A new solution algorithm for skip-free processes to the left

Claus Bauer

Dolby Laboratories,
San Francisco, 94103, USA
email: `cb@dolby.com`

ABSTRACT

This paper proposes a new solution algorithm for steady state models describing skip-free processes to the left where each level has one phase. The computational complexity of the algorithm is independent of the number of levels of the system. If the *skip parameter* of the skip-free process is significantly smaller than the number of levels of the system, our algorithm numerically outperforms existing algorithms for skip-free processes. The proposed algorithm is based on a novel method for applying generalized Fibonacci series to the solution of steady state models.

RESUMEN

Este artículo propone un nuevo algoritmo solución para modelos estado-steady describiendo procesos libres-salto para la izquierda donde todo nivel tiene una fase. La complejidad computacional del algoritmo es independiente del número de niveles del sistema. Si el *parámetro de salto* de los procesos libre-salto es significativamente pequeño respecto del número de niveles del sistema, nuestro algoritmo numérico supera algoritmos existentes para procesos libre-salto. El algoritmo propuesto se basa en un método reciente para aplicar series de Fibonacci generalizados para la solución de modelos-steady.

Key words and phrases: *Skip-free processes, Markovian environment, stationary distribution*

AMS 2000 Subj. Class.: *60J10, 60J99*

1   Introduction

Skip-free processes have been widely researched as a a way to study packet based communication systems. In particular, *skip-free processes to the left* have been applied to model the dynamics of *finite* buffers of switches in data networks that experience the arrival of several data packets at a time while they are only being able to forward one packet per time slot. Using a discrete time model and common queueing theoretic terminology, a skip-free process to the left defines a state model where the (queue) occupancy can move down by one level in a time slot, but might move up by several levels in a time slot. The dynamics of these finite queues are commonly modeled as $M/G/1/K$ queues where K denotes the size of the queue. Similarly, buffers that experience only one packet arrival at a time, but can forward more than one packet per time slot can be modeled as *skip-free processes to the right.*

Algorithms to solve steady state models for skip-free processes were first investigated in [14], [15] by Neuts. In these papers, skip-free processes are considered as a subset of a more general class of steady state models and the application of matrix-geometric methods to solve this class of steady state models is investigated. A steady state analysis that takes explicitly into account the special structure of skip-free processes is proposed in [13, chap. 13]. Combining methods from [5] and [6], skip-free processes are modeled as a special case of Quasi-Birth-and-Death Processes (QBDs). Following the terminology introduced in [13], we define a QBD process as a skip-free process where the system can not move more than one level both downwards or upwards. Several other variations of skip-free processes have been researched in [1], [3], [4], [20], [22], [23].

In this paper, we investigate homogeneous finite skip-free processes to the left, i.e., we assume a finite number of levels and we assume that all levels have the same number of phases. In particular, we assume that each level has one phase. We first give a result of primarily theoretical interest by presenting a new way to derive a closed-form solution solution of a steady state model for skip-free processes to the left. In a second step, we show that the structure of this closed-form solution can be characterized by specific linear recurrent equations. We then apply the theory of general Fibonacci sequences [19] to these linear recurrent equations,. This allows us to derive the main contribution of this paper which is a new solution algorithm for skip-free processes to the left.

This solution algorithm has a complexity that is independent of the number of levels of the system. Previous solution algorithms for skip-free models have complexities that also depend on the number of levels of the system. Prior to our work, solution algorithms for steady state models that have a complexity independent of the number of levels of the system were only known for specific classes of QBDs [13, chap. 10.4], [7], [18]. These classes of QBDs do not contain the QBDs used to model skip-free processes in [13, chap. 13].

Finally, we perform numerical experiments to compare the numerical complexity of our algorithm with the numerical complexity of previously known algorithms. Our experiments show that the complexity of our method is lower than the complexity of previous algorithm if the *skip parameter* is small compared to the overall number of levels of the system. The *skip parameter* is defined as the maximum number of levels that the queue occupancy can increase in a time slot.

We note - without presenting any details in this paper - that our methods can also be applied to derive corresponding results for skip-free processes to the right.

In the next section, we provide the exact problem definition. In sec. 3 - 5, we prove and analyze our Theorem 3.1 which gives a closed-form solution for skip-free processes to the left. In sec. 6, we show that the obtained solution can be described by a set of linear recurrent equations. Based

on this observation, we apply the theory of generalized Fibonacci sequences to obtain a new solution algorithm for skip-free processes to the left. In sec. 7 and 8, we compare the computational complexity of the solution algorithm with previous techniques. We summarize our results in sec. 9.

2   Problem formulation

We consider a discrete time Markov chain X_t, $t \in \mathbb{N}$ on the two-dimensional state space $\{(n, i) : 0 \leq n \leq N, 1 \leq i \leq p\}$, which we partition as $\bigcup_{n \geq 1} l(n)$, where $l(n) = \{(n, 1), (n, 2), ..., (n, p)\}$ for $0 \leq n \leq N$. The first coordinate n is called the level, and the second coordinate j is called the phase of the state (n, j). In this paper, we only consider the case $p = 1$. We suppose that the discrete Matrix chain is described by a transition matrix $T = (T_{i,j})_{0 \leq i,j \leq N}$, where $T_{i,j}$ is the transition probability from the level i to level j. We assume T to be of the following form:

$$
T \;=\; \begin{matrix} & & j \\ & 0 \quad\quad\quad \to \quad\quad\quad N \\ i \downarrow & \begin{pmatrix}
E & B_2 & B_3 & .. & B_m & & & & & \\
B_0 & B_1 & B_2 & .. & B_{m-1} & B_m & & & & \\
 & B_0 & B_1 & .. & .. & B_{m-1} & B_m & & & \\
 & & B_0 & .. & .. & .. & .. & B_m & & \\
 & & & .. & .. & .. & .. & .. & .. & \\
 & & & & & B_0 & & & B_{m-1} & C_m \\
 & & & & & & B_0 & & B_{m-2} & C_{m-1} \\
 & & & & & & & .. & .. & .. & .. \\
 & & & & & & & .. & .. & .. & .. \\
 & & & & & & & B_0 & B_1 & C_2 \\
 & & & & & & & & B_0 & C_1
\end{pmatrix} \end{matrix}
$$

$$(2.1)$$

Here the expressions B_j, $0 \leq j \leq m$, $E = B_0 + B_1$, $C_i = \sum_{j=i}^{m} B_j$, $1 \leq i \leq m$ are real numbers in the open interval $(0, 1)$. We see that the structure of the matrix T depends essentially on the *skip parameter m* as the parameter defines which levels can be reached from a current level in a one-step transition. Any level i can be reached from all stages $i - j$, $-1 \leq j \leq m - 1$ if $m - 1 \leq i \leq N - 1$. By the definition of a transition matrix, we have

$$
\sum_{j=1}^{N} T_{i,j} \;=\; 1, \qquad \forall i, \; 0 \leq i \leq N,
$$

which implies that

$$\sum_{l=0}^{m} B_l = 1. \tag{2.2}$$

Defining the steady state vector as $\pi = (\pi_0, .., \pi_N)$, we can write the steady state equation $\pi = \pi T$ as

$$\pi_1 B_0 = \pi_0(1 - E), \tag{2.3}$$

$$\pi_k B_0 = \pi_{k-1}(1 - B_1) - \sum_{l=2}^{k} \pi_{k-l} B_l, \qquad 2 \le k \le m - 1, \tag{2.4}$$

$$\pi_k B_0 = \pi_{k-1}(1 - B_1) - \sum_{l=2}^{m} \pi_{k-l} B_l, \qquad m \le k \le N - 1, \tag{2.5}$$

$$\pi_N(C_1 - 1) = -\sum_{l=1}^{m-1} \pi_{N-l} C_{l+1}. \tag{2.6}$$

We now define new variables

$$C = B_0, \tag{2.7}$$

$$A_{m-l} = 1 - \sum_{n=0}^{l} B_n, \qquad 1 \le l \le m - 1. \tag{2.8}$$

Using eqn. (2.2), we see that $A_1 = B_m = C_m$. Using the definitions (2.7) and (2.8), we can rewrite the eqn. (2.3) - (2.5) as follows:

$$\pi_1 C = \pi_0^* A_{m-1}, \qquad \pi_0^* = \pi_0(1 - E)A_{m-1}^{-1} \tag{2.9}$$

$$\pi_k C = \pi_{k-1}(A_{m-1} + C) + \sum_{l=2}^{k} \pi_{k-l}(A_{m-l} - A_{m-(l-1)}), \qquad 2 \le k \le m - 1, \tag{2.10}$$

$$\pi_k C = \pi_{k-1}(A_{m-1} + C) + \sum_{l=2}^{m-1} \pi_{k-l}(A_{m-l} - A_{m-(l-1)}) - \pi_{k-m} A_1, \tag{2.11}$$
$$m \le k \le N - 1,$$

$$\pi_N(C_1 - 1) = -\sum_{l=1}^{m-1} \pi_{N-l} C_{l+1}. \tag{2.12}$$

3   Closed form solution of the steady state model

In this section, we give a closed form solution of the steady state model defined via the transition matrix T in (2.1) or equivalently in (2.9) - (2.12). We show the following Theorem:

Theorem 3.1.

$$\pi_0^* = \left(1 + \sum_{k=1}^{N-1} \sum_{b_l}^{k} E(b_{m-1}, .., b_1) + Y(C_0 - 1)^{-1}\right)^{-1}, \tag{3.1}$$

$$\pi_k = \pi_0^* \sum_{b_l}^{k} E(b_{m-1}, ..., b_1), \qquad 1 \le k \le N - 1, \tag{3.2}$$

$$\pi_N = \pi_0^* Y(C_1 - 1)^{-1}, \tag{3.3}$$

where

$$E(b_{m-1}, ..., b_1) = C^{-d} \begin{pmatrix} d \\ c_{m-1} \ c_{m-2} \ .... \ c_1 \end{pmatrix} \prod_{l=1}^{m-1} A_l^{c_l}, \tag{3.4}$$

$$c_l = b_l - 2b_{l-1} + b_{l-2}, \ 3 \le l \le m-1, \tag{3.5}$$

$$c_2 = b_2 - 2b_1, \tag{3.6}$$

$$c_1 = b_1, \tag{3.7}$$

$$d = b_{m-1} - b_{m-2}. \tag{3.8}$$

The summation $\sum_{b_l}^k$ runs over all $b_l \ge 0$, $1 \le l \le m-1$ such that $c_l \ge 0$ for $1 \le l \le m-1$, In this summation, the variable b_{m-1} only takes the fixed value $b_{m-1} = k$. Further,

$$Y = -\sum_{l=1}^{m-1} C_{l+1} \sum_{b_l}^{N-l} E(b_{m-1}, ..., b_1). \tag{3.9}$$

Remarks: We state some remarks for later use in this paper:

1. By definition,

$$\sum_{l=1}^{m-1} c_l = d. \tag{3.10}$$

2. For any fixed b_{l+1} over which is summed in the summation $\sum_{b_l}^k$, the number of b_l over which is summed in $\sum_{b_l}^k$ is limited by

$$b_l \le \frac{l}{l+1} b_{l+1}. \tag{3.11}$$

This follows from the summation condition $c_2 \ge 0$ and the eqn. (3.6) for $l = 1$. For $l \ge 2$, we apply the induction principle. Assuming that eqn. (3.11) holds for l, then by the summation condition $c_{l+2} \ge 0$

$$2b_{l+1} \le b_l + b_{l+2} \le \frac{l}{l+1} b_{l+1} + b_{l+2},$$

which implies eqn. (3.11) for $l + 1$.

3. The eqn. (3.11) implies that

$$d \ge 1. \tag{3.12}$$

4 Proof of Theorem 3.1

We first recall a well-known identity for multinomial coefficients. For any set of strictly positive integers $a_1, .., a_k$ with $n = \sum_{j=1}^{k} a_j$, there is

$$\begin{pmatrix} n \\ a_1 \ a_2 \ ... \ a_k \end{pmatrix} = \sum_{j=1}^{k} \begin{pmatrix} n-1 \\ a_1 \ .. \ (a_j - 1) \ ... \ a_k \end{pmatrix}. \tag{4.1}$$

For any integer x, $1 \leq x \leq m - 1$ and a given set of variables $b_1, ..., b_{m-1}$ as defined in (3.4), we introduce the additional variables b_l^x and c_l^x defined as follows:

$$b_l^x = b_l - \max(0, x - (m - 1 - l)), \ l \geq 1.. \tag{4.2}$$

$$c_l^x = b_l^x - 2b_{l-1}^x + b_{l-2}^x, \ 3 \leq l \leq m - 1, \tag{4.3}$$

$$c_2^x = b_2^x - 2b_1^x, \tag{4.4}$$

$$c_1^x = b_1^x, \tag{4.5}$$

$$d^x = b_{m-1}^x - b_{m-2}^x. \tag{4.6}$$

We now state two lemmas which we will need for the proof of Theorem 3.1.

Lemma 4.1. *For* $1 \leq x \leq m - 1$,

$$c_l^x = c_l - \left\{ \begin{array}{ll} 1 & \text{if } l = m - x, \\ 0 & \text{else} . \end{array} \right\} \tag{4.7}$$

$$d^x = d - 1. \tag{4.8}$$

Proof of Lemma 4.1: We prove the eqn. (4.7) by considering different ranges of the variables l, m and x :

     Range 1: For $l \geq m - x + 1$, $l \geq 3$,
$c_l^x = c_l - x + m - 1 - l + 2x - 2m + 2 + 2l - 2 - x + m - 1 - l + 2 = c_l$.
     Range 2: For $l = m - x$, $l \geq 2$,
$c_l^x = c_l - x + m - 1 - l + 2x - 2m + 2 + 2l - 2 = c_l + x - m + l - 1 = c_l - 1$.
     Range 3: For $l = m - x = 1$
$c_l^x = c_l - 1$.
     Range 4: For $l \leq m - x - 1$, $l \geq 1$,
$c_l^x = c_l$.

We now prove the eqn. (4.8):

$$d^x = b_{m-1} - x - (b_{m-2} - max(0, (x - 1)) = d - 1.$$

For later usage, we note that the eqn. (3.10), (4.7), and (4.8) imply that for $1 \leq x \leq m - 1$, there is

$$\sum_{l=1}^{m-1} c_l^x = d^x. \tag{4.9}$$

Lemma 4.2.

$$E(b_{m-1}, .., b_1) = \sum_{x=1, \, c_{m-x} \neq 0}^{m-1} E^x(b_{m-1}^x, ..., b_1^x) A_{m-x} C^{-1}.$$

Proof of Lemma 4.2: We first note that in the definition (3.4), if for any x there is $c_{m-x} = 0$, then

$$C^{-d} \begin{pmatrix} d \\ c_{m-1} \ \ c_{m-2} \ \ c_{m-x+1} \ \ c_{m-x} \ \ c_{m-x-1}.... \ \ c_1 \end{pmatrix} \prod_{l=1}^{m-1} A_l^{c_l}$$

$$= \ C^{-d} \begin{pmatrix} d \\ c_{m-1} \ \ c_{m-2} \ \ c_{m-x+1} \ \ c_{m-x-1}.... \ \ c_1 \end{pmatrix} \prod_{l=1, \, l \neq m-x}^{m-1} A_l^{c_l},$$

i.e., we can neglect the contribution of the quantity c_{m-x}. Now, Lemma 4.2 follows from the definition (3.4) and the eqn. (3.12), (4.1), (4.7), and (4.8).

Proof of Theorem 3.1:

First, we prove eqn. (3.2) by induction over k. For $k = 1$, we see $b_{m-1} = c_{m-1} = d = 1$, whereas $b_l = c_l = 0$ for $l \leq m - 2$. Thus, we obtain from eqn. (3.2) that $\pi_1 = \pi_0^* A_{m-1} C^{-1}$ as required by eqn. (2.9). For the inductive step, we now assume that eqn. (3.2) holds for π_i, $1 \leq i \leq k - 1$, and prove it for $i = k < N$. In view of the eqn. (2.10) and (2.11), we see that in order to show that the eqn. (3.2) holds for π_k, it is sufficient to prove the following two equations:

$$\pi_k \ = \ \sum_{x=1}^{\min(m-1,k)} \pi_{k-x} A_{m-x} C^{-1}, \tag{4.10}$$

$$\pi_{k-1} \ = \ \sum_{x=1}^{\min(m-1,k-1)} \pi_{k-1-x} A_{m-x} C^{-1}. \tag{4.11}$$

For the proof of the eqn. (4.10), we first consider the case $m - 1 < k$. Replacing the probabilities π_l, $k - m + 1 \leq l \leq k$, in eqn. (4.10) with the right hand side of eqn. (3.2), we obtain

$$\sum_{b_l}^{k} E(b_{m-1}, ..., b_1) \ = \ \sum_{x=1}^{m-1} A_{m-x} C^{-1} \sum_{\tilde{b}_l}^{k-x} E(\tilde{b}_{m-1}, ..., \tilde{b}_1). \tag{4.12}$$

Applying Lemma 4.2 to the left hand side of eqn. (4.12), we see that in order to prove eqn. (4.10) we have to show the following identity:

$$\sum_{x=1, \, c_{m-x} \neq 0}^{m-1} A_{m-x} C^{-1} \sum_{b_l}^{k} E(b_{m-1}^x, ..., b_1^x) \ = \ \sum_{x=1}^{m-1} A_{m-x} C^{-1} \sum_{\tilde{b}_l}^{k-x} E(\tilde{b}_{m-1}, ..., \tilde{b}_1). \tag{4.13}$$

For the proof of eqn. (4.13), it is sufficient to prove the following two claims:

1. For each integer x, $1 \leq x \leq m - 1$, and for each set $b_{m-1}^x, ..., b_1^x$ derived via the relation (4.2) from a set $b_{m-1}, ..., b_1$ over which is summed in the sum $\sum_{b_l}^{k}$, and for which $c_{m-x} \neq 0$, there exists a set $\tilde{b}_{m-1}, ..., \tilde{b}_1$ over which is summed in the sum $\sum_{\tilde{b}_l}^{k-x}$ for which holds $c_l^x = \tilde{c}_l$ for all l, $1 \leq l \leq m - 1$, and $d^x = \tilde{d}$.

2. For each integer x, $1 \leq x \leq m - 1$, and for each set $\tilde{b}_{m-1}, ..., \tilde{b}_1$ over which is summed in the sum $\sum_{\tilde{b}_l}^{k-x}$, there exists a set $b_{m-1}^x, ..., b_1^x$ derived via the relation (4.2) from a set $b_{m-1}, ..., b_1$

over which is summed in the sum $\sum_{b_l}^{k}$, for which $c_{m-x} \neq 0$, and $\tilde{c}_l = c_l^x$ for all l, $1 \leq l \leq m-1$, and $\tilde{d} = d^x$.

In order to show claim 2, we set $e_l^{\tilde{b}} = \tilde{b}_l + \max(0, x - (m-1-l))$. Further, we define $e_l^{\tilde{c}}$ and $e_l^{\tilde{d}}$ via $e_l^{\tilde{b}}$ in the same way c_l and d are defined via b_l as in (4.3) - (4.6). By definition, we see that $(e_l^{\tilde{b}})^x$ - defined as in (4.2) - equals $\tilde{b}_l$ which implies that $\tilde{c}_l = (e_l^{\tilde{c}})^x$ and $\tilde{d}_l = (e_l^{\tilde{d}})^x$. Now, in order to prove claim 2, it remains to show that the set $e_{m-1}^{\tilde{b}}, ..., e_1^{\tilde{b}}$ belongs to the summation $\sum_{b_l}^{k}$, i.e., $e_{m-1}^{\tilde{b}} = k$, $e_{m-l}^{\tilde{c}} \geq 0$ if $l \neq x$ and $e_{m-x}^{\tilde{c}} > 0$. The first claim is obvious by the definition of $e_{m-1}^{\tilde{b}} = \tilde{b}_{m-1} + x = k - x + x = k$. We note that reversing the proof of the relation (4.7), we can show that

$$e_l^{\tilde{c}} = \tilde{c}_l + \left\{ \begin{array}{ll} 1 & l = m - x \\ 0 & else. \end{array} \right\} \tag{4.14}$$

As $\tilde{c}_l \geq 0$, the eqn. (4.14) implies that $e_l^{\tilde{c}} \geq 0$, $1 \leq l \leq m-1$, $l \neq m - x$, and $e_{m-x}^{\tilde{c}} > 0$, q.e.d.

In order to show claim 1, we have to show that the set $b_{m-1}^x, ..., b_1^x$ belongs to the summation $\sum_{b_l}^{k-x}$. For this purpose, we have to show that $b_{m-1}^x = k - x$ and $c_l^x \geq 0$, $1 \leq x \leq m-1$. The first inequality follows from eqn. (4.2). The second relation follows from eqn. (4.7) and the fact that on the left-hand side of eqn. (4.13) we only sum over $c_{m-x} \neq 0$.

In the case $m - 1 \geq k$, we argue as above and have to show that

$$\sum_{x=1, c_{m-x} \neq 0}^{k} A_{m-x} C^{-1} \sum_{b_l}^{k} E(b_{m-1}^x, ..., b_1^x) = \sum_{x=1}^{k} A_{m-x} C^{-1} \sum_{\tilde{b}_l}^{k-x} E(\tilde{b}_{m-1}, ..., \tilde{b}_1). \tag{4.15}$$

Eqn. (4.15) is shown in the same way as eqn. (4.13). Eqn. (4.11) is shown in the same way as eqn. (4.10). Eqn. (3.3) follows from (2.12) and (3.2). Eqn. (3.1) follows from (3.2), (3.3), and the eqn. $\sum_{i=0}^{N} \pi_i = 1$.

5 Complexity of the calculation of π_k using Theorem 3.1

The inequality (3.11) gives an upper bound for the number of b_l over which is summed in the summation $\sum_{b_l}^{k}$. Using $b_{m-1} = k$, eqn. (3.11), and the well-known relation,

$$\sum_{k=1}^{N} k^m = \frac{1}{m+1} N^{m+1} + O(N^m),$$

we obtain

$$\sum_{b_{m-2}=1}^{\lfloor \frac{(m-2)k}{m-1} \rfloor} \sum_{b_{m-3}=1}^{\lfloor \frac{(m-3)b_{m-2}}{m-2} \rfloor} \cdots \sum_{b_1=1}^{\lfloor \frac{b_2}{2} \rfloor} 1 = \frac{1}{(m-1)!} \prod_{l=2}^{m-1} \left(\frac{l-1}{l} \right)^l k^{m-1} + O(k^{m-2}).$$

Thus, the complexity of the calculation of the steady state probability of π_0^* using the relation (3.1) is $\sim \frac{1}{(m-1)!} \prod_{l=2}^{m-1} \left(\frac{l-1}{l}\right)^l \sum_{k=1}^{N-1} k^{m-1} \sim \frac{1}{m!} \prod_{l=2}^{m-1} \left(\frac{l-1}{l}\right)^l N^m$. Here, we assume that the values of the multinomial coefficients have been pre-computed as they do not depend on the actual values of the transition matrix T. Once π_0^* is calculated, the calculation of π_k for any $2 \leq k \leq N-1$ requires $O(1)$ steps if we assume that the values of the sums $\sum_{b_l}^{k}$ already determined for the calculation of π_0^* have been stored. The calculation of π_N requires $O(m)$ steps.

In summary, we see that the overall computational complexity of Theorem 3.1 is $\sim \frac{1}{m!} \prod_{l=2}^{m-1} \left(\frac{l-1}{l}\right)^l N^m$. For $m > 3$ and large values of N, this complexity is too high for any practical applications. In the next section, we will show how the complexity of Theorem 3.1 can be significantly reduced.

6 A reduction of the complexity of Theorem 3.1

In this section, we apply the theory of linear recurrence equations to Theorem 3.1. This will allows to find a computationally very efficient way of computing the sum $\sum_{b_l}^{k} E(b_{m-1}, .., b_1)$.

First, we recall some facts from the theory of linear recurrent equations:

Lemma 6.1. *We consider a sequence of real numbers x_n, $n \geq 1$ defined via a set of real numbers $a_1, .., a_{n-1}$ and a set of strictly positive real numbers $b_1, .., b_{n-1}$ as follows:*

$$x_i = a_i, \qquad 1 \leq i \leq n-1, \tag{6.1}$$

$$x_i = \sum_{k=1}^{n-1} b_k x_{i-k}, \qquad i \geq n. \tag{6.2}$$

Then, we have:

$$x_n = \sum_{u=1}^{w} \sum_{v=0}^{f_u-1} c_{u,v} n^v r_u^n. \tag{6.3}$$

The numbers r_u, f_u, and w are defined via the characteristic polynomial

$$y^{n-1} + \sum_{k=1}^{n-1} b_k y^{n-1-k}. \tag{6.4}$$

r_u are the roots, f_u are the multiplicities of the roots r_u, and w is the number of different roots of the characteristic polynomial (6.4). Obviously, $\sum_{u=1}^{w} f_u = n-1$. The numbers $c_{u,v}$ are the solutions of the $(n-1)-$ equations

$$a_i = \sum_{u=1}^{w} \sum_{v=0}^{f_u-1} c_{u,v} i^v r_u^i, \qquad 1 \leq i \leq n-1. \tag{6.5}$$

Proof: The proof can be found in [10].

We now apply Lemma 6.1 to the eqn. (4.12). Setting $x_k = \sum_{b_l}^{k} E(b_{m-1}, .., b_1)$ in (6.1) and (6.2), we obtain from Lemma 6.1:

Lemma 6.2.

$$\sum_{b_l}^{k} E(b_{m-1}, .., b_1) \quad = \quad \sum_{u=1}^{w} \sum_{a=0}^{g_u-1} t_{u,a} k^a h_u^k,$$

Here, the numbers g_u denote the respective multiplicities of the w different roots h_u of the characteristic polynomial

$$x^{m-1} - \sum_{u=1}^{m-1} A_{m-u} C^{-1} x^{m-1-u}. \tag{6.6}$$

The numbers $t_{u,h}$ are the solutions of the $(m-1)$- linear equations

$$\sum_{b_l}^{k} E(b_{m-1}, .., b_1) \quad = \quad \sum_{u}^{m-1} \sum_{h=0}^{g_u-1} t_{u,h} v^h h_u^v, \qquad 1 \le v \le m-1.$$

We now use Lemma 6.2 to simplify Theorem 3.1 as follows:

Theorem 6.1.

$$\pi_0^* \quad = \quad \left(1 + \sum_{k=1}^{N-1} F_k + Y^*(1 - C_0) \right)^{-1}, \tag{6.7}$$

$$\pi_k \quad = \quad \pi_0^* F_k, \qquad 0 \le k \le N - 1, \tag{6.8}$$

$$\pi_N \quad = \quad \pi_0^* Y^*(C_1 - 1)^{-1}, \tag{6.9}$$

where

$$F_k \quad = \quad \sum_{u=1}^{w} \sum_{a=0}^{g_u-1} t_{u,a} k^a h_u^k. \tag{6.10}$$

The numbers w, h_u, g_u, $t_{u,h}$ are defined as in Lemma 6.2, and

$$Y^* \quad = \quad - \sum_{l=1}^{m-1} C_{l+1} F_{N-l}.$$

Proof: The eqn. (6.7) - (6.10) follow directly from Theorem 3.1 and Lemma 6.2.

7 The complexity of Theorem 6.1

7.1 Preliminaries

In this section, we investigate the computational complexity of Theorem 6.1. We see from the formulation of Theorem 6.1 that the complexity of the calculation of a steady state probability π_k mainly derives from the calculation of the sum $\sum_{k=1}^{N-1} F_k$, the solution of the $(m-1)$ eqn. (6.5), and the calculation of the roots of the characteristic polynomial (6.6). It is well-known that the solution of the system of equations (6.5) using Gaussian elimination requires $O(m^3)$ operations. We consider the complexity of the other two operations separately in the next two sections.

7.2   The calculation of the sum $\sum\limits_{k=1}^{N-1} F_k$

We see from eqn. (6.10) that the sum $\sum\limits_{k=1}^{N-1} F_k$ can be rewritten as follows:

$$
\begin{aligned}
\sum_{k=1}^{N-1} F_k &= \sum_{u=1}^{w} \sum_{a=0}^{g_u-1} t_{u,a} \sum_{k=1}^{N-1} k^a h_u^k \\
&:= \sum_{u=1}^{w} \sum_{a=0}^{g_u-1} t_{u,a} H_{N-1,a,h_u},
\end{aligned}
\tag{7.1}
$$

where

$$
H_{N,q,x} := \sum_{k=1}^{N} k^q x^k.
$$

For the calculation of $H_{N,q,x}$ we prove the following lemma:

Lemma 7.1.

$$
H_{N,q,1} = \sum_{k=1}^{q+1} \frac{(-1)^{q-k+1} BE_{q-k+1}}{p+1} \binom{p+1}{k} N^k,
\tag{7.2}
$$

$$
H_{N,0,x} = x\frac{x^N-1}{x-1},
\tag{7.3}
$$

$$
H_{N,q,x} = x\frac{(N+1)^q x^N + (N+1)^q - 2}{x-1} - \frac{x}{x-1}\sum_{b=0}^{q-1} H_{N,b,x}\binom{q}{b}, \quad x \neq 1, q > 1.
\tag{7.4}
$$

In eqn. (7.2), BE_k denotes the $k-th$ Bernoulli number [2].

   Proof: The relation (7.2) is explained in [2] and the relation (7.3) is a known formula for geometric sums. For the proof of eqn. (7.4), we will use the technique of partial summation defined in [17] as follows: For any complex series a_k, b_k, $M < n \leq N$, there is

$$
\sum_{M<n\leq N} a_k b_k = A_N b_{N+1} + \sum_{m<k\leq N} A_k(b_k - b_{k+1}),
\tag{7.5}
$$

where

$$
A_N = \sum_{M<k\leq N} a_k.
$$

Applying (7.5) with $a_k = x^k$ and $b_k = k^q$, and the formula for geometric sums as applied in (7.3), we see

$$
\begin{aligned}
H_{N,q,x} &= (N+1)^q x \frac{x^N - 1}{x-1} + \sum_{k=1}^{N} x \frac{x^k - 1}{x-1} \left(k^q - (k+1)^q\right) \\
&= (N+1)^q x \frac{x^N - 1}{x-1} + \frac{x}{x-1} \sum_{k=1}^{N} x^k \left(k^q - (k+1)^q\right) - \frac{x}{x-1} \sum_{k=1}^{N} \left(k^q - (k+1)^q\right) \\
&= x \frac{(N+1)^q(x^N - 1)}{x-1} - \frac{x}{x-1} \sum_{k=1}^{N} x^k \sum_{b=0}^{q-1} k^b \binom{q}{b} - x \frac{1 - (N+1)^q}{x-1} \\
&= x \frac{(N+1)^q x^N + (N+1)^q - 2}{x-1} - \frac{x}{x-1} \sum_{b=0}^{q-1} H_{N,b,x} \binom{q}{b}. \qquad \square
\end{aligned}
$$

Lemma 7.1 shows that for $x \neq 1$, the sum $H_{N,q,x}$ can be calculated recursively from the sums $H_{N,b,x}$, $0 \leq b \leq q - 1$, with a complexity of $O(q)$. Thus, the calculation of all sums $H_{N,b,x}$, $0 \leq b \leq q$ has a complexity of $O(q^2)$. In consequence, the complexity of the calculation of the right- hand side of eqn. (7.1) is

$$
O \left(\sum_{u=1}^{w} g_u^2 \right) = O(m^2),
$$

because of $\sum_{u=1}^{w} g_u = m - 1$. In particular, we note that whereas the skip parameter m influences the complexity of the calculations, the variable N in the sum $\sum_{k=1}^{N_1}$ does not influence the complexity. The constant N appears as an exponent in the expressions (7.2) and (7.3). However, the exponential function is usually implemented using a Taylor expansion approximation, the complexity of which depends on the chosen precision of the approximation, but not on the specific value to be calculated.

7.3   The calculation of the roots of the characteristic polynomial

In order to apply Theorem 6.1, one needs to find the roots of the polynomial (6.6). We recall some know results from the theory of polynomial equations for polynomials with real coefficients. The general solution formula for quadratic equations is well known. Also, solution formulas for real polynomials of third and forth degree were developed by Cardano and Ferrari, respectively [12, chap. 2]. Thus, we see that for a degree ≤ 4, the calculation of the roots of the polynomial (6.6) can be done in constant time.

A well-known theorem from algebraic field theory due to Abel from characterizes the set of polynomials over a general field K for which the roots can be found explicitly. We quote Abel's theorem [9], p. 308:

Lemma 7.2. *Let K be a field, p a positive number, and $f_p(x)$ a real polynomial of degree p in the real variable x. The equation $f_p(x) = 0$ is solvable by radicals for all polynomials $f_p(x)$ of degree p if $p \leq 4$.*

In other words, Abel's theorem states that for polynomials of degree $m > 4$ no general exact solutions exists. Thus, the solution of polynomials of higher degree requires the application of approximative methods. An overview of these methods is given in [8]. Among these methods, Laguerre's method has been widely researched [16]. Whereas its convergent behavior is well understood, the speed of its convergence is not well described in the literature.

Therefore, we performed numerical experiments in order to understand the computational complexity of the calculation of the roots of the characteristic polynomial (6.6). For each degree m, we evaluated 10000 polynomials of degree m, and for each polynomial we used a random generator to generate the coefficients of the polynomial. For each polynomial, we measured the execution time $s(m)$ - measured in milliseconds - needed to determine all roots of the polynomial on a Pentium IV, 2.4 Ghz PC. The graph in fig. 1 shows the logarithm - to the base 10 - of $s(m)$ as a function of the logarithm - to the base 10 - of the degree m of the polynomial. An least mean square analysis of the simulation data shows that $\log(s(m))$ can be described as a linear function of $\log m$ as follows:

$$\log(s(m)) = 1.99 \log m + c$$

for some constant c. Thus, the execution time $s(m)$ can be approximately described as a second order polynomial of the polynomial degree m.
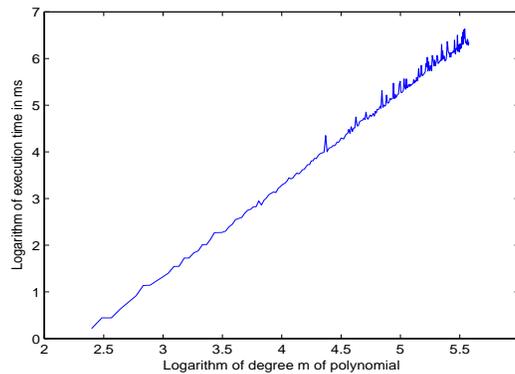

Figure 1: Speed of convergence of the calculation of the roots of the characteristic polynomial

7.4 Combined complexity of Theorem 6.1

The combinations of the results of sec. (7.1) - (7.3) shows that the overall computational complexity required to calculate a specific steady state probability π_k using Theorem 6.1 is $O(m^3)$.

8 Comparison of Theorem 6.1 with previous work

In this section, we compare the computational complexity of Theorem 6.1 with known methods to solve steady state models as defined via the transition matrix (2.1).

First, we present a well-known recursive way of solving the system described by the matrix (2.1).

Then, we follow an idea from [6] and model the Markov process defined via (2.1) as a QBD process
with $u := \left\lceil \frac{N+1}{m-1} \right\rceil$ levels. We discuss two known methods to solve QBD models due to Gaver et. al [5],
[13, chap. 13] and Ye and Li [21], respectively. Finally, we compare the computational complexity of
these approaches with the complexity of Theorem 6.1.

8.1   Recursive calculation

The equations (2.3) - (2.6) allow a recursive calculation of the steady state probabilities $\pi_k, 0 \leq k \leq N$.
We see from these equations that each π_k can be expressed as a linear function of all π_l, $0 \leq l \leq k-1$.
Thus, using a recursive argument each π_l can be expressed as the product of π_0 and a real number
k_l. As $\sum_{l=0}^{N} k_l = \pi_0^{-1}$, we can can calculate all steady state probabilities π_k. These calculations require
$O(N^2)$ calculations.

   We note that a similar approach can be used to derive a closed - form solution for π_k. In particular,
we consider the unconstrained random walk corresponding to (2.1) without a reflecting barrier on the
right, i.e., with an infinite number of levels. In order to define this random walk formally, we generalize
the transition probabilities B_n defined in (2.1) to

$$D_n = \left\{ \begin{array}{ll} B_n, & if\ 0 \leq n \leq m, \\ 0, & n \in \mathbb{Z}\backslash[0,m]. \end{array} \right\}. \tag{8.6}$$

We define the transition probabilities from state i to state j of the unconstrained random walk as

$$p_{ij} \quad = \quad \left\{ \begin{array}{ll} D_0 + D_1, & if\ i = j = 0, \\ D_{j-i+1}, & else. \end{array} \right\}.$$

We denote by $\{m_j : j \geq 0\}$ the invariant measure of this Markov chain and set $m_0 = 1$. Then,
$m_j = \sum_{i \geq 0} p_{ij} m_i$. Using the definition (8.6), we see

$$m_1 = \frac{1 - D_0 - D_1}{D_0}, \qquad D_0 m_{j+1} = m_j - \sum_{i=0}^{j} m_i D_{j-i+1} \qquad (j \geq 1). \tag{8.7}$$

The first N columns of the transition matrix defined by (8.6) are identical to the first N columns
of the transition matrix T defined in (2.1). Thus, there exists a constant K such that $\pi_j = K m_j$ if
$j < N$. There is

$$\pi_N = K \sum_{i=1}^{m-1} C_{m-i+1} m_{N-m+i} + \pi_N C_1,$$

and

$$\pi_N C_1 = \pi_N (1 - D_0) = \pi_N - D_0 + D_0 K \sum_{j=0}^{N-1} m_j, \tag{8.8}$$

which implies

$$K \quad = \quad \frac{D_0}{D_0 \sum_{i=0}^{N-1} m_i + \sum_{i=1}^{m-1} C_{m-i+1} m_{N-m+i}}.$$

Let $D(s) := \sum_{k \geq 0} s^k D_k$ be the generating function of the series D_k. Using standard manipulations with generating functions yields

$$M(s) \quad := \quad \sum_{k \geq 0} m_k s^k = D_0 \frac{1 - s}{B(s) - s}, \tag{8.9}$$

valid in the disc $|s| < q$, where q is the least positive solution of $D(s) = s$. As $D_n = 0$ for $n > m$, $B(s)$ is a polynomial and therefore $M(s)$ is meromorphic function with a finite number of poles. Using a partial fraction expansion of $M(s)$ and picking the coefficients of s^k on the right hand side of (8.9) gives a representation of the m_k. The calculation of the coefficients uses the recursive relations (8.7) and relation (8.8). Thus, it has a complexity of $O(N^2)$.

8.2 Modeling of the Markov process as a QBD process with $u := \left\lceil \frac{N+1}{m-1} \right\rceil$ levels

First, we describe how the matrix T can be redefined as a block matrix that describes a QBD process. For the sake of simplicity, we first assume that $N + 1 = u(m - 1)$ for some integer u. Then, we can rewrite T as defined in (2.1) as a $u \times u$ block matrix as follows:

$$T = \begin{pmatrix}
P_{1,1} & A_0 & 0 & 0 & .. & .. & .. & & .. & 0 \\
A_2 & A_1 & A_0 & .. & .. & .. & .. & & .. & 0 \\
0 & A_2 & A_1 & .. & .. & .. & .. & & .. & .. \\
0 & .. & .. & .. & .. & .. & .. & & .. & .. \\
0 & .. & .. & .. & .. & .. & .. & & .. & .. \\
.. & .. & .. & .. & .. & .. & .. & & .. & .. \\
.. & .. & .. & .. & .. & .. & A_1 & A_0 & 0 \\
.. & .. & .. & .. & .. & .. & A_2 & A_1 & A_0 \\
0 & 0 & .. & .. & .. & .. & .. & A_2 & P_{u,u}
\end{pmatrix}, \tag{8.10}$$

$$A_0 = \begin{pmatrix}
B_{m+1} & 0 & 0 & & .. & .. & .. & .. & & .. & & .. \\
B_m & B_{m+1} & 0 & & .. & .. & .. & .. & & .. & & .. \\
B_{m-1} & B_m & B_{m+1} & & .. & .. & .. & .. & & .. & & .. \\
.. & .. & .. & & .. & .. & .. & .. & & .. & & .. \\
.. & .. & .. & & .. & .. & .. & .. & & .. & & .. \\
.. & .. & .. & & .. & .. & .. & .. & & .. & & .. \\
B_5 & .. & .. & & .. & .. & .. & B_{m+1} & & 0 & & 0 \\
B_4 & B_5 & .. & & .. & .. & .. & B_m & & B_{m+1} & & 0 \\
B_3 & B_4 & .. & & .. & .. & .. & .. & & B_m & & B_{m+1}
\end{pmatrix},$$

$$
A_1 \;=\; \begin{pmatrix}
B_2 & B_3 & B_4 & .. & .. & .. & .. & .. & B_m \\
B_1 & B_2 & B_3 & .. & .. & .. & .. & .. & B_{m-1} \\
0 & B_1 & B_2 & .. & .. & .. & .. & .. & B_{m-2} \\
0 & 0 & B_1 & B_2 & .. & .. & .. & .. & B_{m-3} \\
.. & .. & .. & B_1 & B_2 & .. & .. & .. & .. \\
.. & .. & .. & .. & B_1 & B_2 & .. & .. & .. \\
.. & .. & .. & .. & .. & B_1 & B_2 & .. & .. \\
.. & .. & .. & .. & .. & 0 & B_1 & B_2 & B_3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & B_1 & B_2
\end{pmatrix}, \tag{8.11}
$$

$$
A_2 \;=\; \begin{pmatrix}
.. & .. & .. & .. & .. & .. & .. & 0 & B_1 \\
.. & .. & .. & .. & .. & .. & .. & 0 & 0 \\
.. & .. & .. & .. & .. & .. & .. & .. & 0 \\
.. & .. & .. & .. & .. & .. & .. & .. & .. \\
.. & .. & .. & .. & .. & .. & .. & .. & .. \\
.. & .. & .. & .. & .. & .. & .. & .. & .. \\
.. & .. & .. & .. & .. & .. & .. & .. & .. \\
.. & .. & .. & .. & .. & .. & .. & .. & .. \\
.. & .. & .. & .. & .. & .. & .. & .. & ..
\end{pmatrix},
$$

$$
P_{1,1} \;=\; \begin{pmatrix}
E & B_3 & B_4 & .. & .. & .. & .. & .. & B_m \\
B_1 & B_2 & .. & .. & .. & .. & .. & .. & B_{m-1} \\
0 & B_1 & B_2 & .. & .. & .. & .. & .. & B_{m-2} \\
0 & 0 & B_1 & .. & .. & .. & .. & .. & .. \\
.. & .. & .. & .. & .. & .. & .. & .. & .. \\
.. & .. & .. & .. & .. & .. & .. & .. & .. \\
.. & .. & .. & .. & .. & .. & .. & .. & .. \\
.. & .. & .. & .. & .. & .. & .. & .. & .. \\
.. & .. & .. & .. & .. & .. & .. & B_1 & B_2
\end{pmatrix}, \tag{8.12}
$$

$$
P_{u,u} \;=\; \begin{pmatrix}
B_2 & B_3 & B_4 & .. & .. & .. & .. & B_{m-1} & C_{m-1} \\
B_1 & B_2 & B_3 & .. & .. & .. & .. & B_{m-2} & C_{m-2} \\
0 & B_1 & B_2 & .. & .. & .. & .. & B_{m-3} & C_{m-3} \\
0 & 0 & B_1 & B_2 & .. & .. & .. & B_{m-4} & C_{m-4} \\
.. & .. & .. & B_1 & B_2 & .. & .. & .. & .. \\
.. & .. & .. & .. & B_1 & B_2 & .. & .. & .. \\
.. & .. & .. & .. & .. & B_1 & B_2 & .. & .. \\
.. & .. & .. & .. & .. & 0 & B_1 & B_2 & C_2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & B_1 & C_1
\end{pmatrix}.
$$

Table 1: Comparison of complexity of different solution algorithms

Method	Complexity
Recursive calculation	$O(N^2)$
Linear level reduction	$O(Nm^2)$
Method of folding	$O(m^3 log\frac{N}{m})$
Theorem 6.1	$O(m^3)$

The matrices $A_0, A_1, A_2, P_{1,1}$ and $P_{u,u}$ are $(m-1) \times (m-1)$ matrices. In the case that $N+1$ is not an integer multiple of $m-1$, i.e, $N+1 = u(m-1) - c, 0 < c < m-1$, the presentation (8.10) changes as follow. The matrix $P_{u,u}$ is replaced by the matrix $(m-1-c) \times m$ matrix $P^*_{n,n}$ which consists of the $m-1-c$ right columns of $P_{n,n}$. Similarly, the matrix A_0 in right-most column of (8.10) is replaced by the $(m-c) \times m$ matrix A^*_0 which consists of the $m-c$ right columns of A_2.

Gaver et. al [5], [13, chap. 13] propose the *linear level reduction* method to solve QBD systems. Applied to (8.10), the complexity of this solution algorithm is $O(um^3) = O(Nm^2)$. Ye and Li [21] propose the *method of folding.* The application of this algorithm to solve (8.10) requires $O(m^3 log_2 u) = O(m^3 log_2 \frac{N}{m})$ computational steps.

8.3 Comparison of Complexities

We see from Table 1 that the complexity of Theorem 6.1 is lower than the complexity of the recursive calculations if $m = o(N^{2/3})$. Its complexity is lower than the complexity of the linear reduction based method if $m = o(N)$. Theorem 6.1 also outperforms the method of folding if N is sufficiently larger than m. However, the performance gain is only of the order $(log\frac{N}{m})^{-1}$, whereas Theorem 6.1 achieves a performance gain of the order m/N compared to the linear level reduction method. We note that in contrast to the recursive calculation, the linear level reduction approach and the folding method, the number of computations required to calculate a specific steady state probability π_k using Theorem 6.1 does not depend on the number of phases $N+1$.

In summary, we see that Theorem 6.1 outperforms the other approaches presented in this section if N is significantly larger than m. Thus, the applicability of Theorem 6.1 to the efficient solution of Markov models for skip-free processes depends on the specific choice of the parameters N and m describing the skip-free process.

9   Conclusions

This paper proposes a new solution algorithm for skip-free processes. The complexity of the algorithm is independent of the number of levels of the system. In consequence, it numerically outperforms previous algorithms if the skip parameter is small compared to the number of system levels. It is based on a novel method for deriving a closed-form solution for skip-free processes and analyzing this solution using Fibonacci sequences.

References

[1] A. Adhikari, *Skip free processes.* Ph.D thesis, Berkeley, 1986.

[2] C. B. Boyer, *Pascal's Formula for the Sums of Powers of the Integers.* Scripta Math. 9, 237-244, 1943.

[3] P.J. Brockwell, *The extinction time of a birth, death and catastrophe process and of a related diffusion model.* Advances Applied Probability, Vol. 17, 1985, 17 - 42.

[4] M.F. Chen, *Single birth processes,* Chinese Ann. Math. Ser. A, 20:77-82, 1999.

[5] D.P. Gaver, P.A. Jacobs and G. Latouche, *Finite birth-and-death models in randomly changing environments.* Advances in Applied Probability, 16:715-731, 1984.

[6] W.K Grassmann and D.A. Stanford, *Matrix analytic methods.* In: Computational Probability, WK Grassmann (Ed.), Kluwer, 2000, pp 153-202.

[7] L. Guen and A.M Makowski, *Matrix-geometric solution for finite capacity queues with phase-type distributions.* Performance'87, edited by Courtois, P.J.; Latouche, G.; Amsterdam, 1987, p. 269 - 282.

[8] E. Hansen, M. Patrick and J. Rusnack, *Some modifications of Laguere's method.* BIT, 17(1977), 409 -417.

[9] T. W. Hungerford, *Algebra.* Springer Publishing House, 1974.

[10] W.G. Kelley and A.C. Peterson, *Difference equations, An introduction with applications,* Academic Press, Inc. 1991.

[11] P.A. Jacobs, D.P. Gaver and G. Latouche, *Finite markov chain models skip free in one direction.* Naval Research Logistics Quarterly, 31, 1984, pp. 571-588.

[12] E. Kuntz, *Algebra.,* Vieweg Verlag, Braunschweig, 1991.

[13] G. Latouche and V. Ramaswami, *Introduction to matrix analytic methods in stochastic modeling.* Society for Industrial and Applied Mathematics, 1999.

[14] M.F. Neuts, *Matrix-geometric solutions in stochastic models. An Algorithmic Approach.* The John Hopkins University Press, Baltimore, MD, 1981.

[15] M.F. Neuts, *Structured stochastic matrices of M/G/1 type and their applications.* Marcel Dekker, New York, 1989.

[16] Y.V. Pan, *Solving a polynomial equation : Some history and recent progress.* Siam Rev., Vol. 39, No. 2, pp. 187 -220, June 1997.

[17] K. Prachar, *Primzahlverteilung.*, Berlin, Heidelberg, New York, Springer Verlag, 1978.

[18] W.J. STEWART, *On the use of numerical methods for ATM model.* Modeling and performance evaluation of ATM technology, edited by Perros, H.; Pujolle, G.; Takahashi, p. 375 - 396.

[19] D.A. WOLFRAM, *Solving generalized Fibonacci recurrences.* The Fibonacci Quarterly 36.2. May 1998, 129-45.

[20] S.J. YAN AND M.F. CHEN, *Multidimensional Q-processes.* Chin. Ann. Math. Ser. A,7:90-110, 1986

[21] J. YE AND S.Q. LI, *Folding algorithm: A computational method for finite QBD processes with level dependent transitions.* IEEE Transactions on Communications., 42:625-639, 1994.

[22] J.K. ZHANG, *Generalized birth-death processes.* Acta Mathematica Sinica, Vol. 46, 1984, 241 - 259.

[23] Y.H. ZHANG, *Strong ergodicity for single-birth processes.* Journal of Applied Probability, Vol. 38, 2001, 207 - 277.