
Methodology and Computing in Applied Probability, 7, 459–471, 2005

An Efficient Algorithm for Exact Distribution of Discrete Scan Statistics

MORTEZA EBNESHAHRASHOOB mortezae@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

Received September 14, 2004; Revised April 15, 2005; Accepted June 23, 2005

Abstract. Waiting time random variables and related scan statistics have a wide variety of interesting and useful applications. In this paper, exact distribution of discrete scan statistics for the cases of homogeneous two-state Markov dependent trials as well as i.i.d. Bernoulli trials are discussed by utilizing probability generating functions. A simple algorithm has been developed to calculate the distributions. Numerical results show that the algorithm is very efficient and is capable of handling large problems.

Keywords: i.i.d. Bernoulli trials, two-state Markov dependent trials, waiting time random variables, probability generating functions, success runs

AMS 2000 Subject Classification: 60J22, 60E05, 60J10

1. Introduction

Waiting time random variables and related scan statistics have been studied extensively during the last four decades. There are four recent books in the area of runs, scan statistics, and their applications. They provide excellent and comprehensive review of historical as well as current developments in the distribution theory of runs and scan statistics. They are in order of publications: Scan Statistics and Applications by Glaz and Balakrishnan (1999), Scan Statistics by Glaz et al. (2001), Runs and Scans with Applications by Balakrishnan and Koutras (2002), and Distribution Theory of Runs and Patterns and its Applications by Fu and Lou (2003). In these books chapters are devoted to discrete scan statistics and many of their interesting and useful applications. They also provide excellent references in this research area. Due to complexity of computations of exact distribution of discrete scan statistics, most attention has been given to providing bounds, inequalities, and approximations for the distributions. There are several methods for investigating exact distribution of discrete scan statistics. Most notably, the combinatorial method [e.g., Naus (1974)] and the finite Markov chain imbedding method [e.g., Fu (2001)]. To the best of our knowledge, the leading algorithm up to date for

calculating exact distribution of scan statistics is the one given in Fu (2001) and Fu et al. (2003).

In this paper, we will apply the probability generating function (pgf) method to calculate exact distribution of discrete scan statistics. The pgf method has many other interesting applications [e.g., Balakrishnan et al. (1997), Ebneshahrashoob and Sobel (1990) and Ebneshahrashoob et al. (2005)]. In Section 2, we formally define discrete scan statistic $S_n(w)$ and waiting time random variable $WT(w, s)$ and discuss relationship between their distributions. Then we explain how to apply the pgf method to obtain the pgf of $WT(w, s)$ by symbolically solving a system of conditional probability generating functions and how to calculate the exact distribution of $WT(w, s)$ from its pgf. A special example is given to explain the pgf method. Due to the difficulty of symbolically obtaining the pgf of $WT(w, s)$, in Section 3, a new approach is proposed to directly calculate the distribution of $WT(w, s)$ without symbolically obtaining the pgf. We first provide an efficient method to generate all conditional probability generating functions related to the random variable $WT(w, s)$. Then we show that the exact distribution of $WT(w, s)$ can be obtained by simple multiplications of sparse matrices and vectors, which leads to an efficient algorithm for calculating the exact distribution of the scan statistic $S_n(w)$. Other scan statistics closely related to $S_n(w)$ and $WT(w, s)$ are also briefly discussed. In Section 4, numerical results are given to show that our algorithm is very efficient and is capable of handling large problems.

2. Probability Generating Function Method

Let $\{X_i\}_{i=1}^n$ be either a sequence of independent identically distributed (i.i.d.) Bernoulli trials, with outcomes success (or 1) and failure (or 0) and probabilities $P(X_i = 1) = p$ and $P(X_i = 0) = q = 1 - p$, or a sequence of homogeneous two-state Markov dependent trials with initial probabilities

$$p = P(X_1 = 1), \qquad q = P(X_1 = 0) \tag{1}$$

and transition probabilities

$$p_{ij} = P(X_k = j \mid X_{k-1} = i), \quad k \geq 2,\ 0 \leq i,\ j \leq 1 \tag{2}$$

with $p_{11} + p_{10} = p_{01} + p_{00} = 1$. The scan statistic $S_n(w)$ of window size w for the sequence $\{X_i\}_{i=1}^n$ is defined as

$$S_n(w) = \max\left\{ \sum_{j=i}^{w+i-1} X_j \,:\, 1 \leq i \leq n - w + 1 \right\}.$$

Let $WT(w, s)$ denote the waiting time until we first observe at least s successes (or 1's) in a window of size w. Then $S_n(w)$ and $WT(w, s)$ are related by

$$P(S_n(w) \geq s) = P(WT(w, s) \leq n). \tag{3}$$

In the remainder of this paper, we will use the probability generating function (pgf) method to obtain the exact distribution of $WT(w, s)$ and then obtain the exact distribution of $S_n(w)$ according to (3). With all $P(S_n(w) \geq s)$'s available, we can easily obtain the expectation and variance of $S_n(w)$ by

$$E(S_n(w)) = \sum_{s=1}^{w} P(S_n(w) \geq s) \tag{4}$$

and

$$\sigma^2(S_n(w)) = 2\sum_{s=1}^{w} sP(S_n(w) \geq s) - E(S_n(w))(1 + E(S_n(w))). \tag{5}$$

In the pgf method, we first establish a system of linear equations consisting of conditional probability generating functions. The solution of this system leads to an expression for the pgf of $WT(w, s)$. We use the special case of $s = 3$, $w = 5$ and any $0 \leq p$, $p_{ij} \leq 1$ as an example to illustrate the pgf method. General rules for generating the conditional pgf's and for eliminating redundant conditional pgf's are given in equations (9) and (10) at the end of this section. In this special case we have a total of $\binom{w}{s-1}+1 = \binom{5}{2} + 1 = 11$ equations:

$$\begin{aligned}
\phi &= pt\phi_1 + qt\phi_0,\\
\phi_0 &= p_{01}t\phi_1 + p_{00}t\phi_0,\\
\phi_1 &= p_{11}t\phi_{12} + p_{10}t\phi_2,\\
\phi_2 &= p_{01}t\phi_{13} + p_{00}t\phi_3,\\
\phi_3 &= p_{01}t\phi_{14} + p_{00}t\phi_4 = p_{01}t\phi_{14} + p_{00}t\phi_0,\\
\phi_{12} &= p_{11}t\phi_{123} + p_{10}t\phi_{23} = p_{11}t + p_{10}t\phi_{23},\\
\phi_{13} &= p_{11}t\phi_{124} + p_{10}t\phi_{24} = p_{11}t + p_{10}t\phi_{24},\\
\phi_{14} &= p_{11}t\phi_{125} + p_{10}t\phi_{25} = p_{11}t + p_{10}t\phi_2,\\
\phi_{23} &= p_{01}t\phi_{134} + p_{00}t\phi_{34} = p_{01}t + p_{00}t\phi_{34},\\
\phi_{24} &= p_{01}t\phi_{135} + p_{00}t\phi_{35} = p_{01}t + p_{00}t\phi_3,\\
\phi_{34} &= p_{01}t\phi_{145} + p_{00}t\phi_{45} = p_{01}t + p_{00}t\phi_0,
\end{aligned} \tag{6}$$

where p, q, p_{00}, p_{01}, p_{10} and p_{11} are defined in (1) and (2), and t acts as the parameter of the pgf's. In the equations above, ϕ is the pgf of $WT(w, s)$ which depends on the

conditional pgf's ϕ_1 and ϕ_0. The subscripts 1 of ϕ_1 and 0 of ϕ_0 denote the occurrence of success and failure on the first trial respectively. In general, the subscripts from left to right of the pgf's denote the number of steps back when successes have occurred. For example, ϕ_{13} denotes the pgf of the conditional distribution of the waiting time $WT(w, s)$ given that there were two successes one step back and three steps back respectively and no other in the window that extends $w = 5$ steps back. Note that $\phi_4 \equiv \phi_0$ since we are waiting for 3 successes in a window of size 5 and thus the success occurred at four steps back will no longer contribute to our experiment. Similarly, $\phi_{25} \equiv \phi_2$, $\phi_{35} \equiv \phi_3$, and $\phi_{45} \equiv \phi_0$. Also note that $\phi_{123} = \phi_{124} = \phi_{125} = \phi_{134} = \phi_{135} = \phi_{145} \equiv 1$ since three successes have occurred in a window of size $w = 5$. Let (using T for transpose)

$$\Phi = (\phi,\ \phi_0,\ \phi_1,\ \phi_2,\ \phi_3,\ \phi_{12},\ \phi_{13},\ \phi_{14},\ \phi_{23},\ \phi_{24},\ \phi_{34})^T.$$

Then the equations in (6) can be written in matrix form

$$\Phi(t) = tA\Phi(t) + tb \tag{7}$$

with the matrix A (omitting zeros in A except the first zero) and the vector b given by

$$A = \begin{pmatrix}
0 & q & p & & & & & & & & \\
 & p_{00} & p_{01} & & & & & & & & \\
 & & & p_{10} & & p_{11} & & & & & \\
 & & & & p_{00} & & p_{01} & & & & \\
 & p_{00} & & & & & & p_{01} & & & \\
 & & & & & & & & p_{10} & & \\
 & & & & & & & & & p_{10} & \\
 & & & p_{10} & & & & & & & \\
 & & & & & & & & & & p_{00} \\
 & & & & p_{00} & & & & & & \\
 & p_{00} & & & & & & & & &
\end{pmatrix},$$

$$b = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ p_{11} \\ p_{11} \\ p_{11} \\ p_{01} \\ p_{01} \\ p_{01} \end{pmatrix}. \tag{7}$$

Symbolically solving (7) for ϕ, we obtain

$$\phi(t) = (pt(1 - p_{00}t) + qtp_{01}t)\frac{Q(t)}{D(t)} \tag{8}$$

where

$$\begin{aligned} Q(t) = {} & p_{11}^2 t^2 + 2p_{01}p_{10}p_{11}t^3 + p_{01}^2 p_{10}^2 t^4 + 2p_{00}p_{01}p_{10}p_{11}t^4 \\ & - p_{00}p_{01}p_{10}p_{11}^2 t^5 - p_{00}p_{01}^2 p_{10}^2 p_{11}^2 t^7 - p_{00}^2 p_{01}^2 p_{10}^2 p_{11} t^7 \\ & - p_{00}p_{01}^3 p_{10}^3 p_{11} t^8 - p_{00}^2 p_{01}^3 p_{10}^3 p_{11} t^9 \end{aligned}$$

and

$$\begin{aligned} D(t) = {} & 1 - p_{00}t - p_{00}p_{01}p_{10}t^3 - p_{00}p_{01}^2 p_{10}^2 t^5 - p_{00}^2\, p_{01}p_{10}p_{11}t^5 \\ & + p_{00}^3\, p_{01}^2 p_{10}^2 p_{11} t^8 + p_{00}^3\, p_{01}^3 p_{10}^3\, p_{11} t^{10}. \end{aligned}$$

For the special case of i.i.d. Bernoulli trials, the first two equations in (6) become identical. The pgf of $WT(w, s)$ for this special case can be obtained from (8) by letting $p_{01} = p_{11} = p$ and $p_{00} = p_{10} = q$.

In general, let $\phi(t)$ be the pgf of the distribution of the waiting time $WT(w, s)$ which we solve for, let ϕ_0 denote the probability generating function of the conditional distribution of the waiting time given that the first trial was a failure, and let $\phi_{i_1,i_2,\ldots,i_k}(t)$ with $k \le s$ and $1 \le i_1 < i_2 < \cdots < i_k < w$ denote the probability generating function of the conditional distribution of the waiting time given that there was one success i_j steps back for each $j = 1,\ldots,k$ and no other in the window that extends w steps back. Then these pgf's can be obtained according to the following rules: the *main rules* for generating the pgf's are

$$\begin{aligned} \phi(t) &= pt\phi_1(t) + qt\phi_0(t), \\ \phi_0(t) &= p_{01}t\phi_1(t) + p_{00}t\phi_0(t), \\ \phi_{i_1,i_2,\ldots,i_k}(t) &= p_{11}t\phi_{1,i_1+1,i_2+1,\ldots,i_k+1}(t) + p_{10}t\phi_{i_1+1,i_2+1,\ldots,i_k+1}(t), \\ &\qquad \text{if } k < s, i_1 = 1, \\ \phi_{i_1,i_2,\ldots,i_k}(t) &= p_{01}t\phi_{1,i_1+1,i_2+1,\ldots,i_k+1}(t) + p_{00}t\phi_{i_1+1,i_2+1,\ldots,i_k+1}(t), \\ &\qquad \text{if } k < s, i_1 \ne 1, \end{aligned} \tag{9}$$

and the *reduction rules* for eliminating redundant pgf's are

$$\begin{aligned} \phi_{i_1}(t) &= \phi_0(t), && \text{if } w - i_1 < s - 1, \\ \phi_{i_1,i_2,\ldots,i_k}(t) &= \phi_{i_1,i_2,\ldots,i_{k-1}}(t), && \text{if } w - i_k < s - k, \\ \phi_{i_1,i_2,\ldots,i_k}(t) &\equiv 1, && \text{if } k = s \end{aligned} \tag{10}$$

Note that any information about successes that is more than w steps back will not affect the outcome of the experiment and thus can be dropped.

REMARK In the pgf methodology, the system of conditional pgf equations are constructed by considering the condition of one-step ahead from every condition [e.g., see Balakrishnan et al. (1997) and Section 4 of Chapter XIV in Feller (1957)]. Let Y be any random variable which takes only the integer values 0, 1, 2,.... The pgf of the distribution of Y can be formally written as $\sum_{n=0}^{\infty} P(Y=n)t^n$ which is convergent for any $0 \le t \le 1$. For the first equation in (9), we write the pgf's ϕ, ϕ_1 and ϕ_0 as

$$\phi(t) = \sum_{n=0}^{\infty} P(WT(w,s) = n)t^n,$$

$$\phi_1(t) = \sum_{n=0}^{\infty} P((WT(w,s) \mid X_1 = 1) = n)t^n,$$

$$\phi_0(t) = \sum_{n=0}^{\infty} P((WT(w,s) \mid X_1 = 0) = n)t^n$$

Due to the stopping rule of observing s successes in a window of size w, we have

$$\begin{aligned} &P(WT(w,s) = n+1) \\ &\quad = p\,P(WT(w,s) = n+1 \mid X_1 = 1) + q\,P(WT(w,s) = n+1 \mid X_1 = 0) \\ &\quad = p\,P((WT(w,s) \mid X_1 = 1) = n) + q\,P((WT(w,s) \mid X_1 = 0) = n) \end{aligned}$$

which leads to the pgf equation

$$\phi(t) = pt\phi_1(t) + qt\phi_0(t)$$

since the coefficients of t^n in both sides of the equation are same for all n. The second pgf equation in (9) is based on the fact

$$\begin{aligned} &P((WT(w,s) \mid X_1 = 0) = n+1) \\ &\quad = p_{01}\,P((WT(w,s) \mid X_1 = 0) = n+1 \mid X_2 = 1) \\ &\qquad + p_{00}\,P((WT(w,s) \mid X_1 = 0) = n+1 \mid X_2 = 0) \\ &\quad = p_{01}\,P((WT(w,s) \mid X_1 = 0 \,\&\, X_2 = 1) = n) \\ &\qquad + p_{00}\,P((WT(w,s) \mid X_1 = 0 \,\&\, X_2 = 0) = n) \\ &\quad = p_{01}\,P((WT(w,s) \mid X_1 = 1) = n) \\ &\qquad + p_{00}\,P((WT(w,s) \mid X_1 = 0) = n). \end{aligned}$$

Other pgf equations in (9) and (10) are constructed similarly.

If the pgf $\phi(t)$ of $WT(w, s)$ for any given values of the parameters w, s, p, p_{01} and p_{11} is available, it is well-known that the probabilities $P(WT(w, s) = k)$ for $k = 0, 1, \ldots, n$ satisfy

$$\begin{aligned} &\phi(0) = P(WT(w,s) = 0), \\ &\phi^{(k)}(0) = k!P(WT(w,s) = k), \quad k = 1, 2, \ldots, n, \end{aligned} \tag{11}$$

where $\phi^{(k)}$ is the k-th derivative of $\phi(t)$ [e.g., Theorem 3.4.1 in Evans and Rosenthal (2004)].

For large values of s and w, solving the system of conditional pgf's for $\phi(t)$ symbolically is not always feasible because of computer memory and time restriction. In Section 3, we first discuss how to efficiently generate the conditional pgf's and then propose an alternative method to calculate the derivative values of $\phi(t)$ at $t = 0$ directly without explicitly solving for $\phi(t)$.

3. The Algorithm

First, we discuss how to efficiently generate all conditional probability generating functions $\phi_{i_1,i_2,\ldots,i_k}(t)$ of $WT(w, s)$. With the main rules (9) and the reduction rules (10), it can be verified that for any integer k with $1 \le k < s$ there are exactly $\binom{w-s+k}{k}$ possible pgf's $\phi_{i_1,i_2,\ldots,i_k}$ with $1 \le i_1 < i_2 < \ldots < i_k \le w$. For each fixed value of k, the indices of the pgf's $\phi_{i_1,i_2,\ldots,i_k}(t)$ are exactly the combinations of choosing k numbers from the numbers 1 to $w - s + k$. If these $\binom{w-s+k}{k}$ pgf's are arranged according to the lexicographical order of their indices, they can be easily generated by a computer program. Including the pgf's ϕ and ϕ_0, we have the total

$$\begin{aligned} N &= 1 + \binom{w-s}{0} + \binom{w-s+1}{1} + \cdots + \binom{w-1}{s-1} \\ &= 1 + \binom{w}{s-1} \end{aligned} \tag{12}$$

possible non-constant pgf's. Let (using T for transpose)

$$\Phi(t) = (\phi(t), \phi_0(t), \phi_1(t), \ldots, \phi_{w-s+1,\ldots,w-2,w-1}(t))^T$$

be the vector of these pgf's arranged according to the ascending order of their numbers of indices and then the lexicographical order of their indices among the ones with the same number of indices. Then for any given parameters s, w, p, p_{01} and p_{11}, the system of the pgf's can be written in matrix form

$$\Phi(t) = tA\Phi(t) + tb \tag{13}$$

where A is an $N \times N$ (constant) matrix and b is an N-dimensional (constant) vector.

Note that because of the special ordering of the elements in $\Phi(t)$, the position of any pgf $\phi_{i_1,i_2,\ldots,i_k}(t)$ with $k \ge 1$ in the vector $\Phi(t)$ is determined by

$$\begin{aligned} &1 + \binom{w-s}{0} + \binom{w-s+1}{1} + \cdots + \binom{w-s+k-1}{k-1} \\ &\quad + \binom{w-s+k}{k} - \binom{w-s+k-i_1}{k} - \binom{w-s+k-i_2}{k-1} \\ &\quad - \binom{w-s+k-i_3}{k-2} - \cdots - \binom{w-s+k-i_k}{1} \end{aligned} \tag{14}$$

with the convention that $\binom{m}{n} = 0$ if $m < n$. The entries in A and b can be easily determined by symbolically generating the pgf's according to the main rules (9) and then applying the reduction rules (10) 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 equation

$$\phi_{i_1,i_2,\ldots,i_k}(t) = p_{01}t\phi_{1,i_1+1,i_2+1,\ldots,i_k+1}(t) + p_{00}t\phi_{i_1+1,i_2+1,\ldots,i_k+1}(t).$$

Let α be the position of $\phi_{i_1,i_2,\ldots,i_k}(t)$ in the vector $\Phi(t)$ according to (14). If $\phi_{1,i_1+1,i_2+1,\ldots,i_k+1}(t) \equiv 1$ after the reduction rules (10) applied, then the value of the α-th component of b will be p_{01}. If both $\phi_{1,i_1+1,i_2+1,\ldots,i_k+1}(t)$ and $\phi_{i_1+1,i_2+1,\ldots,i_k+1}(t)$ are non-constant after the reduction rules (10) applied, let β_1 and β_2 be the positions of the two reduced pgf's in the vector $\Phi(t)$ according to (14) respectively, then the values of the β_1-th and β_2-th components of the α-th row of A will be p_{01} and p_{00}. 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 two nonzero entries according to the main rules and the reduction rules. 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)]. It is also very interesting to point out that for given w and s our matrix A from the pgf method and the matrix $\mathcal{N}_{w,s}$ from the finite Markov chain imbedding method in Fu (2001) are the same under permutations of rows and columns.

Now, with the matrix A and vector b in (13) available, we can easily calculate the probabilities $P(WT(w, s) = k)$ for $k = 0, 1, \ldots, n$ according to (11). Since $P(WT(w,s) = k) = \frac{1}{k!}\phi^{(k)}(0)$, differentiating the Equation (13) up to n times, we obtain

$$\Phi'(t) = A\Phi(t) + tA\Phi'(t) + b,$$

$$\Phi^{(k)}(t) = kA\Phi^{(k-1)}(t) + tA\Phi^{(k)}(t), \quad k = 2, \ldots, n.$$

Plugging in $t = 0$, these equations become

$$\Phi'(0) = b,$$

$$\Phi^{(k)}(0) = kA\Phi^{(k-1)}(0), \quad k = 2, \ldots, n$$

and can be simply written as

$$\Phi^{(k)}(0) = k!A^{k-1}b, \quad k = 1, 2, \ldots, n. \tag{15}$$

Note that $\phi(t)$ is the first component of the vector $\Phi(t)$. By (11) and (15), we have

$$P(WT(w,s) = 0) = 0,$$

$$P(WT(w,s) = k) = \text{the first component of } A^{k-1}b, \quad \text{for all } k = 1, 2, \ldots, n \tag{16}$$

Since the matrix A is very sparse, the calculation of Ab can be easily done. In fact, it involves no more than $2N$ multiplications of real numbers, here $N = 1 + \binom{w}{s-1}$ is the dimension of the matrix A according to (12). Since $A^k b$ can be calculated from $A(A^{k-1}b)$

and $P(WT(w, s) = n)$ equals the first component of $A^{n-1}b$, the calculation of $P(WT(w, s) = k)$ for all $k = 0, 1, \ldots, n$ (i.e., $P(WT(w, s) \leq n)$) involves no more than $2N(n - 1)$ multiplications of real numbers. Thus with w and s fixed, the complexity of our algorithm for calculating $P(S_n(w) \geq s) = P(WT(w, s) \leq n)$ is no more than $2N(n-1) = 2(n-1)\left(1 + \binom{w}{s-1}\right)$ multiplications of real numbers. This complexity dictates the efficiency of our algorithm. Note that in our calculation of $P(S_n(w) \geq s)$ for all $s = 1, \ldots, w$, the largest value of N happens when s equals the integer part of $w/2$. Another numerical concern about calculating $A^k b$ for $k = 1, \ldots, n$ is its stability when the value of n is large. Let A' be the resulting matrix of deleting the first row and the first column from the matrix A in (13). Then the matrix A' is irreducible according to the nature of the problem. By Taussky theorem [e.g., Theorem 2 on page 376 in Lancaster and Tismenetsky (1985)], it can be shown that the spectral radius of A' is less than 1 and thus the spectral radius of A is also less than 1. This warrants the stability of the algorithm. Though the algorithm is numerically stable, this kind of computations should be always done in double precision to minimize accumulative errors for large n.

Our algorithm can now efficiently and safely calculate the distribution of $S_n(w)$ by utilizing $P(S_n(w) \geq s) = P(WT(w, s) \leq n)$. The algorithm can be easily applied to many other related statistics. Here we list a few important ones.

(1) The smallest number of consecutive trials that contain s ones, denoted by $W_s = \min_{s \leq w \leq n} \{w : S_n(w) = s\}$: [e.g., Glaz et al. (2001)]

$$P(W_s \leq w) = P(S_n(w) \geq s).$$

(2) The size of the longest number of consecutive trials that have at most r zeros, denoted by V_r and the special case V_0, the size of the longest run of ones: [e.g., Glaz et al. (2001) or Fu et al. (2003)]

$$P(V_r \geq s + r) = P(S_n(s + r) \geq s).$$

(3) The reliability of s-within-consecutive-w-out-of-n: F system consisting of n linearly ordered components is given by $P(S_n(w) < s)$ [e.g., Boutsikas and Koutras (2000) and references therein]. The system will fail if and only if there are w consecutive components which include among them, at least, s failed components.

(4) Type I negative binomial random variable, $WT_r^I(w, s)$, is the r-fold convolution of $WT(w, s)$ [e.g., Chapter 10 of Balakrishnan and Koutras (2002)]. Here, the pgf of $WT_r^I(w, s)$ is the r-th power of the pgf of $WT(w, s)$ for the case of i.i.d. Bernoulli trials.

(5) The number of non-overlapping windows of size at least w containing exactly s successes each, observed in n Bernoulli trials is denoted by $N_{n,s}^{(w,I)}$ where I stands for Type I enumeration scheme: [e.g., Chapter 11 of Balakrishnan and Koutras (2002)]

$$P(N_{n,s}^{(w,I)} \geq r) = P(WT_r^I(w, s) \leq n).$$

4. Numerical Results

A computer program in C++ based on the algorithm discussed in Section 3 has been successfully implemented and tested on various combinations of the parameters w, s, p, p_{01}, and p_{11}. Our algorithm is very efficient and matches results on all examples given in Tables 1 and 2 on page 914 in Fu (2001), results in Fu et al. (2003) and Table 4.1 on page 48 in Glaz et al. (2001). An executable code of our algorithm for operating system Windows NT/2000/XP or Linux is available upon request from the first author of this paper or can be directly downloaded from http://www.csulb.edu/~mortezae/ScanStat/. In this section, four tables of results will be given to show the efficiency of our algorithm. The computations using double precision were carried out on a 900 MHz Intel Xeon Pentium III with 2 Gb memory running Redhat Linux operating system. All numerical results listed for probabilities, expectations and standard deviations are rounded to four digits after the decimal point.

Tables 1 and 2 list our results of the two-state Markov dependent trials and i.i.d. Bernoulli trials for $w = 25$ and various values of n and probabilities. The last row of each table gives the CPU times for solving the respective given problems. The CPU times in

Table 1. Probabilities $P(S_n(w) \geq s)$, expectations E and standard deviations σ for two-state Markov dependent trials with $w = 25$. Last row Time stands for CPU times.

	$p = 0.5, p_{11} = 0.3, p_{01} = 0.4$			$p = 0.5, p_{11} = 0.4, p_{01} = 0.6$		
s	$n = 50$	$n = 100$	$n = 200$	$n = 50$	$n = 100$	$n = 200$
5	0.9999	1	1	1	1	1
6	0.9993	1	1	1	1	1
7	0.9952	1	1	1	1	1
8	0.9774	0.9998	1	1	1	1
9	0.9244	0.9972	1	0.9998	1	1
10	0.8114	0.9787	0.9997	0.9986	1	1
11	0.6347	0.9061	0.9938	0.9909	1	1
12	0.4276	0.7373	0.9445	0.9604	0.9994	1
13	0.2422	0.4931	0.7731	0.8764	0.9922	1
14	0.1136	0.2612	0.4866	0.7150	0.9486	0.9983
15	0.0437	0.1086	0.2255	0.4949	0.8074	0.9719
16	0.0137	0.0357	0.0781	0.2791	0.5540	0.8292
17	0.0035	0.0093	0.0208	0.1248	0.2865	0.5256
18	0.0007	0.0019	0.0044	0.0435	0.1091	0.2273
19	0.0001	0.0003	0.0007	0.0116	0.0307	0.0677
20			0.0001	0.0023	0.0064	0.0144
21				0.0003	0.0010	0.0022
22					0.0001	0.0002
E	11.1875	12.5292	13.5273	14.4978	15.7354	16.6368
σ	1.9199	1.6036	1.3771	1.7550	1.4466	1.2360
Time	5m32s	10m55s	21m55s	5m38s	10m58s	21m36s

Table 2. Probabilities $P(S_n(w) \geq s)$, expectations E and standard deviations σ for i.i.d. Bernoulli trials with $w = 25$ and $n = 100$. Last row Time stands for CPU times.

s	$p = 0.01$	$p = 0.05$	$p = 0.1$	$p = 0.2$	$p = 0.3$	$p = 0.4$	$p = 0.5$
1	0.6340	0.9941	1	1	1	1	1
2	0.1494	0.8962	0.9979	1	1	1	1
3	0.0172	0.5877	0.9671	1	1	1	1
4	0.0013	0.2525	0.8271	0.9995	1	1	1
5	0.0001	0.0744	0.5559	0.9933	1	1	1
6		0.0162	0.2798	0.9565	0.9999	1	1
7		0.0028	0.1071	0.8415	0.9982	1	1
8		0.0004	0.0323	0.6315	0.9865	1	1
9			0.0079	0.3875	0.9387	0.9995	1
10			0.0016	0.1926	0.8179	0.9956	1
11			0.0003	0.0782	0.6184	0.9757	0.9999
12				0.0264	0.3914	0.9105	0.9987
13				0.0075	0.2046	0.7697	0.9907
14				0.0018	0.0885	0.5610	0.9572
15				0.0004	0.0318	0.3413	0.8641
16				0.0001	0.0096	0.1708	0.6892
17					0.0024	0.0701	0.4620
18					0.0005	0.0236	0.2519
19					0.0001	0.0065	0.1098
20						0.0014	0.0378
21						0.0002	0.0101
22							0.0020
23							0.0003
E	0.8019	2.8243	4.7770	8.1166	11.0884	13.8259	16.3739
σ	0.7312	1.1413	1.3888	1.6531	1.7693	1.7916	1.7379
Time	4s	2m6s	6m40s	10m12s	10m14s	10m14s	10m15s

Table 1 matches the result that with w fixed, the complexity of our algorithm is proportional to n. For the calculation of the results in these two tables, the algorithm requires about 200 Mb memory. And the algorithm is terminated when the condition $P(S_n(w) \geq s_0) < 10^{-6}$ is satisfied for some $1 \leq s_0 \leq w$ since $P(S_n(w) \geq s)$ for $s = s_0 + 1, \ldots,$ w will be much smaller. This explains the difference in CPU times for the cases with same values of n and w in the tables.

Table 3. Probabilities $P(S_n(w) \geq w)$ for two-state Markov dependent trials with $p = 0.5$, $p_{11} = 0.75$, $p_{01} = 0.25$. The second row Time stands for CPU times.

	$w = 40$		$w = 60$		$w = 80$	
	$n = 10^6$	$n = 10^7$	$n = 10^6$	$n = 10^7$	$n = 10^6$	$n = 10^7$
Prob.	0.8129	1	0.0053	0.0518	0.00002	0.00017
Time	1.5s	14.7s	2.3s	22.5s	2.9s	28.5s

Table 4. CPU times of calculating $P(S_n(w) \geq w)$ for two-state Markov dependent trials with $p = 0.5$, $p_{11} = 0.75$, $p_{01} = 0.25$, $w = 500$ and large values of n.

$n = 10^4$	$n = 10^5$	$n = 10^6$	$n = 10^7$	$n = 10^8$
0.18s	1.76s	17.52s	2m55s	29m8s

Tables 3 and 4 list our results for the longest success run in a sequence of two-state Markov dependent trials for the cases $p = 0.5$, $p_{11} = 0.75$, $p_{01} = 0.25$ and various values of w and n as discussed in Fu et al. (2003). The distribution of the longest success run is given by $P(S_n(w) \geq w)$ and the dimension of the corresponding matrix A is only $N = w + 1$. With window size w and number of trials n given, the complexity of our algorithm for calculating $P(S_n(w) \geq w)$ is no more than $2(w + 1)(n - 1)$ multiplications of real numbers, which enables our algorithm to efficiently handle large problems. The CPU times in Tables 3 and 4 reflect this complexity of our algorithm very well. It appears that our algorithm is more efficient than the leading algorithms for exact distributions of scan statistics developed in Fu (2001) and Fu et al. (2003). As an example, for the case $w = 500$ and $n = 10{,}000$ as shown in Table 4, our algorithm written in C++ takes about 0.18 second on a 900 MHz Pentium III while the algorithm in Fu et al. (2003) written in MatLab takes about 33 seconds on a 733 MHz Pentium III as reported in the paper. But making strict comparison of these algorithms is impossible since algorithms written in different computer languages may result in different CPU times.

Acknowledgments

The authors sincerely thank both the referees for their constructive suggestions and comments that led to substantial improvements in the presentation and J.C. Fu, J. Glaz and W.Y. Lou for their e-mail communications during the preparation of this paper.

References

N. Balakrishnan and M. V. Koutras, *Runs and Scans with Applications*, Wiley: New York, 2002.

N. Balakrishnan, S. G. Mohanty, and S. Aki, "Start-up demonstration tests under Markov dependence model with corrective actions," *Annals of the Institute of Statistical Mathematics* vol. 49 pp. 155–169, 1997.

M. V. Boutsikas and M. V. Koutras, "Reliability approximation for Markov chain imbeddable systems," *Methodology and Computing in Applied Probability* vol. 2 pp. 393–411, 2000.

M. Ebneshahrashoob and M. Sobel, "Sooner and later waiting time problems for Bernoulli trials: Frequency and run quotas," *Statistics & Probability Letters* vol. 9 pp. 5–11, 1990.

M. Ebneshahrashoob, T. Gao, and M. Sobel, "Sequential window problems," *Sequential Analysis* vol. 24 pp. 159–175, 2005.

M. J. Evans and J. S. Rosenthal, *Probability and Statistics, The Science of Uncertainty*, W. H. Freeman and Company: New York, 2004.

W. Feller, *An Introduction to Probability Theory and Its Applications*, Wiley: New York, 1957.

J. C. Fu, "Distribution of the scan statistics for a sequence of bistate trials," *Journal of Applied Probability* vol. 38 pp. 908–916, 2001.

J. C. Fu and W. Y. Lou, *Distribution Theory of Runs and Patterns and Its Applications*, World Scientific Publisher: Singapore, 2003.

J. C. Fu, L. Q. Wang, and W. Y. Lou, "On exact and large deviation approximation for the distribution of the longest run in a sequence of two-state Markov dependent trials," *Journal of Applied Probability* vol. 40 pp. 346–360, 2003.

J. Glaz and N. Balakrishnan (eds.), *Scan Statistics and Applications*, Birkhäuser: Boston, 1999.

J. Glaz, J. I. Naus, and S. Wallenstein, *Scan Statistics*, Springer-Verlag: New York, 2001.

P. Lancaster and M. Tismenetsky, *The Theory of Matrices: With Applications*, Academic Press: Orlando, 2nd edition, 1985.

J. I. Naus, "Probabilities for a generalized birthday problem," *Journal of the American Statistical Association* vol. 69 pp. 810–815, 1974.

Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM: Philadelphia, 2003.