)+1)^{\theta}+d-1}, \quad \theta \in (1,\infty).$$

Next, consider c_u. It can be easily seen that the inflection point of h_2 provides a lower bound x_c on the root of h_2. Using the inflection point x_c for $\theta \neq 1$ and solving for c we have:

$$x = \frac{(d-1)(1+\theta)}{d+\theta-1} \quad \text{and} \quad x_c = \frac{1-\alpha}{c} - (d-1) \Rightarrow c = \frac{(1-\alpha)(d-1+\theta)}{(d-1)(2\theta+d)}, \quad \theta \neq 1$$

and for $\theta = 1$

$$x = \frac{d}{2} \quad \text{and} \quad x_c = \frac{1-\alpha}{c} - (d-1) \Rightarrow c = \frac{1-\alpha}{3d/2-1}, \quad \theta = 1$$

as stated. □

While proposition 5.4.5 is concerned with Par(θ) margins, the following proposition investigates some of the properties of the crucial function h (in (5.4.10)) for Wang's approach for computing $\overline{\text{VaR}}_\alpha(L^+)$ in the homogeneous case.

Proposition 5.4.6. *Let $d \geqslant 3$ and F be a distribution function with a positive density on $[\beta,\infty)$ and $\alpha \in [F(\beta),1)$. Then*

1. *F^- equals F^{-1} on $[F(\beta),1]$ and F^{-1} is strictly increasing and continuous there. Furthermore, h given by* (5.4.10) *is continuous;*
2. *If F^{-1} is continuously differentiable, h has a negative one-sided derivative at the right endpoint.*

Proof.

1. Since F has a positive density on $[\beta,\infty)$, F is strictly increasing and continuous there and it is well known that $F^- = F^{-1}$ (see Embrechts and Hofert [63], remark

2.2), i.e., the ordinary inverse of F. Furthermore, F^{-1} is strictly increasing and continuous. The latter immediately implies continuity of h.

2. Let $a(c) = a_c$, $b(c) = b_c$ for a_c, b_c as in (5.4.4), $c_l \geqslant 0$, $c_u = (1-\alpha)/d$, $c \in (c_l, c_u)$ and $G = F^{-1}$, and note that $a(c_u) = b(c_u) = 1-(1-\alpha)/d =: p \in [1-(1-F(\beta))/d, 1) \subseteq (F(\beta), 1)$. The use of a change of variables $y = r(c,x) := a(c) + (b(c) - a(c))x$ leads to

$$\bar{I}(c) = \frac{\int_{a(c)}^{b(c)} G(y)\,dy}{b(c)-a(c)} = \int_0^1 G(a(c) + (b(c)-a(c))x)\,dx = \int_0^1 G(r(c,x))\,dx$$

and thus $h(c) = \int_0^1 G(r(c,x))\,dx - \frac{d-1}{d}G(a(c)) - \frac{1}{d}G(b(c))$. It follows from

$$r(c_u, x) = \alpha + (d-1)c_u + (1-\alpha-dc_u)x = p \tag{5.4.15}$$

that $h(c_u) = G(p) - \frac{d-1}{d}G(p) - \frac{1}{d}G(p) = 0$. Furthermore, $a'(c) = d-1$, $b'(c) = -1$ so that, by Leibniz's rule for differentiation under the integral sign,

$$\begin{aligned} h'(c) &= \int_0^1 G'(r(c,x))\frac{\partial}{\partial c}r(c,x)\,dx - \frac{d-1}{d}G'(a(c))a'(c) - \frac{1}{d}G'(b(c))b'(c) \\ &= \int_0^1 G'(r(c,x))(d-1-xd)\,dx - \frac{(d-1)^2}{d}G'(a(c)) + \frac{1}{d}G'(b(c)) \end{aligned} \tag{5.4.16}$$

and thus, by Equation (5.4.15),

$$\lim_{c\uparrow c_u} h'(c) = \int_0^1 G'(p)(d-1-xd)\,dx - \frac{(d-1)^2}{d}G'(p) + \frac{1}{d}G'(p) = G'(p)(1-d/2).$$

The claim follows by noting that $d \geqslant 3$ and, by Part 1, G is strictly increasing, so $G' > 0$ on $(F(\beta), 1)$ and thus $G'(p) > 0$.

□

Given that the assumptions stated in Proposition 5.4.6 hold for the particular F under consideration, h is known to have a unique root on $(0, (1-\alpha)/d)$. In what follows, drawing heavily on Leoni [83, Chapter 3], we present some of the selected properties of absolutely continuous functions in Lemma 5.4.9, 5.4.11 and 5.4.12. Later on, Lemmas 5.4.11 and 5.4.12 are used to prove the existence of the root for the function h in proposition 5.4.13.

Definition 5.4.7. *Let $I \subset \mathbb{R}$ be an interval. A function $f : I \to \mathbb{R}$ is* absolutely continuous *on I if for every $\epsilon > 0$ there exists a $\delta > 0$ such that*

$$\sum_{i=1}^{n} |f(b_i) - f(a_i)| \leqslant \epsilon$$

for every finite number of nonoverlapping intervals $(a_i, b_i), i = 1, \cdots, n$, with $[a_i, b_i] \subset I$

and

$$\sum_{i=1}^{n}(b_i - a_i) \leqslant \delta$$

Definition 5.4.8. *A function $f : I \to \mathbb{R}$ is* locally absolutely continuous *if it is absolutely continuous in $[a, b]$ for every $[a, b] \subset I$.*

The following lemma immediately follows from the definition 5.4.8:

Lemma 5.4.9. *Let $F : (0, \frac{1-\alpha}{d}) \to \mathbb{R}$ be a locally absolutely continuous function, and let $\lim_{x\uparrow(\frac{1-\alpha}{d})} F(x) = \infty$. Then $\limsup_{x\uparrow(\frac{1-\alpha}{d})} F'(x) = \infty$.*

Proof. Fix an $a \in (0, 1)$. By absolute continuity, for almost all x, $F(x) = F(a) + \int_a^x F'(y)\, dy$. If F' is bounded above by K, then the second term is bounded above by $F(a) + K(x - a)$, contradicting thereby the hypothesis that $F \to \infty$ as $x \to (\frac{1-\alpha}{d})$. □

The following is a combination of a stated results in Leoni [83] (See Corollary 3.50 and Exercise 3.51, page 97). $AC_{loc}(I)$ denotes the space of all locally absolutely continuous functions $f : I \to \mathbb{R}$.

Lemma 5.4.10. *Let I, J be two intervals, let $f \in AC_{loc}(J)$ and let $u : I \to J$ be monotone and AC_{loc}. Then $f \circ u \in AC_{loc}(I)$ and $(f \circ u)'(x) = f'(u(x))u'(x)$ almost everywhere.*

By combining the results of the lemmas 5.4.9 and 5.4.10 we have the following lemma:

Lemma 5.4.11. *Let $H : (0, \frac{1-\alpha}{d}) \to \mathbb{R}$ be increasing and absolutely continuous. If $\lim_{x\uparrow(\frac{1-\alpha}{d})} H(x) = \infty$ then $\limsup_{x\uparrow(\frac{1-\alpha}{d})} \frac{H'(x)}{H(x)} = \infty$*

Proof. Let $F(x) = \log(H(x))$. By Lemma 5.4.10, F is locally absolutely continuous, and $F' = \frac{H'(x)}{H(x)}$ almost everywhere. The result then follows by Lemma 5.4.9. □

Lemma 5.4.12. *Let f and g be functions such that $\lim_{x\uparrow(\frac{1-\alpha}{d})} f(x) = \infty$ and $\limsup_{x\uparrow(\frac{1-\alpha}{d})} \frac{g(x)}{f(x)} = \infty$, and let $C_1 > 0$, $C_2 > 0$. Then $\limsup_{x\uparrow(\frac{1-\alpha}{d})} C_1 g(x) - C_2 f(x) = \infty$.*

Proof. Let $M > 0$, and let $x_n \uparrow (\frac{1-\alpha}{d})$ be such that $\lim \frac{g(x_n)}{f(x_n)} = \infty$. There exists N_1, N_2 such that for $n \geqslant N_1$, $f(x_n) \geqslant 1$, for $n \geqslant N_2$, $\frac{g(x_n)}{f(x_n)} \geqslant \frac{M+C_2}{C_1}$. Then $C_1 g(x_n) - C_2 f(x_n) = f(x_n)\left(C_1 \frac{g(x_n)}{f(x_n)} - C_2\right) \geqslant M$ for $n \geqslant N_0 = \max(N_1, N_2)$. □

Proposition 5.4.13. *Under the assumptions in Proposition 5.4.6 $h(c)$ has a root in $(0, \frac{1-\alpha}{d})$.*

Proof. Using lemma 5.4.9 we can see that the (one-sided) derivative of h at $\frac{1-\alpha}{d}$ is negative, so existence of a root will follow if we can show that $\liminf_{c\downarrow 0} h(c) = -\infty$. Using the definition of h, and the fact that $b_c - a_c \to 1-\alpha$ as $c \downarrow 0$, for any $k < 1-\alpha$, and c small enough

$$\begin{aligned} h(c) &= \left(\frac{1}{b_c - a_c}\int_\alpha^{b_c} G(y)\,dy - \frac{1}{d}G(b_c)\right) - \left(\frac{1}{b_c - a_c}\int_\alpha^{a_c} G(y)\,dy + \frac{d-1}{d}G(a_c)\right) \\ &\leqslant \left(\frac{1}{k}\int_\alpha^{b_c} G(y)\,dy - \frac{1}{d}G(b_c)\right) - \left(\frac{1}{b_c - a_c}\int_\alpha^{a_c} G(y)\,dy + \frac{d-1}{d}G(a_c)\right) \\ &= T_1(c) - T_2(c) \end{aligned}$$

Recalling that $b_c = 1 - c$, and the smoothness assumptions on G, the fact that $\lim_{c\downarrow 0} G(b_c) = \infty$ implies, after applying Lemmas 5.4.11 and 5.4.12 that $\liminf_{c\downarrow 0} T_1(c) = -\infty$. Continuity of G implies that $\lim_{c\downarrow 0} T_2(c) = \frac{d-1}{d}G(\alpha)$. □

Example 5.4.14. *As an example, consider* $\mathrm{Par}(\theta)$ *risks and the confidence level* $\alpha = 0.99$. *Figure 5.2 illustrates the objective function* $h(c)$ *as a function of* $c \in (0, (1-\alpha)/d]$ *(see* (5.4.10)*) for various* θ *and* $d = 8$ *(left-hand side) and* $d = 100$ *(right-hand side). Non-positive values* $h(c)$ *have been omitted, so that the y-axis could be given in log-scale; note how steep* h *can be (we have evaluated* h *at* $2^{13} + 1$ *points equally spaced between 0 and* $(1-\alpha)/d$*), especially for small* θ *(and large* d*). For an even larger number of evaluation points* c*, one sees that the first and last positive values of* h *indeed approach 0. Overall, the objective function can be evaluated without any numerical problems on* $(0, (1-\alpha)/d]$ *for our chosen* θ *and* d.

Figure 5.3 displays $\underline{\mathrm{VaR}}_\alpha(L^+)$ *and* $\overline{\mathrm{VaR}}_\alpha(L^+)$ *calculated using Wang's approach as a function of* $1-\alpha$ *for various* θ *and* $d = 8$ *(left-hand side) and* $d = 100$ *(right-hand side).*

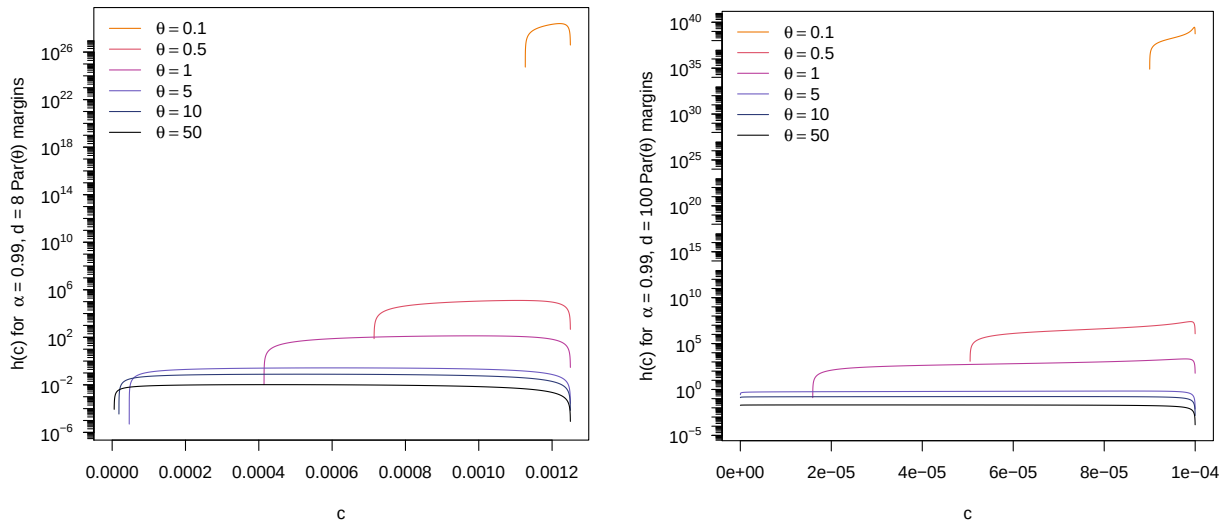


FIGURE 5.2: *Objective function* $h(c)$ *for* $\alpha = 0.99$, F *being* $\mathrm{Par}(\theta)$, $d = 8$ *(left-hand side) and* $d = 100$ *(right-hand side).*

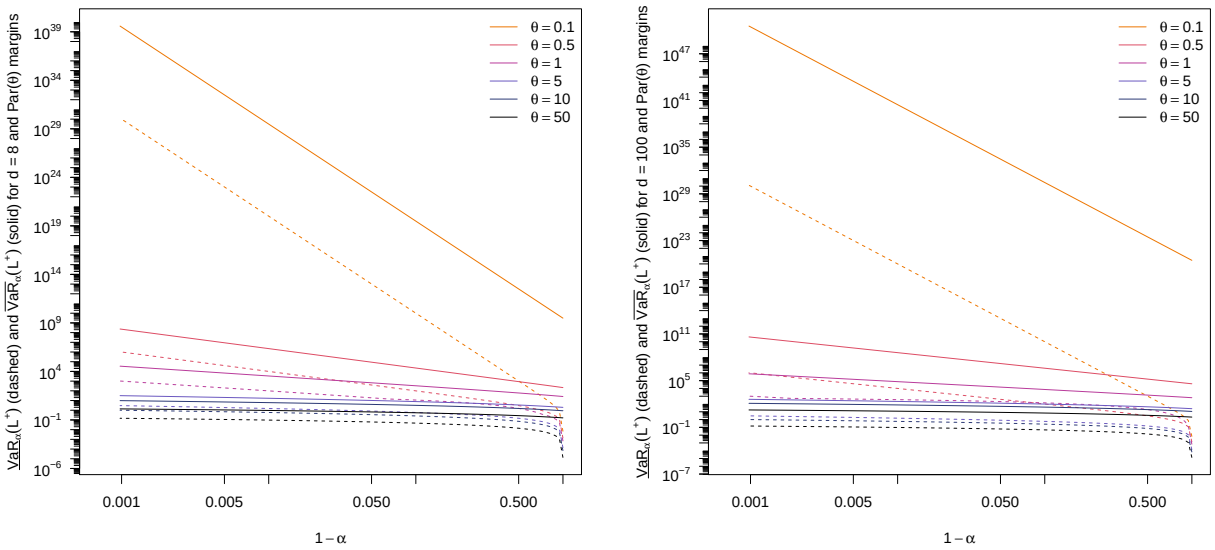


FIGURE 5.3: *Upper and lower* VaR *as functions of* $1-\alpha$ *for* F *being* $\mathrm{Par}(\theta)$, $d = 8$ *(left-hand side) and* $d = 100$ *(right-hand side).*

To obtain numerically reliable results (over these wide ranges of parameters, indeed we tested much higher dimensions as well), one has to be careful when computing the root of h *for* $c \in (0, (1-\alpha)/d)$. *First, choosing a smaller root-finding tolerance is crucial. Figure 5.4 below (see also Example 5.4.15) shows what happens if this is not considered. Second, it turned out that it is required to adjust the* theoretically valid *initial interval described in Proposition 5.4.5 further in order to guarantee that* h *is* numerically *of an opposite sign at the interval end points. In short, one should be very careful when implementing supposedly "explicit solutions" for computing* $\underline{\mathrm{VaR}}_\alpha(L^+)$ *or* $\overline{\mathrm{VaR}}_\alpha(L^+)$ *in the homogeneous case with* $\mathrm{Par}(\theta)$ *(and most likely also other) margins.*

Example 5.4.15. *Again let us consider* $\mathrm{Par}(\theta)$ *risks and the confidence level* $\alpha = 0.99$. *Figure 5.4 compares Wang's approach (using a numerical integration), Wang's approach (with an analytical formula for the integral* $\bar{I}(c)$*), Wang's approach (with an analytical formula for the integral* $\bar{I}(c)$ *and auxiliary function* h*), Wang's approach (with analytical formula for the integral* $\bar{I}(c)$*, smaller tolerance and adjusted initial interval) and the lower and upper bounds obtained from the RA; see Section 5.5. The two plots (for* $d = 8$ *and* $d = 100$*, respectively) show that comparable results are obtained by the different approaches and why it is important to use a smaller tolerance in Wang's approach. Let us again stress how important the initial interval* $[c_l, c_u]$ *is. One could be tempted to simply choose* $c_u = (1-\alpha)/d$ *and force the auxiliary function* h *to be of an opposite sign at* c_u. *Figure 5.5 shows graphs similar to the left-hand sides of Figures 5.3 (but for* $\overline{\mathrm{VaR}}_\alpha(L^+)$ *only) and 5.4 (standardized with respect to the upper bound obtained from the RA). In particular,* $\overline{\mathrm{VaR}}_\alpha(L^+)$ *is not monotone in* α *anymore (see the left-hand side of Figure 5.5) and the computed* $\overline{\mathrm{VaR}}_\alpha(L^+)$ *values are not correct anymore (see the right-hand side of Figure 5.5).*

After carefully considering all of the numerical issues, we can now look at $\underline{\mathrm{VaR}}_\alpha(L^+)$ *and* $\overline{\mathrm{VaR}}_\alpha(L^+)$ *from a different perspective. The right-hand side of Figure 5.6 shows*

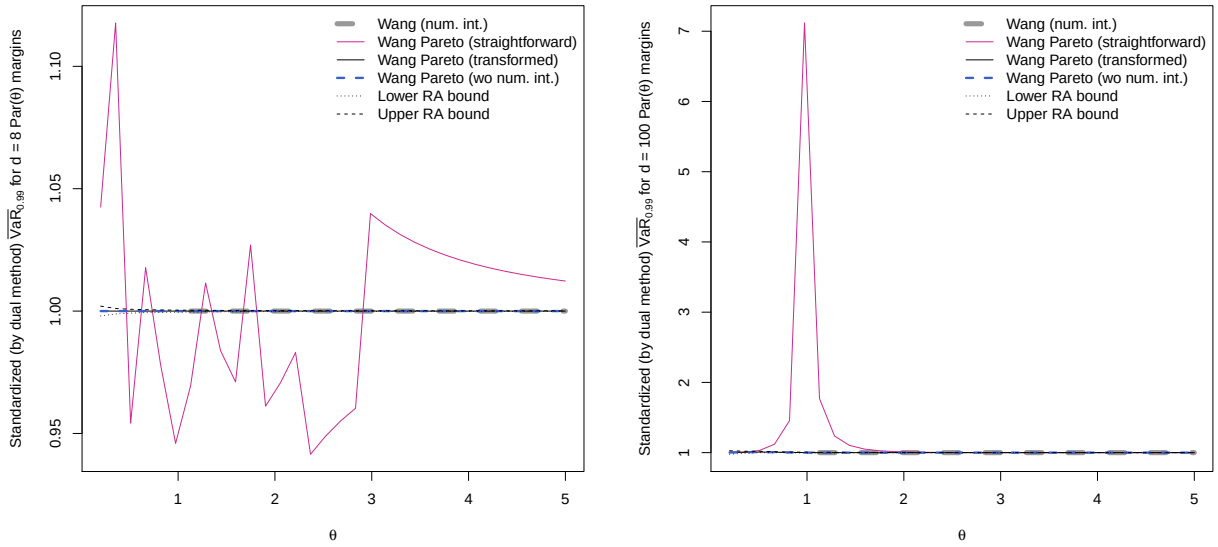


FIGURE 5.4: *Comparisons of Wang's approach (using a numerical integration), Wang's approach (with an analytical formula for the integral $\bar{I}(c)$), Wang's approach (with an analytical formula for the integral $\bar{I}(c)$ and auxiliary function h transformed to $(1,\infty)$), Wang's approach (with an analytical formula for the integral $\bar{I}(c)$, smaller tolerance and adjusted initial interval), and the lower and upper bounds obtained from the RA; all of the results are grouped based on the values obtained from the dual bound approach to facilitate comparison. The left-hand side shows the case $d = 8$, the right-hand side $d = 100$.*

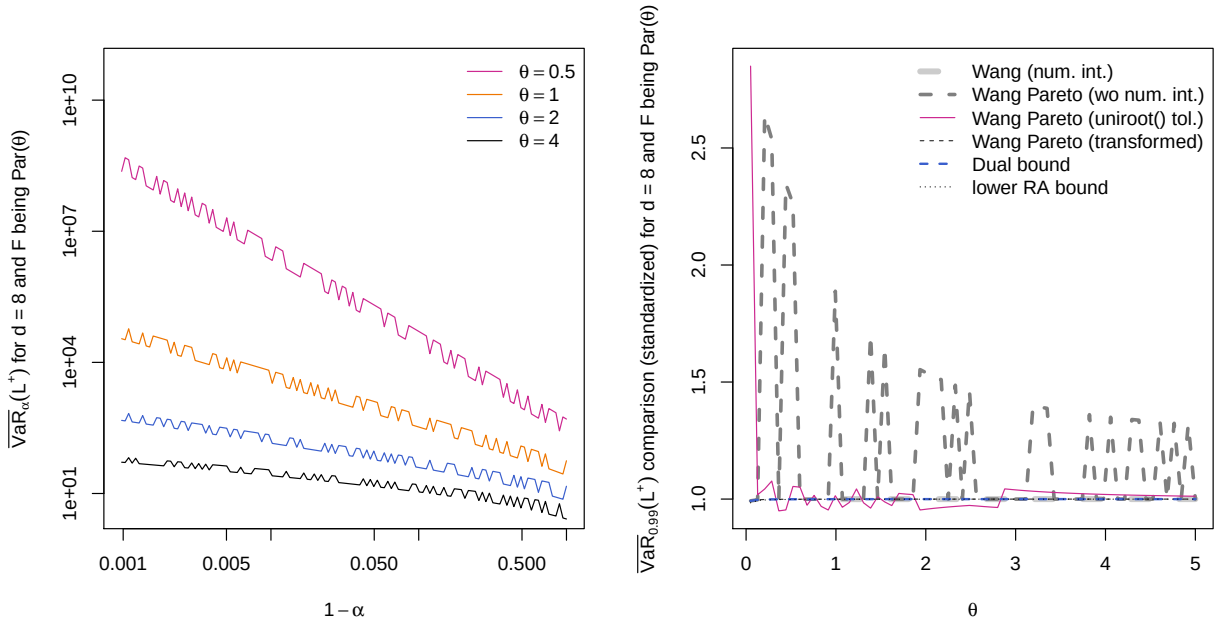


FIGURE 5.5: *Figures corresponding to the left-hand side of Figures 5.3 ($\overline{\text{VaR}}_\alpha(L^+)$ only) and 5.4 (standardized with respect to the upper bound obtained from the RA bounds) but for $h((1-\alpha)/d)$.*

$\underline{\mathrm{VaR}}_\alpha(L^+)$ *and* $\overline{\mathrm{VaR}}_\alpha(L^+)$ *as functions of the dimension* d*. The linearity of* $\overline{\mathrm{VaR}}_\alpha(L^+)$ *in the log-log scale suggests that* $\overline{\mathrm{VaR}}_\alpha(L^+)$ *is nearly a power function in* d*. To the best of our knowledge, this result has not been well explored so far.*

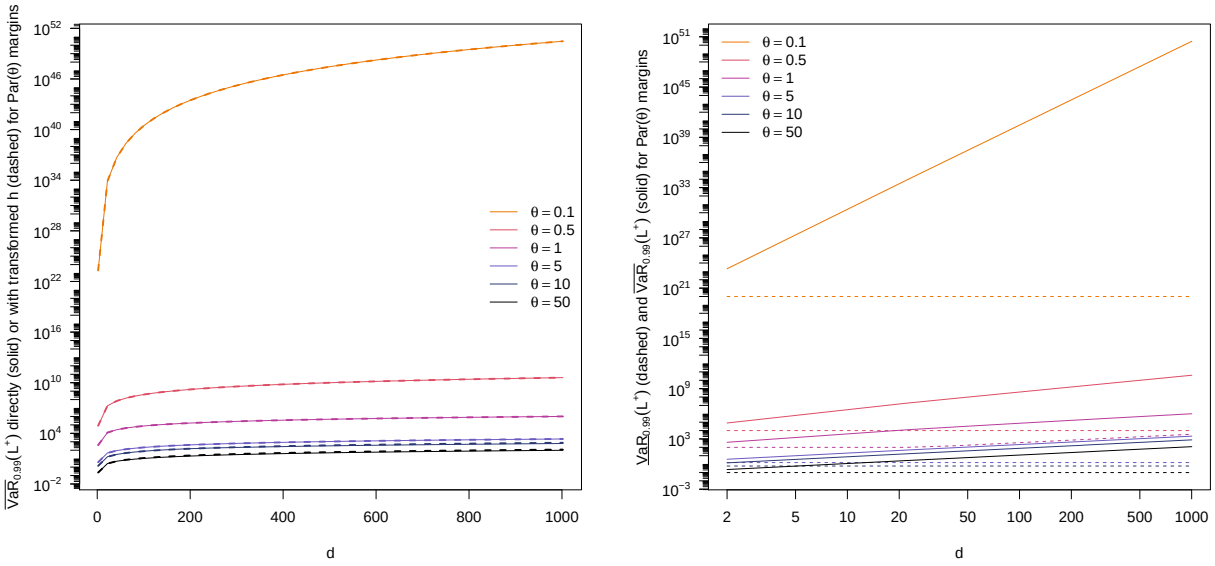


FIGURE 5.6: *A comparison of the upper* VaR *bound computed using Wang's approach (solid line) with the approach based on transforming the auxiliary function* h *to a root-finding problem on* $(1, \infty)$ *(dashed line) as described in the proof of Proposition 5.4.5 (left-hand side); lower* VaR *bound (dashed line) and* $\overline{\mathrm{VaR}}_\alpha(L^+)$ *(solid line) as functions of* d *on log-log scale (right-hand side).*

5.5 How the Rearrangement Algorithm Works

The RA can be applied to approximate the best VaR, $\underline{\mathrm{VaR}}_\alpha(L^+)$, or the worst VaR, $\overline{\mathrm{VaR}}_\alpha(L^+)$, for any set of marginals F_j, $j \in \{1, \dots, d\}$. In what follows our focus is on $\overline{\mathrm{VaR}}_\alpha(L^+)$. To understand the algorithm, note that two columns $\mathbf{a}, \mathbf{b} \in \mathbb{R}^N$ are called *oppositely ordered* if:

$$\forall i, j \in \{1, \dots, N\} \quad (a_i - a_j)(b_i - b_j) \leqslant 0$$

Given a number N of discretization points of the marginal quantile functions F_j^-, $j \in \{1, \dots, d\}$, the RA constructs two (N, d)-matrices, denoted by $\underline{X}^\alpha$ and $\overline{X}^\alpha$; the first matrix serves to construct an approximation of $\overline{\mathrm{VaR}}_\alpha(L^+)$ from below, the second matrix is used to construct an approximation of $\overline{\mathrm{VaR}}_\alpha(L^+)$ from above. Separately for each of these matrices, the RA iterates over its columns and permutes each of them so that it is oppositely ordered to the sum of all other columns. This iteration is repeated until the minimal row sum changes by less than a given $\varepsilon > 0$. As Embrechts et al. [5] state, one then typically ends up with matrices whose minimal row sums are close to each other and roughly equal to $\overline{\mathrm{VaR}}_\alpha(L^+)$. Note that if one such iteration over all columns of one of the matrices does not lead to any change in that matrix, then each column of the

matrix is oppositely ordered to the sum of all others and thus there is also no change in the minimal row sum.

An important question that we should address is whether or not the set of all matrices where columns are permutations of the columns of a given matrix X and any columns is oppositely ordered to row-sum of the remaining columns is non-empty. More formally, given an $X \in \mathbb{R}^{n\times d}$, we are interested in knowing whether the following set is empty or not:

$$O^+ = \left\{ X \in \mathcal{P}(X) : \forall c_i, c_j \in \mathrm{COL}(X) \quad c_i \text{ is oppositely ordered to } \sum_{j\neq i} c_j \right\} \tag{5.5.1}$$

where $\mathcal{P}(X)$ denotes the set of all matrices obtained by permuting the elements of each column of X and $\mathrm{COL}(X)$ represents the column space of X. The following well known lemma helps us to prove non-emptiness of O^+.

Lemma 5.5.1. *Suppose $a_1 < a_2,\ b_1 < b_2$ and f is strictly convex. Then:*

$$f(a_1 + b_2) + f(a_2 + b_1) < f(a_1 + b_1) + f(a_2 + b_2) \tag{5.5.2}$$

Proof. $a_2 + b_1 = \lambda(a_2 + b_2) + (1 - \lambda)(a_1 + b_1)$ with:

$$\lambda = \frac{a_2 - a_1}{(a_2 - a_1) + (b_2 - b_1)} \quad 1 - \lambda = \frac{b_2 - b_1}{(a_2 - a_1) + (b_2 - b_1)}$$

and $\lambda \in (0, 1)$ so:

$$f(a_2 + b_1) < \frac{a_2 - a_1}{(a_2 - a_1) + (b_2 - b_1)} f(a_2 + b_2) + \frac{b_2 - b_1}{(a_2 - a_1) + (b_2 - b_1)} f(a_1 + b_1)$$

similarly $a_1 + b_2 = \tilde{\lambda}(a_2 + b_2) + (1 - \tilde{\lambda})(a_1 + b_1)$ with:

$$\tilde{\lambda} = \frac{b_2 - b_1}{(a_2 - a_1) + (b_2 - b_1)} \quad 1 - \tilde{\lambda} = \frac{a_2 - a_1}{(a_2 - a_1) + (b_2 - b_1)}$$

and $\tilde{\lambda} \in (0, 1)$ so:

$$f(a_1 + b_2) < \frac{b_2 - b_1}{(a_2 - a_1) + (b_2 - b_1)} f(a_2 + b_2) + \frac{a_2 - a_1}{(a_2 - a_1) + (b_2 - b_1)} f(a_1 + b_1)$$

The result follows immediately from adding the two inequalities. □

Lemma 5.5.2. *O^+ is not empty.*

Proof. Consider any strictly convex function f and minimize $\sum_{i=1}^{N} f\left(\sum_{j=1}^{d} \tilde{x}_{ij}\right)$ over all matrices $\tilde{X}$ that are derived from X by permuting the elements of each column. Since there is a finite number of such permuted matrices, the minimum is attained by some matrix X^*. Such an X^* is in O^+. If this is not the case, then there exists a column

(without loss of generality we can assume that it is the first column) and a pair of i, i' such that $x^*_{i1} > x^*_{i'1}$ and $\sum_{j=2}^d x^*_{ij} > \sum_{j=2}^d x^*_{i'j}$. Using lemma 5.5.1 and switching the positions of the elements x^*_{i1} and $x^*_{i'1}$ we get a new permutation for the first column, resulting in a matrix with a strictly lower objective value. The proof follows immediately from this contradiction. □

In chapter 6, we provide an adapted version of the RA of Embrechts et al. [5] for bounding $\overline{\mathrm{VaR}}_\alpha(L^+)$ from below. To this end, let

$$s(X) = \min_{1 \leqslant i \leqslant N} \sum_{1 \leqslant j \leqslant d} x_{ij}$$

denote the minimum of the row sums of an $(N \times d)$-matrix $X = (x_{ij})$. The version of the RA given below contains more practical information than the one presented in Embrechts et al. [5]; e.g., how infinite quantiles are dealt with and clear termination conditions based on provided inputs.

Algorithm 5.5: Rearrangement Algorithm for Computing $\overline{\mathrm{VaR}}_\alpha(L^+)$

1. Assume that the chosen confidence level $\alpha \in (0,1)$, marginal quantile functions $F_1^-, \dots, F_d^-$, an integer $N \in \mathbb{N}$ and the desired convergence tolerance $\varepsilon \geqslant 0$ are given.
2. Compute the lower bound for $\overline{\mathrm{VaR}}_\alpha(L^+)$:
   - 2.1. Define the matrix $\underline{X}^\alpha = (\underline{x}^\alpha_{ij})$ for $\underline{x}^\alpha_{ij} = F_j^-(\alpha + \frac{(1-\alpha)(i-1)}{N})$, $i \in \{1, \dots, N\}$, $j \in \{1, \dots, d\}$.
   - 2.2. Permute randomly the elements in each column of $\underline{X}^\alpha$.
   - 2.3. Set $\underline{Y}^\alpha = \underline{X}^\alpha$. For $1 \leqslant j \leqslant d$, rearrange the jth column of the matrix $\underline{Y}^\alpha$ so that it becomes oppositely ordered to the sum of all other columns.
   - 2.4. While $s(\underline{Y}^\alpha) - s(\underline{X}^\alpha) > \varepsilon$, set $\underline{X}^\alpha$ to $\underline{Y}^\alpha$ and repeat Step 2.3..
   - 2.5. Set $\underline{s}_N = s(\underline{Y}^\alpha)$.
3. Compute the upper bound for $\overline{\mathrm{VaR}}_\alpha(L^+)$:
   - 3.1. Define the matrix $\overline{X}^\alpha = (\overline{x}^\alpha_{ij})$ for $\overline{x}^\alpha_{ij} = F_j^-(\alpha + \frac{(1-\alpha)i}{N})$, $i \in \{1, \dots, N\}$, $j \in \{1, \dots, d\}$. If (for $i = N$ and) for any $j \in \{1, \dots, d\}$, $F_j^-(1) = \infty$, adjust it to $F_j^-(\alpha + \frac{(1-\alpha)(N-1/2)}{N})$.
   - 3.2. Permute randomly the elements in each column of $\overline{X}^\alpha$.
   - 3.3. Set $\overline{Y}^\alpha = \overline{X}^\alpha$. For $1 \leqslant j \leqslant d$, rearrange the jth column of the matrix $\overline{Y}^\alpha$ so that it becomes oppositely ordered to the sum of all other columns.
   - 3.4. While $s(\overline{Y}^\alpha) - s(\overline{X}^\alpha) > \varepsilon$, set $\overline{X}^\alpha$ to $\overline{Y}^\alpha$ and repeat Step 3.3.
   - 3.5. Set $\overline{s}_N = s(\overline{Y}^\alpha)$.
4. Return $(\underline{s}_N,\ \overline{s}_N)$.

The main feature of the RA is the concept of *oppositely ordering* two vectors (step 2.3.) against each other, which is designed to reduce the variance of the row sums after each iteration, i.e., iteratively rearranging the elements of each column, as proposed by 5.5, drives down the variance among the row sums. The initial idea behind the RA is attributed to Rüschendorf [75, Theorem 2] which introduced *rearrangement-inequalities* for the discrete distribution function.

The randomization of the initial input, performed in the steps 2.2. and 3.2., is in place to ensure different starting points for each simulation when we calculate $\overline{\mathrm{VaR}}_\alpha(L^+)$. This is done because it has been shown that there are certain initial points for which the RA does not converge (see Haus [84, Lemma 6]). Randomization of the input matrix aims at reducing the possibility of this happening in practice.

5.5.1 Practical Challenges of Choosing Input Parameters

Some words of warning are in order. Besides the confidence level α and the marginal quantile functions $F_1^-, \dots, F_d^-$, the RA relies on two sources of input, namely $N \in \mathbb{N}$ and $\varepsilon \geqslant 0$, for which Embrechts et al. [5] do not provide any practical guidance on reasonable defaults, leaving room for interpretation on how to use it.

Concerning N, it obviously needs to be "sufficiently large", but a practitioner is left alone with such a choice. Another issue is the use of the *absolute* error ε in the algorithm. There are two problems with this. The first problem is that it is more natural to use a relative error than an absolute error in this context. Without (roughly) knowing the minimal row sum in Steps 2.4. and 3.4., a pre-specified absolute error does not guarantee that the change in the minimal row sum from $\underline{X}^\alpha$ to $\underline{Y}^\alpha$ is of the right order (and such order depends at least on d and the chosen quantile functions). If ε is chosen to be too large, the computed bounds $\underline{s}_N$ and $\overline{s}_N$ would carry too much uncertainty, whereas if it is too small, the RA has an unnecessarily long run time; the latter seems to be the case for Embrechts et al. [5, Table 3], where the chosen $\varepsilon = 0.1$ is roughly 0.000004% of the computed $\overline{\mathrm{VaR}}_\alpha(L^+)$ (for $\alpha = 0.99$), with a small absolute error.

The second problem is that the absolute error ε is only used *individually* for checking "convergence" of $\underline{s}_N$ and of $\overline{s}_N$. It does not guarantee that $\underline{s}_N$ and $\overline{s}_N$ are sufficiently close for a reasonable approximation to $\overline{\mathrm{VaR}}_\alpha(L^+)$. We are aware of the theoretical hurdles underlying the algorithm which are still open questions at this point (e.g., the probability of convergence of $\underline{s}_N$ and $\overline{s}_N$ to $\overline{\mathrm{VaR}}_\alpha(L^+)$ or that $\overline{\mathrm{VaR}}_\alpha(L^+) \leqslant \overline{s}_N$ for sufficiently large N), but from a computational point of view one should still check that $\underline{s}_N$ and $\overline{s}_N$ are close to each other.

The algorithm should return convergence and other useful information, e.g., the relative dependence uncertainty spread (i.e., the relative error $|(\overline{s}_N - \underline{s}_N)/\overline{s}_N|$), and the actual absolute errors reached for each of the bounds, the number of iterations used, the actual minimal row sums computed after each iteration or the actual number of oppositely ordered columns determined from the final matrices.

5.6 Empirical Performance Under Various Setups

In order to empirically investigate the performance of the RA, we consider 8 scenarios. Each scenario consists of one out of two studies combined with one of the four portfolios, which are described in what follows.

We consider the following:

Study 1: $N \in \{2^7, 2^8, \ldots, 2^{17}\}$ and $d = 20$.

Study 2: $N = 2^8 = 256$ and $d \in \{2^2, 2^3, \ldots, 2^{10}\}$.

These choices allow us to investigate the impact of the upper tail discretization parameter N (in Study 1) and the impact of the number of risk factors d (within a reasonable span; (in Study 2)) on the performance of the RA. The case of $d = 2$ is included for pedagogical reasons, sorting each column and oppositely ordering them produces the optimal solution.

The different portfolios we consider specify different marginal tail behaviors based on the Pareto distribution with the following distribution function

$$F_j(x) = 1 - (1 + x)^{-\theta_j}, \quad x \geqslant 0,$$

for a given tail parameter $\theta_j > 0$.

Portfolio 1: $\theta_1, \ldots, \theta_d$ are uniformly chosen from 0.6 to 0.4, representing a portfolio with marginals with similar tail behavior (i.e. a heavy-tailed distribution).

Portfolio 2: $\theta_1, \ldots, \theta_d$ are uniformly chosen from 0.5 to 1.5, representing a portfolio with marginals with different tail behavior (i.e. from a very heavy-tailed distribution to a not so heavy-tailed distribution).

Portfolio 3: $\theta_1, \ldots, \theta_d$ are uniformly chosen from 1.4 to 1.6, representing a portfolio with marginals with similar tail behavior (i.e. a not so heavy-tailed distribution).

Portfolio 4: $\theta_2, \ldots, \theta_d$ are chosen as in Portfolio3 and $\theta_1 = 0.5$, representing a portfolio with one heavy-tailed marginal loss distribution.

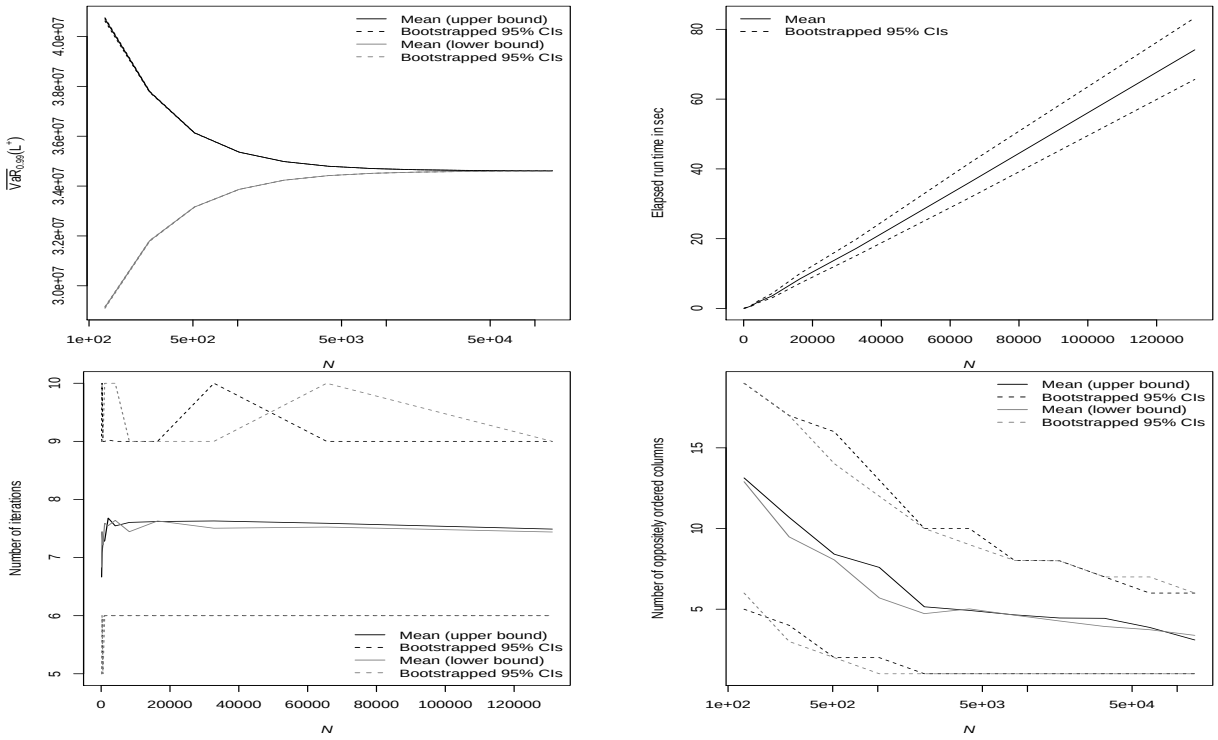


FIGURE 5.7: *Study 1, Portfolio 1:* $\mathrm{VaR}_{0.99}$ *bounds, run time, number of iterations at convergence and number of oppositely ordered columns.*

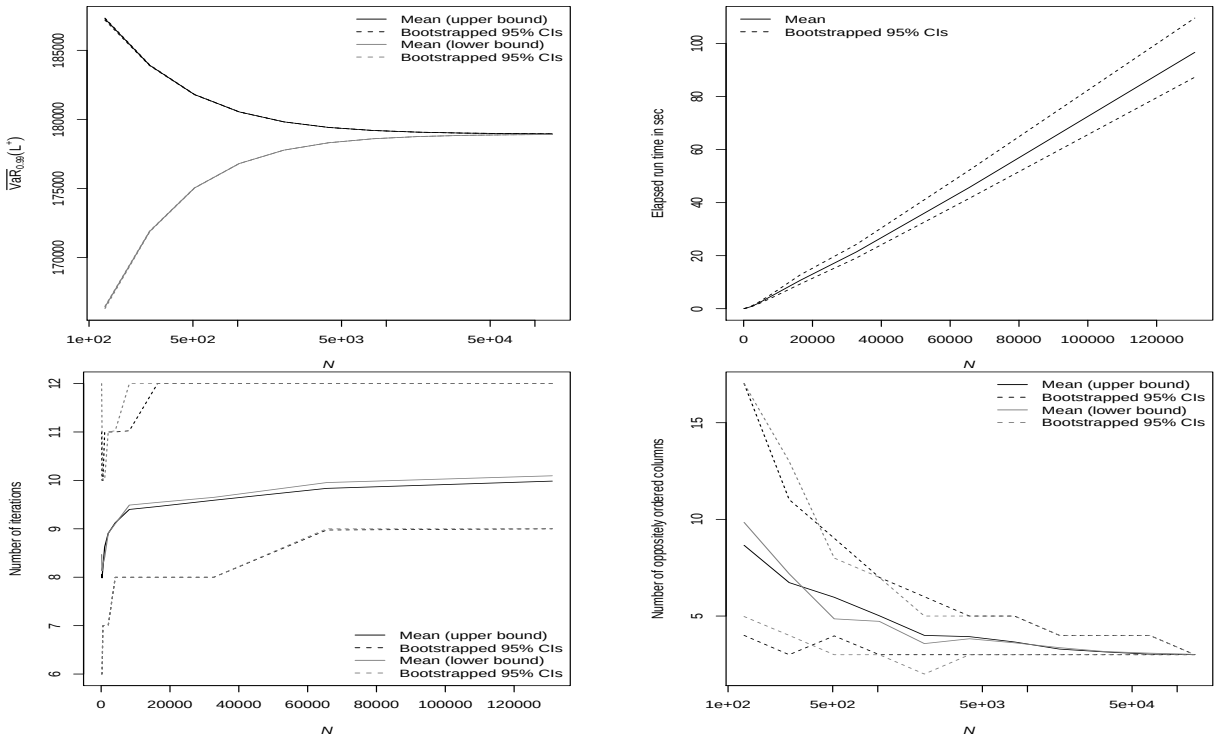


FIGURE 5.8: *Study 1, Portfolio 2:* $\mathrm{VaR}_{0.99}$ *bounds, run time, number of iterations at convergence and number of oppositely ordered columns.*

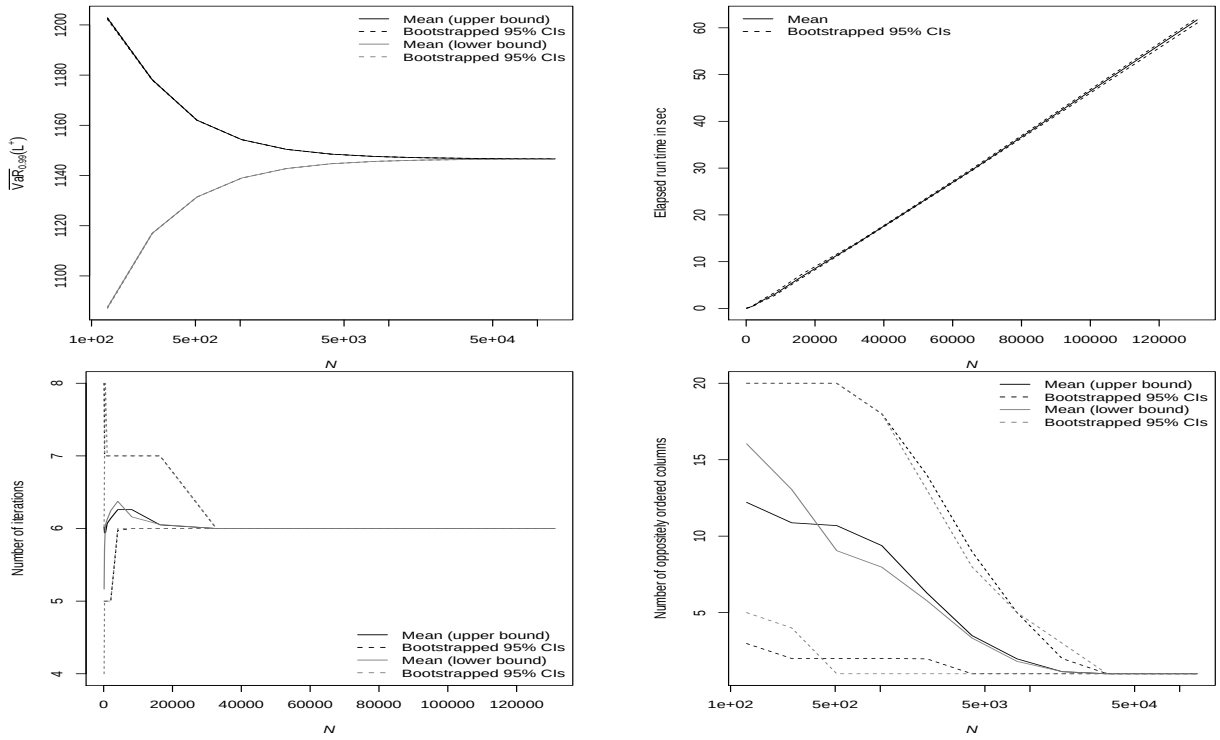


FIGURE 5.9: *Study 1, Portfolio 3:* $\mathrm{VaR}_{0.99}$ *bounds, run time, number of iterations at convergence and number of oppositely ordered columns.*

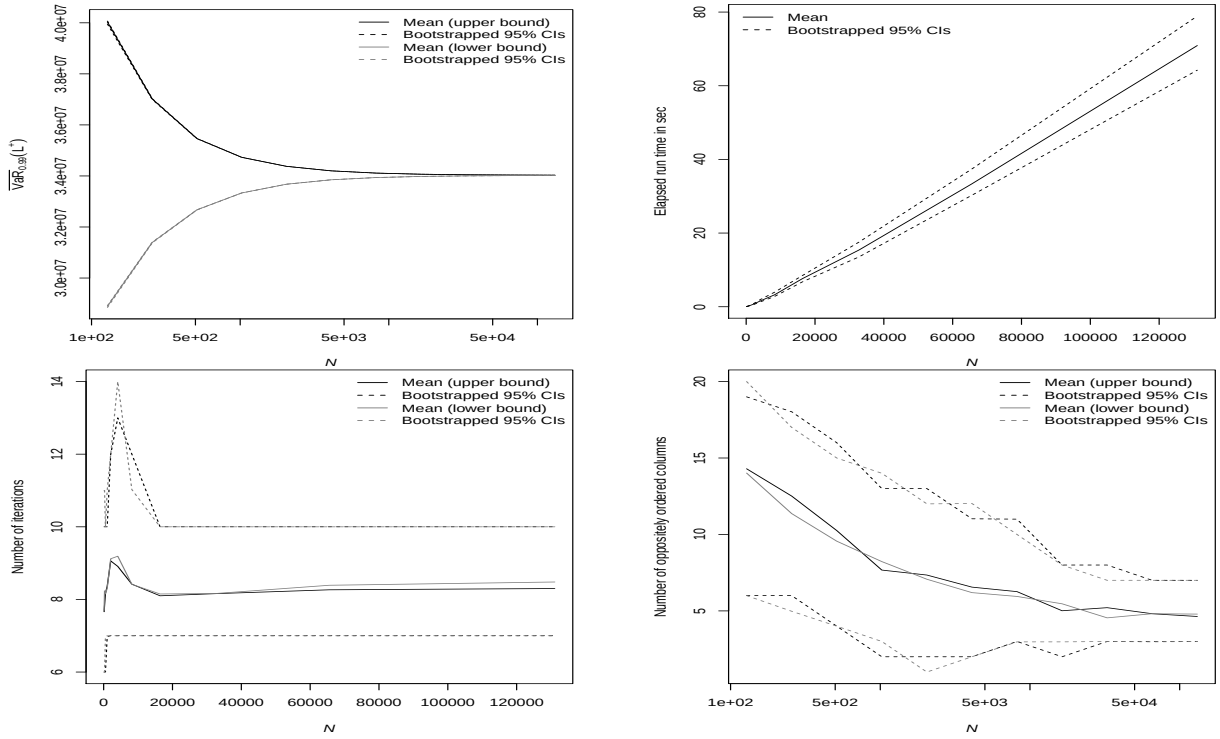


FIGURE 5.10: *Study 1, Portfolio 4:* $\mathrm{VaR}_{0.99}$ *bounds, run time, number of iterations at convergence and number of oppositely ordered columns.*

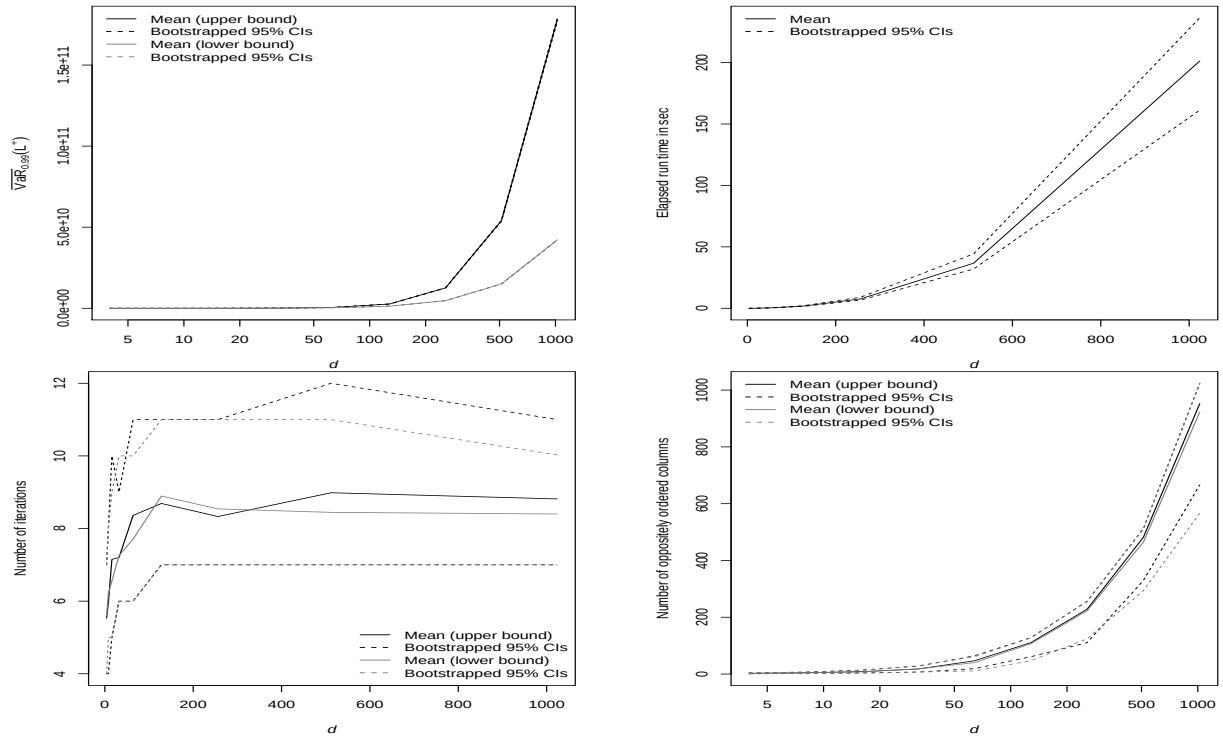


FIGURE 5.11: *Study 2, Portfolio 1:* $\text{VaR}_{0.99}$ *bounds, run time, number of iterations at convergence and number of oppositely ordered columns.*

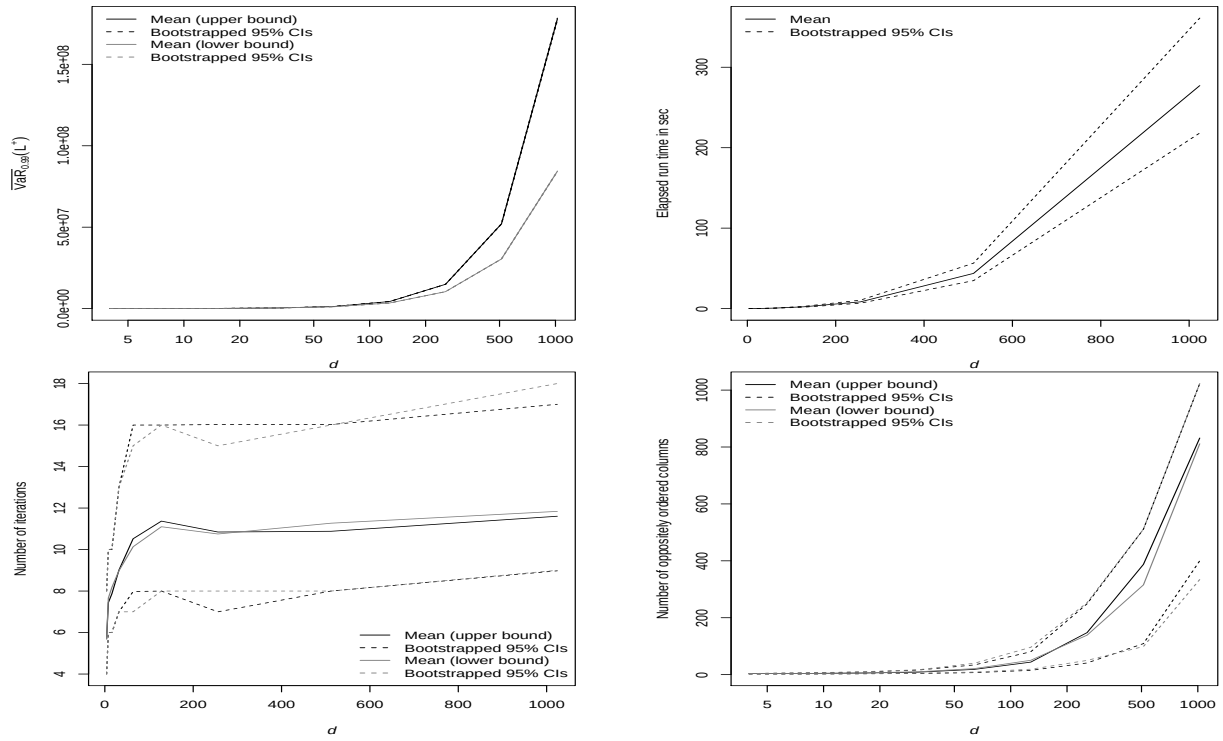


FIGURE 5.12: *Study 2, Portfolio 2:* $\text{VaR}_{0.99}$ *bounds, run time, number of iterations at convergence and number of oppositely ordered columns.*

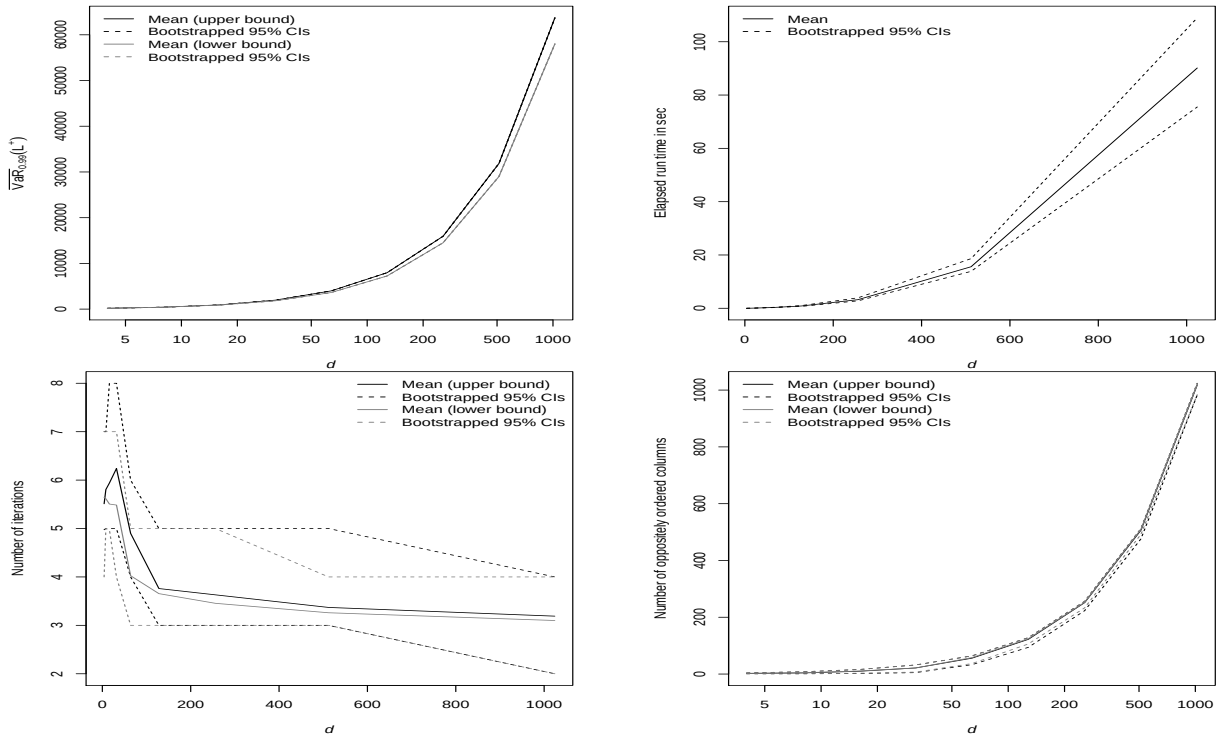


FIGURE 5.13: *Study 2, Portfolio 3:* $\mathrm{VaR}_{0.99}$ *bounds, run time, number of iterations at convergence and number of oppositely ordered columns.*

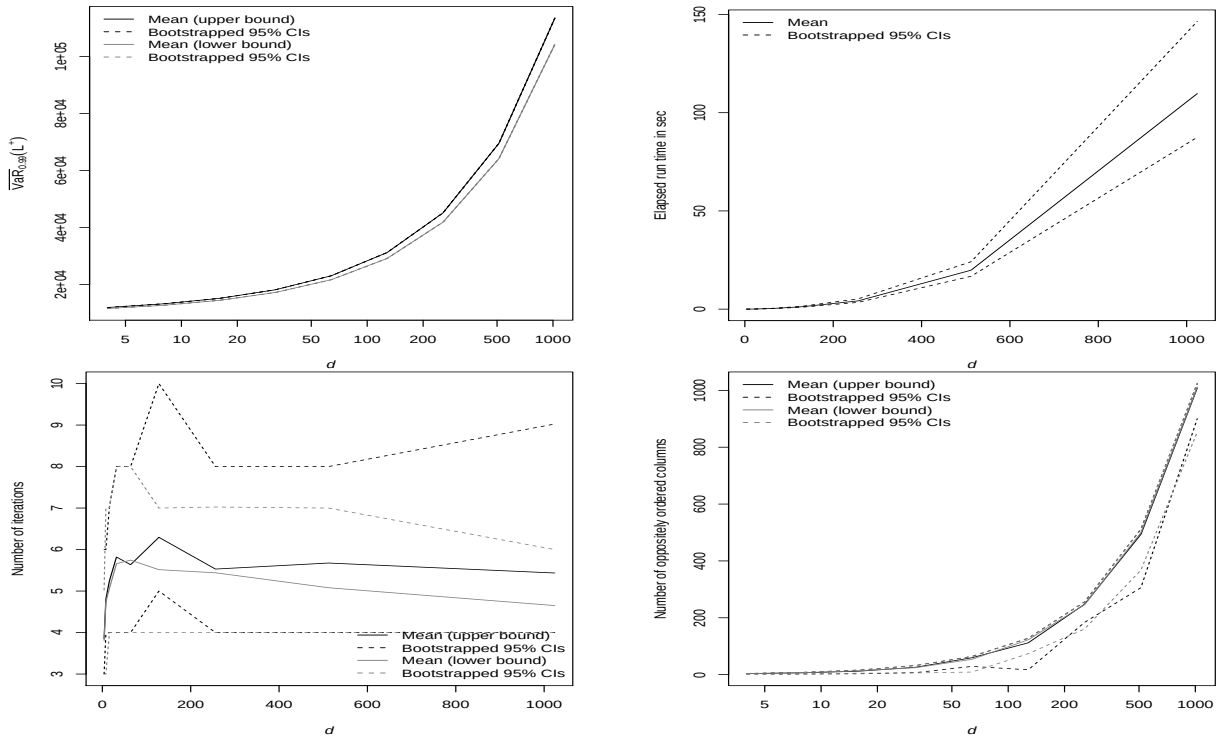


FIGURE 5.14: *Study 2, Portfolio 4:* $\mathrm{VaR}_{0.99}$ *bounds, run time, number of iterations at convergence and number of oppositely ordered columns.*

The results presented for each scenario are based on $B = 200$ simulations run on an AMD 3.2 GHz Phenom II X4 955 processor (with 16 GB RAM). The simulation results for Study 1 can be summarized as follows:

- As can be seen from the first graphs of figures (5.7), (5.8), (5.9) and (5.10), both the mean upper bound and mean lower bound of $\text{VaR}_{0.99}$ converge as N increases.
- The second graphs in figures (5.7), (5.8), (5.9) and (5.10) indicate that as N increases, so does the mean elapsed time. The mean run time in figure (5.9) is the smallest with the least variability compared to all other cases due to the particular choice of Pareto distribution parameter.
- The third graph in figures (5.7), (5.8), (5.9) and (5.10) show that the maximum of number of iterations hardly exceeds 10 as N increases; this is an important observation as it will be used later on to choose the maximum number of iterations required in the ARA.
- Finally, the last graphs indicate that the rate at which the number of oppositely ordered columns changes depends on the distribution characteristics of the input matrix X. The first and last graphs follow the same pattern, as the presence of a large first column (due to $\theta_1 = 0.5$) in the input matrix X in both cases dominates the variability of the number of oppositely ordered columns.

Figures (5.11), (5.12), (5.13) and (5.14) show the performance of the RA in the second study; here we are more interested in analyzing the impact of the number of risk factors, d, on portfolios which exhibit different marginal tail behaviors. Based on 200 simulations of the performance of the RA in each case:

- The first graphs in figures (5.11), (5.12), (5.13) and (5.14) exhibit the same pattern: the mean upper bound and mean lower bound for $\text{VaR}_{0.99}$ diverge from one another; this is due to the fact that we have kept N the same for all cases in Study 2.
- Similar to what we have seen in Study 1, the second graphs in Study 2 indicate that the mean run time of RA increases as the number of risk factors goes up. The third portfolio in Study 2 has the lowest run time on average as $\theta_1, \ldots, \theta_d$ uniformly chosen in $[1.4, 1.6]$ results in smaller elements of the input matrix X with a smaller relative range of entries compared to other portfolios.
- The mean number of iterations at convergence is consistently below 12 across all cases. More important is the uniform behaviour of the mean number of iterations at convergence as d increases. As can be seen from the third graph, in each case this number remains stable around the mean for large d's ($d \geqslant 128$). In addition, the upper and lower confidence bounds in these graphs exhibit the same pattern.
- The number of oppositely ordered columns in each case increases as d (and the number of columns in the input matrix X) increases. This finding does not contradict what we have seen in the first study as we have kept N the same here.

In the next chapter, we will introduce the ARA and show how the observations that we have made in this chapter help us set meaningful initial values of the parameters that are used in the proposed algorithm.

Chapter 6

The Adaptive Rearrangement Algorithm

6.1 Overview

As we have shown in chapter 5, the RA is approximating worst VaR for portfolios with arbitrary marginal loss distributions. In section 5.6 we have shown that the convergence tolerance ε and the tail discretization parameter N have a significant impact on the algorithm's performance, the run-time and the accuracy of the VaR bounds calculated by the RA. Moreover, once we commit to using a particular ε and N, there is no guarantee that the given convergence tolerance level ε can be achieved using the given N.

In this chapter we present the ARA. The proposed algorithm improves on the Rearrangement Algorithm on several points: first it addresses the problem of choosing the tail discretization parameter N iteratively. Second the convergence tolerance ε is replaced by two relative convergence tolerances, namely ε_1 and ε_2. The first relative tolerance, ε_1, is used to determine the individual convergence of each of the lower bound $\underline{s}_N$ and the upper bound $\overline{s}_N$, when computing the lower and upper bounds for $\overline{\mathrm{VaR}}_\alpha(L^+)$, while the second relative tolerance, ε_2, is used to control how far apart $\underline{s}_N$ and $\overline{s}_N$ are in each iteration of the algorithm. Moreover, the newly introduced convergence tolerance ε_2 along with ε_1 are used to define a new stopping criterion for the algorithm when computing the lower bound $\underline{s}_N$ and the upper bound $\overline{s}_N$ for $\overline{\mathrm{VaR}}_\alpha(L^+)$.

6.2 $\mathrm{VaR}_\alpha(L^+)$ Bounds, NP-Completeness and Computational Complexity

Before proceeding to the ARA, we look at the worst VaR problem from another angle. Minimizing the variance of the sum of random variables with given marginals is a classical problem in simulation and variance reduction, studied in Fishman [85] and Reiher [86] extensively. More formally assume that we are given a set of n random variables, X_i, $i = 1, \ldots, n$, each of which has a given marginal distribution F. We are interested in minimizing

the variance of the sum of these random variables, i.e.,

$$\min_{X_i \sim F} \text{Var}(X_1 + \ldots + X_n) \tag{6.2.1}$$

For $n = 2$ the above problem can be easily solved by using antithetic pairs $X_1 = F^-(U)$ and $X_2 = F^-(1-U)$ where F^- represents the inverse cumulative distribution function as before and U is a unif$(0, 1)$ random variable. However given an $n \geqslant 3$ and an arbitrary cumulative distribution function F, problem (6.2.1) becomes highly intractable.

An important question that can be asked here is for which cumulative distribution functions F, when $X_i \sim F$, we get a constant $\sum_{i=1}^n X_i$, i.e. what is the optimal value of the problem (6.2.1)? In addressing this question Wang and Wang [87] introduced the notion of *complete mixability* for the marginal distributions which can attain constant sum. More formally:

Definition 6.2.1. *Suppose that n is a positive integer. A probability distribution F on $\mathbb{R}$ is called* completely mixable with an index n *if there exist n random variables $X_1, \ldots, X_n$, each $X_i \sim F, \quad i = 1, \ldots, n$, such that $X_1 + \ldots + X_n$ is constant. In this case the distribution of $(X_1, \ldots, X_n)$ is called an n-*complete mix.

Using the above notion, Wang and Wang [87] provide bounds for the worst VaR in the case of homogenous marginals $F_1 = \ldots = F_n$.

Similarly, a matrix is called completely mixable if the entries of its columns can be permuted such that all the resulting row sums are equal. Considering a collection of permutations of an N-point discretization of the marginal quantile functions in the RA, finding a constant row sum is ideal when computing $\overline{s}_N$ and $\underline{s}_N$ defined in algorithm 5.5.

If a matrix is not completely mixable, determining the smallest maximal and largest minimal row sum that are attainable is of interest. The problem of estimating the α-quantile of the aggregate of random variables with an unknown dependence structure is connected to the discrete approximation of the bounds on $\text{VaR}_\alpha(L^+)$. Haus [84] points out that estimating the bounds on the α-quantile of the aggregate of random variables with an unknown dependence structure, when we are only given the marginal distributions of the underlying random variables (seen as estimating bounds on $\text{VaR}_\alpha(L^+)$), is related to the multidimensional bottleneck assignment problem and determining whether a given matrix is completely mixable is in general $\mathcal{NP}$-complete.

In addition to the above, Haus [84] also provides a counterexample for which the RA simply does not converge:

$$X = \begin{pmatrix} 1 & N & N \\ 2 & N-1 & N-1 \\ \vdots & \vdots & \vdots \\ N & 1 & 1 \end{pmatrix} \tag{6.2.2}$$

By running the RA on the input matrix (6.2.2), when we oppositely order each column to the row sums of the remaining columns, we will get the original matrix (6.2.2) after the

first iteration of the RA algorithm and no changes in the input matrix (6.2.2) will occur at the end of this process.

One suggestion is to iterate over all possible permutations of the columns of $N \times d$ matrices, derived from discretizing the tail of each of the d risk factors using N points to calculate $\overline{s}_N$ and $\underline{s}_N$ and obtain an estimate for $\overline{\mathrm{VaR}}_\alpha(L^+)$. This option is not practical: enumeration of the total number of possible matrices when we permute all but one column will result in $(N!)^{d-1}$ possible matrices. Consider a portfolio of 10 instruments ($d = 10$) and choose $N = 20$; the total number of the resulting matrices becomes $(20!)^9 \approxeq 10^{165}$. Note that the choices of both $N = 20$ and $d = 10$ are extremely conservative in this example since in practice N can easily be as large as 1,000 to 100,000 and many portfolios have at least 20 instruments. Therefore enumeration of the set of all possible candidate $N \times d$ matrices in both real-world applications and in the simple theoretical example (with for example $d = 10$ and $N = 20$) is nearly impossible.

The aforementioned problems motivate us to enhance the performance of the Rearrangement Algorithm.

6.3 How the ARA Works

In this section we present ARA for computing bounds for $\overline{\mathrm{VaR}}_\alpha(L^+)$. Recall from section 5.6 that the choice of two of the main input parameters in the RA, namely the discretization parameter N and the relative error ε impacts both the run-time and the accuracy of the resulting $\overline{\mathrm{VaR}}_\alpha(L^+)$ bounds. In order to choose initial values for these parameters, Puccetti and Rüschendorf [7] provide the following information:

- Discretization parameter N: it has been suggested that it should be chosen "large enough" but no further suggestion is provided.
- Relative error ε: the choice of this input is left to the user of the RA. However the choice of ε is not straightforward without a priori knowledge of the minimal row sums.

In addition to the issues mentioned above, an important problem that the RA does not address is the connection between the discretization parameter N and the relative error ε and how it impacts the overall run-time and the accuracy of attaining $\underline{s}_N$ and $\overline{s}_N$ when the $\overline{\mathrm{VaR}}_\alpha(L^+)$ bounds are estimated.

In what follows we present an algorithmically improved version of the RA, with more meaningful tuning parameters. Recall that given an input (N, d)-matrix $X = (x_{ij})$ (alternatively $\underline{X}^\alpha$ or $\overline{X}^\alpha$ as defined in algorithm 5.5),

$$s(X) = \min_{1 \leqslant i \leqslant N} \sum_{1 \leqslant j \leqslant d} x_{ij} \tag{6.3.1}$$

represented the minimum of the row sums of the input matrix. The adaptive rearrangement algorithm addresses the problems described above through an iterative process for choosing

the discretization parameter N, coupled with a new bivariate vector of the relative errors. More specifically,

- Discretization parameter N: it is chosen *adaptively*. Based on a pre-defined set of input $N \in \mathbf{N}$, the algorithm progresses through this set and chooses the most relevant N based on the convergence criterion, defined through a bivariate vector of the relative errors.
- Relative error ε is replaced by a bivariate vector of relative errors $\varepsilon = (\varepsilon_{\mathbf{1}}, \varepsilon_{\mathbf{2}})$:
  - Individual error ε_1 is used to ensure the *individual* convergence of the lower bound and upper bound, namely $\underline{s}_N$ and $\overline{s}_N$, in computing $\overline{\mathrm{VaR}}_\alpha(L^+)$ bounds.
  - Joint relative error ε_2 is used to ensure the *relative* convergence of the lower bound and upper bound in computing $\overline{\mathrm{VaR}}_\alpha(L^+)$ bounds.

For notational simplicity and given two $N \times d$ matrices X and Y let

$$\mathrm{err}_i(X, Y) = \left| \frac{s(X) - s(Y)}{s(X)} \right| \tag{6.3.2}$$

denote the individual error function. Similarly, using the function $s(\cdot)$ defined in equation (6.3.1) let

$$\mathrm{err}_j(\overline{s}_N, \underline{s}_N) = \left| \frac{\overline{s}_N - \underline{s}_N}{\overline{s}_N} \right| \tag{6.3.3}$$

be the joint relative error, where $\underline{s}_N = s(\underline{X}^\alpha)$ and $\overline{s} = s(\overline{X}^\alpha)$. Using the above notation, we introduce the following algorithm:

Algorithm 6.3: Adaptive Rearrangement Algorithm for Computing $\overline{\mathrm{VaR}}_\alpha(L^+)$

---

1. Assume that the chosen confidence level $\alpha \in (0,1)$, marginal quantile functions $F_1^-, \ldots, F_d^-$, an integer vector $\mathbf{K} \in \mathbb{N}^{\mathbf{l}}$, $l \in \mathbb{N}$, (containing the numbers of discretization points which are adaptively used), a bivariate vector of relative convergence tolerances $\varepsilon = (\varepsilon_{\mathbf{1}}, \varepsilon_{\mathbf{2}})$ (containing the individual relative tolerance $\varepsilon_1 > 0$ and the joint relative tolerance $\varepsilon_2 > 0$) and the maximal number of iterations used for each $k \in \mathbf{K}$ are given.
2. For $N = 2^k$, $k \in \mathbf{K}$, do:
   2.1. Compute the lower bound for $\overline{\mathrm{VaR}}_\alpha(L^+)$:
      2.1.1. Define the matrix $\underline{X}^\alpha = (\underline{x}_{ij}^\alpha)$ for $\underline{x}_{ij}^\alpha = F_j^-(\alpha + \frac{(1-\alpha)(i-1)}{N})$, $i \in \{1, \ldots, N\}$, $j \in \{1, \ldots, d\}$.
      2.1.2. Permute randomly the elements in each column of $\underline{X}^\alpha$.
      2.1.3. Set $\underline{Y}^\alpha = \underline{X}^\alpha$ and for $j \in \{1, 2, \ldots, d\}$, rearrange the jth column of the matrix $\underline{Y}^\alpha$ so that it becomes oppositely ordered to the sum of all other columns. Call the resulting matrix $\underline{Y}^\alpha$.

2.1.4. While the maximal number of the column rearrangements is not reached *and*

$$\mathrm{err}_i(\underline{X}^\alpha, \underline{Y}^\alpha) > \varepsilon_1 \tag{6.3.4}$$

set $\underline{X}^\alpha = \underline{Y}^\alpha$ and goto step 2.1.3..

2.1.5. Set $\underline{s}_N = s(\underline{Y}^\alpha)$.

2.2. Compute the upper bound for $\overline{\mathrm{VaR}}_\alpha(L^+)$:

2.2.1. Define the matrix $\overline{X}^\alpha = (\overline{x}^\alpha_{ij})$ for $\overline{x}^\alpha_{ij} = F_j^-(\alpha + \frac{(1-\alpha)i}{N})$, $i \in \{1, \ldots, N\}$, $j \in \{1, \ldots, d\}$. If (for $i = N$ and) for any $j \in \{1, \ldots, d\}$, $F_j^-(1) = \infty$, adjust it to $F_j^-(\alpha + \frac{(1-\alpha)(N-1/2)}{N})$.

2.2.2. Permute randomly the elements in each column of $\overline{X}^\alpha$.

2.2.3. Set $\overline{Y}^\alpha = \overline{X}^\alpha$. For $j \in \{1, 2, \ldots, d\}$, rearrange the jth column of the matrix $\overline{Y}^\alpha$ so that it becomes oppositely ordered to the sum of all other columns. Call the resulting matrix $\overline{Y}^\alpha$.

2.2.4. While the maximal number of the column rearrangements is not reached *and*

$$\mathrm{err}_i(\overline{X}^\alpha, \overline{Y}^\alpha) > \varepsilon_1 \tag{6.3.5}$$

set $\overline{X}^\alpha = \overline{Y}^\alpha$ and goto step 2.2.3..

2.2.5. Set $\overline{s}_N = s(\overline{Y}^\alpha)$.

2.3. Determine convergence based on both the individual and joint relative convergence tolerances:

If (6.3.4) and (6.3.5) hold, and if $\quad \mathrm{err}_j(\overline{s}_N, \underline{s}_N) \leqslant \varepsilon_2 \quad$ break.

3. Return $(\underline{s}_N,\ \overline{s}_N)$.

Concerning the choices of the input parameters in algorithm 6.3, note that if $\mathbf{K} = \{\mathbf{k} = \log_{\mathbf{2}} \mathbf{N}\}$, where we have a single number of discretization points, then the ARA reduces to an improved RA that uses more meaningful relative errors instead of absolute errors and not only checks what we termed *individual errors*, i.e., convergence of $\underline{s}_N$ and $\overline{s}_N$ individually, but also their *joint error*, i.e., the relative tolerance between $\underline{s}_N$ and $\overline{s}_N$ in computing the bounds for $\overline{\mathrm{VaR}}_\alpha(L^+)$.

Based on the simulation studies in section 5.6, useful defaults for the choice of $\mathbf{K}$ and the maximal number of iterations in algorithm 6.3 are $\mathbf{K} = \{\mathbf{8}, \mathbf{9}, \ldots, \mathbf{19}\}$ and $10d$ respectively. Given the high model uncertainty and the (often) rather large values of $\overline{\mathrm{VaR}}_\alpha(L^+)$ (especially in heavy-tailed test cases), a useful choice for the bivariate vector of relative errors ε is $\varepsilon = (\mathbf{0.001},\ \mathbf{0.01})$.

It is important to note that the choice of the powers of 2 for the discretization points N is purely for investigative purposes and for covering a wide range for this input in the studies conducted in sections 6.3.2, 6.3.3 and 6.3.4. Moreover, as we will show in sections 6.3.2, 6.3.3 and 6.3.4, the choices of both N and ε_2 have a significant impact on the run time of the ARA, but, using the results in section 6.4 and 6.5, as the number of risk factors

d increases, the importance of N is less pronounced compared to ε_2, i.e., determining whether or not the convergence criteria for the joint relative error, as specified by ε_2, is met, plays a more important role in meeting the final stopping criteria of the ARA when computing the lower bound and upper bound for $\overline{\text{VaR}}_\alpha(L^+)$ and hence in determining the overall run time of ARA.

6.3.1 Empirical Performance Under Various Setups

One of the main features of the ARA is the *dynamic* choice of the upper tail discretization parameter N. Therefore in defining the main studies in which we analyze the performance of the ARA, the choice of N plays no role. Instead we consider two main studies, the first one with $d = 20$ and the second with $d = 100$ risk factors . We consider the case of Pareto marginal distributions.

While this setup is similar to the one that we used in section 5.6 in chapter 5, we have modified it when testing the empirical performance of the ARA by using Student's t-distributions and log-normal distributions. More specifically, we consider three portfolios with $d = 20$, $d = 40$ and $d = 80$ for the Student's t-distribution and log-normal distribution studies, in order to analyze the impact of doubling the number of risk factors on the performance of the ARA. For each of these portfolios a similar setup for the bivariate vector $\varepsilon = (\varepsilon_1, \varepsilon_2)$ is used in which $\varepsilon_1 = 0.1\%$ and $\varepsilon_2 = 0.5\%$, 1% and 2% respectively.

In each of the three studies we present the mean and 95% confidence interval for the lower and upper bounds of $\overline{\text{VaR}}_{0.99}(L^+)$, the mean and 95% confidence interval for the N *used* for computing each of the lower and upper bounds, as well as run-time, number of oppositely ordered columns and number of iterations. In addition to that we have used the boxplots (also known as the box-and-whiskers plots) of the aforementioned inputs and outputs in each study to provide a graphical summary of their distributions. For example figure 6.1 shows the boxplot of the run-time and the N *used* when computing the $\overline{\text{VaR}}_{0.99}(L^+)$ bounds for $d = 100$ for Portfolio2 with Pareto marginals described in section 6.3.2.

The graph on the left in figure 6.1 shows the run-time of computing the $\overline{\text{VaR}}_{0.99}$ bounds using $B = 200$ replications of Algorithm 6.3 in which the mean, the first and third quartile and the whiskers are visible, demonstrating the overall distribution of the 200 run-times where each iteration has had a different stopping time. On the other hand, in the graph on the right in figure 6.1, which shows the boxplot of N *used* for computing the upper bound $\overline{s}_N$), since all iterations have used $N = 2^{15}$, there are no visible whiskers and outliers and the first and third quartiles have *collapsed* on the mean.

Another advantage of using boxplots is that since the boxplots of a given data set displays the median, the first and third quartiles (and hence the interquartile range of the data set), the minimum and maximum as well as the central tendency, range, symmetry, and presence of outliers, given a fixed number of risk factors ($d = 20, 40, 80$ or 100), and by side-by-side presentation of each of these boxplots (for each of the four portfolios in each study) we provide a visual comparison for the distributions of the parameter that is analyzed.

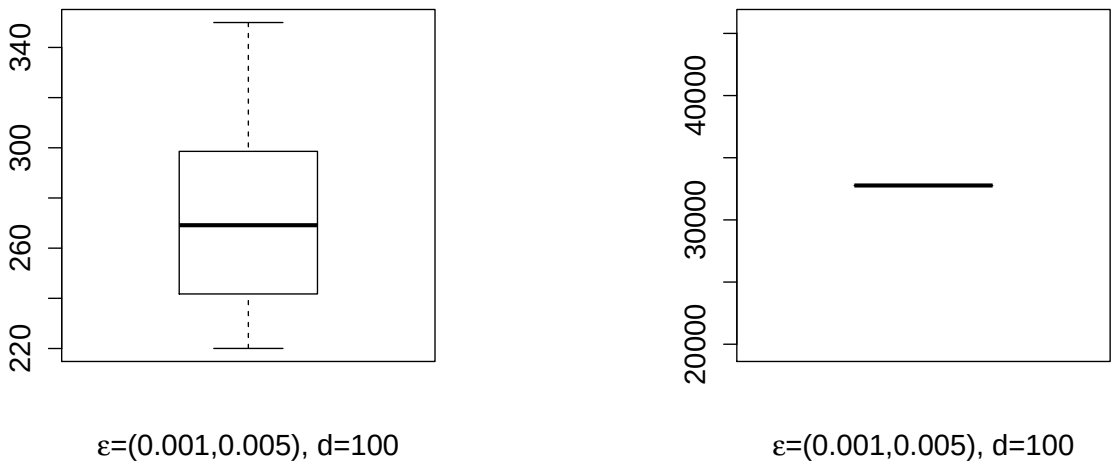


FIGURE 6.1: *Boxplots of the run-time in seconds (left-hand side) and* N used *(right-hand side, in computing the upper bound* $\overline{s}_N$*) based on* $B = 200$ *replications when computing Pareto distribution* $\overline{\mathrm{VaR}}_{0.99}$ *bounds with the ARA for Portfolio2.*

Moreover, to compute the confidence interval we have used the percentile bootstrap (more formally known as percentile bootstrap confidence intervals) methodology. This methodology uses the percentiles of the bootstrap distribution for a given statistic to compute the confidence interval (see Davison and Hinkley [88, Equation 5.18]) at the confidence level α. For example, using $B = B_0$ bootstrapped values for the run-time, namely $t_1, t_2, \ldots, t_{B_0-1}, t_{B_0}$, define the ordered run-times as:

$$t^*_{(1)} \leqslant t^*_{(2)} \leqslant \ldots \leqslant t^*_{(B_0-1)} \leqslant t^*_{(B_0)}$$

To determine the lower and upper $\alpha/2$ points for the empirical distribution of the $B = B_0$ bootstrapped run-times, let

$$i_l = \left\lfloor \frac{\alpha}{2}(B_0 + 1) \right\rfloor \quad \text{and} \quad i_u = (B_0 + 1) - i_l$$

then the interval $[t^*_{i_l}, t^*_{i_u}]$ contains the middle $(1 - \alpha)$ fraction for the empirical distribution of the bootstrapped run-times.

6.3.2 Generalized Pareto Distribution Results

Under each study (corresponding to $d = 20$ and $d = 100$ respectively) we investigate 12 examples. Each example consists of one the following 4 portfolios, where the Pareto(θ_j)-distributed losses are defined by the cumulative distribution function

$$F_j(x) = 1 - (1 + x)^{-\theta_j}, \quad x \geqslant 0,$$

for a given tail parameter $\theta_j > 0$.

Portfolio1: $\theta_1, \ldots, \theta_d$ are uniformly chosen from 0.6 to 0.4. This case represents a portfolio with marginals with a similar tail behavior (i.e., a heavy-tailed distribution).

Portfolio2: $\theta_1, \ldots, \theta_d$ are uniformly chosen from 0.5 to 1.5. This case represents a portfolio with marginals with differing tail behavior (i.e., from a very heavy-tailed distribution to a not so heavy-tailed distribution).

Portfolio3: $\theta_1, \ldots, \theta_d$ are uniformly chosen from 1.4 to 1.6. This case represents a portfolio with marginals with similar tail behavior (i.e., a not so heavy-tailed distribution).

Portfolio4: $\theta_2, \ldots, \theta_d$ are chosen as in Portfolio3 and $\theta_1 = 0.5$. This case represents a portfolio with marginals with only one heavy-tailed loss distribution.

Each of the above portfolios represent different marginal tail behaviors based on the Pareto distribution, coupled with one of the following three choices of the bivariate vector of relative errors $\varepsilon = (\varepsilon_{\mathbf{1}}, \varepsilon_{\mathbf{2}})$ defined as:

Case 1: $\varepsilon = (\mathbf{0.1}\%, \mathbf{0.5}\%)$, i.e., individual relative error ε_1 is chosen as 0.1% and the joint relative error ε_2 is 0.5%.

Case 2: $\varepsilon = (\mathbf{0.1}\%, \mathbf{1}\%)$, i.e., individual relative error ε_1 is the same as before and the joint relative error ε_2 is twice the previous case.

Case 3: $\varepsilon = (\mathbf{0.1}\%, \mathbf{2}\%)$, i.e., individual relative error ε_1 is kept at the same level and again we have doubled the joint relative error ε_2.

Therefore the performance of the ARA is investigated in 24 different examples. As before, the results shown for each test are based on $B = 200$ simulations and we report the mean and 95% confidence interval for the lower and upper bounds of $\overline{\text{VaR}}_{0.99}$, the N *used* in the final iteration of the ARA, mean and confidence interval for each example in table (6.1) as well as the number of oppositely ordered columns and number of iterations for each example in table (6.2). Our findings indicate that:

- In table 6.1, we see that although in both studies ($d = 20$ and $d = 100$), the length of the confidence intervals for $\overline{\text{VaR}}_{0.99}$ increases as the joint relative error, ε_2 increases, the mean and lower and upper confidence bounds for $\overline{\text{VaR}}_{0.99}$ remain close to each other and within these bounds. Moreover for a fixed level of individual relative error, as ε_2 increases, we do not observe a drastic shift in both lower and upper bounds for the mean across different examples.

- The mean and the 95% confidence interval for the N *used* in the final iteration of the ARA is shown in table 6.2. An important observation about this parameter is that in virtually all examples, both the upper and lower bounds for the 95% confidence interval of N *used* remain the same. This observation can be used in practice for

portfolios which exhibit the same marginal tail behavior to reduce the run-time of the ARA. For example when the underlying risk factors in the portfolio are similar to those which we defined earlier in the case of Pareto(θ_j)-distributed marginals, and for a fixed bivariate vector of relative errors $\varepsilon = (\varepsilon_1, \varepsilon_2)$, $\mathbf{K} = \mathbf{11}, \ldots, \mathbf{17}$ can be used in algorithm 6.3.

- The last column in table (6.1) shows the mean run-time of the ARA in each of the 24 examples. Doubling the joint relative error reduces the run-time more than 50% across all examples.

- The mean number of iterations when computing $\underline{s}_N$ and $\overline{s}_N$, as well as both the lower and upper bounds consistently remain below 5.

- Finally figures 6.2, 6.3, 6.4, 6.5, and 6.6 show the impact of the randomization of the initial input matrix X. As can be seen from our simulations, randomizing the input has minimal impact on the outputs and parameters such as *N-used* in the final iteration of the ARA in each of these 24 examples.

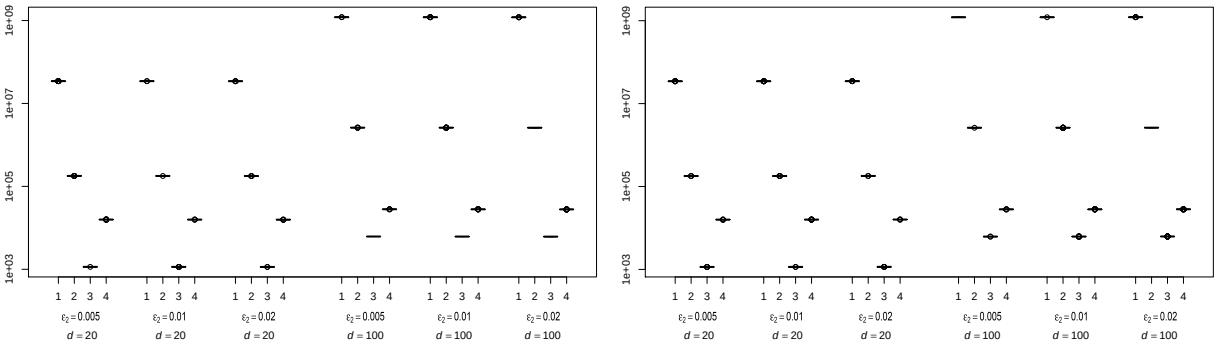


FIGURE 6.2: *Boxplots of lower (left-hand side) and upper (right-hand side) Pareto distribution* $\overline{\mathrm{VaR}}_{0.99}$ *bounds computed with the ARA based on* $B = 200$ *replications.*

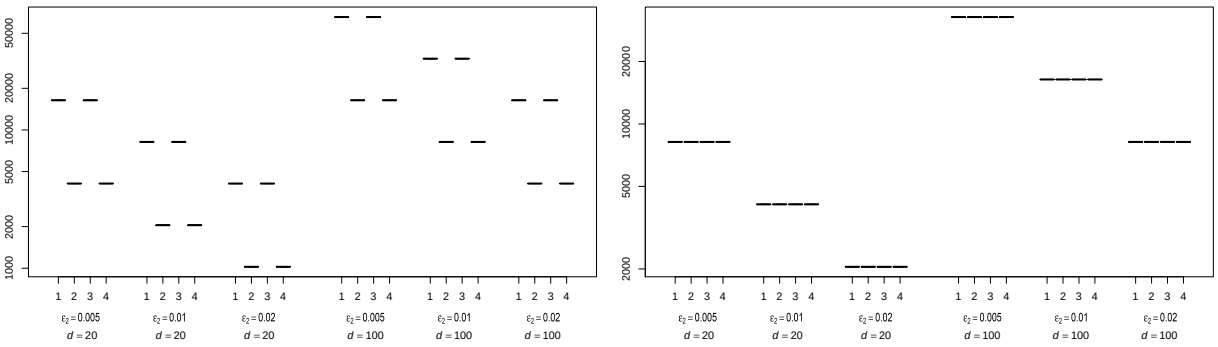


FIGURE 6.3: *Boxplots of the actual N used for computing lower (left-hand side) and upper (right-hand side) Pareto distribution* $\overline{\mathrm{VaR}}_{0.99}$ *bounds with the ARA based on* $B = 200$ *replications.*

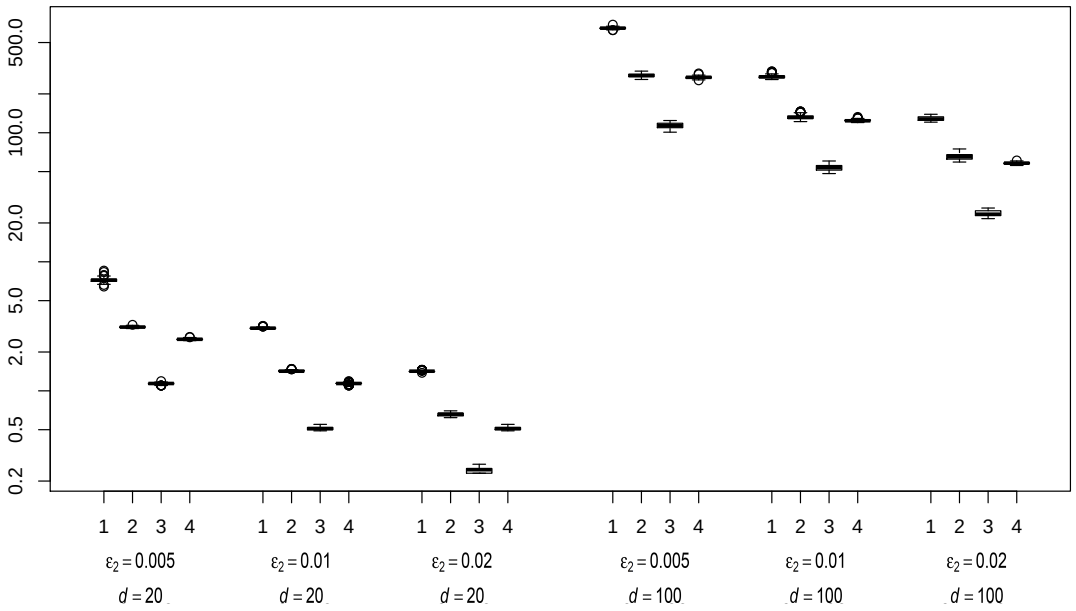


FIGURE 6.4: *Boxplots of the run-time in seconds for computing Pareto distribution* $\overline{\text{VaR}}_{0.99}$ *bounds with the ARA based on* $B = 200$ *replications.*

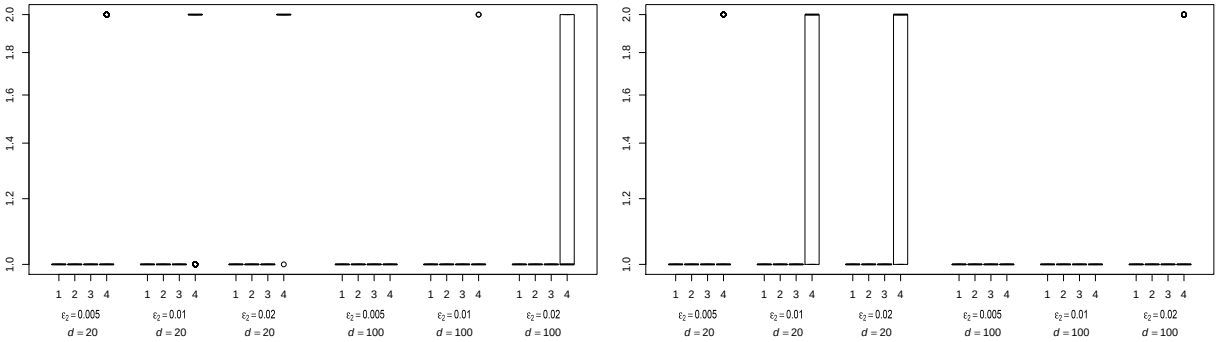


FIGURE 6.5: *Boxplots of the number of oppositely ordered columns for computing lower (left-hand side) and upper (right-hand side) Pareto distribution* $\overline{\text{VaR}}_{0.99}$ *bounds with the ARA based on* $B = 200$ *replications.*

6.3.3 Student's t-distribution Analysis

Besides the Pareto case, we test the performance of the ARA using marginals that are Student's t-distributed. Student's t-distribution is one of the most commonly used fat-tailed distributions in risk management and its probability density function is given by:

$$f(x) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\Gamma(\frac{\nu}{2})}\left[1 + \frac{x^2}{\nu}\right]^{-\frac{\nu+1}{2}}$$

where ν is the degrees of freedom and $\Gamma(\cdot)$ represents the gamma function. It is well known that the degrees of freedom parameter governs the tail behavior of this distribution. The smaller the value of ν, the heavier the tail becomes. As $\nu \to \infty$ Students's t-

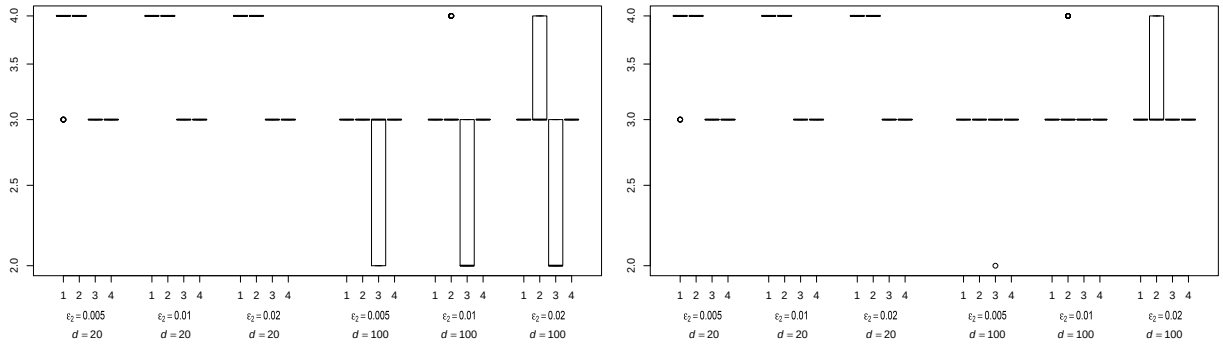


FIGURE 6.6: *Boxplots of the number of iterations for computing lower (left-hand side) and upper (right-hand side) Pareto distribution* $\overline{\text{VaR}}_{0.99}$ *bounds with the ARA based on* $B = 200$ *replications.*

			$\overline{\text{VaR}}_{0.99}$			
			Lower		Upper	
d	ε (in %)	Case	Mean	95% CI	Mean	95% CI
20	(0.1, 0.5)	1	3.4592e7	(3.4559e7, 3.4560e7)	3.4653e7	(3.4653e7, 3.4654e7)
		2	1.7857e5	(1.7857e5, 1.7857e5)	1.7916e5	(1.7916e5, 1.7917e5)
		3	1.1446e3	(1.1446e3, 1.1446e3)	1.1484e3	(1.1484e3, 1.1484e3)
		4	1.5839e4	(1.5839e4, 1.5840e4)	1.5905e4	(1.5905e4, 1.5905e4)
	(0.1, 1)	1	3.4513e7	(3.4513e7, 3.4513e7)	3.4700e7	(3.4700e7, 3.4700e7)
		2	1.7827e5	(1.7827e4, 1.7828e4)	1.7938e5	(1.7937e5, 1.7939e5)
		3	1.1426e3	(1.1426e3, 1.1427e3)	1.1530e3	(1.1503e3, 1.1503e3)
		4	1.5807e4	(1.5807e4, 1.5807e4)	1.5938e4	(1.5938e4, 1.5938e4)
	(0.1, 2)	1	3.4420e7	(3.4419e7, 3.4420e7)	3.4793e7	(3.4793e7, 3.4794e7)
		2	1.7773e5	(1.7772e5, 1.7774e5)	1.7978e5	(1.7976e5, 1.7979e5)
		3	1.1389e3	(1.1388e3, 1.1389e3)	1.1542e3	(1.1541e3, 1.1542e3)
		4	1.5739e4	(1.5738e4, 1.5739e4)	1.5991e4	(1.5990e4, 1.5991e4)
100	(0.1, 0.5)	1	1.2054e9	(1.2054e9, 1.2054e9)	1.2095e9	(1.2095e9, 1.2095e9)
		2	2.6073e7	(2.6073e7, 2.6074e7)	2.6162e7	(2.6162e7, 2.6163e7)
		3	6.1760e3	(6.1759e3, 6.1761e3)	6.2018e3	(6.2018e3, 6.2018e3)
		4	2.8035e4	(2.8035e4, 2.8035e4)	2.8156e4	(2.8156e4, 2.8156e4)
	(0.1, 1)	1	1.2034e9	(1.2033e9, 1.2034e9)	1.2116e9	(1.2116e9, 1.2116e9)
		2	2.6029e7	(2.6028e7, 2.6032e7)	2.6194e7	(2.6194e7, 2.6196e7)
		3	6.1631e3	(6.1630e3, 6.1632e3)	6.2148e3	(6.2148e3, 6.2148e3)
		4	2.7972e4	(2.7972e4, 2.7972e4)	2.8212e4	(2.8212e4, 2.8212e4)
	(0.1, 2)	1	1.1992e9	(1.1992e9, 1.1993e9)	1.2157e9	(1.2157e9, 1.2157e9)
		2	2.5950e7	(2.5948e7, 2.5952e7)	2.6252e7	(2.6250e7, 2.6254e7)
		3	6.1379e3	(6.1378e3, 6.1380e3)	6.2412e3	(6.2411e3, 6.2412e3)
		4	2.7853e4	(2.7853e4, 2.7853e4)	2.8278e4	(2.8278e4, 2.8278e4)

TABLE 6.1: *Pareto distribution* $\overline{\text{VaR}}_{0.99}$ *bounds with the ARA based on* $B = 200$ *bootstrap replications.*

distribution converges to the normal distribution. The following remark (see McNeil et al. [65, Remark 2.22]) formalizes this connection in the context of quantitative risk management.

Remark 6.3.1. *It is possible to derive results on the asymptotics of the shortfall to*

			N used					
			Lower		Upper		Time (in s)	
d	ε (in %)	Case	Mean	95% CI	Mean	95% CI	Mean	95% CI
20	(0.1, 0.5)	1	16384	(16384, 16384)	8192	(8192, 8192)	7.21	(6.79, 7.76)
		2	4096	(4096, 4096)	8192	(8192, 8192)	3.12	(3.06, 3.20)
		3	16384	(16384, 16384)	8192	(8192, 8192)	1.14	(1.10, 1.17)
		4	4096	(4096, 4096)	8192	(8192, 8192)	2.51	(2.45, 2.57)
	(0.1, 1)	1	8192	(8192, 8192)	4096	(4096, 4096)	3.06	(3.02, 3.13)
		2	2048	(2048, 2048)	4096	(4096, 4096)	1.42	(1.40, 1.46)
		3	8192	(8192, 8192)	4096	(4096, 4096)	0.51	(0.49, 0.54)
		4	2048	(2048, 2048)	4096	(4096, 4096)	1.14	(1.11, 1.17)
	(0.1, 2)	1	4096	(4096, 4096)	2048	(2048, 2048)	1.41	(1.40, 1.45)
		2	1024	(1024, 1024)	2048	(2048, 2048)	0.65	(0.64, 0.69)
		3	4096	(4096, 4096)	2048	(2048, 2048)	0.24	(0.23, 0.26)
		4	1024	(1024, 1024)	2048	(2048, 2048)	0.51	(0.49, 0.55)
100	(0.1, 0.5)	1	65536	(65536, 65536)	32768	(32768, 32768)	647.25	(629.52, 666.05)
		2	16384	(16384, 16384)	32768	(32768, 32768)	278.24	(263.75, 295.46)
		3	65536	(65536, 65536)	32768	(32768, 32768)	113.72	(104.28, 122.44)
		4	16384	(16384, 16384)	32768	(32768, 32768)	268.64	(259.07, 279.24)
	(0.1, 1)	1	32768	(32768, 32768)	16384	(16384, 16384)	271.93	(261.79, 292.40)
		2	8192	(8192, 8192)	16384	(16384, 16384)	132.27	(125.50, 143.53)
		3	32768	(32768, 32768)	16384	(16384, 16384)	53.62	(49.04, 58.60)
		4	8192	(8192, 8192)	16384	(16384, 16384)	124.57	(120.94, 129.17)
	(0.1, 2)	1	16384	(16384, 16384)	8192	(8192, 8192)	128.66	(121.98, 136.47)
		2	4096	(4096, 4096)	8192	(8192, 8192)	65.48	(60.17, 74.24)
		3	16384	(16384, 16384)	8192	(8192, 8192)	23.73	(22.16, 25.74)
		4	4096	(4096, 4096)	8192	(8192, 8192)	58.15	(56.45, 60.14)

TABLE 6.2: *N-used and run-time in computing $\overline{\mathrm{VaR}}_{0.99}$ with the ARA based on $B = 200$ bootstrap replications for the Pareto distribution.*

quantile ratio $\mathrm{CVaR}_\alpha(L)/\mathrm{VaR}_\alpha(L)$ *as* $\alpha \to 1$. *For the normal distribution we have:*

$$\lim_{\alpha\to 1} \frac{\mathrm{CVaR}_\alpha(L)}{\mathrm{VaR}_\alpha(L)} = 1$$

while for the Student's t-distribution with $\nu > 1$ *degrees of freedom we have*

$$\lim_{\alpha\to 1} \frac{\mathrm{CVaR}_\alpha(L)}{\mathrm{VaR}_\alpha(L)} = \frac{\nu}{\nu - 1}$$

Remark 6.3.1 indicates that, given a more heavy-tailed distribution, the difference between $\mathrm{CVaR}_\alpha(L)$ and $\mathrm{VaR}_\alpha(L)$ becomes more pronounced compared to the normal distribution.

It is important to note that using realistic estimates of the degrees of freedom parameter that governs extreme events essential in analyzing the performance of the ARA. Jorion [89] indicates that values of ν ranging from 4 to 7 for the Student's t-distribution are a good representative of the tail behavior of financial data. Besides this study, tail behavior exhibits further diversity: according to Stoyanov et al. [90] 27% of all stocks have a $\nu > 7$ while 13% of them have $\nu < 4$. In a similar study Rachev et al. [91] show that 21% of the S&P 500 stocks are very fat-tailed, with a $\nu < 4$, while 35% have $\nu > 7$.

			Number of oppositely ordered columns				Number of iterations			
			Lower		Upper		Lower		Upper	
d	ε (in %)	Case	Mean	95% CI	Mean	95% CI	Mean	95% CI	Mean	95% CI
20	(0.1, 0.5)	1	1	(1,1)	1	(1,1)	3.975	(3.975,4)	3.98	(4,4)
		2	1	(1,1)	1	(1,1)	4	(4,4)	4	(4,4)
		3	1	(1,1)	1	(1,1)	3	(3,3)	3	(3,3)
		4	1.195	(1,2)	1.12	(1,2)	3	(3,3)	3	(3,3)
	(0.1, 1)	1	1	(1,1)	1	(1,1)	4	(4,4)	4	(4,4)
		2	1	(1,1)	1	(1,1)	4	(4,4)	4	(4,4)
		3	1	(1,1)	1	(1,1)	3	(3,3)	3	(3,3)
		4	1.795	(1,2)	1.715	(1,2)	3	(3,3)	3	(3,3)
	(0.1, 2)	1	1	(1,1)	1	(1,1)	4	(4,4)	4	(4,4)
		2	1	(1,1)	1	(1,1)	4	(4,4)	4	(4,4)
		3	1	(1,1)	1	(1,1)	3	(3,3)	3	(3,3)
		4	1.995	(2,2)	1.67	(1,2)	3	(3,3)	3	(3,3)
100	(0.1, 0.5)	1	1	(1,1)	1	(1,1)	3	(3,3)	3	(3,3)
		2	1	(1,1)	1	(1,1)	3	(3,3)	3	(3,3)
		3	1	(1,1)	1	(1,1)	2.585	(2,3)	2.995	(3,3)
		4	1	(1,1)	1	(1,1)	3	(3,3)	3	(3,3)
	(0.1, 1)	1	1	(1,1)	1	(1,1)	3	(3,3)	3	(3,3)
		2	1	(1,1)	1	(1,1)	3.045	(3,4)	3.05	(3,4)
		3	1	(1,1)	1	(1,1)	2.49	(2,3)	3	(3,3)
		4	1.01	(1,1)	1	(1,1)	3	(3,3)	3	(3,3)
	(0.1, 2)	1	1	(1,1)	1	(1,1)	3	(3,3)	3	(3,3)
		2	1	(1,1)	1	(1,1)	3.305	(3,4)	3.265	(3,4)
		3	1	(1,1)	1	(1,1)	2.36	(2,3)	3	(3,3)
		4	1.45	(1,2)	1.07	(1,2)	3	(3,3)	3	(3,3)

TABLE 6.3: *Number of oppositely ordered columns and number of iterations in computing* $\overline{\mathrm{VaR}}_{0.99}$ *with the ARA based on* $B = 200$ *replications for the Pareto distribution.*

Based on the above results, we have considered the following ranges for the tail parameter $\nu_j > 0$.

Portfolio 1: $\nu_1, \ldots, \nu_d$ are uniformly chosen from 1 to 3. This case represents a portfolio with marginals with similar tail behavior (i.e., a heavy-tailed distribution).

Portfolio 2: $\nu_1, \ldots, \nu_d$ are uniformly chosen from 1 to 8. This case represents a portfolio with marginals with differing tail behavior (i.e., from a very heavy-tailed distribution to a not so heavy-tailed distribution).

Portfolio 3: $\nu_1, \ldots, \nu_d$ are uniformly chosen from 6 to 8. This case represents a portfolio with marginals with similar tail behavior (i.e., a not so heavy-tailed distribution).

Portfolio 4: $\nu_2, \ldots, \nu_d$ are chosen as in Portfolio 3 and $\nu_1 = 2$; this case represents a portfolio with only one heavy-tailed marginal loss distribution.

In each of the portfolios described in 6.3.3, three bivariate vectors of relative errors $\varepsilon = (\varepsilon_{\mathbf{1}}, \varepsilon_{\mathbf{2}})$ are used:

Case 1: $\varepsilon = (\mathbf{0.1}\%, \mathbf{0.5}\%)$, i.e., individual relative error ε_1 is chosen as 0.1% and the joint relative error ε_2 is 0.5%.

Case 2: $\varepsilon = (\mathbf{0.1}\%, \mathbf{1}\%)$, i.e., individual relative error ε_1 is the same as before and the joint relative error ε_2 is twice the previous case.

Case 3: $\varepsilon = (\mathbf{0.1}\%, \mathbf{2}\%)$, i.e., individual relative error ε_1 is kept at the same level and we have doubled the joint relative error ε_2.

			$\overline{\text{VaR}}_{0.99}$			
			Lower		Upper	
d	ε (in %)	Case	Mean	95% CI	Mean	95% CI
20	(0.1, 0.5)	1	515.806	(515.804, 515.808)	517.335	(517.333, 517.337)
		2	218.368	(218.366, 218.370)	219.435	(219.433, 219.436)
		3	75.519	(75.518, 75.519)	75.802	(75.802, 75.803)
		4	83.640	(83.640, 83.641)	83.966	(83.965, 83.967)
	(0.1, 1)	1	515.053	(515.050, 515.056)	517.908	(517.904, 517.911)
		2	217.847	(217.844, 217.849)	219.812	(219.808, 219.814)
		3	75.383	(75.382, 75.384)	75.869	(75.869, 75.870)
		4	83.484	(83.483, 83.485)	84.053	(84.051, 84.054)
	(0.1, 2)	1	511.154	(511.146, 511.160)	520.690	(520.671, 520.697)
		2	216.924	(216.919, 216.928)	220.453	(220.447, 220.457)
		3	75.383	(75.382, 75.384)	75.869	(75.869, 75.870)
		4	83.484	(83.483, 83.485)	84.053	(84.051, 84.054)
40	(0.1, 0.5)	1	1086.709	(1086.708, 1086.710)	1091.982	(1091.980, 1091.983)
		2	434.532	(434.531, 434.533)	436.159	(436.158, 436.160)
		3	151.010	(151.009, 151.010)	151.577	(151.577, 151.577)
		4	159.846	(159.846, 159.847)	160.482	(160.481, 160.482)
	(0.1, 1)	1	1084.163	(1084.161, 1084.164)	1093.785	(1093.783, 1093.787)
		2	433.754	(433.752, 433.755)	436.719	(436.717, 436.720)
		3	150.741	(150.740, 150.741)	151.712	(151.712, 151.713)
		4	159.544	(159.543, 159.544)	160.653	(160.652, 160.653)
	(0.1, 2)	1	1079.571	(1079.568, 1079.573)	1096.877	(1096.867, 1096.881)
		2	432.329	(432.327, 432.330)	437.691	(437.689, 437.693)
		3	150.741	(150.740, 150.741)	151.712	(151.712, 151.713)
		4	159.544	(159.543, 159.544)	160.653	(160.652, 160.653)
80	(0.1, 0.5)	1	2285.346	(2285.345, 2285.346)	2295.125	(2295.124, 2295.125)
		2	882.010	(882.010, 882.011)	884.672	(884.672, 884.673)
		3	301.992	(301.992, 301.992)	303.126	(303.126, 303.127)
		4	311.300	(311.300, 311.300)	312.558	(312.557, 312.558)
	(0.1, 1)	1	2280.631	(2280.630, 2280.632)	2298.357	(2298.355, 2298.358)
		2	878.40 4	(878.402, 878.404)	887.109	(887.108, 887.110)
		3	301.456	(301.456, 301.456)	303.399	(303.399, 303.399)
		4	310.701	(310.700, 310.701)	312.895	(312.895, 312.895)
	(0.1, 2)	1	2272.161	(2272.160, 2272.163)	2303.854	(2303.853, 2303.855)
		2	874.316	(874.313, 874.319)	889.733	(889.731, 889.734)
		3	301.456	(301.456, 301.456)	303.399	(303.399, 303.399)
		4	310.701	(310.700, 310.701)	312.895	(312.895, 312.895)

TABLE 6.4: *Student's $t-$distribution $\overline{\text{VaR}}_{0.99}$ bounds with the ARA based on $B = 200$ bootstrap replications.*

The above setup for the student's t-distribution results in 36 different examples, each of which is created using $B = 200$ simulations, and as before we report the mean and 95% confidence interval for the lower and upper bounds of $\overline{\text{VaR}}_{0.99}$ for each example in table 6.4. The mean and 95% confidence interval for the N *used* in computing each of the lower and upper bounds, as well as the total run-time's mean and 95% confidence interval are

			N used					
			Lower		Upper		Time (in s)	
d	ε (in %)	Case	Mean	95% CI	Mean	95% CI	Mean	95% CI
20	(0.1, 0.5)	1	4096	(4096, 4096)	1024	(1024, 1024)	1.97	(1.94, 2.24)
		2	512	(512, 512)	512	(512, 512)	0.44	(0.42, 0.47)
		3	4096	(4096, 4096)	1024	(1024, 1024)	0.19	(0.17, 0.21)
		4	512	(512, 512)	512	(512, 512)	0.19	(0.17, 0.21)
	(0.1, 1)	1	2048	(2048, 2048)	512	(512, 512)	0.96	(0.94, 1)
		2	256	(256, 256)	256	(256, 256)	0.20	(0.19, 0.21)
		3	2048	(2048, 2048)	512	(512, 512)	0.07	(0.05, 0.08)
		4	256	(256, 256)	256	(256, 256)	0.07	(0.05, 0.09)
	(0.1, 2)	1	512	(512, 512)	256	(256, 256)	0.21	(0.19, 0.22)
		2	256	(256, 256)	256	(256, 256)	0.07	(0.06, 0.09)
		3	512	(512, 512)	256	(256, 256)	0.07	(0.05, 0.08)
		4	256	(256, 256)	256	(256, 256)	0.07	(0.05, 0.08)
40	(0.1, 0.5)	1	4096	(4096, 4096)	2048	(2048, 2048)	5.40	(5.30, 5.70)
		2	512	(512, 512)	512	(512, 512)	2.48	(2.41, 2.78)
		3	4096	(4096, 4096)	2048	(2048, 2048)	0.47	(0.44, 0.48)
		4	512	(512, 512)	512	(512, 512)	0.46	(0.44, 0.48)
	(0.1, 1)	1	2048	(2048, 2048)	1024	(1024, 1024)	2.53	(2.47, 2.81)
		2	256	(256, 256)	256	(256, 256)	1.17	(1.13, 1.22)
		3	2048	(2048, 2048)	1024	(1024, 1024)	0.16	(0.14, 0.18)
		4	256	(256, 256)	256	(256, 256)	0.16	(0.14, 0.18)
	(0.1, 2)	1	1024	(1024, 1024)	512	(512, 512)	1.18	(1.16, 1.22)
		2	256	(256, 256)	256	(256, 256)	0.52	(0.50, 0.55)
		3	1024	(1024, 1024)	512	(512, 512)	0.16	(0.14, 0.18)
		4	256	(256, 256)	256	(256, 256)	0.16	(0.14, 0.18)
80	(0.1, 0.5)	1	8192	(8192, 8192)	4096	(4096, 4096)	36.58	(35.78, 37.51)
		2	512	(512, 512)	512	(512, 512)	16.86	(16.32, 17.24)
		3	8192	(8192, 8192)	4096	(4096, 4096)	1.26	(1.23, 1.31)
		4	512	(512, 512)	512	(512, 512)	1.26	(1.23, 1.31)
	(0.1, 1)	1	4096	(4096, 4096)	1024	(1024, 1024)	16.91	(16.68, 17.17)
		2	256	(256, 256)	256	(256, 256)	3.45	(3.38, 3.79)
		3	4096	(4096, 4096)	1024	(1024, 1024)	0.44	(0.42, 0.46)
		4	256	(256, 256)	256	(256, 256)	0.44	(0.42, 0.46)
	(0.1, 2)	1	2048	(2048, 2048)	512	(512, 512)	7.76	(7.60, 8.07)
		2	256	(256, 256)	256	(256, 256)	1.48	(1.44, 1.84)
		3	2048	(2048, 2048)	512	(512, 512)	0.44	(0.42, 0.46)
		4	256	(256, 256)	256	(256, 256)	0.44	(0.42, 0.45)

TABLE 6.5: *N-used and run-time in computing Student's t-distribution $\overline{\text{VaR}}_{0.99}$ with the ARA based on $B = 200$ bootstrap replications.*

shown in table 6.5. Similar statistics have been reported for the number of oppositely ordered columns and number of iterations in table 6.6. Based on the simulation results:

- The mean and the 95% confidence interval reported in table 6.4 indicate that across all studies ($d = 20$, $d = 40$ and $d = 80$), while the length of the confidence intervals for $\overline{\text{VaR}}_{0.99}$ increases as the joint relative error, ε_2, and d increase, the mean and lower and upper confidence bounds for $\overline{\text{VaR}}_{0.99}$ do not deviate from the mean and remain very close to the observed mean. Moreover table 6.4 shows that as we double d across examples, the length of the confidence intervals remains relatively stable.

- *N used* in the final iteration of the ARA is shown in table 6.5. Similar to what

			Number of oppositely ordered columns				Number of iterations			
			Lower		Upper		Lower		Upper	
d	ε (in %)	Case	Mean	95% CI	Mean	95% CI	Mean	95% CI	Mean	95% CI
20	(0.1, 0.5)	1	1	(1, 1)	1	(1, 1)	3	(3, 3)	3	(3, 3)
		2	1	(1, 1)	1	(1, 1)	3	(3, 3)	3	(3, 3)
		3	5.595	(2, 10)	4.425	(1, 9)	2	(2, 2)	2	(2, 2)
		4	2.51	(1, 7.02)	2.055	(1, 3.050)	2.19	(2, 3)	2.15	(2, 3)
	(0.1, 1)	1	1	(1, 1)	1	(1, 1)	3	(3, 3)	3	(3, 3)
		2	1.055	(1, 2)	1.005	(1, 1)	3	(3, 3)	3	(3, 3)
		3	10.855	(5, 16)	9.745	(4.975, 15)	2	(2, 2)	2	(2, 2)
		4	4.01	(2, 15)	4.07	(1.975, 14)	2.215	(2, 3)	2.25	(2, 3)
	(0.1, 2)	1	1.025	(1, 1.025)	1.08	(1, 2)	3	(3, 3)	3	(3, 3)
		2	2.645	(1, 4)	1.175	(1, 2)	3	(3, 3)	3	(3, 3)
		3	10.855	(5, 16)	9.745	(4.975, 15)	2	(2, 2)	2	(2, 2)
		4	4.01	(2, 15)	4.07	(1.975, 14)	2.215	(2, 3)	2.25	(2, 3)
40	(0.1, 0.5)	1	1	(1, 1)	1	(1, 1)	3	(3, 3)	3	(3, 3)
		2	1	(1, 1)	1	(1, 1)	3	(3, 3)	3	(3, 3)
		3	29.17	(21, 36)	29.13	(22, 36.025)	2	(2, 2)	2	(2, 2)
		4	3.75	(2, 11)	4.045	(2, 12)	2	(2, 2)	2	(2, 2)
	(0.1, 1)	1	1	(1, 1)	1	(1, 1)	3	(3, 3)	3	(3, 3)
		2	1	(1, 1)	1	(1, 1)	3	(3, 3)	3	(3, 3)
		3	34.445	(28.975, 39)	34.585	(29, 39)	2	(2, 2)	2	(2, 2)
		4	15.04	(2.975, 30)	14.095	(2, 26)	2	(2, 2)	2	(2, 2)
	(0.1, 2)	1	1.025	(1, 1.025)	1	(1, 1)	3	(3, 3)	3	(3, 3)
		2	1.205	(1, 2)	1.155	(1, 2)	3	(3, 3)	3	(3, 3)
		3	34.445	(28.975, 39)	34.585	(29, 39)	2	(2, 2)	2	(2, 2)
		4	15.04	(2.975, 30)	14.095	(2, 26)	2	(2, 2)	2	(2, 2)
80	(0.1, 0.5)	1	1	(1, 1)	1	(1, 1)	3	(3, 3)	3	(3, 3)
		2	1	(1, 1)	1	(1, 1)	3	(3, 3)	3	(3, 3)
		3	74.57	(69, 79)	74.155	(68, 79)	2	(2, 2)	2	(2, 2)
		4	46.825	(12.95, 64.025)	45.735	(15.85, 62)	2	(2, 2)	2	(2, 2)
	(0.1, 1)	1	1	(1, 1)	1	(1, 1)	3	(3, 3)	3	(3, 3)
		2	1.025	(1, 1.025)	1.015	(1, 1)	3	(3, 3)	3	(3, 3)
		3	77.15	(71.95, 80)	77.015	(71.975, 80)	2	(2, 2)	2	(2, 2)
		4	64.745	(42.95, 76)	63.93	(46, 74)	2	(2, 2)	2	(2, 2)
	(0.1, 2)	1	1	(1, 1)	1	(1, 1)	3	(3, 3)	3	(3, 3)
		2	1.255	(1, 4.025)	2.235	(1, 7)	3	(3, 3)	3	(3, 3)
		3	77.15	(71.95, 80)	77.015	(71.975, 80)	2	(2, 2)	2	(2, 2)
		4	64.745	(42.95, 76)	63.93	(46, 74)	2	(2, 2)	2	(2, 2)

TABLE 6.6: *Number of oppositely ordered columns and number of iterations in computing Student's t-distribution* $\overline{\text{VaR}}_{0.99}$ *with the ARA based on* $B = 200$ *replications.*

we observed in the case of Pareto marginals, in all examples, the upper and lower bounds for the 95% confidence interval of N *used* remain the same. This observation can be used for portfolios which exhibit the same marginal tail behavior to reduce the run-time of the ARA. Moreover the N *used* for computing the upper bound is smaller than the N *used* for computing the lower bound on $\overline{\text{VaR}}_{0.99}$ in 18 out of the 36 examples and it is the same in the remaining 18 examples, indicating that the N *used* for computing the upper bound is at least as large as that of the lower bound.

- The last two columns in table 6.5 show the mean run-time and the 95% confidence interval of the ARA for the Student's t-distribution. While as before doubling the

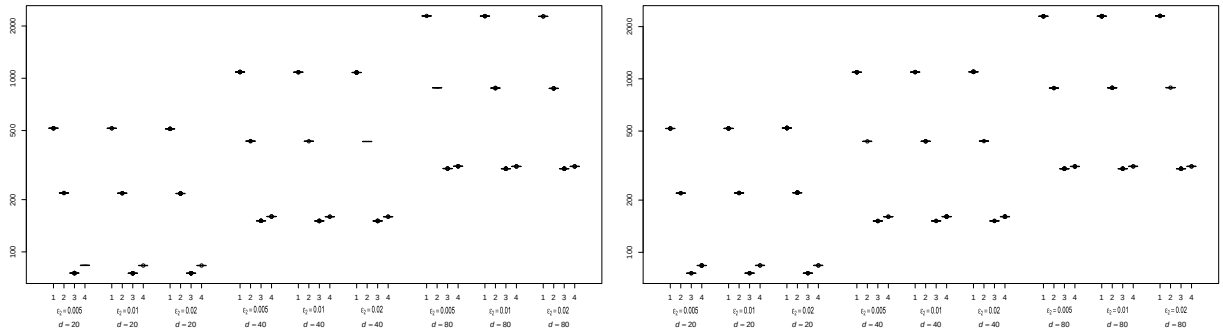


FIGURE 6.7: *Boxplots of lower (left-hand side) and upper (right-hand side) Student's t-distribution* $\overline{\mathrm{VaR}}_{0.99}$ *bounds computed with the ARA based on* $B = 200$ *replications.*

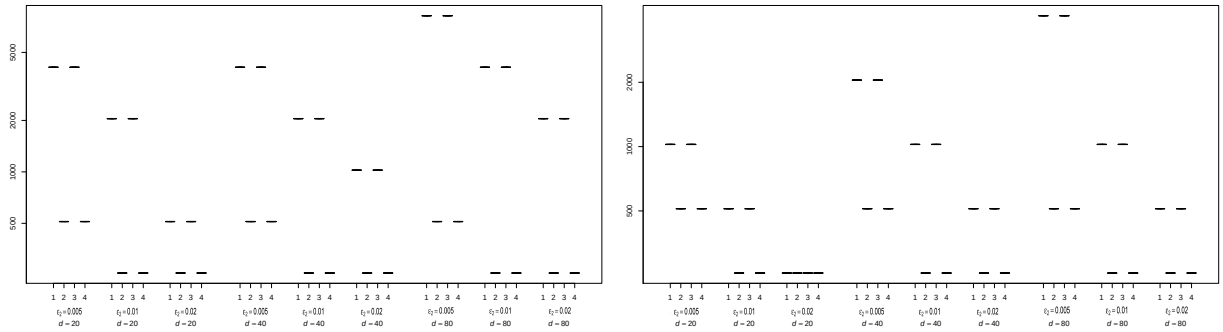


FIGURE 6.8: *Boxplots of the actual* N *used for computing lower (left-hand side) and upper (right-hand side) Student's t-distribution* $\overline{\mathrm{VaR}}_{0.99}$ *bounds with the ARA based on* $B = 200$ *replications.*

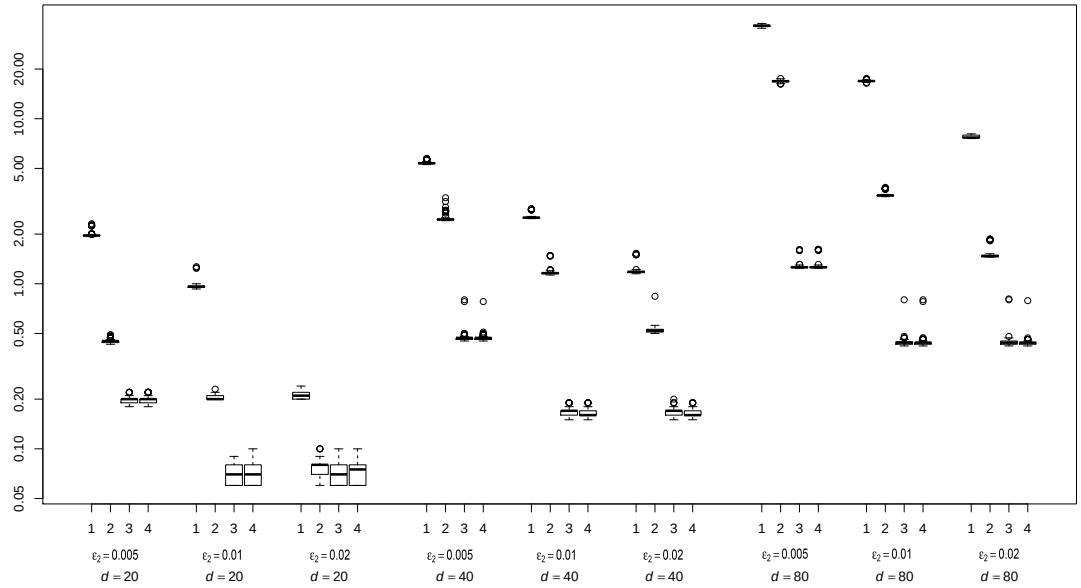


FIGURE 6.9: *Boxplots of the run time in seconds for computing Student's t-distribution* $\overline{\mathrm{VaR}}_{0.99}$ *bounds with the ARA based on* $B = 200$ *replications.*

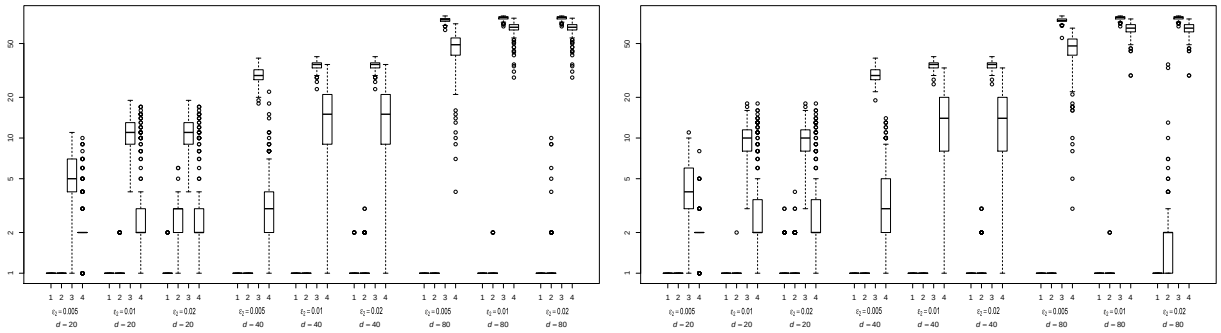


FIGURE 6.10: *Boxplots of the number of oppositely ordered columns for computing lower (left-hand side) and upper (right-hand side) Student's t-distribution* $\overline{\text{VaR}}_{0.99}$ *bounds with the ARA based on* $B = 200$ *replications.*

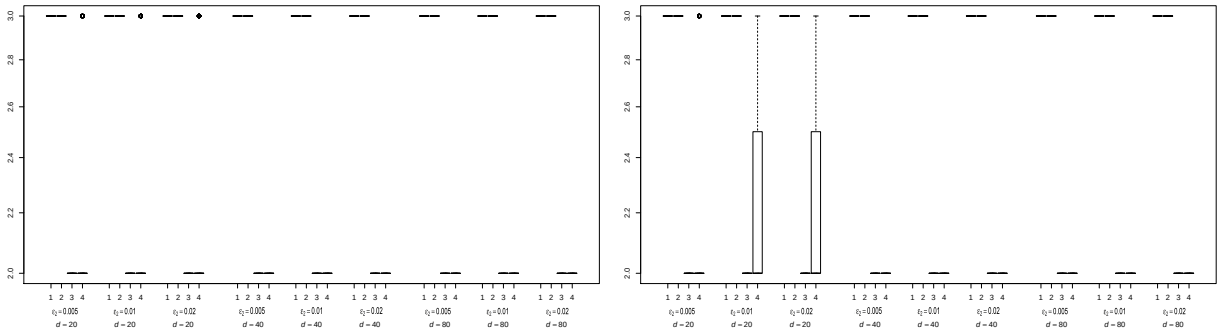


FIGURE 6.11: *Boxplots of the number of iterations for computing lower (left-hand side) and upper (right-hand side) Student's t-distribution* $\overline{\text{VaR}}_{0.99}$ *bounds with the ARA based on* $B = 200$ *replications.*

joint relative error reduces the run-time more than 50% across all examples, we observe a different pattern when we double the number of risk factors; this increases the run-time by at least a factor of 3 (up to 7 in some cases) across examples.

- The mean number of iterations when $\underline{s}_N$ and $\overline{s}_N$ are computed as well as both the lower and upper bound are at most 3. In addition to that the lower and upper 95% confidence interval for the mean number of iterations remains the same as the mean number of iterations in 33 out of 36 examples.

- Figures 6.7, 6.8, 6.9 and 6.11 show minimal impact of the randomization of the initial input matrix X on the $\overline{\text{VaR}}$ bounds, N *used*, run-time and number of iterations for computing these bounds, while the boxplots for the number of oppositely ordered columns for computing the lower and upper Student's t-distribution $\overline{\text{VaR}}$ bounds in figure 6.10 indicate that randomization of the initial input matrix X impacts this parameter more than the others.

6.3.4 Log-normal Distribution Analysis

The last distribution that we use for analyzing the performance of the ARA is the log-normal distribution. This distribution is often used to model losses in the areas of risk

management (such as in credit and operational risk management) in which the observed losses are highly skewed. Its probability density function is given by:

$$f(x) = \frac{1}{\sigma x\sqrt{2\pi}} e^{-(\ln x-\mu)^2/(2\sigma^2)} \quad \mu \in \mathbb{R}, \quad \sigma > 0 \tag{6.3.6}$$

It is a widely used distribution for modelling severity and tail behavior. According to Embrechts and Hofert [92] the most common choices for modelling the severity distribution are log-normal (33%) and Weibull (17%), while for the tail, generalized Pareto and log-normal at (31%) and (14%) constitute the largest two classes of distributions among the models.

Using $\mu = 0$ in (6.3.6) and across all scenarios, we have considered the following ranges for the tail parameter $\sigma_j > 0$ in our studies:

Portfolio 1: $\sigma_1, \ldots, \sigma_d$ are uniformly chosen from 12 to 16. This case represents a portfolio with marginals with similar tail behavior (i.e., a heavy-tailed distribution).

Portfolio 2: $\sigma_1, \ldots, \sigma_d$ are uniformly chosen from 16 to 1. This case represents a portfolio with marginals with differing tail behavior (i.e., from a very heavy-tailed distribution to a not so heavy-tailed distribution).

Portfolio 3: $\sigma_1, \ldots, \sigma_d$ are uniformly chosen from 1 to 2. This case represents a portfolio with marginals with similar tail behavior (i.e., a not so heavy-tailed distribution).

Portfolio 4: $\sigma_2, \ldots, \sigma_d$ are chosen as in Portfolio 3 and $\sigma_1 = 16$. This case represents a portfolio with only one heavy-tailed marginal loss distribution.

The portfolios used in this section, as well as the bivariate vectors of relative errors are the same as for the Student's t-distribution. We have considered three main studies with $d = 20$, $d = 40$ and $d = 80$, each of which are coupled with one of the bivariate vectors of relative errors $\varepsilon = (\varepsilon_{\mathbf{1}}, \varepsilon_{\mathbf{2}})$ chosen as: $\varepsilon = (\mathbf{0.1}\%, \mathbf{0.5}\%)$, $\varepsilon = (\mathbf{0.1}\%, \mathbf{1}\%)$ and $\varepsilon = (\mathbf{0.1}\%, \mathbf{2}\%)$ respectively.

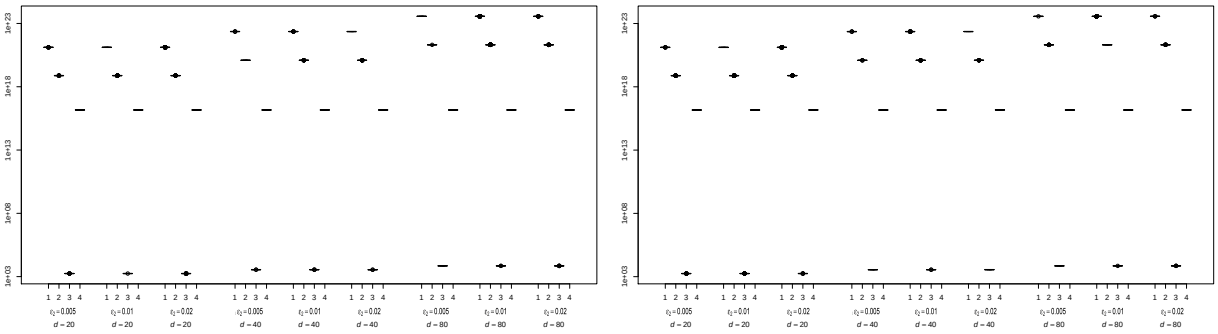


FIGURE 6.12: *Boxplots of lower (left-hand side) and upper (right-hand side) log-normal distribution* $\overline{\mathrm{VaR}}_{0.99}$ *bounds computed with the ARA based on* $B = 200$ *replications.*

			$\overline{\text{VaR}}_{0.99}$			
			Lower		Upper	
d	ε (in %)	Case	Mean	95% CI	Mean	95% CI
20	(0.1, 0.5)	1	1.30502e21	(1.30500e21, 1.30504e21)	1.30848e21	(1.30846e21, 1.30850e21)
		2	7.59062e18	(7.58973e18, 7.59075e18)	7.61253e18	(7.61147e18, 7.61273e18)
		3	1.75236e3	(1.75236e3, 1.75237e3)	1.75889e3	(1.75888e3, 1.75889e3)
		4	1.46258e16	(1.46258e16, 1.46258e16)	1.46687e16	(1.46687e16, 1.46687e16)
	(0.1, 1)	1	1.30330e21	(1.30325e21, 1.30335e21)	1.31020e21	(1.31016e21, 1.31026e21)
		2	7.57893e18	(7.57785e18, 7.57923e18)	7.62211e18	(7.62109e18, 7.62243e18)
		3	1.74915e3	(1.74914e3, 1.74916e3)	1.76144e3	(1.76143e3, 1.76144e3)
		4	1.46258e16	(1.46258e16, 1.46258e16)	1.47118e16	(1.47118e16, 1.47118e16)
	(0.1, 2)	1	1.29988e21	(1.29980e21, 1.29990e21)	1.31369e21	(1.31360e21, 1.31372e21)
		2	7.55731e18	(7.55615e18, 7.55781e18)	7.63812e18	(7.63703e18, 7.63863e18)
		3	1.74317e3	(1.74315e3, 1.74318e3)	1.76607e3	(1.76605e3, 1.76608e3)
		4	1.46258e16	(1.46258e16, 1.46258e16)	1.47984e16	(1.47984e16, 1.47984e16)
40	(0.1, 0.5)	1	2.29986e22	(2.29985e22, 2.29987e22)	2.30568e22	(2.30566e22, 2.30569e22)
		2	1.24262e20	(1.24262e20, 1.24263e20)	1.24589e20	(1.24588e20, 1.24589e20)
		3	3.52970e3	(3.52969e3, 3.52971e3)	3.54085e3	(3.54084e3, 3.54086e3)
		4	1.46258e16	(1.46258e16, 1.46258e16)	1.46687e16	(1.46687e16, 1.46687e16)
	(0.1, 1)	1	2.29676e22	(2.29672e22, 2.29694e22)	2.30869e22	(2.30850e22, 2.30873e22)
		2	1.23792e20	(1.23789e20, 1.23794e20)	1.25006e20	(1.25002e20, 1.25008e20)
		3	3.52425e3	(3.52424e3, 3.52426e3)	3.54496e3	(3.54495e3, 3.54498e3)
		4	1.46258e16	(1.46258e16, 1.46258e16)	1.47118e16	(1.47118e16, 1.47118e16)
	(0.1, 2)	1	2.28724e22	(2.27945e22, 2.29114e22)	2.31818e22	(2.31426e22, 2.32604e22)
		2	1.23180e20	(1.23170e20, 1.23185e20)	1.25522e20	(1.25512e20, 1.25529e20)
		3	3.49611e3	(3.49609e3, 3.49612e3)	3.56396e3	(3.56393e3, 3.56399e3)
		4	1.46258e16	(1.46258e16, 1.46258e16)	1.47984e16	(1.47984e16, 1.47984e16)
80	(0.1, 0.5)	1	3.67899e23	(3.67896e23, 3.67903e23)	3.69689e23	(3.69685e23, 3.69693e23)
		2	2.05190e21	(2.05187e21, 2.05193e21)	2.06138e21	(2.06135e21, 2.06140e21)
		3	7.06116e3	(7.06116e3, 7.06116e3)	7.09510e3	(7.09509e3, 7.09510e3)
		4	1.46258e16	(1.46258e16, 1.46258e16)	1.46687e16	(1.46687e16, 1.46687e16)
	(0.1, 1)	1	3.66995e23	(3.66984e23, 3.67013e23)	3.70576e23	(3.70564e23, 3.70594e23)
		2	2.04719e21	(2.04713e21, 2.04728e21)	2.06542e21	(2.06535e21, 2.06550e21)
		3	7.04484e3	(7.04483e3, 7.04484e3)	7.10549e3	(7.10548e3, 7.10549e3)
		4	1.46258e16	(1.46258e16, 1.46258e16)	1.47118e16	(1.47118e16, 1.47118e16)
	(0.1, 2)	1	3.65222e23	(3.65204e23, 3.65233e23)	3.72380e23	(3.72368e23, 3.72392e23)
		2	2.03818e21	(2.03806e21, 2.03824e21)	2.07322e21	(2.07310e21, 2.07327e21)
		3	7.01596e3	(7.01595e3, 7.01596e3)	7.12125e3	(7.12124e3, 7.12126e3)
		4	1.46258e16	(1.46258e16, 1.46258e16)	1.47984e16	(1.47984e16, 1.47984e16)

TABLE 6.7: *Log-normal distribution* $\overline{\text{VaR}}_{0.99}$ *bounds with the ARA based on* $B = 200$ *bootstrap replications.*

- The $\overline{\text{VaR}}_{0.99}$ mean, lower and upper confidence bounds, shown in table 6.7 depict similar results to those for the Student't t-distribution and Pareto within each study (corresponding to $d = 20$, $d = 40$ and $d = 80$). Increasing the joint relative error ε_2 does not cause a substantial change in the length of the confidence intervals compared to the mean that is reported.

- Table 6.8 shows the mean run-time and the 95% confidence interval of the ARA for the log-normal distribution. A 50% reduction in the run-time across all examples is seen as before when we double the joint relative error. Similarly doubling the number of risk factors increases the run-time by at least a factor of 4 (and up to 7

			N used					
			Lower		Upper		Time (in s)	
d	ε (in %)	Case	Mean	95% CI	Mean	95% CI	Mean	95% CI
20	(0.1, 0.5)	1	32768	(32768, 32768)	16384	(16384, 16384)	12.60	(11.53, 13.80)
		2	2048	(2048, 2048)	2048	(2048, 2048)	7.89	(7.49, 8.20)
		3	32768	(32768, 32768)	16384	(16384, 16384)	0.48	(0.46, 0.59)
		4	2048	(2048, 2048)	2048	(2048, 2048)	0.27	(0.25, 0.36)
	(0.1, 1)	1	16384	(16384, 16384)	8192	(8192, 8192)	6.31	(5.86, 6.88)
		2	1024	(1024, 1024)	1024	(1024, 1024)	3.42	(3.21, 3.58)
		3	16384	(16384, 16384)	8192	(8192, 8192)	0.24	(0.22, 0.27)
		4	1024	(1024, 1024)	1024	(1024, 1024)	0.13	(0.11, 0.15)
	(0.1, 2)	1	8192	(8192, 8192)	4096	(4096, 4096)	2.93	(2.70, 3.11)
		2	512	(512, 512)	512	(512, 512)	1.49	(1.38, 1.61)
		3	8192	(8192, 8192)	4096	(4096, 4096)	0.11	(0.09, 0.13)
		4	512	(512, 512)	512	(512, 512)	0.06	(0.05, 0.07)
40	(0.1, 0.5)	1	65536	(65536, 65536)	32768	(32768, 32768)	73.43	(71.53, 75.76)
		2	4096	(4096, 4096)	2048	(2048, 2048)	42.26	(41.09, 44.07)
		3	65536	(65536, 65536)	32768	(32768, 32768)	2.54	(2.42, 2.65)
		4	4096	(4096, 4096)	2048	(2048, 2048)	0.79	(0.77, 0.91)
	(0.1, 1)	1	32276.48	(16384, 32768)	8192	(8192, 8192)	36.16	(34.30, 38.95)
		2	2048	(2048, 2048)	1024	(1024, 1024)	8.97	(8.75, 9.37)
		3	32440.32	(32768, 32768)	8192	(8192, 8192)	1.15	(1.10, 1.27)
		4	2048	(2048, 2048)	1024	(1024, 1024)	0.37	(0.35, 0.40)
	(0.1, 2)	1	13107.2	(8192, 16384)	4096	(4096, 4096)	15.15	(8.62, 19.57)
		2	512	(512, 512)	512	(512, 512)	4.00	(3.88, 4.11)
		3	14254.08	(8192, 16384)	4096	(4096, 4096)	0.26	(0.23, 0.30)
		4	512	(512, 512)	512	(512, 512)	0.17	(0.15, 0.19)
80	(0.1, 0.5)	1	65536	(65536, 65536)	32768	(32768, 32768)	317.61	(304.92, 335.92)
		2	4096	(4096, 4096)	2048	(2048, 2048)	147.05	(142.70, 153.13)
		3	65536	(65536, 65536)	32768	(32768, 32768)	9.61	(9.44, 9.79)
		4	4096	(4096, 4096)	2048	(2048, 2048)	2.97	(2.87, 3.09)
	(0.1, 1)	1	32768	(32768, 32768)	16384	(16384, 16384)	156.56	(150.52, 171.23)
		2	2048	(2048, 2048)	1024	(1024, 1024)	67.01	(63.25, 72.43)
		3	32768	(32768, 32768)	16384	(16384, 16384)	4.25	(4.15, 4.40)
		4	2048	(2048, 2048)	1024	(1024, 1024)	1.30	(1.27, 1.46)
	(0.1, 2)	1	16384	(16384, 16384)	8192	(8192, 8192)	73.77	(71.99, 75.32)
		2	1024	(1024, 1024)	512	(512, 512)	33.62	(31.84, 34.46)
		3	16384	(16384, 16384)	8192	(8192, 8192)	1.90	(1.86, 2.07)
		4	1024	(1024, 1024)	512	(512, 512)	0.56	(0.53, 0.72)

TABLE 6.8: *N-used and run-time in computing $\overline{\mathrm{VaR}}_{0.99}$ with the ARA based on $B = 200$ bootstrap replications for the log-normal distribution.*

in some cases) across all of the examples considered.

- As can be seen in table 6.8, *N used* in the final iteration of the ARA, as well as the upper and lower bounds for the 95% confidence interval of *N used* remain the same, a result comparable to what we have observed for the Student't t-distribution and Pareto marginals. Similarly, the *N used* for computing the upper bound is smaller than the *N used* for computing the lower bound on $\overline{\mathrm{VaR}}_{0.99}$ in 28 out of the 36 examples and it is the same in the rest of the 8 examples.

- Table 6.9 shows that the mean number of iterations for both the lower and upper bounds exceeds 4 in only 2 examples out of 36; note that in these two examples the

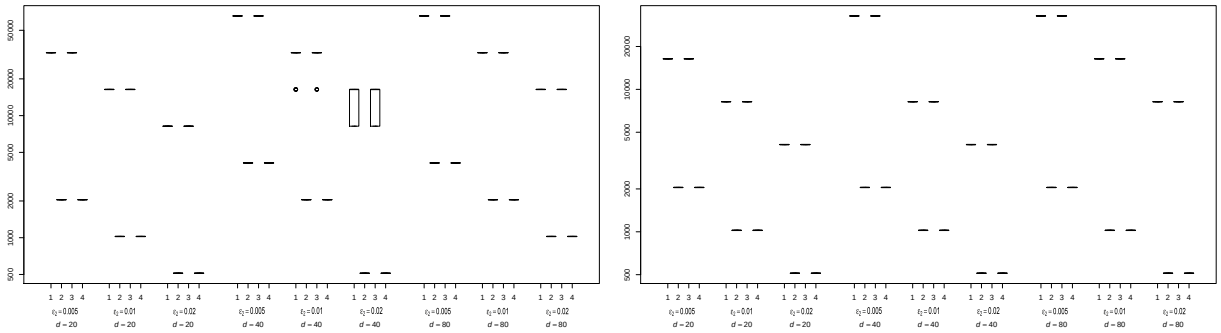


FIGURE 6.13: *Boxplots of the actual N used for computing lower (left-hand side) and upper (right-hand side) log-normal distribution $\overline{\mathrm{VaR}}_{0.99}$ bounds with the ARA based on $B = 200$ replications.*

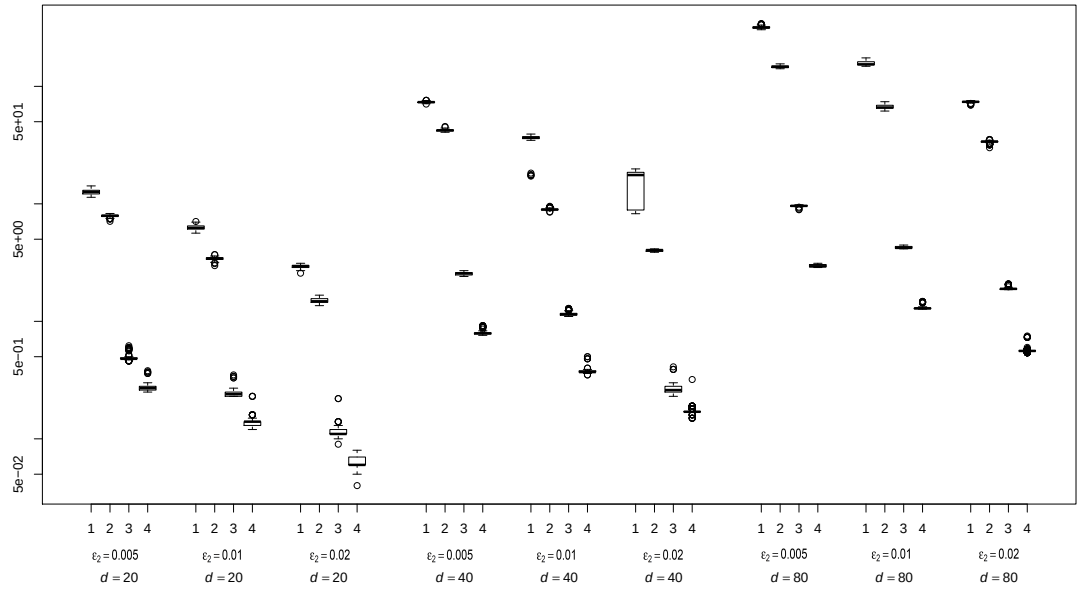


FIGURE 6.14: *Boxplots of the run-time in seconds for computing log-normal distribution $\overline{\mathrm{VaR}}_{0.99}$ bounds with the ARA based on $B = 200$ replications.*

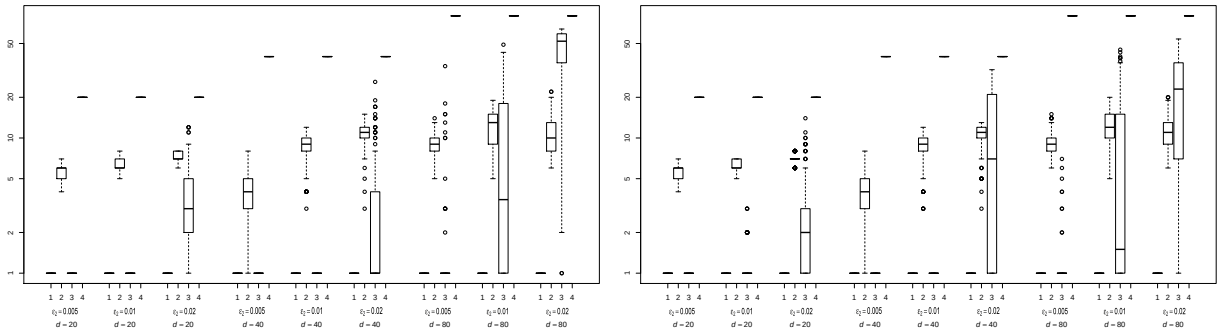


FIGURE 6.15: *Boxplots of the number of oppositely ordered columns for computing lower (left-hand side) and upper (right-hand side) log-normal distribution $\overline{\mathrm{VaR}}_{0.99}$ bounds with the ARA based on $B = 200$ replications.*

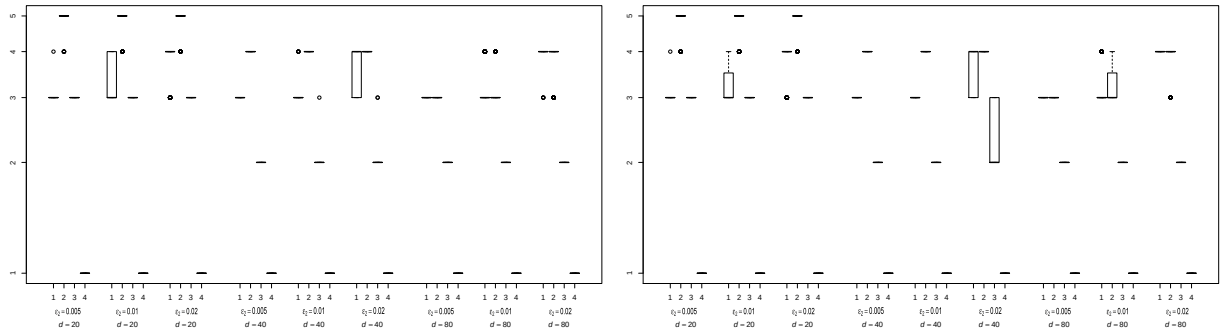


FIGURE 6.16: *Boxplots of the number of iterations for computing lower (left-hand side) and upper (right-hand side) log-normal distribution* $\overline{\text{VaR}}_{0.99}$ *bounds with the ARA based on* $B = 200$ *replications.*

mean is much closer to the upper 95% confidence bound.

- The impact of the randomization of the initial input matrix X for the log-normally distributed losses is similar to what we observed for the Student's t-distribution: while figures 6.12, 6.13, 6.14 and 6.16 show minimal impact of the randomization of the initial input matrix X on the $\overline{\text{VaR}}$ bounds, N *used*, run-time and number of iterations, the boxplots in figure 6.15 show that randomization of the initial input matrix X impacts the number of oppositely ordered columns (in computing both the lower and upper $\overline{\text{VaR}}$ bounds in the final iteration) more than the others.

6.4 Enhanced Adaptive Rearrangement Algorithm

As we have shown in the studies in sections 6.3.2, 6.3.3 and 6.3.4, the mean number of iterations (at most 4 for Pareto marginals and 3 and 5 for Student's t-distribution and log-normal respectively) when computing $\underline{s}_N$ and $\overline{s}_N$ remains relatively small. Therefore, as can be seen in tables 6.2, 6.5 and 6.8, the *N-used* for discretizing the marginal distributions in many studies is less than 10,000 (in 70 out of 96 examples for computing $\underline{s}_N$ and in 84 out of 96 examples for computing $\overline{s}_N$), resulting in much less time being spent on sorting in each iteration of algorithm 6.3. Besides, having a control over the convergence tolerances is one of the main benefits of using the ARA over the RA and as we have shown in the studies conducted in sections 6.3.2, 6.3.3 and 6.3.4, the *N-used* in the ARA on average is much smaller compared to what has been proposed in Embrechts et al. [5] for the RA.

We now consider the impact of doubling the number of risk factors, d, on the mean run-time. The studies conducted using the Student's t-distribution and log-normal marginals (see the last two columns of tables 6.5 and 6.8) show that doubling the number of risk factors results in the run-time going up by a factor of at least 3 (and even up to 7 in some of the examples). Since the ARA aims at solving an NP-complete problem heuristically, having a faster algorithm which achieves the same level of accuracy can be beneficial when running simulations.

The *Enhanced Adaptive Rearrangement Algorithm (EARA)* aims at addressing this problem. This is done by iterating through the given input matrix $X_{N\times d}$ at least once

			Number of oppositely ordered columns				Number of iterations			
			Lower		Upper		Lower		Upper	
d	ε (in %)	Case	Mean	95% CI	Mean	95% CI	Mean	95% CI	Mean	95% CI
20	(0.1, 0.5)	1	1	(1, 1)	1	(1, 1)	3.005	(3, 3)	3.005	(3, 3)
		2	5.71	(5, 7)	5.585	(5, 6.025)	4.97	(4, 5)	4.915	(4, 5)
		3	1	(1, 1)	1	(1, 1)	3	(3, 3)	3	(3, 3)
		4	20	(20, 20)	20	(20, 20)	1	(1, 1)	1	(1, 1)
	(0.1, 1)	1	1	(1, 1)	1	(1, 1)	3.3	(3, 4)	3.25	(3, 4)
		2	6.45	(5, 7)	6.32	(5, 7)	4.905	(4, 5)	4.87	(4, 5)
		3	1	(1, 1)	1.095	(1, 2)	3	(3, 3)	3	(3, 3)
		4	20	(20, 20)	20	(20, 20)	1	(1, 1)	1	(1, 1)
	(0.1, 2)	1	1	(1, 1)	1	(1, 1)	3.895	(3, 4)	3.88	(3, 4)
		2	7.205	(6, 8)	7.095	(6, 8)	4.875	(4, 5)	4.855	(4, 5)
		3	3.845	(1, 9.05)	2.84	(1, 9)	3	(3, 3)	3	(3, 3)
		4	20	(20, 20)	20	(20, 20)	1	(1, 1)	1	(1, 1)
40	(0.1, 0.5)	1	1	(1, 1)	1	(1, 1)	3	(3, 3)	3	(3, 3)
		2	4.15	(2, 8)	4.225	(2, 7.02)	4	(4, 4)	4	(4, 4)
		3	1	(1, 1)	1	(1, 1)	2	(2, 2)	2	(2, 2)
		4	40	(40, 40)	40	(40, 40)	1	(1, 1)	1	(1, 1)
	(0.1, 1)	1	1	(1, 1)	1	(1, 1)	3.025	(3, 3.02)	3	(3, 3)
		2	8.38	(4, 11)	8.37	(4, 11)	4	(4, 4)	4	(4, 4)
		3	1	(1, 1)	1	(1, 1)	2.005	(2, 2)	2	(2, 2)
		4	40	(40, 40)	40	(40, 40)	1	(1, 1)	1	(1, 1)
	(0.1, 2)	1	1	(1, 1)	1	(1, 1)	3.575	(3, 4)	3.57	(3, 4)
		2	10.81	(7.97, 13)	10.435	(5, 13)	4	(4, 4)	4	(4, 4)
		3	3.39	(1, 15.05)	11.11	(1, 31)	2.01	(2, 2)	2.38	(2, 3)
		4	40	(40, 40)	40	(40, 40)	1	(1, 1)	1	(1, 1)
80	(0.1, 0.5)	1	1	(1, 1)	1	(1, 1)	3	(3, 3)	3	(3, 3)
		2	8.965	(6, 12)	9.2	(6, 13.025)	3	(3, 3)	3	(3, 3)
		3	1.655	(1, 11.05)	1.135	(1, 3.025)	2	(2, 2)	2	(2, 2)
		4	80	(80, 80)	80	(80, 80)	1	(1, 1)	1	(1, 1)
	(0.1, 1)	1	1	(1, 1)	1	(1, 1)	3.175	(3, 4)	3.21	(3, 4)
		2	12.27	(6, 18)	12.055	(6, 18)	3.21	(3, 4)	3.25	(3, 4)
		3	11.09	(1, 41.025)	8.365	(1, 37.05)	2	(2, 2)	2	(2, 2)
		4	80	(80, 80)	80	(80, 80)	1	(1, 1)	1	(1, 1)
	(0.1, 2)	1	1	(1, 1)	1	(1, 1)	3.98	(4, 4)	4	(4, 4)
		2	11.04	(6, 19)	11.19	(7, 19)	3.84	(3, 4)	3.95	(3, 4)
		3	45.61	(3, 63)	22.7	(1, 49)	2	(2, 2)	2	(2, 2)
		4	80	(80, 80)	80	(80, 80)	1	(1, 1)	1	(1, 1)

TABLE 6.9: *Number of oppositely ordered columns and number of iterations in computing log-normal distribution* $\overline{\text{VaR}}_{0.99}$ *with the ARA based on* $B = 200$ *replications for the log-normal distribution.*

and then reducing the number of the required iterations afterward over the entire input matrix for meeting the convergence criteria that is given by $\varepsilon = (\varepsilon_1, \varepsilon_2)$. More specifically, after the first iteration over an input matrix $X_{N\times d}$ (and oppositely ordering at least $d+1$ columns of the input matrices $\underline{X}^{\alpha}$ and $\overline{X}^{\alpha}$ to the sum of all other remaining columns respectively), *EARA* proceeds only if the convergence criteria given by $\varepsilon = (\varepsilon_1, \varepsilon_2)$ is not met. In that case and from the second iteration onward, after oppositely ordering each column, EARA checks the convergence criteria given by $\varepsilon = (\varepsilon_1, \varepsilon_2)$ and stops if they are both satisfied. As we will show in sections 6.4.1 and 6.5, given a large number of risk factors, d, stopping in such a scenario while the algorithm has gone through several

iterations (in which we have used a much larger N in the successive iteration, resulting in less time being spent on sorting and not oppositely ordering of *all* of the columns of the input matrices when using the new stopping criteria) can significantly reduce the overall run-time.

As before let $\mathrm{err}_i(X,Y)$ and $\mathrm{err}_j(X,Y)$ be the individual and joint relative functions defined in 6.3.2 and 6.3.3 respectively.

Algorithm 6.4: Enhanced Adaptive Rearrangement Algorithm for Computing $\overline{\mathrm{VaR}}_\alpha(L^+)$

---

1. Fix a confidence level $\alpha \in (0,1)$, marginal quantile functions $F_1^-, \ldots, F_d^-$, an integer vector $\mathbf{K} \in \mathbb{N}^{\mathbf{l}}$, $l \in \mathbb{N}$, (containing the numbers of discretization points which are adaptively used), a bivariate vector of relative convergence tolerances $\varepsilon = (\varepsilon_{\mathbf{1}}, \varepsilon_{\mathbf{2}})$ (containing the individual relative tolerance $\varepsilon_1 > 0$ and the joint relative tolerance $\varepsilon_2 > 0$) and the maximal number of iterations used for each $k \in \mathbf{K}$.

2. For $N = 2^k$, $k \in \mathbf{K}$, do:

   2.1. Compute the lower bound for $\overline{\mathrm{VaR}}_\alpha(L^+)$:

      2.1.1. Define the matrix $\underline{X}^\alpha = (\underline{x}_{ij}^\alpha)$ for $\underline{x}_{ij}^\alpha = F_j^-(\alpha + \frac{(1-\alpha)(i-1)}{N})$, $i \in \{1, \ldots, N\}$, $j \in \{1, \ldots, d\}$.

      2.1.2. Permute randomly the elements in each column of $\underline{X}^\alpha$.

      2.1.3. Set $\underline{Y}^\alpha = \underline{X}^\alpha$. For $j \in \{1, 2, \ldots, d, 1, 2, \ldots, d, \ldots\}$, rearrange the jth column of the matrix $\underline{Y}^\alpha$ so that it becomes oppositely ordered to the sum of all other columns. After having rearranged at least $d+1$ columns, set, after every column rearrangement, the matrix $\underline{X}^\alpha$ to the matrix $\underline{Y}^\alpha$ from d rearrangement steps earlier and stop if the maximal number of column rearrangements is reached or if

$$\mathrm{err}_i(\underline{X}^\alpha, \underline{Y}^\alpha) \leqslant \varepsilon_1 \tag{6.4.1}$$

      2.1.4. Set $\underline{s}_N = s(\underline{Y}^\alpha)$.

   2.2. Compute the upper bound for $\overline{\mathrm{VaR}}_\alpha(L^+)$:

      2.2.1. Define the matrix $\overline{X}^\alpha = (\overline{x}_{ij}^\alpha)$ for $\overline{x}_{ij}^\alpha = F_j^-(\alpha + \frac{(1-\alpha)i}{N})$, $i \in \{1, \ldots, N\}$, $j \in \{1, \ldots, d\}$. If (for $i = N$ and) for any $j \in \{1, \ldots, d\}$, $F_j^-(1) = \infty$, adjust it to $F_j^-(\alpha + \frac{(1-\alpha)(N-1/2)}{N})$.

      2.2.2. Permute randomly the elements in each column of $\overline{X}^\alpha$.

      2.2.3. Set $\overline{Y}^\alpha = \overline{X}^\alpha$. For $j \in \{1, 2, \ldots, d, 1, 2, \ldots, d, \ldots\}$, rearrange the jth column of the matrix $\overline{Y}^\alpha$ so that it becomes oppositely ordered to the sum of all other columns. After having rearranged at least $d+1$ columns, set, after every column rearrangement, the matrix $\overline{X}^\alpha$ to the matrix $\overline{Y}^\alpha$ from d rearrangement steps earlier and stop if the maximal number of column rearrangements is reached or if

$$\mathrm{err}_i(\overline{X}^\alpha, \overline{Y}^\alpha) \leqslant \varepsilon_1 \tag{6.4.2}$$

2.2.4. Set $\overline{s}_N = s(\overline{Y}^{\alpha})$.

2.3. Determine convergence based on both the individual and the joint relative convergence tolerances:

If (6.4.1) and (6.4.2) hold, and if $\text{err}_j(\overline{s}_N, \underline{s}_N) \leqslant \varepsilon_2$ then break.

3. Return $(\underline{s}_N,\ \overline{s}_N)$.

The new stopping criterion, described above, is implemented in (6.4.1) and (6.4.2) of the algorithm 6.4 in which we iteratively evaluate the stipulated conditions *only* based on oppositely ordering *column-by-column* (after at least oppositely ordering $d+1$ columns at the beginning). The matrices $\underline{X}^{\alpha}$ and $\underline{Y}^{\alpha}$ (and $\overline{X}^{\alpha}$ and $\overline{Y}^{\alpha}$) store the results from the previous d and $d+1$ column rearrangements respectively when evaluating the condition (6.4.1) (and (6.4.2)), given that the maximum number of the column rearrangements is not reached.

In the following section we compare the performance of the ARA with its enhanced counterpart in the case of Pareto marginals. We refer to algorithm 6.4 as the EARA.

6.4.1 Enhanced Adaptive Rearrangement Algorithm: Pareto Marginals

In order to analyze the performance of the EARA, we use the main studies, corresponding to $d = 20$ and $d = 100$ that we considered in section 6.3.2. Recall that under each study, we investigate 12 examples, each of which consisted of one the following 4 cases

$$F_j(x) = 1 - (1+x)^{-\theta_j}, \quad x \geqslant 0,$$

for a given tail parameter $\theta_j > 0$.

Portfolio 1: $\theta_1, \ldots, \theta_d$ are uniformly chosen from 0.6 to 0.4; this case represents a portfolio with marginals with similar tail behavior (i.e., a heavy-tailed distribution).

Portfolio 2: $\theta_1, \ldots, \theta_d$ are uniformly chosen from 0.5 to 1.5; this case represents a portfolio with marginals with differing tail behavior (i.e., from a very heavy-tailed distribution to a not so heavy-tailed distribution).

Portfolio 3: $\theta_1, \ldots, \theta_d$ are uniformly chosen from 1.4 to 1.6; this case represents a portfolio with marginals with similar tail behavior (i.e., a not so heavy-tailed distribution).

Portfolio 4: $\theta_2, \ldots, \theta_d$ are chosen as in Portfolio 3 and $\theta_1 = 0.5$; this case represents a portfolio with only one heavy-tailed marginal loss distribution.

Using the above portfolios, coupled with one of the three choices of the bivariate vector of relative errors $\varepsilon = (\varepsilon_{\mathbf{1}}, \varepsilon_{\mathbf{2}})$ given by $\varepsilon = (0.1\%, 0.5\%)$, $\varepsilon = (0.1\%, 1\%)$ and $\varepsilon = (0.1\%, 2\%)$

we investigate the performance of the EARA to that of the ARA given Pareto marginals. The results of comparing the run-times as well as the percentage change in the $\overline{\text{VaR}}_{0.99}$ calculated using the ARA and the EARA are reported in table 6.10.

		Ratio of EARA mean run-time to ARA mean run-time		Maximum % change
d	ε (in %)	Minimum	Maximum	in mean $\overline{\text{VaR}}_{0.99}$
20	(0.1,0.5)	28.1%	32.7%	0.48%
	(0.1,1)	27.2%	31.2%	0.39%
	(0.1,2)	25.9%	30.5%	0.46%
100	(0.1,0.5)	6.9%	9.1%	0.21%
	(0.1,1)	5.7%	8.5%	0.18%
	(0.1,2)	4.7%	7.4%	0.14%

TABLE 6.10: *Comparison of the mean run-time and % change in computing the mean* $\overline{\text{VaR}}_{0.99}$ *bounds with the EARA and the ARA based on* $B = 200$ *bootstrap replications.*

To simplify the presentation of the results, we are only reporting the minimum and maximum of the ratio of the mean run-times calculated based on 200 bootstrap replications from the EARA to that of the ARA across all of the cases that are described in 6.4.1. These ratios are shown in table 6.10.

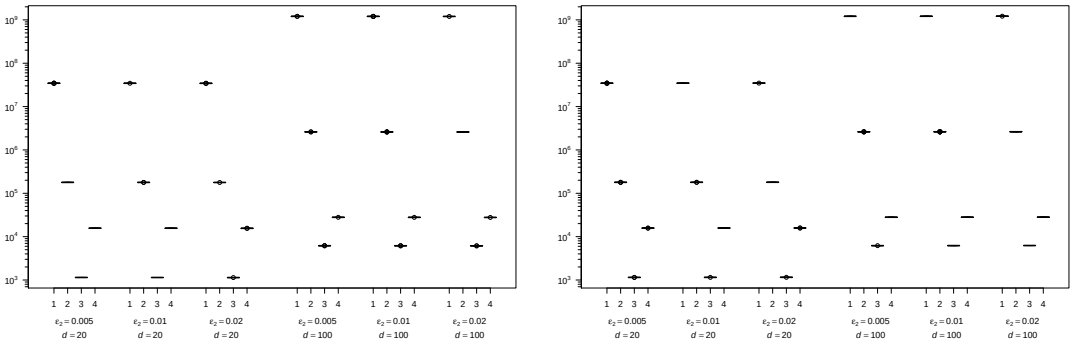


FIGURE 6.17: *Boxplots of the lower (left-hand side) and upper (right-hand side)* $\overline{\text{VaR}}_{0.99}(L^+)$ *bounds computed with the EARA for* $\varepsilon_1 = 0.001$ *using* $B = 200$ *replications.*

The impact that increasing the dimensionality has on the performance and run-time of these algorithms becomes clear from this analysis: while the EARA decreases the run-time in both studies, its performance is significantly better with a larger number of risk factors, namely $d = 100$.

The last column shows the maximum of the relative change in the mean $\overline{\text{VaR}}_{0.99}$ calculated through the ARA and the EARA. Note that since both algorithms satisfy the convergence criteria that are given by the bivariate vector of relative errors, we should not see a drastic change in this column. As can be seen in this column, the mean $\overline{\text{VaR}}_{0.99}$ bounds calculated through these algorithms in each example are very close to each other

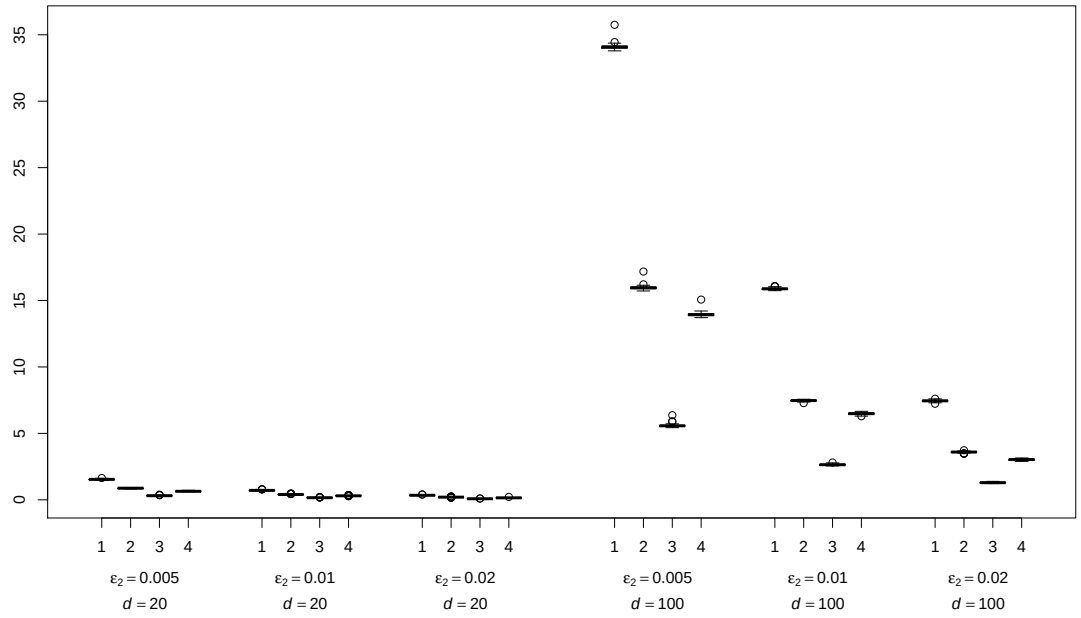


FIGURE 6.18: *Boxplots of the run-time (in seconds) for computing the lower and upper* $\overline{\mathrm{VaR}}_{0.99}(L^+)$ *bounds (on left-hand side and right-hand side respectively) with the EARA for* $\varepsilon_1 = 0.001$ *based on* $B = 200$ *replications.*

and only differ by at most 0.48% in the $d = 20$ study and only by 0.21% in the $d = 100$ study.

6.5 Operational Risk Data, ARA and EARA

So far we have analyzed the performance of the ARA and EARA by using distributions that were chosen a priori. In this section we aim at analyzing the performance of these algorithms on a real-world data set and explore an additional feature of these algorithms that impacts their run-time significantly.

Measuring and managing operational risk are important in practice. According to the operational risk guidelines issued by the Basel Committee on Banking Supervision (BCBS) [93], operational risk is defined as "The risk of a loss resulting from inadequate or failed internal processes, people and systems or from external events. This definition includes legal risk, but excludes strategic and reputational risk.".

In the following example we use the operational risk data set of [82]. Using 950 data points that represent inflation adjusted gross losses, we focus on 10 risk factors (each of which is associated with a business line) and fit a Generalized Pareto distribution to them. Skewness among these losses ranges from 1.75 to 8.60, indicating that these losses are quiet positively skewed. More statistics on these losses (such as mean, standard deviation, median, etc.) can be found in [82], page 14. We refer to this data set as OpRisk data. Figure 6.19 shows the Q-Q plot of the residuals for the business lines Trading and Sales (TS) and Agency Services (AS). The rest of the Q-Q plots are shown in Appendix D.

Using the OpRisk data set we investigate the impact of the order of inputting the

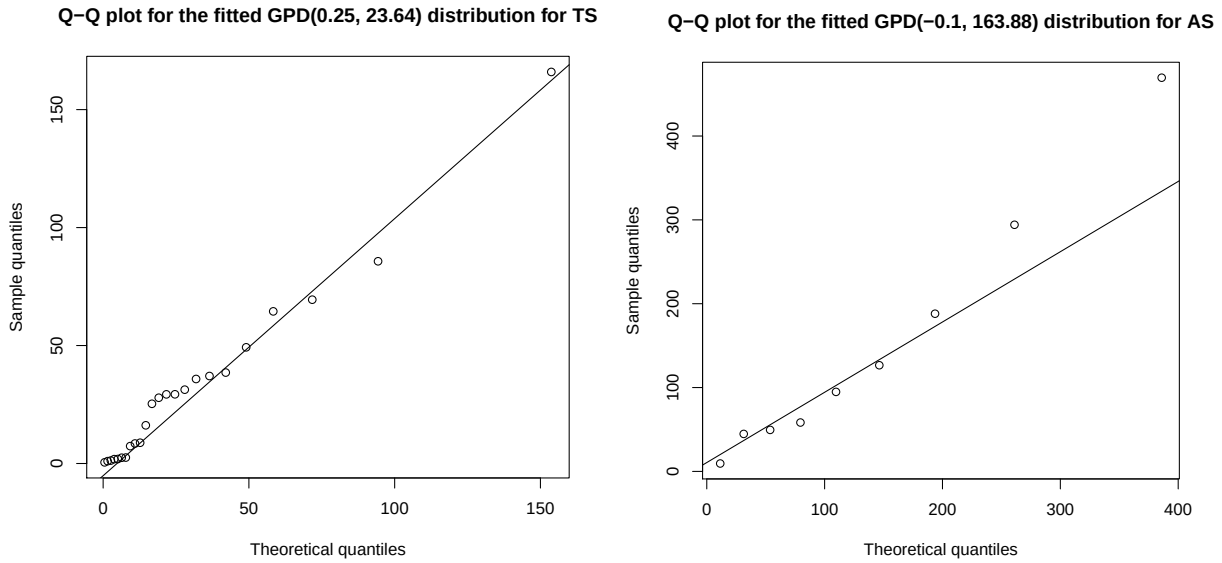


FIGURE 6.19: *Q-Q plots of fitting the Generalized Pareto Distribution to Trading and Sales (TS) and Agency Services (AS) losses.*

respective marginal distributions into the input matrix $X_{N\times d}$ on the run-time of the ARA and the EARA. Given a fixed bivariate vector of relative errors $\varepsilon = (\varepsilon_{\mathbf{1}}, \varepsilon_{\mathbf{2}})$ we compare the run-time of the EARA and the ARA algorithm in one of the following studies:

- ***Heavy first***: we start by inputting the risk factor with the heaviest marginal (found through fitting the Generalized Pareto distribution to the losses) in the first column and subsequently substituting the rest of the discretized marginals in the successive columns based on decreasing order of the heavy-tailedness of the respective remaining marginal. We refer to this case as *Heavy* in our simulation.

- ***Random order***: the order in which each discretized marginal is put in each column is random in this study. Moreover we make sure that the heaviest margin is not in the first column. We refer to this case as *Random* in our simulation.

As can be seen in table 6.11, while the EARA consistently outperforms the ARA across all studies, the ratio of the mean run times for $d = 10$ is on average higher than those for $d = 20$ and $d = 100$, presented earlier in section 6.4.1. This finding is comparable to that of section 6.4.1: when using the EARA the mean run-time decreases as d increases.

The main finding of this section is that the order of inserting marginals into the input matrix $X_{N\times d}$ has a significant impact on the run-time: as can be seen in table 6.11, both the EARA and the ARA in *heavy* studies outperform their randomized counterpart and take much less time to satisfy the given convergence criteria as given by $\varepsilon = (\varepsilon_{\mathbf{1}}, \varepsilon_{\mathbf{2}})$. A similar observation is made for the mean number of iterations when $\underline{s}_N$ and $\overline{s}_N$ are computed in both studies.

		Ratio of mean run-time				Mean number of iterations			
		EARA		ARA		EARA		ARA	
d	ε (in %)	Heavy	Random	Heavy	Random	Heavy	Random	Heavy	Random
10	(0.1, 0.5)	38.7%	47.2%	82.5%	100%	2.25	3.25	4.25	5
	(0.1, 1)	24.8%	41.5%	77.2%	100%	2	3	4	5
	(0.1, 2)	29.9%	33.2%	74.3%	100%	2	3	4	5

TABLE 6.11: *Ratio of the mean run-time of the EARA and ARA to the run-time of the ARA with randomized columns and the mean number of iterations of the EARA and ARA based on 200 replications.*

6.6 Conclusion

Chapters 5 and 6 presented two contributions to the problem of computation of the worst VaR for a sum of losses with given marginals in the context of Quantitative Risk Management.

In chapter 5 we considered the homogeneous case (i.e., when all margins are equal) and addressed the dual bound approach based on [5, Proposition 4] and Wang's approach based on [6, Proposition 1] for computing worst VaR. Although both of these approaches are mathematically "explicit", care has to be exercised when computing worst VaR bounds with these algorithms in practice. Several numerical and computational hurdles in their implementation were identified and addressed.

Several numerical steps such as computing initial intervals for the root-finding procedure involved were covered. A particular example which highlights the *numerical challenges* when computing worst VaR in general is the case of equal Pareto margins (for which we also showed uniqueness of the root even in the infinite-mean case).

Further, we considered the general, i.e., inhomogeneous case. We first investigated the performance of the Rearrangement Algorithm in various studies and focused on answering the questions that the original algorithm leaves open concerning the concrete choice of two of its tuning parameters. These parameters were shown to have a substantial impact on the algorithm's performance and thus need to be chosen with care.

In chapter 6 we therefore presented ARA. The latter improves the original RA in that it addresses the aforementioned two tuning parameters and improves on the underlying algorithmic design. The number of discretization points is chosen automatically in an adaptive way (hence the name of the algorithm). The absolute convergence tolerance is replaced by two relative convergence tolerances. Since they are relative tolerances, their choice is much more intuitive. The first relative tolerance is used to determine the individual convergence of each of the lower bound $\underline{s}_N$ and the upper bound $\overline{s}_N$ for worst VaR and the second relative tolerance is used to control how far apart $\underline{s}_N$ and $\overline{s}_N$ are. The original version of the Rearrangement Algorithm does not allow for such a control.

Based on our findings in investigating the performance of the ARA, we proposed an enhanced version of this algorithm which significantly reduces the run-time compared to the ARA for portfolios with larger numbers of risk factors. Finally, the impact of various orders of inserting discretized marginal distributions into the input matrix was investigated.

There are still several interesting questions left to be investigated. First of all, as for the RA, the theoretical convergence properties of the algorithm remain an open problem. Also, as mentioned above, it remains unclear how the rows and columns of the input matrices for the RA or the ARA can be set up in an initialization such that the run-time is minimal.

Another interesting question is whether or not we can use the rearranged matrices from previous iterations (with $N = 2^{k-1}$ used as the tail discretization parameter) to construct new input matrices (with $N = 2^k$) for which the convergence criteria, given by $\varepsilon = (\varepsilon_{\mathbf{1}}, \varepsilon_{\mathbf{2}})$ is met faster than when we start with a randomized input matrix X with the tail discretization parameter $N = 2^k$?

Bibliography

[1] P. Glasserman and L. Yang. Bounding wrong-way risk in CVA calculation. *Mathematical Finance*, 2016. 2, 6, 8, 12, 14, 17, 19, 22, 30, 44, 47

[2] D. Rosen and D. Saunders. CVA the wrong way. *Journal of Risk Management in Financial Institutions*, 5(3):252–272, 2012. 2, 6, 12, 14, 17, 40

[3] M. Pykhtin and D. Rosen. Pricing counterparty risk at the trade level and CVA allocations. 2010. 2, 6, 8, 31, 32, 34

[4] A. Memartoluie, D. Saunders, and T. Wirjanto. Wrong-way risk bounds in counterparty credit risk management. *Journal of Risk Management in Financial Institutions*, 10(2):150–163, 2017. 3

[5] P. Embrechts, G. Puccetti, and L. Rüschendorf. Model uncertainty and VaR aggregation. *Journal of Banking and Finance*, 37(8):2750–2764, 2013. 3, 66, 67, 70, 71, 72, 74, 84, 86, 87, 117, 124

[6] P. Embrechts, G. Puccetti, L. Rüschendorf, R. Wang, and A. Beleraj. An Academic Response to Basel 3.5. *Risks*, 2(1):25–48, 2014. 3, 67, 70, 73, 74, 75, 124

[7] G. Puccetti and L. Rüschendorf. Computation of sharp bounds on the distribution of a function of dependent risks. *Journal of Computational and Applied Mathematics*, 236(7):1833–1840, 2012. 3, 44, 67, 68, 69, 70, 97

[8] M. Hofert, A. Memartoluie, D. Saunders, and T. Wirjanto. Improved algorithms for computing worst value-at-risk. *Statistics & Risk Modeling*. 4

[9] Basel Committee on Banking Supervision (BCBS). Strengthening the resilience of the banking sector. Technical report, Bank for International Settlements, 2012. http://www.bis.org/publ/bcbs164.htm. 5

[10] E. Derman. Model risk, quantitative strategies research notes. *Goldman Sachs, New York, NY*, 1996. 7

[11] I. A. Cooper and A. S. Mello. The default risk of swaps. *The Journal of Finance*, 46 (2):597–620, 1991. 7

[12] D. Brigo and M. Masetti. Risk neutral pricing of counterparty risk. 2005. 7

[13] D. Brigo and A. Pallavicini. Counterparty risk pricing under correlation between default and interest rates. *Numerical Methods for Finance*, 63, 2007. 7, 8

[14] D. Brigo, K. Chourdakis, and I. Bakkar. Counterparty risk valuation for energy-commodities swaps: Impact of volatilities and correlation. *Available at SSRN 1150818*, 2008. 7, 8

[15] L. Rüschendorf. Comparison of multivariate risks and positive dependence. *Journal of Applied Probability*, 41:391–406, 2004. 7

[16] L. Rüschendorf. Risk bounds, worst case dependence and optimal claims and contracts. In *Proceedings AFMATH Conference Brussel*. Citeseer, 2012. 7

[17] J. Haase, M. Ilg, and R. Werner. Model-free bounds on bilateral counterparty valuation. Technical report, 2010. Available at http://mpra.ub.uni-muenchen.de/24796/. 7

[18] D. Brigo and K. Chourdakis. Counterparty risk for credit default swaps: Impact of spread volatility and default correlation. *International Journal of Theoretical and Applied Finance*, 12(07):1007–1026, 2009. 8

[19] S. Y. Leung and Y. K. Kwok. Credit default swap valuation with counterparty risk. *The Kyoto Economic Review*, 74(1):25–45, 2005. 8

[20] A. Lipton and A. Sepp. Credit value adjustment for credit default swaps via the structural default model. *The Journal of Credit Risk*, 5(2):127–150, 2009. 8

[21] S. Crépey, M. Jeanblanc, and B. Zargari. *Counterparty risk on a CDS in a Markov chain copula model with joint defaults*. 2010. 8

[22] J. C. Hull and A. D. White. Valuing credit default swaps II: Modeling default correlations. *The Journal of Derivatives*, 8(3):12–21, 2001. 8

[23] M. B. Walker. Credit default swaps with counterparty risk: A calibrated Markov model. *Journal of Credit Risk*, 2(1):31–49, 2006. 8

[24] E. Sorensen and T. Bollier. Pricing swap default risk. *Financial Analysts Journal*, pages 23–33, 1994. 8

[25] D. Duffie and M. Huang. Swap rates and credit quality. *The Journal of Finance*, 51 (3):921–949, 1996. 8

[26] D. Brigo and A. Capponi. Bilateral counterparty risk valuation with stochastic dynamical models and application to credit default swaps. *Available at SSRN 1318024*, 2009. 8

[27] D. Brigo, A. Pallavicini, and V. Papatheodorou. Arbitrage-free valuation of bilateral counterparty risk for interest-rate products: impact of volatilities and correlations. *International Journal of Theoretical and Applied Finance*, 14(06):773–802, 2011. 8

[28] J. Gregory. *Counterparty credit risk and credit value adjustment: A continuing challenge for global financial markets*. John Wiley & Sons, 2012. 8

[29] G. Cesari, J. Aquilina, N. Charpillon, Z. Filipovic, G. Lee, and I. Manda. *Modelling, pricing, and hedging counterparty credit exposure: A technical guide*. Springer Science & Business Media, 2009. 8, 11

[30] D. Brigo, M. Morini, and A. Pallavicini. *Counterparty Credit Risk, Collateral and Funding: With Pricing Cases for All Asset Classes*. John Wiley & Sons, 2013. 8

[31] P. Glasserman and X. Xu. Robust risk measurement and model risk. *Quantitative Finance*, 14(1):29–58, 2014. 8

[32] M. Beiglböck, P. Henry-Labordère, and F. Penkner. Model-independent bounds for option prices, a mass transport approach. *Finance and Stochastics*, 17(3):477–501, 2013. 8, 43

[33] A. Pallavicini, D. Perini, and D. Brigo. Funding valuation adjustment: A consistent framework including CVA, DVA, collateral, netting rules and re-hypothecation. *arXiv preprint arXiv:1112.1521*, 2011. 9

[34] M. Morini and A. Prampolini. Risky funding: A unified framework for counterparty and liquidity charges. *Available at SSRN 1669930*, 2010. 9

[35] C. Burgard and M. Kjaer. Generalised CVA with funding and collateral via semi-replication. *Available at SSRN 2027195*, 2012. 9

[36] A. Arvanitis and J. Gregory. *Credit. The complete guide to pricing, hedging and risk management*. Risk books, 2001. 11

[37] J.C. Garcia-Cespedes, J.A. de Juan Herrero, D. Rosen, and D. Saunders. Effective modeling of wrong-way risk, CCR capital and alpha in Basel II. *Journal of Risk Model Validation*, 4(1):71–98, 2010. 11, 56

[38] Basel Committee on Banking Supervision (BCBS). Review of the credit valuation adjustment risk framework. Technical report, Bank for International Settlements, 2015. Available at http://www.bis.org/bcbs/publ/d325.pdf. 13

[39] Basel Committee on Banking Supervision (BCBS). An explanatory note on the Basel II IRB risk weight functions. Technical report, Bank for International Settlements, 2005. Available at http://www.bis.org/bcbs/irbriskweight.pdf. 14, 45

[40] C. Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008. 30

[41] Basel Committee on Banking Supervision (BCBS). Basel III: A global regulatory framework for more resilient banks and banking systems - revised version June 2011. Technical report, Bank for International Settlements, 2011. Available at http://www.bis.org/publ/bcbs189.htm. 42

[42] Basel Committee on Banking Supervision (BCBS). Developments in modelling risk aggregation. Technical report, Bank for International Settlements, 2010. Available at http://www.bis.org/publ/joint25.pdf. 42

[43] D. Bertsimas and I. Popescu. On the relation between option and stock prices: A convex optimization approach. *Operations Research*, 50(2):358–374, 2002. 43

[44] D. Hobson, P. Laurence, and T.H. Wang. Static-arbitrage optimal subreplicating strategies for basket options. *Insurance: Mathematics and Economics*, 37(3):553–572, 2005. 43

[45] D. Hobson, P. Laurence, and T.H. Wang. Static-arbitrage upper bounds for the prices of basket options. *Quantitative Finance*, 5(4):329–342, 2005. 43

[46] P. Laurence and T.H. Wang. What is a basket worth? *Risk Magazine*, February 2004. 43

[47] P. Laurence and T.H. Wang. Sharp upper and lower bounds for basket options. *Applied Mathematical Finance*, 12(3):253–282, 2005. 43

[48] X. Chen, G. Deelstra, J. Dhaene, and M. Vanmaele. Static super-replicating strategies for a class of exotic options. *Insurance: Mathematics and Economics*, 42 (3):1067–1085, 2008. 43

[49] G. D. Makarov. Estimates for the distribution function of a sum of two random variables when the marginal distributions are fixed. *Theory of Probability & its Applications*, 26(4):803–806, 1982. 43, 68

[50] R. C. Williamson and T. Downs. Probabilistic arithmetic. I. Numerical methods for calculating convolutions and dependency bounds. *International Journal of Approximate Reasoning*, 4(2):89–158, 1990. 43, 69

[51] M. Denuit, C. Genest, and É. Marceau. Stochastic bounds on sums of dependent risks. *Insurance: Mathematics and Economics*, 25(1):85–104, 1999. 43, 69

[52] S. P. Firpo and G. Ridder. Bounds on functionals of the distribution treatment effects. Technical Report 201, Escola de Economia de Sẽo Paulo, Getulio Vargas Foundation (Brazil), 2010. 43, 68

[53] P. Embrechts, A. Höing, and A. Juri. Using copula to bound the value-at-risk for functions of dependent risks. *Finance and Stochastics*, 7(2):145–167, 2003. 43, 69

[54] P. Embrechts and G. Puccetti. Aggregating risk capital, with an application to operational risk. *The Geneva Risk and Insurance Review*, 30(2):71–90, 2006. 44, 69

[55] G. Puccetti and L. Rüschendorf. Bounds for joint portfolios of dependent risks. *Statistics & Risk Modeling*, 29(2):107–132, 2012. 44, 70

[56] D. Talay and Z. Zhang. Worst case model risk management. *Finance and Stochastics*, 6:517–537, 2002. 44

[57] M. Kervarec. Modeles non domines en mathematiques financieres. *These de Doctorat en Mathematiques*, 2008. 44

[58] Bion-Nadal J. and Kervarec M. Risk measuring under model uncertainty. *Annals of Applied Probability*, 22(1):213–238, 2012. 44

[59] P. Artzner, F. Delbaen, J. Eber, and D. Heath. Coherent measures of risk. *Mathematical Finance*, 9(3):203–228, 1999. 47, 48

[60] A. Schied. Risk measures and robust optimization problems. Available at www.aschied.de/PueblaNotes8.pdf, 2008. 49

[61] D. Rosen and D. Saunders. Computing and stress testing counterparty credit risk capital. In E. Canabarro, editor, *Counterparty Credit Risk*, pages 245–292. Risk Books, London, 2010. 56, 61, 64

[62] Basel Committee on Banking Supervision (BCBS). Overview of the new basel capital accord. Technical report, Bank for International Settlements, 2001. Available at http://www.bis.org/publ/bcbsca02.pdf. 66

[63] P. Embrechts and M. Hofert. A note on generalized inverses. *Mathematical Methods of Operations Research*, 77(3):423–432, 2013. 66, 78

[64] P. Embrechts, H. Furrer, and R. Kaufmann. Different kinds of risk. In T. G. Andersen, R. A. Davis, J.-P. Kreiß, and T. Mikosch, editors, *Handbook of Financial Time Series*, pages 729–751. Springer, 2009. 66

[65] A. J McNeil, R. Frey, and P. Embrechts. *Quantitative risk management: Concepts, techniques, and tools*. Princeton university press, 2010. 66, 105

[66] M. Hofert and A. J. McNeil. Subadditivity of value-at-risk for Bernoulli random variables. *Statistics & Probability Letters*, 2014. 66

[67] C. Bernard, L. Rüschendorf, and S. Vanduffel. Value-at-risk bounds with variance constraints. *Journal of Risk and Insurance*, 2015. 67

[68] C. Bernard, M. Denuit, and S. Vanduffel. Measuring portfolio risk under partial dependence information. *Journal of Risk and Insurance*, 2016. 67

[69] E. L. Lehmann. Some concepts of dependence. *The Annals of Mathematical Statistics*, pages 1137–1153, 1966. 69

[70] H. Cossette, M. Denuit, and É. Marceau. Distributional bounds for functions of dependent risks. *Schweiz. Aktuarver. Mitt.*, 1:45–65, 2002. 69

[71] P. Embrechts, F. Lindskog, and A. J. McNeil. Modelling dependence with copulas and applications to risk management. In S. Rachev, editor, *Handbook of Heavy Tailed Distributions in Finance*, pages 329–384. Elsevier, 2003. 69

[72] P. Embrechts and G. Puccetti. Bounds for functions of dependent risks. *Finance and Stochastics*, 10(3):341–352, 2006. 69

[73] L. Rüschendorf. Random variables with maximum sums. *Advances in Applied Probability*, 18:623–632, 1982. 69

[74] L Rüschendorf. On the multidimensional assignment problem. *Methods of OR*, 47: 107–113, 1983. 70

[75] L. Rüschendorf. Solution of a statistical optimization problem by rearrangement methods. *Metrika*, 30(1):55–61, 1983. 70, 87

[76] E. Jakobsons, X. Han, and R. Wang. General convex order on risk aggregation. *Scandinavian Actuarial Journal*, 2016(8):713–740, 2016. 70, 74

[77] C. Bernard and D. McLeish. Algorithms for finding copulas minimizing convex functions of sums. *Asia-Pacific Journal of Operational Research*, 33(05):1650040, 2016. 70

[78] R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis*. Springer, 1998. 73

[79] R. Wang, L. Peng, and J. Yang. Bounds for the sum of dependent risks and worst value-at-risk with monotone marginal densities. *Finance and Stochastics*, 17(2): 395–417, 2013. 73

[80] C. Bernard, X. Jiang, and R. Wang. Risk aggregation with dependence uncertainty. *Insurance: Mathematics and Economics*, 54:93–108, 2014. 74, 76

[81] M. Hofert and M. V. Wüthrich. Statistical review of nuclear power accidents. *Asia-Pacific Journal of Risk and Insurance*, 7(1), 2012. doi: 10.1515/2153-3792.1157. 77

[82] V. Chavez-Demoulin, P. Embrechts, and M. Hofert. An extreme value approach for modeling operational risk losses depending on covariates. *Journal of Risk and Insurance*, 2015. 77, 122

[83] G. Leoni. *A first course in Sobolev spaces*. American Mathematical Society Providence, RI, 2009. 79, 80

[84] U. Haus. Bounding stochastic dependence, complete mixability of matrices, and multidimensional bottleneck assignment problems. *arXiv preprint arXiv:1407.6475*, 2014. 87, 96

[85] G. S. Fishman. Variance reduction in simulation studies. *Journal of Statistical Computation and Simulation*, 1(2):173–182, 1972. 95

[86] W. Reiher. Monte carlo methods. *Biometrical Journal*, 8(3):209–209, 1966. 95

[87] B. Wang and R. Wang. The complete mixability and convex minimization problems with monotone marginal densities. *Journal of Multivariate Analysis*, 102(10):1344–1360, 2011. 96

[88] A. C. Davison and D. V. Hinkley. *Bootstrap methods and their application*, volume 1. Cambridge university press, 1997. 101

[89] P. Jorion. *Value at risk: the new benchmark for managing financial risk*, volume 3. McGraw-Hill New York, 2007. 106

[90] S. V. Stoyanov, S. T. Rachev, B. Racheva-Yotova, and F. J. Fabozzi. Fat-tailed models for risk estimation. *The Journal of Portfolio Management*, 37(2):107–117, 2011. 106

[91] Z. Rachev, B. Rocheva-Yotova, and S. Stoyanov. Capturing fat tails. *Risk*, 23(5):72, 2010. 106

[92] P. Embrechts and M. Hofert. Practices and issues in operational risk modeling under Basel II. *Lithuanian mathematical journal*, 51(2):180–193, 2011. 113

[93] Basel Committee on Banking Supervision (BCBS). Operational risk - revisions to the simpler approaches - consultative document. Technical report, Bank for International Settlements, 2014. Available at http://www.bis.org/publ/bcbs291.pdf. 122

[94] D. Rosen and D. Saunders. *Re-Thinking CVA: Valuations, Counterparty Credit Risk, and Model Risk*, pages 183–227. Risk books, (E. Canabarro and M. Pykhtin, Editor), 2014. 134, 136

[95] S. L. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of financial studies*, 6(2):327–343, 1993. 134

[96] J. C. Hull. *Options, Futures, and Other Derivatives*. Prentice Hall Series in Finance, 7 edition, 2008. 138

[97] R.T. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2(3):21–42, 2000. 141

[98] J. Hartung. An extension of Sion's minimax theorem with an application to a method for constrained games. *Pacific J. Math*, 103(2):401–408, 1982. 144

[99] M. Sion. On general minimax theorems. *Pacific J. Math*, 8(1):171–176, 1958. 144

[100] ILOG Inc. Ilog cplex 12.6. *User Manual*, 2013. URL `https://www.ibm.com/support/knowledgecenter/SSSA5P_12.6.1/ilog.odms.studio.help/pdf/usrcplex.pdf`. 151

Appendix A

Model Uncertainty and CVA

In this appendix we highlight important issues surrounding model uncertainty and the effect of misspecification of parameters on counterparty exposure profile through the following example based on the work done in Rosen and Saunders [94].

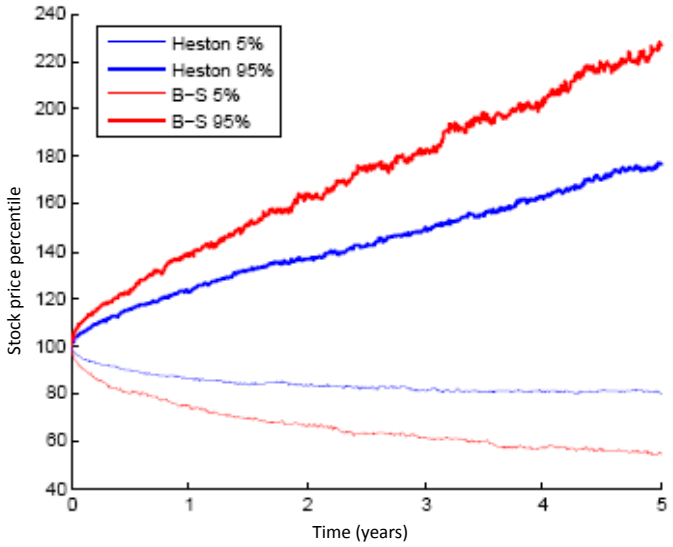


FIGURE A.1: *Percentiles of the stock index under Heston and Black-Scholes models*

Consider a portfolio consisting of a single European call option on an equity index, assuming that the true data generating process (DGP) for the equity index price is a Heston's stochastic volatility model (see Heston [95]):

$$\begin{aligned} dS_t &= rS_t dt + \sqrt{\nu_t} dW_t \\ d\nu_t &= \kappa(\theta - \nu_t)dt + \xi\sqrt{\nu_t} dZ_t \end{aligned}$$

where W_t and Z_t are correlated standard Brownian motions with correlation ρ. Table A.1 provides a summary of the parameters used in our simulations. First consider a scenario

Parameter	Notation	Default value
Initial stock index price	S_0	100
Initial volatility	ν_0	0.0375
Reversion rate of volatility	κ	1
Long-run volatility	θ	0.0375
Volatility-of-volatility	ξ	0.35
Correlation between Z_t and W_t	ρ	-0.3
Risk-free rate	r	4%

TABLE A.1: *Heston model base case parameters*

in which the analyst may decide to use a simpler model in which the equity index follows a geometric Brownian motion and not a Heston model; this is an instance of possible misspecification of the stochastic model for a given set of underlying parameters.

$$dS_t = rS_t dt + \sigma s_t dW_t$$

Assume an initial stock index value of $S = 100$ with volatility $\sigma = \sqrt{\theta} = 0.1936$. Using 1,000 simulated scenarios over a 5 year period, we graph the standard deviation and percentiles of these stock indices. As can be seen in figures A.1 and A.2, the simulations based on Heston and Black-Scholes generate noteworthy differences; using simpler Black-Scholes dynamics clearly results in larger volatility; this volatility increases as time increases.

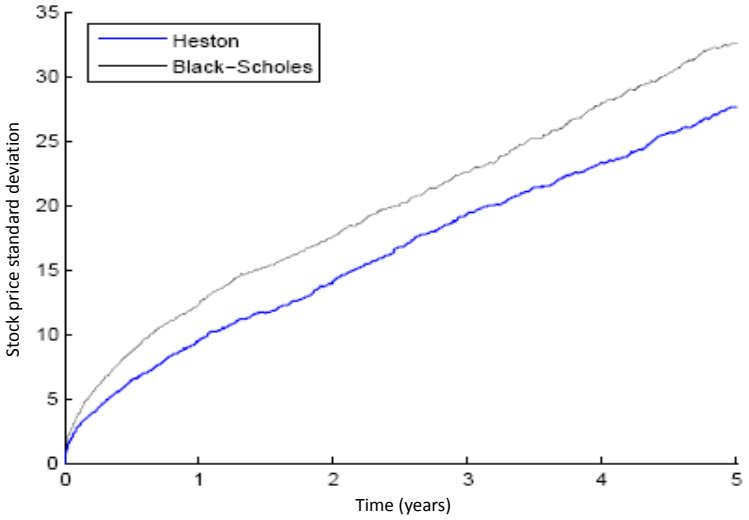


FIGURE A.2: *Standard deviation of the stock index under Heston and Black-Scholes models*

Next we analyze the impact of parameter estimation error while the model is correctly specified. To demonstrate this, we use the Heston model with base-case parameters presented in table A.1 and compare it against the case in which long-run volatility is mis-specified, assuming the mis-specified parameter values are $\nu_0 = 0.01$ and $\theta = 0.01$ respectively.

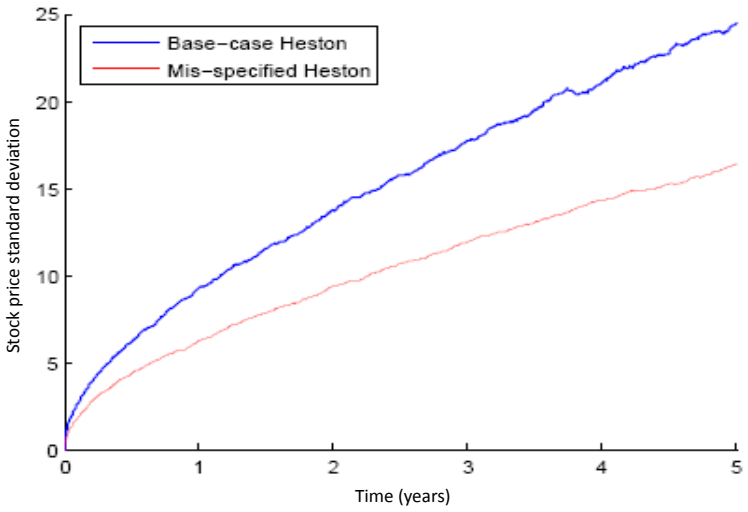


FIGURE A.3: *Standard deviation of stock index under base-case and misspecified Heston models*

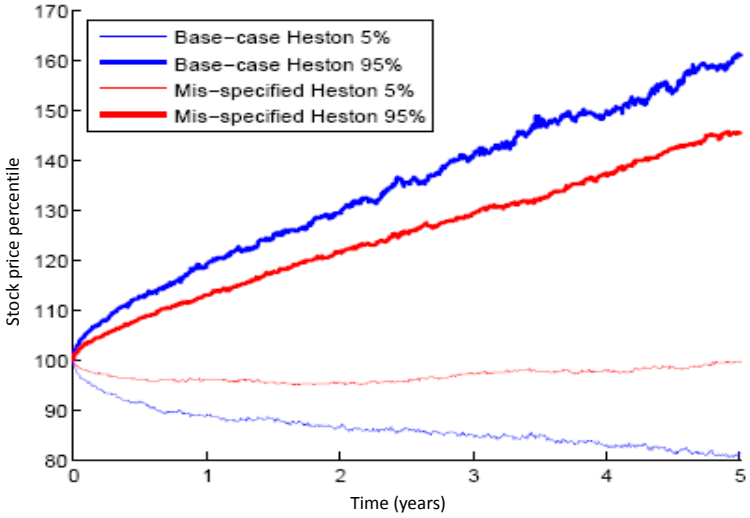


FIGURE A.4: *Percentiles of the stock index under base-case and misspecified Heston model*

As can be seen in figures A.4 and A.3, even if the model used by the analyst is the right one, but incorrect parameters are used, quiet large differences are introduced in pricing process; further examples and an in-depth discussion of issues surrounding model uncertainty and CVA calculation are given in Rosen and Saunders [94].

A.0.1 Implications of Model Uncertainty in Practice

In section A only a few simple implications of model and parameters misspecification when the portfolio consisted of only *one* instrument was investigated; In practice a given portfolio can potentially have thousands of transactions and *scenario generation, instrument valuation* and *aggregation at portfolio level* play an important role in calculating CVA. Scenario generation, valuation models and computational feasibility of pricing instruments across future scenarios are among the most important challenges that one faces in calculating CVA in practice. Since Monte Carlo frameworks are the most commonly used tools by financial institutions in CVA calculations, we briefly look at some of these challenges in implementing them; the process usually involves:

- identifying the underlying risk factors and conducting (forward or spot) simulation of prices, using a specific model and the existing correlations to generate scenarios;
- combining the previous information and utilizing existing pricing functions for valuation of the instruments;
- calculating CVA charge and other statistical quantities (such as CVA VaR).

Although the steps described above seem fairly straightforward, several issues should be considered in implementing them:

Scenario Generation: The same underlying model (the family of models accordingly) should be used to generate *consistent* scenarios across the portfolio; even if the correlation between various asset classes is ignored (assuming for example that equities are not correlated with interest rate products), using the same discount factors as those used for FX or interest rates can be very difficult to implement across various simulation processes.

Valuation Model: calculating counterparty exposures depends on the entire simulated scenario history; different formats used in different pricing systems are not designed to accommodate the integration of counterparty exposure calculations, which are an important input to the CVA calculation.

Computational Complexity: if there is no analytical solution for pricing a product, Monte Carlo methods or solving PDEs over grids are typically used for pricing. Computing exposure in this case would consist of a Monte Carlo simulation for various scenarios and *then* using the existing Monte Carlo or PDE pricing method for each instrument; this is simply not feasible computationally in many cases.

To summarize, computing CVA is among the most complex computational problems in practice; the uncertainty of future credit exposures due to the sensitivity of various instruments to market parameters, combined with the fact that both the current value of the instrument and its future value across different time steps are the main drivers in this. In the next section we take a closer look at the principles of CVA calculation in practice.

A.1 CVA: A Simple Example

The following is based on an example in Hull [96]. The credit exposure on a derivatives transaction with counterparty C from the bank's perspective falls into one of the following three cases:

1. The derivatives contract with counterparty C is always a liability for the bank B. An example of a derivative in this category is a short option position. Derivatives in this category have no credit risk to the bank B if the counterparty C defaults and there will be no losses for the bank since the derivative is one of the counterparty's assets and it will be either retained, closed out, or sold to a third party.

2. The derivatives contract with counterparty C is always an asset for the bank B. An example of a derivative in this category is a long option position. Derivatives in this category bear credit risk for the bank B if the counterparty C defaults and there will be potential losses for the bank. In this case the bank has to make a claim against the assets of the counterparty and may receive a certain percentage of the value of the derivative.

3. The derivatives contract with counterparty C can be either an asset or a liability for the bank B. A forward contract falls in this category. A derivative in this category may or may not have credit risk for the bank. If the counterparty C defaults when the value is negative for the bank, the bank experiences no loss, while if the counterparty C defaults when the value of the derivative is positive for the bank, it experiences loss and can claim the remaining assets of the counterparty C.

CVA represents the price adjustment in the value of a derivative instrument when the counterparty can default. Consider that our portfolio with counterparty C consists of only one derivative, maturing at time T, and that today's value of this derivative, assuming no defaults, is f_0. Assume for the sake of simplicity that defaults can only occur at a set of discrete points in time namely $t_1, t_2, \ldots, t_n = T$ and the respective value of the derivative contract at these time steps, assuming no default, is $f_1, f_2, \ldots, f_n$. Let the risk-neutral probability of default at each time step t_i be $q_i, (i = 1, \ldots, n)$ and the expected recovery rate, in case of a default, be R_{C}.

At time t_i, the exposure of the bank to the counterparty C is its potential loss, i.e., $\max\{f_i, 0\}$. Assume further that the recovery rate and the probability of default of counterparty C are independent of the derivative's value. If the counterparty C defaults at time t_i the expected recovery is $R_{\mathrm{C}} \max\{f_i, 0\}$. Then the risk-neutral expected loss from the default of counterparty C at time t_i is given by:

$$L = q_i(1 - R_{\mathrm{C}}) \,\mathrm{E}[\max\{f_i, 0\}] \tag{A.1}$$

where E[.] represents the expected value with respect to the risk-neutral probabilities. Unilateral CVA (CVA^U) is simply found by taking the present values of equation (A.1):

$$\mathrm{CVA}^U = \sum_{i=1}^{n} q_i(1 - R_{\mathrm{C}})v_i \tag{A.2}$$

where v_i represents the value of an instrument that pays off the exposure on the derivative at time t_i for which we are computing CVA.

Let us consider the value of the derivative, f_i, at the default time t_i in each of the three cases presented earlier: Since in the first case the derivative is a liability for the bank and the value of f_i is always negative, the total expected loss from the default of counterparty C given by equation (A.2) is zero and no adjustments are made by the bank for the defaults in this case.

In the second case that the derivative is an asset for the bank and the value of the derivative, f_i, is always positive and in equation (A.1) we have $\max\{f_i, 0\} = f_i$. For the sake of simplicity assume that the only payoff from the derivative is at time T. Then f_0 must be equal to the present value of f_i such that $v_i = f_0, (i = 1, 2, \ldots, n)$. Therefore equation (A.2) for the present value of the cost of default of counterparty C becomes:

$$\text{CVA}^U = f_0 \sum_{i=1}^{n} q_i (1 - R_\text{C})$$

Let f_0^* denote the value of the derivative, allowing for the default of counterparty C, then:

$$f_0^* = f_0 - f_0 \sum_{i=1}^{n} q_i (1 - R_\text{C}) = f_0 \left(1 - \sum_{i=1}^{n} q_i (1 - R_\text{C}) \right) \tag{A.3}$$

An example of an instrument in the second case is an unsecured zero-coupon bond issued by the counterparty C that pays \$1 at the maturity time T. Let B_0 be the value of the bond assuming no possibility of default for the counterparty C and B_0^* be the value of the bond when the counterparty can default. If we assume that the recovery rate R_C on the bond as a percentage of its no-default value is the same as that on the derivative then we have:

$$B_0^* = B_0 \left(1 - \sum_{i=1}^{n} q_i (1 - R_\text{C}) \right) \tag{A.4}$$

and using equations (A.3) and (A.4)

$$\frac{f_0^*}{f_0} = \frac{B_0^*}{B_0} \tag{A.5}$$

Assuming that y is the yield on a risk-free zero-coupon bond that matures at time T and y^* is the yield on the zero-coupon bond maturing at T, issued by counterparty C, we have $B_0 = \exp(-yT)$ and $B_0^* = \exp(-y^*T)$. Therefore equation (A.5) becomes:

$$f_0^* = f_0 \exp(-(y^* - y)T) \tag{A.6}$$

Equation (A.6) simply indicates that an increased discount rate of y^* should be applied to a derivative that matures at time T compared to the discount rate of y, applied in a risk-neutral world. For example suppose that an over-the-counter option with maturity of $T = 5$ is sold by counterparty C (without possibility of default) for \$10. Moreover, a 5-year zero-coupon bond issued by counterparty C has a yield that is 2% greater than that of a

risk-free zero-coupon bond maturing in $T = 5$ years. In this case the value of the option, assuming the possibility of the default of counterparty C, is $\$10 \exp(-0.02 \times 5) = \$9.05$. Hence in this case $\text{CVA}^U = \$10 - \$9.05 = \$0.95$.

In the third case, where the derivative contract with counterparty C can be either an asset or a liability for the bank B, the sign of f_i is uncertain and the variable v_i in equation (A.2) is seen as a call option, with a strike price of zero, on f_i. While it is possible to analytically calculate v_i for certain derivatives, simulating the underlying market risk factors over the life of the derivative and estimating v_i is the most common method used in practice.

Appendix B

Derivation of Worst-Case Conditional Value at Risk

Recall that by using $\rho = \mathrm{CVaR}_\alpha$ as our choice of risk measure in 3.5.1 the worst case joint distribution problem can be recast as:

$$\begin{aligned} &\max_{\vartheta} \mathrm{CVaR}_\alpha(L) \\ &\sum_{i=1}^{N} \vartheta_{ij} = p_j, \quad j = 1, \ldots, M \\ &\sum_{j=1}^{M} \vartheta_{ij} = q_i, \quad i = 1, \ldots, N \\ &\vartheta_{ij} \geqslant 0, \quad i = 1, \ldots, N, \quad j = 1, \ldots, M \end{aligned} \tag{B.1}$$

in which $\vartheta \in \mathfrak{F}(F_Y, F_Z)$, the *Fréchet class* of all possible joint distributions of the market risk factors Y and credit risk factors Z, matching the given marginals p_j, $j = 1, \ldots, M$ and q_i, $i = 1, \ldots, N$ respectively. Using the discretized representation of CVaR_α (see Rockafellar and Uryasev [97]) the above problem can be simplified to:

$$\begin{aligned} &\max_{\vartheta} \left(\min_{x} \left(x + (1-\alpha)^{-1} \sum_{i=1}^{N} \sum_{j=1}^{M} \vartheta_{ij} [l_{ij} - x]^+ \right) \right) \\ &\sum_{i=1}^{N} \vartheta_{ij} = p_j \quad j = 1, \ldots, M \\ &\sum_{j=1}^{M} \vartheta_{ij} = q_i \quad i = 1, \ldots, N \\ &\vartheta_{ij} \geqslant 0, \quad i = 1, \ldots, N, \quad j = 1, \ldots, M \end{aligned}$$

where $[l_{ij} - x]^+ = \max\{l_{ij} - x, 0\}, i = 1, \ldots, N, j = 1, \ldots, M$. By applying a minimax theorem to (B.1) (See appendix C for details), this is equivalent to solving:

$$\min_x \left(x + (1-\alpha)^{-1} \max_{\vartheta} \sum_{i=1}^{N} \sum_{j=1}^{M} \vartheta_{ij} [l_{ij} - x]^+ \right) \tag{B.2}$$

$$\sum_{i=1}^{N} \vartheta_{ij} = p_j, \quad j = 1, \ldots, M$$

$$\sum_{j=1}^{M} \vartheta_{ij} = q_i, \quad i = 1, \ldots, N$$

$$\vartheta_{ij} \geqslant 0, \quad i = 1, \ldots, N, \quad j = 1, \ldots, M$$

using the notation $l_{ij}^x = [l_{ij} - x]^+$ the inner maximization can be written as:

$$\max_{\vartheta} \sum_{i=1}^{N} \sum_{j=1}^{M} \vartheta_{ij} l_{ij}^x$$

$$\sum_{i=1}^{N} \vartheta_{ij} = p_j, \quad j = 1, \ldots, M$$

$$\sum_{j=1}^{M} \vartheta_{ij} = q_i, \quad i = 1, \ldots, N$$

$$\vartheta_{ij} \geqslant 0, \quad i = 1, \ldots, N, \quad j = 1, \ldots, M$$

Taking the dual of this linear program (and by letting $p = (p_1, \ldots, p_M), \psi = (\psi_1, \ldots, \psi_M), q = (q_1, \ldots, q_N)$ and $\theta = (\theta_1, \ldots, \theta_N)$) yields:

$$\min_{\psi, \theta} (p \cdot \psi + q \cdot \theta)$$

$$\psi_m + \theta_n \geqslant l_{nm}^x, \quad n = 1, ..., N, \quad m = 1, ..., M$$

The optimal ϑ can be recovered as the optimal dual variables of the above problem. Denoting the constraint set of the above LP by $\widetilde{C}_x$, and using the duality theorem for linear programming, problem B.2 becomes:

$$\min_x \left(x + (1-\alpha)^{-1} \min_{(\psi,\theta) \in \widetilde{C}_x} p \cdot \psi + q \cdot \theta \right) = \min_{x, (\psi,\theta) \in \widetilde{C}_x} \left(x + (1-\alpha)^{-1} (p \cdot \psi + q \cdot \theta) \right)$$

We can rewrite this single optimization problem more explicitly as:

$$\min_{x, \psi, \theta} \left(x + (1-\alpha)^{-1} (p \cdot \psi + q \cdot \theta) \right)$$

$$\psi_m + \theta_n \geqslant l_{nm}^x, \quad n = 1, ..., N, \quad m = 1, ..., M$$

To convert this into a linear program we introduce the auxiliary variables ϵ_{nm}:

$$\begin{gathered}
\min_{x,\psi,\theta,\epsilon} \Big(x + (1-\alpha)^{-1}(p\cdot\psi + q\cdot\theta)\Big) \\
\psi_m + \theta_n \geqslant \epsilon_{mn}, \quad n = 1,...,N, \quad m = 1,...,M \\
\epsilon_{nm} \geqslant l_{nm} - x, \quad n = 1,...,N, \quad m = 1,...,M \\
\epsilon_{nm} \geqslant 0, \qquad n = 1,...,N, \quad m = 1,...,M
\end{gathered}$$

Thus, the problem is reduced to a single large linear program (with $MN + N + M + 1$) variables and $3MN$ constraints). The optimal ϑ appears as the dual variables of the first set of the constraints in the optimization problem above.

We can simplify the above problem further. Taking the dual of the above LP yields:

$$\begin{gathered}
\max_{z,\nu} \quad \sum_{i=1}^{N}\sum_{j=1}^{M} l_{ij}.\nu_{ij} \\
\sum_{i=1}^{N} z_{ij} = (1-\alpha)^{-1} p_j, \quad j = 1,\dots,M \\
\sum_{j=1}^{M} z_{ij} = (1-\alpha)^{-1} q_i, \quad i = 1,\dots,N \\
\nu_{ij} - z_{ij} \leqslant 0 \quad i = 1,\dots,N, \quad j = 1,\dots,M \\
\sum_{i=1}^{N}\sum_{j=1}^{M} \nu_{ij} = 1 \\
z_{ij} \geqslant 0, \quad \nu_{ij} \geqslant 0, \quad i = 1,\dots,N, \quad j = 1,\dots,M
\end{gathered}$$

Making the substitutions $\vartheta_{ij} = (1-\alpha)z_{ij}$ and $\mu_{ij} = (1-\alpha)\nu_{ij}$, $i = 1,\dots,N$, $j = 1,\dots,M$ leads to a much more straightforward and intuitive formulation of the problem of finding the worst case joint distribution for CVaR for a joint distribution with the given marginals.

$$\begin{gathered}
\max_{\vartheta,\mu} \quad \frac{1}{1-\alpha}\sum_{i=1}^{N}\sum_{j=1}^{M} l_{ij}\cdot\mu_{ij} \\
\sum_{i=1}^{N} \vartheta_{ij} = p_j, \quad j = 1,\dots,M \\
\sum_{j=1}^{M} \vartheta_{ij} = q_i, \quad i = 1,\dots,N \\
\sum_{i=1}^{N}\sum_{j=1}^{M} \mu_{ij} = 1-\alpha, \\
0 \leqslant \mu_{ij} \leqslant \vartheta_{ij}, \quad i = 1,\dots,N, \quad j = 1,\dots,M
\end{gathered}$$

Appendix C

Proof of Switching Min-Max in B.1

To show that:

$$\max_{\vartheta\in\mathcal{P}}\min_{x\in\mathbb{R}}\left(x+(1-\alpha)^{-1}\sum_{i=1}^{N}\sum_{j=1}^{M}\vartheta_{ij}[l_{ij}-x]^{+}\right)=$$

$$\min_{x\in\mathbb{R}}\max_{\vartheta\in\mathcal{P}}\left(x+(1-\alpha)^{-1}\sum_{i=1}^{N}\sum_{j=1}^{M}\vartheta_{ij}[l_{ij}-x]^{+}\right) \tag{C.1}$$

where $\mathcal{P}$ is

$$\mathcal{P}=\left\{\vartheta\in\mathbb{R}^{NM}\Big|\sum_{i=1}^{N}\vartheta_{ij}=p_j,\quad\sum_{j=1}^{M}\vartheta_{ij}=q_i,\quad\vartheta_{ij}\geqslant 0,\quad j=1,\ldots,M\quad i=1,\ldots,N\right\} \tag{C.2}$$

$\mathcal{P}$ represents the *Fréchet class* of all possible joint distributions of market risk factors, Y, and credit risk factors, Z, matching the given marginal distributions, we apply an extension of Sion's minimax theorem, found in [98].

Definition C.0.1. *A function $f: X\to\mathbb{R}$ is called:*

- *inf-compact if $\{x|x\in\mathbb{X}, f(x)\leqslant a\}$, is compact for all $a\in\mathbb{R}$,*
- *sup-compact if $\{x|x\in\mathbb{X}, f(x)\geqslant a\}$, is compact for all $a\in\mathbb{R}$.*

A function $f:\mathbb{X}\times\mathbb{Y}\to\mathbb{R}$ is called (x,y) sup inf-compact, if for all $(x,y)\in\mathbb{X}\times\mathbb{Y}$, $f(x,.)$ is inf-compact and $f(.,y)$ is sup-compact.

If $f:\mathbb{X}\times\mathbb{Y}\to\mathbb{R}$ is u.s.c.-l.s.c. (upper semi-continuous-lower semi-continuous), i.e., $f(x,y)$ is upper semi-continuous in x for each $y\in\mathbb{Y}$ and lower semi-continuous in y for each $x\in\mathbb{X}$ and $\mathbb{X}$ and $\mathbb{Y}$ are compact sets, then $f(x,y)$ is (x,y)-sup inf-compact for all $(x,y)\in\mathbb{X}\times\mathbb{Y}$. We use the following theorem, a generalization of the Sion's theorem (see [99]):

Theorem C.0.1. *Let $\mathbb{X}$ and $\mathbb{Y}$ be convex sets, and let $f : \mathbb{X} \times \mathbb{Y} \to \mathbb{R}$ be an u.s.c.-l.s.c. and quasi-concave-convex function, that is (x, y)-sup inf-compact. Then we have*

$$\max_{x\in\mathbb{X}} \min_{y\in\mathbb{Y}} f(x, y) = \min_{y\in\mathbb{Y}} \max_{x\in\mathbb{X}} f(x, y)$$

Take $\mathbb{X} = \mathcal{P}$:

$$\mathcal{P} = \mathbb{X} = \left\{ (\vartheta_1, \ldots, \vartheta_{MN}) \quad \Big| \sum_{i=1}^{N} \vartheta_{ij} = p_j, \quad \sum_{j=1}^{M} \vartheta_{ij} = q_i, \quad \vartheta_i \geqslant 0 \right\}$$

To show that $\mathcal{P}$ is convex, consider $\vartheta^1, \vartheta^2 \in \mathcal{P}$ and $\lambda \in (0, 1)$. Clearly $\lambda\vartheta^1 + (1-\lambda)\vartheta^2 \geqslant 0$ as using (C.2) $\mathcal{P}$ can be written as $A\vartheta = b, \vartheta \geqslant 0$, so it is convex and closed, i.e.,

$$\lambda\vartheta^1 + (1-\lambda)\vartheta^2 \geqslant 0$$

and

$$A(\lambda\vartheta^1 + (1-\lambda)\vartheta^2) = \lambda A\vartheta^1 + (1-\lambda)A\vartheta^2 = b$$

implying that $\mathcal{P}$ is convex. Moreover, let

$$\vartheta^n \in \mathcal{P} \quad \text{and} \quad \vartheta^n \to \vartheta$$

$$\vartheta^n \geqslant 0 \Rightarrow \vartheta \geqslant 0 \quad \text{and} \quad b = \lim_{n\to\infty} A\vartheta^n = A \lim_{n\to\infty} \vartheta^n = A\vartheta$$

hence $\mathcal{P}$ is clearly bounded and therefore compact and closed. Next we show that

$$\max_{\vartheta\in\mathcal{P}} \min_{x\in\mathbb{R}} \left(x + (1-\alpha)^{-1} \sum_{i=1}^{N} \sum_{j=1}^{M} \vartheta_{ij} [l_{ij} - x]^+ \right) =$$

$$\max_{\vartheta\in\mathcal{P}} \min_{x\in\mathbb{Y}} \left(x + (1-\alpha)^{-1} \sum_{i=1}^{N} \sum_{j=1}^{M} \vartheta_{ij} [l_{ij} - x]^+ \right)$$

where

$$\mathbb{Y} \;=\; \Big\{ y \in \mathbb{R}^+ \big| \min(L) \leqslant y \leqslant \max(L) \Big\}$$

and for a fixed x

$$L = \{\max(l_k - x, 0), \quad k = 1, \ldots, MN\}$$

i.e. in our computational problem, y would be restricted between the minimum and maximum of the loss vector, because for a fixed $\vartheta \in \mathcal{P}$ (and respectively constant l_{ij}), since $\sum_{i=1}^{N} \sum_{j=1}^{M} \vartheta_{ij} = 1$, we have:

$$\lim_{x\to+\infty} x + (1-\alpha)^{-1} \sum_{i=1}^{N} \sum_{j=1}^{M} \vartheta_{ij} [l_{ij} - x]^+ = +\infty$$

and since the confidence level $0 < \alpha < 1$, we get

$$\lim_{x \to -\infty} x + (1-\alpha)^{-1} \sum_{i=1}^{N} \sum_{j=1}^{M} \vartheta_{ij} [l_{ij} - x]^+ = +\infty$$

Furthermore, let $f(\cdot, \cdot)$ be defined as:

$$f(\vartheta, x) \quad := \quad x + (1-\alpha)^{-1} \sum_{i=1}^{N} \sum_{j=1}^{M} \vartheta_{ij} [l_{ij} - x]^+ \tag{C.3}$$

and $L_{\min} = \min(L)$ and $L_{\max} = \max(L)$. Then for a fixed $\vartheta \in \mathcal{P}$,
If $x > L_{\max}$

$$f(\vartheta, x) = x > L_{\max} = f(\vartheta, L_{\max})$$

If $x < L_{\min}$

$$\begin{aligned} f(\vartheta, x) &= (1 - (1-\alpha)^{-1}) x + (1-\alpha)^{-1} \sum_{i=1}^{N} \sum_{j=1}^{M} \vartheta_{ij} [l_{ij} - x]^+ \\ &> (1 - (1-\alpha)^{-1}) L_{\min} + (1-\alpha)^{-1} \sum_{i=1}^{N} \sum_{j=1}^{M} \vartheta_{ij} L_{\min} \end{aligned}$$

implying that

$$\min_{x \in \mathbb{R}} f(\vartheta, x) = \min_{x \in [L_{\min}, L_{\max}]} f(\vartheta, x) \quad \text{for all} \quad \vartheta \in \mathcal{P}$$

Let

$$g(x) = \max_{\vartheta \in \mathcal{P}} \left(x + (1-\alpha)^{-1} \sum_{i=1}^{N} \sum_{j=1}^{M} \vartheta_{ij} [l_{ij} - x]^+ \right)$$

If $x > L_{\max}$

$$\begin{aligned} g(x) &= \max_{\vartheta \in \mathcal{P}} \left(x + (1-\alpha)^{-1} \sum_{i=1}^{N} \sum_{j=1}^{M} \vartheta_{ij} [l_{ij} - x]^+ \right) = \max_{\vartheta \in \mathcal{P}} x = x \\ &> L_{\max} = g(L_{\max}) \end{aligned}$$

If $x < L_{\min}$

$$\begin{aligned}
g(x) &= \max_{\vartheta \in \mathcal{P}} \left(x + (1-\alpha)^{-1} \sum_{i=1}^{N} \sum_{j=1}^{M} \vartheta_{ij} [l_{ij} - x]^+ \right) \\
&= \max_{\vartheta \in \mathcal{P}} \left((1-(1-\alpha)^{-1}) x + (1-\alpha)^{-1} \sum_{i=1}^{N} \sum_{j=1}^{M} \vartheta_{ij} [l_{ij} - x]^+ \right) \\
&= (1-(1-\alpha)^{-1}) x + \max_{\vartheta \in \mathcal{P}} \left((1-\alpha)^{-1} \sum_{i=1}^{N} \sum_{j=1}^{M} \vartheta_{ij} [l_{ij} - x]^+ \right) \\
&> (1-(1-\alpha)^{-1}) L_{\min} + \max_{\vartheta \in \mathcal{P}} \left((1-\alpha)^{-1} \sum_{i=1}^{N} \sum_{j=1}^{M} \vartheta_{ij} [l_{ij} - x]^+ \right) \\
&= \max_{\vartheta \in \mathcal{P}} \left(L_{\min} + (1-\alpha)^{-1} \sum_{i=1}^{N} \sum_{j=1}^{M} \vartheta_{ij} (l_{ij} - L_{\min}) \right) = g(L_{\min})
\end{aligned}$$

Therefore $\mathbb{Y}$ can be restricted to a compact interval in $\mathbb{R}$ and would be convex and compact as well.

Next we show that $f(\cdot, \cdot)$ defined in (C.3) is upper semi-continuous with respect to ϑ for each fixed x and lower semi-continuous with respect to x for each fixed ϑ. This is easily verified, as for a fixed x we have a linear function with respect to ϑ. Alternatively, for a fixed ϑ, we obtain a continuous piecewise linear function with respect to x and joint continuity of $f(\cdot, \cdot)$ is attained.

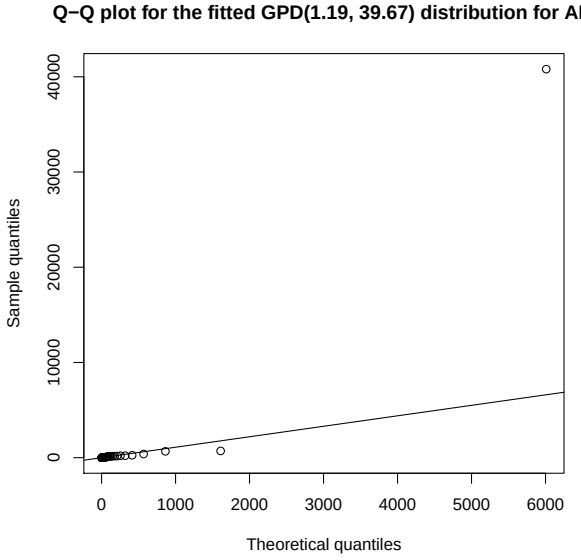
Lastly we demonstrate that $f(\cdot, \cdot)$ is a quasi-concave-convex function, i.e., it is quasi-concave with respect to ϑ for a fixed x and quasi-convex with respect to x for each fixed ϑ. For a fixed x, f is linear with respect to ϑ_{ij}, so it would be quasi-concave. Furthermore, for a fixed ϑ, since we restricted x to $[L_{\min}, L_{\max}]$, $f(\vartheta, \cdot)$ is a sum of nonnegative constants multiplied by convex function and is therefore convex, hence quasi-convex.

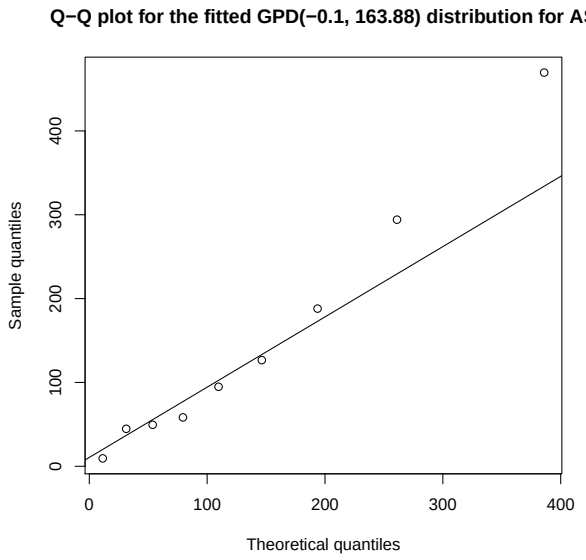
Appendix D

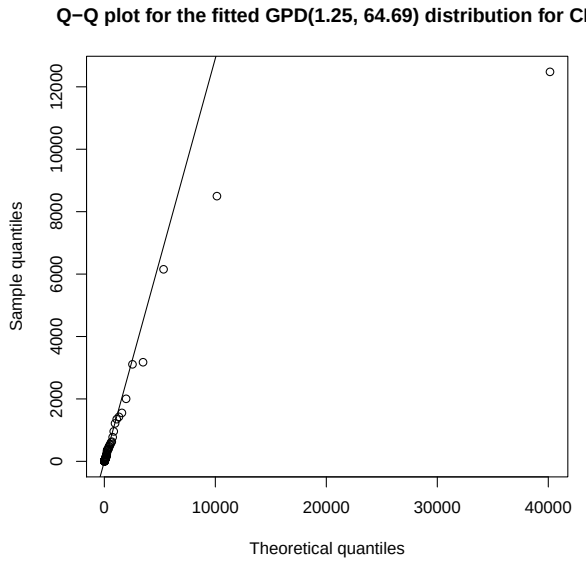
Q-Q plots for the Fitted Generalized Pareto Distribution to Op-Risk Data

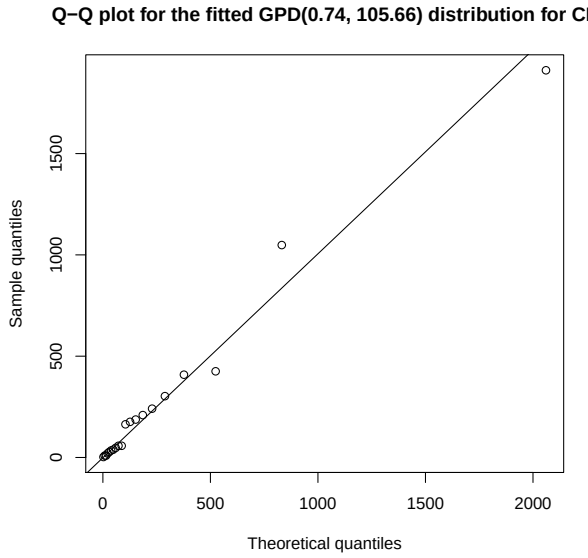
The 10 business lines that are used as the 10 risk factors in section 6.5 are Agency Services (AS), Asset Management (AM), Commercial Banking (CB), Corporate Finance (CF), Insurance (I), Payment and Settlement (PS), Retail Banking (RBa), Retail Brokerage (RBr), Trading and Sales (TS), or an unallocated business line (UBL).

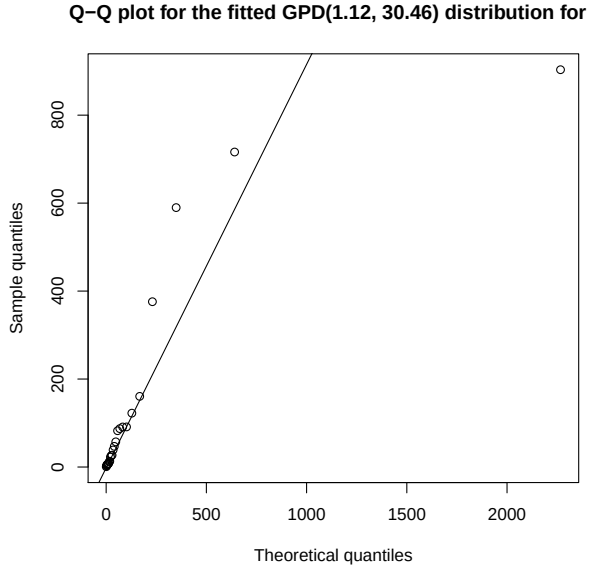
The Q-Q plots of fitted Generalized Pareto Distribution to each of these risk factors, as well as the estimated parameters for the estimated GPD are shown below.

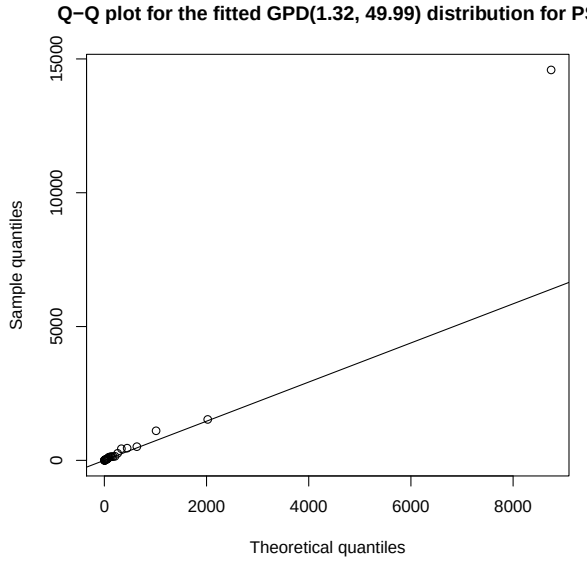


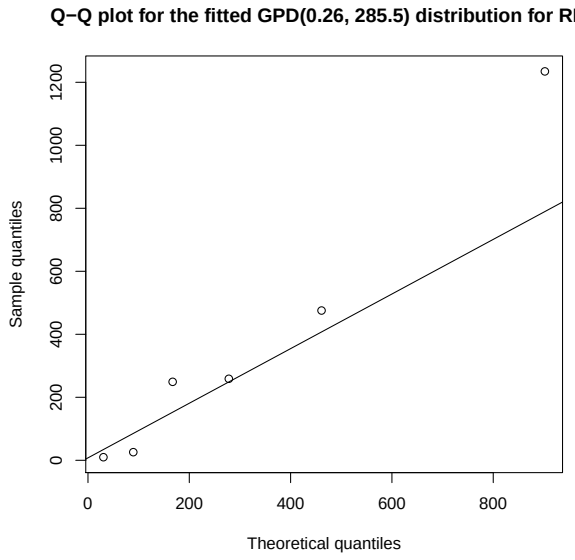


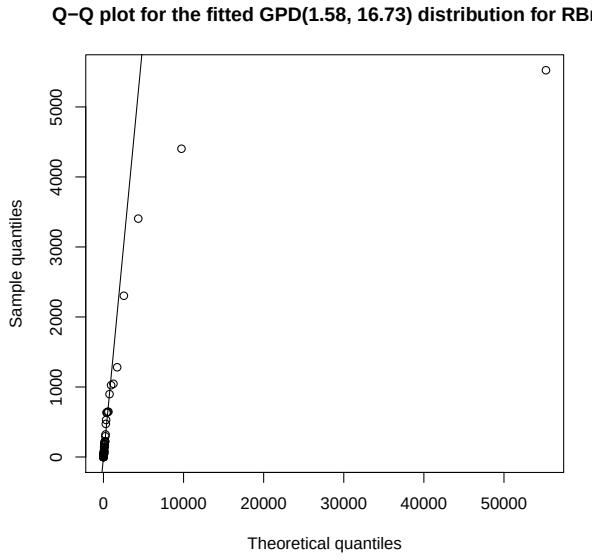

Q−Q plot for the fitted GPD(1.25, 64.69) distribution for CB
Sample quantiles
Theoretical quantiles
0
2000
4000
6000
8000
10000
12000
0
10000
20000
30000
40000

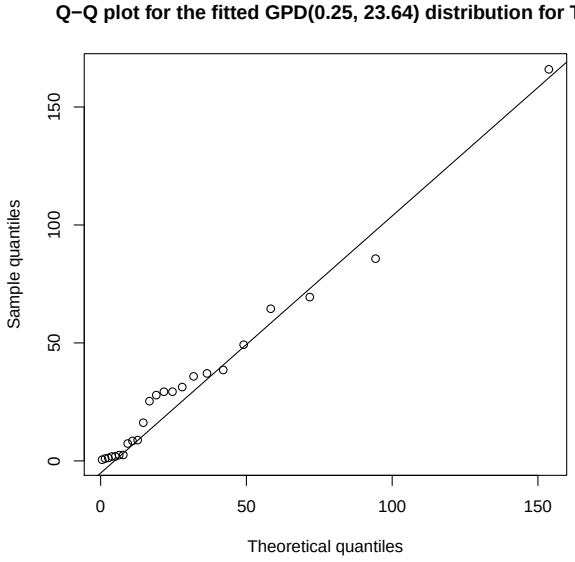

Q−Q plot for the fitted GPD(0.74, 105.66) distribution for CF
Sample quantiles
Theoretical quantiles
0
500
1000
1500
0
500
1000
1500
2000

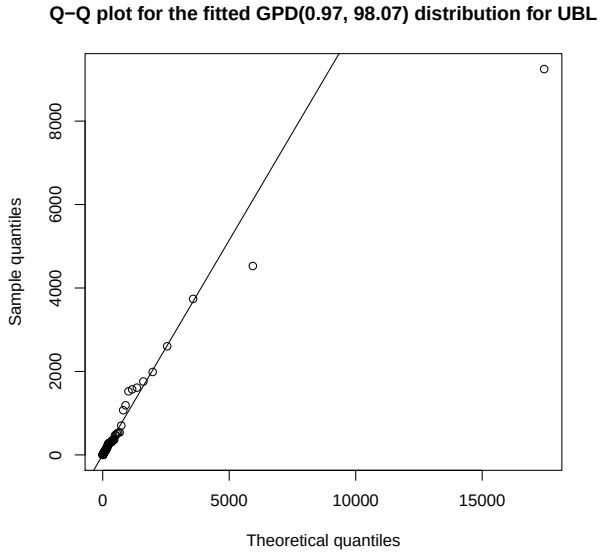

Q−Q plot for the fitted GPD(1.12, 30.46) distribution for I
Sample quantiles
Theoretical quantiles
0
200
400
600
800
0
500
1000
1500
2000


Q−Q plot for the fitted GPD(1.32, 49.99) distribution for PS
Sample quantiles
Theoretical quantiles
0
5000
10000
15000
0
2000
4000
6000
8000


Q−Q plot for the fitted GPD(0.26, 285.5) distribution for RBa
Sample quantiles
Theoretical quantiles
0
200
400
600
800
1000
1200
0
200
400
600
800


Q−Q plot for the fitted GPD(1.58, 16.73) distribution for RBr
Sample quantiles
Theoretical quantiles
0
1000
2000
3000
4000
5000
0
10000
20000
30000
40000
50000


Q−Q plot for the fitted GPD(0.25, 23.64) distribution for TS
Sample quantiles
Theoretical quantiles
0
50
100
150


Q−Q plot for the fitted GPD(0.97, 98.07) distribution for UBL
Sample quantiles
Theoretical quantiles
0
2000
4000
6000
8000
0
5000
10000
15000

Appendix E

CPLEX Optimizer

In order to solve linear programming problems (2.6.11), (2.6.22), (2.7.13) and (2.7.14) in chapter 2 and (4.1.6) in chapter 3, we have used IBM ILOG CPLEX Optimization Studio. CPLEX Optimization Studio solves linearly constrained optimization problems where the objective is expressed as a linear function. More specifically it solves all linear programming problems expressed as:

$$\min_x(\text{ or } \max_x) \quad c_1x_1 + c_2x_2 + \ldots + c_nx_n \tag{E.1}$$

subject to

$$\begin{gathered} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \simeq b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \simeq b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \simeq b_m \\ l_1 \leqslant x_1 \leqslant u_1, \quad l_2 \leqslant x_2 \leqslant u_2, \ldots, \quad l_n \leqslant x_n \leqslant u_n, \end{gathered}$$

in which the relation $\simeq$ represents greater than or equal to, less than or equal to, or simply equal to. The lower bounds l_i and upper bounds u_i $(i = 1, \ldots, n)$ may be positive infinity, negative infinity, or any real number. CPLEX allows the user to specify the variables as either continuous or to constrain them to take only integer values. CPLEX implements optimizers that are based on the simplex algorithms, both primal and dual simplex, primal-dual logarithmic barrier algorithms and a sifting algorithm to solve linear programming problems (see table E.1).

After initiating a problem, we solve it by calling one of the optimizers available in the CPLEX Component Libraries. The default setting of the LP method (**LPMethod** = 0) lets CPLEX decide which algorithm to use for solving the linear programming problems. It is recommended by the CPLEX Manual (see ILOG Inc. [100]) that users choose this option unless there is a compelling reason to choose another method for a particular linear programming problem. The automatic setting will choose the dual simplex optimizer unless:

Setting of LPMethod	Meaning
0	Default setting
1	Primal simplex
2	Dual simplex
3	Network simplex
4	Barrier
5	Sifting
6	Concurrent Dual, Barrier, and Primal in opportunistic parallel mode

TABLE E.1: *Settings of the **LPMethod** parameter for choosing an optimizer in CPLEX.*

- There is an advanced basis present that is ascertained to be primal feasible,

- The specified problem is so small that the overhead costs of setting up concurrent optimization exceed possible performance gains, or

- The memory emphasis parameter has been enabled.

We provide an overview of the various optimization methods supported by CPLEX below:

- **Dual simplex optimizer**: In this method CPLEX uses linear programming and the fact that a linear programming problem can be stated in either primal or dual form, and an optimal solution (if one exists) of the dual has a direct relationship to an optimal solution of the primal model. CPLEX reports the solution in terms of the primal model. The dual simplex method is the first choice for optimizing a linear programming problem and it is used especially when we solve primal-degenerate problems (when there is little variability in the righthand side coefficients but significant variability in the cost coefficients.). We have used this method for solving the linear programming problems in chapters 2, 3 and 4.

- **Primal simplex optimizer**: CPLEX primal simplex optimizer can also effectively solve a wide variety of linear programming problems, but it is not the recommended choice for a first try at optimizing a linear programming problem. This method will at times work better on linear programming problems where the number of variables exceeds the number of constraints significantly, or on linear programming problems that exhibit little variability in the cost coefficients.

- **Network simplex optimizer**: The CPLEX network optimizer recognizes a special class of linear programming problems with network structure and is used for solving this class of problems. The optimizer uses highly efficient network algorithms on that part of the problem; this solution is then used to construct an advanced basis for the rest of the linear programming problem. Using this advanced basis, CPLEX then iterates to find a solution to the full problem.

- **Barrier optimizer**: The barrier optimizer is an efficient approach for solving large, sparse linear programming problems when the constraint matrix has a staircase structures or banded structures. The barrier optimizer utilizes a primal-dual logarithmic barrier algorithm to generate a sequence of strictly positive primal and dual solutions for the given problem.

- **Sifting optimizer**: The sifting optimizer exploits the characteristics of models with large aspect ratios (i.e., when we have a large ratio of the number of columns to the number of rows). The method starts by solving a subproblem (known as the working problem) that consists of all rows but only a small subset of the full set of columns, assuming an arbitrary value (such as its lower bound) for the solution value of each of the remaining columns. The reduced costs of the remaining columns are evaluated using this solution. In the next sifting iteration, any columns whose reduced costs violate the optimality criterion become candidates to be added to the working problem. The sifting terminates when no candidates are present and the solution of the working problem is optimal for the full problem.

- **Concurrent optimizer**: On a computer where CPLEX can use parallel threads, the concurrent optimizer launches distinct optimizers. Note that if this optimizer is launched on a single-threaded platform, it only calls the dual simplex optimizer.

E.0.1 Accessing solution status

Once CPLEX optimization is terminated, the solution status codes (available through `cplex.Solution.status` in Matlab) describe the status of the solution. Table E.2 provides a list the first 10 codes and their respective descriptions.

It is a well-known fact of linear programming that if residual vectors are zero then the current solution is primal and dual feasible (Complementary Slackness Theorem). Moreover a primal-dual feasible pair that has zero duality gap is an optimal solution (Strong Duality Theorem). Furthermore, a wide range of optional fields is provided by CPLEX to assess the quality of the solution. These dynamic properties of the solution are generated and updated by the `cplex.solve`. We have used the following three properties to assess the quality of the solution:

1. `quality.objgap`: Is used to access the objective value gap between the primal and dual objective value solution.

Solution status code	Description
1	Optimal solution is available
2	Problem has an unbounded ray
3	Problem has been proven infeasible
4	Problem has been proven either infeasible or unbounded
5	Optimal solution is available, but with infeasibilities after unscaling
6	Solution is available, but not proved optimal, due to numeric difficulties during optimization
10	Stopped due to limit on number of iterations
11	Stopped due to a time limit
12	Stopped due to an objective limit

TABLE E.2: *List of solution status codes in CPLEX.*

2. `quality.sumprimalresidual` Is used to access the sum of the elements of the primal residual vector.

3. `quality.sumdualresidual` Is used to access the sum of the values of the dual residual vector.

All results were produced on a platform that uses an AMD 3.2 GHz Phenom II X4 955 processor with 16 GB RAM.

E.0.2 Run Time and Solution Quality in Solving Linear Programming Problem 2.6.22

λ_B	λ_C	T	M	N_1+1	N_2+1	run time (in seconds)	sum of primal residuals	sum of dual residuals	duality gap	solution status code
0.5	0.5	5	10000	61	61	744.4787	2.4169e-07	2.1752e-07	3.5316e-09	1
	1	5	10000	61	61	699.0864	4.0391e-07	3.6352e-07	8.2119e-09	1
	1.5	5	10000	61	61	698.9253	9.6455e-08	8.6809e-08	1.5403e-10	1
	2	5	10000	61	61	616.2281	1.3197e-07	1.1878e-07	4.3024e-10	1
	2.5	5	10000	61	61	559.9946	9.4205e-07	8.4785e-07	1.6899e-09	1
	3	5	10000	61	61	686.9247	9.5613e-07	8.6052e-07	6.4912e-09	1
	3.5	5	10000	61	61	638.8797	5.7521e-07	5.1769e-07	7.3172e-09	1
	4	5	10000	61	61	571.9748	5.978e-08	5.3802e-08	6.4775e-09	1
	4.5	5	10000	61	61	611.0261	2.3478e-07	2.113e-07	4.5092e-09	1
1	0.5	5	10000	61	61	658.1126	8.1763e-07	7.3586e-07	5.5016e-09	1
	1	5	10000	61	61	742.9386	7.9483e-07	7.1535e-07	6.2248e-09	1
	1.5	5	10000	61	61	727.5713	6.4432e-07	5.7989e-07	5.8704e-09	1
	2	5	10000	61	61	601.3208	3.7861e-07	3.4075e-07	2.0774e-09	1
	2.5	5	10000	61	61	606.4141	8.1158e-07	7.3042e-07	3.0125e-09	1
	3	5	10000	61	61	694.6784	5.3283e-07	4.7954e-07	4.7092e-09	1

	3.5	5	10000	61	61	619.3651	3.5073e-07	3.1565e-07	2.3049e-09	1
	4	5	10000	61	61	599.1491	9.39e-07	8.451e-07	8.4431e-09	1
	4.5	5	10000	61	61	598.9228	8.7594e-07	7.8835e-07	1.9476e-09	1
1.5	0.5	5	10000	61	61	747.9748	2.2175e-07	1.9957e-07	2.922e-10	1
	1	5	10000	61	61	693.887	1.1742e-07	1.0568e-07	9.2885e-09	1
	1.5	5	10000	61	61	638.8881	2.9668e-07	2.6701e-07	7.3033e-09	1
	2	5	10000	61	61	624.1935	3.1878e-07	2.869e-07	4.8861e-09	1
	2.5	5	10000	61	61	609.128	4.2417e-07	3.8175e-07	5.7853e-09	1
	3	5	10000	61	61	590.5104	5.0786e-07	4.5707e-07	2.3728e-09	1
	3.5	5	10000	61	61	676.2212	8.5516e-08	7.6964e-08	4.5885e-09	1
	4	5	10000	61	61	589.7157	2.6248e-07	2.3623e-07	9.6309e-09	1
	4.5	5	10000	61	61	721.1216	8.0101e-07	7.2091e-07	5.4681e-09	1
2	0.5	5	10000	61	61	564.4477	3.8462e-07	3.4616e-07	3.4388e-09	1
	1	5	10000	61	61	585.5235	5.8299e-07	5.2469e-07	5.8407e-09	1
	1.5	5	10000	61	61	612.3728	2.5181e-07	2.2663e-07	1.0777e-09	1
	2	5	10000	61	61	669.0924	2.9044e-07	2.614e-07	9.0631e-09	1
	2.5	5	10000	61	61	607.1747	6.1709e-07	5.5538e-07	8.7965e-09	1
	3	5	10000	61	61	698.2022	2.6528e-07	2.3875e-07	8.1776e-09	1
	3.5	5	10000	61	61	662.0612	8.2438e-07	7.4194e-07	2.6073e-09	1
	4	5	10000	61	61	591.0493	9.8266e-07	8.844e-07	5.9436e-09	1
	4.5	5	10000	61	61	672.6188	7.3025e-07	6.5722e-07	2.2513e-10	1
2.5	0.5	5	10000	61	61	672.404	9.1599e-07	8.2439e-07	3.5763e-10	1
	1	5	10000	61	61	583.2167	1.1511e-09	1.036e-09	1.7587e-09	1
	1.5	5	10000	61	61	565.5608	4.6245e-07	4.162e-07	7.2176e-09	1
	2	5	10000	61	61	615.5538	4.2435e-07	3.8191e-07	4.7349e-09	1
	2.5	5	10000	61	61	571.948	4.6092e-07	4.1482e-07	1.5272e-09	1
	3	5	10000	61	61	717.5332	7.7016e-07	6.9314e-07	3.4112e-09	1
	3.5	5	10000	61	61	650.6715	3.2247e-07	2.9022e-07	6.0739e-09	1
	4	5	10000	61	61	589.783	7.8474e-07	7.0627e-07	1.9175e-09	1
	4.5	5	10000	61	61	688.6771	4.7136e-07	4.2422e-07	7.3843e-09	1
3	0.5	5	10000	61	61	680.1306	3.4788e-07	3.1309e-07	5.3998e-09	1
	1	5	10000	61	61	620.4466	4.4603e-07	4.0142e-07	7.0692e-09	1
	1.5	5	10000	61	61	616.7064	5.4239e-08	4.8816e-08	9.9949e-09	1
	2	5	10000	61	61	696.7068	1.7711e-07	1.594e-07	2.8785e-09	1
	2.5	5	10000	61	61	714.8198	6.6281e-07	5.9653e-07	4.1452e-09	1
	3	5	10000	61	61	647.4772	3.3083e-07	2.9775e-07	4.6484e-09	1
	3.5	5	10000	61	61	734.2207	8.9849e-07	8.0864e-07	7.6396e-09	1
	4	5	10000	61	61	594.0967	1.1816e-07	1.0634e-07	8.182e-09	1
	4.5	5	10000	61	61	564.59	9.8842e-07	8.8958e-07	1.0022e-09	1
3.5	0.5	5	10000	61	61	690.6955	6.7865e-07	6.1079e-07	6.9667e-09	1
	1	5	10000	61	61	724.8706	4.9518e-07	4.4566e-07	5.8279e-09	1
	1.5	5	10000	61	61	567.4416	1.8971e-07	1.7074e-07	8.154e-09	1
	2	5	10000	61	61	571.0037	4.9501e-07	4.4551e-07	8.7901e-09	1
	2.5	5	10000	61	61	618.1476	1.4761e-07	1.3285e-07	9.8891e-09	1
	3	5	10000	61	61	596.5936	5.4974e-08	4.9477e-08	5.2238e-12	1
	3.5	5	10000	61	61	641.0049	8.5071e-07	7.6564e-07	8.6544e-09	1
	4	5	10000	61	61	661.1706	5.6056e-07	5.045e-07	6.1257e-09	1
	4.5	5	10000	61	61	663.6293	9.2961e-07	8.3665e-07	9.8995e-09	1
4	0.5	5	10000	61	61	708.5987	7.6903e-07	6.9213e-07	2.0941e-09	1
	1	5	10000	61	61	674.6735	5.8145e-07	5.233e-07	5.5229e-09	1
	1.5	5	10000	61	61	583.3584	9.2831e-07	8.3548e-07	6.2988e-09	1
	2	5	10000	61	61	641.6517	5.8009e-07	5.2208e-07	3.1991e-10	1
	2.5	5	10000	61	61	587.404	1.6983e-08	1.5285e-08	6.1471e-09	1
	3	5	10000	61	61	583.9055	1.2086e-07	1.0877e-07	3.6241e-09	1
	3.5	5	10000	61	61	577.0248	8.6271e-07	7.7644e-07	4.9533e-10	1
	4	5	10000	61	61	739.0752	4.843e-07	4.3587e-07	4.8957e-09	1
	4.5	5	10000	61	61	748.2303	8.4486e-07	7.6037e-07	1.9251e-09	1
4.5	0.5	5	10000	61	61	600.0884	5.6498e-07	5.0848e-07	6.2096e-09	1
	1	5	10000	61	61	596.4199	6.4031e-07	5.7628e-07	5.7371e-09	1
	1.5	5	10000	61	61	605.4817	4.1703e-07	3.7533e-07	5.2078e-10	1
	2	5	10000	61	61	662.3932	2.0598e-07	1.8538e-07	9.312e-09	1
	2.5	5	10000	61	61	699.0357	9.4793e-07	8.5314e-07	7.2866e-09	1
	3	5	10000	61	61	564.7002	8.2071e-08	7.3864e-08	7.3784e-09	1
	3.5	5	10000	61	61	737.3927	1.0571e-07	9.5138e-08	6.3405e-10	1

	4	5	10000	61	61	677.0294	1.4204e-07	1.2784e-07	8.6044e-09	1
	4.5	5	10000	61	61	629.1539	1.6646e-07	1.4981e-07	9.3441e-09	1

TABLE E.3: *Run time, sum of primal residuals, sum of dual residuals, duality gap and solution status code for solving linear programming problems 2.6.22 and generating figure 2.6 with the bank and the counterparties default time parameters given in 2.6.27.*

λ_B	λ_C	T	M	N_1+1	N_2+1	run time (in seconds)	sum of primal residuals	sum of dual residuals	duality gap	solution status code
1	1	10	10000	121	121	963.2083	7.4003e-07	6.6603e-07	5.9794e-09	1
	2	10	10000	121	121	877.7275	2.3483e-07	2.1134e-07	7.8936e-09	1
	3	10	10000	121	121	961.0353	7.3496e-07	6.6146e-07	3.6765e-09	1
	4	10	10000	121	121	938.2507	9.706e-07	8.7354e-07	2.0603e-09	1
	5	10	10000	121	121	885.9282	8.6693e-07	7.8024e-07	8.6667e-10	1
	6	10	10000	121	121	917.1186	8.6235e-08	7.7611e-08	7.7193e-09	1
	7	10	10000	121	121	1015.3812	3.6644e-07	3.2979e-07	2.0567e-09	1
	8	10	10000	121	121	875.0869	3.692e-07	3.3228e-07	3.8827e-09	1
	9	10	10000	121	121	1008.3186	6.8503e-07	6.1653e-07	5.5178e-09	1
2	1	10	10000	121	121	959.1499	6.9475e-07	6.2528e-07	4.876e-09	1
	2	10	10000	121	121	990.5315	7.581e-07	6.8229e-07	7.6896e-09	1
	3	10	10000	121	121	939.5154	4.3264e-07	3.8938e-07	3.9601e-09	1
	4	10	10000	121	121	961.2284	6.555e-07	5.8995e-07	2.7294e-09	1
	5	10	10000	121	121	871.5572	1.0976e-07	9.878e-08	3.7235e-10	1
	6	10	10000	121	121	920.843	9.3376e-07	8.4038e-07	6.7329e-09	1
	7	10	10000	121	121	889.6467	1.8746e-07	1.6871e-07	4.2956e-09	1
	8	10	10000	121	121	1046.4423	2.6618e-07	2.3956e-07	4.5174e-09	1
	9	10	10000	121	121	906.7515	7.9783e-07	7.1805e-07	6.0986e-09	1
3	1	10	10000	121	121	1015.5573	1.6825e-07	1.5143e-07	1.8443e-09	1
	2	10	10000	121	121	877.7077	1.9625e-07	1.7662e-07	2.1203e-09	1
	3	10	10000	121	121	896.8791	3.1748e-07	2.8573e-07	7.7347e-10	1
	4	10	10000	121	121	939.1182	3.1643e-07	2.8479e-07	9.138e-09	1
	5	10	10000	121	121	1013.1766	2.1756e-07	1.9581e-07	7.0672e-09	1
	6	10	10000	121	121	962.65	2.5104e-07	2.2594e-07	5.5779e-09	1
	7	10	10000	121	121	936.5697	8.9292e-07	8.0363e-07	3.1343e-09	1
	8	10	10000	121	121	959.8594	7.0322e-07	6.329e-07	1.662e-09	1
	9	10	10000	121	121	935.7973	5.5574e-07	5.0016e-07	6.225e-09	1
4	1	10	10000	121	121	887.9328	5.3063e-07	4.7757e-07	7.4255e-09	1
	2	10	10000	121	121	965.437	8.3242e-07	7.4918e-07	4.2433e-09	1
	3	10	10000	121	121	911.8655	5.9749e-07	5.3774e-07	4.2936e-09	1
	4	10	10000	121	121	966.1134	3.3531e-07	3.0178e-07	1.2487e-09	1
	5	10	10000	121	121	874.1888	2.9923e-07	2.693e-07	2.4434e-10	1
	6	10	10000	121	121	862.8889	4.5259e-07	4.0733e-07	2.9019e-09	1
	7	10	10000	121	121	985.0777	4.2265e-07	3.8038e-07	3.1752e-09	1
	8	10	10000	121	121	1018.5536	3.5961e-07	3.2365e-07	6.5369e-09	1
	9	10	10000	121	121	920.5851	5.5832e-07	5.0249e-07	9.5694e-09	1
5	1	10	10000	121	121	927.1331	3.1743e-07	2.8569e-07	8.6675e-09	1
	2	10	10000	121	121	956.4187	8.1454e-07	7.3309e-07	6.3119e-09	1
	3	10	10000	121	121	873.267	7.8907e-07	7.1017e-07	3.5507e-09	1
	4	10	10000	121	121	1017.1202	8.5226e-07	7.6704e-07	9.97e-09	1
	5	10	10000	121	121	878.4787	5.0564e-07	4.5507e-07	2.2417e-09	1
	6	10	10000	121	121	885.7939	6.3566e-07	5.721e-07	6.5245e-09	1
	7	10	10000	121	121	908.0952	9.5089e-07	8.558e-07	6.0499e-09	1
	8	10	10000	121	121	989.0762	4.4396e-07	3.9957e-07	3.8725e-09	1
	9	10	10000	121	121	1031.614	6.0019e-08	5.4017e-08	1.4219e-09	1
6	1	10	10000	121	121	948.1822	1.8214e-07	1.6393e-07	6.7664e-09	1
	2	10	10000	121	121	1018.3839	4.182e-08	3.7638e-08	9.883e-09	1
	3	10	10000	121	121	871.6264	1.0694e-07	9.6247e-08	7.6683e-09	1
	4	10	10000	121	121	1003.4138	6.1644e-07	5.548e-07	3.367e-09	1
	5	10	10000	121	121	1038.5359	9.3966e-07	8.4569e-07	6.6238e-09	1
	6	10	10000	121	121	860.0995	3.5446e-07	3.1901e-07	2.4417e-09	1
	7	10	10000	121	121	887.4062	4.1063e-07	3.6957e-07	2.9551e-09	1
	8	10	10000	121	121	963.7869	9.8435e-07	8.8591e-07	6.8018e-09	1

	9	10	10000	121	121	928.2198	9.4558e-07	8.5102e-07	5.2785e-09	1
7	1	10	10000	121	121	860.4108	5.1015e-07	4.5914e-07	8.116e-09	1
	2	10	10000	121	121	940.091	9.0636e-07	8.1573e-07	4.8565e-09	1
	3	10	10000	121	121	945.5834	6.2892e-07	5.6603e-07	8.9445e-09	1
	4	10	10000	121	121	1005.7295	1.0153e-07	9.138e-08	1.3755e-09	1
	5	10	10000	121	121	872.7505	3.9085e-07	3.5177e-07	3.9e-09	1
	6	10	10000	121	121	981.194	5.4617e-08	4.9155e-08	9.2736e-09	1
	7	10	10000	121	121	932.1018	5.0128e-07	4.5115e-07	9.1749e-09	1
	8	10	10000	121	121	983.8956	4.3172e-07	3.8855e-07	7.1357e-09	1
	9	10	10000	121	121	971.0146	9.9756e-07	8.978e-07	6.1834e-09	1
8	1	10	10000	121	121	891.1791	4.2298e-08	3.8068e-08	1.9257e-10	1
	2	10	10000	121	121	913.3843	9.7296e-07	8.7566e-07	8.3874e-10	1
	3	10	10000	121	121	1030.676	1.8921e-07	1.7029e-07	9.748e-09	1
	4	10	10000	121	121	899.6219	6.6712e-07	6.0041e-07	6.5135e-09	1
	5	10	10000	121	121	901.0406	5.8644e-07	5.278e-07	2.3124e-09	1
	6	10	10000	121	121	862.2951	6.7511e-07	6.076e-07	4.0349e-09	1
	7	10	10000	121	121	985.3142	3.6102e-07	3.2492e-07	1.2202e-09	1
	8	10	10000	121	121	994.9444	6.2028e-07	5.5825e-07	2.6844e-09	1
	9	10	10000	121	121	1046.353	8.1115e-07	7.3004e-07	2.5785e-09	1
9	1	10	10000	121	121	963.3504	5.9753e-07	5.3777e-07	5.212e-09	1
	2	10	10000	121	121	947.8444	8.8402e-07	7.9562e-07	3.7231e-09	1
	3	10	10000	121	121	1005.9841	9.4373e-07	8.4936e-07	9.3713e-09	1
	4	10	10000	121	121	919.9181	5.4916e-07	4.9424e-07	8.2953e-09	1
	5	10	10000	121	121	856.059	7.2839e-07	6.5555e-07	8.4909e-09	1
	6	10	10000	121	121	1048.0904	5.7676e-07	5.1908e-07	3.7253e-09	1
	7	10	10000	121	121	921.338	2.5857e-08	2.3272e-08	5.9318e-09	1
	8	10	10000	121	121	1030.082	4.4653e-07	4.0188e-07	8.7255e-09	1
	9	10	10000	121	121	860.3889	6.463e-07	5.8167e-07	9.335e-09	1

TABLE E.4: *Run time, sum of primal residuals, sum of dual residuals, duality gap and solution status code for solving linear programming problems 2.6.22 and generating figure 2.7 with the bank and the counterparties default time parameters given in 2.6.28.*

Glossary

CVA^B Bilateral CVA. 10

CVA^U Unilateral CVA. 10

CWI creditworthiness index. 14, 44

EAD Exposure at Default. 44, 45

EPE Expected Positive Exposure. 45

LGD Loss Given Default. 45

MTY Maturity. 45

PD Probability of Default. 44, 45

PFE Potential Future Exposure. 45

$\overline{\rho}$ Market-credit correlation in the ordered sceenario copula methodology. 16

τ_{C} Counterparty's default time. 10

τ_B Bank's default time. 11

EE^{B+} Discounted expected exposure for the counterparty C when computing CVA^B. 11

EE^{B-} Discounted expected exposure for the bank B when computing CVA^B. 11

EE^{U+} Discounted expected exposure for the counterparty C when computing CVA^U. 10

r_0 Risk-free rate. 10, 20

ARA Adaptive Rearrangement Algorithm. 1, 4, 67, 68

CCR Counterparty Credit Risk. 1

CVA Credit Valuation Adjustment. 1

CVaR Conditional Value-at-Risk. 1, 44

E(t) Counterparty-level exposure at time t. 9

EARA Enhanced Adaptive Rearrangement Algorithm. 4, 67, 68, 120

OSC Ordered Scenario Copula. 2

QRM Quantitative Risk Management. 1

RA Rearrangement Algorithm. 1, 3, 67

VaR Value-at-Risk. 1