An Efficient Algorithm for Exact Distributions of Soonest Waiting Time Problems in Markov Dependent Trials

Morteza Ebneshahrashoob morteza@csulb.edu
Department of Mathematics and Statistics, California State University, Long Beach, CA 90840, USA

Tangan Gao tgao@csulb.edu
Department of Mathematics and Statistics, California State University, Long Beach, CA 90840, USA

Mengnien Wu mwu@mail.tku.edu.tw
Department of Mathematics, Tamkang University, Danshui, Taiwan 251

Abstract. Waiting time random variables have a wide variety of interesting and useful applications. In this paper, we study the soonest waiting time for the independent multinomial as well as first-order Markov dependent trials via conditional probability generating functions. Our model is very general and covers various models proposed by many researchers. A computer algorithm is also developed to calculate the distributions, marginal probabilities, expectations and standard deviations. Numerical results show that the algorithm is very efficient and is capable of handling large problems.

Keywords: Inverse sampling, Probability generating functions, Independent multinomial trials, First-order Markov dependent trials.

AMS 2000 Subject Classification: 60E05, 60J10, 62L99.

1. Introduction

Waiting time random variables have been studied extensively for many decades and has many interesting applications (e.g., Balakrishnan and Koutras (2002)). Ebneshahrashoob and Sobel (1990) proposed an interesting class of waiting time problems for a sequence of independent Bernoulli trials with sooner and later stopping rules. This type of waiting time problems and its generalizations have been studied by many researchers since then, e.g, Aki, Balakrishnan and Mohanty (1996), Aki and Hirano (1999), Antzoulakos (1999), Antzoulakos and Philippou (1997), Balasubramanian, Viveros and Balakrishnan (1993), Chadjiconstantinidis, Antzoulakos and Koutras (2000), Doi and Yamamoto (1999), Ebneshahrashoob and Sobel (1994), Ebneshahrashoob and Sobel (1995), Fu and Lou (2003), Han and Aki (2000), Kolev and Minkova (1997), Koutras and Alexandrou (1997), Ling and Low (1993), Sobel and Ebneshahrashoob (1992), and Uchida and Aki (1995). In a recent book by Balakrishnan and Koutras (2002), chapters 6, 7, and 8 are devoted to these topics and their applications.

A general model which covers many models in the research mentioned above was proposed by Sobel and Ebneshahrashoob (1992) for independent multinomial trials. They used the Dirichlet methodology as a computational tool for the model. But in general the Dirichlet method is not computationally efficient for large application problems. The main goal of this paper is to generalize the model to first-order Markov dependent trials and develop efficient computer algorithms to solve large scale problems of the model.

Our first-order Markov dependent (or independent multinomial as a special case) $(\alpha, \beta, 1)$ model considered in this paper is given with $\alpha + \beta + 1$ disjoint cells. The first α cells are designated as frequency cells with integer frequency quotas $f_1, \ldots, f_\alpha$ and the next β cells are called run cells with integer run quotas $r_1, \ldots, r_\beta$. The last cell is a slack cell with no quota associated with it. For the first-order Markov dependent case, the initial cell probabilities are

$$(p_1, \ldots, p_\alpha, q_1, \ldots, q_\beta, p_0) \quad \text{with} \sum_{i=1}^{\alpha} p_i + \sum_{j=1}^{\beta} q_j + p_0 = 1$$

and the transaction cell probabilities

$$(p_{k1}, \ldots, p_{k\alpha}, q_{k1}, \ldots, q_{k\beta}, p_{k0}) \quad \text{with} \sum_{i=1}^{\alpha} p_{ki} + \sum_{j=1}^{\beta} q_{kj} + p_{k0} = 1$$

if the current event occurring in the k-th cell, $k = 1, \ldots, \alpha + \beta + 1$, or, for the independent multinomial case, the cell probabilities of the $\alpha + \beta + 1$ cells are

$$(p_1, \ldots, p_\alpha, q_1, \ldots, q_\beta, p_0) \quad \text{with} \sum_{i=1}^{\alpha} p_i + \sum_{j=1}^{\beta} q_j + p_0 = 1.$$

The scheme is to stop sampling as soon as one of these $\alpha + \beta$ quotas is satisfied. Let $WT(\alpha, \beta)$ denote the waiting time until at least one of the frequency or run quotas met. We wish to develop efficient methods to calculate exact distribution, marginal probabilities, expectation and standard deviation of the random variable $WT(\alpha, \beta)$.

There are at least four different methods for investigating the distributions and moments of waiting time random variables. They are: (i) the combinatorial method (e.g., Han and Aki (2000)), (ii) the probability generating function (pgf) method (e.g., Ebneshahrashoob, Gao and Sobel (2005)), (iii) the Dirichlet integral method (e.g., Ebneshahrashoob and Sobel (1994)), and (iv) the finite Markov chain imbedding method (e.g., Fu and Lou (2003)). To the best of our knowledge, there is no efficient algorithm available for solving large scale problems of our model.

The probability generating function (pgf) method has been used in research and education for many decades (e.g., Feller (1957)). The method provides a way to obtain the probability generating function and has many other interesting applications (e.g., Balakrishnan and Koutras (2002)). Since the difficulty of *symbolically* obtaining the probability generating functions of random variables, the pgf method is commonly regarded as a research tool, not a computational tool. During last several years, we have introduced sparse matrix computational tools into the pgf method and opened a new phase of the pgf method for large scale applications (e.g., Ebneshahrashoob, Gao and Sobel (2004), Ebneshahrashoob, Gao and Sobel (2005), and Ebneshahrashoob, Gao and Wu (2006)).

In Sections 2 and 3, we consider the independent multinomial case and the first-order Markov dependent case of the model respectively by using the pgf method and provide efficient methods for calculating the exact distribution, expectation, and standard deviation of $WT(\alpha, \beta)$. The calculation of marginal probabilities of any frequency and run cells chosen is discussed in Section 4. Numerical results are presented in Section 5 to show that our algorithm is very efficient and is capable of handling large problems. Executable codes of our algorithms for operating systems Windows NT/2000/XP and Linux are available upon request from the first author of this paper or can be directly downloaded from http://www.csulb.edu/∼mortezae/WTAlphaBeta/.

2. Independent Multinomial Case

Consider the independent multinomial case $(\alpha, \beta, 1)$ with cell probabilities

$$(p_1, \dots, p_\alpha, q_1, \dots, q_\beta, p_0) \quad \text{with} \sum_{i=1}^{\alpha} p_i + \sum_{j=1}^{\beta} q_j + p_0 = 1$$

of the $\alpha + \beta + 1$ cells and frequency and run quotas $(f_1, \dots, f_\alpha, r_1, \dots, r_\beta)$ of the α frequency cells and β run cells. The stopping rule is to stop sampling as soon as one of these $\alpha + \beta$ quotas is satisfied.

Let Y be a random variable which takes integer values $0, 1, 2, \dots$. The pgf of the distribution of Y can be formally written as

$$\sum_{n=0}^{\infty} P(Y = n) t^n$$

which is absolutely convergent for any $0 \le t \le 1$. Let $\phi[m_1, \dots, m_\alpha; n_1, \dots, n_\beta](t)$ denote the conditional pgf given that up to the present we have total frequency m_i in the i-th frequency cell for $i = 1, \dots, \alpha$ and a run of n_j elements in the j-th run cell for $j = 1, \dots, \beta$, where t is the parameter of the pgf. Clearly at most one of the values $n_1, \dots, n_\beta$ can be nonzero at any one time point and $\phi[0, \dots, 0; 0, \dots, 0](t)$ is the pgf of $WT(\alpha, \beta)$. The pgf method is to establish a system of linear equations consisting of these conditional pgf's and then solve the system for results related to $\phi[0, \dots, 0; 0, \dots, 0](t)$, the pgf of $WT(\alpha, \beta)$.

By the definition of the pgf's and the total probability formula, the pgf's of this system can be obtained according to the following two rules: the *main rule* for generating the pgf's is

$$\begin{aligned}
&\phi[m_1,\dots,m_\alpha;n_1,\dots,n_\beta](t)\\
&= p_1t\phi[m_1+1,\dots,m_\alpha;0,\dots,0](t)+\\
&\qquad p_2t\phi[m_1,m_2+1\dots,m_\alpha;0,\dots,0](t)+\cdots+\\
&\qquad p_\alpha t\phi[m_1,\dots,m_\alpha+1;0,\dots,0](t)+\\
&\quad q_1t\phi[m_1,\dots,m_\alpha;n_1+1,0,\dots,0](t)+\\
&\qquad q_2t\phi[m_1,\dots,m_\alpha;0,n_2+1,0,\dots,0](t)+\cdots+\\
&\qquad q_\beta t\phi[m_1,\dots,m_\alpha;0,\dots,0,n_\beta+1](t)+\\
&\quad p_0t\phi[m_1,\dots,m_\alpha;0,\dots,0](t)
\end{aligned} \tag{1}$$

if $m_i < f_i$ for $i = 1,\dots,\alpha$ and $n_j < r_j$ for $j = 1,\dots,\beta$, and the *reduction rule* is

$$\phi[m_1,\dots,m_\alpha;n_1,\dots,n_\beta](t) \equiv 1 \tag{2}$$

if $m_i = f_i$ for some $i = 1,\dots,\alpha$ or $n_j = r_j$ for some $j = 1,\dots,\beta$.

If we currently have total frequency m_i in the i-th frequency cell for $i = 1,\dots,\alpha$ and a run of n_i elements in the i-th run cell for $i = 1,\dots,\beta$, the next event may fall into one of the $\alpha+\beta+1$ cells with probabilities $p_1,\dots,p_\alpha,q_1,\dots,q_\beta$ and p_0. With this observation, the total probability formula leads to the equation in the main rule (1). The reduction rule (2) simply means that the pgf is constant when one of the quotas met.

With the main rule (1) and the reduction rule (2), it can be easily verified that there are a total of

$$\prod_{i=1}^{\alpha} f_i$$

possible non-constant pgf's for each fixed set of run values $(n_1,\dots,n_\beta)$, all zero or just one nonzero. Thus the total number of possible non-constant pgf's is

$$N \equiv [\sum_{j=1}^{\beta} r_j - (\beta-1)]\prod_{i=1}^{\alpha} f_i. \tag{3}$$

For efficiently generating the pgf's, we group the pgf's into

(i) $\phi[m_1,\dots,m_\alpha;0,\dots,0](t)$ for all possible $(m_1,\dots,m_\alpha)$, $0 \le m_i < f_i$, $i = 1,\dots,\alpha$, i.e., the case of no run in all run cells;

(ii) $\phi[m_1,\dots,m_\alpha;0,\dots,0,n_j,0,\dots,0](t)$ for all possible $(m_1,\dots,m_\alpha)$, $0 \le m_i < f_i$, $i = 1,\dots,\alpha$ and $0 < n_j < r_j$ for some $1 \le j \le \beta$, i.e., the case of a run in one of the run cells.

The pgf's in part (i) can be easily enumerated and the pgf's in part (ii) can be obtained by simply repeating the enumeration in part (i) when n_j going through $1, 2, \ldots, r_j - 1$ and j going through $1, 2, \ldots, \beta$. To apply this enumeration method, let (using T for transpose)

$$\Phi(t) = (\phi[0, \ldots, 0; 0, \ldots, 0](t), \ldots, \phi[f_1 - 1, \ldots, f_\alpha - 1; 0, \ldots, 0, r_\beta - 1](t))^T$$

be the vector of all non-constant pgf's $\phi[m_1, \ldots, m_\alpha; n_1, \ldots, n_\beta](t)$ arranged first according to the lexicographical order of their run values $(n_1, \ldots, n_\beta)$ and then according to the lexicographical order of their frequency values $(m_1, \ldots, m_\alpha)$. With this arrangement of the pgf's, the position for any given non-constant pgf $\phi[m_1, \ldots, m_\alpha; n_1, \ldots, n_\beta](t)$ in the vector $\Phi(t)$ can be easily determined by

$$\begin{array}{ll} 1 + \sum_{i=1}^{\alpha} m_i \prod_{k=i+1}^{\alpha} f_k & \text{if all } n_j = 0, \text{ or} \\ 1 + \sum_{i=1}^{\alpha} m_i \prod_{k=i+1}^{\alpha} f_k + (\sum_{i=j+1}^{\beta} (r_i - 1) + n_j) \prod_{i=1}^{\alpha} f_i & \text{if some } n_j \neq 0. \end{array} \tag{4}$$

Write the system of the pgf's in matrix form

$$\Phi(t) = tA\Phi(t) + tb \tag{5}$$

where A is an $N \times N$ (constant) matrix and b is an N-dimensional (constant) vector with N given in (3). The entries in A and b can be efficiently determined by symbolically generating the pgf's according to the main rule (1) and then applying the reduction rule (2) to every newly generated pgf to determine whether it is a new pgf, a constant pgf, or equivalent to one of the previous pgf's. For example, consider the pgf's at the right-hand side of the main rule (1). The position of each of the non-constant pgf's in the vector $\Phi(t)$ can be easily calculated by (4). These positions in the vector $\Phi(t)$ determine the nonzero entries in the matrix A and then the corresponding coefficient values p_i and q_j of the non-constant pgf's are assigned to the entries. If a pgf at the right-hand side of the main rule (1) is constant after the reduction rule applied, its corresponding coefficient p_i or q_j will be added to the corresponding entry of the vector b. Thus the matrix A and vector b can be very efficiently generated by a computer program. Also each row of the matrix A has no more than $\alpha + \beta + 1$ nonzero entries according to the main rule and the reduction rule. Thus the matrix A is very sparse and can be easily handled even when its dimension is very large (e.g., Section 3.4 of Saad (2003)).

If the pgf $\phi[0, \ldots, 0; 0, \ldots, 0](t)$ of $WT(\alpha, \beta)$ is available, it is well-known that the probabilities $P(WT(\alpha, \beta) = k)$ for $k = 0, 1, 2, \ldots$ satisfy

$$\begin{aligned} \phi[0, \ldots, 0; 0, \ldots, 0](0) &= P(WT(\alpha, \beta) = 0), \\ \phi^{(k)}[0, \ldots, 0; 0, \ldots, 0](0) &= k! P(WT(\alpha, \beta) = k), \quad k = 1, 2, \ldots, \end{aligned} \tag{6}$$

where $\phi^{(k)}[0,\ldots,0;0,\ldots,0](t)$ denotes the k-th derivative of the pgf $\phi[0,\ldots,0;0,\ldots,0](t)$ (e.g., Theorem 3.4.1 in Evans and Rosenthal (2004)). But for large values of α, β and frequency and run quotas, solving the system of pgf's in (5) for $\phi[0,\ldots,0;0,\ldots,0](t)$ and then for its derivatives symbolically is not always feasible because of computer memory and time restriction. Thus numerical method without explicitly solving for $\phi[0,\ldots,0;0,\ldots,0](t)$ becomes essential for solving this kind of problems with large parameter values.

With the matrix A and vector b in (5) available, calculating the probabilities $P(WT(\alpha,\beta)=k)$ for $k=0,1,2,\ldots$ is in fact very easy. Since

$$P(WT(\alpha,\beta)=k)=\frac{1}{k!}\phi^{(k)}[0,\ldots,0;0,\ldots,0](0),$$

repeatedly differentiating the equation (5), we obtain

$$\begin{aligned}\Phi'(t) &= A\Phi(t)+tA\Phi'(t)+b,\\ \Phi^{(k)}(t) &= kA\Phi^{(k-1)}(t)+tA\Phi^{(k)}(t), \quad k=2,3,\ldots.\end{aligned}$$

Plugging in $t=0$, these equations become

$$\begin{aligned}\Phi'(0) &= b,\\ \Phi^{(k)}(0) &= kA\Phi^{(k-1)}(0), \quad k=2,3,\ldots,\end{aligned}$$

and can be simply written as

$$\Phi^{(k)}(0)=k!A^{k-1}b, \quad k=1,2,\ldots. \tag{7}$$

Note that $\phi[0,\ldots,0;0,\ldots,0](t)$ is the first component of the vector $\Phi(t)$. By (6) and (7), we have

$$\begin{aligned}P(WT(\alpha,\beta)=0) &= 0,\\ P(WT(\alpha,\beta)=k) &= \text{the first component of } A^{k-1}b, \\ &\qquad \text{for all } k=1,2,\ldots.\end{aligned} \tag{8}$$

Since the matrix A is very sparse, the calculation of Ab can be easily done. In fact, it involves no more than $N(\alpha+\beta+1)$ multiplications of real numbers, here N is the dimension of the matrix A given in (3). Since A^kb can be calculated from $A(A^{k-1}b)$ and $P(WT(\alpha,\beta)=k)$ equals the first component of $A^{k-1}b$, the calculation of $P(WT(\alpha,\beta)=k)$ for all $k=0,1,\ldots,n$ (i.e., $P(WT(\alpha,\beta)\le n)$) involves no more than $(n-1)(\alpha+\beta+1)[\sum_{j=1}^{\beta}r_j-(\beta-1)]\prod_{i=1}^{\alpha}f_i$ multiplications of real numbers. Thus with parameter values of the problem fixed, this complexity dictates the efficiency of our algorithm. According to the nature of the problem, it can be shown that the matrix A is irreducible and thus its spectral radius $\rho(A)$ is less than 1 by Taussky theorem (e.g., Theorem 2 on page 376 in Lancaster and Tismenetsky (1985)). From (8), it can be shown that $P(WT(\alpha,\beta)=k)$ tends to zero as fast as $\rho(A)^{k-1}$ as k increases. When $\rho(A)$ is much smaller than 1 $P(WT(\alpha,\beta)=n)$ will be almost zero for small value of n and the whole computation can be terminated quickly. For the cases with large value of n, i.e., when $\rho(A)$ is very close to 1, $\rho(A)<1$ warrants the stability of calculating $A^n b$. Though the algorithm is numerically stable, this kind of computations should be always done in double precision to minimize accumulative errors for large n.

3. First-Order Markov Dependent Case

Consider the first-order Markov dependent case $(\alpha,\beta,1)$ with initial cell probabilities

$$(p_1,\ldots,p_\alpha,q_1,\ldots,q_\beta,p_0) \quad \text{with } \sum_{i=1}^{\alpha} p_i + \sum_{j=1}^{\beta} q_j + p_0 = 1$$

of the $\alpha+\beta+1$ cells and the transaction probabilities

$$(p_{k1},\ldots,p_{k\alpha},q_{k1},\ldots,q_{k\beta},p_{k0}) \quad \text{with } \sum_{i=1}^{\alpha} p_{ki} + \sum_{j=1}^{\beta} q_{kj} + p_{k0} = 1$$

if the current event belongs to the k-th cell of the $\alpha+\beta+1$ cells for $k=1,\ldots,\alpha+\beta+1$, and frequency and run quotas $(f_1,\ldots,f_\alpha,r_1,\ldots,r_\beta)$ of the α frequency cells and β run cells. The stopping rule is to stop sampling as soon as one of these $\alpha+\beta$ quotas is satisfied.

Let $\phi[m_1,\ldots,m_\alpha;n_1,\ldots,n_\beta]_k(t)$ denote the conditional pgf given that up to the present we have total frequency m_i in the i-th frequency quota cell for $i=1,\ldots,\alpha$ and a run of n_j elements in the j-th run cell for $j=1,\ldots,\beta$, and the current event belongs to the k-th cell, $k=1,\ldots,\alpha+\beta+1$, where t is the parameter of the pgf. The pgf system of this problem can be obtained according to the following two rules: the *main rule* for generating the pgf's is

$$\begin{aligned}
&\phi[0,\dots,0;0,\dots,0](t)\\
&= p_1 t\phi[1,0,\dots,0;0,\dots,0]_1(t)+\\
&\qquad p_2 t\phi[0,1,0\dots,0;0,\dots,0]_2(t)+\cdots+\\
&\qquad p_\alpha t\phi[0,\dots,0,1;0,\dots,0]_\alpha(t)+\\
&\ \ q_1 t\phi[0,\dots,0;1,0,\dots,0]_{\alpha+1}(t)+\\
&\qquad q_2 t\phi[0,\dots,0;0,1,0,\dots,0]_{\alpha+2}(t)+\cdots+\\
&\qquad q_\beta t\phi[0,\dots,0;0,\dots,0,1]_{\alpha+\beta}(t)+\\
&\ \ p_0 t\phi[0,\dots,0;0,\dots,0]_{\alpha+\beta+1}(t),\\
&\phi[m_1,\dots,m_\alpha;n_1,\dots,n_\beta]_k(t)\\
&= p_{k1} t\phi[m_1+1,\dots,m_\alpha;0,\dots,0]_1(t)+\\
&\qquad p_{k2} t\phi[m_1,m_2+1\dots,m_\alpha;0,\dots,0]_2(t)+\cdots+\\
&\qquad p_{k\alpha} t\phi[m_1,\dots,m_\alpha+1;0,\dots,0]_\alpha(t)+\\
&\ \ q_{k1} t\phi[m_1,\dots,m_\alpha;n_1+1,0,\dots,0]_{\alpha+1}(t)+\\
&\qquad q_{k2} t\phi[m_1,\dots,m_\alpha;0,n_2+1,0,\dots,0]_{\alpha+2}(t)+\cdots+\\
&\qquad q_{k\beta} t\phi[m_1,\dots,m_\alpha;0,\dots,0,n_r+1]_{\alpha+\beta}(t)+\\
&\ \ p_{k0} t\phi[m_1,\dots,m_\alpha;0,\dots,0]_{\alpha+\beta+1}(t)
\end{aligned} \tag{9}$$

if $m_i < f_i$ for $i = 1,\dots,\alpha$ and $n_j < r_j$ for $j = 1,\dots,\beta$, and the *reduction rule* is

$$\phi[m_1,\dots,m_\alpha;n_1,\dots,n_\beta]_k(t) \equiv 1 \tag{10}$$

if $m_i = f_i$ for some $i = 1,\dots,\alpha$ or $n_j = r_j$ for some $j = 1,\dots,\beta$. As in the independent multinomial case, the equation in the main rule (9) is based on the total probability formula and the reduction rule (10) simply means that the pgf is constant when one of the $\alpha+\beta$ quotas is satisfied. It can be verified that the total number of non-constant pgf's is

$$\begin{aligned}
N \ \equiv\ & 1+(r_1+\cdots+r_\beta-(\beta-1))f_1\cdots f_\alpha\\
&+\sum_{j=1}^{\alpha} j\ (\sum_{1\le i_1<i_2<\cdots<i_j\le\alpha}(f_{i_1}-1)\cdots(f_{i_j}-1)).
\end{aligned} \tag{11}$$

Since the transaction probabilities may be different when the current event falls into different cells, the pgf's $\phi[m_1,\ldots,m_\alpha;n_1,\ldots,n_\beta]_k(t)$ with the same $(m_1,\ldots,m_\alpha;n_1,\ldots,n_\beta)$ but different k's may also be different. We can group the pgf's into

(i) the initial pgf $\phi[0,\ldots,0;0,\ldots,0](t)$ of $WT(\alpha,\beta)$;

(ii) $\phi[m_1,\ldots,m_\alpha;0,\ldots,0]_{\alpha+\beta+1}(t)$ for all possible $(m_1,\ldots,m_\alpha)$, $0\le m_i<f_i$, $i=1,\ldots,\alpha$, i.e., the case of no run in all run cells and the current event in the slack cell;

(iii) $\phi[m_1,\ldots,m_\alpha;0,\ldots,0]_k(t)$ for all possible $(m_1,\ldots,m_\alpha)$, $0\le m_i<f_i$, $i=1,\ldots,\alpha$ and $0<m_k<f_k$ for some $1\le k\le\alpha$, i.e., the case of no run in all run cells and the current event in one of the frequency cells;

(iv) $\phi[m_1,\ldots,m_\alpha;0,\ldots,0,n_j,0,\ldots,0]_{\alpha+j}(t)$ for all possible $(m_1,\ldots,m_\alpha)$, $0\le m_i<f_i$, $i=1,\ldots,\alpha$ and $0<n_j<r_j$ for some $1\le j\le\beta$, i.e., the case of the current event in one of the run cells.

There is a natural 1-1 correspondence between the pgf's in parts (ii) and (iv) and the ones for the independent multinomial case in Section 2. To adapt the efficient way of generating the system of pgf's in Section 2 by using this correspondence, we arrange the vector $\Phi(t)$ of all non-constant pgf's in the following order: (a) the pgf in part (i) is its first component; (b) then followed by the pgf's in parts (ii) and (iv) as in Section 2; after this arrangement (c) all pgf's $\phi[m_1,\ldots,m_\alpha;0,\ldots,0]_k(t)$ with $(m_1,\ldots,m_\alpha)$ fixed in part (iii) are inserted, according to their indices of nonzero components, immediately after the position of their corresponding pgf $\phi[m_1,\ldots,m_\alpha;0,\ldots,0]_{\alpha+\beta+1}(t)$ in part (ii). With this arrangement of the pgf vector, the system of these pgf's

$$\Phi(t)=tA\Phi(t)+tb \tag{12}$$

can be generated by the same efficient method in Section 2 for the pgf's in parts (ii) and (iv) according to the main rule (9) and reduction rule (10), and all pgf's $\phi[m_1,\ldots,m_\alpha;0,\ldots,0]_k(t)$ with $(m_1,\ldots,m_\alpha)$ fixed in part (iii) can be obtained immediately after generating the corresponding pgf $\phi[m_1,\ldots,m_\alpha;0,\ldots,0]_{\alpha+\beta+1}(t)$ in part (ii).

Note that $\phi[0,\ldots,0;0,\ldots,0](t)$ is the first component of the vector $\Phi(t)$. Similar to the independent multinomial case, we have

$$\begin{aligned}P(WT(\alpha,\beta)=0)&=0,\\ P(WT(\alpha,\beta)=k)&=\text{the first component of }\ A^{k-1}b,\\ &\qquad\text{for all } k=1,2,\ldots.\end{aligned}\tag{13}$$

Since the matrix A is very sparse, the calculation of Ab involves no more than $N(\alpha+\beta+1)$ multiplications of real numbers, here N is the dimension of the matrix A given in (11). Since A^kb

can be calculated from $A(A^{k-1}b)$ and $P(WT(\alpha,\beta)=k)$ equals the first component of $A^{k-1}b$, the calculation of $P(WT(\alpha,\beta)=k)$ for all $k=0,1,\ldots,n$ (i.e., $P(WT(\alpha,\beta)\leq n)$) involves no more than $N(n-1)(\alpha+\beta+1)$ multiplications of real numbers with N given in (11). According to the nature of the problem, it can be shown that the sub-matrix from deleting the first row and first column of A is irreducible and its spectral radius is less than 1 by Taussky theorem (e.g., Theorem 2 on page 376 in Lancaster and Tismenetsky (1985)). Since the entries in the first column of A are all zero the spectral radius $\rho(A)$ of the matrix A is also less than 1. From (13), it can be shown that $P(WT(\alpha,\beta)=k)$ tends to zero as fast as $\rho(A)^{k-1}$ as k increases. When $\rho(A)$ is much smaller than 1 $P(WT(\alpha,\beta)=n)$ will be almost zero for small value of n and the whole computation can be terminated quickly. For the cases with large value of n, i.e., when $\rho(A)$ is very close to 1, $\rho(A)<1$ warrants the stability of calculating $A^n b$.

Remark 1: Though we assume that the cell probabilities and the frequency and run quotas are positive, our method and algorithm still work if some of the values are zero. Thus our method can be applied to special cases, say, when frequency cell, run cell and/or slack cell are not present.

4. Computation of Marginal Probabilities

In this section we discuss how to calculate the marginal probability (e.g., probability of acceptance and probability of rejection) for one or more frequency and/or run cells chosen. Let C be a set of frequency and/or run cells chosen for calculating their marginal probability.

For the independent multinomial case, we still use the main rule (1) to generate a system of pgf's

$$\Phi(t)=tA\Phi(t)+t\bar{b} \tag{14}$$

but with the following *reduction rule*: if a quota is reached in one of the frequency and run cells,

$$\phi[m_1,\ldots,m_\alpha;n_1,\ldots,n_\beta](t)\equiv\begin{cases}1 & \text{if the cell is in } C,\\ 0 & \text{if the cell not in } C.\end{cases} \tag{15}$$

or for the first-order Markov dependent case, we still use the main rule (9) to generate a system of pgf's

$$\Phi(t)=tA\Phi(t)+t\bar{b} \tag{16}$$

but with the following *reduction rule*: if a quota is reached in one of the frequency and run cells,

$$\phi[m_1,\ldots,m_\alpha;n_1,\ldots,n_\beta]_k(t)\equiv\begin{cases}1 & \text{if the cell is in } C,\\ 0 & \text{if the cell not in } C.\end{cases} \tag{17}$$

In both cases, the marginal probability of the cells in C is the sum of the first components of the vectors $A^{k-1}\bar{b}$, k=1,2,3,.... Note that calculating the marginal probability is as expensive as calculating the distribution of $WT(\alpha, \beta)$.

In the systems (14) and (16) we abuse the pgf notation $\Phi(t)$ to illustrate how to obtain the vector $\bar{b}$. Note that the matrix A in (14) is the same as the matrix in (5) and the matrix A in (16) is the same as the matrix in (12). The only difference is in the vector $\bar{b}$ in both cases. The vector $\bar{b}$ can be easily obtained simultaneously with the new reduction rule (15) or (10) when the algorithm generates the vector b.

Remark 2: The marginal probability above is alway associated with the slack cell unless the slack cell probability is zero.

5. Numerical Results

A computer program in C++ based on the methods discussed in Sections 2, 3 and 4 has been successfully implemented and tested on various combinations of the parameters α, β, f_i's, r_j's and probabilities. Our numerical results match those of the examples given in Balakrishnan and Koutras (2002). Our extensive testing shows that our algorithm is very efficient and is capable of solving large scale problems.

Executable codes of our methods for operating systems Windows NT/2000/XP and Linux are available upon request from the first author of this paper or can be directly downloaded from the web page http://www.csulb.edu/∼mortezae/WTAlphaBeta/.

All computation using double precision for the results in the three tables below was carried out on a 3.6 GHz Intel Xeon Pentium IV with 2 Gb memory running RedHat Enterprise Linux operating system. The algorithm is terminated when the condition $P(WT(\alpha, \beta) = n) < 10^{-10}$ is satisfied for some $n > 1$ since $P(WT(\alpha, \beta) = k)$ for $k = n+1, n+2, \ldots$ will be much smaller. The largest value of n for the results in Tables II and III is 452 for the first-order Markov dependent case with $(\alpha, \beta) = (4, 4)$. Numerical values are rounded to at least four digits after the decimal point.

Example 1: Consider a problem of small size given on page 278 of Balakrishnan and Koutras (2002): $\alpha = \beta = 1$, $f_1 = 6$, $r_1 = 10$, $p_1 = 0.5, 0.4, 0.3, 0.2$, $q_1 = 1 - p_1$ (and thus $p_0 = 0$) for the independent multinomial case. The results for this problem are presented in Table I, where MP_1 and MP_2 are the marginal probabilities for the first cell and the second cell. Only portions of the distributions $P(WT(\alpha, \beta) = k)$ for $k = 10, \ldots, 15$ are listed.

Table I. Probabilities $P(WT(\alpha,\beta)=k)$, expectations E, standard deviations σ and marginal probabilities MP for Example 1.

k	$p_1 = 0.5$	$p_1 = 0.4$	$p_1 = 0.3$	$p_1 = 0.2$
10	0.12402	0.07293	0.05030	0.11068
11	0.12354	0.08268	0.03935	0.02676
12	0.11328	0.09071	0.04810	0.02923
13	0.09717	0.09323	0.05602	0.03210
14	0.07904	0.09096	0.06256	0.03529
15	0.06158	0.08506	0.06737	0.03867
E	11.95905	14.68573	18.11318	20.54015
σ	3.39040	4.37296	5.51147	7.22784
MP_1	0.99415	0.96426	0.84204	0.50584
MP_2	0.00585	0.03574	0.15796	0.49416

Example 2: Consider the problems with parameters $(\alpha,\beta)=(1,2)$, (2,2), (2,3), (3,3), (3,4), (4,4), $f_i=20$ for $i=1,\ldots,\alpha$ and $r_j=10$ for $j=1,\ldots,\beta$, and cell probabilities

$$p_i = q_j = p_0 = \frac{1}{\alpha+\beta+1},\ i=1,\ldots,\alpha, j=1,\ldots,\beta$$

for the independent multinomial case and initial cell probabilities

$$p_i = q_j = p_0 = \frac{1}{\alpha+\beta+1},\ i=1,\ldots,\alpha, j=1,\ldots,\beta$$

and transaction cell probabilities

$$p_{ki} = q_{kj} = \frac{1}{\alpha+\beta+k},\ p_{k0} = 1-\sum_{i=1}^{\alpha} p_{ki} - \sum_{j=1}^{\beta} q_{kj},$$

$i=1,\ldots,\alpha, j=1,\ldots,\beta, k=1,\ldots,\alpha+\beta+1$ for the first-order Markov dependent case. Tables II and III list our results of the distributions, expectations, standard deviations and marginal probabilities of the first frequency cell. Only portions of the distributions $P(WT(\alpha,\beta)=k)$ for $k=100,\ldots,105$ are listed. The last row of each table gives the CPU times for solving the respective given problems. The CPU times in Tables II and III roughly match the computational complexity results of the problems mentioned in Sections 2 and 3. For example, the computation for the case (4,4) is expected at least $f_4=20$ times slower than that for the case (3,4). From the CPU times in Tables II and III we can conclude that our algorithm is very efficient and is capable of handling large problems.

Table II. Probabilities $P(WT(\alpha,\beta)=k)$, expectations E, standard deviations σ and marginal probability MP_1 for the independent multinomial case in Example 2. Last row Time stands for CPU times.

(α,β)	(1,2)	(2,2)	(2,3)	(3,3)	(3,4)	(4,4)
$k=100$	0.00986	0.01816	0.02271	0.01645	0.00630	0.00284
101	0.00913	0.01694	0.02295	0.01724	0.00679	0.00311
102	0.00843	0.01574	0.02310	0.01800	0.00730	0.00340
103	0.00777	0.01456	0.02319	0.01873	0.00782	0.00371
104	0.00715	0.01340	0.02319	0.01943	0.00836	0.00404
105	0.00656	0.01229	0.02311	0.02009	0.00892	0.00439
E	79.9952	87.4623	104.9553	114.3129	130.6433	140.4671
σ	15.4960	13.8815	17.2757	17.1237	20.0411	20.3418
MP_1	0.99989	0.49999	0.50000	0.33333	0.33333	0.25000
Time	0.01s	0.07s	0.12s	3.74s	6.48s	2m45.53s

Table III. Probabilities $P(WT(\alpha,\beta)=k)$, expectations E, standard deviations σ and marginal probabilities MP_1 for the first-order Markov dependent case in Example 2. Last row Time stands for CPU times.

(α,β)	(1,2)	(2,2)	(2,3)	(3,3)	(3,4)	(4,4)
k=100	0.01522	0.00948	0.00217	0.00061	0.00015	0.00003
k=101	0.01541	0.00995	0.00235	0.00068	0.00014	0.00004
k=102	0.01558	0.01042	0.00253	0.00074	0.00016	0.00004
k=103	0.01571	0.01088	0.00272	0.00082	0.00017	0.00005
k=104	0.01583	0.01134	0.00292	0.00090	0.00019	0.00005
k=105	0.01591	0.01180	0.00313	0.00098	0.00021	0.00006
E	113.6661	127.3584	155.5183	171.3894	197.7208	214.0576
σ	25.4990	23.9425	29.6004	29.7493	34.6162	35.3018
MP_1	0.99999	0.50000	0.50000	0.33333	0.33333	0.25000
Time	0.01s	0.11s	0.27s	7.86s	13.30s	6m2.64s

References

Aki, S., Balakrishnan, N., Mohanty, S.G. (1996). Sooner and later waiting time problems for success and failure runs in higher order Markov dependent trials. *Ann. Inst. Statist. Math.* 48:773-787.

Aki, S., Hirano, K. (1999). Sooner and later waiting time problems for runs in Markov dependent bivariate trials. *Ann. Inst. Statist. Math.* 51: pp.17-29.

Antzoulakos, D.L. (1999). On waiting time problems associated with runs in Markov dependent trials. *Ann. Inst. Statist. Math.* 51:323–330.

Antzoulakos, D.L., Philippou, A.N. (1997). Probability distribution functions of succession quotas in the case of Markov dependent trials. *Ann. Inst. Statist. Math.* 49:531–539.

Balakrishnan, N., Koutras, M.V. (2002). *Runs and scans with applications.* New York: Wiley.

Balasubramanian, K., Viveros, R., Balakrishnan, N. (1993). Sooner and later waiting time problems for Markovian Bernoulli trials. *Statist. Probab. Lett.* 18:153–161.

Chadjiconstantinidis, S., Antzoulakos, D.L., Koutras, M.V. (2000). Joint distributions of successes, failures and patterns in enumeration problems. *Adv. in Appl. Probab.* 32:866–884.

Doi, M., Yamamoto, E. (1999). On the joint distribution of runs in a sequence of multi-state trials. *Statist. Probab. Lett.* 39:133–141.

Ebneshahrashoob, M., Gao, T., Sobel, M. (2004). Double window acceptance sampling. *Naval Research Logistics* 51:297–306.

Ebneshahrashoob, M., Gao, T., Sobel, M. (2005). Sequential window problems. *J. Sequential Analysis* 24:159–175.

Ebneshahrashoob, M., Gao, T., Wu, M. (2006). An efficient algorithm for exact distribution of scan statistics. *Methodology and Computing in Applied Probability.* To appear in 2006.

Ebneshahrashoob, M., Sobel, M. (1990). Sooner and later waiting time problems for Bernoulli trials: Frequency and run quotas. *Statist. and Prob. Letters* 9:5–11.

Ebneshahrashoob, M., Sobel, M. (1994). Probability, waiting-time results for pattern and frequency quotas in the same inverse sampling problem via the Dirichlet. in *Approximation, probability, and related fields* (Santa Barbara, CA, 1993), pp. 179–186. New York:Plenum.

Ebneshahrashoob, M., Sobel, M. (1995). Dirichlet analysis for inverse multinomial with both quota and quota-free cells. *Computational Statistics & Data Analysis* 19:293–307.

Evans, M.J., Rosenthal, J.S. (2004). *Probability and statistics, the science of uncertainty.* New York: W. H. Freeman and Company.

Feller, W. (1957). *An introduction to probability theory and its applications.* New York: Wiley.

Fu, J.C., Lou, W.Y. (2003). *Distribution theory of runs and patterns and its applications.* Singapore: World Scientific Publisher.

Han, Q., Aki, S. (2000). Waiting time problems in a two-state Markov chain. *Ann. Inst. Statist. Math.* 52:778–789.

Kolev, N., Minkova, L. (1997). Run and frequency quotas in a multi-state Markov chain. *Comm. Statist. Theory Methods* 28:2223–2233.

Koutras, M.V., Alexandrou, V.A. (1997). Sooner waiting time problems in a sequence of trinary trials. *J. Appl. Probab.* 34:593–609.

Lancaster, P., Tismenetsky, M. (1985). *The theory of matrices : with applications*, 2nd Ed.. Orlando: Academic Press.

Ling, K.D., Low, T.Y. (1993). On the soonest and latest waiting time distributions: succession quotas., *Comm. Statist. Theory Methods* 22:2207–2221.

Saad, Y. (2003). *Iterative methods for sparse linear systems.* Philadelphia: SIAM.

Sobel, M., Ebneshahrashoob, M. (1992). Quota sampling for multinomial via Dirichlet. *J. Statistical Planning and Inference* 33:157-164.

Uchida, M., Aki, S. (1995). Sooner and later waiting time problems in a two-state Markov chain. *Ann. Inst. Statist. Math.* 47:415–433.