A combinatorial problem from sooner waiting time problems with run and frequency quotas

Tangan Gao\textsuperscript{a}, Mengnien Wu\textsuperscript{b,\*}

\textsuperscript{a} Department of Mathematics and Statistics, California State University, Long Beach, CA 90840, USA
\textsuperscript{b} Department of Mathematics, Tamkang University, Danshui, Taiwan 251, Taiwan

Received 30 January 2006
Available online 24 April 2006
Submitted by William F. Ames

Abstract

In this paper, we study the problem of finding the number of integer solutions solving

\[ z_1 + \cdots + z_k \leq w, \quad 1 \leq z_i \leq r, \quad i = 1, \ldots, k, \quad 1 \leq k < f \]

for given \( f, r, w \in \mathbb{N} \) with \( w \geq \max(f, r) \). This problem is naturally from calculating exact distributions of some sooner waiting time random variables of run and frequency quotas in statistics. We present several solutions to the problem and develop an algorithm for the sooner waiting time problems. Numerical results are given to show the efficiency of our algorithm for calculating the exact distributions of the sooner waiting time random variable.

\( \text{©} \ 2006 \text{ Elsevier Inc. All rights reserved.} \)

\textbf{Keywords:} Combinatorial counting problem; Multinomial coefficients; Markov dependent trials; Waiting time random variables; Probability generating functions

\* Corresponding author.
\textit{E-mail address:} mwu@mail.tku.edu.tw (M. Wu).

0022-247X/ – see front matter \( \text{©} \ 2006 \text{ Elsevier Inc. All rights reserved.} \)
doi:10.1016/j.jmaa.2006.03.071
1. Introduction

A sequence of 0 and 1 randomly entering a window of size $w$ from the left with given probabilities,

\[ \cdots 00110010101001011010110110 \rightarrow \]

and this window is the only part of the sequence we can see. At any time, keep monitoring if (a) the number of consecutive 0’s in the window is fewer than a fixed number $r \leq w$, and (b) the number of 1’s in the window is fewer than a fixed number $f \leq w$, otherwise, stop the process. We would like to know how many different states (parts of the sequence seen in the window) and how to enumerate them efficiently. This problem can be modeled as Problem 1: Given $f, r, w \in \mathbb{N}$ with $w \geq \max(f, r)$, find the number of integer solutions solving

\[ z_1 + \cdots + z_k \leq w \quad \text{with } 1 \leq z_i \leq r \quad \text{for } i = 1, \ldots, k, \quad 1 \leq k < f. \tag{1} \]

If let $\bar{\chi}_k$ be the accumulated amount $z_1 + \cdots + z_k$ and $\bar{\chi}_0 := 0$, then (1) becomes

\[ \bar{\chi}_k \leq w \quad \text{with } 1 \leq z_i = \bar{\chi}_i - \bar{\chi}_{i-1} \leq r \quad \text{for } i = 1, \ldots, k, \quad 1 \leq k < f. \]

Therefore Problem 1 is equivalent to: find the cardinality of $\bigcup_{k=1}^{f-1} S_k$ defined in (2). Define

\[ n_k^w := \begin{cases} 0, & \text{if } w < \min(k, r), \\ \text{the cardinality of } S_k, & \text{if } \min(k, r) \leq w \leq rk. \end{cases} \]

where $\bar{\chi}_j$ = position of $j$th 1 (from the left), and $k$ = number of 1’s in the window.

The problem above is naturally from the study of some sooner waiting time random variables by using probability generating function (pgf) method. Since 1990 many papers have been written on the study of distributions and moments of sooner waiting time random variables and their applications with inverse sampling schemes in a Bernoulli or multinomial setting [1,6]. A closely related concept is the idea of scan (window) statistics [7,8]. The pgf method provides a way to obtain the pgf’s of random variables and has many other interesting applications (e.g., [11]). Due to the difficulty of symbolically obtaining the pgf’s, this 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., [2–5]).

In this paper, we present several results for the cardinality of $\bigcup_{k=1}^{f-1} S_k$ in Section 2 and also apply the results to solve another related problem. As an application of the results in Section 2.3, we study the exact distributions of some sooner waiting time random variables in Section 3 and present a numerical algorithm to calculate the distributions. Numerical results show that our algorithm is very efficient and capable of handling large problems.

2. Main results

In this section, for $f, r, w \in \mathbb{N}$ with $w \geq \max(f, r)$ given, we present several formulae for the cardinality of $\bigcup_{k=1}^{f-1} S_k$ defined in (2). Define

\[ n_k^w := \begin{cases} 0, & \text{if } w < \min(k, r), \\ \text{the cardinality of } S_k, & \text{if } \min(k, r) \leq w \leq rk. \end{cases} \]
It is clear that for fixed window size \( w \), (a) \( S_k \)'s are disjoint, (b) each \( S_k \) corresponds exactly \( k \) check marks (or 1's), and (c) the last mark \( \check{v}_k \) cannot be placed further than \( r_k \), which implies that the cardinality of \( S_k \) remains the same for windows of size larger than \( r_k \), i.e., \( n^w_k = n^r_k \) for \( w \geq r_k \).

2.1. A recursive formula

\[
\begin{array}{c|c|c|c|c|c|c}
\ldots & \ldots & \check{v}_k & \ldots \\
\end{array}
\]

\[
\cong \begin{array}{c|c|c|c|c|c|c}
\ldots & \ldots & \check{v}_k & \ldots \\
\end{array}
\]

For a fixed window of size \( w \leq r_k \), suppose there are \( k \) check marks in it. Placing the last mark \( \check{v}_k \) at the \( j \)th slot is nothing more than placing \( k-1 \) check marks into the window of size \( j-1 \) as shown. Therefore,

Formulra 1. The cardinality of \( S_k \) is

\[
n^w_k := \sum_{\ell = w-r}^{w-1} n^\ell_{k-1}
\]

with initial values \( n^w_1 = 1, 2, \ldots, r-1, r, r, \ldots \), for \( w = 1, 2, 3, \ldots \).

Note that if \( \check{v}_k \) is \( w \), then the left most \( \check{v}_{k-1} \) is \( w-r \). That is why the summation of \( n^\ell_{k-1} \)'s starts from \( \ell = w-r \).

2.2. Combinatorial formulae

Consider the multinomial expansion of a degree \( k \) homogeneous polynomial

\[
(b_0 + \cdots + b_{r-1}) \cdots (b_0 + \cdots + b_{r-1}) = \sum_{\|p\|_\infty < r} b_{p_1} \cdots b_{p_k},
\]

where \( b_{p_i} \) is the term chosen from the \( i \)th factor when multiplying. Then we may read \( p_i \) as “the number of blanks between \( \check{v}_{i-1} \) and \( \check{v}_i \)” and \( p := (p_1, \ldots, p_k) \) can precisely represent a check mark state

\[
\begin{array}{c|c|c|c|c|c|c}
dummy \check{v}_0 & \check{v}_1 & \check{v}_2 & \check{v}_3 & \ldots \\
p_1 & 0 & 1 & 0 & 0 & 1 & \ldots \\
p_2 & 1 & 0 & 0 & 1 & 0 & \ldots \\
p_3 & 0 & 0 & 1 & 0 & 0 & \ldots \\
\end{array}
\]

Define \( \ell(p) := \) position of the last mark = \( \check{v}_k \). Clearly, \( k \leq \ell(p) \leq r_k \), and \( \ell(p) = p_1 + \cdots + p_k + k \). Group all terms in the expansion by the \( \ell(p) \),

\[
(b_0 + \cdots + b_{r-1})^k = \sum_{\|p\|_\infty < r} b_{p_1} \cdots b_{p_k} = \sum_{\|p\|_\infty < r} \sum_{\ell(p)=k} b_{p_1} \cdots b_{p_k},
\]
where \( \|p\|_\infty \) is the \( \infty \)-norm of \( p \). Set \( b_0 = b_1 = \cdots = b_{r-1} = 1 \),

\[
r^k = \sum_{\|p\|_\infty < r} 1 = \sum_{\|p\|_\infty < r} \sum_{\ell(p) = k} 1.
\]

If we restrict \( \ell(p) \) within \( w \leq rk \), then the cardinality of \( \{p : \|p\|_\infty < r \text{ and } \ell(p) \leq w\} \) equals \( n_k^w \) exactly, and therefore we have

**Formula 2.** The cardinality of \( S_k \) is

\[
n_k^w := \begin{cases} \sum_{\|p\|_\infty < r} \sum_{\ell(p) = k} 1, & \text{if } k \leq w \leq rk, \\ \|p\|_\infty = k, & \text{if } w \geq rk. \end{cases}
\]

After combining like-terms in (4), a monomial \( b_0^{q_0}b_1^{q_1} \cdots b_{r-1}^{q_{r-1}} \) may come from several different check mark states \( p \)'s, and for each \( p \), there are exactly \( q_i \) many check marks which have \( i \) consecutive blanks (an \( i \)-blank set, \( 0 \leq i < r \)) on their left, namely, \( q_i = \text{number of } i \)-blank sets = number of \( i \)'s among \( p_1, \ldots, p_k \). Note that \( q_0 + \cdots + q_{r-1} = k \) and \( p_1, \ldots, p_k \) must be one of \( 0, 1, \ldots, r - 1 \). Let \( q := (q_0, \ldots, q_{r-1}) \) and \( \|q\|_1 \) be the \( 1 \)-norm of \( q \). Define \( \ell(q) := \text{position of the last mark} \). Clearly, if \( \|q\|_1 = k \), then \( k \leq \ell(q) = k \leq rk \) and \( \ell(q) = 0 \cdot q_0 + 1 \cdot q_1 + 2 \cdot q_2 + \cdots + (r - 1) \cdot q_{r-1} + k \).

**Example 3.** For \( r = 2 \) and \( k = 3 \),

\[
(b_0 + b_1)(b_0 + b_1)(b_0 + b_1) = \sum_{\|p\|_\infty < 2} b_{p_1}b_{p_2}b_{p_3} = \sum_{\|q\|_1 = 3} \frac{3!}{q_0!q_1!} b_0^{q_0}b_1^{q_1}.
\]

<table>
<thead>
<tr>
<th>( (p_1, p_2, p_3) )</th>
<th>Check marks</th>
<th>( (q_0, q_1) )</th>
<th>( \ell = \sqrt{3} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( (0, 0, 0) )</td>
<td>( \checkmark \checkmark \checkmark )</td>
<td>( (3, 0) )</td>
<td>( 3 )</td>
</tr>
<tr>
<td>( (0, 0, 1) )</td>
<td>( \checkmark \checkmark \checkmark )</td>
<td>( (2, 1) )</td>
<td>( 4 )</td>
</tr>
<tr>
<td>( (0, 1, 0) )</td>
<td>( \checkmark \checkmark \checkmark )</td>
<td>( (2, 1) )</td>
<td>( 4 )</td>
</tr>
<tr>
<td>( (0, 1, 1) )</td>
<td>( \checkmark \checkmark \checkmark )</td>
<td>( (1, 2) )</td>
<td>( 5 )</td>
</tr>
<tr>
<td>( (1, 0, 0) )</td>
<td>( \checkmark \checkmark \checkmark )</td>
<td>( (2, 1) )</td>
<td>( 4 )</td>
</tr>
<tr>
<td>( (1, 0, 1) )</td>
<td>( \checkmark \checkmark \checkmark )</td>
<td>( (1, 2) )</td>
<td>( 5 )</td>
</tr>
<tr>
<td>( (1, 1, 0) )</td>
<td>( \checkmark \checkmark \checkmark )</td>
<td>( (1, 2) )</td>
<td>( 5 )</td>
</tr>
<tr>
<td>( (1, 1, 1) )</td>
<td>( \checkmark \checkmark \checkmark \checkmark )</td>
<td>( (0, 3) )</td>
<td>( 6 )</td>
</tr>
</tbody>
</table>

\[
(b_0 + b_1)^3 = b_0b_0b_0 + b_0b_0b_1 + b_0b_1b_0 + b_1b_0b_0 + b_0b_1b_1 + b_1b_0b_1 + b_1b_1b_0 + b_1b_1b_1 + b_1b_1b_1
\]

\[
= b_0^3 + b_0^2b_1 + b_0b_1^2 + b_1^3.
\]

Group all monomials in the expansion by \( \ell(q) \),
\[(b_0 + b_1 + \cdots + b_{r-1})^k = \sum_{\|q\|_1 = k} \frac{k!}{q_0! \cdots q_{r-1}!} b_0^{q_0} \cdots b_{r-1}^{q_{r-1}}
\]
\[= \sum_{\|q\|_1 = k} \sum_{\ell(q) = k} \frac{k!}{q_0! \cdots q_{r-1}!} b_0^{q_0} \cdots b_{r-1}^{q_{r-1}}.
\]
Set \(b_0 = b_1 = \cdots = b_{r-1} = 1,\)
\[r^k = \sum_{\|q\|_1 = k} \frac{k!}{q_0! \cdots q_{r-1}!} = \sum_{\|q\|_1 = k} \sum_{\ell(q) = k} \frac{k!}{q_0! \cdots q_{r-1}!}.
\]
If we restrict \(\ell(q)\) within \(w \leq r^k\), then we have

**Formula 4.** The cardinality of \(S_k\) is
\[n_k^w := \left\{ \sum_{\|q\|_1 = k} \sum_{\ell(q) = k} \frac{k!}{q_0! \cdots q_{r-1}!} \right\}, \quad \text{if } k \leq w < r^k,
\]
\[\text{if } w \geq r^k.
\]
For the special case \(r = 2,\)
\[n_k^w := \left\{ \sum_{\ell=0}^{w-k} \binom{k}{\ell-k} \right\}, \quad \text{if } k \leq w < 2k,
\]
\[\text{if } w \geq 2k.
\]

### 2.3. A non-recursive formula

Let us first consider the cardinality of \(S_k\) for large windows with \(w \geq r^k\). For each fixed \(\vee_1, \vee_{i+1}\) has \(r\) many choices. So, there are total
\[n_1^w + n_2^w + \cdots + n_{f-1}^w = r + r^2 + \cdots + r^{f-1}
\]
many check mark states. In our next example, we will enumerate all \((\vee_1, \ldots, \vee_k)\)'s following a certain rule and check which one is qualified when \(w\) is limited.

**Example 5.** \((f, r) = (4, 3)\). If \(w = 5\), then (see Fig. 1)
\[S_1: (\vee_1) = (1) \Rightarrow (2) \Rightarrow (3) \Rightarrow n_1^5 = 3,
\]
\[S_2: (\vee_1, \vee_2) = (1, 2) \Rightarrow (2, 3) \Rightarrow (3, 4) \Rightarrow (1, 3) \Rightarrow (2, 4) \Rightarrow (3, 5) \Rightarrow (1, 4) \Rightarrow (2, 5) \Rightarrow (3, 6) \times n_2^5 = 3 + 3 + 2,
\]
\[S_3: (\vee_1, \vee_2, \vee_3) = (1, 2, 3) \Rightarrow (2, 3, 4) \Rightarrow (3, 4, 5) \Rightarrow (1, 2, 4) \Rightarrow (2, 3, 5) \Rightarrow (3, 4, 6) \times n_3^5 = 3 + 2 + 1 + 2 + 0 + 0
\]
\[\times (1, 2, 5) \times (2, 3, 6) \times (3, 4, 7) \times + 1 + 0 + 0,
\]
where \(\times\) stands for “not a valid check mark state,” and “\(\Rightarrow\)” (called a plus-one-shift) means “adding one to every entry.” It is easier to enumerate all check mark states by first fixing \(\vee_1\) as 1,
then shift \( \rightarrow \) them \( r-1 \) many times. Very likely, there is a regularity behind this enumerating rule for windows with \( w < rk \).

If ordering all elements of \( S_k \) in lexicographic precedence ("\( \prec \)") using backtracking scheme, all of the transitional order pairs form a totally ordered set in a balanced \( r \)-tree structure, and elements of \( S_k \) are actually leaves of the tree.

**Definition 6.** The \( r \)-tree for \( S_k \) with window size \( w \) is defined as follows: the root is \( \emptyset \) (referring to dummy \( \checkmark_0 \)), and a node \( P \) is an increasing finite sequence

\[
P = (z_1, z_1 + z_2, \ldots, z_1 + \cdots + z_m)
\]

for any choice of \( 1 \leq z_i \leq r \) for \( i = 1, 2, \ldots, m, m \leq k \) as in (1), and let \( \ell(P) := m \) be the length of \( P \). It is equivalent to regard a node as a path from the root to the node itself, or a node as a subtree in the usual sense. There are several terms defined as follows:

1. It is said that node \( P \) precedes node \( Q \) (denote \( P \prec Q \)) if there exists \( j \) such that \( P(i) = Q(i) \) for \( i < j \) and \( P(j) < Q(j) \). Moreover, if \( d := \ell(Q) - \ell(P) > 0 \) and \( P(i) = Q(i) \) for \( i = 1, \ldots, \ell(P) \), then we say \( Q \) is a descendant of \( P \), denoted by \( P = Q^{-d} \), and the path from \( P \) to \( Q \) is uniquely determined by the partial sequence of \( Q \) from index \( \ell(P) \) to index \( \ell(Q) \), denoted by \( [P, Q] \).
2. Every non-leaf node \( P \) of length \( m \) has exactly \( r \) many length \( m+1 \) descendants \( P_1 < \cdots < P_r \) with \( P_i(m+1) = P(m) + i \) for \( i = 1, \ldots, r \), and these nodes are called siblings.
3. \( P \) is a shift of \( Q \) if there exists \( d \in \mathbb{Z} \) such that for any descendant \( P' \) of \( P \), \([P, P']\) with entries plus \( d \) equals \([Q, Q']\) for some descendant \( Q' \) of \( Q \), and conversely, for any descendant \( Q' \) of \( Q \), \([Q, Q']\) with entries minus \( d \) equals \([P, P']\) for some descendant \( P' \) of \( P \).
Proposition 7. In the $r$-tree for $S_k$, if two nodes are siblings, one is a shift of the other, in the sense of subtrees.

Proof. Let $P$ be a node of length $m$ and $Q$ be a sibling of $P$, i.e., $P(i) = Q(i)$ for $i = 1, 2, \ldots, m - 1$ and $P(m) + d = Q(m)$ for some $d \neq 0$. Then for any length $n$ descendant $P'$ of $P$,

$$[P, P'] = (P(m), P(m) + z_{m+1}, \ldots, P(m) + z_{m+1} + \cdots + z_n)$$

and for any length $n$ descendant $Q'$ of $Q$,

$$[Q, Q'] = (Q(m), Q(m) + \tau_{m+1}, \ldots, Q(m) + \tau_{m+1} + \cdots + \tau_n),$$

where $1 \leq z_i \leq r$ and $1 \leq \tau_i \leq r$ for $i = m + 1, \ldots, n$. As long as the choices of $(z_{m+1}, \ldots, z_n)$ and $(\tau_{m+1}, \ldots, \tau_n)$ are the same, entries of $[P, P'] + d$ are equal to entries of $[Q, Q']$, and $P'$ and $Q'$ are of the same order respectively (between 1 and $r^{k-m}$) in subtree $P$ and subtree $Q$. \hfill $\Box$

In Proposition 7, if $d > 0$, i.e., $P < Q$, we shall use “$Q = P + d$” to denote the shift: when regarding $P$ as a node, $P + d = (P(1), \ldots, P(m - 1), P(m) + d)$; when regarding $P$ as a subtree, let $P'$ be any length $n$ descendant of $P$, then $P' + d = (P(1), \ldots, P(m - 1), P(m) + d, P(m) + z_{m+1} + d, \ldots, P(m) + z_{m+1} + \cdots + z_n + d)$ for some $z_i$'s, i.e., node $P' + d$ is a descendant of subtree $P + d$. For convenience, we shall abuse our previous notation

“$P \Rightarrow Q$” or “$Q$ is a plus-one-shift of $P$” for “$Q = P + 1$”

so “$Q = P + d$” can be read as “$Q$ is $d$ many plus-one-shifts of $P$.”

Let $v(P)$ denote the number of valid leaves in subtree $P$. Clearly, every non-leaf $P$ has $r$ descendants $P_1 < \cdots < P_r$ with $P_i^{-1} = P$, and $v(P) = v(P_1) + \cdots + v(P_r)$. Eventually, we need to obtain $v(\emptyset)$, i.e., the number of valid leaves in the $r$-tree for $S_k$, which is precisely $n_k^w$, the cardinality of $S_k$.

Corollary 8. Starting with node $P_* := (1, 2, \ldots, k - 1)$, let $a_* := v(P_*)$, $i_0 := w - k + 1$, and $s := \min(i_0 - r, r - 1)$.

1. If $a_* = r$, then $i_0 > r$, and the number of valid leaves is still $r$ in the first $s$ many plus-one-shifts of $P_*$, i.e., $v(P_* + d) = r$ for $d = 0, 1, 2, \ldots, s$.
2. If $a_* < r$, then $i_0 = a_*$, and every plus-one-shift of $P_*$ results in one fewer valid leaf, until no valid leaf left, i.e., $v(P_* + d) = \max(a_* - d, 0)$ for $d = 0, 1, 2, \ldots, r - 1$. 

(4) A node is valid if every of its entry is no bigger than $w$, otherwise, invalid.

Some trivial facts from Definition 6:

- For $m < n \leq k$, every length $m$ node has $r^{n-m}$ many length $n$ descendants.
- If $P = Q^{-d}$, we do not write “$Q = P + d$” since the parent is unique, but the descendant is not, for every node.
- If $P$ is a shift of $Q$, then they have the same number of descendants.
- If a node is valid, so are all its precedent siblings; If a node is invalid, so are all its descendants and its following siblings.
Proof. Clearly, $a_\star$ is at most $r$. $P_\star$ has $r-1$ siblings and by Proposition 7, they are all considered as shifts of $P_\star$ in the following:

1. If $a_\star = r$, i.e., $(k-1)+1 < \cdots < (k-1)+r \leq w$. Since $w-(k-1+r)=i_0-r$ is the room between $(k-1)+r$ and $w$, the first $s := \min(i_0-r, r-1)$ many following siblings of $P_\star$ have the same number of valid leaves.

2. If $a_\star < r$, i.e., $(k-1)+1 < \cdots < (k-1)+a_\star \leq w < (k-1)+a_\star+1 < \cdots < (k-1)+r$, then $a_\star$ must be equal to $i_0 := w-k+1<r$. Hence, the first $i_0-1 (< r-1)$ many following siblings of $P_\star$ have $i_0-1, i_0-2, \ldots, 2, 1$ valid leaves, and the rest (the last $r-i_0$ many) siblings have no valid leaf. \hfill \square

Directly from Corollary 8, $v(P_{\star}^{-1})$ can be written explicitly as

$$v(P_{\star}^{-1}) = \begin{cases} \phantom{\overbrace{}} r \text{ terms} \\ \overbrace{a_{\star} + \cdots + a_{\star} + (a_{\star} - 1) + (a_{\star} - 2) + \cdots + (s + 1) + s}^{s \text{ many}} \text{ if } a_{\star} = r, \\ a_{\star} + (a_{\star} - 1) + \cdots + 2 + 1 + 0 + \cdots + 0, \text{ if } a_{\star} < r. \\ \overbrace{r-i_0 \text{ many}} \end{cases}$$ (7)

Define a new sequence $\{a_j\}_{j=-\infty}^{\infty}$

$$\cdots a_{-2} a_{-1} a_0 \underbrace{a_1 a_2 \cdots a_{r-1}}_{r} a_r \underbrace{a_{r+1}}_{r} a_{r+2} \cdots \\
\phantom{\cdots} \underbrace{0 0 0}_{r} \underbrace{1 2 \cdots r-1 r}_{r} r \phantom{\cdots}$$ (8)

Then $a_\star = a_{i_0}$ and (7) becomes

$$v(P_{\star}^{-1}) = a_{i_0} + a_{i_0-1} + \cdots + a_{i_0-r+1} = \sum_{i_1=0}^{r-1} a_{i_0-i_1}. \quad (9)$$

Let $P$ be a length $k-2$ node and its length $k-1$ descendants are $P_1 < \cdots < P_r$. If $P$ is not the last node of length $k-2$, then $\Rightarrow$ is applicable to it and

$$v(P+1) = v(P_1+1) + v(P_2+1) + \cdots + v(P_{r-1}+1) + v(P_r+1)$$
$$= v(P_2) + v(P_3) + \cdots + v(P_r) + v(P_{r+1})$$

$v(P_{r+1}) = \max(v(P_r) - 1, 0)$ by Corollary 8. Therefore inductively we have

Corollary 9. A node $P$ of length $m < k$ has $\mu := r^{k-m-1}$ many descendants of length $k-1$. Let $P_1 < \cdots < P_\mu$ be these $\mu$ descendants and $v_1 = v(P_1), \ldots, v_\mu = v(P_\mu)$, then

$$v(P) = (v_1 + v_2 + \cdots + v_r) + \cdots + (v_{\mu-r+1} + v_{\mu-r+2} + \cdots + v_{\mu}), \quad \text{and}$$

$$v(P+1) = (v_2 + \cdots + v_r + v'_r) + \cdots + (v_{\mu-r+2} + \cdots + v_{\mu} + v'_{\mu}),$$

where $v'_i := \max(v_i - 1, 0)$ for $i = r^1, r^2, \ldots, r^{k-m-1} = \mu$.

With (9), apply Corollary 9 repeatedly,
\[ v(P_{\ast}^{r-2}) = v(P_{\ast}^{r-1}) + v(P_{\ast}^{r-1} + 1) + \cdots + v(P_{\ast}^{r-1} + r - 1) \]
\[ = \sum_{i_1=0}^{r-1} a_{i_0-i_1} + \sum_{i_1=0}^{r-1} a_{i_0-i_1-1} + \cdots + \sum_{i_1=0}^{r-1} a_{i_0-i_1-(r-1)} \]
\[ = \sum_{i_2=0}^{r-1} \sum_{i_1=0}^{r-1} a_{i_0-i_1-i_2}, \]
\[ \vdots \]
\[ v(P_{\ast}^{r-(k-1)}) = v(P_{\ast}^{r-(k-2)}) + v(P_{\ast}^{r-(k-2)} + 1) + \cdots + v(P_{\ast}^{r-(k-2)} + r - 1) \]
\[ = \sum_{i_{k-1}=0}^{r-1} \cdots \sum_{i_1=0}^{r-1} a_{i_0-i_1-\cdots-i_{k-1}}. \]

Since \( P_{\ast} = (1, 2, \ldots, k - 1) \), \( P_{\ast}^{r-(k-1)} \) is the root of the \( r \)-tree for \( S_k \) and thus \( v(P_{\ast}^{r-(k-1)}) = n_k^w \). Therefore,

**Formula 10.** With \( i_0 := w - k + 1 \) and \( \{a_j\}_{j=-\infty}^{\infty} \) as in (8), for any \( w \), the general formula for the cardinality of \( S_k \) is
\[
n_k^w = \begin{cases} 
    a_{i_0}, & \text{if } k = 1, \\
    \sum_{i_{k-1}=0}^{r-1} \cdots \sum_{i_1=0}^{r-1} a_{i_0-i_1-\cdots-i_{k-1}}, & \text{if } k > 1.
\end{cases} \tag{10}
\]

**Example 11.** \( r = 3, k = 4, w = 7 \), by (10), \( i_0 = 4 > r \), and \( n_4^7 \) equals
\[
\begin{array}{c}
(3+3+2)+(3+2+1)+(2+1+0)+(3+2+1)+(2+1+0)+(1+0+0)+(2+1+0)+(1+0+0)+(0+0+0), \\
\Rightarrow (1,2) \Rightarrow (1,3) \Rightarrow (1,4) \Rightarrow (2,3) \Rightarrow (2,4) \Rightarrow (2,5) \Rightarrow (3,4) \Rightarrow (3,5) \Rightarrow (3,6) \Rightarrow \emptyset
\end{array}
\]
i.e., \( n_4^7 = a_4+a_3+a_2+ a_3+a_2+a_1 +3+2+1 + a_2+a_1+a_0 +2+1+0 + a_3+a_2+a_1 +3+2+1 + a_2+a_1+a_0 +2+1+0 + a_1+a_0+a_{-1} +1+0+0 + a_2+a_1+a_0 +2+1+0 + a_1+a_0+a_{-1} +1+0+0 + a_0+a_{-1}+a_{-2} +0+0+0 \).

2.4. A related problem

A related problem can be modeled as Problem 2:
Given $f, r, w \in \mathbb{N}$ with $w \geq \max(f, r)$, find the cardinality of $\bigcup_{k=1}^{\bar{f}-1} \bar{S}_k$ with

$$\bar{S}_k := \{(\sqrt{1}, \ldots, \sqrt{k}) \mid 1 \leq \sqrt{1} < \cdots < \sqrt{k} \leq w \text{ and } \sqrt{j} - \sqrt{j-1} \leq r \text{ for } j = 2, \ldots, k\}. \quad (11)$$

Note that this problem is actually our main problem (2) without the condition $\sqrt{1} - \sqrt{0} \leq r$. Since $\sqrt{1}$ can be placed at any position between 1 and $w - (k - 1)$ for each fixed $k$, it is clear that the general formula for the cardinality of $\bar{S}_k$ is

$$\bar{n}_k^w := \sum_{\ell=k-1}^{w-1} n_{k-1}^\ell, \quad (12)$$

where $n_{k-1}^\ell$ can be computed by (3), (5), (6), or (10).

3. An application

Let $\{X_i\}_{i=1}^n$ be a sequence of homogeneous two-state Markov dependent trials with outcomes success (or 1) and failure (or 0), initial probabilities

$$p = P(X_1 = 1), \quad q = P(X_1 = 0) \quad (13)$$

and transition probabilities

$$p_{ij} = P(X_k = j \mid X_{k-1} = i), \quad k \geq 2, \quad 0 \leq i, j \leq 1 \quad (14)$$

with $p_{11} + p_{10} = p_{01} + p_{00} = 1$. Let $WT(f, r, w)$ denote the waiting time until we first observe at least $f$ successes (or 1’s) or a run of $r$ failures (or 0’s) in a window of size $w$. We are interested in finding the distribution of the waiting time random variable $WT(f, r, w)$ by using the probability generating function (pgf) method. The pgf method is to establish a system of linear equations consisting of conditional pgf’s at different states of the experiment of $WT(f, r, w)$ and then solve the system for results related to the pgf of $WT(f, r, w)$.

Let $\phi(t)$ be the pgf of the distribution of the waiting time random variable $WT(f, r, w)$ which we solve for and let $\phi_{i_1, i_2, \ldots, i_k}(t)$ with $k \leq w$ and $0 \leq i_j \leq 1$, $j = 1, \ldots, k$, denote the pgf of the conditional distribution of the waiting time given that there was one success (if $i_j = 1$) or one failure (if $i_j = 0$) $j$ steps back for each $j = 1, \ldots, k$, and no other in the window that extends $w$ steps back. Then with the probabilities $p, q, p_{00}, p_{01}, p_{10},$ and $p_{11}$ given in (13) and (14), these pgf’s can be obtained according to the following rules: the main rules for generating the pgf’s are for $k < w$,

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

$$\phi_{i_1, i_2, \ldots, i_k}(t) = p_{11}t\phi_{1, i_1, i_2, \ldots, i_k}(t) + p_{10}t\phi_{0, i_1, i_2, \ldots, i_k}(t), \quad \text{if } i_1 = 1,$$

$$\phi_{i_1, i_2, \ldots, i_k}(t) = p_{01}t\phi_{1, i_1, i_2, \ldots, i_k}(t) + p_{00}t\phi_{0, i_1, i_2, \ldots, i_k}(t), \quad \text{if } i_1 = 0,$$ \quad (15)

and the reduction rules for eliminating redundant pgf’s are for $k \leq w$,

$$\phi_{i_1, i_2, \ldots, i_k}(t) = \phi_{i_1, i_2, \ldots, i_{k-1}}(t), \quad \text{if } n_1 < f, \ i_k = 1, \ w-k < f-n_1,$$

$$\phi_{i_1, i_2, \ldots, i_k}(t) = \phi_{i_1, i_2, \ldots, i_{k-1}}(t), \quad \text{if } n_1 \neq 0, \ i_k = 0,$$

$$\phi_{i_1, i_2, \ldots, i_k}(t) \equiv 1, \quad \text{if } n_1 = f \text{ or } n_0 = r,$$ \quad (16)

where $n_1$ is the number of 1’s, $n_0$ is the numbers of leading 0’s in the sequence $i_1, i_2, \ldots, i_k$, and $t$ acts as the parameter of the pgf’s.
The main rules and the reduction rules are obtained according to the total probability formula and the nature of the problem. For the first equation in (15), we formally write the pgf’s $\phi$, $\phi_1$, and $\phi_0$ as

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

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

$$
\phi_0(t) = \sum_{n=0}^{\infty} P((WT(f, r, w) | X_1 = 0) = n) t^n
$$

which converge for $0 \leq t \leq 1$. Due to the stopping rule of observing $f$ successes or a run of $r$ failures in a window of size $w$, by the total probability formula,

$$
P(WT(f, r, w) = n + 1) = p P(WT(f, r, w) = n + 1 | X_1 = 1) + q P(WT(f, r, w) = n + 1 | X_1 = 0)
$$

$$
= p P((WT(f, r, w) | X_1 = 1) = n) + q P((WT(f, r, w) | X_1 = 0) = n)
$$

which leads to the first pgf equation in (15) since the coefficients of $t^n$ in both sides of the equation are the same for all $n$. Other two equations in (15) can be explained similarly. Note that the stopping rules for the experiment are $f$ successes or a run of $r$ failures in a window of size $w$. Any information about successes that is more than $w$ steps back will not affect the outcome of the experiment and thus can be dropped, which explains the first equation in (16). Only the leading 0’s will contribute to the stopping rule of a run of $r$ failures and thus the trailing 0’s are always omitted, which explains the second equation in (16). The third equation in (16) is true because there are $f$ successes or a run of $r$ failures occurring in a window of size $w$. For instance, with $f = 3, r = 3,$ and $w = 5$, $\phi_{000}(t) = \phi_{0001}(t) = \phi_{00011}(t) \equiv 1$ since a run of 3 failures occurs in the window. Similarly, $\phi_{111}(t) = \phi_{1101}(t) = \phi_{1011}(t) = \phi_{11001}(t) = \phi_{110101}(t) = \phi_{10011}(t) \equiv 1$ since three successes occur in the window. And $\phi_{01001}(t) = \phi_{01}(t)$ and $\phi_{00101}(t) = \phi_{001}(t)$ since the success occurred 5 steps back will no longer contribute to the stopping rule of 3 successes in the window.

It is clear that the subscripts of the pgf’s for $WT(f, r, w)$ (excluding $\phi$ and those with only 0 indices) correspond to Problem 1 with parameters $(f, r - 1, w - 1)$ in Section 2. For the general case $WT(f, r, w)$, its pgf’s can be efficiently generated by using the tree structures in Section 2.3 with the main rules (15) and reduction rules (16) applied. Let

$$
\Phi(t) = (\phi(t), \phi_0(t), \ldots)^T
$$

be the column vector of the pgf’s. The dimension $N$ of $\Phi(t)$ can be easily determined by adding $r$ to the cardinality of Problem 1 with parameters $(f, r - 1, w - 1)$. Then the system of the pgf’s can be written in a matrix form

$$
\Phi(t) = tA\Phi(t) + tb,
$$

where $A$ is an $N \times N$ matrix and $b$ is a $N$-dimensional vector with all nonzero entries from $p, q, p_{00}, p_{01}, p_{10}$ or $p_{11}$. By (17), the $k$th derivative of $\phi$ at 0 are

$$
\phi^{(k)}(0) = k! P(WT(f, r, w) = k), \quad k = 0, 1, 2, \ldots,
$$

(19)
Table 1

| Probabilities $P(WT(f, r, w) = k)$, expectations $E$, and standard deviations $\sigma$ |
|---|---|---|---|
| $(f, r, w)$ | $(7, 5, 10)$ | $(10, 5, 15)$ | $(17, 5, 20)$ | $(20, 5, 25)$ |
| $k = 10$ | 0.0318 | 0.3487 | 0.0000 | 0.0000 |
| $k = 20$ | 0.0000 | 0.0000 | 0.1594 | 0.1216 |
| $k = 30$ | 0.0000 | 0.0000 | 0.0040 | 0.0015 |
| $\sigma$ | 0.9154 | 1.0344 | 3.5339 | 1.9786 |
| CPU time | 0.0001 s | 0.01 s | 0.19 s | 8.06 s |

and the $k$th derivatives of (18) at 0 are $\Phi'(0) = b$ and $\Phi^{(k)}(0) = kA\Phi^{(k-1)}(0)$ for $k = 2, 3, \ldots$, which can be simply written as

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

By (19) and (20), we get $P(WT(f, r, w) = 0) = 0$ and

$$P(WT(f, r, w) = k) = \text{the 1st component of } A^{k-1}b, \quad k = 1, 2, \ldots,$$  

which determines the exact distribution of the waiting time random variable $WT(f, r, w)$ and provides a numerical method to calculate the distribution.

The calculation of column vector $Ab$ involves no more than $2N$ multiplications of real numbers since each row of $A$ has no more than two nonzero. Hence, the calculation of $P(WT(f, r, w) = k)$, i.e., the first component of $A^{k-1}b = A(A^{k-2}b) = \cdots$, involves no more than $2N(k - 1)$ multiplications, and this dictates the efficiency of our algorithm. According to the nature of the problem, it can be shown that the spectral radius $\rho(A)$ of $A$ is less than 1, and from (21), $P(WT(f, r, w) = k)$ approaches zero as fast as $\rho(A)^{k-1}$ while $k$ increases. $\rho(A) < 1$ also warrants the stability of calculating $A^n b$.

A computer program in C++ for the exact distribution of the waiting time random variable $WT(f, r, w)$, based on the method discussed in this section, has been successfully implemented. An extensive testing shows that our algorithm is very efficient and is capable of solving large scale problems. Table 1 lists some numerical results of $P(WT(f, r, w) = k)$, expectations $E$ and standard deviations $\sigma$ of $WT(f, r, w)$ with parameters $(f, r, w) = (7, 5, 10), (10, 5, 15), (17, 5, 20), (20, 5, 25)$ and $p = 0.9, p_{01} = 0.95, p_{11} = 0.9$. All computation using double precision for the results in the table 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 $1 - P(WT(f, r, w) \leq n) < 10^{-10}$ is satisfied for some $n > 1$, and numerical values in the table are truncated after four decimal places. The largest value of $n$ for the results in Table 1 is 127 for the case of $(f, r, w) = (17, 5, 20)$.

References