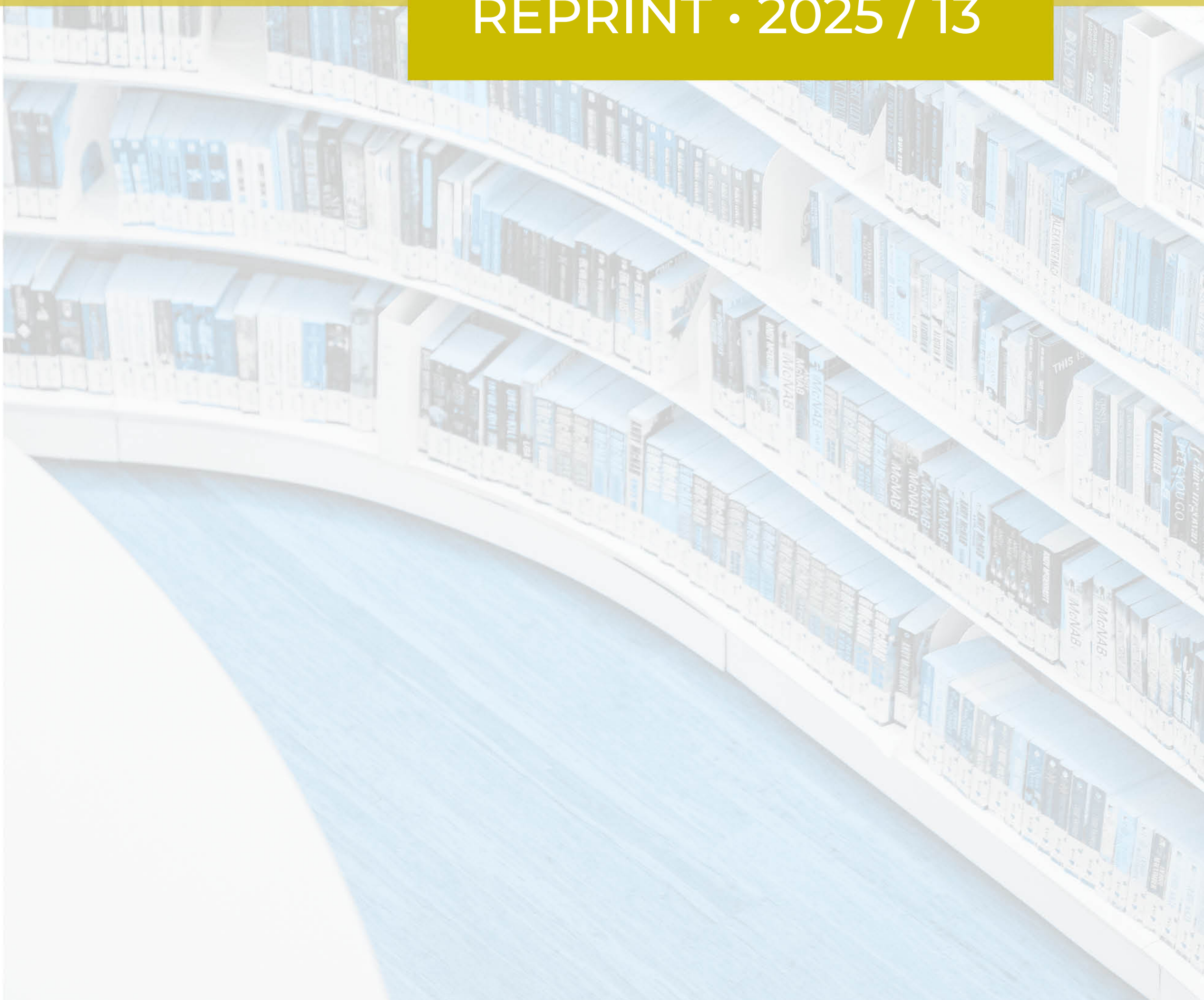
CREDIT SELECTION IN COLLATERALIZED LOAN OBLIGATION: EFFICIENT APPROXIMATION THROUGH LINEARIZATION AND CLUSTERING

Arnaud Germain, Frédéric Vrins

REPRINT • 2025 / 13



UCLouvain

LiDAM
LFIN | Louvain Institute of Data Analysis and Modeling in economics and statistics

LFIN

Voie du Roman Pays 34, L1.03.01 B-1348 Louvain-la-Neuve

Tel (32 10) 47 43 04

Email: lidam-library@uclouvain.be

https://uclouvain.be/en/research-institutes/lidam/lfin/publications

European Journal of Operational Research xxx (xxxx) xxx

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/eor

Interfaces with Other Disciplines

Credit selection in collateralized loan obligation: Efficient approximation through linearization and clustering

Arnaud Germain [a,*], Frédéric Vrins [a,b]

[a] Louvain Institute of Data Analysis and Modeling in economics and statistics (LIDAM/LFIN), UCLouvain, Belgium
[b] Département des sciences de la décision, HEC Montréal, Canada

ARTICLE INFO

Emanuele Borgonovo

Keywords:
Portfolio optimization
OR in banking
Mixed-integer nonlinear programming
Collateralized loan obligation

ABSTRACT

Despite its role in the global financial crisis, collateralized loan obligation (CLO) remains a powerful tool to direct funds towards the real economy. In particular, it enables development banks to increase credit supply to small and medium-sized enterprises (SMEs). Public financial institutions thus face the challenge of identifying a subset of credits to be pooled in a CLO for the sake of reaching a specific financial target. The resulting problem is a mixed-integer nonlinear program, which is NP-hard. In this paper, we propose an approximate optimization problem that combines a large pool approximation, linearization of ancillary variables, and clustering. This approximate optimization problem delivers a solution (i) that improves the value of the objective function under the exact criterion compared to the solution of a derivative-free mixed-integer algorithm solving the exact problem and (ii) in a much lower computation time. As illustration, we consider two realistic CLO objective functions. We rely on the celebrated one-factor Gaussian copula in the main examples, but make clear that this assumption is not a restriction and can be relaxed. Our results contribute to reduce the funding cost of SMEs and are of direct interest for securitization stakeholders such as public financial institutions, commercial banks and pension funds.

1. Introduction

Small and medium-sized enterprises (SMEs) play a key role in the economy by contributing to job creation and global economic development. In the European Union alone, there are 24.7 million SMEs, representing 99.8 % of all non-financial businesses, accounting for 64.4 % of all employment and 51.8 % of the value added in the non-financial business economy (European Commission, 2023). SMEs are typically much riskier than larger firms because of their limited size and financial capacity, which makes them less resilient to economic shocks (OECD, 2023). According to a recent report of the European Commission (2023), "firms across the EU have experienced an increase in bankruptcies since the second half of 2022, which is largely due to regulatory changes (…), the delayed effect of the pandemic, a greater prevalence of late payments and increased difficulties in accessing finance (…)." The latter is identified as one of the main issues: "In 2022, the share of SMEs not applying for bank loans due to fear of possible rejection rose to 6.6 %, up from 5.4 % in 2021 (with small firms being most impacted) - whereas the corresponding share among large firms decreased". Although the portion of SME loans to total business loans has increased to reach 44 % in recent years (OECD, 2023), about half of the SMEs have unmet credit needs (Owens & Wilhelm, 2017). This mismatch between credit supply and demand can be explained by a reduced risk tolerance of private banks but, to some extent, also results from decisions taken by central authorities themselves such as changes in bankruptcy regulation, more stringent capital requirements, or rise of interest rates to fight inflation, to name but a few.

The question of SMEs' access and cost of funds is thus a primary concern for policymakers. The European Commission has launched several financing programs to reduce both interest rates and collateral requirements, and to increase financing volumes. Many of these programs are handled by the European Investment Bank via the European Investment Fund thanks to securitization, and collateralized loan obligation (CLO) in particular. In a nutshell, the idea is that a public financial institution (PFI) such as the European Investment Fund purchases loans by commercial banks and turns them into tradable securities which can then be sold on financial markets. Although securitization is sometimes perceived as a *weapon of wealth destruction* since the global financial crisis, it is also considered by many stakeholders as *the greatest financial innovation of the past century*. Securitization of SME loans can be appealing, with four positive outcomes. First, it allows one to transfer risk to external investors. Commercial banks can use the released capital to issue new loans, thereby increasing the amount of private money injected in the economy. Second, these securities form a new class of financial prod-

* Corresponding author.
E-mail address: arnaud.germain@uclouvain.be (A. Germain).

https://doi.org/10.1016/j.ejor.2025.11.015
Received 15 October 2024; Accepted 10 November 2025
Available online 16 November 2025

ucts, allowing investors such as pension funds to better diversify their portfolios. Third, the PFI can focus on specific types of loans and boost the development of high-priority sectors (e.g., cybersecurity, sustainable activities, renewable energy, etc). Finally, focusing on benefits for SMEs, the securitized assets benefit from a credit enhancement mechanism (via CLO tranches) that will reduce the risk premium required by external investors. This, in turn, leads to a substantial decrease of SMEs' funding costs while putting only a limited amount of PFI's public money at risk. Therefore, securitization provides a unique opportunity for governments willing to support SMEs.

In this context, the problem of optimal loan selection consists in assigning to each candidate loan — usually among a couple of thousands — a selection indicator specifying whether the loan needs to be securitized in a CLO designed to achieve a specific financial goal. Mathematically, this amounts to determining the entries of a binary vector for the sake of optimizing an objective function, leading to a high-dimensional mixed-integer nonlinear program (MINLP) known to be NP-hard. In addition, evaluating the objective function can be very costly, as it often features the joint distribution of the losses, hence, requires time-consuming numerical methods such as Monte Carlo sampling or Laplace transform. The main contribution of this paper is to propose an approximate optimization problem which delivers a solution (i) that is better than the solution of a derivative-free mixed-integer algorithm solving the exact problem and (ii) in a much lower computation time. We consider that the solution of algorithm A is better compared to the solution of algorithm B if the exact objective function evaluated at the solution of A is lower than the exact objective function evaluated at the solution of B, assuming one is minimizing the objective. The considered approximations enable us to circumvent the most challenging aspects of the problem by decreasing its dimension, relaxing the binary constraints and tackling the nonlinearity issues.

The first attempt to address the problem of a credit portfolio construction from a quantitative perspective is due to Bennett (1984). The author argues that the main concepts inherent to Modern Portfolio Theory also apply to loan selection when constructing credit portfolios. Dependence between losses, in particular, is of primary importance. Since then, most of the developments dealt with the modeling and computational aspects of the loss on a pool of loans. For instance, extensive research has been conducted to understand the limit behavior of portfolio losses, leading to the well-known large pool (LP) approximations; see e.g. (Vasicek, 1991) for homogeneous pools and (Gordy, 2003) for the extension to heterogeneous loans, which form the cornerstone of Basel's IRB regulation (BCBS, 2023). Corrections for finite-size, granularity or multi-factor adjustments have been proposed (Düllmann & Masschelein, 2007; Gordy & Lutkebohmert, 2013; Laurent & Sestier, 2016; Pykhtin, 2004). Similarly, efficient computational methods for estimating loss distributions or quantiles of finite pools have been studied, including Laplace transform, saddlepoint or advanced Monte Carlo approximations; see (Egloff & Leippold, 2010; Glasserman & Ruiz-Mata, 2006; Gordy, 2002), among others.

Some authors focus on the optimization of the waterfall design in the securitization context (Huang et al., 2007). Surprisingly, the literature on asset selection for securitization is relatively scarce. Some papers study the problem of designing optimal synthetic Collateralized Debt Obligations (CDOs), whose underlying assets are not loans but credit default swaps (CDSs). In this problem, the decision variables represent the relative notional amounts of each CDS — called *investment weights* hereafter — and are modeled as continuous variables in $[0, 1]$ defined on the corresponding simplex. For instance, Jewan et al. (2009) consider a multi-objective optimization problem which consists in determining the investment weights associated with each CDS listed in the iTraxx Europe S5 series. Their objective is to minimize the tail conditional expectation of the CDO while maximizing the spread of a given tranche. This problem is solved under various constraints using a genetic algorithm and the objective function is evaluated via Monte Carlo. Similarly, Veremyev et al. (2012) consider three optimization problems for designing synthetic CDOs where the decision variables are the attachment and detachment points of tranches, sometimes combined with the investment weights. The authors solve the problems at hand using the S&P CDO Evaluator and the Portfolio Safeguard solver of AORDA. Bo and Capponi (2014) consider a dynamic programming approach to optimally invest in a money-market account and in a portfolio of CDSs under contagion risk. These papers depart from the problem at hand in the sense that they do not feature integer constraints, short-selling might be permitted, the numerical applications focus on much smaller pools (of size 50, 53 and 2, respectively) and seem hardly scalable to face typical CLO sizes (tens of thousands of loans).

Westerfeld and Weber (2009) tackle the problem of optimal loan selection for CLOs. Their objective function takes the form of a weighted sum of the loans' risk contribution where the weights are the membership indicators and the risk measure is the unexpected loss contribution of the loan, that is, the variance of the loan's loss scaled by its exposure. They propose a heuristic exploiting the property that the objective function only relies on the marginal characteristics of the loans and not on their joint distribution, thereby avoiding the need to postulate a portfolio loss model. Closest to our work is Saunders et al. (2007), who determine the optimal investment weights associated with a given number of buckets made of homogeneous loans for the sake of minimizing a portfolio's risk measure (VaR or CVaR). In this setup, the objective function takes the form of a sum of buckets' contributions scaled by the investment weights, leading to a linear program whose dimension is the number of buckets. The individual contribution of each bucket is evaluated thanks to the loss distribution provided by the large homogeneous pool assumption associated with a single risk factor model. This approach relies on two key features. First, there must be a single systematic factor and the objective function must display the portfolio invariance property, i.e., can be decomposed as the sum of the buckets' contribution. Second, it is assumed that there are infinitely many homogeneous loans in each bucket (rating class), such that the decision variables are investment weights rather than membership indicators. Sirignano et al. (2016) consider a pool of finite size and face the challenging aspects related to the number of loans and the integer constraint by relying on an approximation of the initial problem at hand. They consider a class of objective functions taking the form of a function of the expectation of transformed returns and a rather general discrete factor model whereby each loan is characterized by a set of static features , some time-varying credit state variables, and a common process driving the correlation between the loans' states. They propose an approximate version of the initial optimization program which relies on the asymptotic (large pool) law of the portfolios for this class of models. They discretize the feature space using a sparse grid. Their optimization program searches for a distribution over the space of probability measures whose dimension is the number of loans to be selected, assumed to be given. Although (Saunders et al., 2007) and (Sirignano et al., 2016) represent significant progress, they focus on rather specific settings. Saunders et al. (2007) rely on portfolio invariance and infinitely large homogeneous rating buckets, resulting in a linear but restrictive formulation of the objective. Sirignano et al. (2016) accommodate finite heterogeneous pools via sparse-grid discretization but with a pre-specified selection size and objective functions taking the form of the expectation of transformed returns.

We contribute to the field by proposing an approximate optimization problem to the loan selection problem featuring a general, nonlinear objective function and a large but finite loan pool. This approximate optimization problem delivers a solution that improves the objective value under the exact criterion and that is much faster. We combine clustering and a linearization of ancillary variables to reduce dimensionality without imposing portfolio invariance or fixing the number of selected loans ex ante. This preserves the generality of the objective and the factor model, and scales to large, finite portfolios. Clustering reduces dimension and relaxes the binary constraint. We first group the loans sharing similar characteristics into clusters and treat their centroids as "prototypical loans". Then, we determine the optimal amount to invest while relaxing the binary constraint. The intuition behind this idea is close to the homogeneous pool assumption of Saunders et al. (2007): By grouping the loans sharing similar characteristics into clusters, we create buckets of loans that are rather homogeneous. We compute explicitly the selection indicators by projecting the solution on the set of conformable vectors with binary entries. Although these approximations can be considered separately, combining linearization and clustering delivers promising results. Besides relying on the LP approximation and Laplace transform, we propose a granularity adjustment (GA) to compute the expected loss of a tranche. We apply our approach on two objective functions which aim to be potential targets that a structurer may want to consider, namely (i) a rating defined as a function of the expected loss and the weighted average life of a CLO tranche and (ii) the amount of capital that is released.

The paper is organized as follows. The principles of securitization are recalled in Section 2. Section 3 gives a summary of the main financial variables that may be of interest for the structurer and proposed "standard" modeling choices for each of them as illustration. We describe a general form for the optimization problem in Section 4 and discuss several approximations to efficiently solve the problem. Section 5 presents two financial applications that fit the proposed general problem in the one-factor Gaussian copula model. Section 6 presents an empirical analysis and discusses the results. Finally, Section 7 explains how this approach can be extended to other loss models, based on multiple factors and/or other copulae.

2. Securitization

We consider the problem of an originator willing to design a CLO starting from an initial pool $\mathcal{S} = \{1, 2, \dots, I\}$ of fixed-rate defaultable loans taken from credit portfolios collected from issuers such as banks or other credit institutions. The securitization process consists in two main steps: (i) analyzing the loans in $\mathcal{S}$ and (ii) designing the waterfall (that is, the algorithm that will allocate the cashflows across tranches) on a portfolio $\mathcal{P}$ of loans selected from $\mathcal{S}$ that will be embedded in the structure. This second step consists in designing the pool so that the CLO meets specific features in terms of risk-return trade-off, rating or capital requirements. In this paper, we focus on a standard sequential architecture featuring J tranches given exogenously by a structurer. The tranches bear losses in reverse seniority order; a tranche $j \in \{1, 2, \dots, J\}$ will incur principal losses if and only if the notional of the more junior tranches has been fully wiped out. Each tranche j is characterized by a pair of attachment and detachment points (A_j, D_j) satisfying $0 \le A_j < D_j \le 1$. The J tranches form a partition of the $[0, 1]$ interval in the sense that $A_1 = 0$, $D_J = 1$ and $A_{j+1} = D_j$. Assuming the waterfall to be given, it remains to identify the portfolio $\mathcal{P}$ of loans so as to reach the structurer's objectives. Mathematically, $\mathcal{P}$ can be represented as the vector of IDs of the loans being selected, $\mathcal{P} \subseteq \mathcal{S}$, or via the *membership vector* $\boldsymbol{w} = (w_1, \dots, w_I)$ where $w_i := \mathbb{1}_{\{i \in \mathcal{P}\}}$ is a binary weight indicating whether loan i is embedded in the securitization or not. The membership vector is found by solving an optimization problem. We assume that the loans are purchased from the issuers such that the lifetime of $\mathcal{P}$ corresponds to the maturity of the longest loan.

2.1. Portfolio of defaultable loans

The loss associated with the default of loan $i \in \mathcal{S}$ with maturity M_i can be decomposed as

$$L_i := N_i \times \lambda_i \times B_i \,.$$

In this expression, $N_i > 0$ stands for the outstanding notional, $\lambda_i \in [0, 1]$ is the loss given default (LGD) defined as the ratio of N_i that is lost upon default prior to the maturity M_i, and $B_i := \mathbb{1}_{\{\tau_i \le M_i\}}$ is the associated default indicator with τ_i the default time of the loan i.[1]

The relative credit loss on a non-empty portfolio $\mathcal{P}$ with membership vector $\boldsymbol{w}$ reads as

$$L(\boldsymbol{w}) := \frac{\sum_{i=1}^{I} w_i L_i}{N(\boldsymbol{w})} \quad \text{where} \quad N(\boldsymbol{w}) := \sum_{i=1}^{I} w_i N_i$$

is the pool's outstanding notional at inception. Because the membership vector $\boldsymbol{w}$ belongs to the set $\mathbb{B}^I$ of I-dimensional bit arrays, the size of the portfolio $\mathcal{P}$ is just its Hamming weight: $|\boldsymbol{w}| := \sum_{i=1}^{I} w_i$. The relative loss is thus 0 if no loan in $\mathcal{P}$ defaults prior to the pool maturity $\max_{i\in\mathcal{P}}(M_i)$. On the other hand, the loss is $\frac{1}{N(\boldsymbol{w})}\sum_{i=1}^{I} w_i \lambda_i N_i \le 1$ if all the loans in $\mathcal{P}$ default before T, where the equality holds only if all the LGDs are equal to one.

The expected loss (EL) on $\mathcal{P}$ is the average loss incurred by an investor holding the pool until T:

$$\mathrm{EL}(\boldsymbol{w}) := \mathbb{E}(L(\boldsymbol{w})) = \frac{\sum_{i=1}^{I} w_i \mathbb{E}\big(L_i\big)}{N(\boldsymbol{w})} \,.$$

In the special case where the LGDs are considered as constant, the EL on loan i reads as

$$\mathbb{E}\big(L_i\big) = N_i \lambda_i p_i \quad \text{with} \quad p_i := \mathbb{E}(B_i) = \mathbb{P}(\tau_i \le M_i) \,.$$

2.2. Tranches of a portfolio

The size of tranche j relative to $N(\boldsymbol{w})$ is $S_j := D_j - A_j > 0$, satisfying $\sum_{j=1}^{J} S_j = 1$. The loss on the tranche relative to its outstanding notional $N_j(\boldsymbol{w}) = S_j \times N(\boldsymbol{w})$ becomes:

$$L_j(\boldsymbol{w}) := \left[\frac{L(\boldsymbol{w}) - A_j}{S_j}\right] \quad \text{where} \quad [x] := \min(1, \max(0, x)) \,.$$

Observe that $L_j(\boldsymbol{w}) = 0$ if and only if $L(\boldsymbol{w}) \le A_j$ and $L_j(\boldsymbol{w}) = 1$ if and only if $L(\boldsymbol{w}) \ge D_j$. Note the distinction: L_j denotes the loss on loan j in the master pool $\mathcal{S}$, while $L_j(\boldsymbol{w})$ denotes the loss on tranche j in the selected portfolio $\mathcal{P}$, with membership vector $\boldsymbol{w}$. The same distinction applies to N_j and $N_j(\boldsymbol{w})$. The sum of the losses on the J tranches is equal to the total loss experienced by the portfolio: $\sum_{j=1}^{J} L_j(\boldsymbol{w}) N_j(\boldsymbol{w}) = L(\boldsymbol{w}) N(\boldsymbol{w})$.

Similar to the EL on a portfolio $\mathcal{P}$, one can compute the EL on a tranche j of $\mathcal{P}$, defined as the expected value of the loss on the tranche relative to its notional:

$$\mathrm{EL}_j(\boldsymbol{w}) := \mathbb{E}(L_j(\boldsymbol{w})) = \mathbb{E}\left(\left[\frac{L(\boldsymbol{w}) - A_j}{S_j}\right]\right) \,. \tag{1}$$

Because of the nonlinear min-max operator $x \mapsto [x]$, computing EL_j requires the distribution of $L(\boldsymbol{w})$. There are two classes of portfolio loss models. Top-down models aim to directly model the total losses on the credit portfolio $L(\boldsymbol{w})$ as a single variable (Giesecke et al., 2011). CreditRisk+, for instance, postulates a model for the number of defaults in the portfolio by assuming for the latter a binomial distribution with random (gamma-distributed) mean. Bottom-up models, instead, start from the marginal distributions of the individual losses and postulate some copula to obtain a joint distribution, from which the distribution of $L(\boldsymbol{w})$ can be inferred. This is the setup adopted in the Basel regulation, which assumes a one-factor Gaussian copula framework to couple the default

[1] Another common decomposition is to replace N_i by EAD_i, the exposure at default of loan i. The latter is typically smaller than the former in presence of amortization, hence, considering N_i instead of EAD_i is a conservative approach. However, note that this difference does not really matter in our setup because it simply amounts to redefine the LGD factor. Considering the LGDs as known constants may seem unrealistic but our point in this section is to sketch the salient features of the securitization process and analyze the resulting optimization problem. In practice, one could consider variants of this setup featuring stochastic LGDs; see Section 7.

times (see Section 3.1). This is also the setup adopted by Jarrow and Van Deventer (2023) to compute CLO tranches' loss probabilities and capital factors. For a comparison between top-down and bottom-up approaches, we refer to Giesecke (2008). We focus on bottom-up models in the sequel. Besides modeling the individual characteristics of the loans (N_i, λ_i and p_i) one thus needs to assume a specific dependence structure between the L_i's, hence, between the B_i's. Factor models will be discussed in Sections 3 and 7.

3. Main financial variables

The features of the securitized products depend both on the waterfall and on the loans embedded in the CLO, that is, on the characteristics of the portfolio $\mathcal{P}$. In Modern Portfolio Theory, the objective of the portfolio manager is to identify the variables that matter to the investors (e.g., the expected return and variance in Markowitz's theory), build an objective function out of them that would depict the investor's preference (quadratic expected utility), and solve an optimization problem to determine how to invest in the assets of a given universe. Similarly, the purpose of the structurer is to identify the variables that matter in the design of the product, build an objective function trading off between those, and solve for the membership vector $\boldsymbol{w}$. It is not the purpose of the paper to identify the most relevant objective function a structurer may want to consider, as many different choices can be relevant. Instead, we discuss some standard indicators extensively used in asset-backed securities (ABS) such as the expected loss on the tranche discussed above, the weighted average coupon, the weighted average maturity, the weighted average life and the capital requirements. Two possible objective functions built upon these variables will be considered in Section 5 for illustration purposes.

3.1. Expected loss on the tranche

The expected loss on a tranche (ELoT) is perhaps the most important risk factor of a tranche. It is driven by the joint behavior of the default indicators B_i's. In line with the Basel model used to compute capital requirements discussed in Section 3.2, specific attention is given to the one-factor Gaussian copula (1FGC) approach, which is used for illustration purposes throughout the text. Alternative choices are possible, as discussed in Section 7. This formulation provides the unexpected loss using asymptotic results that rely on the LP assumption.

In the one-factor model, the default event of loan i is modeled indirectly using latent variables Z_i's taking values below a given threshold z_i: $\{\tau_i \leq M_i\} \Leftrightarrow \{B_i = 1\} \Leftrightarrow \{Z_i \leq z_i\}$.

Letting F_X denote the (invertible) cumulative distribution function of a continuous random variable X, setting $z_i := F_{Z_i}^{-1}(p_i)$ guarantees that $\mathbb{P}(Z_i \leq z_i) = p_i$, hence, allows one to preserve the marginal default probability (PD) of the loans. The dependence structure between the default indicators is handled by decomposing the Z_i's as a weighted sum of two standard i.i.d. variables: a common (systematic) random variable Z and a firm-specific (idiosyncratic) random variable ϵ_i,

$$Z_i = a_i Z + \sqrt{1 - a_i^2}\,\epsilon_i \quad \text{where} \quad Z, \epsilon_1, \dots, \epsilon_I \text{ are independent and standardized.} \tag{2}$$

The loading factors $a_i \in (-1, 1)$ control the correlation between the credit worthiness variables Z_i's, hence, between the default indicators B_i's. In such a model, the default events are independent given Z and the conditional probability of default of firm i is $p_i(Z) := \mathbb{P}(Z_i \leq z_i | Z)$.

The ELoT (1) can be estimated using numerical approaches, such as Monte Carlo. However, for specific portfolio loss models more efficient methods exist. In single-factor models, for instance, one can exploit conditional independence and Laplace transform: Compute the product of the moment generating functions associated with the conditional distributions of the individual losses $p_i(Z)$, get the portfolio loss distribution using inverse Fast Fourier Transform (FFT), and integrate out Z. This approach is very efficient and delivers very good results for small or moderate pool sizes. Otherwise, numerical issues can appear in the inverse FFT, and choosing an appropriate number of points can be critical (see Supplementary Material A.1 for more details about this approach). For large pools, the ELoT can be efficiently approximated by relying on the LP approximation (Gordy, 2003). If the portfolio is infinitely granular in the sense that it has a nearly infinite number of loans with infinitesimal relative exposures, the impact of the idiosyncratic variables can be completely diversified away and the portfolio loss converges in law to its conditional expectation:

$$L(\boldsymbol{w}) \xrightarrow{I\to\infty} L(\boldsymbol{w}; Z) := \mathbb{E}(L(\boldsymbol{w})|Z) \quad \text{with}$$
$$L(\boldsymbol{w}; z) = \frac{1}{N(\boldsymbol{w})} \sum_i w_i N_i \lambda_i p_i(z)\,. \tag{3}$$

In other words, the randomness embedded in $L(\boldsymbol{w}) = L(\boldsymbol{w}; Z, \epsilon)$ results solely from Z, asymptotically: In the limit where $I \to \infty$, the impact of the idiosyncratic risks ϵ vanishes. The asymptotic ELoT on tranche j is thus obtained by replacing $L(\boldsymbol{w})$ by its asymptotic version $L(\boldsymbol{w}; Z)$ in (1). We conclude that for I large enough, the ELoT on tranche j can be estimated from its LP approximation as an integral of the asymptotic EL with respect to Z:

$$\mathrm{EL}_j(\boldsymbol{w}) := \mathbb{E}(L_j(\boldsymbol{w})) \approx \mathbb{E}(L_j(\boldsymbol{w}; Z)) = \int_{-\infty}^{\infty} \left[\frac{L(\boldsymbol{w}; z) - A_j}{S_j}\right] dF_Z(z)$$
$$=: \mathrm{EL}_j^{\infty}(\boldsymbol{w})\,. \tag{4}$$

Remark 1. The 1FGC model corresponds to the special case where $Z, \epsilon_1, \dots, \epsilon_I \sim \mathcal{N}(0, 1)$. This is convenient because then $Z_i \sim \mathcal{N}(0, 1)$ for all i and any choice of the loadings. Letting Φ denote the cumulative distribution of the standard normal distribution, $\phi = \Phi'$ its density, $z_i = \Phi^{-1}(p_i)$ its p_i-quantile and setting the loadings $a_i = \sqrt{\rho_i}$ for $\rho_i \in [0, 1]$ yields,

$$p_i(Z) = \Phi\left(\frac{\Phi^{-1}(p_i) - \sqrt{\rho_i}Z}{\sqrt{1 - \rho_i}}\right). \tag{5}$$

The Gaussian coupling is very popular because of its use in Basel regulation. Nevertheless, this assumption can be easily relaxed; what matters for the LP trick to work is the independence of the underlying variables $Z, \epsilon_1, \dots, \epsilon_I$. Our method encompasses any model to which the LP approximation applies such as the t-copula or multi-factors models, including some stochastic LGD extensions. Examples of families of models that can be accommodated are provided in Section 7.

Clearly, computing $\mathrm{EL}_j^{\infty}(\boldsymbol{w})$ in (4) only requires to perform an integral with respect to a standard normal variable, which is much easier than evaluating $\mathrm{EL}_j(\boldsymbol{w})$ in (1). Moreover, in contrast with the latter, the former can be linearized in the weights (see Section 5.3). This feature will play a central role later. Nevertheless, it remains true that $\mathrm{EL}_j^{\infty}(\boldsymbol{w})$ is an asymptotic approximation to the actual ELoT, $\mathrm{EL}_j(\boldsymbol{w})$: Eq. (3) implicitly assumes that the conditional variance of the loss collapses to zero, i.e., that the idiosyncratic risks are fully diversified away. Clearly, this is a strong assumption when I is not too large (see Supplementary Material A.2 for a discussion about the quality of the approximation $\mathrm{EL}_j^{\infty}(\boldsymbol{w}) \approx \mathrm{EL}_j(\boldsymbol{w})$). It is possible to adjust the LP expression, correcting for the finite size pool. Such GA have been studied in details in the context of quantile estimation, at the heart of value-at-risk and Basel regulation on capital requirements (Gordy & Lutkebohmert, 2013). Inspired by the latter and using the expressions derived in Barbagli and Vrins (2025), we propose a GA tailored for the ELoT in Supplementary Material A.3. In a nutshell, the idea consists in performing an Edgeworth expansion of L around $\mathbb{E}(L|Z)$ in a single risk factor model, truncating this expansion at the second moment of $L - \mathbb{E}(L|Z)$. Compared to the asymptotic case, this extra term captures the variance of the conditional portfolio loss — which, in general, is zero only when the pool size is infinite. By correcting for the residual idiosyncratic risk, the resulting expression is expected to provide a more accurate view of the portfolio loss, hence, of the ELoT.[2] The adjustment for a tranche (A_j, D_j) with size $S_j = D_j - A_j$ and constant LGDs is $\mathrm{EL}_j^{GA}(\boldsymbol{w}) = \mathrm{EL}_j^{\infty}(\boldsymbol{w}) + \hat{\Delta}\mathrm{EL}_j(\boldsymbol{w})$, where, in the special case of the 1FGC model,[3]

$$\hat{\Delta}\mathrm{EL}_j(\boldsymbol{w}) = \frac{h(\mu^{-1}(A_j)) - h(\mu^{-1}(D_j))}{2S_j} \quad \text{with}$$

$$h(x) = \frac{-\phi(x)\sigma^2(x)}{\mu'(x)}\ , \quad \mu(x) = \sum_i w_i\lambda_i p_i(x) \quad \text{and}$$

$$\sigma^2(x) = \sum w_i^2\lambda_i^2 p_i(x)(1-p_i(x)).$$

Interestingly, $\hat{\Delta}\mathrm{EL}_j$ is analytical, hence, comes for free in terms of computational cost compared to EL_j^{∞}. Featuring a second-order adjustment, though, EL_j^{GA} is nonlinear in the weights.

3.2. Capital requirements

In Basel regulation, capital requirements correspond to the unexpected losses computed under the 1FGC model. It is the difference between the value-at-risk (at 99.9 % confidence level) and the expected loss (4) when relying on (3) and (5). The corresponding provisions add up,[4] that is, the capital $K(\boldsymbol{w})$ associated with holding $\mathcal{P}$ is the sum of the capital requirements associated with the underlying loans:

$$K(\boldsymbol{w}) = \frac{\sum_{i=1}^I w_i K_i N_i}{N(\boldsymbol{w})}\ , \tag{6}$$

where K_i is the capital requirement of loan i per unit of outstanding exposure. Capital requirements at the loan level are governed by the Basel guidelines and salient points associated with their computation are recalled in Supplementary Material B. The capital requirement $K_j(\boldsymbol{w})$ for the tranche j (relative to its notional) is a regulatory variable that is defined by Basel guidelines. Leveraging on the operator [.], its expression in the official documents (BCBS, 2019) simplifies to:

$$K_j(\boldsymbol{w}) := \delta_j(\boldsymbol{w}) + (1-\delta_j(\boldsymbol{w}))K_j^{\mathrm{CLO}}(\boldsymbol{w}) \quad \text{where} \quad \delta_j(\boldsymbol{w}) := \left[\frac{K(\boldsymbol{w}) - A_j}{S_j}\right] \tag{7}$$

and

$$K_j^{\mathrm{CLO}}(\boldsymbol{w}) := \frac{e^{a(\boldsymbol{w})u_j(\boldsymbol{w})} - e^{a(\boldsymbol{w})l_j(\boldsymbol{w})}}{a(u_j(\boldsymbol{w}) - l_j(\boldsymbol{w}))}\ , \quad \text{with}$$

$$a(\boldsymbol{w}) := \frac{-1}{K(\boldsymbol{w})}\ , \quad u_j(\boldsymbol{w}) := D_j - K(\boldsymbol{w}) \quad \text{and} l_j(\boldsymbol{w}) := \max(A_j - K(\boldsymbol{w}); 0).$$

Again, note the difference between K_j, the capital associated with loan j in the pool $\mathcal{S}$, and $K_j(\boldsymbol{w})$, which represents the capital requirement of tranche j on the portfolio $\mathcal{P} \subseteq \mathcal{S}$. Because $K_j(\boldsymbol{w})$ depends on $\boldsymbol{w}$ via $K(\boldsymbol{w})$ only, we introduce for further reference the notation $k_j(\cdot)$:

$$K_j(\boldsymbol{w}) = k_j(K(\boldsymbol{w}))\ . \tag{8}$$

The capital released by selling tranche j is given by the difference of the capital on the portfolio $\mathcal{P}$ and the capital that one must hold when keeping all the tranches of $\mathcal{P}$ but the jth one:

$$\Delta K_j(\boldsymbol{w}) = K(\boldsymbol{w}) - \sum_{k=1}^{J} \mathbb{1}_{\{k\neq j\}} S_k K_k(\boldsymbol{w}) > 0\ . \tag{9}$$

Remark 2. In general, $\Delta K_j(\boldsymbol{w}) \neq S_j K_j(\boldsymbol{w})$ because the portfolio invariance property holds for loans but not for tranches: The capital requirements of a tranched structure exceed the capital requirements on the portfolio in the sense that $\sum_{j=1}^J S_j K_j(\boldsymbol{w}) > K(\boldsymbol{w})$ when $J > 1$.[5] However, the total capital on the structure $\sum_{j=1}^J S_j K_j(\boldsymbol{w})$ does not depend on the number of tranches nor on the choice of the attachment/detachment points.

3.3. Weighted average maturity, coupon and life

Besides credit risk, investors typically also care about the maturity and the coupon of the product they invest in. However, a portfolio is often made of thousands of loans, each having its own maturity M_i and rate r_i. To circumvent this problem, investors in the ABS market rely on meta-variables that describe the aggregated behavior of the portfolio as a whole. The weighted average maturity (WAM) and weighted average coupon (WAC) provide the "portfolio equivalents" of the loans' maturity and coupon rate. The corresponding expressions (defined relative to the pool notional for consistency with the earlier sections) are given by the average of the maturities and coupons of the loans embedded in $\mathcal{P}$, weighted by the relative notional $N_i/N(\boldsymbol{w})$, see e.g. (Veronesi, 2010) or (Petitt et al., 2015):

$$\mathrm{WAC}(\boldsymbol{w}) := \frac{1}{N(\boldsymbol{w})}\sum_{i=1}^I w_i N_i r_i \quad \text{and} \quad \mathrm{WAM}(\boldsymbol{w}) := \frac{1}{N(\boldsymbol{w})}\sum_{i=1}^I w_i N_i M_i. \tag{10}$$

A crucial risk variable in fixed income analysis is *duration*, aiming to capture the sensitivity of the value of a security to interest risk. A similar (although slightly different) concept in ABS is the weighted average life (WAL), aiming to represent the expected amount of time needed to repay one unit of principal. Similarly to the ELoT, a detailed calculation of the weighted average life of a tranche (WALoT) would require a stochastic model and a description of the waterfall driving the amortization profile. Nevertheless, the goal of the WAL is merely to give an indication of the "average lifetime" of the portfolio rather than to be an accurate estimation of interest-rate risk, as illustrated by the market conventions and the lack of academic literature on the topic. In the ABS industry, the WAL of $\mathcal{P}$ is practically defined as the sum of the payment times t_i weighted by the part of the cashflow paid at t_i that relates to capital. Assuming we are at time $t = 0$ and $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$ is the set of payment dates of the cashflows coming from the loans:

$$\mathrm{WAL}(\boldsymbol{w}) = \frac{\sum_{t_i\in\mathcal{T}} t_i \times p(t_i;\boldsymbol{w})}{\sum_{t_i\in\mathcal{T}} p(t_i;\boldsymbol{w})}\ ,$$

where $p(t;\boldsymbol{w})$ is the amount of capital repayment paid at time t in the portfolio $\mathcal{P}$. The function $p(t;\boldsymbol{w})$ depends on the loan characteristics such as the coupon rates, the maturities and the possible prepaid amounts.[6]

In practice, the WAL of a tranche is often modeled using a top-down approach where the portfolio is taken as a single loan whose "maturity" and "coupon rate" can be regarded as $T = \mathrm{WAM}(\boldsymbol{w})$ and $r = \mathrm{WAC}(\boldsymbol{w})$. In this setup, the WAL on a portfolio $\mathcal{P}$ depends on $\boldsymbol{w}, r, T$ and a parameter (possibly functional) γ called *prepayment rate* which controls the rate at

[2] The magnitude of this correction heavily depends on the size of the pool, the loading factors and the design of the tranches. See the difference between columns $F^{\infty,*}$ and $F^{GA,*}$ in Tables 4 and 5, as well as Fig. 4 in the Supplementary Material for an illustration of the impact of the loading factor.

[3] Note that μ is monotonic in x, hence, its inversion is easily performed using root search. The expression in the Supplementary Material is more general and can accommodate stochastic LGDs, possibly correlated with Z.

[4] This property is called *portfolio invariance*.

[5] In other words, keeping the portfolio $\mathcal{P}$ is less costly in terms of capital that keeping all the J tranches of that portfolio when $J > 1$. This seems inconsistent, but is nonetheless inherent to the Basel framework. We presume that this is because it is practically irrelevant: There is no incentive in tranching a portfolio if all the tranches are kept.

[6] Note that the WAL is computed by assuming that the capital payment cashflows are deterministic which is contradictory with the existence of credit risk. This can be seen as a convention. The WAL aims to convey information about other features of $\mathcal{P}$. In practice, the lifetime of a portfolio is impacted by the early termination related to the defaults, which can be considered as a form of prepayment. Therefore, the impact of credit risk on the actual lifetime of a pool could be captured by playing with the prepayment rate.

which the underlying principal is repaid ahead of schedule. Prepayment is a phenomenon that depends on macro-economic environment (such as interest rates) and private events impacting the customer's life (divorce, moves, fire sales, death, etc), and is therefore difficult to model. Standard market practice is to describe prepayment using the PSA model which provides a specific shape for the monthly prepayment rate (CPR) profile that can be adjusted to the portfolio dynamics by tuning a parameter α called "PSA level". Specifically, the CPR profile grows linearly with the month index i until the 30th month and is constant afterwards:[7]

$$\mathrm{CPR}(i) = \alpha \times 6\,\% \times \min\left(1; \frac{i}{30}\right).$$

A 100 % PSA pool corresponds to $\alpha = 1$. For more information about PSA we refer to Veronesi (2010) which provides a very comprehensive treatment of ABS notions.

Recall that the focus of this paper is on the optimization problem related to securitization designs, not to propose a precise model for a particular waterfall. Therefore, we model the WAL using a schematic continuous-time prepayment framework inspired from the discrete-time PSA model leading to a semi-closed form solution. Specifically, we assume that

$$\mathrm{WAL}(\boldsymbol{w}) = \frac{\int_0^T t \times p(t;\boldsymbol{w})dt}{\int_0^T p(t;\boldsymbol{w})dt}$$

where $T = \mathrm{WAM}(\boldsymbol{w})$ is the WAM of $\mathcal{P}$. The dynamics of $p(\cdot,\boldsymbol{w})$ depend on the principal repayment, on the continuously compounded version $\tilde{r} = \ln(1+r)$ of the WAC coupon rate r (which controls the amortization profile, hence, the stream of so-called *scheduled principal payments*) as well as on the profile of the portfolio's prepayment rate $\gamma(\cdot)$ (stream of *unscheduled principal payments*). Even in the case of a homogeneous pool ($r_i = r$ and $M_i = M$), the portfolio coupon (that is, the cashflow paid) at time t depends on t whenever $\gamma(t) > 0$. This is because the cashflow related to principal payment at t increases in case of prepayment such that, afterwards, the principal payments drop.

We show in Supplementary Material C that, for a tranche j and assuming that $p(t;\boldsymbol{w})$ represents the principal cashflows relative to the notional of the pool, the WAL on a tranche j can be modeled as:

$$\mathrm{WAL}_j(\boldsymbol{w}) := \frac{\int_{T_{j-1}}^{T_j} t \times (c(t) + (\gamma(t) - \tilde{r})n(t))dt}{S_j}, \tag{11}$$

where

$$c(t) := \frac{\tilde{r}\mathrm{e}^{-\int_0^t \gamma(u)du}}{1 - e^{-\tilde{r}T}} \quad \text{and} \quad n(t) := 1 + c(t)e^{\tilde{r}t}\int_0^t \frac{\tilde{r} - \gamma(u) - c(u)}{c(u)e^{\tilde{r}u}}du \tag{12}$$

are the coupon per unit of time and outstanding notional of $\mathcal{P}$ relative to $N(\boldsymbol{w})$ at time t, respectively, T_j is the time at which tranche j is fully repaid and the denominator comes from the fact that the sum of the principal payment on tranche j relative to the notional of $\mathcal{P}$ is equal to $N_j/N(\boldsymbol{w}) = S_j$. We consider specific profiles for the prepayment rate function $t \mapsto \gamma(t)$ inspired from the PSA model and show that, in such cases, the times T_j are easily found by computing the root of monotonic functions. Note that since the WAL of a tranche is a function of r, T and γ,

$$\mathrm{WAL}_j(\boldsymbol{w}) = \mathrm{wal}_j(\mathrm{WAC}(\boldsymbol{w}), \mathrm{WAM}(\boldsymbol{w})) \tag{13}$$

for some function $\mathrm{wal}_j(\cdot,\cdot)$ because $\gamma(\cdot)$ is a function of $\mathrm{WAC}(\boldsymbol{w})$ and $\mathrm{WAM}(\boldsymbol{w})$, too.

[7] A potential justification for this profile is to notice that people take time before thinking of refinancing their loans as the market conditions or their private situations need to evolve in order for prepayment to make sense.

4. Optimization problems

Having given an overview of the main financial variables driving the risks associated with credit portfolio tranches and some corresponding models to evaluate them, we proceed with the analysis of the related optimization program and derive some alternatives that help circumvent the computational issue resulting from its NP-hard nature. Two parametric forms of objective functions that a structurer may want to minimize and that will be considered in our numerical experiments are introduced in Section 5. Before detailing the different optimization programs, Table 1 summarizes the main ELoT approximations introduced in Section 3, together with their implementation scope.

4.1. Original problem

The goal of the structurer is to find the vector $\boldsymbol{w}$ solving:

Program 1

$$\min_{\boldsymbol{w}\in\mathbb{B}^I} F(\boldsymbol{w}) \quad \text{s.t.} \quad \begin{cases} \boldsymbol{w} & \in \mathcal{W} \\ N(\boldsymbol{w}) & \geq \underline{N} \end{cases}$$

where $\underline{N} > 0$ is a strictly positive finite lower bound on the pool's notional and $\mathcal{W}$ represents a set of optional linear constraints involved in the securitization, e.g., aiming to foster diversification across regions or economic sectors. Function $F(\cdot)$ represents the preferences of the originator and is typically nonlinear in the main financial variables, hence, in $\boldsymbol{w}$. This is a knapsack problem (Wolsey, 2020): Program 1 is a MINLP which, moreover, is typically high-dimensional (the number of loans is often between 1000 and 100,000) and for which the derivative of the objective function is hard to compute. This is due to the fact that the derivatives of some of the main financial variables are not available in closed form, like the exact ELoT (see Supplementary Material A.1) and the WALoT (see Supplementary Material C). This problem is NP-hard. Therefore, finding efficiently the solution to an approximate problem might be better than tackling the exact problem using derivative-free solvers. As stated by Sirignano et al. (2016), "MINLP solvers' success can vary in practice depending upon the particular problem. A priori, the computational performance of a particular MINLP solver on a problem is difficult to anticipate", a statement that our empirical analysis confirms. This is the purpose of the next section to propose approximate optimization problems.

4.2. Approximate optimization problems

The main challenges associated with Program 1 are of two types. First, Program 1 is a high-dimensional problem whose optimization domain is discrete ($\boldsymbol{w} \in \mathbb{B}^I$ where I is large). Second, F is typically nonlinear (see Section 5 for examples of objective functions). In this section, we consider three approximations of Program 1 that mitigate these issues.

4.2.1. Clustering and projection

The main idea of this approximation is to notice that if two loans are very similar, choosing one or the other has a limited impact on the objective function F. One could thus group the I loans into Q clusters and, instead of assigning a binary weight w_i to each loan, assign a nonnegative real number $\tilde{w}_q$ to each cluster $q \in \{1, 2, \dots, Q\}$ representing the notional that we invest in each cluster relative to the maximum outstanding notional $\overline{N}_q$ available in the latter. Noting $q(i)$ the cluster of loan i, one simply gets $\overline{N}_q = \sum_{i=1}^I \mathbb{1}_{\{q(i)=q\}} N_i$. By doing so, the optimization domain of the sub-program changes from $\mathbb{B}^I$ to $\mathbb{R}_+^Q$ defined as the set of Q-dimensional vectors with nonnegative real entries. The benefits of doing this are twofold. First, the dimension is reduced from I to Q where $Q \ll I$, typically. Second, the integer constraint is relaxed. Of course, the solution vector $\tilde{\boldsymbol{w}} \in \mathbb{R}_+^Q$ of amounts (relative to the maximum outstanding notional) to be invested in each cluster will then need

Table 1
Summary of ELoT approximations and their main characteristics. Programs are introduced in this section; this table serves as an overview of the approximations discussed in Section 3.

	ELoT approximations			
Property	**LP**	**GA**	**Laplace**	**CLT**
Linearity in w	Yes	No	No	No
Typical pool size	Large ($I \gtrsim 1000$)	Medium-Large	Small-Medium ($I \lesssim 1000$)	Small-Medium
Computational cost	Very low	Low	High	Very high
Accuracy	Moderate	High	Very high	Very high
Feasibility in programs	All	1-2	1-2	1-2

to be projected on the set of attainable portfolios to obtain $\boldsymbol{w} \in \mathbb{B}^I$. This is tackled in a second step.

Specifically, we first run a clustering algorithm on the pool $\mathcal{S}$. Next, we consider the pool $\tilde{\mathcal{S}}$ where each "prototypical loan" q has the characteristics of its centroid, $C(q)$, except that its "notional" is set to $\overline{N}_q$ and that fractional shares of such loans can now be purchased. This leads to the following program:

Program 2: (Clustered version of Program 1)

$$\min_{\tilde{\boldsymbol{w}} \in \mathbb{R}_+^Q} F(\tilde{\boldsymbol{w}}) \quad \text{s.t.} \quad \begin{cases} \tilde{\boldsymbol{w}} & \leq \mathbf{1} \\ \tilde{\boldsymbol{w}} & \in \tilde{\mathcal{W}} \\ N(\boldsymbol{w}) & \geq \underline{N} \end{cases}$$

where $\mathbf{1} \in \mathbb{R}^Q$ is a vector of ones and $N(\tilde{\boldsymbol{w}}) := \sum_{q=1}^{Q} \tilde{w}_q \overline{N}_q$ is the total notional invested. Finally, we need to map the "optimal cluster weight vector" $\tilde{\boldsymbol{w}}^* \in \mathbb{R}_+^Q$ to a portfolio $\boldsymbol{w}^* \in \mathbb{B}^I$. To this end, we select the loans closest to their centroid in each cluster until the optimal proportion is reached.

Projection 2:

$$\boldsymbol{w}^* \in \operatorname{argmin}_{\boldsymbol{w} \in \mathbb{B}^I} \sum_{i=1}^{I} w_i d_i \;\text{ s.t. } \sum_{i=1}^{I} w_i N_i \mathbb{1}_{\{q(i)=q\}} \geq \tilde{w}_q^* \overline{N}_q \;\forall q \in \{1, 2, \dots, Q\},$$

where d_i is the distance of loan i from its centroid $C(q(i))$. By doing so, one obtains a portfolio $\boldsymbol{w}^* \in \mathbb{B}^I$ which allocates the wealth across clusters in a similar way as $\tilde{\boldsymbol{w}}^* \in \mathbb{R}_+^Q$, while being close to the prototypical loans forming $\tilde{\boldsymbol{w}}^*$. Compared to Program 1, Program 2 is much more tractable. It can be solved using a derivative-free algorithm and a Mixed-Integer Linear Program (MILP) for Projection 2. Moreover, the projection step is trivial because it simply consists in selecting, for each cluster q, the loans being closest from their centroid while a notional threshold is not reached.

4.2.2. Linearization

Another common procedure to address the complexity of optimization problems is to consider approximations exploiting the shape of the functions at hand such as linearization schemes. To this end, let us write the objective F as a function of n ancillary variables $X_0(\boldsymbol{w}), X_1(\boldsymbol{w}), \dots, X_n(\boldsymbol{w})$ and propose, for each of them, a linear approximation $X_i(\boldsymbol{w}) \approx \boldsymbol{w}^\top \boldsymbol{c}_i =: \hat{X}_i(\boldsymbol{w})$ for $i \in \{0, 1, \dots, n\}$ where $\boldsymbol{c}_i \in \mathbb{R}^I$. This yields an alternative version of F featuring the linearized ancillary variables:

$$F(\boldsymbol{w}) := f\left(X_0(\boldsymbol{w}), \dots, X_n(\boldsymbol{w})\right) \approx f\left(\hat{X}_0(\boldsymbol{w}), \dots, \hat{X}_n(\boldsymbol{w})\right) =: \hat{F}(\boldsymbol{w}) .$$

Note that we do not linearize F, only the ancillary variables X_i. In particular, f can be nonlinear. In our application, the ancillary variables could be the expected loss, the capital requirement or the weighted average life of the tranche. We assume that F is monotonic in at least one of them, and choose the convention that F is increasing in X_0. One can thus optimize F indirectly as follows:

Program 3 (Linear approximation of Program 1)

$$\min_{\hat{\boldsymbol{x}} \in \mathbb{R}^n} \quad g(\hat{\boldsymbol{x}}) := F(\boldsymbol{w}^*(\hat{\boldsymbol{x}}))$$

where

Sub-program 3

$$\boldsymbol{w}^*(\hat{\boldsymbol{x}}) \in \operatorname{argmin}_{\boldsymbol{w} \in \mathbb{B}^I} \hat{X}_0(\boldsymbol{w}) \quad \text{s.t.} \quad \begin{cases} \boldsymbol{w} & \in \mathcal{W} \\ N(\boldsymbol{w}) & \geq \underline{N} \\ \hat{X}_i(\boldsymbol{w}) & = \hat{x}_i \;, \quad i \in \{1, 2, \dots, n\} \;. \end{cases}$$

Concretely, we seek to find the membership vector $\boldsymbol{w}^*$ minimizing the objective function $F(\boldsymbol{w})$ indirectly, via a function g of a vector $\hat{\boldsymbol{x}} = (\hat{x}_1, \dots, \hat{x}_n)$ of scalar variables. The value $g(\hat{\boldsymbol{x}})$ coincides with the objective function F evaluated at the portfolio $\boldsymbol{w}^*(\hat{\boldsymbol{x}})$ minimizing the linear approximation $\hat{X}_0$ of the monotonic variable X_0 over a restriction of $\mathbb{B}^I$ satisfying $\hat{X}_i(\boldsymbol{w}) = \hat{x}_i$, $i \in \{1, \dots, n\}$. Instead of minimizing F by solving a MINLP in the space $\mathbb{B}^I$ (Program 1) we now have a nonlinear program (NLP) in dimension $n \ll I$ (Program 3) which features the solution of a MILP (Sub-program 3) at each evaluation of the objective function g. Note that this sub-program could be *infeasible* for some vectors $\hat{\boldsymbol{x}}$ imposed for the linearized ancillary variables, as the set of portfolios meeting these constraints might be empty. In practice, replacing the equality constraints of Sub-program 3 by inequalities helps decrease the number of cases for which Sub-program 3 is infeasible. Infeasible cases are ruled out by forcing then g to return a very large number. Moreover, in the empirical analysis, it is trivial to find lower and upper bounds for each element of $\hat{\boldsymbol{x}}$ for a given pool $\mathcal{S}$ of loans and for a given $\underline{N}$. This also decreases the number of infeasible cases.

4.2.3. Combining linearization, clustering and projection

Finally, we consider a third variant where the clustering and the linearization approaches are combined. This leads to a program similar to Program 3 but where Sub-program 3 is replaced by Sub-program 4 (optimizing over $\mathbb{R}_+^Q$ instead of $\mathbb{B}^I$), and project the solution of Sub-program 4 $\tilde{\boldsymbol{w}}^* \in \mathbb{R}_+^Q$ onto $\mathbb{B}^I$ to get our "optimal" portfolio $\boldsymbol{w}^*$ using Projection 2. Note that relaxing the binary constraint also reduces the number of cases where Sub-program 3 is infeasible. This leads to the following program:

Program 4 (Clustered version of Program 3)

$$\min_{\hat{\boldsymbol{x}} \in \mathbb{R}^n} \quad g(\hat{\boldsymbol{x}}) := F(\boldsymbol{w}^*(\hat{\boldsymbol{x}}))$$

where

1. **Sub-program 4** (Clustered version of Sub-program 3):

$$\tilde{\boldsymbol{w}}^* \in \operatorname{argmin}_{\tilde{\boldsymbol{w}} \in \mathbb{R}_+^Q} \hat{X}_0(\tilde{\boldsymbol{w}}) \quad \text{s.t.} \quad \begin{cases} \tilde{\boldsymbol{w}} & \leq \mathbf{1} \\ \tilde{\boldsymbol{w}} & \in \tilde{\mathcal{W}} \\ N(\tilde{\boldsymbol{w}}) & \geq \underline{N} \\ \hat{X}_i(\tilde{\boldsymbol{w}}) & = \hat{x}_i \;, \quad i \in \{1, 2, \dots, n\} \end{cases}$$

2. **Projection 2** $\tilde{\boldsymbol{w}}^* \mapsto \boldsymbol{w}^*$ (as in Program 2).

Program 4 combines the best of the two worlds. It can be solved using a derivative-free algorithm that will compute at each evaluation of the objective function g the solution of Sub-program 4, that is, a Linear Program (LinP) and the solution of Projection 2, that is, a MILP. As stated before, finding a solution to the MILP is trivial. This program is fast and provides high-quality solutions in our applications, as shown in Section 6.

5. Financial applications

We now introduce two different applications of loan selection for securitization. In line with the existing literature (Jewan et al., 2009; Sirignano et al., 2016), we first consider the case of a structurer working for a development bank whose mission is to stimulate local economy by reducing the SMEs' cost of funds. A common procedure to achieve this goal is by securitizing loans bought from commercial banks, and issuing the senior tranche. The risk associated with the first tranche is kept by the investment bank. The senior tranche, however, benefits from a credit enhancement mechanism and can now be sold to risk-averse institutional investors such as pension funds. The main objective of the structurer is thus to design a "AAA" tranche.[8] To mimic the rating score procedure, we consider two key drivers for the rating of a tranche: the ELoT and the WALoT. This is consistent with both previous studies (Ayotte & Gaon, 2011; Chen et al., 2024; He et al., 2012) and industry practice. Clearly, the attractiveness of a tranche decreases with the expected loss but increases with the average life.[9]

The second application takes the point of view of a PFI aiming to maximize the economic impact of the money invested, as motivated in the introduction. For this purpose, a PFI would theoretically finance as many loans as possible. Unfortunately, they are constrained by the capital requirements needed for each loan. The idea behind securitization is to release the capital of the PFI by selling those loans to investors as securitized assets, the PFI keeping only the most junior tranche of the CLO (and therefore capital requirement on this tranche). However, releasing this capital has a cost. We consider an objective function where we minimize the cost of releasing one unit of capital. This objective function is particularly relevant because it describes the trade-off faced by the structurer on the riskiness of the loans to include. By releasing a riskier tranche, the PFI releases more capital but the investor is expecting a higher return and this increases the cost of release. Our objective function therefore encapsulates existing formulations in the literature that optimize risk metrics of tranches. Note that the objective of optimizing the rating of the senior tranche and minimizing the cost of capital release could also be the goal of a private structurer.

5.1. Target rating

As first application, we consider a structurer willing to maximize the attractiveness of the most senior tranche, J. The attractiveness of a tranche is represented via a risk score function (the lower the score, the more attractive the tranche ceteris paribus). Here, we consider a risk score bounded between 0 and a 20 as per the following form for the objective function:

$$F(\boldsymbol{w}) = \min\left(20; \max\left(0; a \times \sqrt{\mathrm{EL}_J(\boldsymbol{w})} - b \times \log(\mathrm{WAL}_J(\boldsymbol{w}))\right)\right) \text{ with } a, b > 0. \quad (14)$$

Since $\mathrm{WAL}_J(\boldsymbol{w}) = \mathrm{wal}_J(\mathrm{WAC}(\boldsymbol{w}), \mathrm{WAM}(\boldsymbol{w}))$, $F(\boldsymbol{w})$ is a deterministic function of $\boldsymbol{w}$ via the ancillary variables $\mathrm{EL}_J(\boldsymbol{w}), \mathrm{WAC}(\boldsymbol{w})$ and $\mathrm{WAM}(\boldsymbol{w})$,

$$\begin{aligned} F(\boldsymbol{w}) &:= f(\mathrm{EL}_J(\boldsymbol{w}), \mathrm{WAM}(\boldsymbol{w}), \mathrm{WAC}(\boldsymbol{w})) \\ &\approx f\left(\widehat{\mathrm{EL}}_J(\boldsymbol{w}), \widehat{\mathrm{WAM}}(\boldsymbol{w}), \widehat{\mathrm{WAC}}(\boldsymbol{w})\right) =: \hat{F}(\boldsymbol{w}) , \end{aligned} \quad (15)$$

where $\widehat{\mathrm{EL}}_J(\boldsymbol{w}), \widehat{\mathrm{WAM}}(\boldsymbol{w}), \widehat{\mathrm{WAC}}(\boldsymbol{w})$ are the linearized version of the corresponding financial variable (see Section 5.3). It is clear from (14) that $F(\boldsymbol{w})$ is increasing in $\mathrm{EL}_J(\boldsymbol{w})$, such that the latter can serve as X_0. One can rely on the approximate optimization problems of Section 5.4 to optimize $F(\boldsymbol{w})$. Note that finding the best functional form to approximate the rating as a function of ELoT and WALoT is out of the scope of this paper. Any rating function that only depends of $\boldsymbol{w}$ via the ELoT and the WALoT (and any other key driver of the rating that could be reasonably linearized) could be optimized using the approximate optimization problems, if the rating function is monotonic in at least one of the drivers.

5.2. Cost of capital release

As a second application, we consider a structurer willing to minimize the cost of releasing one unit of capital. For the sake of simplicity, we consider for this section a setup with two tranches: a junior one (referred to as tranche 1) kept by the issuer, and a senior one (tranche 2) sold to investors. In such a case, the objective function takes the form

$$F(\boldsymbol{w}) = \frac{c_2(\boldsymbol{w})}{\Delta K_2(\boldsymbol{w})} . \quad (16)$$

The denominator corresponds to the release of the regulatory provisions associated with tranche 2 given in (9). The numerator $c_2(\boldsymbol{w})$ corresponds to the annualized cashflows that will be lost as a result of transferring the senior tranche to investors.[10] Relying on expressions relative to the pool notional $N(\boldsymbol{w})$, the cost of releasing tranche 2 is the product of the tranche spread (an annualized rate, $s_2(\boldsymbol{w})$) with the tranche size:

$$c_2(\boldsymbol{w}) = s_2(\boldsymbol{w}) \times S_2 . \quad (17)$$

There is no clear expression for the spread of a tranche as a function of the pool's characteristics. This is due to markets' incompleteness: Each product is unique (portfolios differ) and the cashflows cannot be replicated because the underlying loans are not tradable. This contrasts with synthetic CDOs, where the underlying CDSs can be traded. Since the spread strongly depends on the risk score, we use the same drivers as in the first application: the ELoT and the WALoT. A reasonable parametric form is obtained by assuming that the cashflows collected during the lifetime of the tranche should be an increasing function of its WAL (time-value of money) and EL (credit losses):

$$s_2(\boldsymbol{w}) \times \mathrm{WAL}_2(\boldsymbol{w}) = \alpha \times \mathrm{WAL}_2(\boldsymbol{w}) + \beta \times \mathrm{EL}_2(\boldsymbol{w}) \quad \text{with} \quad \alpha, \beta \geq 0 .$$

The scaling coefficients α, β control the weight of each risk in the spread. This yields the expression for the tranche spread as a linear function of the annualized expected loss:

$$s_2(\boldsymbol{w}) = \alpha + \beta \times \frac{\mathrm{EL}_2(\boldsymbol{w})}{\mathrm{WAL}_2(\boldsymbol{w})} . \quad (18)$$

Note that the spread increases with the ELoT but decreases with the WALoT. Therefore, the latter acts as a penalty ruling out the portfolios consisting of loans bearing low credit risk because of expiring shortly. Clearly, $F(\boldsymbol{w}) = f(\mathrm{EL}_2(\boldsymbol{w}), K(\boldsymbol{w}), \mathrm{WAC}(\boldsymbol{w}), \mathrm{WAM}(\boldsymbol{w}))$ because $\Delta K_2(\boldsymbol{w}) = K(\boldsymbol{w}) - S_1 k_1(K(\boldsymbol{w}))$ and $\mathrm{WAL}_2(\boldsymbol{w})$ is a function of $(\mathrm{WAM}(\boldsymbol{w}), \mathrm{WAC}(\boldsymbol{w}))$. It is obvious from (16), (17) and (18) that $F(\boldsymbol{w})$ is a monotonic function of $\mathrm{EL}_2(\boldsymbol{w})$. One can rely on the approximate optimization problems of Section 5.4 to optimize $F(\boldsymbol{w})$ if one can approximate $\mathrm{EL}_2(\boldsymbol{w})$, $K(\boldsymbol{w})$, $\mathrm{WAC}(\boldsymbol{w})$, $\mathrm{WAM}(\boldsymbol{w})$ by a linearized version $\widehat{\mathrm{EL}}_2(\boldsymbol{w})$, $\hat{K}(\boldsymbol{w})$, $\widehat{\mathrm{WAM}}(\boldsymbol{w})$, $\widehat{\mathrm{WAC}}(\boldsymbol{w})$. This is the purpose of the next section.

[8] Although the loans being securitized therein are mortgages and not SME loans, a typical example of a two-tranches CLO is Belgian-lion-RMBS-I issued by ING Belgium. The folder can be found here https://assets.ing.com/m/45ab9323d4442c8a/original/RMBS-I-Fitch-Presale-Report.pdf. One can read in the "Financial structure" section (p.3) of that document that "The issuer [buys] from the seller the initial portfolio of EUR 5,330,415,230 (the current balance as of the cut-off date), constituting 56,899 loans."

[9] In this setup, we do not include the return of the tranche because we adopt the viewpoint of an investment bank willing to target a specific rating. The corresponding spread will be determined accordingly.

[10] Note that the cashflow on the numerator is the annual premium for bearing the credit risk of tranche 2 while the capital requirement in the denominator aims to cover the unexpected losses only, that is, the losses exceeding $\mathrm{EL}_2(\boldsymbol{w})$. One could reduce the numerator by the annualized ELoT in order to focus on the unexpected loss and related cashflows, but we stick to (16) as it does not change the expression of F.

5.3. Linearization of the main financial variables

Both objective functions introduced above feature $K(\boldsymbol{w})$, WAC($\boldsymbol{w}$), WAM($\boldsymbol{w}$) and $\mathrm{EL}_2(\boldsymbol{w})$ as ancillary variables. Note that one needs to specify which of the Laplace (EL_2), LP (EL_2^∞) or GA-adjusted (EL_2^{GA}) estimation of the ELoT we shall consider in the objective function. We highlight this choice via a superscript in the objective, leading to F, F^∞ and F^{GA}, respectively. We now propose linear approximations for these four ancillary variables. As mentioned in Section 3.1, the ELoT computed using Laplace or its GA estimation are both nonlinear in the weights. The LP expression is more convenient but, still, EL_j^∞ departs from a finite weighted sum of the membership vector $\boldsymbol{w}$ for three reasons: It features an integral, it relies on the min-max operator $[.]$ and the pool notional $N(\boldsymbol{w})$ shows up in the denominator.

To deal with the first problem, we replace the integral in (4) by a finite sum using a quadrature scheme. Quadratures are efficient procedures to estimate the value of an integral using a finite sum of P terms. Some quadrature schemes are tailored for expectations. For instance,

$$\mathbb{E}(h(Z)) = \int_{-\infty}^{\infty} h(x)p_Z(x)dx \approx \sum_{k=1}^{P} \pi_k h(x_k)p_Z(x_k)\ ,$$

where (x_k, π_k) are the P-points quadrature nodes and weights, respectively. The approximation is exact for polynomials of degree $2P-1$ or less. Moreover, the approximation can be made arbitrary good in the sense that the RHS tends to the LHS as $P \to \infty$. Applying this to (4) leads to the LP approximation to the exact $\mathrm{EL}_j(\boldsymbol{w})$, which can be plugged in the objective function to get F^∞:

$$\mathrm{EL}_j(\boldsymbol{w}) \approx \mathrm{EL}_j^\infty(\boldsymbol{w}) = \sum_{k=1}^{P} \pi_k \left[\frac{\sum_{i=1}^{I} w_i N_i \lambda_i p_i(x_k) - A_j N(\boldsymbol{w})}{S_j N(\boldsymbol{w})} \right] p_Z(x_k)\ , \tag{19}$$

where (x_k, π_k) are associated with a P-point quadrature. When $Z \sim \mathcal{N}(0,1)$, for instance, $p_Z = \phi$. The fact that EL_j^∞ is still nonlinear due to the operator $[.]$ is not an issue for optimization purposes because it can be easily circumvented using slack variables.[11] Finally, we replace $N(\boldsymbol{w})$ in the denominator of (3) by the lower bound $\underline{N}$ used in the constraints, which amounts to replace $N(\boldsymbol{w})$ by $\underline{N}$ in (19). The resulting expression will serve as ancillary variable $\hat{X}_0(\boldsymbol{w})$.

We deduce from (6) and (10) that applying the same trick to the denominator of $K(\boldsymbol{w})$, WAC($\boldsymbol{w}$) and WAM($\boldsymbol{w}$) makes them linear in $\boldsymbol{w}$, hence, could serve as ancillary variables, too. We note $\widehat{\mathrm{EL}}_j(\boldsymbol{w})$, $\widehat{\mathrm{WAC}}(\boldsymbol{w})$, $\widehat{\mathrm{WAM}}(\boldsymbol{w})$ and $\hat{K}(\boldsymbol{w})$ the financial variables linearized as explained above.

5.4. Overview of the optimization problems

To build the four different optimization programs, we have used 2 different approximations: (i) linear approximations of the ancillary variables and (ii) clustering loans to reduce the dimensionality and relax the binary constraint. Table 2 summarizes the various programs at hand as well as which assumptions is used in each of them. Observe that Programs 1 and 2 can handle any of the three variants of the objective (F, F^∞ or F^{GA}, indifferently) although the evaluation of the first (exact) objective is much more time-consuming than the last two. In contrast, Programs 3 and 4 deal with F^∞ only, because they require a linearized form; see Section 5.3.

6. Empirical analysis

Let us now assess the performance of the approximate optimization problems proposed in Section 5.4 on either financial applications proposed in Section 5.

[11] Alternative schemes can be considered, dropping the $[.]$ operator and/or relying on quadrature schemes for truncated distributions, but this result in a nonlinear expression in $\boldsymbol{w}$, hence, are not appropriate for our purpose.

Table 2
Optimization programs, related approximations, and variants of the objective function which they can handle.

Pgm	Type	Linearization	Clustering	Objective
1	MINLP	No	No	F, F^∞, F^{GA}
2	NLP + projection	No	Yes	F, F^∞, F^{GA}
3	NLP (MILP at each iteration)	Yes	No	F^∞
4	NLP (LinP + projection at each iteration)	Yes	Yes	F^∞

6.1. Data

Testing empirically the performance of the approximate optimization problems in our context requires to make some assumptions about the credit pool's characteristics and model parameters. The proposed setup is motivated by previous academic research or, for more business-related items such as loans' interest rate or rating functions, based on extensive discussions with major players active in the field of SME loans securitization. As for the pool characteristics, two setups can be adopted: a deterministic or a stochastic approach. Regarding the default probabilities, for example, Glasserman and Ruiz-Mata (2006) sets the marginal PD's in $[0, 0.02]$ using a deterministic data generating process $p_i = 0.01(1 + \sin(16\pi i/I))$ while Gordy (2002) sample from the exponential law, $p_i \sim \mathrm{Exp}(\xi)$ i.i.d. for some scale parameter $\xi > 0$. In this empirical study, we adopt the stochastic approach, which allows us to assess the impact of various parameters on different pools which are statistically equivalent.

We build our heterogeneous pools according to the stochastic model summarized in Table 3. The notional N_i and the maturity M_i of each loan are positive variables that can be generated using i.i.d. samples from a Gamma distribution. Based on discussions with practitioners, we calibrate the parameters of the notional (in units of 5k) distribution to satisfy $\mathbb{E}(N_i) = 5$ and $\mathbb{V}(N_i) = 3.5$. For the maturity (in years), we choose the Gamma parameters to match $\mathbb{E}(M_i) = 5$ and $\mathbb{V}(M_i) = 1$. As for the LGD λ_i, we choose to draw from a Beta distribution to comply with the $[0,1]$ interval. We set the parameters to match the first and second moments reported in previous studies (Caselli et al., 2008; Kaposty et al., 2022). Regarding the marginal default probabilities, we are interested here in two horizons: the probability that the loan defaults before maturity, $p_i = \mathbb{P}(\tau_i \le M_i)$, and the 1-year PD that matters for regulatory capital requirements, $\mathrm{PD}_i = \mathbb{P}(\tau_i \le 1)$. In order to have consistent probabilities bounded in $[0,1]$, we adapt the procedure of Gordy (2002) and draw exponential marginal cumulative probability *curves*, $F_i(t) = \mathbb{P}(\tau_i \le t)$. To this end, we first draw a default rate $\xi_i > 0$ for each loan using i.i.d. Gamma samples, $\xi_i \sim \Gamma(a, b)$, where a and b are chosen to match the first and second moments of previous studies (Andreeva et al., 2016; Kim & Sohn, 2010; Stevenson et al., 2021). Then, we postulate an exponential law with parameter ξ_i for the marginal default distributions,

$$F_i(t) = 1 - e^{-\xi_i t}\ . \tag{20}$$

Then, we simply set $p_i = F_i(M_i)$ and $\mathrm{PD}_i = F_i(1)$. The expected default rate in the pool by a fixed time horizon t is

$$\mathbb{E}\left(\frac{1}{I}\sum_{i=1}^{I} F_i(t)\right) = 1 - \frac{1}{I}\sum_{i=1}^{I} \mathbb{E}(1 - \mathrm{e}^{-\xi_i t}) = 1 - (1 + bt)^{-a}\ ,$$

such that the expected 1-year PD is 3.43 %.[12] The annual rate r_i of loan i is taken as a deterministic function of the maturity M_i and the 1-year relative expected loss $\mathrm{PD}_i \lambda_i$ of the loan:

$$r_i = \frac{50 + 0.8\ M_i + 8000\ \mathrm{PD}_i \lambda_i}{10000}\ . \tag{21}$$

[12] Note that this setup is merely used to set the parameters of the pool, not to actually sample default times.

Table 3
Stochastic model used to generate the parameters of the loans. The notional is given in units of 5k euros and maturity in years. The default times follow an exponential law with parameter ξ_i. The expectations and standard deviations are computed analytically.

Symbol	Variable	Distribution	Mean	Std Dev.
N_i	notional	$\Gamma(2, 2.5)$	5	3.535
λ_i	LGD	$\beta(2, 2)$	0.5	0.224
M_i	maturity	$\Gamma(20, 0.25)$	5	1.118
ξ_i	default rate	$\Gamma(7, 0.005)$	0.035	0.013
p_i	PD at maturity		0.158	0.063
PD_i	PD at 1Y horizon		0.034	0.013
r_i	interest rate		0.019	0.008

In Supplementary Material D, we present histograms and descriptive statistics for the characteristics of the loans in a pool of size $I = 10,000$, and show that they comply with the moments in Table 3.[13]

Having fixed the pool parameters, we can now proceed with the model driving the portfolio losses. Although other choices can be made, we consider for illustration purpose the 1FGC setup, and assume that the expression of the loading factors controlling the dependence between the defaults coincides with the Basel formula of capital requirements (see Supplementary Material B).

With regard to the CLO design, we assume that the structurer wants to select $\underline{N} = 0.75$ of the total outstanding of the loans in the portfolio. For the sake of simplicity, we do not consider any constraint in $\mathcal{W}$ (see Supplementary Material J for a case with sector diversification constraints). For the tranching structure, in line with (Antoniades & Tarashev, 2014), we assume a CLO with 3 tranches: a junior one with attachment and detachment points equal respectively to 0 % and 10 %, a mezzanine tranche with attachment and detachment points of respectively 10 % and 20 % and a senior one with attachment and detachment points of respectively 20 % and 100 %. Since the second application features only two tranches, we merge the mezzanine and the senior tranches to get $A_1 = 0$, $D_1 = A_2 = 0.1$, $D_2 = 1$. In the first application, the parameters a and b are set respectively to 300 and 0.5. In the second application, the parameters α and β are respectively set to 0.0004 and 0.5. With regard to the integral in the ELoT, we rely on a change of variable $x = \tan(\pi/2u)$ combined with the Gauss-Legendre quadrature:

$$\int_{-\infty}^{\infty} h(x)dx = \frac{\pi}{2}\int_{-1}^{1} \frac{h(\tan(\pi/2u))}{\cos^2(\pi/2u)}du \approx \frac{\pi}{2}\sum_{k=1}^{P} \pi_k \frac{h(x_k)}{\cos^2(\tilde{u}_k)}\,, \tag{22}$$

where $\tilde{u}_k = u_k\pi/2$, $x_k = \tan(\tilde{u}_k)$ and the (u_k, π_k) pairs are obtained using `gauss.quad(P,'legendre')` available in the `statmod` library of the statistical software `R`, using $P = 100$ points.[14] A 100 % PSA model is considered to assess the impact of prepayments on the WAL. For the clustering procedure, we use k-means algorithm on the standardized set of all loans characteristics used in the optimization and we report the performance for different number of clusters Q.

6.2. Solutions of the programs and heuristics

In this section, we compare the efficiency of the various methods summarized in Table 2 in terms of the terminal value of the objective function, the similarities in the solution portfolios, and computation time.

We consider two different algorithms to solve Program 1: a Genetic Algorithm (GEN) from `R` package `GA`[15] and a binary Particle Swarm Optimization algorithm (PSO) as in Khanesar et al. (2007).[16] Programs 2, 3 and 4 are solved using Nelder-Mead algorithm from `R` package `nloptr` using default convergence criterion: Stop when an optimization step changes every parameter by less than $1 \times e^{-04}$ multiplied by the absolute value of the parameter. The notional constraint is added to the objective as a penalty in Programs 1 and 2. All MILPs (Program 3) and LPs (Program 4) are solved with Gurobi using default parameters. We compare the solutions of these five programs to those of four heuristics, leading to analyze a total of 9 different algorithms. First, we consider a heuristic (called "Heuristics (EL)" in the sequel) exploiting the connection between the riskiness of a tranche and that of the underlying loans. It consists in selecting the loans in ascending order of their expected losses until all the constraints are met. Alternatively, the priority rule can be determined by sorting the loans in descending order of their maturity, capital requirement or coupon, leading to Heuristics (M), (K) and (r), respectively.

Tables 4 and 5 display a comparison of the algorithms' performance respectively for Application 1 and 2 considering a pool of either 100, 1000 or 10,000 loans. The first 4 lines are heuristics. The next 3 lines show the results for Programs 1 and 2 when the iterations call the exact (Laplace) objective function, F. The next 6 lines show the same but when the objective function being optimized is F^{GA} or F^{∞}. Finally, the last 2 lines refer to the solution of our algorithms (Programs 3 and 4) featuring F^{∞} and linearization, without or with clustering. Columns $F^{\infty,*}$ refer to $F^{\infty}(\boldsymbol{w}^*)$, i.e., show the terminal values of the objective function featuring the LP approximation of the ELoT. Columns $F^{GA,*}$ and F^* are similar but when using the GA approximation or the exact (Laplace) EloT in the objective, respectively. The computation time is capped to 1 h.[17] In Program 4, we fix the number of clusters to $Q = 20$ for the 100 loans dataset, $Q = 200$ for the 1000 loans dataset and $Q = 2,500$ for the 10,000 loans dataset.

Let us first consider Table 4 (Application 1). Observe first the differences between the results in columns $F^{\infty,*}$, $F^{GA,*}$ and F^* for a given pool size, which underpins the impact of considering EL_j^{∞}, EL_j^{GA} or EL_j (Laplace) in the expression of the objective. As expected, the gap is especially large when the pool size I is small, and vanishes as I increases. Second, the GA is efficient in the sense that $F^{GA,*}$ is, in most cases, a better estimation of F^* compared to $F^{\infty,*}$, and comes at no extra computational cost compared to the latter. Not surprisingly, this improvement is mainly relevant when the size of the pool is small or moderately large. Third, it is worth noting that despite these differences, evaluating the relative performance of the algorithms based on $F^{\infty,*}$, $F^{GA,*}$ or F^* leads to similar rankings. Indeed, searching for the program delivering the lowest value of $F^{\infty,*}$, $F^{GA,*}$ or F^* (identified in bold) lead to a same algorithm (here, Program 3 for all considered pool sizes).[18] Fourth, heuristic portfolios perform badly compared to any of the optimization program, except for Heuristics (EL). This underlines the usefulness of solving the optimization problem rather than selecting the "best" loans based on marginal criteria. Fifth, interestingly, the terminal values of the objective function (column F^*) associated with Programs 1 and 2 are better

[13] Of course, many different setups could be considered. Still, it is comparable to credit pools investigated in other studies. Düllmann and Masschelein (2007), for instance, consider a homogeneous pool of $I = 6,000$ SME exposure with 1-year default probability PD = 2 %, an LGD of $\lambda = 45\,\%$ and a notional of $N = 1,000$. In this work, we consider larger PD's and notional amounts in order to better stick to actual figures. In addition, we consider a heterogeneous pool as the loan selection problem is trivial otherwise, all loans being mutually exchangeable.

[14] Note that the `gauss.quad.prob(P,'normal')` command yields the points (x_k, π_k) such that $\mathbb{E}(h(Z)) = \int h(x)\phi(x)dx \approx \sum_{k=1}^{P}\pi_k h(x_k)$ and seems therefore more appropriate to compute integrals with respect to the normal distribution, making the economy of a change of variable and of the evaluation of the $\phi(x_k)$s. However, the Gauss-Legendre scheme proves to be much more efficient when dealing with ELoTs, as noted in Vrins (2009).

[15] Parameters `popSize` and `maxiter` are set to match a desired running time (e.g. 1 h).

[16] Parameter `maxit` is set to match a desired running time (e.g. 1 h).

[17] We use a 12th Gen Intel(R) Core(TM) i7-12700H processor, 2300 MHz, 14 cores, 20 logical processors with a RAM of 16 Go.

[18] Recall that the GA comes at no computational cost compared to LP, but is nonlinear in the weights, hence, cannot be applied to Programs 3 and 4.

Table 4

Application 1. Terminal values of the objective as a function of the method used to compute the ELoT: LP (F^{∞}), GA (F^{GA}) or "exact" (F, Laplace transform). Columns labelled $F^{\infty,*}$ refer to $F^{\infty}(\boldsymbol{w}^*)$, i.e., to the objective where the ELoT at the solution portfolio $\boldsymbol{w}^*$ is computed using the LP approximation. Similarly, columns $F^{GA,*}$ and F^* give $F^{GA}(\boldsymbol{w}^*)$ and $F(\boldsymbol{w}^*)$, the values of the objective at the same final solution $\boldsymbol{w}^*$ when the ELoT is computed using LP + GA or Laplace, respectively. The ELoT approximation associated with the objective that is called by the programs during the iterations are indicated in the left column. Computation time is capped at 1 h.

Algorithm	100 loans			1,000 loans			10,000 loans		
	$F^{\infty,*}$	$F^{GA,*}$	F^*	$F^{\infty,*}$	$F^{GA,*}$	F^*	$F^{\infty,*}$	$F^{GA,*}$	F^*
Heuristics (EL)	1.0174	1.8082	1.8987	2.3103	2.4601	2.4654	2.1534	2.1681	2.1855
Heuristics (M)	6.9262	9.2183	9.5705	9.1633	9.4992	9.5049	8.7901	8.8226	8.8795
Heuristics (K)	10.0900	12.7597	13.0493	14.2832	14.6671	14.6711	14.1319	14.1697	14.2240
Heuristics (r)	9.7924	12.5302	12.8448	14.1423	14.5322	14.5363	14.0525	14.0908	14.1398
Pgm 1 (GEN) on F	2.4478	3.6426	3.8849	6.7578	7.0489	7.0550	7.0079	7.0371	6.9811
Pgm 1 (PSO) on F	2.8065	4.2618	4.5915	6.0255	6.3527	6.3617	7.6836	7.7282	7.7138
Pgm 2 on F	4.1476	5.8782	6.2857	5.4913	5.7365	5.7426	7.1217	7.1541	7.1001
Pgm 1 (GEN) on F^{GA}	0.6448	1.2667	1.2712	4.8916	5.1165	5.1222	6.5646	6.5929	6.5697
Pgm 1 (PSO) on F^{GA}	0.7224	1.3933	1.4475	4.6312	4.8759	4.8828	6.6520	6.6914	6.6644
Pgm 2 on F^{GA}	0.8230	1.5018	1.7211	4.1423	4.3669	4.3736	5.3306	5.3551	5.3927
Pgm 1 (GEN) on F^{∞}	0.6524	1.2780	1.3568	4.6743	4.9017	4.9076	6.4991	6.5268	6.5120
Pgm 1 (PSO) on F^{∞}	0.8804	1.6880	1.8339	4.3707	4.6073	4.6142	6.6520	6.6914	6.6644
Pgm 2 on F^{∞}	0.7830	1.4566	1.6132	4.3354	4.5634	4.5700	5.3440	5.3686	5.4070
Pgm 3 on F^{∞}	**0.6338**	**1.2459**	**1.2308**	**2.0764**	**2.2195**	**2.2247**	**1.8895**	**1.9031**	**1.9303**
Pgm 4 on F^{∞}	0.6903	1.3247	1.3424	2.1072	2.2488	2.2538	1.8913	1.9050	1.9322

when the programs feature F^{∞} or F^{GA} (lines) in the iterations instead of F. This shows that using an approximate expression of the objective function in the iterations can lead to a better solution $\boldsymbol{w}^*$, measured in terms of the *exact* objective function. This can be intuitively explained by the fact that evaluating F^{∞} or F^{GA} is much faster than evaluating F (Laplace or, worse, Monte Carlo). Because Programs 1 and 2 run until the one-hour time out, a much smaller number of iterations can be performed when optimizing F instead of its approximations. Furthermore, recall that despite the values of $F^{\infty}(\boldsymbol{w}), F^{GA}(\boldsymbol{w})$ and $F(\boldsymbol{w})$ may differ for a same pool $\boldsymbol{w}$, the link between them seems to be "comonotonic" (improving $F^{\infty}(\boldsymbol{w})$ or $F^{GA}(\boldsymbol{w})$ seems to enhance $F(\boldsymbol{w})$, too). Sixth, and most importantly, Program 3 and 4, which rely on F^{∞}, deliver the best solutions for all pool sizes. The fact that Program 1 (GEN) on F^{GA} seems competitive when $I = 100$ can be explained by the conjunction of two phenomena. First, the performance of Program 1 is expected to be good because the dimension of the search space is lower and F^{GA} is a cheap but yet reliable estimation of F. At the same time, Programs 3 and 4 rely on "large dimension" approximations and, therefore, do not aim to accommodate small pool sizes. Interestingly enough, though, the best solution is found by Program 3 for all pool sizes, while Program 4 yields comparable results.

Most of these conclusions apply to Application 2 (Table 5). The main difference is that for small pool sizes (100 loans) Program 1 (GEN) iterating based on F^{GA} delivers the best solution, while Program 4 is best in all other configurations (improving slightly the results found by Program 3).

Those results underline the usefulness of relying on both the clustering and the linearization approximations in spite of the fact that they iterate upon an objective function that features the LP approximation of the ELoT. In Supplementary Material E, we report the value of the ancillary variables for solution portfolios. We also show that the value of the objective function before projection and/or using quadrature remains very close to the final value. In Supplementary Material F, we investigate further the similarities of the solution portfolios by looking at the overlap between membership vectors. Program 1 leads to a solution that is relatively different compared to the others. Programs 3 and 4 yield quite similar solutions with an overlap rate higher than 90 %. Complementary figures about the variability of performance across simulation runs can be found in Supplementary Material G. Considering Application 2 and a pool of 1000 loans, numerical results confirm the superiority of Program 4: Not only is it the best in all runs, but it also exhibits the lowest standard deviation. Moreover, the first five places of the ranking, starting from the best algorithm, are Programs 4, 3, 2, 1 (GEN) and 1 (PSO) in this order, for all the simulated pools and regardless of the ELoT approximations used to evaluate the terminal values. The heuristics complete the rankings, in an order that can slightly vary across runs.

Since, in practice, pools of less than 1000 loans are very rare due to the lack of information about the underlying loans combined with the large exposure to idiosyncratic risks, we focus for the rest of the paper on pools of 1000 and 10,000 loans. Moreover, since Programs 1 and 2 deliver better results when applied on F^{∞} than when applied on F^{GA} or F for those pool sizes, we only give the results associated with the former in the rest of the paper.

6.3. Convergence

The results reported in Section 6.2 show that our approximations help increase the quality of the solution. Interestingly, they also help drastically reduce the computation time. Fig. 1 illustrates the convergence speed of the algorithms applied on F^{∞} for Application 1 for a pool featuring 1000 (a) and 10,000 (b) loans. The solution provided by Program 1 (GEN) improves at a very low pace, especially when the pool size is large. Note that first iteration of Program 1 — hence, the time taken to return the first value of the objective function — takes more time ($\approx$ 10 sec) than the ones of Programs 2, 3 and 4 due to the nature of the Genetic Algorithm. Program 2 behaves slightly better in this case, but exhibits longstanding plateaus. The solutions of Programs 3 and 4 are much better than those returned by Programs 1 and 2, as reported in Tables 9 and 10. Additionally, Programs 1 and 2 run up to the one-hour time out, while Programs 3 and 4 converge — hence, stop — much before (in less than 20 sec for 1000 loans and less than 120 sec for 10,000 loans), although Program 4 is about twice as fast as Program 3 when facing large pools. It is worth noting that it is very common for structurers to run a program several times (for instance to analyze the impact of the attachment point of the senior tranche on the objective function). Therefore, cutting the computation time by a factor 30 *while improving the results simultaneously* is a genuine benefit. The results for Application 2 are shown in Fig. 1 (panels c and d) and exhibit a similar pattern. In particular, the quality of the solutions of Programs 3 and 4 are comparable, and much better than that of the solutions of Programs

Table 5
Same as Table 4 but for application 2.

Algorithm	100 loans			1,000 loans			10,000 loans		
	$F^{\infty,*}$	$F^{GA,*}$	F^*	$F^{\infty,*}$	$F^{GA,*}$	F^*	$F^{\infty,*}$	$F^{GA,*}$	F^*
Heuristics (EL)	0.0959	0.1197	0.1208	0.0522	0.0536	0.0535	0.0542	0.0544	0.0543
Heuristics (M)	0.0528	0.0628	0.0625	0.0541	0.0552	0.0552	0.0534	0.0535	0.0535
Heuristics (K)	0.0522	0.0610	0.0606	0.0542	0.0551	0.0550	0.0538	0.0538	0.0539
Heuristics (r)	0.0538	0.0628	0.0624	0.0571	0.0580	0.0580	0.0560	0.0560	0.0561
Pgm 1 (GEN) on F	0.0451	0.0556	0.0555	0.0513	0.0525	0.0525	0.0518	0.0519	0.0519
Pgm 1 (PSO) on F	0.0486	0.0614	0.0614	0.0521	0.0535	0.0535	0.0522	0.0524	0.0524
Pgm 2 on F	0.0520	0.0629	0.0628	0.0486	0.0496	0.0496	0.0517	0.0518	0.0519
Pgm 1 (GEN) on F^{GA}	**0.0433**	**0.0529**	**0.0528**	0.0468	0.0479	0.0479	0.0506	0.0507	0.0507
Pgm 1 (PSO) on F^{GA}	0.0437	0.0538	0.0536	0.0477	0.0489	0.0489	0.0522	0.0523	0.0523
Pgm 2 on F^{GA}	0.0470	0.0574	0.0572	0.0470	0.0482	0.0482	0.0485	0.0486	0.0485
Pgm 1 (GEN) on F^{∞}	0.0433	0.0537	0.0536	0.0471	0.0482	0.0482	0.0505	0.0506	0.0506
Pgm 1 (PSO) on F^{∞}	0.0453	0.0566	0.0564	0.0480	0.0492	0.0492	0.0522	0.0523	0.0523
Pgm 2 on F^{∞}	0.0453	0.0551	0.0550	0.0453	0.0464	0.0464	0.0484	0.0485	0.0484
Pgm 3 on F^{∞}	0.0457	0.0559	0.0560	0.0417	0.0429	0.0428	0.0416	0.0417	0.0416
Pgm 4 on F^{∞}	0.0447	0.0545	0.0544	**0.0414**	**0.0425**	**0.0424**	**0.0412**	**0.0413**	**0.0414**

Table 6
Distribution of the parameters for each stochastic model.

		Low-quality loans					
		High variance			Low variance		
Symbol	Variable	Distribution	Mean	Std Dev.	Distribution	Mean	Std Dev.
λ_i	LGD	$\beta(1.5, 1)$	0.6	0.262	$\beta(3.5, 3)$	0.54	0.182
M_i	maturity	$\Gamma(8, 0.5)$	4	1.412	$\Gamma(40, 0.1)$	4	0.63
ξ_i	default rate	$\Gamma(1, 0.05)$	0.05	0.05	$\Gamma(10, 0.005)$	0.05	0.016
		High-quality loans					
		High variance			Low variance		
Symbol	Variable	Distribution	Mean	Std Dev.	Distribution	Mean	Std Dev.
λ_i	LGD	$\beta(1, 1.25)$	0.44	0.275	$\beta(3, 4)$	0.429	0.175
M_i	maturity	$\Gamma(20, 0.3)$	6	1.341	$\Gamma(60, 0.1)$	6	0.775
ξ_i	default rate	$\Gamma(5, 0.006)$	0.03	0.013	$\Gamma(50, 0.0005)$	0.025	0.003

1 and 2. Furthermore, they also converge much faster, and Program 4 proves to be superior to Program 3 for either pool sizes.

6.4. Sensitivity analysis

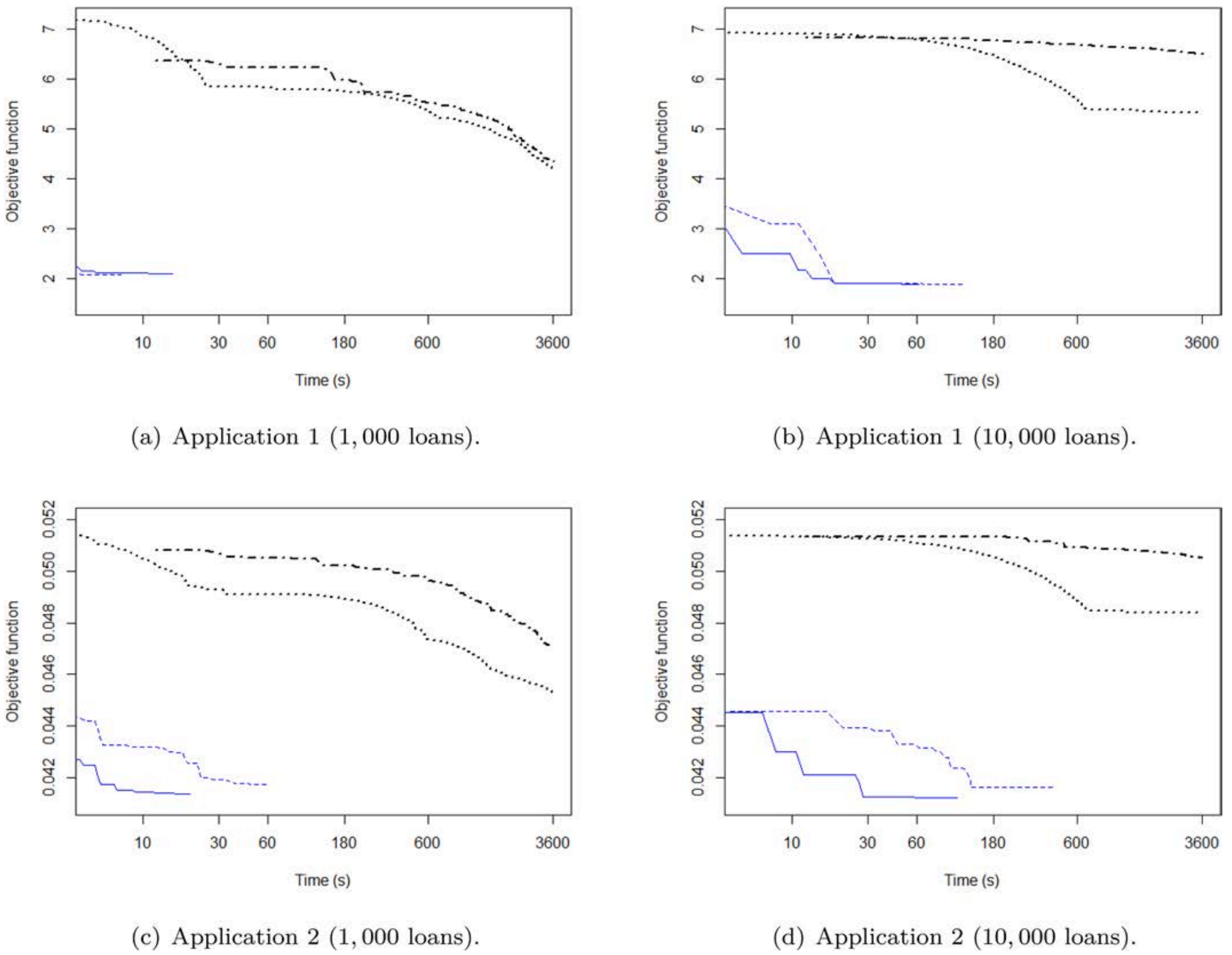
In this section, we stress the parameters of Table 3 by proposing four new stochastic models to generate the parameters of the loans. We consider either *low-quality* loans (compared to the original setup of Table 3) by increasing the average default rate and LGD and by decreasing the average maturity or *high-quality* loans by decreasing the average default rate and LGD and by increasing the average maturity. Moreover, we consider respectively a high variance or a low variance by increasing or decreasing the variance of each distribution compared to the original setup of Table 3. For the sake of conciseness, we do not display the values of p_i, PD_i and r_i; which are functions of the other parameters. The corresponding distributions are given in Table 6.

Table 7 displays the results of each heuristic and each program applied on F^{∞} for each stochastic model, considering both Application 1 and 2 and featuring 1000 or 10,000 loans. As before, we cap the computation time to 1 h and we fix the number of clusters to $Q = 200$ for the 1000 loans dataset and $Q = 2,500$ for the 10,000 loans dataset. We evaluate each terminal value using F^{GA} because it is the best compromise between computation time and accuracy. The results in terms of programs' performance are similar to the ones obtained previously. The best heuristic is different for each configuration which underlines the complexity of the optimization problem. Programs 3 and 4 find the best solution in every configuration. Note that Program 4 finds the best solution in a large majority of configurations and, in the few configurations where Program 3 performs better than Program 4, the two solutions are extremely close (in terms of terminal value of the objective function). This underlines again the usefulness to combine both linearization and clustering.

6.5. Impact of the number of clusters and the clustering algorithm

A central parameter in Programs 2 and 4 is Q, the number of clusters. In this section, we discuss its impact on the quality of the solution. We also assess the impact of the clustering algorithm on the quality of the solution by considering besides k-means three other clustering algorithms: a hierarchical clustering with average linkage, a model-based clustering based on Gaussian mixture models and DBSCAN (density-based spatial clustering of applications with noise). Note that DBSCAN selects automatically the number of clusters and that this number is very low (between 2 and 4 clusters, depending on the configurations) if one considers a standard setup of DBSCAN.[19] In line with the standard use of k-means, we run the clustering algorithm n_k times and select the best within-cluster sum-of-squares run. We set $n_k = 100$, which offers a good

[19] The number of clusters chosen by DBSCAN will be driven by the two required input parameters: *minPts*, the minimum number of points required to form a dense region and ε, the radius of a neighborhood. We follow the recommendations of the DBSCAN authors to determine those two parameters, by choosing ε where there is an elbow in the k-distance graph with $k = minPts - 1$ and by setting $minPts = 2p$ where p is the number of variables used for clustering.



(a) Application 1 (1,000 loans).
(b) Application 1 (10,000 loans).
(c) Application 2 (1,000 loans).
(d) Application 2 (10,000 loans).

Fig. 1. Evolution of the objective function $F^{\infty}(\boldsymbol{w})$ with respect to time for each program in Application 1 (top) and 2 (bottom). Logarithmic scale for x-axis. Dash-dotted (black, top) line for Program 1 (GEN). Dotted (black, top) line for Program 2. Dashed (blue, bottom) line for Program 3. Solid (blue, bottom) line for Program 4. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

compromise between stability and computational time. Other clustering algorithms are deterministic given their inputs.

We draw ten independent credit portfolios of 10,000 loans according to Table 3 (i.e., we have ten different independent copies of a pool of size $I = 10,000$)[20] and report the solution of Program 4 after a running time of 60 s for different number of clusters and different clustering algorithms as a form of boxplots. We compare it to the solution of Program 3 after the same running time. Note that both are applied on F^{∞} and we also use F^{∞} to evaluate the terminal values. The results are shown in Fig. 2. In Application 1 (panel a), increasing Q quickly improves the solution up to $Q = 500$. After this point, the function is rather flat which suggests that the solution is not too sensitive to the number Q of clusters chosen, at least when this number is not too low compared to the pool size I. The optimum is around $Q = 1,000$ and the function is U-shaped; increasing Q further deteriorates slightly the solution, with an average deterioration of 10 % in relative value of the objective function between $Q = 1,000$ and $Q = 10,000$. In Application 2 (panel b), the pattern is similar and the U-shape is more striking but the solution does not critically depend on Q in the sense that Program 4 performs better, on average, than Program 3 (last box) for every Q, and is quite insensitive to Q for reasonable choices: The average deterioration of the objective function in relative value between the optimum and the worst point is 2.5 %. The U-shape of the solution with respect to Q was expected because the approximation is too rough at the extremes. The clustering is too rough when Q is small, such that the projected solution $\boldsymbol{w}^*$ can deviate substantially from $\tilde{\boldsymbol{w}}^*$. When Q is too large, many clusters will contain a single loan, in which case the procedure essentially amounts to relax the binary constraint and then project the solution.

The optimal order of magnitude of the number of clusters is likely to be linked to the number of underlying loans and also, intuitively, to what extent are the loans similar. If we stress the parameters of Table 3 as we did in Section 6.4, we observe empirically that the optimal number of clusters is generally lower for the low-volatility cases, which confirms the initial intuition. For example, the optimal number of clusters for Application 2 considering 10,000 loans is 25 for low-volatility cases and 2500 for high-volatility cases. Although DBSCAN performs much worse, which can be explained by the extremely low number of clusters being selected, the performance of our approach is rather insensitive to the clustering method when the number of clusters is not too low. As a rule of thumb, we recommend choosing a Q between 5 % and 25 % of the pool size. Furthermore, we recommend using k-means in this context.

[20] Results are similar for $I = 1,000$.

Table 7
Terminal values of the objective function found by each heuristic and each program applied on F^{∞} after max. 1 h for each stochastic model, for each application and for each dataset. Terminal values are evaluated using F^{GA}.

Algorithm	Low-quality loans							
	High variance				Low variance			
	1,000 loans		10,000 loans		1,000 loans		10,000 loans	
	App. 1	App. 2	App. 1	App. 2	App. 1	App. 2	App. 1	App. 2
Heuristics (EL)	1.4835	0.0407	1.8364	0.0365	5.6333	0.0669	5.7032	0.0657
Heuristics (M)	10.2020	0.0685	13.2227	0.0757	13.4829	0.0828	13.9514	0.0827
Heuristics (K)	18.2856	0.0773	20.0000	0.0834	18.5864	0.0834	19.5665	0.0845
Heuristics (r)	18.7188	0.0866	20.0000	0.0910	18.5262	0.0880	19.4896	0.0884
Pgm 1 (GEN)	3.4384	0.0461	8.2967	0.0643	8.0738	0.0731	11.2549	0.0773
Pgm 1 (PSO)	5.0872	0.0489	9.5275	0.0693	10.2294	0.0744	11.7656	0.0791
Pgm 2	2.5526	0.0378	6.6394	0.0573	7.5045	0.0728	9.8221	0.0753
Pgm 3	1.2240	**0.0323**	**1.5555**	0.0321	**5.3042**	**0.0640**	**5.4666**	0.0627
Pgm 4	**1.2183**	0.0325	1.5564	**0.0319**	5.3422	**0.0640**	5.4720	**0.0626**
Algorithm	High-quality loans							
	High variance				Low variance			
	1,000 loans		10,000 loans		1,000 loans		10,000 loans	
	App. 1	App. 2	App. 1	App. 2	Appli 1	App. 2	App. 1	App. 2
Heuristics (EL)	0.4313	0.4053	0.2615	1.6153	0.7148	0.1378	0.5442	0.1760
Heuristics (M)	6.6616	0.0474	5.6645	0.0442	3.4836	0.0415	3.2782	0.0408
Heuristics (K)	13.4579	0.0460	12.0468	0.0430	6.4326	0.0366	6.3540	0.0356
Heuristics (r)	13.3356	0.0470	11.8613	0.0441	6.3828	0.0370	6.3032	0.0359
Pgm 1 (GEN)	1.9302	0.0419	3.9170	0.0432	1.8152	0.0389	2.5417	0.0410
Pgm 1 (PSO)	5.2759	0.0435	4.4104	0.0446	4.0135	0.0396	2.8643	0.0424
Pgm 2	0.9689	0.0392	2.8197	0.0404	1.6840	0.0373	1.9994	0.0386
Pgm 3	0.1481	0.0378	0.0871	0.0359	0.8016	0.0402	0.4752	0.0413
Pgm 4	**0.1260**	**0.0365**	**0.0287**	**0.0344**	**0.6413**	**0.0364**	**0.4553**	**0.0349**

First, k-means is extremely fast.[21] Dealing with more greedy algorithms triggers a question: How should the computation time be allocated to handle optimization vs. clustering tasks? Moreover, k-means can feature a large number of cluster Q compared to the number of loans I while some algorithms such as model-based clustering might fail for large Q.[22] Last but not least, k-means is particularly suited in our context because, for the projection step, we select for each cluster the loans being closest from their centroid. This is perfectly aligned with the way k-means assigns a loan to a cluster.

6.6. Impact of the number of tranches

In the considered applications, the number of tranches does not have a substantial impact: In the first application, the goal is to optimize the rating of the most senior tranche without any constraint on the other tranches and, in the second application, the goal is to minimize the cost of capital release for the issuer. In this context, one needs to differentiate between which tranches are kept by the issuer and which tranches are sold to investors. We assumed so far for this application that there were only two tranches, one kept by the issuer and one sold to investors. One could release this assumption by assuming that the numerator becomes the cost of releasing all the tranches that are sold. However, for the first application, one might argue that the issuer should not only be interested in the rating of the most senior tranches but in the rating of all the tranches of the deal. This will lead to a multi-objective optimization. In Supplementary Material H, we consider a CLO of 10 tranches based on Filomeni and Baltas (2023) and we optimize the rating of the most senior tranche while constraining the rating of all the other tranches to a target. Program 4 is the best in terms of value of the objective function. It is also the only one that is able to satisfy all the constraints. This underlines once again that combining both clustering and linearization can be useful.

7. Extensions of the credit loss model

Our approach is illustrated using the one-factor Gaussian copula loss model. This choice is motivated by the fact that the 1FGC loss model is not central in our paper and can be expressed in a concise way. Moreover, in spite of its simplicity, it remains widely used in the industry and for regulation purposes. Nevertheless, it is important to realize that this choice is not a restriction of our approach and can be easily relaxed. For instance, Remark 1 highlights that only the shape of $p_i(z)$ in (5) changes when changing the distribution of the latent variables $Z, \epsilon_1, \ldots, \epsilon_I$. In addition, one could consider other kinds of coupling schemes (copula or number of factors). Moreover, the LGD can also be dependent upon the factors. In this section, we introduce some variants of the 1FGC setup that can be easily tackled in our framework. Some related results are reported in Supplementary Materials. They illustrate that the conclusions drawn in the previous section are not only valid in the 1FGC setup, but hold in those frameworks, too.

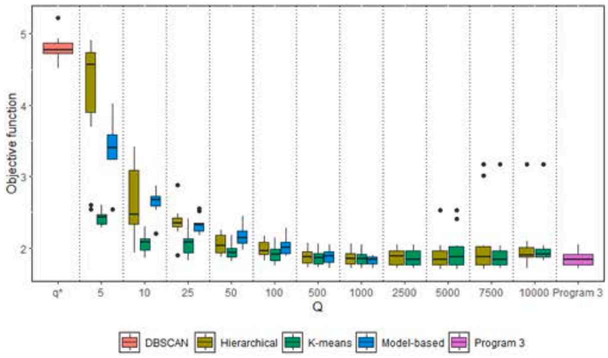
7.1. Non-Gaussian copula

A first extension of the 1FGC approach deals with the type of copula controlling the dependence structure between the default events. For instance, dealing with the LP approximation in a t-copula is not much different. This model is obtained by rescaling the Z_i's in (2) with another systematic component $\sqrt{v/V}$ where $V \sim \chi^2(v)$ is independent from the standard normal variables $Z, \epsilon_1, \ldots, \epsilon_I$. Hence, we have independence between individual losses when conditioning on the pair (Z, V), leading to a 2-factor model. One gets that for $I \to \infty$,

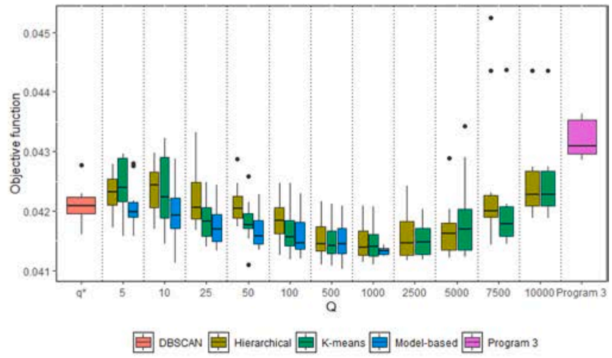
$$L(\boldsymbol{w}) \to \mathbb{E}(L(\boldsymbol{w})|Z, V) = \frac{1}{N(\boldsymbol{w})} \sum_{i=1}^{I} w_i N_i \lambda_i p_i(Z, V)$$

[21] If one consider 10,000 loans and 500 clusters and default settings for each algorithm, K-means takes 0.08 s, hierarchical clustering takes 2.75 s and model-based clustering takes 636 s.

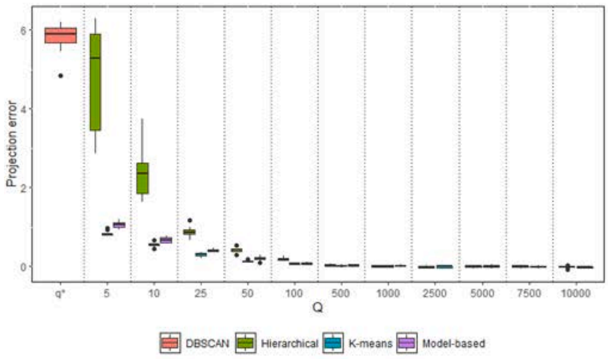
[22] The *mclust* function in R fails when considering $Q = 1.000$ and $I = 10.000$.



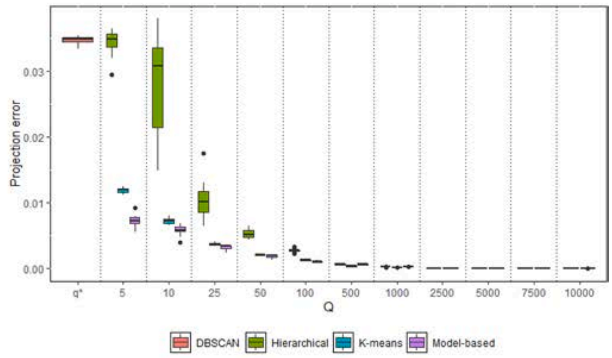
(a) Application 1.



(b) Application 2.



(c) Application 1.



(d) Application 2.

Fig. 2. Distribution of the objective function $F^\infty(\boldsymbol{w}^*)$ (top) and of the projection error $(\hat{F}^\infty(\tilde{\boldsymbol{w}}^*) - F^\infty(\boldsymbol{w}^*))$ in Program 4 as a function of the number Q of clusters estimated by running the algorithm on 10 independent copies of a portfolio of size $I = 10,000$ loans. The last boxplot in the top panels refers to Program 3 (no clustering) on F^∞. In the first boxplot, q^* refers to the number of clusters chosen by DBSCAN that can vary from 2 to 4 depending on the dataset.

$$\textit{where} \quad p_i(Z,V) = \Phi\left(\frac{\sqrt{\frac{V}{\nu}}\Phi_\nu^{-1}(p_i) - a_i Z}{\sqrt{1-a_i^2}}\right)$$

and Φ_ν^{-1} is the quantile function of the standard Student-t distribution with ν degrees of freedom. The integral in (4) is now bi-dimensional (one needs to integrate V away on the top of Z), which remains tractable. Because the Chi-squared belongs to the Gamma family, the integration with respect to V can be achieved using appropriate quadrature schemes. Supplementary Material I displays the solution of each program when a t-copula is considered. The results in terms of programs' performance are similar to the ones obtained previously with the 1FGC.

7.2. Multi-factor models

The loss model can also feature several systematic factors, for example, capturing regional or industry effects. The multi-factor setup corresponds to the case where the default events are all mutually independent conditional upon a set of factors $\mathbf{Z} = (Z_1, \dots, Z_k)^\top$. In this model, Eq. (2) changes to

$$Z_i = a_i Y_i + \sqrt{1-a_i^2}\epsilon_i \text{ where } \quad Y_i = \boldsymbol{b}_i^\top \mathbf{Z} \text{ and } \boldsymbol{b}_i \in \mathbb{R}^k \quad \text{satisfies } \|\boldsymbol{b}_i\| = 1. \tag{23}$$

In this case, the LP approximation yields

$$L(\boldsymbol{w}) \to \mathbb{E}(L(\boldsymbol{w})|\mathbf{Z}) = \frac{1}{N(\boldsymbol{w})}\sum_{i=1}^{I} w_i N_i \lambda_i p_i(\mathbf{Z}) \quad \text{as} \qquad I \to \infty\ ,$$

where $p_i(\mathbf{Z})$ is the conditional PD of i given $\mathbf{Z}$. This shows that extending our setup to multiple and arbitrary factors is straightforward; the only price to pay is that the dimension of the integral in (4) becomes multidimensional. This is not an obstacle to using our quadrature approximation as long as the number k of factors and the number of quadrature points are not too large. Note that this setup can be combined with the t-copula, it suffices to scale the Z_i's in (23) by $\sqrt{\nu/V}$. One could also proxy this multi-factor model using a *comparable single-factor* (CSF) model following (Pykhtin, 2004). Supplementary Material J illustrates the performance of our approach in such a case using evidence from Hahnenstein (2004), and includes diversification constraints for each sector. The conclusions are similar to those obtained with the 1FGC. In particular, optimizing the objective function associated with the CSF model using Programs 3 and 4 leads to a substantial improvement compared to running Programs 1 and 2. Note that it is hardly possible to apply Programs 1 and 2 to the exact EL_j computed using the multi-factor model, as each evaluation of the latter is extremely time-consuming.

7.3. PD-LGD dependency

Finally, our setup can also accommodate stochastic LGDs (Λ_i) provided that the default events and the losses all remain mutually independent conditionally upon the systematic factor, Z. For any such coupling scheme, it suffices to replace λ_i in (3) by $\lambda_i(Z)$ where $\lambda_i(Z) := \mathbb{E}(\Lambda_i | B_i = 1, Z)$ and the linearity in $\boldsymbol{w}$ is preserved. See Barbagli and Vrins (2023) for more details and some tractable examples. This also applies when considering a multi-factor setup and/or the t-copula (conditioning with respect to $\mathbf{Z}$, (Z, V) or $(\mathbf{Z}, V)$, depending on the approach chosen).

8. Conclusion

SMEs account for an important part of wealth creation in modern economies. Yet, they are typically riskier than large firms and, consequently, face more difficulties to get financed. In an era of costly energy, high inflation, and scarce public investment capacities, reducing the funding costs of SMEs is therefore more critical than ever.

Among the various strategies put in force by public financial institutions (PFIs), securitization proves to be particularly efficient for this purpose. Thanks to the credit enhancement mechanism inherent to collateralized loan oblogations (CLO), transferring SME loans from commercial credit institutions to private investors while the junior tranche is held by PFIs is a way to reduce the requirements regarding both interest rates and collateral. On the other hand, these loans are removed from the balance sheet of commercial banks, leading to capital release and, consequently, increasing the available financing volumes. Finally, these tranches form a new asset class with various risk-return profiles, enabling fund managers to build more efficient portfolios.

A key step in the securitization process is the identification of which loans to embed in a CLO. Indeed, the loan selection problem is a high-dimensional mixed-integer nonlinear optimization program (NP-hard) featuring complex objective functions whose evaluation relies on costly numerical methods such as heavy Monte Carlo simulations and whose derivative is typically not available in closed form. In this paper, we circumvent this challenge by relying on clustering and linearization. Specifically, we derive approximations of the original problem which are lower dimensional, relax the binary constraints and tackle nonlinearity issues. Our approach can accommodate pools made of several thousands of loans, various forms of objective functions and different loss models. Our simulation studies feature realistic pools and show that the proposed approach delivers a solution leading to an improved value of the objective function associated with the exact criterion. Moreover, it does so in a lower computation time compared to standard derivative-free mixed-integer algorithms in every tested configuration.

The traditional approach to select loans to embed in a given CLO is to identify a few portfolios by relying on heuristics or running brute force algorithms, compare the corresponding values of the objective function, and select the best solution. The philosophy of the approach we propose is to suggest the structurer to also involve, in his comparison, portfolios solving approximate problems such as Programs 3 and/or Programs 4. Indeed, our numerical results indicate that those solution portfolios are easy to find and might significantly outperform those found with the traditional approach in terms of the actual waterfall.

Our contribution is useful for any stakeholder involved in the selection step of the loans securitization process, including PFIs such as development and central banks, as well as the structuring teams of commercial banks. Moreover, by enhancing the securitization procedure, our research also benefits to public authorities, private investors such as pension funds, as well as to the SME sector as a whole.

CRediT authorship contribution statement

Arnaud Germain: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Conceptualization; **Frédéric Vrins:** Writing – review & editing, Writing – original draft, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

None.

Acknowledgment

The work of Arnaud Germain is funded by the ING chair. Frédéric Vrins benefits from the financial support of the Fonds de la Recherche Scientifique F.S.R.-FNRS (grant J.0225.24), the Belgian Federal Science Policy Office (grant ARC 18-23/089) as well as of the ING chair. The authors warmly thank Aurélien Mouly, Lilan Lecomte and Jacques Makutonin (European Investment Fund) as well as Guillermo Navas and Dylan Cissou (Morningstar DBRS) for interesting discussions about this project and their guidance regarding the design of the objective functions and empirical setup. The authors are grateful to Erick Delage (HEC Montréal), Laurence Wolsey (CORE, UCLouvain), Georges Hübner (HEC Liège), the participants of AFMath24 (Belgium), ICCF24 (Amsterdam), AFFI2025 (Dijon) and FEBS2025 (Montpellier) whose constructive comments and feedback contributed to improve preliminary versions. This paper builds upon the master thesis (Jonard, 2021) supervised by F. Vrins, which received the second prize of the 2021 CFA Quant Awards held in Amsterdam.

Supplementary material

Supplementary material associated with this article can be found in the online version, at 10.1016/j.ejor.2025.11.015

References

Andreeva, G., Calabrese, R., & Osmetti, S. A. (2016). A comparative analysis of the UK and italian small businesses using generalised extreme value models. *European Journal of Operational Research*, *249*(2), 506–516.

Antoniades, A., & Tarashev, N. A. (2014). Securitisations: Tranching concentrates uncertainty. *BIS Quarterly Review December*, 37–53.

Ayotte, K., & Gaon, S. (2011). Asset-backed securities: Costs and benefits of "bankruptcy remoteness". *The Review of Financial Studies*, *24*(4), 1299–1335.

Barbagli, M., & Vrins, F. (2023). Accounting for PD-LGD dependency: A tractable extension to the basel ASRF framework. *Economic Modelling*, *125*, 106321.

Barbagli, M., & Vrins, F. (2025). Efficient Monte Carlo estimation of credit concentration risk. Technical Report LFIN DP-2025/03 UCLouvain.

BCBS (2019). Calculation of RWA for credit risk - Securitisation: Standardised approach. Technical Report Bank for International Settlements.

BCBS (2023). Calculation of RWA for credit risk - IRB approach: Risk weight functions. Technical Report Bank for International Settlements.

Bennett, P. (1984). Applying portfolio theory to global bank lending. *Journal of Banking & Finance*, *8*(2), 153–169.

Bo, L., & Capponi, A. (2014). Optimal investment in credit derivatives portfolio under contagion risk. *Mathematical Finance*, *26*(4), 785–834.

Caselli, S., Gatti, S., & Querci, F. (2008). The sensitivity of the loss given default rate to systematic risk: New empirical evidence on bank loans. *Journal of Financial Services Research*, *34*, 1–34.

Chen, X., Cheng, Q., & Luo, T. (2024). The economic value of blockchain applications: Early evidence from asset-backed securities. *Management Science*, *70*(1), 439–463.

Düllmann, K., & Masschelein, N. (2007). A tractable model to measure sector concentration risk in credit portfolios. *Journal of Financial Services Research*, *32*(1-2), 55–79. https://doi.org/10.1007/s10693-007-0014-3

Egloff, D., & Leippold, M. (2010). Quantile estimation with adaptive importance sampling. *The Annals of Statistics*, *38*(2), 1244–1278. https://doi.org/10.1214/09-AOS745

European Commission (2023). SME and High inflation (summary report). Technical Report Directorate-General for Internal market, Industry, Entrepreneurship and SMEs.

Filomeni, S., & Baltas, K. (2023). Senior-subordinated structure: buffer or signal in securitisation?. *The European Journal of Finance*, *29*(3), 329–362.

Giesecke, K. (2008). Portfolio credit risk: Top-down versus bottom-up approaches. In *Frontiers in quantitative finance* chapter 10. (pp. 251–267). John Wiley & Sons, Ltd. https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118266915.ch10. https://doi.org/10.1002/9781118266915.ch10

Giesecke, K., Goldberg, L., & Ding, X. (2011). A top-down approach to multi-name credit. *Operations Research*, *59*(2), 283–300.

Glasserman, P., & Ruiz-Mata, J. (2006). Computing the credit loss distribution in the gaussian copula model: A comparison of methods. *Journal of Credit Risk*, *2*(4), 33–66.

Gordy, M. (2002). Saddlepoint approximation of creditrisk+. *Journal of Banking & Finance*, *26*, 1335–1353.

Gordy, M. B. (2003). A risk-factor model foundation for ratings-based bank capital rules. *Journal of Financial Intermediation*, *12*(3), 199–232.

Gordy, M. B., & Lutkebohmert, E. (2013). Granularity adjustment for regulatory capital assessment. *International Journal of Central Banking*, *9*(3), 33–71. https://www.ijcb.org/journal/ijcb13q3a2.htm.

Hahnenstein, L. (2004). Calibrating the CreditMetrics (TM) correlation concept–Empirical evidence from Germany. *Financial Markets and Portfolio Management*, *18*(4), 358–381.

He, J., Qian, J., & Strahan, P. E. (2012). Are all ratings created equal? the impact of issuer size on the pricing of mortgage-backed securities. *The Journal of Finance* *67*(6), 2097–2137.

Huang, D., Kai, Y., Fabozzi, F. J., & Fukushima, M. (2007). An optimal design of collateralized mortgage obligation with PAC-companion structure using dynamic cash reserve. *European Journal of Operational Research*, *177*(2), 1134–1152.

Jarrow, R., & Van Deventer, D. R. (2023). A bottom-up, reduced form credit risk model approach for the determination of collateralised loan obligation capital. *Journal of Risk Management in Financial Institutions*, *16*(3), 237–255.

Jewan, D., Guo, R., & Witten, G. (2009). Optimal bespoke CDO design via NSGA-II. *Journal of Applied Mathematics and Decision Sciences*, 925169. https://doi.org/10.1155/2009/925169

Jonard, M. (2021). Selection of securities for Collateralized Debt Obligations contracts: An optimization approach applied to European SME loan portfolios. CEMS Master in International Management and Business Engineering UCLouvain. https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119606475. https://doi.org/10.1002/9781119606475

Kaposty, F., Klein, P., Löderbusch, M., & Pfingsten, A. (2022). Loss given default in SME leasing. *Review of Managerial Science*, *16*(5), 1561–1597.

Khanesar, M. A., Teshnehlab, M., & Shoorehdeli, M. A. (2007). A novel binary particle swarm optimization. In *2007 Mediterranean conference on control & automation* (pp. 1–6). IEEE.

Kim, H. S., & Sohn, S. Y. (2010). Support vector machines for default prediction of SMEs based on technology credit. *European Journal of Operational Research*, *201*(3), 838–846.

Laurent, J.-P., & Sestier, M. (2016). Trading book and credit risk: How fundamental is the Basel review ? *Journal of Banking & Finance*, *73*, 211–223.

OECD (2023). OECD financing SMEs and entrepreneurs scoreboard: 2023 highlights. (36). https://www.oecd-ilibrary.org/content/paper/a8d13e55-en. https://doi.org/10.1787/a8d13e55-en

Owens, J., & Wilhelm, L. (2017). Alternative data transforming SME finance. Technical Report World Bank. http://documents.worldbank.org/curated/en/701331497329509915/Alternative-data-transforming-SME-finance.

Petitt, B., Pinto, J., & Pirie, W. (2015). Fixed income analysis. CFA Institute Investment Series (third edition ed.). Wiley.

Pykhtin, M. (2004). Multi-factor adjustment. *Risk*, *17*, 85–90.

Saunders, D., Xiouros, C., & Zenios, S. (2007). Credit risk optimization using factor models. *Annals of Operations Research*, *152*, 49–77.

Sirignano, J. A., Tsoukalas, G., & Giesecke, K. (2016). Large-scale loan portfolio selection. *Operations Research*, *64*(6), 1239–1255.

Stevenson, M., Mues, C., & Bravo, C. (2021). The value of text for small business default prediction: A deep learning approach. *European Journal of Operational Research*, *295*(2), 758–771.

Vasicek, O. (1991). Limiting loan loss probability distribution. Technical Report.

Veremyev, A., Tsyurmasto, P., & Uryasev, S. (2012). Optimal structuring of CDO contracts: Optimization approach. *Journal of Credit Risk*, *8*(4), 133–155.

Veronesi, P. (2010). Fixed income securities: Valuation, risk and risk management. Wiley.

Vrins, F. (2009). Double t copula pricing of structured credit products: Practical aspects of a trustworthy implementation. *Journal of Credit Risk*, *5*(3), 91–109.

Westerfeld, S., & Weber, F. (2009). Selecting credit portfolios for collateralized loan obligation transactions: A heuristic algorithm. *Journal of Credit Risk*, *5*(4), 65.

Wolsey, L. (2020). Integer programming. John Wiley & Sons, Ltd. https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119606475. https://doi.org/10.1002/9781119606475