J. Appl. Prob. **43,** 755–766 (2006)
Printed in Israel

A MATRIX-ANALYTIC APPROACH TO THE *N*-PLAYER RUIN PROBLEM

YVIK C. SWAN * ** AND
F. THOMAS BRUSS,* *Université Libre de Bruxelles*

Abstract

Consider N players, respectively owning $x_1, x_2, \ldots, x_N$ monetary units, who play a sequence of games, winning from and losing to each other integer amounts according to fixed rules. The sequence stops as soon as (at least) one player is ruined. We are interested in the ruin process of these N players, i.e. in the probability that a given player is ruined first, and also in the expected ruin time. This problem is called the N-player ruin problem. In this paper, the problem is set up as a multivariate absorbing Markov chain with an absorbing state corresponding to the ruin of each player. This is then discussed in the context of phase-type distributions where each phase is represented by a vector of size N and the distribution has as many absorbing points as there are ruin events. We use this modified phase-type distribution to obtain an explicit solution to the N-player problem. We define a partition of the set of transient states into different levels, and on it give an extension of the folding algorithm (see Ye and Li (1994)). This provides an efficient computational procedure for calculating some of the key measures.

Keywords: Markov process; phase-type distribution; hitting time; hitting probability; folding algorithm

2000 Mathematics Subject Classification: Primary 60J10
Secondary 60J20

1. Introduction

Suppose that N players are going to play a sequence of games, during each of which they will either win or lose an integer amount of money. We are interested in the probability that a given player is ruined first. If the games are played according to fixed rules, then this probability will only depend – through the structure of each game – on the initial distribution of capital, say $\{x_1, \ldots, x_N\}$. Define Σ as the total capital (i.e. $\Sigma = x_1 + \cdots + x_N$). We are studying games in which, at each stage, the total available capital remains constant and equal to Σ. Therefore, we choose to represent the flow of capitals as a N-dimensional random walk $\{J_t\} \in \mathbb{Z}^N$ in the lattice

$$\{(X_1, \ldots, X_N) \in \mathbb{Z}^N : X_1 + \cdots + X_N = \Sigma\}. \tag{1.1}$$

Each state of the random walk represents a possible distribution of wealth among the N players. The ruin of a player occurs only when the random walk reaches a state in which one (or more) of the coordinates satisfies $X_i \leq 0$.

Remark 1.1. Note that we have not specified that the only possible flows are those between capitals which lead to ruin without debt. Since we use no martingale argument at the core of

Received 22 February 2005; revision received 20 April 2006.
* Postal address: Département de Mathématique et ISRO, Université Libre de Bruxelles, Campus Plaine, CP 210, B-1050 Brussels, Belgium.
** Email address: yvswan@ulb.ac.be

the present paper, the stronger requirement that $X_i = 0$ at ruin time is not needed, as it is in, e.g. Engel (1993), Stirzaker (1994), and Bruss *et al*. (2003).

1.1. Classification of states

By the *interior points* of the lattice (1.1) we will refer to the collection of points all of whose coordinates are positive, i.e. the subset of (1.1) given by

$$\{X_1 + \cdots + X_N = \Sigma \colon X_i > 0,\ i = 1, \ldots, N\}. \tag{1.2}$$

Clearly, all these points are transient for the process J_t. We also notice that there are as many points in the set (1.2) as there are vectors of $\mathbb{R}^N$ whose components are strictly positive integers summing to Σ. This is the same as the number of ways of distributing Σ indistinguishable balls between N urns, each urn containing at least one ball; i.e. there are

$$p = \binom{\Sigma + N - 1 - N}{\Sigma - N} = \binom{\Sigma - 1}{N - 1}$$

transient states for J_t.

By the *edge* of the lattice (1.1) we will refer to the collection of points corresponding to the ruin of at least one player, i.e. the subsimplex of (1.1) made up of all the points which have at least one nonpositive coordinate. All these points are clearly absorbing for J_t, since we suppose that once a player is ruined the game is finished.

Clearly, all the points at which the same combination of coordinates is nonpositive define the same ruin events for the N-player game. Therefore, there are as many inequivalent disjoint edge points as there are different ruin events. Since there are as many ruin events as there are ways of choosing subgroups of sizes 1 to $N - 1$ among the N initial players, we see that, in the context of the N-player ruin problem, there are in fact only $2^N - 2$ inequivalent edge points. To each of the possible $2^N - 2$ sets of indices $J = \{i_1, \ldots, i_k\}$ we associate the absorbing state

$$O_J = \{X_1 + \cdots + X_N = \Sigma \colon X_j \le 0 \text{ for all } j \in J \text{ and } X_l > 0 \text{ for all } l \notin J\},$$

and together these states form a partition of the set of edge points.

Example 1.1. Let us consider the case in which $N = 3$ and $\Sigma = 7$. The lattice defined by (1.1) can be laid out as a triangle in the plane with 36 points, 15 of which are interior points. We partition the 21 edge points into six absorbing states labeled O_1 to O_6, as in Figure 1. Each of these states defines a different ruin event (the points O_2, O_4, and O_6 correspond to the ruin of a single player and O_1, O_3, and O_5 correspond to the three possible combinations of two players being ruined at the same time).

The way in which the points are ordered is arbitrary, but the choice of an order will obviously determine the shape of the transition matrix of the process.

Remark 1.2. In the rest of the text, we suppose that the

$$p = \binom{\Sigma - 1}{N - 1}$$

transient states as well as the $n = 2^N - 2$ absorbing states are ordered in an unambiguous way.

https://doi.org/10.1239/jap/1158784944 Published online by Cambridge University Press

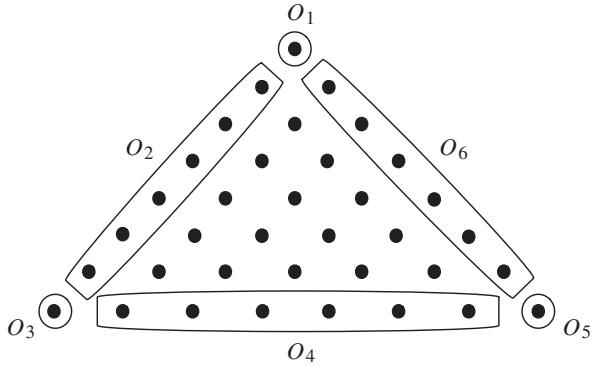


FIGURE 1: The lattice defined by (1.1) for $N = 3$ and $\Sigma = 7$.

2. A solution to the N-player problem

2.1. Notation

Let $n = 2^N - 2$. We respectively denote by $\boldsymbol{r}^1, \ldots, \boldsymbol{r}^n$ the (column) vectors of one-step probabilities of absorption in $O_1, \ldots, O_n$, i.e. $r_i^\zeta = \mathrm{P}[J_t \in O_\zeta \mid J_{t-1} = i]$, $\zeta = 1, \ldots, n$.

We denote by $\boldsymbol{T}$ the transition matrix corresponding to the transient states and use $\boldsymbol{I}$ for the $n \times n$ identity matrix. The transition matrix of the random walk corresponding to the ruin problem can then be written as

$$\boldsymbol{P} = \begin{pmatrix} \boldsymbol{T} & \boldsymbol{r}^1 \cdots \boldsymbol{r}^n \\ \boldsymbol{0} & \boldsymbol{I} \end{pmatrix}. \tag{2.1}$$

As stated before, the outcome of the game depends on the initial distribution of wealth, i.e. on the initial distribution of the random walk. We therefore introduce the (row) vector $\boldsymbol{\tau}$, which has zero entries everywhere except the entry corresponding to the position of the initial state, which is set to 1.

Proposition 2.1. 1. *Since $\boldsymbol{P}$ is stochastic, we have the relationship $\boldsymbol{r}^1 + \cdots + \boldsymbol{r}^n + \boldsymbol{T}\boldsymbol{1} = \boldsymbol{1}$, where $\boldsymbol{1} = (1, \ldots, 1)^\top$, '$^\top$' denoting transpose.*

2. *From the structure of $\boldsymbol{P}$ we immediately obtain, for $k \geq 1$,*

$$\boldsymbol{P}^k = \begin{pmatrix} \boldsymbol{T}^k & (\boldsymbol{I} + \cdots + \boldsymbol{T}^{k-1})\boldsymbol{r}^1 \cdots (\boldsymbol{I} + \cdots + \boldsymbol{T}^{k-1})\boldsymbol{r}^n \\ \boldsymbol{0} & \boldsymbol{I} \end{pmatrix}.$$

2.2. PH^m random variables

The methods and terminology we are going to apply are borrowed from matrix analysis theory. These methods have been gradually developed from the foundations laid by Marcel Neuts, who coined the terminology of 'matrix-geometric distributions' and 'phase-type processes' (see, for example, Neuts (1975), (1978), (1981)). More specifically, we use different results, of Latouche and Ramaswami (1999), in which those authors describe applications and algorithms derived from and for matrix analysis theory.

Take $\boldsymbol{P}$ to be any stochastic matrix of the form (2.1) and $\boldsymbol{\tau}$, as above, to be the initial distribution of a random walk on a grid with m transient states and n absorbing states. Following Latouche and Ramaswami (1999), we define the concept of PH^m distributions.

https://doi.org/10.1239/jap/1158784944 Published online by Cambridge University Press

Definition 2.1. The distribution of the time, X, until absorption of a random walk with transition matrix $\boldsymbol{P}$ and initial distribution $\boldsymbol{\tau}$ is called a PH^m distribution with representation $(\boldsymbol{\tau}, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_n, \boldsymbol{T})$. We will write $X \sim \mathrm{PH}^m(\boldsymbol{\tau}, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_n, \boldsymbol{T})$ to indicate this.

Given a PH^m random variable X, we can define a random walk (which we will consistently denote by J_t) on a grid with m transient states and n absorbing states such that X represents the time until absorption of J_t.

However, it must be noted that there can be more than one such random walk and, therefore, more than one choice of $(\boldsymbol{\tau}, \boldsymbol{r}^1, \ldots, \boldsymbol{r}^n, \boldsymbol{T})$. This is the reason why we refer to $(\boldsymbol{\tau}, \boldsymbol{r}^1, \ldots, \boldsymbol{r}^n, \boldsymbol{T})$ as a *representation* of X. This ambiguity will not cause any problems, since in practice the random walk is a-priori defined by the structure of the ruin problem.

2.2.1. *The distribution.* In the next two propositions we study the law of a PH^m random variable. These results are a generalization of known results on PH random variables which can be found in Latouche (1989) and Latouche and Ramaswami (1999, p. 49), and are therefore stated without proof.

Proposition 2.2. *Let X follow a $\mathrm{PH}^m(\boldsymbol{\tau}, \boldsymbol{r}^1, \ldots, \boldsymbol{r}^n, \boldsymbol{T})$ distribution starting in a state α, $1 \leq \alpha \leq m$ (which is not a border point). The distribution of X conditional on this initial state is then given by*

(i) $\mathrm{P}_\alpha[X = 0] = 0$;

(ii) $\mathrm{P}_\alpha[X = k] = \boldsymbol{\tau}\boldsymbol{T}^{k-1}\mathbf{1} - \boldsymbol{\tau}\boldsymbol{T}^k\mathbf{1}$;

(iii) $F_X(k) := \mathrm{P}_\alpha[X \leq k] = \boldsymbol{\tau}(\boldsymbol{I} - \boldsymbol{T}^k)\mathbf{1}$.

Now, it can be shown (see, for example, Householder (1964, p. 54)) that the series $\sum_{n\geq 0} \boldsymbol{M}^n$ converges if and only if the spectral radius, $\rho(\boldsymbol{M})$, of $\boldsymbol{M}$ is strictly less than 1, in which case $\sum_{n\geq 0} \boldsymbol{M}^n = (\boldsymbol{I} - \boldsymbol{M})^{-1}$. Moreover, for any strict submatrix $\boldsymbol{N}$ of a nondegenerate matrix $\boldsymbol{M}$, we have $\rho(\boldsymbol{N}) < \rho(\boldsymbol{M})$. Since all stochastic matrices are nondegenerate (i.e. have no line of zeros) and have spectral radius 1, this means that we can use the identity $\sum_{n\geq 0} \boldsymbol{M}^n = (\boldsymbol{I} - \boldsymbol{M})^{-1}$ for all strict submatrices $\boldsymbol{M}$ of stochastic matrices.

Using this and Proposition 2.2, we obtain the second proposition.

Proposition 2.3. *The expected duration of the three-player game is given by*

$$\mathrm{E}[X] = \boldsymbol{\tau}(\boldsymbol{I} - \boldsymbol{T})^{-1}\mathbf{1},$$

where $\boldsymbol{\tau}$ is the initial distribution. Furthermore, the probability that the random walk is absorbed in state O_ζ, $\zeta = 1, \ldots, n$, starting from any point α interior to the grid is given by

$$\mathrm{P}_\alpha[\text{ruin in } \zeta] = \boldsymbol{\tau}(\boldsymbol{I} - \boldsymbol{T})^{-1}\boldsymbol{r}^\zeta. \tag{2.2}$$

Solving the ruin problem means finding the probabilities of ruin in any absorbing state O_l starting from any point α in the grid. It therefore means solving (2.2), where the transition matrix is determined by the structure of the game.

3. Ruin by folding

The algorithm we shall present and explain in this section is based on the *folding algorithm* developed by Ye and Li (1994) for the study of quasi-birth-and-death processes. We need the following preliminaries.

https://doi.org/10.1239/jap/1158784944 Published online by Cambridge University Press

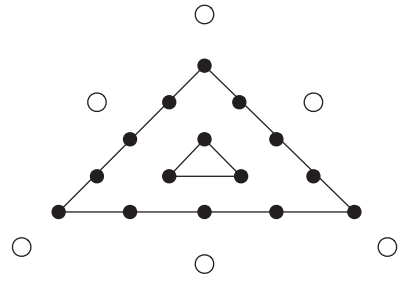


FIGURE 2: The lattice defined by (1.1) for $N = 3$ and $\Sigma = 7$ can be divided into 2 levels as shown.

3.1. Preparation

Definition 3.1. We partition the lattice $\{X_1 + \cdots + X_N = \Sigma\}$ into *levels* $L_j,\ j \geq 1$, each level being defined as the set of points at least one of whose coordinates (per point) is equal to j with all the other coordinates being greater than j, i.e.

$$L_j = \{X_1 + \cdots + X_N = \Sigma \colon X_i \geq j \text{ for all } i \text{ and } X_{j_0} = j \text{ for some } j_0\}.$$

Example 3.1. From Example 1.1 we know that, for $N = 3$ and $\Sigma = 7$, we have a random walk on a set with 15 transient interior points and six absorbing edge points. We can rearrange these points into two levels, which are represented in Figure 2 by concentric equilateral triangles. The exterior level (which is made up of the $n = 2^N - 2$ absorbing points) is not counted as a true level.

The next proposition serves to count the number of levels and the number of states on each level.

Proposition 3.1. *Let* $\Sigma = Nk + l,\ 0 \leq l \leq N - 1$.

1. *The grid is divided into k levels (without counting the ruin level).*

2. *Each level* $L_j,\ j < k$, *has*

$$b_j = \binom{\Sigma - Nj + N - 1}{\Sigma - Nj} - \binom{\Sigma - Nj - 1}{\Sigma - N(j+1)} \tag{3.1}$$

points. On L_k *the number of points is given by*

$$b_k = \binom{l + N - 1}{l}.$$

Proof. We first prove statement 1. To do so, let us fix some integer j and consider level L_j. We want to find a necessary and sufficient condition for the level to be nonempty. Since $X_i \geq j$ for all i and at least one player has capital $X = j$, we see that the condition $\{X_1 + \cdots + X_N = \Sigma\}$ will be satisfied if and only if there exist $N - 1$ nonnegative integers $\tilde{x}_i,\ i = 1, \ldots, N - 1$, such that $\tilde{x}_1 + \cdots + \tilde{x}_{N-1} = \Sigma - Nj$. These $\tilde{x}_i$ will exist if and only if $\Sigma - Nj \geq 0$. Writing Σ uniquely as $Nk + l$ (Euclidean division of Σ by N) in the last inequality, we obtain $N(j - k) \leq l$ as a necessary and sufficient condition for the nonemptiness of level L_j. Since $0 \leq l \leq N - 1$, this is satisfied if and only if $0 \leq j \leq k$. When j is equal to 0, we are at level 0; therefore, there are only k different nonempty nonruin levels.

https://doi.org/10.1239/jap/1158784944 Published online by Cambridge University Press

To prove the second part of the proposition, we assume that $1 \le j \le k-1$. Clearly, the number of points on level L_j, namely $\#(L_j)$, will be given by

$$\#(L_j) = \#\{\tilde{x}_i \ge j,\ i = 1, \dots, N\} - \#\{\tilde{x}_i \ge j+1,\ i = 1, \dots, N\}.$$

Equation (3.1) is then obtained by using the same urn model arguments as those used in the classification of states in Section 1.

This argument does not apply when we are looking for the number of points on the last level, L_k. However, subtracting from the total number of points the sum of the numbers of points on levels of lower order yields the result.

3.2. An imbedded quasi-birth-and-death process

For a fixed ruin problem it is clear that level transitions of the random walk will only be possible from a level L_j to certain other levels L_{j+k}, independently of j. We are going to exploit this property. However, to make matters (and notation) less cumbersome, we will restrict ourselves to ruin games in which the random walk does not make transitions of more than one level at a time. The argument we are going to give will be in principle adaptable to different situations which arise in practice. This restriction does not impair the value of the result, since it is satisfied by most ruin games studied in the literature (see, for instance, Stirzaker (1994) and Bruss *et al*. (2003)).

3.2.1. *The transition matrix of the game.* Every point α in the grid is defined by two coordinates: we write $\alpha = (j, l)$, where j, $1 \le j \le k$, represents the level on which the point lies and l, $1 \le l \le b_j$, represents the position of this point on the level. With the restriction just imposed, it is clear that from L_j the random walk can only stay on L_j, go up one level, or go down one level. Thus, for each level L_j, $j \ge 1$, we define the submatrices $\boldsymbol{A}_0^{(j)}$, $\boldsymbol{A}_1^{(j)}$, and $\boldsymbol{A}_2^{(j)}$ entrywise as follows:

$$(A_0^{(j)})_{\alpha\beta} = \mathrm{P}[J_{t+1} = \beta \in L_{j+1} \mid J_t = \alpha \in L_j], \quad \text{corresponding to} \quad L_j \to L_{j+1},$$

$$(A_1^{(j)})_{\alpha\beta} = \mathrm{P}[J_{t+1} = \beta \in L_j \mid J_t = \alpha \in L_j], \quad \text{corresponding to} \quad L_j \to L_j,$$

$$(A_2^{(j)})_{\alpha\beta} = \mathrm{P}[J_{t+1} = \beta \in L_{j-1} \mid J_t = \alpha \in L_j], \quad \text{corresponding to} \quad L_j \to L_{j-1}.$$

Each of these matrices represents an admissible transition between levels (as described). Since transitions to the absorbing states are only possible from the exterior level, L_1, if we order the levels in decreasing order then the transition matrix $\boldsymbol{P}$ becomes

$$\boldsymbol{P} = \left[\begin{array}{cccccc|c} \boldsymbol{A}_1^{(k)} & \boldsymbol{A}_2^{(k)} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{A}_0^{(k-1)} & \boldsymbol{A}_1^{(k-1)} & \boldsymbol{A}_2^{(k-1)} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_0^{(k-2)} & \boldsymbol{A}_1^{(k-2)} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{A}_1^{(2)} & \boldsymbol{A}_2^{(2)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{A}_0^{(1)} & \boldsymbol{A}_1^{(1)} & \boldsymbol{R} \\ \hline \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{array}\right], \tag{3.2}$$

where the matrix $\boldsymbol{R}$ is given (entrywise) by

$$R_{\alpha\zeta} = \mathrm{P}[J_{t+1} \in O_\zeta \mid J_t = \alpha \in L_1], \qquad \zeta = 1, \dots, n.$$

https://doi.org/10.1239/jap/1158784944 Published online by Cambridge University Press

Remark 3.1. The transient submatrix of (3.2) has the form of the transition matrix of an inhomogeneous quasi-birth-and-death process. The notation $\boldsymbol{A}_0^{(j)}$, $\boldsymbol{A}_1^{(j)}$, $\boldsymbol{A}_2^{(j)}$ has been chosen to be consistent with the notation used in the standard text books on this subject. See Latouche and Ramaswami (1999, p. 130) for more details.

Example 3.2. Let us consider a three-player game in which the admissible transitions are given by

$$(a, b, c) \to \begin{cases} (a-1, b+1, c), \\ (a, b-1, c+1), \\ (a+1, b, c-1), \end{cases}$$

each occurring with the same probability. We take $\Sigma = 10$. The random walk of the game then runs on a set with three levels of, respectively, 3, 12, and 21 points and a ruin level of 6 points. We fix the counting of the points on each level to start (in our standard, triangular lattice) from the highest point of the equilateral triangle and proceed clockwise. As an illustration we write the level transition matrices starting from L_3 and L_2:

$$\boldsymbol{A}_1^3 = \begin{bmatrix} 0 & 0 & \frac{1}{3} \\ \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \end{bmatrix}, \qquad \boldsymbol{A}_2^3 = \begin{bmatrix} 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 \end{bmatrix},$$

$$\boldsymbol{A}_0^2 = \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3} \\ \frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad \boldsymbol{A}_1^2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} \\ \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 \end{bmatrix},$$

$$\boldsymbol{A}_2^2 = \begin{bmatrix} 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 \end{bmatrix}.$$

For obvious space reasons we do not write down the last three submatrices (i.e. $\boldsymbol{A}_i^{(1)}$, $i = 0, 1, 2$). They have the same structure as those displayed.

https://doi.org/10.1239/jap/1158784944 Published online by Cambridge University Press

For $j = 1, \ldots, k$ and $l = 1, \ldots, n$, we define the *ruin vectors* $\boldsymbol{r}_l^{(j)}$, whose entries are given by $(r_l^{(j)})_i = \mathrm{P}[\text{final ruin in } O_l \text{ starting from } (j, i)]$, $i = 1, \ldots, b_j$. Solving the ruin problem means finding the values of the ruin vectors.

Proposition 3.2. *For $l = 1, \ldots, n$,*

$$\begin{aligned} \boldsymbol{r}_l^{(1)} &= (\boldsymbol{I} - \boldsymbol{A}_1^{(1)})^{-1}(\boldsymbol{R}_{\bullet l} + \boldsymbol{A}_0^{(1)}\boldsymbol{r}_l^{(2)}), \\ \boldsymbol{r}_l^{(j)} &= (\boldsymbol{I} - \boldsymbol{A}_1^{(j)})^{-1}(\boldsymbol{A}_2^{(j)}\boldsymbol{r}_l^{(j-1)} + \boldsymbol{A}_0^{(j)}\boldsymbol{r}_l^{(j+1)}), \qquad j = 1, \ldots, k-1, \\ \boldsymbol{r}_l^{(k)} &= (\boldsymbol{I} - \boldsymbol{A}_1^{(k)})^{-1}(\boldsymbol{A}_2^{(k)}\boldsymbol{r}_l^{(k-1)}), \end{aligned} \tag{3.3}$$

where $\boldsymbol{R}_{\bullet l} = (R_{1l}, \ldots, R_{b_1 l})^\top$, a column vector whose components are entries of the matrix $\boldsymbol{R}$.

Proof. We have

$$\begin{aligned} (r_l^{(j)})_i &= \mathrm{P}[\text{final ruin in } O_l \text{ starting from } (j, i)] \\ &= \sum_{t=1}^{b_j} (A_1^{(j)})_{it}(r_l^{(j)})_t + \sum_{t=1}^{b_{j+1}} (A_0^{(j)})_{it}(r_l^{(j+1)})_t + \sum_{t=1}^{b_{j-1}} (A_2^{(j)})_{it}(r_l^{(j-1)})_t, \end{aligned}$$

i.e.

$$\begin{aligned} \boldsymbol{r}_l^{(j)} &= \boldsymbol{A}_1^{(j)}\boldsymbol{r}_l^{(j)} + \boldsymbol{A}_0^{(j)}\boldsymbol{r}_l^{(j+1)} + \boldsymbol{A}_2^{(j)}\boldsymbol{r}_l^{(j-1)} \\ &= (\boldsymbol{A}_1^{(j)})^2\boldsymbol{r}_l^{(j)} + (\boldsymbol{I} + \boldsymbol{A}_1^{(j)})(\boldsymbol{A}_0^{(j)}\boldsymbol{r}_l^{(j+1)} + \boldsymbol{A}_2^{(j)}\boldsymbol{r}_l^{(j-1)}) \\ &= \lim_{K\to\infty}\left[(\boldsymbol{A}_1^{(j)})^K\boldsymbol{r}_l^{(j)} + \sum_{t=0}^{K}(\boldsymbol{A}_1^{(j)})^t(\boldsymbol{A}_0^{(j)}\boldsymbol{r}_l^{(j+1)} + \boldsymbol{A}_2^{(j)}\boldsymbol{r}_l^{(j-1)})\right]. \end{aligned}$$

From arguments given previously, we know that $\lim_{K\to\infty}(\boldsymbol{A}_1^{(j)})^K = \boldsymbol{0}$ and that $\sum_{t=0}^{\infty}(\boldsymbol{A}_1^{(j)})^t = (\boldsymbol{I} - \boldsymbol{A}_1^{(j)})^{-1}$.

The two other equations are proved using similar arguments.

Equations like (3.3) and those appearing later in the paper admit an explicit interpretation. To facilitate the understanding of this interpretation, we give a detailed explanation of the second expression in (3.3), i.e.

$$\boldsymbol{r}_l^{(j)} = (\boldsymbol{I} - \boldsymbol{A}_1^{(j)})^{-1}(\boldsymbol{A}_2^{(j)}\boldsymbol{r}_l^{(j-1)} + \boldsymbol{A}_0^{(j)}\boldsymbol{r}_l^{(j+1)}),$$

for some fixed, intermediate $j \in \{2, \ldots, k-1\}$, as follows.

We condition on the first level visited after leaving L_j. Once the random walk leaves L_j it has two options: either it goes up to L_{j+1} (which happens with 'probability' $\boldsymbol{A}_0^{(j)}$) and is absorbed in O_l from there (which happens with 'probability' $\boldsymbol{r}_l^{(j+1)}$), explaining the term $\boldsymbol{A}_0^{(j)}\boldsymbol{r}_l^{(j+1)}$; or it goes down to L_{j-1} (which happens with 'probability' $\boldsymbol{A}_2^{(j)}$) and is absorbed in O_l from there (which happens with 'probability' $\boldsymbol{r}_l^{(j-1)}$), explaining the term $\boldsymbol{A}_2^{(j)}\boldsymbol{r}_l^{(j-1)}$. Before leaving L_j and being ruined, there is a probability that the random walk first returns a number of times to L_j. This happens with 'probability' $\sum_{i\geq 1}(\boldsymbol{A}_1^{(j)})^i$, explaining the presence of the term $(\boldsymbol{I} - \boldsymbol{A}_1^{(j)})^{-1}$.

Hence, we may summarize the meaning of these equations by saying that absorption probabilities are mutually expressed in terms of 'neighboring' absorption probabilities.

https://doi.org/10.1239/jap/1158784944 Published online by Cambridge University Press

3.3. Ruin probability by folding

Suppose that we are solving the N-player ruin problem in a system with k levels. Let us consider the restriction of J_t to the set of even-numbered levels, $L_2, L_4, \ldots, L_{2\lfloor k/2 \rfloor}$, where $\lfloor \cdot \rfloor$ is the integer-part function. This yields a new random walk, by which transitions from level L_{2i} to level L_{2j} are made at the first hitting time of L_{2j} from L_{2i} of the initial random walk J_t. From the viewpoint of J_t, we consider the transitions from state α to state β if and only if α and β are two states on the same level or if β is the first state on an even-numbered level that the random walk visits after leaving α. This new random walk runs on a grid with $\lfloor k/2 \rfloor$ levels.

Now, solving the ruin equations (2.2) for this smaller system, i.e. with smaller matrices, will yield the ruin vectors for even levels. From Proposition 3.2 we see that we can then apply (3.3) to determine the ruin vectors for the whole system.

This restriction of the initial random walk to a new set with half the number of levels is what we call the *folding* of the process. The new random walk is the *folded* random walk.

The transition matrix of the folded random walk is given by the following proposition (which is proved using arguments similar to those used to prove Proposition 3.2).

Proposition 3.3. *Take $s = \lfloor k/2 \rfloor$. If $s \geq 3$ then define*

$$\begin{aligned} \boldsymbol{A}_0^{\star(j)} &= \boldsymbol{A}_0^{(2j)}(\boldsymbol{I} - \boldsymbol{A}_1^{(2j+1)})^{-1}\boldsymbol{A}_0^{(2j+1)}, \\ \boldsymbol{A}_1^{\star(j)} &= \boldsymbol{A}_1^{(2j)} + \boldsymbol{A}_2^{(2j)}(\boldsymbol{I} - \boldsymbol{A}_1^{(2j-1)})^{-1}\boldsymbol{A}_0^{(2j-1)} + \boldsymbol{A}_0^{(2j)}(\boldsymbol{I} - \boldsymbol{A}_1^{(2j+1)})^{-1}\boldsymbol{A}_2^{(2j+1)}, \\ \boldsymbol{A}_2^{\star(j)} &= \boldsymbol{A}_2^{(2j)}(\boldsymbol{I} - \boldsymbol{A}_1^{(2j-1)})^{-1}\boldsymbol{A}_2^{(2j-1)}, \end{aligned} \tag{3.4}$$

for $j = 2, \ldots, s-1$ if k is even and for $j = 2, \ldots, s$ if k is odd. Also, if k is even, define

$$\begin{aligned} \boldsymbol{A}_2^{\star(s)} &= \boldsymbol{A}_2^{(k)}(\boldsymbol{I} - \boldsymbol{A}_1^{(k-1)})^{-1}\boldsymbol{A}_2^{(k-1)}, \\ \boldsymbol{A}_1^{\star(s)} &= \boldsymbol{A}_1^{(k)} + \boldsymbol{A}_2^{(k-1)}(\boldsymbol{I} - \boldsymbol{A}_1^{(k-1)})^{-1}\boldsymbol{A}_0^{(k-1)}. \end{aligned}$$

Finally, define

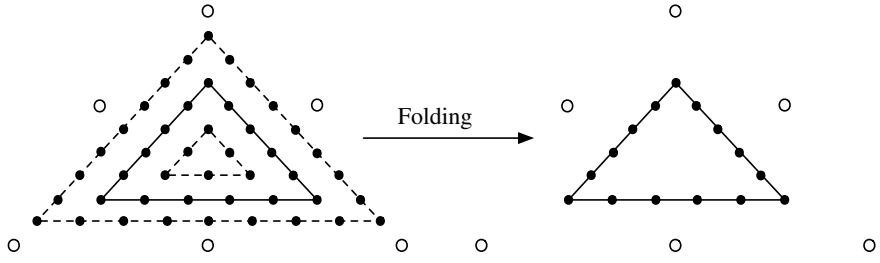
$$\begin{aligned} \boldsymbol{A}_1^{\star(1)} &= \boldsymbol{A}_1^{(2)} + \boldsymbol{A}_0^{(2)}(\boldsymbol{I} - \boldsymbol{A}_1^{(3)})^{-1}\boldsymbol{A}_2^{(3)} + \boldsymbol{A}_2^{(2)}(\boldsymbol{I} - \boldsymbol{A}_1^{(1)})^{-1}\boldsymbol{A}_0^{(1)}, \\ \boldsymbol{A}_0^{\star(1)} &= \boldsymbol{A}_0^{(2)}(\boldsymbol{I} - \boldsymbol{A}_1^{(3)})^{-1}\boldsymbol{A}_2^{(3)}, \\ \boldsymbol{R}^{\star} &= \boldsymbol{A}_0^{(2)}(\boldsymbol{I} - \boldsymbol{A}_1^{(1)})^{-1}\boldsymbol{R}. \end{aligned}$$

With this notation, the transition matrix of the process restricted to even levels is given by

$$\boldsymbol{P} = \left[\begin{array}{cccccc|c} \boldsymbol{A}_2^{\star(s)} & \boldsymbol{A}_0^{\star(s)} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{A}_0^{\star(s-1)} & \boldsymbol{A}_1^{\star(s-1)} & \boldsymbol{A}_2^{\star(s-1)} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_0^{\star(s-2)} & \boldsymbol{A}_1^{\star(s-2)} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{A}_1^{\star(2)} & \boldsymbol{A}_2^{\star(2)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{A}_0^{\star(2)} & \boldsymbol{A}_1^{\star(2)} & \boldsymbol{R}^{\star} \\ \hline \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{array}\right].$$

Example 3.3. If $N = 3$ and $\Sigma = 9$ then we have three levels (see Figure 3(a)). After folding, we are left with a system with one level (see Figure 3(b)).

https://doi.org/10.1239/jap/1158784944 Published online by Cambridge University Press



(a) Three levels (b) One level

FIGURE 3: Folding of the lattice defined by (1.1) for $N = 3$ and $\Sigma = 9$.

If, after the initial folding, the ruin problem runs on a set which still has more than two levels, we can repeat the folding and obtain a random walk on a smaller set. The folding can be repeated over and over until the set on which the random walk runs has only one or two levels left. The ruin equations (2.2) can be solved on this set, and with these results we can recursively apply (3.3) to obtain the ruin vectors for the whole system.

3.3.1. *Recursive algorithm.* From the last proposition, it is clear how we can devise a recursive algorithm with which to compute the ruin probabilities. Let $\Sigma = Nk + l$ (Euclidean division of Σ by N) and let $s = \lfloor k/2 \rfloor$. The steps of the algorithm are as follows.

(i) Recursively apply Proposition 3.3 until $s \leq 2$.

(ii) Apply (2.2) to compute the ruin probabilities associated with the last system.

(iii) Recursively apply (3.3) to compute the ruin vectors for each level of the game.

3.3.2. *Comparison.* It is well known that the inversion of a $p \times p$ matrix takes $O(p^3)$ operations and that the multiplication of a $p \times q$ matrix with a $q \times r$ matrix takes $O(pqr)$ operations. Therefore, direct inversion of the ruin problem using (2.2) takes $O(p^3)$ operations with

$$p^3 = \binom{\Sigma - 1}{N - 1}^3.$$

The next proposition gives a rough upper bound for the number of operations involved in the folding algorithm.

Proposition 3.4. *The number of operations involved in the folding algorithm is bounded above by*

$$\frac{\Sigma}{(\Sigma - 1)^3} N^5 p^3.$$

Remark 3.2. For fixed N and $\Sigma > N$, this simple upper bound already shows considerable savings in the number of operations, of $O(\Sigma^2)$.

Proof of Proposition 3.4. We use the same notation as in Proposition 3.1. After each folding, the new random walk runs on a set with $\lfloor k/2 \rfloor$ levels. Therefore, after at most $\lfloor \log_2 k \rfloor$ foldings, there will be strictly less than three levels left. We denote by $N(l)$ the number of operations involved in the lth folding. We will only take into account multiplications and inversions appearing in (3.4). For each $j \in \{1, \ldots, k\}$ there are 12 such operations. Counting the number

https://doi.org/10.1239/jap/1158784944 Published online by Cambridge University Press

of operations in the same way as above and using the fact that the b_j are decreasing in j, it is then straightforward to see that

$$N(j) \le 12\frac{k}{2^j}b_{2^{j-1}}^3.$$

This implies that the total number of operations involved in the folding of the process is of order

$$k\sum_{j=1}^{\lfloor \log_2 k \rfloor}\frac{1}{2^j}b_{2^{j-1}}^3 \le kb_1^3\sum_{j=1}^{\lfloor \log_2 k \rfloor}\frac{1}{2^j} \le kb_1^3.$$

We can show that

$$b_1 \le N\binom{\Sigma-2}{\Sigma-N} = N\frac{N-1}{\Sigma-1}p;$$

therefore, a rough upper bound for the number of operations involved in the folding of the process is given by

$$k\left(N\frac{N-1}{\Sigma-1}p\right)^3.$$

Application of (2.2) to obtain the ruin vectors on the folded set will not change the order of the number of operations. Also, the recursive application of (3.3) to obtain the ruin vectors of the whole system will take the same number of operations as will the folding and, therefore, does not change the order of the number of operations either. Since $k \le \Sigma/N$, we therefore see that the number of operations demanded by the folding algorithm admits the upper bound

$$\frac{\Sigma}{(\Sigma-1)^3}N^5p^3.$$

4. Conclusion

We recall that, for fair games and specifically for $N = 3$, asymptotic methods provide an interesting alternative to the approach presented here (see Ferguson (1995), Alabert *et al.* (2004), and Swan and Bruss (2004)). The applications of the present paper must be seen as being confined to 'real-world' problems. If the number of players, N, is not too large, then the saving in operations required to find the solution is, as we have seen, considerable. If N were to become large, the relative worth of this approach would become negligible and an asymptotic approach would gain independent interest.

Acknowledgement

The authors are pleased to thank the referee for his/her interest in the problem and very helpful comments.

References

Alabert, A., Farré, M. and Roy, R. (2004). Exit times from equilateral triangles. *Appl. Math. Optimization* **49,** 43–53.

Bruss, F. T., Louchard, G. and Turner, J. W. (2003). On the N-tower-problem and related problems. *Adv. Appl. Prob.* **35,** 278–294.

Engel, A. (1993). The computer solves the three tower problem. *Amer. Math. Monthly* **100,** 62–64.

Ferguson, T. S. (1995). *Gambler's Ruin in Three Dimensions*. Unpublished manuscript. Available at http://www.math.ucla.edu/~tom/.

Householder, A. S. (1964). *The Theory of Matrices in Numerical Analysis*. Blaisdell, New York.

Latouche, G. (1989). Distribution de type phase: tutorial. *Cahiers Centre Études Rech. Opér.* **31,** 3–11.

LATOUCHE, G. AND RAMASWAMI, V. (1999). *Introduction to Matrix Analytic Methods in Stochastic Modeling*. Society for Industrial and Applied Mathematics, Philadelphia, PA.

NEUTS, M. F. (1975). Probability distributions of phase type. In *Liber Amicorum Professor Emeritus H. Florin*, Department of Mathematics, University of Louvain, pp. 173–206.

NEUTS, M. F. (1978). Renewal processes of phase type. *Naval. Res. Logistics* **25,** 445–454.

NEUTS, M. F. (1981). *Matrix-Geometric Solutions in Stochastic Models. An Algorithmic Approach*. The Johns Hopkins University Press, Baltimore, MD.

STIRZAKER, D. (1994). Tower problems and martingales. *Math. Scientist* **19,** 52–59.

SWAN, Y. AND BRUSS, F. T. (2004). The Schwarz–Christoffel transformation as a tool in applied probability. *Math. Scientist* **29,** 21–32.

YE, J. AND LI, S. Q. (1994). Folding algorithm: a computational method for finite QBD processes with level dependent transitions. *IEEE Trans. Commun.* **45,** 625–639.