Optimization Problems in the Simulation of Multifactor Portfolio Credit Risk

Wanmo Kang[1] and Kyungsik Lee[2,⋆,⋆⋆]

[1] Columbia University, New York, NY, USA
[2] Hankuk University of Foreign Studies, Yongin, Korea

Abstract. We consider some optimization problems arising in an efficient simulation method for the measurement of the tail of portfolio credit risk. When we apply an importance sampling (IS) technique, it is necessary to characterize the important regions. In this paper, we consider the computation of directions for the IS, which becomes hard in multifactor case. We show this problem is NP-hard. To overcome this difficulty, we transform the original problem to subset sum and quadratic optimization problems. We support numerically that these reformulation is computationally tractable.

1 Introduction

Measurement of portfolio credit risk is an important problem in financial industry. To reserve economic capital or to summarize the potential risk of a company, the portfolio credit risk is calculated frequently. Some of key properties of this measurement are the importance of dependence structure of obligors constituting the portfolio and the rare-event characteristic of large losses. Dependence among obligors incurs large losses more frequently, even though they are still rare. Gaussian copula is one of the most popular correlation structure in practice. (See [5].) Since there is no known analytical or numerical way to compute the tail losses under Gaussian copula framework, Monte Carlo method is a viable way to accomplish this task. (See [1], [4], [6], [8], and [9]) However, the rareness of large losses makes a crude Monte Carlo method impractical. To accelerate the simulation, one effective way is the application of IS. When applying IS, the identification of important region is the key for the efficiency enhancement. In this paper, we consider the problem of identifying important regions. The combinatorial complexity underlying this problem makes it a hard problem. We re-formulate this problem as a combination of quadratic optimizations and subset sum problems. The worst case complexity is not reduced, but the subset sum problems can be solved very fast in practice. Consequently this new approach works very well for actual problem instances.

⋆ Corresponding author.
⋆⋆ This work was supported by Hankuk University of Foreign Studies Research Fund.

M. Gavrilova et al. (Eds.): ICCSA 2006, LNCS 3982, pp. 777–784, 2006.

2 Portfolio Credit Risk and Importance Sampling

We briefly introduce the portfolio credit risk model and Gaussian copula framework. We consider the distribution of losses from defaults over a fixed horizon. We are interested in the estimation of the probability that the credit loss of a portfolio exceeds a given threshold. As it is difficult to estimate correlations among the default events of obligors, latent variables are introduced as default triggers and the dependence structure is imposed on the latent variables indirectly. A linear factor model is adopted for the correlations among them. We use the following notation:

m : the number of obligors to which the portfolio is exposed;
Y_k : default indicator (= 1 for default, = 0 otherwise) for the k-th obligor;
p_k : marginal probability that the k-th obligor defaults;
c_k : loss resulting from default of the k-th obligor;
$L_m = c_1 Y_1 + \cdots + c_m Y_m$: total loss from defaults.

We are interested in the estimation of $\mathsf{P}(L_m > x)$ for a given threshold x when the event $\{L_m > x\}$ is rare. We call such one as a *large loss* event. We introduce latent normal random variables X_k for each Y_k. X_k's are standard normal random variables and We set $Y_k = \mathbf{1}\{X_k > \Phi^{-1}(1 - p_k)\}$, with Φ the cumulative normal distribution. For a linear factor representation of X_k, we assume the following: There are d factors and t types of obligors. $\{\mathcal{I}_1, \ldots, \mathcal{I}_t\}$ is a partition of the set of obligors $\{1, \ldots, m\}$ into types. If $k \in \mathcal{I}_j$, then the k-th obligor is of type j and its latent variable is given by

$$X_k = \mathbf{a}_j^\top \mathbf{Z} + b_j \, \varepsilon_k$$

where $\mathbf{a}_j \in \mathbb{R}^d$ with $0 < \|\mathbf{a}_j\| < 1$, $\mathbf{Z}$ is a d dimensional standard normal random vector, $b_j = \sqrt{1 - \mathbf{a}_j^\top \mathbf{a}_j}$ and ε_k are independent standard normal random variables. This dependence structure is called a *Gaussian copula model*. (See [2].) $\mathbf{Z}$ represents common systematic risk factors and ε_k an idiosyncratic risk factor. We set $x = q \sum_{k=1}^m c_k$ for a given q, $0 < q < 1$. Denote the average loss of each type by $C_j = \sum_{k \in \mathcal{I}_j} c_k / |\mathcal{I}_j|$ and total average loss by $C = \sum_{k=1}^m c_k / m$. Then index sets (sets of types) important for the large losses exceeding qC can be characterized by the following index set $\mathcal{J} \subset \{1, \ldots, t\}$ (See [3]):

$$\max_{\mathcal{J}' \subsetneq \mathcal{J}} \sum_{j \in \mathcal{J}'} C_j < qC \leq \sum_{j \in \mathcal{J}} C_j. \tag{1}$$

This characterization of an important index set can be interpreted as follows: to observe samples with large losses, the common factors should have values which enable the sum of average losses of the types in the index set to exceed the loss threshold.

After identifying these index sets (say $\mathcal{J}$, the point to shift the mean vectors of Gaussian common factors is found by

$$\boldsymbol{\mu}_{\mathcal{J}} := \operatorname{argmin} \{\|\mathbf{z}\| \ : \ \mathbf{z} \in G_{\mathcal{J}}\} \tag{2}$$

where

$$G_j := \left\{ \mathbf{z} \in \mathbb{R}^d : \mathbf{a}_j^\top \mathbf{z} \geq d_j \right\} \quad \text{and} \quad G_{\mathcal{J}} := \bigcap_{j \in \mathcal{J}} G_j \,.$$

$d_j > 0$ is a constant for each type calculated from the problem instance. In this paper, the positivity of d_j is sufficient.

Now returning to the Monte Carlo simulation, we sample the common factors from the mixture distribution of $N(\boldsymbol{\mu}_{\mathcal{J}}, \mathbf{I})$ for all the $\mathcal{J}$'s satisfying (1). $\boldsymbol{\mu}_{\mathcal{J}}$ is the minimum distant point to the important region for the large losses and we sample from the normal distribution shifted to those points. As usual, we compensate this change of measure by multiplying likelihood ratios. (See [3] for details.)

Define $\mathcal{S}_q$ be the set of index sets satisfying (1). In principle, we can use $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ in the simulation as the mean vectors of mixture distribution if $K = |\mathcal{S}_q|$. However, K becomes very large as t increases. The size depends on q, but for q values near 0.5, the order of K follows exponential to t. So identifying $\mathcal{S}_q$ first and then finding corresponding $\mu_{\mathcal{J}}$'s are impractical. In the next section, we exploit some structural properties and re-formulate the problem into a tractable one.

3 Re-formulation of Problem

The first idea comes from the fact that we use $\boldsymbol{\mu}_{\mathcal{J}}$ for the shift of mean vectors but $\mathcal{J}$ is not explicitly used. So if $\boldsymbol{\mu}_{\mathcal{J}} = \boldsymbol{\mu}_{\mathcal{J}'}$ for two different index sets $\mathcal{J}$ and $\mathcal{J}'$, we don't need to know what the two index sets are, but just need $\boldsymbol{\mu}_{\mathcal{J}}$. Hence we focus on the characterization of

$$\mathcal{V} := \{ \boldsymbol{\mu}_{\mathcal{J}} : \mathcal{J} \subset \{1, \ldots, t\}, |\mathcal{J}| \leq d, G_{\mathcal{J}} \neq \emptyset \}$$

instead of $\mathcal{S}_q$ which possibly consists of exponentially many elements with respect to the number of types t. The issue here is how to find the candidate IS distributions as fast as possible when the number of types, t, and the dimension of factors, d, are fixed.

3.1 Reduction of Candidate Mean Vectors

For a given problem instance, the size of $\mathcal{S}_q$ depends on the value q. In the worst case, the size of $\mathcal{S}_q$ will be $\binom{t}{[t/2]}$, in which case the application of IS is intractable for instances with a large number of types. To avoid this difficulty, we need to devise a method that does not involve an explicit enumeration of the index sets in $\mathcal{S}_q$. The key fact is the following lemma.

Lemma 1. *For any $\mathcal{J} \in \mathcal{S}_q$ satisfying $G_{\mathcal{J}} \neq \emptyset$, there exists a $\mathcal{J}' \subset \mathcal{J}$ with $|\mathcal{J}'| \leq d$ such that*

$$\boldsymbol{\mu}_{\mathcal{J}} = \boldsymbol{\mu}_{\mathcal{J}'} \,.$$

Proof. (Sketch of Proof) Recall that the definition (2) implies that $\boldsymbol{\mu}_{\mathcal{J}}$ is the optimal solution of linear programming (LP), $\min\{\boldsymbol{\mu}_{\mathcal{J}}^{\top}\mathbf{z} : \mathbf{a}_j^{\top}\mathbf{z} \geq d_j \quad \text{for } j \in \mathcal{J}\}$. The LP duality gives a dual optimal solution π with $\mathcal{P} := \{j : \pi_j > 0\}$ and $|\mathcal{P}| \leq d$. The complementary slackness condition shows that $\boldsymbol{\mu}_{\mathcal{J}}$ and $\frac{\pi_j^*}{\|\mathbf{v}\|}$, $j \in \mathcal{P}$ satisfy KKT optimality conditions for $\min\{\|\mathbf{z}\| : \mathbf{a}_j^{\top}\mathbf{z} \geq d_j \quad \text{for } j \in \mathcal{P}\}$ and its dual. We can take $\mathcal{J}' = \mathcal{P}$ and this completes the proof. Refer [3] for details. □

The lemma tells us that we don't have to spend our effort to solve (2) if $|\mathcal{J}| > d$. This gives a large reduction of our search space. From this lemma, we also have the following upper bound on the number of (2) which we have to solve.

Lemma 2. *For an instance with d factors and t types,*

$$\left|\{\boldsymbol{\mu}_{\mathcal{J}} : \mathcal{J} \in \mathcal{S}_q\}\right| \leq \binom{t}{d} + \binom{t}{d-1} + \cdots + t < t^d.$$

Proof. Note that the righthand side of inequality is the number of ways of choosing d or less constraints from t candidates. Combining with Lemma 1, we complete the proof. □

3.2 Derivation of the Subset Sum Problem

Recall that $\{\boldsymbol{\mu}_{\mathcal{J}} : \mathcal{J} \in \mathcal{S}_q\} \subset \mathcal{V}$ from Lemma 1. The upper bound in Lemma 2 is also an upper bound on $|\mathcal{V}|$. Our approach is to find $\mathcal{V}$, as reduced candidate mean vectors, and use it to get $\{\boldsymbol{\mu}_{\mathcal{J}} : \mathcal{J} \in \mathcal{S}_q\}$ from $\mathcal{V}$. Assume a representation $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ and define $\mathcal{H}(\mathbf{v}) := \{j : \mathbf{a}_j^{\top}\mathbf{v} \geq d_j, j = 1, \ldots, t\}$ for $\mathbf{v} \in \mathcal{V}$. $\mathcal{H}(\mathbf{v})$ is the maximal index set satisfying $\mathbf{v} = \boldsymbol{\mu}_{\mathcal{H}(\mathbf{v})}$. Consider, for each $\mathbf{v} \in \mathcal{V}$, all the minimal constraints sets forming the optimization problem whose unique optimal solution is $\mathbf{v}$; denote this family by $\mathcal{F}(\mathbf{v}) = \{F : F \subset \mathcal{H}(\mathbf{v}), \mathbf{v} = \boldsymbol{\mu}_F, \mathbf{v} \neq \boldsymbol{\mu}_{F\setminus\{j\}} \quad \text{for all } j \in F\}$. Note that $|F| \leq d$ for each $F \in \mathcal{F}(\mathbf{v})$ by Lemma 1 and hence the cardinality of $\bigcup_{\mathbf{v}\in\mathcal{V}} \mathcal{F}(\mathbf{v})$ has the same upper bound as the one in Lemma 2. Because we search $\mathcal{V}$ by probing all index sets of cardinality less than or equal to d, we get $\mathcal{F}(\mathbf{v})$'s as by-products of the search. To simplify notations, we abuse the symbol $\mathcal{V}$ to denote the collection of pairs $(\mathbf{v}, F)$ for each $\mathbf{v}$ and each $F \in \mathcal{F}(\mathbf{v})$.

To identify $\{\boldsymbol{\mu}_{\mathcal{J}} : \mathcal{J} \in \mathcal{S}_q\}$ from $\mathcal{V}$ according to our scheme, we have to decide whether there is a $\mathcal{J} \in \mathcal{S}_q$ such that $\mathbf{v} = \boldsymbol{\mu}_{\mathcal{J}}$ for each $\mathbf{v} \in \mathcal{V}$. For this decision, we can use some information on $\mathbf{v}$, $\mathcal{H}(\mathbf{v})$ and $\mathcal{F}(\mathbf{v})$, which can be collected from the computation of $\mathcal{V}$ with no additional cost. We formulate this problem as a *minimal cover* problem (MCP). Then we transform MCP into a knapsack problem. To simplify notations, we define $C_J := \sum_{j\in J} C_j$ for any index set J.

MCP: An index set N is given. $\{C_i\}_{i\in N}$ with $C_i > 0$ and a subset $F \subset N$ $(F \neq \emptyset)$ are given. For a given positive number b, is there a subset $J \subset N \setminus F$ such that

$$C_{J\cup F} \geq b \qquad \text{and} \qquad C_{J\cup F\setminus\{k\}} < b \text{ for all } k \in J \cup F \ ?$$

Then we have the following lemma:

Lemma 3. *The answer to MCP is YES if and only if there exists a* $J \subset N \setminus F$ *such that*

$$\text{i)}\, C_{J\cup F} \geq b,\ \text{ii)}\, C_{J\cup F\setminus\{k\}} < b \text{ for all } k \in J, \text{ and iii)}\, C_{J\cup F} - \min_{i\in F} C_i < b.$$

Proof. If we notice the relation $C_{J\cup F\setminus\{k\}} < b$ for all $k \in F \Leftrightarrow C_{J\cup F} - \min_{i\in F} C_i < b$, then the proof is complete. □

Set $b' := b - C_F$. Using Lemma 3, we can rewrite the MCP as

MCP′: $\{C_i\}_{i\in N}$ with $C_i > 0$ and a subset $F \subset N$ are given. For a given positive number b, is there a subset $J \subset N \setminus F$ such that

$$\text{i)}\, C_J \geq b',\ \text{ii)}\, C_{J\setminus\{k\}} < b' \text{ for all } k \in J, \text{ and iii)}\, C_J < b' + \min_{i\in F} C_i\ ?$$

Consider the following 0-1 knapsack problem (KP):

$$f^* = \min\left\{ \sum_{j\in N\setminus F} C_j x_j : \sum_{j\in N\setminus F} C_j x_j \geq b', x_j \in \{0,1\} \text{ for all } j \in N \setminus F \right\}.$$

Any set $G \subset N \setminus F$ corresponding to an optimal solution of (KP) satisfies condition i) of MCP′ from the feasibility. If $C_{G\setminus\{k\}} \geq b'$ for some $k \in G$, then $G\setminus\{k\}$ is another feasible set with strictly less optimal value and this contradicts to the optimality of G. Hence G satisfies ii) of MCP′. Therefore, we conclude that $f^* < b' + \min_{i\in F} C_i$ if and only if the answer to MCP is YES. Now set $N = \mathcal{H}(\mathbf{v})$ and take an F from $\mathcal{F}(\mathbf{v})$. Then by setting $b = qC$, MCP solves whether there is a $\mathcal{J}$ such that $F \subset \mathcal{J} \subset \mathcal{H}(\mathbf{v})$, $\mathcal{J} \in \mathcal{S}_q$, and $\mathbf{v} = \boldsymbol{\mu}_{\mathcal{J}}$. Hence by checking this question for all $F \in \mathcal{F}(\mathbf{v})$, we can decide whether $\mathbf{v} \in \{\boldsymbol{\mu}_{\mathcal{J}} : \mathcal{J} \in \mathcal{S}_q\}$. Note that, with $\min_{i\in F} C_i = 1$, MCP′ is equivalent to knapsack feasibility problem and hence MCP is NP-complete.

By transforming MCP′ into the maximization form using the minimal index set notations results in the following SSP:

$$f^* = \max\left\{ \sum_{j\in N\setminus F} C_j x_j : \sum_{j\in N\setminus F} C_j x_j \leq C_N - qC, x_j \in \{0,1\} \text{ for all } j \in N \setminus F \right\}.$$

The procedure identifying $\{\boldsymbol{\mu}_{\mathcal{J}} : \mathcal{J} \in \mathcal{S}_q\}$ is described as the following:

1: Identify $\mathcal{V}$ by solving the norm minimization problems (2) associated with all possible combinations of type indices, $\mathcal{J} \subset \{1, \ldots, t\}$, $|\mathcal{J}| \leq d$.
2: Given q, solve SSP associated with each $N = \mathcal{H}(\mathbf{v})$ and $F \in \mathcal{F}(\mathbf{v})$ for all $\mathbf{v} \in \mathcal{V}$. If $f^* > C_N - qC - \min_{i\in F} C_i$, then include $\mathbf{v}$ among the shifting mean vectors.

We assume that all C_j's are positive integers. This is a necessary assumption for knapsack problems. SSP has a special structure and is called a *subset sum* problem which is NP-complete. However, knapsack problems arising in practice are solved very fast. (See, e.g., Chapter 4 of Kellerer, Pferschy, and Pisinger [7].) For numerical experiment, we measured the time spent to solve 10^6 subset sum problems using a code `subsum.c` available at `http://www.diku.dk/~pisinger`. Each instance consists of 100 randomly generated weights (i.e. $|N \setminus F| = 100$ in SSP) and the weights have their ranges $[1, 10^4]$ (i.e., $1 \leq C_j \leq 10^4$). 21.88 seconds were spent to solve all these 10^6 problems. (All experiments in this paper were executed using a notebook with a CPU of 1.7GHz Intel Pentium M and a 512MB RAM.) This number of problems, 10^6, is roughly the upper bound of the cardinality of $\mathcal{V}$ for a factor model having 100 types ($= |N|$) and three factors. In solving a subset sum problem, the range of weights are crucial for the running time of algorithm. The above input ranges imply that the potential loss amount of each obligor will take its value among the multiples up to 10^4 of some base amount.

Table 1 shows the average cardinalities of $\{\boldsymbol{\mu}_{\mathcal{J}} : \mathcal{J} \in \mathcal{S}_q\}$ and $\mathcal{V}$ for 30 randomly generated 20- or 25-type instances with factor dimension 4 or 5. Note that the values of the upper bound on $|\mathcal{V}|$ in Lemma 2 are 6195, 21699, 15275, and 68405, respectively. However we just need to keep a smaller size (at most 2000 on average) of $\mathcal{V}$ to get $\{\boldsymbol{\mu}_{\mathcal{J}} : \mathcal{J} \in \mathcal{S}_q\}$. The computing time of $\mathcal{V}$ takes about 28, 100, 65, and 300 seconds for each instance, respectively if we use the MATLAB function `quadprog` for the norm minimization (2). (By a specialized algorithm in Section 3.3, the time can be reduced to 0.3, 1, 1, and 6 seconds for each instance, respectively.) And the total times in solving 9 subset sum problems to find $\{\boldsymbol{\mu}_{\mathcal{J}} : \mathcal{J} \in \mathcal{S}_q\}$'s for $q = 0.1, 0.2, \ldots, 0.9$ from $\mathcal{V}$ are at most 0.2, 0.5, 0.4, and 1.5 seconds, respectively. Furthermore, the cardinalities of $\{\boldsymbol{\mu}_{\mathcal{J}} : \mathcal{J} \in \mathcal{S}_q\}$ are much smaller than the theoretical upper bound. These observations imply that we can implement the IS efficiently.

Table 1. The average number of minimum norm points in $\mathbb{R}^d$. n_q denotes the average of $|\{\boldsymbol{\mu}_{\mathcal{J}} : \mathcal{J} \in \mathcal{S}_q\}|$. (This table is taken from [3]).

Types	d	Bound	$\lvert\mathcal{V}\rvert$	$n_{0.1}$	$n_{0.2}$	$n_{0.3}$	$n_{0.4}$	$n_{0.5}$	$n_{0.6}$	$n_{0.7}$	$n_{0.8}$	$n_{0.9}$
20	4	6195	574.6	16.9	36.1	48.5	52.2	44.5	29.6	14.3	3.9	0.2
20	5	21699	932.2	25.0	57.0	78.8	84.4	69.0	44.2	19.5	4.9	0.4
25	4	15275	1224.9	33.5	65.7	90.5	91.7	74.6	44.1	16.0	2.4	0.2
25	5	68405	2036.5	39.7	96.3	138.4	157.1	137.7	79.8	28.2	3.1	0.0

3.3 Quadratic Optimizations

To find $\mathcal{V}$, we need to solve (2). We can apply general quadratic programming (QP) algorithms to these problems. However, we can exploit the hierarchy of QP problems further: we characterize $\mathcal{V}$ by solving a QP for each $\mathcal{J} \subset \{1, \ldots, t\}$, $|\mathcal{J}| \leq d$. This strategy of the search allows us to do it by

solving $\boldsymbol{\nu}_{\mathcal{J}} = \text{argmin}\{\|\mathbf{z}\| : \mathbf{a}_j^\top \mathbf{z} = d_j \text{ for all } j \in \mathcal{J}\}$ instead of the original QP contrained by inequalities. This equality constrained problem can be solved by simple Gaussian eliminations. Because of the change of constraints, we have $\|\boldsymbol{\nu}_{\mathcal{J}}\| \geq \|\boldsymbol{\mu}_{\mathcal{J}}\|$ instead of equality. So we have to detect the case $\|\boldsymbol{\nu}_{\mathcal{J}}\| > \|\boldsymbol{\mu}_{\mathcal{J}}\|$. For this, we adopt the following procedure:

Set $L = \emptyset$
for $i = 1$ to d
  for all $\mathcal{J} \subset \{1, \ldots, t\}$ of $|\mathcal{J}| = i$
    • find $\boldsymbol{\nu}_{\mathcal{J}}$
    • check the existence of $\mathcal{J}' \in L$ so that $\mathcal{J}' \subset \mathcal{J}$ and $\boldsymbol{\nu}_{\mathcal{J}'} \leq \boldsymbol{\nu}_{\mathcal{J}}$
    • if no such $\mathcal{J}'$ then add $\mathcal{J}$ to L.
  end
end

Note that there always exists a $\mathcal{J}' \subset \mathcal{J}$ such that $\boldsymbol{\nu}_{\mathcal{J}'} = \boldsymbol{\mu}_{\mathcal{J}}$ if $\|\boldsymbol{\nu}_{\mathcal{J}}\| > \|\boldsymbol{\mu}_{\mathcal{J}}\|$. Furthermore, $\boldsymbol{\nu}_{\mathcal{J}'} = \boldsymbol{\mu}_{\mathcal{J}'}$. Since the enumeration is done in increasing order of $|\mathcal{J}|$, $\boldsymbol{\nu}_{\mathcal{J}'}$ exists in the list L (because $|\mathcal{J}'| < |\mathcal{J}|$). Hence the $\mathcal{J}$ is discarded before we solve (2) for $\mathcal{J}$. By this implementation, we can reduce substantial amount of time spent to identify $\mathcal{V}$.

4 Concluding Remarks

We considered an optimization problem arising in the simulation of portfolio credit risk. Our re-formulation has the same worst case computational complexity as the original problem, but it allows tractability in practice. The shifting of sampling distribution based on these points enhances the efficiency of simulation quite impressively.

Acknowledgments

The first author thanks Paul Glasserman and the late Perwez Shahabuddin, the coauthors of [3], on which this proceeding is based.

References

1. A. Avranitis and J. Gregory. *Credit: The Complete Guide to Pricing, Hedging and Risk Management.* Risk Books, London, 2001.
2. P. Glasserman. *Monte Carlo Methods in Financial Engineering.* Springer-Verlag, New York, 2004.
3. P. Glasserman, W. Kang, and P. Shahabuddin. Fast simulation of multifactor portfolio credit risk. Technical report, Graduate School of Business and IEOR Department, Columbia University, February 2005.
4. P. Glasserman and J. Li. Importance sampling for portfolio credit risk. *Management Science*, 2005.

5. G. Gupton, C. Finger, and M. Bhatia. *CreditMetrics Technical Document*. J.P. Morgan & Co., New York, NY, 1997.
6. M. Kalkbrener, H. Lotter, and L. Overbeck. Sensible and efficient capital allocation for credit portfolios. *RISK*, January:S19–S24, 2004.
7. H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer-Verlag, Berlin · Heidelberg, Germany, 2004.
8. S. Merino and M. A. Nyfeler. Applying importance sampling for estimating coherent credit risk contributions. *Quantitative Finance*, 4:199–207, 2004.
9. W. Morokoff. An importance sampling method for portfolios of credit risky assets. In R. Ingalls, M. Rossetti, J. Smith, and B. Peters, editors, *Proceedings of the 2004 Winter Simulation Conference*, pages 1668–1676, 2004.
10. J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag, New York, 1999.