Recursive Formulas for the Default Probability Distribution of a Heterogeneous Group of Defaultable Entities

Daniel W.-C. Miao and B. M. Hambly

May 24, 2010

Abstract

The probability distribution of the number of defaults plays an important role in the multiple name credit derivatives pricing problems. When the group size becomes large, it is increasingly difficult to obtain its whole distribution. We base on a financial argument to prove that for these default probabilities, there exists a recursive formula which is useful for the calculation of the whole distribution. A major merit of the proposed formula is that it is model free and allows for general correlation structure among group entities. For a fully heterogeneous group, the use of the proposed formula is somewhat limited because of its high computational complexity. To remedy this difficulty, we first look at the homogeneous case where the formula is reduced to a simple version that is particularly feasible for computation. Based on this simple version, we then incorporate heterogeneity by extending the recursive formula for more general cases where there are multiple subgroups of homogeneous entities. Through numerical examples for a class of Markov chain based reduced-form models, we demonstrate the computational benefits.

Key words: default probability, multiple name credit derivatives, recursive algorithms, reduced-form models, Markov chains, Markov-modulated Poisson processes.

1 Introduction

The probability distribution of N_t^n, the number of defaults in a size n group at time t, is central to the pricing of credit derivatives involving multiple names, such as credit default swaps (CDSs) and collateralized debt obligations (CDOs). The distribution typically has a complicated structure as the default correlation between entities has a major impact on these probabilities. As the group size increases the computation of the whole distribution becomes challenging and has received much attention in the literature (see for example, Schonbucher (2000), Vasicek (2002), Frey and McNeil (2003), Schonbucher (2003), Merino and Nyfeler (2002), Andersen et al. (2003), Mortensen (2006)).

Some earlier attempts to obtain the distribution were based on the assumption that the entities are homogeneous and independent, and then the number of defaults follows a binomial distribution,

$$\mathsf{P}(N_t^n = m) = \binom{n}{m} p^m (1-p)^{n-m},$$

where p is the individual default probability. It is important to introduce correlation into this framework and there are a number of ways of doing this. For example, the binomial expansion technique (BET) is used by the ratings agency Moody's to approximate a group of correlated entities by a smaller group of independent entities, as discussed in Schonbucher (2003). Another way is to use a conditional independence model through the introduction of a common factor Y that is linked to each individual default probability p. Under the assumption that the defaults are independent conditional on the realization of $Y = y$, we have

$$\mathsf{P}(N_t^n = m) = \mathsf{E}[\binom{n}{m} p(Y)^m (1 - p(Y))^{n-m}],$$

where the expectation is taken over the factor Y.

The binomial framework provides a simple model for the distribution of the number of defaults. However, if the group becomes heterogeneous we can no longer use the binomial distribution. Andersen et al. (2003) provide a recursive algorithm for the default probabilities calculation when the defaults between the heterogeneous entities are independent. Mortensen (2006) extends this method by introducing common factors to reflect their mutual dependence. In fact, the use of common factors becomes popular as they avoid specifying how each pair of entities are correlated and instead model the correlation by a number of centralized quantities that affect all entities. This makes factor loadings well suited in a variety of default risk models (for example, the Copula models, as seen in Andersen and Sidenius (2004-5), Hull and White (2004)). However, it is inevitable that the complication from a large heterogeneous group will cause high computation load, especially when the objective is to calculate the whole distribution of N_t^n.

In the following we consider this problem from the viewpoint of reduced-form models. Suppose we are given n heterogeneous entities with default rate processes $\{\lambda_t^1\}, \cdots, \{\lambda_t^n\}$ and their own default times $\tau_1, \cdots, \tau_n$. Let $\tau_{(1)}$ be the first default time, i.e. $\tau_{(1)} = \min_{i=1\cdots n} \tau_i$ and write $\tau_{(m)}$ for the time of the m-th default. Assuming that there are no joint defaults and the default rates are conditionally independent, the first default time $\tau_{(1)}$ is the time of the first event in a Poisson process of rate $\lambda_t^{(1)} = \sum_{i=1}^n \lambda_t^i$.

The whole default process can be considered as a pure death process $\{N_t^n\}$. Let $\lambda_t^{(m)}$ denote the m-th default rate that triggers $\tau_{(m)}$, which is simply the sum of the default rates of the surviving entities at t, i.e.

$$\lambda_t^{(m)}(S^{(m-1)}) = \sum_{j \in S^{(m-1)}} \lambda_t^j,$$

where $S^{(m-1)}$ is the set of entities surviving the $(m-1)$-th default and subject to the m-th default. Note that here $\lambda_t^{(m)}$ is well defined only when $N_t^n = m - 1$ and can be clearly specified only when $S^{(m-1)}$ is known. For convenience, we would like to consider $\{\lambda_t^{(m)}\}$ as a stochastic process which is well defined for all $t \geq 0$. This can be achieved by letting $\lambda_t^{(m)} = 0$ outside the range $\tau_{(m-1)} \leq t < \tau_{(m)}$. Consequently, the m-th default survival probability can be expressed by

$$\mathsf{P}(\tau_{(m)} > t) = \mathsf{E}[e^{-\int_0^t \lambda_s^{(m)} ds}], \quad m = 1, 2, \cdots, n.$$

The probability distribution $\mathsf{P}(N_t^n = m)$ or $\mathsf{P}(N_t^n \leq m)$, $m = 0, 1, \cdots, n$ can be recovered from the survival probabilities as

$$\mathsf{P}(N_t^n \leq m) = \mathsf{P}(\tau_{(m+1)} > t), \quad m = 0, 1, \cdots, n.$$

Consider the simplest case where the n entities are homogeneous and each default rate follows a Poisson process with rate λ, then $\lambda_t^{(m)}$ can be specified as

$$\lambda_t^{(m)} = (n - m + 1)\lambda, \quad \tau_{(m-1)} \leq t < \tau_{(m)},$$

as $\lambda_t^{(m)}$ is determined only by the number of surviving entities. In the general case, $\lambda_t^{(m)}$ requires knowledge about which entities survive during $\tau_{(m-1)} \leq t < \tau_{(m)}$. A clear specification of $\lambda_t^{(m)}$ can be obtained by looking at the default history in the death process $\{N_t^n\}$. If the death process $\{N_t^n\}$ is Markovian (for instance, if each λ_t^i follows a 2-state Markov chain with possibly dependent transition rates, then $\{N_t^n\}$ can be formulated as a large Markov chain), in theory the distribution of $N_t^n, \forall t \geq 0$ can be obtained by solving such a pure death Markov chain with $\lambda_t^{(m)}$ properly defined for each state. However, this approach is viable only when n is not too large and the individual default rate process is a simple enough. In general the whole death tree grows exponentially and it is not computationally feasible to determine the whole distribution in this way (though it is usually possible to compute $\mathsf{P}(N_t^n = m)$ for smaller m, e.g. $m = 2, 3$, by solving a truncated part of the full death chain).

The difficulty of computing $\mathsf{P}(N_t^n = m)$ for general n, m can be also seen from the following formula where we note that the concerned probability is calculated by conditioning on $\tau_{(1)} = s_1 < \cdots < \tau_{(m)} = s_m < t < \tau_{(m+1)}$, and can be expressed in terms of $\lambda_t^{(1)}, \cdots, \lambda_t^{(m+1)}$ as below:

$$\begin{aligned}\mathsf{P}(N_t^n = m) \;=\; \mathsf{P}(\tau_{(m)} \leq t < \tau_{(m+1)}) = \int_0^t \int_{s_1}^t \cdots \int_{s_{m-1}}^t \mathsf{E}\Big[& e^{-\int_0^{s_1} \lambda_u^{(1)} du} \cdot \lambda_{s_1}^{(1)} \cdot e^{-\int_{s_1}^{s_2} \lambda_u^{(2)} du} \cdot \lambda_{s_2}^{(2)} \cdots \\ & \cdots e^{-\int_{s_{m-1}}^{s_m} \lambda_u^{(m)} du} \cdot \lambda_{s_m}^{(m)} \cdot e^{-\int_{s_m}^{t} \lambda_u^{(m+1)} du} \Big] ds_m \cdots ds_2 ds_1.\end{aligned}$$

The above expectation is taken over the filtration which contains the paths of all the default rates $\lambda_t^{(1)}, \cdots, \lambda_t^{(m+1)}$. The formula can not be directly calculated unless n and m are small as the interdependence between these $\lambda_t^{(i)}, i = 1, \cdots, m + 1$ further complicates the calculation when n and m is large.

In this paper, we show that there actually exists a recursive structure in the probabilities $\mathsf{P}(N_t^n \leq m)$, which is helpful in computing the whole probability distribution of N_t^n. This work is inspired by the recursive formula for the basket credit default swap (CDS) prices studied by Huge (2001) and Lando (2004). Our treatment starts by defining an essentially different financial contract which has a direct connection to the probabilities concerned. The price of this contract is proved to have the same recursive structure as the basket CDS prices and consequently, we come to the recursive structure in the probabilities $\mathsf{P}(N_t^n \leq m)$ by using their connection with the contract prices.

With proper initial conditions given, the recursive formula can be used to calculate these default probabilities. However, the use of this formula actually involves the default probability distribution of every possible subset of the original group of entities. When the group size becomes large (e.g. 50), the number of subsets can become unmanageable and the method still breaks down. In order to achieve large group size, we look at the special homogeneous case in which the recursive formula is reduced to a much simpler form and becomes computationally feasible. Based on the homogeneous case (1-G), we then consider the more general cases where there are two (2-G), three (3-G), or more (k-G) homogeneous subgroups to allow for different form of heterogeneity. Again we are able to develop a recursive formula for the default probabilities in these partly heterogeneous cases.

The effectiveness of the recursive formulas are investigated in the numerical studies, where we implement the proposed 1-G and 2-G algorithms for a class of Markov chain based reduced-form models. These models are chosen such that we can solve the whole pure death Markov chain to obtain the desired probabilities $\mathsf{P}(N_t^n \leq m)$ or $\mathsf{P}(N_t^{n_1} \leq m_1, N_t^{n_2} \leq m_2)$ with high precision, which are used as a benchmark to compare with the results from the proposed recursive algorithms. Through the numerical examples we look at the saving in computation time as well as the accuracy of the proposed methods, with some emphasis on the avoidance of the propagation error that is caused by the recursive formulas.

The rest of this paper is organized as follows. In Section 2 we derive the recursive formula for $\mathsf{P}(N_t^n \leq m)$ in the general heterogeneous case, and look at the its reduction to the special homogeneous case. Section 3 extends the homogeneous results to more general cases where there are multiple groups of homogeneous entities. In Section 4 we implement the proposed algorithms for a class of Markov chain models and discuss the numerical examples. Finally the conclusions are given in Section 5.

2 Recursive formulas for $P(N_t^n \leq m)$

Given a group of n defaultable entities, consider a first-m-of-n-to-default CDS, which gives a payoff for the first m default events at $\tau_{(1)}, \cdots, \tau_{(m)}$ if these defaults happen prior to maturity time T. It is shown in Huge (2001) and Lando (2004) that, based on an arbitrage argument, the time t price of such a contract, denoted $U^{m,n}(t)$, satisfies the following recursive formula

$$U^{m,n}(t) = \frac{1}{m-1}\left[\sum_{k=1}^{n} U_k^{m-1,n-1}(t) - (n-m)U^{m-1,n}(t)\right], \tag{1}$$

where $n \geq 2$, $m \geq 2$, $0 \leq t \leq T$. In the above formula, $U_k^{m-1,n-1}(t)$ represents the price of the first-$(m-1)$-of-$(n-1)$-to-default CDS in which the new size $(n-1)$ group is formed by excluding entity k from the original size n group. The formula indicates that, for a fixed integer $\bar{n}$, as long as all $U^{1,n}(t)$, $1 \leq n \leq \bar{n}$ are given, then all the $U^{m,n}(t)$, $1 \leq m \leq n, 1 \leq n \leq \bar{n}$ can be obtained recursively from this formula.

Note that there is no clear connection between (1) and the default probabilities concerned, i.e. $\mathsf{P}(N_t^n \le m)$, as the payoffs are given at several default epoches before maturity[1]. Here our aim is to develop a recursive formula for $\mathsf{P}(N_t^n \le m)$. To this end we consider a similar but essentially different contract $C(m,n)$ which gives payoff at t as below:

$$\text{The } t\text{-payoff of } C(m,n) = \begin{cases} m, & \text{if } N_t^n \le m, \\ 0, & \text{if } N_t^n > m. \end{cases}$$

Note that in the preceding basket CDS contract, the actual total payoff depends on the number of defaults (up to m defaults are covered), and the payoff is given at each random default times $\tau_{(i)}, i \le m$. Here in our contract $C(m,n)$ the payoff is always m regardless of the real number of defaults, as long as it is no greater than m. In addition, the payoff is only given at the prespecified maturity t instead of the random default times.

Let $V^{m,n}(t)$ denote the current (time 0) price of the contract $C(m,n)$. (Note that here (t) refers to the maturity of $C(m,n)$, unlike in $U^{m,n}(t)$ where (t) refers to the time t price of the basket CDS maturing at T.) Similarly, we also define $V_k^{m-1,n-1}(t)$ as the current price of the contract $C_k(m-1,n-1)$ in which the size $(n-1)$ group does not include entity k. We show that $V^{m,n}(t)$ exhibits the same recursive structure as Huge's formula (1).

Theorem 2.1. *The price $V^{m,n}(t)$ of the payoff $C(m,n)$ satisfies the following recursive formula:*

$$V^{m,n}(t) = \frac{1}{m-1}\left[\sum_{k=1}^{n} V_k^{m-1,n-1}(t) - (n-m)V^{m-1,n}(t)\right], \tag{2}$$

where $n \ge 2$, $m \ge 2$, $t \ge 0$.

Proof. Proof. We just check that the values of the claims corresponding to both sides are worth the same in each possible case. Obviously, these claims are

$$\text{LHS claim}: \quad C(m,n), \quad \text{and}$$

$$\text{RHS claim}: \quad \frac{1}{m-1}\left[\sum_{k=1}^{n} C_k(m-1,n-1) - (n-m)C(m-1,n)\right].$$

There are three possible cases:

1. $N_t^n > m$: Both the LHS and RHS claims pay nothing.
2. $N_t^n = m$: The LHS claim pays m. In the RHS claim, $C(m-1,n)$ pays nothing. As to $C_k(m-1,n-1)$, it can be divided into two cases:
   (a) The excluded entity k has defaulted (with probability $\frac{m}{n}$): the remaining set has $(m-1)$ defaulted entities, thus $C_k(m-1,n-1)$ pays $(m-1)$.
   (b) The excluded entity k is a surviving one (with probability $\frac{n-m}{n}$): the remaining set has m defaulted ones, thus $C_k(m-1,n-1)$ pays nothing.

As a consequence, the value of the RHS claim is

$$\frac{1}{m-1}\left[\, n\cdot\frac{m}{n}\cdot(m-1)-0\,\right]=m.$$

3. $N_t^n < m$: The LHS claim pays m. In the RHS claim, both $C_k(m-1,n-1)$ and $C(m-1,n)$ pay $(m-1)$, thus we have

$$\frac{1}{m-1}\left[\, n\cdot(m-1)-(n-m)\cdot(m-1)\,\right]=m.$$

Therefore the proof is completed. □

Since the contract $C(m,n)$ pays off according to whether or not $N_t^n \le m$ is true, the probability $\mathsf{P}(N_t^n \le m)$ will have a similar recursive structure to $V^{m,n}(t)$. Let $N_{k,t}^{n-1}$ be the number of defaults in an $(n-1)$-entity group with entity k excluded. The result is stated as below.

Theorem 2.2. $\mathsf{P}(N_t^n \le m)$ *satisfies the following recursive formula:*

$$\mathsf{P}(N_t^n \le m)=\frac{1}{m}\left[\sum_{k=1}^{n}\mathsf{P}(N_{k,t}^{n-1}\le m-1)-(n-m)\mathsf{P}(N_t^n\le m-1)\right], \tag{3}$$

where $n\ge 2$, $m\ge 1$, $1\le k\le n$, $t\ge 0$.

Proof. Proof. According to the definition of the contract, the following pricing formulas hold (thinking of P as the risk-neutral probability measure):

$$\begin{aligned} V^{m,n}(t) &= \mathsf{P}(N_t^n\le m)\cdot m,\\ V_k^{m-1,n-1}(t) &= \mathsf{P}(N_{k,t}^{n-1}\le m-1)\cdot(m-1). \end{aligned}$$

Then the claim can be proved by inserting these terms back into (2). □

Remark 2.3. *Though the pricing formulas require* P *to be the risk-neutral measure, the relation (3) wouldn't be changed by a change of measure and should hold under any measure.*

Note that (3) is only valid for $n\ge 2$, $m\ge 1$. Since the smallest n,m are $n=1$ and $m=0$, the initial cases that are not covered by (3) are

$$\text{Initial cases}: \begin{cases} n=1, & m=0,1,\\ n\ge 2, & m=0. \end{cases}$$

Also note that $n=1$, $m=1$ is a trivial case since $\mathsf{P}(N_t^1\le 1)=1$. Thus $\mathsf{P}(N_t^n\le 0)$ for $1\le n\le \bar{n}$ are the initial condition (IC) for (3) to work. For a given integer $\bar{n}$, the probabilities concerned are $\mathsf{P}(N_t^{\bar{n}}\le m)$, $0\le m\le \bar{n}$. As long as the initial probabilities are given, we are able to use (3) to obtain all the $\mathsf{P}(N_t^n\le m)$ for $0\le m\le n$, $1\le n\le \bar{n}$.

Though (3) holds for a general heterogeneous group of entities, this approach will not be effective when the group size is large. The use of the recursive formula means we have to compute all the $\mathsf{P}(N_t^n \leq m)$ for $n = 1, \cdots, \bar{n}$ even if we are only interested in $n = \bar{n}$. More importantly, when it comes to N_t^n we are actually working on a size n subset of the original size $\bar{n}$ group. For a fixed n there are in total $\binom{\bar{n}}{n}$ combinations of such size n subsets that need to be considered, making the problem very complicated. As discussed in Huge (2001) and Lando (2004), the implementation of (1) is usually infeasible for the cases with n, m going up to $n = 50$ and $m = 10$. This also marks the limitation of formula (3).

2.1 Homogeneous (1-G) case

The problem is much simpler if we assume the original size $\bar{n}$ group is homogeneous (abbreviated as 1-G). In this case the group consists of $\bar{n}$ homogeneous entities with identical parameters (i.e. with completely symmetric correlation structure) and taking any entity out of a size n subset makes no difference, thus $N_{k,t}^{n-1} = N_t^{n-1}$. Therefore, the recursive formula (3) becomes the following simpler version:

$$\mathsf{P}(N_t^n \leq m) = \frac{1}{m}\left[n\mathsf{P}(N_t^{n-1} \leq m-1) - (n-m)\mathsf{P}(N_t^n \leq m-1) \right], \tag{4}$$

where $n \geq 2, m \geq 1, t \geq 0$.

Below we summarize the whole procedure to calculate $\mathsf{P}(N_t^{\bar{n}} \leq m)$ for the homogeneous 1-G case. There are actually two levels of recursions: for a given $\bar{n}$ the algorithm works recursively from $n = 1, \cdots, \bar{n}$, while for each fixed n there is another recursion running from $m = 0, \cdots, n$.

The procedure for a fixed n, named $\mathbf{P}(n)$:

1. If $n = 1$:

   Both $m = 0, 1$ should be given beforehand (IC).

2. If $n \geq 2$:

   (a) $m = 0$ should be given beforehand (IC).

   (b) $m = 1, \cdots, n$, use formula (4).

The whole 1-G algorithm:
For $n = 1, \cdots, \bar{n}$, do $\mathbf{P}(n)$.

Shown in Figure 1 is a diagram of running such two-level recursions. The light (blue) circles are the initial conditions which have to be ready before the recursions begin. The dark (green) circles are those covered in (4), with arrows showing how the recursions are running.

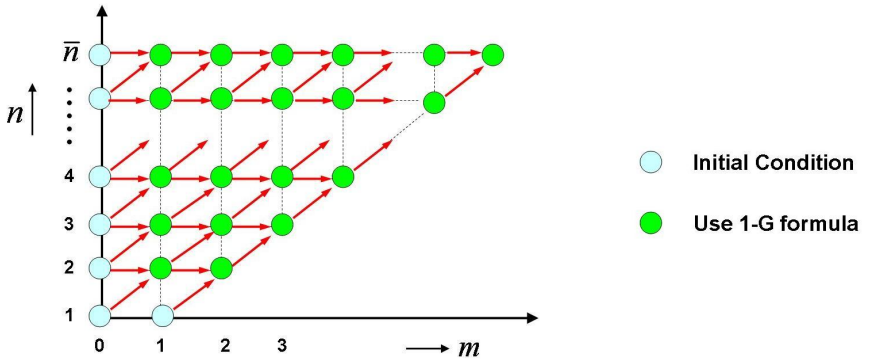

Figure 1: The two-level recursions in the homogeneous 1-G case.

3 Extensions to multiple groups of homogeneous entities

Although (4) is simple and easy to implement (without having to determine all possible size n subsets of the original group), the assumption of homogeneity is rather restrictive. As the formula for the heterogeneous case is hard to implement for large n, to strike a good balance we shall look at cases between these two extremes. In this section we firstly develop the recursive formula for the two-group (2-G) case and we then give the formulas for the three-group (3-G) and the general k-group (k-G) cases.[2]

3.1 Extension to the 2-G case

We assume that in the original size n group there are two subgroups (groups 1 and 2, with sizes n_1, n_2, $n_1 + n_2 = n$) of homogeneous entities and these subgroups have different parameters. Let $N_i^{n_i}(t), i = 1, 2$, be the number of defaults in the subgroup i at time t. Define a contract with payoff $C(m_1, n_1, m_2, n_2)$ at time t given by

$$C(m_1, n_1, m_2, n_2) = \begin{cases} m_1 m_2, & \text{if } N_1^{n_1}(t) \le m_1 \text{ and } N_2^{n_2}(t) \le m_2, \\ 0, & \text{otherwise.} \end{cases}$$

Denote its price by $V^{m_1,n_1,m_2,n_2}(t)$. As in Theorem 2.1, we show that the price satisfies a general version of recursive formula (2) as follows.

Theorem 3.1. *The following recursive formula holds for* $V^{m_1,n_1,m_2,n_2}(t)$*:*

$$\begin{aligned} V^{m_1,n_1,m_2,n_2}(t) = \frac{1}{(m_1-1)(m_2-1)} \Big[& n_1 n_2 V^{m_1-1,n_1-1,m_2-1,n_2-1}(t) \\ & - (n_1 - m_1)(m_2 - 1) V^{m_1-1,n_1,m_2,n_2}(t) \\ & - (m_1 - 1)(n_2 - m_2) V^{m_1,n_1,m_2-1,n_2}(t) \\ & - (n_1 - m_1)(n_2 - m_2) V^{m_1-1,n_1,m_2-1,n_2}(t) \Big], \end{aligned} \tag{5}$$

where $n_1, n_2 \ge 2, m_1, m_2 \ge 2, t \ge 0$.

Proof. Proof.

The LHS claim is $C(m_1, n_1, m_2, n_2)$, while the claim corresponding to the RHS is

$$\frac{1}{(m_1-1)(m_2-1)}\Big[\, n_1 n_2 C(m_1-1, n_1-1, m_2-1, n_2-1) \\ -(n_1-m_1)(m_2-1)C(m_1-1, n_1, m_2, n_2) \\ -(m_1-1)(n_2-m_2)C(m_1, n_1, m_2-1, n_2) \\ -(n_1-m_1)(n_2-m_2)C(m_1-1, n_1, m_2-1, n_2)\Big],$$

where $n_1, n_2 \geq 2, m_1, m_2 \geq 2$. We want to prove the claims in both sides are worth the same for every possible case of $(N_1^{n_1}(t), N_2^{n_2}(t))$.

In case $N_1^{n_1}(t) > m_1$ or $N_2^{n_2}(t) > m_2$, both sides' claims are worth nothing. Therefore we only need to check $N_1^{n_1}(t) \leq m_1$ and $N_2^{n_2}(t) \leq m_2$, in which case the LHS claim pays $m_1 m_2$. This can be further divided into four cases, and in each of them we show the RHS claim also pays $m_1 m_2$. We begin with the observation that

$$\begin{aligned} \mathsf{P}(N_i^{n_i-1}(t) = m_i - 1 | N_i^{n_i}(t) = m_i) &= \mathsf{P}(\text{the excluded one has defaulted}) = \tfrac{m_i}{n_i}, \\ \mathsf{P}(\ N_i^{n_i-1}(t) = m_i\ | N_i^{n_i}(t) = m_i) &= \mathsf{P}(\text{the excluded one has survived}) = \tfrac{n_i - m_i}{n_i}, \end{aligned}$$

$i = 1, 2$. Also note that if the excluded one has defaulted, the remaining size $(n_i - 1)$ set would have $(m_i - 1)$ defaulted entities, making the payment condition of $C(m_1, n_1, m_2, n_2)$ partly satisfied (group i part). We now check the four cases separately.

1. $N_1^{n_1}(t) = m_1, N_2^{n_2}(t) = m_2$: the RHS claim is worth
$$\frac{1}{(m_1-1)(m_2-1)}\left[n_1 n_2 \cdot \frac{m_1}{n_1}\frac{m_2}{n_2} \cdot (m_1-1)(m_2-1) - 0 - 0 - 0 \right] = m_1 m_2.$$
2. $N_1^{n_1}(t) = m_1, N_2^{n_2}(t) < m_2$: the RHS claim is worth
$$\begin{aligned} &\frac{1}{(m_1-1)(m_2-1)}\Big[n_1 n_2 \cdot \frac{m_1}{n_1} \cdot (m_1-1)(m_2-1) - 0 \\ &\qquad -(m_1-1)(n_2-m_2)\cdot(m_1)(m_2-1) - 0 \Big] \\ = \;& m_1 n_2 - m_1(n_2 - m_2) = m_1 m_2. \end{aligned}$$
3. $N_1^{n_1}(t) < m_1, N_2^{n_2}(t) = m_2$: this is case 2. with the roles of m_1 and m_2 swapped.
4. $N_1^{n_1}(t) < m_1, N_2^{n_2}(t) < m_2$: the RHS claim is worth
$$\begin{aligned} &\frac{1}{(m_1-1)(m_2-1)}\Big[n_1 n_2 \cdot (m_1-1)(m_2-1) - (n_1-m_1)(m_2-1)\cdot(m_1-1)(m_2) \\ &\quad -(m_1-1)(n_2-m_2)\cdot(m_1)(m_2-1) - (n_1-m_1)(n_2-m_2)\cdot(m_1-1)(m_2-1) \Big] \\ = \;& n_1 n_2 - (n_1-m_1)m_2 - m_1(n_2-m_2) - (n_1-m_1)(n_2-m_2) \\ = \;& m_1 m_2. \end{aligned}$$

Thus we complete the proof. □

Once again, we may derive the recursive formula for $\mathsf{P}(N_1^{n_1}(t) \le m_1, N_2^{n_2}(t) \le m_2)$ from its relation to $V^{m_1,n_1,m_2,n_2}(t)$. For simplicity we use the streamlined notations $N_i = N_i^{n_i}(t)$ and $N_i' = N_i^{n_i-1}(t)$, $i = 1, 2$, where (t) is dropped as the t-dependence is obvious and not emphasized.

Theorem 3.2. *The following recursive formula holds for* $\mathsf{P}(N_1 \le m_1, N_2 \le m_2)$*:*

$$\begin{aligned}\mathsf{P}(N_1 \le m_1, N_2 \le m_2) = \frac{1}{m_1 m_2}\Big[\, & n_1 n_2 \mathsf{P}(N_1' \le m_1 - 1, N_2' \le m_2 - 1) \\ & - (n_1 - m_1) m_2 \mathsf{P}(N_1 \le m_1 - 1, N_2 \le m_2) \\ & - m_1 (n_2 - m_2) \mathsf{P}(N_1 \le m_1, N_2 \le m_2 - 1) \\ & - (n_1 - m_1)(n_2 - m_2) \mathsf{P}(N_1 \le m_1 - 1, N_2 \le m_2 - 1) \,\Big],\end{aligned} \tag{6}$$

where $n_1, n_2 \ge 2, m_1, m_2 \ge 1, t \ge 0$*.*

Proof. Proof. According to the definition of the contract $C(m_1, n_1, m_2, n_2)$, the prices appearing in Theorem 3.1 are (again thinking of P as the risk-neutral measure):

$$\begin{aligned} V^{m_1-1,n_1-1,m_2-1,n_2-1}(t) &= \mathsf{P}(N_1' \le m_1 - 1, N_2' \le m_2 - 1) \cdot (m_1 - 1)(m_2 - 1), \\ V^{l_1,n_1,l_2,n_2}(t) &= \mathsf{P}(N_1 \le l_1, N_2 \le l_2) \cdot l_1 l_2, \end{aligned}$$

where $l_i = m_i - 1$ or m_i. Substituting these terms into (5) we obtain the claim. □

Note that the above formula is only valid for $n_1, n_2 \ge 2, m_1, m_2 \ge 1$. Since the smallest n_i, m_i are $n_1 = n_2 = 1$ and $m_1 = m_2 = 0$, there are many initial probabilities that are not covered by (6), including

$$\text{Initial probabilities} : \begin{cases} n_1 = 1, n_2 = 1, & \text{all } (m_1, m_2), \\ n_1 \ge 2, n_2 = 1, & \text{all } (m_1, m_2), \\ n_1 = 1, n_2 \ge 2, & \text{all } (m_1, m_2), \\ n_1 \ge 2, n_2 \ge 2, & \begin{cases} m_1 = 0, m_2 = 0, \\ m_1 \ge 1, m_2 = 0, \\ m_1 = 0, m_2 \ge 1. \end{cases} \end{cases}$$

Shown in Figure 2 is an illustration of the probabilities that are or are not covered by (6). There are four examples of (n_1, n_2) correspond to the above four classes. Note that (6) applies only to all the dark (red) circles in the case of $(n_1, n_2) = (4, 3)$ (i.e. $n_1 \ge 2, n_2 \ge 2$). All the remaining circles, including the light (yellow) and unfilled (white) ones (their difference will be explained later) in the four examples, are the initial probabilities required for (6) to work.

In fact, some of these initial probabilities satisfy a version of the 1-G formula (4). These formulas will apply to the light (yellow) circles in Figure 2 and that is why we separate them from the unfilled (white) ones. We first look at the light circles in the case $n_1 \ge 2, n_2 = 1$.

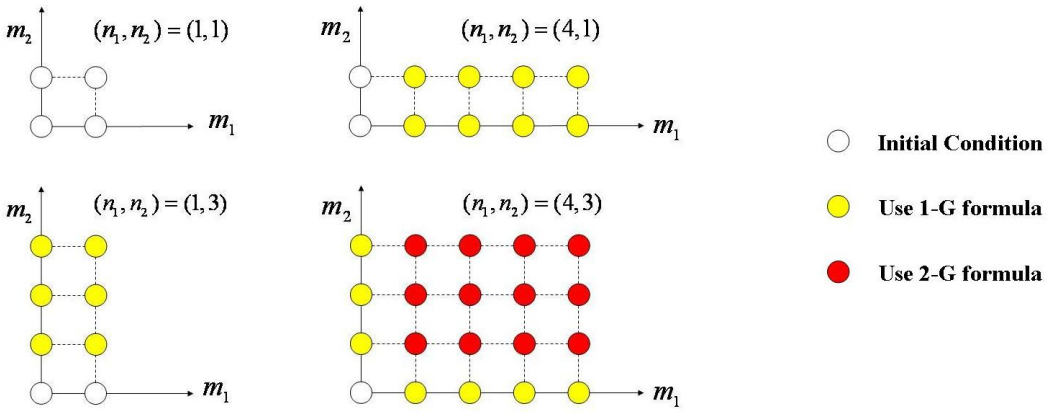

Figure 2: Examples of the four cases of (n_1, n_2).

Theorem 3.3. *Let $n_1 \geq 2, m_1 \geq 1, n_2 \geq 1$. If $m_2 = 0$ or $m_2 = n_2$, the following formula holds:*

$$\begin{aligned} & \mathsf{P}(N_1 \leq m_1, N_2 \leq m_2) \\ = \; & \frac{1}{m_1}\left[n_1 \mathsf{P}(N_1' \leq m_1 - 1, N_2 \leq m_2) - (n_1 - m_1)\mathsf{P}(N_1 \leq m_1 - 1, N_2 \leq m_2)\right]. \end{aligned} \tag{7}$$

Proof. Proof. In the second case $m_2 = n_2$, because $N_2 \leq n_2$ is always true and can be taken out, we recover the 1-G formula for N_1. To prove the first case $m_2 = 0$, we rewrite its LHS and RHS as below:

$$\begin{aligned} \text{LHS} &= m_1 \mathsf{P}(N_1 = m_1, N_2 \leq 0), \\ \text{RHS} &= n_1 \mathsf{P}(N_1' \leq m_1 - 1, N_2 \leq 0) - n_1 \mathsf{P}(N_1 \leq m_1 - 1, N_2 \leq 0). \end{aligned}$$

In order to relate N_1' with N_1, we need to define A_1 as the indicator random variable for the event that the excluded entity in group 1 has defaulted. We then deduce the RHS as follows:

$$\begin{aligned} \text{RHS} &= n_1 \Big[\mathsf{P}(\underbrace{N_1' \leq m_1 - 1}_{N_1 \leq m_1 \,(\because A_1 = 1)}, N_2 \leq 0, A_1 = 1) + \mathsf{P}(\underbrace{N_1' \leq m_1 - 1}_{N_1 \leq m_1 - 1 \,(\because A_1 = 0)}, N_2 \leq 0, A_1 = 0) \\ &\qquad - \mathsf{P}(N_1 \leq m_1 - 1, N_2 \leq 0, A_1 = 1) - \mathsf{P}(N_1 \leq m_1 - 1, N_2 \leq 0, A_1 = 0) \Big] \\ &= n_1 \Big[\mathsf{P}(N_1 \leq m_1, N_2 \leq 0, A_1 = 1) - \mathsf{P}(N_1 \leq m_1 - 1, N_2 \leq 0, A_1 = 1) \Big] \\ &= n_1 \; \mathsf{P}(N_1 = m_1, N_2 \leq 0, A_1 = 1) \\ &= n_1 \; \mathsf{P}(N_1 = m_1, N_2 \leq 0) \underbrace{\mathsf{P}(A_1 = 1 | N_1 = m_1, N_2 \leq 0)}_{\frac{m_1}{n_1}} \\ &= m_1 \; \mathsf{P}(N_1 = m_1, N_2 \leq 0) = \text{LHS}, \end{aligned}$$

which completes the proof. □

Note that if we divide the formula (7) by $\mathsf{P}(N_2 \leq m_2)$ we can see that the conditional distribution of N_1 given $N_2 \leq m_2$ satisfies the 1-G formula.

It is now clear for the case $(n_1, n_2) = (4, 1)$ in Figure 2, (7) can be applied to the two rows of light circles (the two rows correspond to $m_2 = 0$ and $m_2 = n_2$ in (7)). Next, for the case $n_1 = 1, n_2 \geq 2$, a similar formula can be obtained by swapping the roles of n_1, m_1 and n_2, m_2 in (7), as stated below.

Theorem 3.4. *For $n_2 \geq 2, m_2 \geq 1$, $n_1 \geq 1$, if $m_1 = 0$ or $m_1 = n_1$, we have*

$$\begin{aligned} & \mathsf{P}(N_1 \leq m_1, N_2 \leq m_2) \\ = \ & \frac{1}{m_2}\left[n_2 \mathsf{P}(N_1 \leq m_1, N_2' \leq m_2 - 1) - (n_2 - m_2)\mathsf{P}(N_1 \leq m_1, N_2 \leq m_2 - 1)\right]. \end{aligned} \tag{8}$$

As before we see that the conditional distribution of N_2 given $N_1 \leq m_1$ satisfies the 1-G formula.

Again we see the two columns of light circles for the case $(n_1, n_2) = (1, 3)$ correspond to $m_1 = 0$ or $m_1 = n_1$ in (8). As for the case $n_1 \geq 2, n_2 \geq 2$, both (7) and (8) are required for the row and the column of light circles (note that here we use $m_2 = 0$ in (7) and $m_1 = 0$ in (8)).

With these two 1-G formulas, the initial probabilities that remain not covered become the following smaller set (i.e. the unfilled circles in Figure 2):

$$\text{Initial cases}: \begin{cases} n_1 = 1, n_2 = 1, & (m_1, m_2) = (0,0), (0,1), (1,0), (1,1), \\ n_1 \geq 2, n_2 = 1, & (m_1, m_2) = (0,0), (0,1), \\ n_1 = 1, n_2 \geq 2, & (m_1, m_2) = (0,0), (1,0), \\ n_1 \geq 2, n_2 \geq 2, & (m_1, m_2) = (0,0). \end{cases}$$

As long as the probabilities for these initial cases are given in advance, then we may use the formulas (6)−(8) to obtain all the subsequent probabilities. Let $\bar{n}_1$, $\bar{n}_2$ be the target group sizes. In the following we summarize the whole procedure to calculate all the probabilities $\mathsf{P}(N_1^{n_1} \leq m_1, N_2^{n_2} \leq m_2)$, $0 \leq m_i \leq n_i$, $1 \leq n_i \leq \bar{n}_i$, $i = 1, 2$. It is still a two-level process, in which the first level recursion runs for all (n_1, n_2) from $(1, 1)$ to $(\bar{n}_1, \bar{n}_2)$, and for each fixed (n_1, n_2) the second level recursion runs for all (m_1, m_2) from $(0, 0)$ to (n_1, n_2).

The procedure for a fixed (n_1, n_2), named $\mathbf{P}(n_1, n_2)$:

1. If $n_1 = 1, n_2 = 1$:

   All $(m_1, m_2) = (0,0), (0,1), (1,0), (1,1)$ should be given beforehand (IC).

2. If $n_1 \geq 2, n_2 = 1$:

   (a) $m_1 = 0$, i.e. $(m_1, m_2) = (0,0)$ and $(0,1)$ should be given beforehand (IC).

   (b) $m_1 = 1, \cdots, n_1$, use 1-G formula (7).

3. If $n_1 = 1, n_2 \geq 2$:

(a) $m_2 = 0$, i.e. $(m_1, m_2) = (0, 0)$ and $(1, 0)$ should be given beforehand (IC).

(b) $m_2 = 1, \cdots, n_2$, use 1-G formula (8).

4. If $n_1 \geq 2, n_2 \geq 2$:

(a) $m_1 = 0, m_2 = 0$ should be given beforehand (IC).

(b) $m_1 \geq 1, m_2 = 0$, use 1-G formula (7).

(c) $m_1 = 0, m_2 \geq 1$, use 1-G formula (8).

(d) $m_1 \geq 1, m_2 \geq 1$, use 2-G formula (6).

The whole 2-G algorithm:

For $(n_1, n_2) = (1, 1), \cdots, (\bar{n}_1, \bar{n}_2)$, do $\mathbf{P}(n_1, n_2)$.

When all $\mathsf{P}(N_1^{n_1} \leq m_1, N_2^{n_2} \leq m_2)$ are ready, we are able to calculate all $\mathsf{P}(N_1^{n_1} = m_1, N_2^{n_2} = m_2)$ from the following formulas:

$$\begin{aligned}\mathsf{P}(N_1^{n_1} = m_1, N_2^{n_2} = m_2) &= \mathsf{P}(N_1^{n_1} \leq m_1, N_2^{n_2} \leq m_2) - \mathsf{P}(N_1^{n_1} \leq m_1 - 1, N_2^{n_2} \leq m_2) \\ &\quad - \mathsf{P}(N_1^{n_1} \leq m_1, N_2^{n_2} \leq m_2 - 1) + \mathsf{P}(N_1^{n_1} \leq m_1 - 1, N_2^{n_2} \leq m_2 - 1),\end{aligned}$$

for $m_1 \geq 1$ and $m_2 \geq 1$. If one of m_1 and m_2 is 0, then

$$\begin{aligned}\mathsf{P}(N_1^{n_1} = m_1, N_2^{n_2} = 0) &= \mathsf{P}(N_1^{n_1} \leq m_1, N_2^{n_2} \leq 0) - \mathsf{P}(N_1^{n_1} \leq m_1 - 1, N_2^{n_2} \leq 0), \quad \text{if } m_1 \geq 1, \\ \mathsf{P}(N_1^{n_1} = 0, N_2^{n_2} = m_2) &= \mathsf{P}(N_1^{n_1} \leq 0, N_2^{n_2} \leq m_2) - \mathsf{P}(N_1^{n_1} \leq 0, N_2^{n_2} \leq m_2 - 1), \quad \text{if } m_2 \geq 1.\end{aligned}$$

When these are done, the default probabilities $\mathsf{P}(N^n = m)$ and $\mathsf{P}(N^n \leq m)$ where $N^n = N_1^{n_1} + N_2^{n_2}$, $n = n_1 + n_2$, and $m = m_1 + m_2$ (irrespective of which subgroup a default entity belongs to) can be obtained from the following formulas:

$$\mathsf{P}(N^n = m) = \sum_{m_1 + m_2 = m} \mathsf{P}(N_1^{n_1} = m_1, N_2^{n_2} = m_2) \quad \text{and} \quad \mathsf{P}(N^n \leq m) = \sum_{j=0}^{m} \mathsf{P}(N^n = j).$$

3.2 Comparison between heterogeneous, 1-G, and 2-G cases

We now make a comparison of the computational complexity across the heterogeneous, 1-G, and 2-G cases. The aim is to see how the complexity grows with the target group size $\bar{n}$ in each case. Here we only focus on the complexity involved in the recursive formulas, namely, the required initial conditions are assumed given beforehand.

For a given $\bar{n}$, the complexity depends on the total number of recursions required, which is roughly equal to the total number of default probabilities to be calculated. In the heterogeneous case, for each given n there are $\binom{\bar{n}}{n}$ combinations of size n subsets out of the original group. For

each size n subset there are $(n+1)$ probabilities $\mathsf{P}(N^n \le m)$ to be calculated. Thus the total number of default probabilities is

$$\sum_{n=1}^{\bar{n}} \binom{\bar{n}}{n}(n+1) = \sum_{n=0}^{\bar{n}} \binom{\bar{n}}{n}(n+1) - 1 = (\bar{n}+2)2^{\bar{n}-1} - 1.$$

We see the computational complexity is $O(\bar{n}2^{\bar{n}})$, which grows exponentially with $\bar{n}$.

In the 1-G case, since there is no difference between size n subsets, the number of probabilities $\mathsf{P}(N^n \le m)$ becomes $\sum_{n=1}^{\bar{n}}(n+1) = \frac{(\bar{n}+1)(\bar{n}+2)}{2} - 1$, reducing the computational complexity from exponential to second order, i.e. $O(\bar{n}^2)$.

In the 2-G case, given (n_1, n_2) there are $(n_1+1)(n_2+1)$ probabilities $\mathsf{P}(N_1^{n_1} = m_1, N_2^{n_2} = m_2)$ to be calculated. Their total number becomes

$$\sum_{n_1=1}^{\bar{n}_1}\sum_{n_2=1}^{\bar{n}_2}(n_1+1)(n_2+1) = \left[\frac{(\bar{n}_1+1)(\bar{n}_1+2)}{2} - 1\right]\left[\frac{(\bar{n}_2+1)(\bar{n}_2+2)}{2} - 1\right].$$

Therefore, the computational complexity is $O(\bar{n}_1^2\bar{n}_2^2)$, which is second order in both $\bar{n}_1$ and $\bar{n}_2$. If we further assume $\bar{n}_1 = c\bar{n}_2$ for a constant c (i.e. $\bar{n}_1 = \frac{c}{c+1}\bar{n}, \bar{n}_2 = \frac{1}{c+1}\bar{n}$), then the complexity becomes $O(\bar{n}^4)$, i.e. fourth order in $\bar{n}$.

As a summary, from this simple analysis we see that using a grouped homogeneous structure may greatly reduce the complexity from exponential growth to polynomial growth, which indicates that higher $\bar{n}$ can be reached for a given computational power.

3.3 The 3-G and general k-G cases

In fact, the recursive structure can be further extended to general cases with three or more homogeneous groups following a similar treatment. It is clear that the computational complexity must grow to a higher order when more groups are involved. We usually have to trade off the degree of heterogeneity against the computation load. Below we give these general results with proofs omitted to save space. We first look at the 3-G case.

Theorem 3.5. *The recursive formula for the default probabilities for the three homogeneous groups* $\mathsf{P}(N_1 \le m_1, N_2 \le m_2, N_3 \le m_3)$ *is*

$$\begin{aligned} & \mathsf{P}(N_1 \le m_1, N_2 \le m_2, N_3 \le m_3) \\ = \; & \frac{1}{m_1 m_2 m_3}\Big[\, n_1 n_2 n_3 \mathsf{P}(N_1' \le m_1 - 1, N_2' \le m_2 - 1, N_3' \le m_3 - 1) \end{aligned}$$

$$
\begin{aligned}
&- \ (n_1-m_1)m_2m_3\mathsf{P}(N_1 \le m_1-1, N_2 \le m_2, N_3 \le m_3) \\
&- \ m_1(n_2-m_2)m_3\mathsf{P}(N_1 \le m_1, N_2 \le m_2-1, N_3 \le m_3) \\
&- \ m_1m_2(n_3-m_3)\mathsf{P}(N_1 \le m_1, N_2 \le m_2, N_3 \le m_3-1) \\
&- \ (n_1-m_1)(n_2-m_2)m_3\mathsf{P}(N_1 \le m_1-1, N_2 \le m_2-1, N_3 \le m_3) \\
&- \ (n_1-m_1)m_2(n_3-m_3)\mathsf{P}(N_1 \le m_1-1, N_2 \le m_2, N_3 \le m_3-1) \\
&- \ m_1(n_2-m_2)(n_3-m_3)\mathsf{P}(N_1 \le m_1, N_2 \le m_2-1, N_3 \le m_3-1) \\
&- \ (n_1-m_1)(n_2-m_2)(n_3-m_3)\mathsf{P}(N_1 \le m_1-1, N_2 \le m_2-1, N_3 \le m_3-1) \ \Big],
\end{aligned}
$$

where $n_1, n_2, n_3 \ge 2, m_1, m_2, m_3 \ge 1$.

Based on the 3-G formula, it is not difficult to extend the recursive formulas by induction to derive the general k-G case as given below.

Theorem 3.6. *The general recursive formula for the default probabilities for $k < n$ homogeneous groups $\mathsf{P}(N_1 \le m_1, N_2 \le m_2, \ldots, N_k \le m_k)$ is*

$$
\begin{aligned}
&\mathsf{P}(\cup_{i=1}^k N_i \le m_i) = \frac{1}{\prod_{i=1}^k m_i}\Bigg[\prod_{i=1}^k n_i \mathsf{P}(\cup_{i=1}^k N_i' \le m_i - 1) \\
&-\sum_{j=1}^k \sum_{l_1=1}^k \sum_{l_2=l_1+1}^k \cdots \sum_{l_j=l_{j-1}+1}^k \prod_{i=1}^j (n_{l_i}-m_{l_i}) \prod_{i\ne l_1,\ldots,l_j} m_i \mathsf{P}(\cup_{i=1}^j \{N_{l_i} \le m_{l_i}-1\} \cup \cup_{i\ne l_1,\ldots,l_j}\{N_i \le m_i\})\Bigg].
\end{aligned}
$$

Remark 3.7. *To understand the k-G formula we may compare it with the 3-G formula in which we observe that, for the negative terms in the bracket, $(n_i - m_i)$ in the coefficient corresponds to the event $N_i \le m_i - 1$ in the probability, while m_i corresponds to the event $N_i \le m_i$.*

It is clear from the structure of the formula that there is an exponential growth in the number of terms with 2^k terms for the k-G case. We also observe that if we set $m_k = 0$ or $m_k = n_k$, it will reduce to the $(k-1)$-G formula.

4 Numerical examples

In this section we implement the 1-G and 2-G algorithms for a class of Markov chain based reduced-form models. In these models the distribution of N_t^n can be also obtained with high precision from solving their pure death Markov chain, and these results are used to contrast with the results from using the 1-G and 2-G algorithms. Through the numerical examples we compare the computational time and accuracy between these two methods.

Traditionally in the reduced-form default risk models, the default rates are usually described by a diffusion process possibly with jumps (see the monographs, e.g., Lando (2004), Schonbucher (2003), Duffie and Singleton (2003), Bielecki and Rutkowski (2002)). The correlation between the default rates is introduced by the joint movements of either the diffusion terms or the jump

terms. In a sense the effect from the joint movements is similar to that from the common factors in the factor-based models. As pointed out in Schonbucher (2003), Duffie and Gârleanu (2001), the joint movements of the diffusion terms make little contribution to the default correlation and in this setting the only way to generate significant default correlation is through joint jumps (i.e. the default rates are likely to jump up or down simultaneously). For this reason we consider the reduced-form models with default rates taking a finite set of discrete values (i.e. no diffusion terms) and allow joint jumps between these values.

We assume the default rate of each entity follows a 2-state Markov chain (MC), and thus the default arrival is actually a Markov modulated Poisson process (MMPP) (see for example, Heffes and Lucantoni (1986), Ching et al. (1997), Fischer and Meier-Hellstern (1992)).[3] Joint jumps from the low (high) rate state to the high (low) rate state indicate the entities become more (less) risky simultaneously. Note that our purpose here is to have suitable models with which we can test the proposed algorithms and demonstrate their effectiveness, so our models are selected in such a way that they are able to capture important features (e.g. joint jumps) while still keeping its numerically tractability for large group size. Using merely a 2-state MC (instead of a k-state MC, $k > 2$) will help us reach this goal. On one hand, with two states of high and low default rates, we may achieve highest possible default correlation by making the two default rates as contrasting as possible and the joint jump rates as high as possible. On the other hand, the pure death MC for the whole population will be of a manageable size when $\bar{n}$ is large (say $\bar{n} = 50 \sim 60$). Though our models are idealized (they wouldn't allow, for example, successive jumps in default rates), the good balance they strike makes them a good proxy for our numerical study. For more complicated models, our recursive algorithms are still applicable as they are model free, but with even more computational efforts involved in the calculations of initial probabilities.[4]

4.1 Markov chain models for default rates with joint jumps

Consider a group of n inhomogeneous entities. Let λ_t^i represent the i-th entity's default rate, $i = 1, \cdots, n$. We suppose that λ_t^i takes two values, a normal rate and an excited rate and that it is a function of an underlying state variable $X_i(t)$ which evolves as a Markov chain and determines the joint jumps. Define $X_i(t) \in \{0, 1, 2\}$ as below:

$$X_i(t) = \begin{cases} 0 & \text{to represent the } i\text{-th entity in a normal state,} \\ 1 & \text{to represent the } i\text{-th entity in an individual excited state,} \\ 2 & \text{to represent all entities in the joint excited state.} \end{cases}$$

The default rate λ_t^i is specified as

$$\lambda_t^i = \begin{cases} \gamma_i, & \text{normal state, } X_i(t) = 0, \\ \gamma_i + \delta_i, & \text{excited states, } X_i(t) = 1, 2. \end{cases}$$

Thus when the state variable moves from the normal state to an excited state the process λ_t^i has an increase of size δ_i. When an entity's state variable moves to state 2 all the $\lambda_t^i, i = 1, \cdots, n$ are

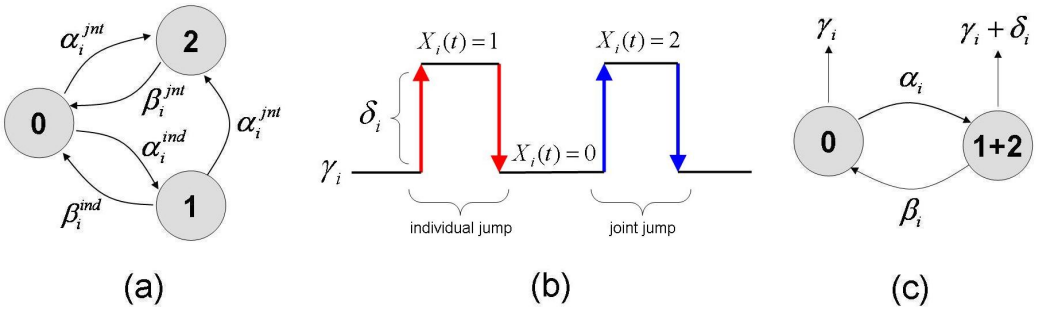

Figure 3: A marginal viewpoint of the states in λ_t^i: (a) 3-state Markov chain of $X_i(t)$, (b) process value of λ_t^i, (c) λ_t^i follows a 2-state Markov chain with two excited states merged if $\beta_i = \beta^{jnt} = \beta_i^{ind}$.

driven to their joint excited states simultaneously, even if some of them have already been in their (individual) excited states. On the other hand, if an entity's state variable $X_i(t)$ moves to state 1, it drives only λ_t^i to its excited state, and this may happen only when $X_i(t) = 0$.

Shown in Figure 3 (a)(b) is an illustration of these states from a marginal viewpoint of an entity i. The jump rates q_{jk} for $X_i(t)$ to move from state j to state k are given by

$$\begin{aligned} q_{01} &= \alpha_i^{ind}, & q_{02} &= \alpha^{jnt}, \\ q_{10} &= \beta_i^{ind}, & q_{12} &= \alpha^{jnt}, \\ q_{20} &= \beta_i^{jnt}, & q_{21} &= 0. \end{aligned}$$

Note that α^{jnt} and β^{jnt} do not have a subscript i as they are common to all entities.

The total up jump rate $\alpha_i = \alpha^{jnt} + \alpha_i^{ind}$ is the sum of the joint and the individual up jump rates. The fraction $\rho_i = \frac{\alpha^{jnt}}{\alpha_i}$ is the probability for entity i to see a joint up jump given a jump happens. This can be seen as a measure of default correlation. In the case $\beta_i = \beta^{jnt} = \beta_i^{ind}$, the two excited states may be merged as the default rate behaves the same way regardless of which kind of excited state it is. Given in Figure 3 (c) is the reduced 2-state Markov chain (or MMPP) for this case.

If we look at a two-entity group, there are 5 possible states for the state vector $(X_1(t), X_2(t))$ including $(0,0), (1,0), (0,1), (1,1), (2,2)$, as sketched in Figure 4 (a). Note that every state (except $(2,2)$) may move to the joint excited state $(2,2)$ with rate α^{jnt}, but the joint excited state may switch back only to the joint normal state $(0,0)$ with rate β^{jnt}. The joint default rate $\lambda_t^{(1)} = \lambda_t^1 + \lambda_t^2$ at each state is given by

$$\lambda_t^1 + \lambda_t^2 = \begin{cases} \gamma_1 + \gamma_2, & \text{if} \quad (X_1(t), X_2(t)) = (0,0), \\ \gamma_1 + \delta_1 + \gamma_2, & \text{if} \quad (X_1(t), X_2(t)) = (1,0), \\ \gamma_1 + \gamma_2 + \delta_2, & \text{if} \quad (X_1(t), X_2(t)) = (0,1), \\ \gamma_1 + \delta_1 + \gamma_2 + \delta_2, & \text{if} \quad (X_1(t), X_2(t)) = (1,1) \text{ or } (2,2). \end{cases}$$

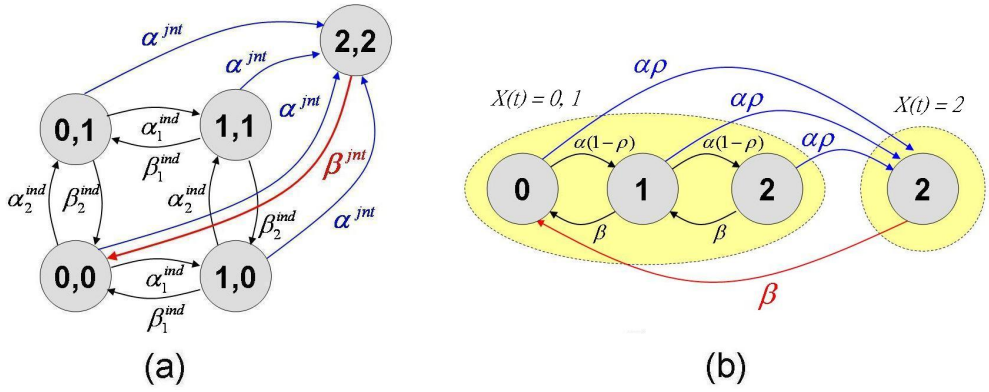

Figure 4: The Markov chains for a two-entity group: (a) the Markov chain $(X_1(t), X_2(t))$, (b) the reduced Markov chain for the homogeneous case. Inside a state circle is the number of entities that are in excited states.

If the two entities are homogeneous with parameter set $(\alpha, \beta, \gamma, \delta)$ as well as ρ being the default correlation index as defined earlier, then a reduced Markov chain can be formulated by specifying only the number of entities in excited states, denoted by j, as shown in Figure 4 (b). In this case the joint default rate is $2\gamma + j\delta$.

4.2 1-G and 2-G models with $\rho = 1$

Now we apply the proposed 1-G and 2-G algorithms to this simple model with larger group size, and discuss the computational aspects of the algorithms. As in the above two-entity homogeneous case, in the 1-G models we assume there are n homogeneous entities and each with parameter set $(\alpha, \beta, \gamma, \delta)$. For simplicity we further assume $\rho = 1$ (the $\rho \neq 1$ case will be discussed later), thus all entities are either in the joint normal state $(X(t) = 0)$ or in the joint excited state $(X(t) = 2)$, and the individual excited state $(X(t) = 1)$ doesn't exist. The pure death process in this case is shown in Figure 5 (a), where we see the corresponding states in the $X(t) = 0$ and $X(t) = 2$ areas jump to each other with rates α and β, and the death rate at each state depends on the number of surviving entities and the state variable $X(t)$.

Generalized from the 1-G models, in the 2-G models we assume there are n_1 and n_2 entities in the groups 1 and 2, with entity parameter sets $(\alpha, \beta, \gamma_1, \delta_1)$ and $(\alpha, \beta, \gamma_2, \delta_2)$ respectively. We also assume $n_1 = n_2 = n$ and $\rho = 1$ for simplicity, thus the Markov chain formulation of the full death process is sketched in Figure 6 (a). Again, the corresponding states jump between each other with rates α and β, and the death rate at each state depends on the number of defaults in each subgroup as well as the state variable $X(t)$. For example, when $X(t) = 2$, the rates of jumping from (m_1, m_2) to $(m_1 + 1, m_2)$ and $(m_1, m_2 + 1)$ are $(n_1 - m_1)(\gamma_1 + \delta_1)$ and $(n_2 - m_2)(\gamma_2 + \delta_2)$.

The default probability distribution $\mathsf{P}(N_t^n \leq m)$ can be obtained either by solving the full Markov chain directly, or by applying the proposed algorithms. Note that in the 1-G algorithm,

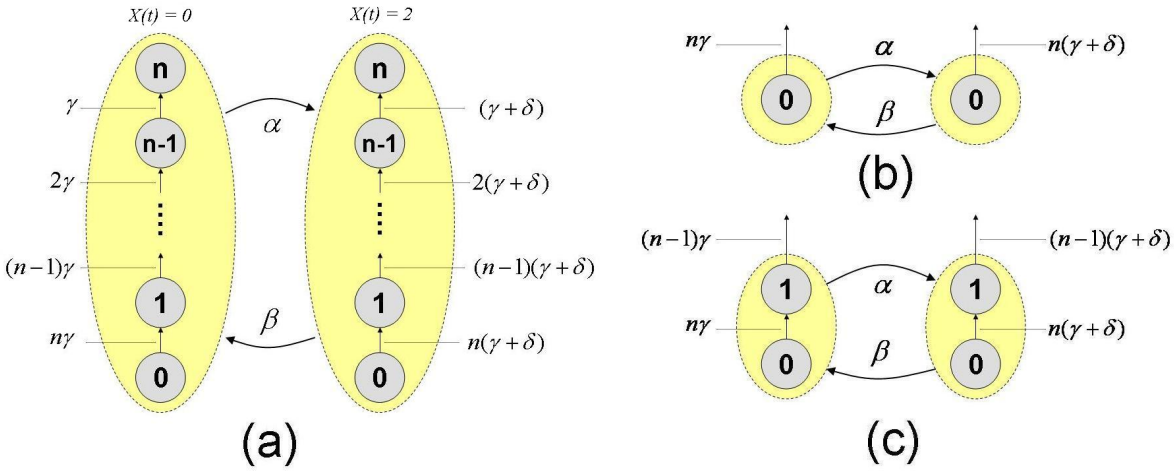

Figure 5: The 1-G model with $\rho = 1$: (a) the full pure death Markov chain, (b) the truncated Markov chain for IC: $m = 0$, (c) the truncated Markov chain for IC: $m \leq 1$. Inside a state circle is the number of defaults.

the initial probabilities $\mathsf{P}(N_t^n \leq m)$ with $m = 0, \forall n$ are required as IC, and these can be obtained by solving a truncated pure death Markov chain with only $m = 0$ as shown in 5 (b). We may also use a larger truncated Markov chain as seen in Figure 5 (c) to calculate $\mathsf{P}(N_t^n \leq m)$ with both $m = 0$ and $1, \forall n$ as the IC. Though this is more than is necessary for our recursive algorithm, its benefits and importance will be discussed later. In the 2-G model, the minimal set of IC required for the 2-G algorithm to work can be obtained by solving the truncated Markov chain as shown in Figure 6 (b), which gives $\mathsf{P}(N_1^{n_1}(t) \leq m_1, N_2^{n_2}(t) \leq m_2)$, $m_1, m_2 \leq 1, \forall n_1, n_2$. Likewise, we may also solve the Markov chain in Figure 6 (c) to obtain the initial probabilities for $m_1, m_2 \leq 2$.

4.2.1 Computation time

We first look at the computation times for solving the full Markov chain directly and using the proposed algorithm. Note that in the former case, we need to solve only one Markov chain that corresponds to the group of target size $\bar{n}$. But when using the proposed recursive algorithms, we need to compute the required IC first by solving the truncated Markov chains corresponding to $n = 1 \sim \bar{n}$ (in 1-G models) or $(n_1, n_2) = (1, 1) \sim (\bar{n}_1, \bar{n}_2)$ (in 2-G models). The results are given in Table 1, which are obtained by running programs on a computer with the following specifications: Intel Pentium M processor 1.73GHz, 1.00GB RAM. Each computation time shown for the 1-G and 2-G algorithms includes the time spent on solving the truncated Markov chains for the IC, as well as on the recursive computations. Table 1 (a) shows the 1-G cases (Figure 5) where all (full and truncated) Markov chains are solved for $t = 0 \sim 5$ by the fourth order Runge-Kutta method (RK4)[5] with $\Delta t = 0.001$, while (b) shows the 2-G cases (Figure 6) where Markov chains are solved by RK4 with $\Delta t = 0.01$.

We see that the computation time for solving the full Markov chain is roughly second order

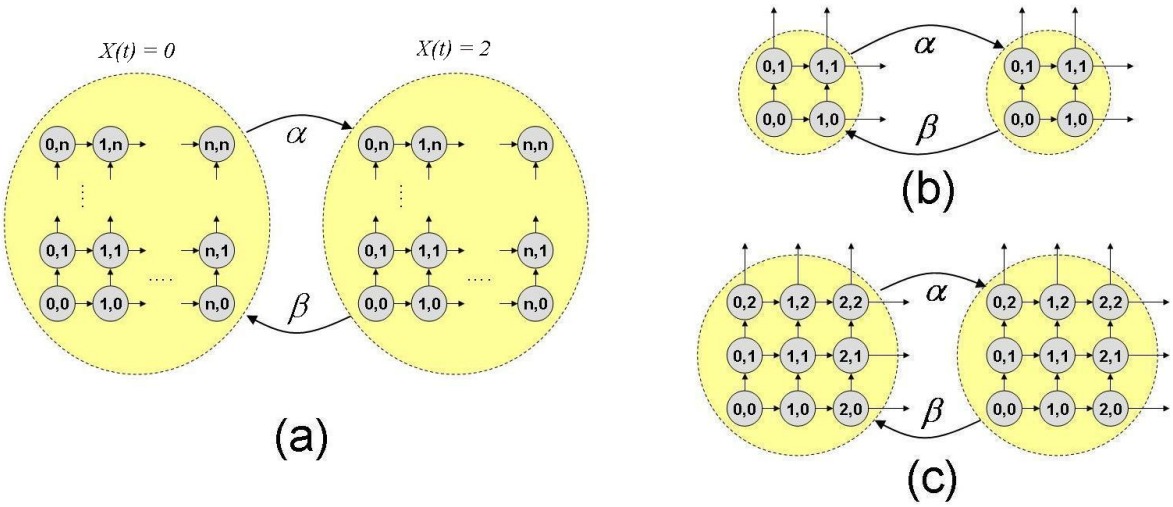

Figure 6: The 2-G model with $\rho = 1$: (a) the full pure death Markov chain, (b) the truncated Markov chain for IC: $m_1, m_2 \leq 1$, (c) the truncated Markov chain for IC: $m_1, m_2 \leq 2$. Inside a state circle are the numbers of defaults in both subgroups.

in the number of states, while the time for using the proposed algorithms is much less. It is noted that the time for the proposed algorithms is roughly $O(\bar{n})$ in the 1-G cases and $O(\bar{n}_1\bar{n}_2)$ in the 2-G cases. This is because the IC requires solving the truncated Markov chains for $\bar{n}$ times in the 1-G cases and $\bar{n}_1\bar{n}_2$ times in the 2-G cases, and the time spent for the IC dominates over the time spent for recursive computations (which is $O(\bar{n}^2)$ or $O(\bar{n}_1^2\bar{n}_2^2)$ as seen in Section 3). The cases (2)(3)(4) in (a) and (b) also show that solving larger truncated Markov chains for greater sets of IC makes the time longer, but the total time is still manageable and much less than solving the full chain.

The above observations justify the use of the proposed algorithms in the sense that the complexity of solving the original pure death Markov chain is shifted to solving the ICs. Because each IC involves solving a MC of a much smaller dimension and the recursive formula involves only simple computation, the saving in the computation time is significant although we need to solve many such MCs and use the recursive formula repeatedly. For more complicated default rate models, our algorithms can also provide such a divide-and-conquer benefit in that we can avoid dealing with a big problem by dealing with many much smaller ones.

4.2.2 Accuracy and propagation errors

We then move on to look at the accuracy of the proposed algorithms. Note that the proposed algorithm is based on a recursive structure in default probabilities, therefore, if the initial conditions are not accurate enough (or prone to be inaccurate, such as having values very close to 0), the error can propagate through the recursions and even yield unreasonable results, such as $\mathsf{P}(N_t^n \leq m) < 0$ or $\mathsf{P}(N_t^n \leq m) > 1$ or $\mathsf{P}(N_t^n \leq m+1) < \mathsf{P}(N_t^n \leq m)$. In addition, the propagation error will

become more significant when the recursion runs farther (n is greater). In fact, to avoid these unreasonable results we have added the following rules in the computer programs:

$$\begin{cases} \text{if } \mathsf{P}(N_t^n \le m) < 0 & \Longrightarrow \quad \text{then set } \mathsf{P}(N_t^n \le m) = 0, \\ \text{if } \mathsf{P}(N_t^n \le m) > 1 & \Longrightarrow \quad \text{then set } \mathsf{P}(N_t^n \le m) = 1, \\ \text{if } \mathsf{P}(N_t^n \le m+1) < \mathsf{P}(N_t^n \le m) & \Longrightarrow \quad \text{then set } \mathsf{P}(N_t^n \le m+1) = \mathsf{P}(N_t^n \le m). \end{cases} \tag{9}$$

Consider a 1-G model where the default rate of each entity is parametrized as earlier, i.e. $(\alpha, \beta, \gamma, \delta) = (0.1, 0.1, 0.01, 0.04)$. Shown in Table 2 is a comparison between solving the full Markov chain and using the 1-G algorithm when $\bar{n} = 20$. We see the differences between (a) and (b) are negligible, namely, the proposed algorithm gives good accuracy. Note that in Table 2 (b), the IC (initial probability with $m = 0$) is not too low (e.g. 1.488797e-1 at $t = 5$), thus the propagation error is insignificant for all subsequent $m = 1, \cdots, 20$.

However, the propagation error becomes more noticeable and sometimes harmful when $\bar{n}$ becomes large and the probabilities in the IC are small. Table 3 shows the case when $\bar{n} = 60$, in which we can see the effect of propagation error by comparing (a) with (b). It can be observed in (b) that the change in each column loses its smoothness around the underlined figures, and the worst case is at the $t = 5$ column where the initial probability ($m = 0$) has the smallest value (1.669630e-2). In fact, at these underlined figures the propagation error has already caused unreasonable results, but due to the use of (9), the figures shown in (b) are seemingly reasonable. The unusual behavior around the underlined figures has revealed that the recursive formula has been unable to work correctly at these figures. In addition, for the column with larger t, the problem becomes worse because the recursion starts with even smaller probability as the IC, which makes the recursive formula break down earlier.

We can fix this problem by using more initial probabilities as the IC. The results are given in Table 4, where the IC includes $m \le 1$ in (a) and $m \le 2$ in (b). We see the improvement in (a) and a further improvement in (b), especially for the larger t columns. We anticipate that by including more initial probabilities in the IC the effect of the propagation error will be further reduced. This is because when more initial probabilities are given, the recursion may start with higher valued probabilities, which are less liable to propagation error. However, in the examples of Tables 2∼4, the probability is relatively low when m is small, e.g. $\mathsf{P}(N_t^n \le 0)$ at $t = 5$ is as low as 1.669630e-02, which is not the usual case in practice as defaults are rare events. In more practical cases where default probabilities are low and survival probabilities are high (e.g. $\mathsf{P}(N_t^n \le 0) = 0.3 \sim 0.5$), the recursion can start with a higher value, in which case the propagation error is much less significant. (It is suggested the initial probabilities should be of a value no less than 0.3 to avoid the effect of propagation errors.)

A graphical presentation of the propagation errors observed in Tables 3∼4 is given in Figure 7, where (a) compares the results from solving the full Markov chain and from using the 1-G algorithm with $m \le 0$, $m \le 1$, $m \le 2$ as its IC. We see the improvement of using more initial probabilities (also see the error curves at the bottom, where the error is defined as the difference between the

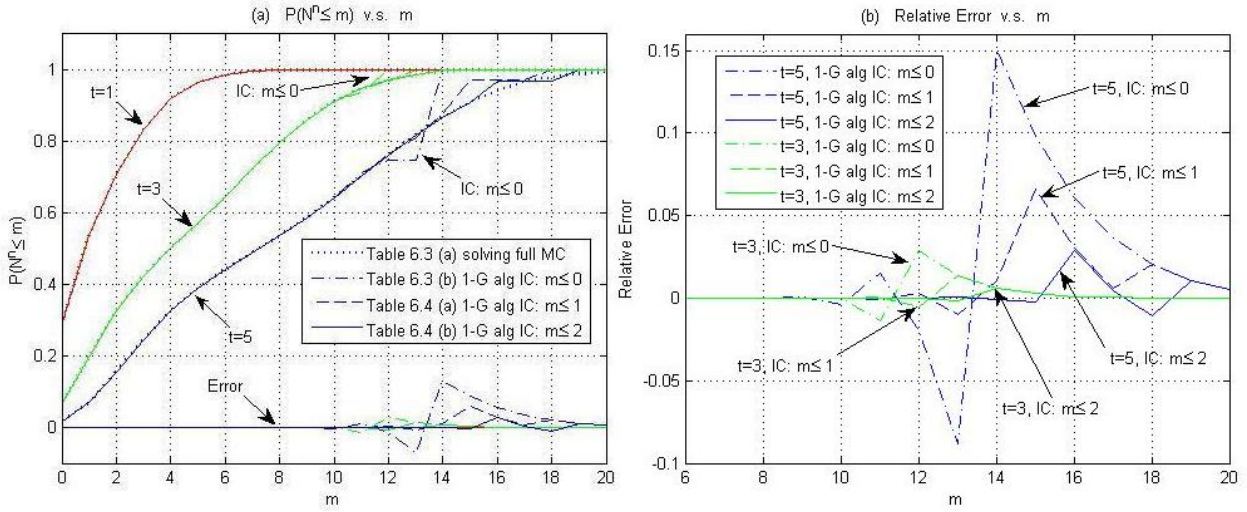

Figure 7: Graphical presentation of the propagation errors observed in Tables 3∼4: (a) $\mathsf{P}(N_t^n \le m)$ v.s. m for $t = 1, 3, 5$, with error curves given at the bottom, (b) relative error v.s. m for $t = 3, 5$ in a larger scale.

results from both methods). Figure 7 (b) shows the improvement in relative errors (the errors normalized by the results from solving the full Markov chain). It is observed in the $t = 5$ curves that when more initial probabilities are used (comparing $m \le 0$ with $m \le 2$), the maximal relative error decreases from 15% at $m = 14$ to 3% at $m = 16$.

4.3 More complicated examples

The above 1-G and 2-G models have such a simple structure because $\rho = 1$. When $\rho \neq 1$, the corresponding pure death Markov chain becomes more complicated and takes much more time to solve. Here we look at the 1-G model with $\rho \neq 1$. Shown in Figure 8 is an example of the pure death Markov chain for such a 1-G model with $n = 5$, where the left part represents the surviving entities at normal or individual excited states ($X(t) = 0, 1$), while the right part represents the joint excited states ($X(t) = 2$). The two tuple (m, j) inside a state circle denotes that there are m surviving entities, where j out of the m are at excited states (either individual or joint). All the states on the left may jump to the corresponding state (with the same number of surviving entities) on the right with rate $\alpha\rho$ (i.e. from $(m, j), j = 1, \cdots, m$ on the left to (m, m) on the right), but the state on the right may jump back only to the first state on the left in the same row (i.e. $(m, 0)$) with rate β (similar to Figure 4 (b)). If we look at a state (m, j) on the left, it may also jump to

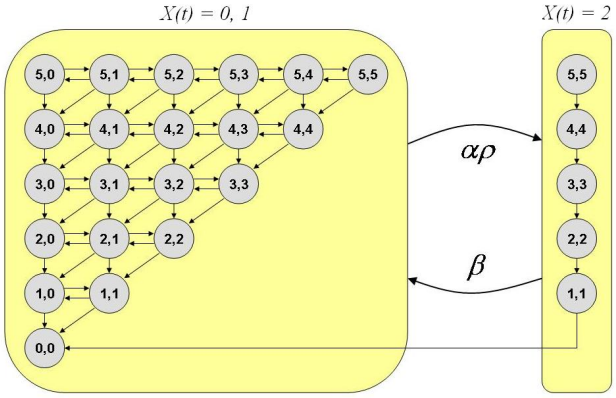

Figure 8: The pure death Markov chain for the 1-G model with $\rho \neq 1$. .

its neighboring states with rates as below:

$$(m,j) \to \begin{cases} (m, j-1), & \text{with rate} \quad j\beta, & \text{if } j \geq 1, \\ (m, j+1), & \text{with rate} \quad (m-j)\alpha(1-\rho), & \text{if } j \leq m-1, \\ (m-1, j-1), & \text{with rate} \quad j(\gamma+\delta), & \text{if } m \geq 1, j \geq 1, \\ (m-1, j), & \text{with rate} \quad (m-j)\gamma, & \text{if } m \geq 1. \end{cases}$$

On the other hand, if we look at a state (m, m) on the right, the only state change is:

$$(m,m) \to (m-1, m-1), \qquad \text{with rate } m(\gamma+\delta), \qquad \text{if } m \geq 1.$$

The $\rho \neq 1$ case for the 2-G models can be also defined in a similar way, but it is not discussed here as the pure death Markov chain will become too complicated.

The next numerical examples show the accuracy of the distribution of N_t^n for the 1-G and 2-G models as the model parameters are varied. $\rho \neq 1$ is allowed in the 1-G models, where we pick $\bar{n} = 50$, $(\alpha, \beta, \gamma, \delta) = (0.1, 0.9, 0.01, 0.06)$ with ρ varying between 0, 0.5, 1.0. In the 2-G models we still assume $\rho = 1$ but vary the parameters across the two groups. We pick $\bar{n}_1 = \bar{n}_2 = 25$, and use the three parameter sets as shown in Table 5.

The three sets represent different degrees of heterogeneity. Set A is actually a homogeneous case with two identical subgroups. While keeping γ_1, δ_1 fixed as in set A, in sets B and C we vary γ_2, δ_2 but still keep the average default rate $\bar{\lambda}$ fixed. Set C has the greatest difference in parameters across the two subgroups. The initial probabilities with $m \leq 1$ are used as the IC for the 1-G algorithm, while those with $m_1 \leq 1, m_2 \leq 1$ are the IC for the 2-G algorithm. The results are given in Figure 9. Once again, we see the proposed algorithms perform nicely compared with solving the full Markov chain. In (b), there is a slight discrepancy seen in the set C curve for $t = 5$ when m is close to 15. This indicates that this case is most prone to propagation error but it is correctable by more initial probabilities being used. In addition, we observe in (a) that ρ may change the shape of the distribution curves, and higher ρ makes the curve more fat-tailed. Similar phenomena are

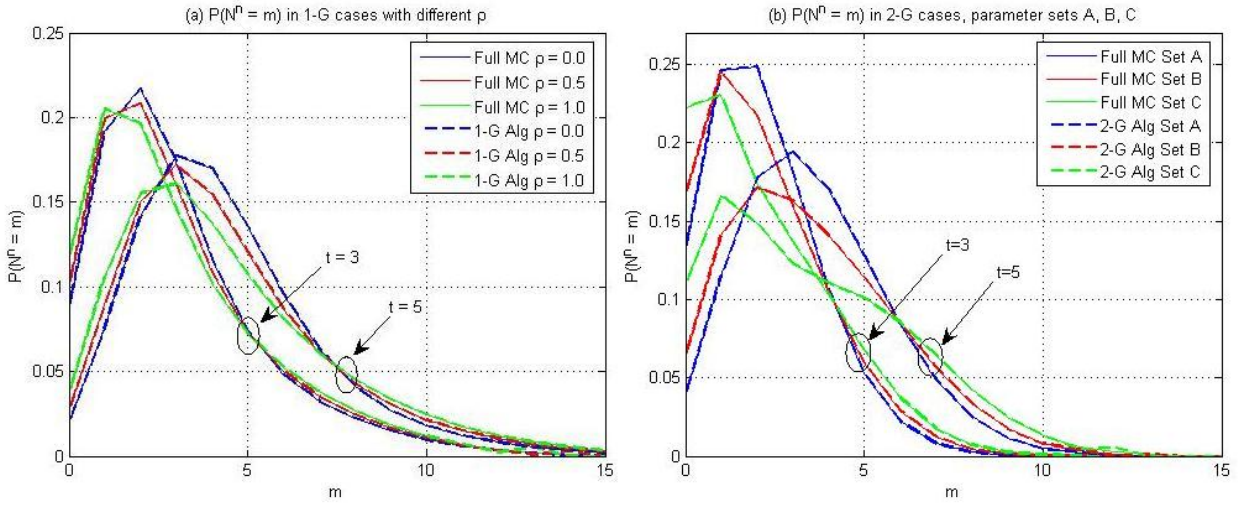

Figure 9: Comparison between solving the full Markov chains and using the proposed 1-G and 2-G algorithms: (a) $\mathsf{P}(N_t^n = m)$ v.s. m in 1-G cases for $t = 3, 5$, with $\rho = 0.0, 0.5, 1.0$, (b) $\mathsf{P}(N_t^n = m)$ v.s. m for $t = 3, 5$ in 2-G cases, with parameter sets A, B, C.

observed in (b) where we see the curves for set C, with the greatest difference in parameters, are the most fat-tailed. In sum, apart from the accuracy of the proposed algorithms, from these examples we also observe that both the correlation structure as well as the heterogeneity style among the group affect the pattern of the distribution curves in a complicated way.

5 Conclusion

The main contribution of this paper is to prove that there exists a recursive structure in the default probabilities in a pool of multiple names. One merit of this structure is that it is model free, namely, it doesn't impose specific assumptions on the default models and correlation structure among entities, and thus has wide applicability. The other merit is that it can be extended to the partly heterogeneous group with multiple homogeneous subgroups for which the recursive algorithms are much more computationally feasible. It serves as a satisfactory technique for the calculation of the whole distribution and strikes a good balance between saving computing power and achieving full heterogeneity.

In the theoretical part of this paper, it is worth noting that the proofs of our main results, which are expressed in the form of probabilistic formulas, are actually based on a pure financial pricing argument. The decomposition of the contingent claim $C(m, n)$ provides us a unique insight into the deduced mathematical results, and also connects our work to the credit derivative literature. In the numerical part of this paper, the proposed algorithms are tested against the standard Markov chain methods under the selected models and shown to perform satisfactorily well. Through the results we demonstrate that in the proposed methods, the computational efficiency does not compromise the

accuracy achieved, as long as the potential problem of propagation error can be properly avoided.

Our results serve as a useful tool for the pricing of the defaultable claims involving multiple names, such as CDO, where the payoff depends mainly on the number of defaults. Though this work originates from the analysis of default risk, its use is not necessarily limited to this area and can be applied to other areas where the probability of an specific event happening to a part of the group members is a concern. This is especially true when there exists a complicated correlation structure among the members because how they are correlated does not pose a problem to our algorithms. Potential examples include estimating the number of faulty products on a manufacture line (operations management) or the number of people catching a specific disease out of a population (epidemiology), etc. In this sense, our results also shed some light on other related areas.

Notes

1. Iscoe and Kreinin (2006) follow Huge (2001) to consider the recursive valuation of the basket CDS contracts and derive a recursive formula for the risk-neutral mth-to-default probability $\mathsf{P}(\tau_{(m)} = \tau_k, \tau_{(m)} \in (t_{i-1}, t_i])$ (for given k, t_{i-1}, t_i) based on order statistics. Such a probability has a connection with the mth-to-default CDS price.

2. In fact, when n is large, it is impractical to assume all the names in a pool are completely heterogeneous. To simplify the problem, these names can instead be categorized into a few classes according to, for instance, their credit ratings from the information provided by rating agencies. As long as the names in a class have similar likelihood to default and symmetric dependence structure, our k-G recursive formulas (with k not too large) are applicable.

3. MMPP has wide applications in such fields as traffic modeling in communication networks or manufacturing systems to reflect the arrival rates may change with time. Here in the field of credit risk modeling, we borrow this idea to model the arrival rates of default events with the purpose of having a computational tractable model in a large group of entities. It is worth noting that in credit risk modeling we care about the transient behavior of MMPP as we only look at the first event for each entity. In other fields the events happen constantly and usually the stationary behavior is of interest.

4. Under more complicated models (e.g. each default rate follows a 3-state MC), the computational benefits of using our algorithms are expected to be more significant because solving the whole pure death MC becomes much more computationally demanding. In contrast, when the proposed algorithm is used, the increased computation costs in obtaining the initial probabilities are not that high and the recursive calculations will not be influenced.

5. The reason we use RK4 is because it strikes a good balance between accuracy and efficiency for solving an ODE system. For the pure death Markov chains the ODE system, which comes

from the Kolmogorov forward equations, is linear, time homogeneous, but of high dimension. The state probabilities are expected to change smoothly over time so the RK4 should provide sufficiently accurate results.

Acknowledgements

The first author would like to thank Prof. Mike Giles (Oxford University), Prof. Harry Zheng (Imperial College London), Prof. San-Lin Chung and Prof. Hung Chen (both with National Taiwan University) for helpful discussions.

References

Andersen, L. and Sidenius, J. (2004-5), 'Extensions to the Gaussian Copula: random recovery and random factor loadings.', *Journal of Credit Risk* **1**, 29–70.

Andersen, L., Sidenius, J. and Basu, S. (2003), 'All your hedges in one basket.', *Risk* **Nov**, 67–72.

Bielecki, T. and Rutkowski, M. (2002), *Credit Risk: Modeling, Valuation and Hedging.*, Springer.

Ching, W., Chan, R. and Zhou, X. (1997), 'Circulant preconditioners for Markov-modulated Poisson processes and their applications to manufacturing systems.', *SIAM Journal on Matrix Analysis and Applications* **18**, 464–481.

Duffie, D. and Gârleanu, N. (2001), 'Risk and valuation of collateralized debt obligations.', *Financial Analysts Journal* **57**, 41–62.

Duffie, D. and Singleton, K. J. (2003), *Credit Risk: Pricing, Measurement, and Management.*, Princeton University Press.

Fischer, W. and Meier-Hellstern, K. S. (1992), 'The Markov-modulated Poisson process cookbook', *Performance Evaluation* **18**, 149–171.

Frey, R. and McNeil, A. (2003), 'Dependent defaults in models of portfolio credit risk.', *Journal of Risk* **6**, 59–92.

Heffes, H. and Lucantoni, D. (1986), 'A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance', *IEEE J. Selected Areas in Commun.* **Vol. SAC-4, No. 6**, 856–868.

Huge, B. (2001), 'On defaultable claims and credit derivatives.', *PhD Thesis, Department of Statistics and Operations Research, University of Copenhagen* .

Hull, J. and White, A. (2004), 'Valuation of a CDO and an nth to default CDS without Monte Carlo simulation.', *Journal of Derivatives* **12**, 8–23.

Iscoe, I. and Kreinin, A. (2006), 'Recursive valuation of basket default swaps.', *Journal of Computational Finance* **9**, 95–116.

Lando, D. (2004), *Credit Risk Modeling*, Princeton Universiy Press.

Merino, S. and Nyfeler, M. (2002), 'Calculating portfolio loss.', *Risk* **Aug**, 82–86.

Mortensen, A. (2006), 'Semi-analytical valuation of basket credit derivatives in intensity-based models.', *Journal of Derivatives* **14**, 8–26.

Schonbucher, P. (2000), 'Factor models for portfolio credit risk.', *Working paper, Department of Statistics, University of Bonn* .

Schonbucher, P. (2003), *Credit Derivatives Pricing Models*, Wiley.

Vasicek, O. (2002), 'Loan portfolio value.', *Risk* **Dec**, 160–162.

(a) Computation time in 1-G models

$\bar{n}$	20	40	60	80	100
Number of states	42	82	122	162	202
(1) Solving the full Markov chain	1.10	4.06	8.86	15.03	23.33
(2) 1-G algorithm (IC: $m = 0$)	0.27	0.44	0.67	0.84	0.95
(3) 1-G algorithm (IC: $m \leq 1$)	0.45	0.81	1.28	1.64	2.30
(4) 1-G algorithm (IC: $m \leq 2$)	0.68	1.41	2.16	2.92	3.80

(b) Computation time in 2-G models

$(\bar{n}_1, \bar{n}_2)$	(10,10)	(20,20)	(30,30)	(40,40)	(50,50)
Number of states	242	882	1922	3362	5202
(1) Solving the full Markov chain	3.16	51.03	234.32	2126.22	5103.73
(2) 2-G algorithm (IC: $m_1, m_2 \leq 1$)	0.70	2.11	4.64	8.35	13.61
(3) 2-G algorithm (IC: $m_1, m_2 \leq 2$)	2.53	9.94	21.84	46.25	61.46
(4) 2-G algorithm (IC: $m_1, m_2 \leq 3$)	5.47	20.86	45.86	82.28	128.45

Table 1: Computation time (seconds) of default probability distribution $\mathsf{P}(N_t^n \leq m)$ for $t = 0 \sim 5$.

(a) 1-G model as in Figure 5, with parameter set $(\alpha, \beta, \gamma, \delta) = (0.1, 0.1, 0.01, 0.04)$. All Markov chains are solved by RK4 with $\Delta t = 0.001$, (b) 2-G model as in Figure 6, with parameter set $(\alpha, \beta, \gamma_1, \delta_1, \gamma_2, \delta_2) = (0.1, 0.1, 0.01, 0.005, 0.02, 0.01)$. All Markov chains are solved by RK4 with $\Delta t = 0.01$.

(a) Solving the Markov chain of the 1-G model in Figure 5 (a)

(n, m)	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$
(20, 0)	5.904740e-01	3.904021e-01	2.755539e-01	2.009936e-01	1.488797e-01
(20, 1)	8.654197e-01	6.799661e-01	5.354986e-01	4.288148e-01	3.472717e-01
(20, 2)	9.649288e-01	8.540239e-01	7.256721e-01	6.125636e-01	5.198219e-01
(20, 3)	9.929703e-01	9.454700e-01	8.592294e-01	7.604515e-01	6.670928e-01
(20, 4)	9.989039e-01	9.836049e-01	9.395851e-01	8.712034e-01	7.920440e-01
(20, 5)	9.998643e-01	9.960120e-01	9.785354e-01	9.411632e-01	8.865881e-01
(20, 6)	9.999864e-01	9.992077e-01	9.936814e-01	9.773494e-01	9.468737e-01
(20, 7)	9.999989e-01	9.998704e-01	9.984538e-01	9.926688e-01	9.787981e-01
(20, 8)	9.999999e-01	9.999824e-01	9.996847e-01	9.980068e-01	9.928260e-01
(20, 9)	1.000000e-00	9.999980e-01	9.999463e-01	9.995454e-01	9.979501e-01
(20,10)	1.000000e-00	9.999998e-01	9.999924e-01	9.999133e-01	9.995076e-01
(20,11)	1.000000e-00	1.000000e-00	9.999991e-01	9.999862e-01	9.999012e-01
(20,12)	1.000000e-00	1.000000e-00	9.999999e-01	9.999982e-01	9.999836e-01
(20,13)	1.000000e-00	1.000000e-00	1.000000e-00	9.999998e-01	9.999978e-01
(20,14)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	9.999998e-01
(20,15)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00
(20,16)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00
(20,17)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00
(20,18)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00
(20,19)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00
(20,20)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00

(b) Using the proposed 1-G algorithm (IC: $m = 0$)

(n, m)	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$
(20, 0)	5.904740e-01	3.904021e-01	2.755539e-01	2.009936e-01	1.488797e-01
(20, 1)	8.654197e-01	6.799661e-01	5.354986e-01	4.288148e-01	3.472717e-01
(20, 2)	9.649288e-01	8.540239e-01	7.256721e-01	6.125636e-01	5.198219e-01
(20, 3)	9.929703e-01	9.454700e-01	8.592294e-01	7.604515e-01	6.670928e-01
(20, 4)	9.989039e-01	9.836049e-01	9.395851e-01	8.712034e-01	7.920440e-01
(20, 5)	9.998643e-01	9.960120e-01	9.785354e-01	9.411632e-01	8.865881e-01
(20, 6)	9.999864e-01	9.992077e-01	9.936814e-01	9.773494e-01	9.468737e-01
(20, 7)	9.999989e-01	9.998704e-01	9.984538e-01	9.926688e-01	9.787981e-01
(20, 8)	9.999999e-01	9.999824e-01	9.996847e-01	9.980068e-01	9.928260e-01
(20, 9)	1.000000e-00	9.999980e-01	9.999463e-01	9.995454e-01	9.979501e-01
(20,10)	1.000000e-00	9.999999e-01	9.999924e-01	9.999133e-01	9.995076e-01
(20,11)	1.000000e-00	9.999999e-01	9.999991e-01	9.999861e-01	9.999011e-01
(20,12)	1.000000e-00	1.000000e-00	1.000000e-00	9.999983e-01	9.999836e-01
(20,13)	1.000000e-00	1.000000e-00	1.000000e-00	9.999997e-01	9.999978e-01
(20,14)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	9.999997e-01
(20,15)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00
(20,16)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00
(20,17)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00
(20,18)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00
(20,19)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00
(20,20)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00

Table 2: The distribution $\mathsf{P}(N_t^n \leq m)$ in the 1-G model where $n = \bar{n} = 20$, $m = 1 \sim 20$.
(a) solving the full Markov chain, (b) using 1-G algorithm (IC: $m = 0$).

(a) Solving the Markov chain of the 1-G model in Figure 5 (a)

(n, m)	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$
(60, 0)	2.906014e-01	1.355761e-01	6.718700e-02	3.348436e-02	1.669630e-02
(60, 1)	5.369080e-01	3.163036e-01	1.965105e-01	1.187822e-01	6.978918e-02
(60, 2)	7.083877e-01	4.571624e-01	3.263505e-01	2.290583e-01	1.545842e-01
(60, 3)	8.347858e-01	5.642999e-01	4.252769e-01	3.289669e-01	2.469917e-01
(60, 4)	9.190443e-01	6.623842e-01	5.010821e-01	4.057875e-01	3.272230e-01
(60, 5)	9.658666e-01	7.570330e-01	5.710216e-01	4.659751e-01	3.904500e-01
(60, 6)	9.875262e-01	8.406514e-01	6.444710e-01	5.206711e-01	4.418050e-01
(60, 7)	9.960102e-01	9.053957e-01	7.212137e-01	5.776554e-01	4.885248e-01
(60, 8)	9.988725e-01	9.492060e-01	7.954959e-01	6.397878e-01	5.360506e-01
(60, 9)	9.997161e-01	9.752972e-01	8.605885e-01	7.058353e-01	5.873450e-01
(60,10)	9.999359e-01	9.890884e-01	9.119939e-01	7.718885e-01	6.431544e-01
(60,11)	9.999869e-01	9.956086e-01	9.486361e-01	8.330704e-01	7.022844e-01
(60,12)	9.999976e-01	9.983846e-01	9.722918e-01	8.852266e-01	7.619923e-01
(60,13)	9.999996e-01	9.994552e-01	9.861775e-01	9.260719e-01	8.187160e-01
(60,14)	9.999999e-01	9.998310e-01	9.936170e-01	9.554677e-01	8.690442e-01
(60,15)	1.000000e-00	9.999517e-01	9.972681e-01	9.749362e-01	9.105894e-01
(60,16)	1.000000e-00	9.999872e-01	9.989148e-01	9.868240e-01	9.424466e-01
(60,17)	1.000000e-00	9.999969e-01	9.995994e-01	9.935297e-01	9.651318e-01
(60,18)	1.000000e-00	9.999993e-01	9.998624e-01	9.970310e-01	9.801401e-01
(60,19)	1.000000e-00	9.999998e-01	9.999559e-01	9.987263e-01	9.893727e-01
(60,20)	1.000000e-00	1.000000e-00	9.999868e-01	9.994889e-01	9.946593e-01

(b) Using the proposed 1-G algorithm (IC: $m = 0$)

(n, m)	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$
(60, 0)	2.906014e-01	1.355761e-01	6.718700e-02	3.348436e-02	1.669630e-02
(60, 1)	5.369080e-01	3.163036e-01	1.965105e-01	1.187822e-01	6.978918e-02
(60, 2)	7.083877e-01	4.571624e-01	3.263505e-01	2.290583e-01	1.545842e-01
(60, 3)	8.347858e-01	5.642999e-01	4.252769e-01	3.289669e-01	2.469917e-01
(60, 4)	9.190443e-01	6.623842e-01	5.010821e-01	4.057875e-01	3.272230e-01
(60, 5)	9.658666e-01	7.570330e-01	5.710216e-01	4.659751e-01	3.904500e-01
(60, 6)	9.875252e-01	8.406513e-01	6.444708e-01	5.206708e-01	4.418048e-01
(60, 7)	9.960234e-01	9.053959e-01	7.212164e-01	5.776590e-01	4.885278e-01
(60, 8)	9.987585e-01	9.492427e-01	7.954688e-01	6.397478e-01	5.360148e-01
(60, 9)	1.000000e-00	9.744343e-01	8.607411e-01	7.061541e-01	5.876720e-01
(60,10)	1.000000e-00	1.000000e-00	9.121080e-01	7.703162e-01	6.408931e-01
(60,11)	1.000000e-00	1.000000e-00	9.356308e-01	8.325292e-01	7.129219e-01
(60,12)	1.000000e-00	1.000000e-00	1.000000e-00	9.929421e-01	7.467941e-01
(60,13)	1.000000e-00	1.000000e-00	1.000000e-00	9.929421e-01	7.467941e-01
(60,14)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00
(60,15)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00
(60,16)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00
(60,17)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00
(60,18)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00
(60,19)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00
(60,20)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00

Table 3: The distribution $\mathsf{P}(N_t^n \leq m)$ in the 1-G model where $n = \bar{n} = 60$, $m = 1 \sim 20$.
(a) solving the full Markov chain, (b) using 1-G algorithm (IC: $m = 0$).

(a) Using the proposed 1-G algorithm (IC: $m \leq 1$)

(n, m)	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$
(60, 0)	2.906014e-01	1.355761e-01	6.718700e-02	3.348436e-02	1.669630e-02
(60, 1)	5.369080e-01	3.163036e-01	1.965105e-01	1.187822e-01	6.978918e-02
(60, 2)	7.083877e-01	4.571624e-01	3.263505e-01	2.290583e-01	1.545842e-01
(60, 3)	8.347858e-01	5.642999e-01	4.252769e-01	3.289669e-01	2.469917e-01
(60, 4)	9.190443e-01	6.623842e-01	5.010821e-01	4.057875e-01	3.272230e-01
(60, 5)	9.658666e-01	7.570330e-01	5.710216e-01	4.659751e-01	3.904500e-01
(60, 6)	9.875262e-01	8.406514e-01	6.444710e-01	5.206711e-01	4.418050e-01
(60, 7)	9.960101e-01	9.053957e-01	7.212137e-01	5.776553e-01	4.885248e-01
(60, 8)	9.988736e-01	9.492055e-01	7.954967e-01	6.397887e-01	5.360514e-01
(60, 9)	9.997095e-01	9.753083e-01	8.605849e-01	7.058285e-01	5.873361e-01
(60,10)	9.999423e-01	9.889263e-01	9.119813e-01	7.719256e-01	6.432280e-01
(60,11)	1.000000e-00	9.974042e-01	9.490961e-01	8.330158e-01	7.018078e-01
(60,12)	1.000000e-00	9.974042e-01	9.667579e-01	8.837122e-01	7.643487e-01
(60,13)	1.000000e-00	1.000000e-00	1.000000e-00	9.481197e-01	8.108047e-01
(60,14)	1.000000e-00	1.000000e-00	1.000000e-00	9.481197e-01	8.784275e-01
(60,15)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	9.708124e-01
(60,16)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	9.708124e-01
(60,17)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	9.708124e-01
(60,18)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00
(60,19)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00
(60,20)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00

(b) Using the proposed 1-G algorithm (IC: $m \leq 2$)

(n, m)	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$
(60, 0)	2.906014e-01	1.355761e-01	6.718700e-02	3.348436e-02	1.669630e-02
(60, 1)	5.369080e-01	3.163036e-01	1.965105e-01	1.187822e-01	6.978918e-02
(60, 2)	7.083877e-01	4.571624e-01	3.263505e-01	2.290583e-01	1.545842e-01
(60, 3)	8.347858e-01	5.642999e-01	4.252769e-01	3.289669e-01	2.469917e-01
(60, 4)	9.190443e-01	6.623842e-01	5.010821e-01	4.057875e-01	3.272230e-01
(60, 5)	9.658666e-01	7.570330e-01	5.710216e-01	4.659751e-01	3.904500e-01
(60, 6)	9.875262e-01	8.406514e-01	6.444710e-01	5.206711e-01	4.418050e-01
(60, 7)	9.960102e-01	9.053957e-01	7.212137e-01	5.776554e-01	4.885248e-01
(60, 8)	9.988725e-01	9.492060e-01	7.954959e-01	6.397878e-01	5.360506e-01
(60, 9)	9.997163e-01	9.752971e-01	8.605886e-01	7.058356e-01	5.873453e-01
(60,10)	9.999355e-01	9.890909e-01	9.119938e-01	7.718869e-01	6.431519e-01
(60,11)	9.999800e-01	9.955776e-01	9.486280e-01	8.330767e-01	7.023026e-01
(60,12)	1.000000e-00	9.986910e-01	9.724219e-01	8.852424e-01	7.618914e-01
(60,13)	1.000000e-00	9.986910e-01	9.849055e-01	9.255562e-01	8.191206e-01
(60,14)	1.000000e-00	1.000000e-00	1.000000e-00	9.606318e-01	8.681914e-01
(60,15)	1.000000e-00	1.000000e-00	1.000000e-00	9.606318e-01	9.083995e-01
(60,16)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	9.693243e-01
(60,17)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	9.693243e-01
(60,18)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	9.693243e-01
(60,19)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00
(60,20)	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00	1.000000e-00

Table 4: The distribution $\mathsf{P}(N_t^n \leq m)$ in the 1-G model where $n = \bar{n} = 60$, $m = 1 \sim 20$.
(a) using 1-G algorithm (IC: $m \leq 1$), (b) using 1-G algorithm (IC: $m \leq 2$).

Parameter sets in 2-G models

Parameter set	α	β	γ_1	δ_1	γ_2	δ_2	$\bar{\lambda}$
A	0.1	0.1	0.01	0.02	0.01	0.02	0.015
B	0.1	0.1	0.01	0.02	0.005	0.025	0.015
C	0.1	0.1	0.01	0.02	0.00	0.03	0.015

Table 5: Three parameter sets in 2-G models where $\bar{n}_1 = \bar{n}_2 = 25$.