

WAITING TIME DISTRIBUTIONS OF COMPETING PATTERNS IN HIGHER-ORDER MARKOVIAN SEQUENCES

JOHN A. D. ASTON,* *Academia Sinica, Taiwan*

DONALD E. K. MARTIN,** *Howard University and U.S. Census Bureau*

Abstract

Competing patterns are compound patterns that compete to be the first to occur pattern-specific numbers of times. They represent a generalisation of the sooner waiting time problem and of start-up demonstration tests with both acceptance and rejection criteria. Through the use of finite Markov chain imbedding, the waiting time distribution of competing patterns in multistate trials that are Markovian of a general order is derived. Also obtained are probabilities that each particular competing pattern will be the first to occur its respective prescribed number of times, both in finite time and in the limit.

Keywords: Multistate trials; finite Markov chain imbedding; sooner waiting time distribution; start-up demonstration test; competing compound patterns

2000 Mathematics Subject Classification: Primary 60E05

Secondary 60J05

1. Introduction

In recent years there has been significant research on distributions associated with patterns. An important segment of this work deals with sooner waiting time distributions. Generally speaking, the sooner waiting time distribution refers to waiting times for the first occurrence of one of several simple patterns. Most of the early research on sooner waiting time distributions dealt with success or failure runs in binary trials; see, for example [4], [3], and [1]. In [10] and [9], run and frequency quotas in multistate Markov chains were studied. Probability generating functions of sooner and later waiting time problems for patterns in Markov chains were considered in [8], while in [5] waiting time distributions for compound patterns were computed. Since compound patterns are the union of simple patterns, their waiting time distribution corresponds to the sooner waiting time distribution for the collection of simple patterns. We mention these papers because they each represent special cases of competing patterns as discussed in this paper.

Also of relevance is work on start-up demonstration tests. In early research on start-up demonstration testing [7], [17], it was assumed that the testing of the power-generation equipment continued until k consecutive successes had occurred, irrespective of the number of failures that occurred prior to it. As this setup is clearly impractical, [2] studied start-up demonstration tests with independent start-ups, where the equipment is accepted or rejected based on whether or not k consecutive successes occur before d total failures. The analysis

Received 28 April 2005; revision received 17 July 2005.

* Postal address: Institute of Statistical Science, Academia Sinica, 128 Academia Road, Sec. 2, Taipei, 115, Taiwan, Republic of China. Email address: jaston@stat.sinica.edu.tw

** Postal address: Department of Mathematics, Howard University, Washington, DC 20059, USA.

Email address: donald.e.martin@census.gov

was extended to trials that form a Markov chain in [12]. This research has subsequently been extended to tests with stopping criteria based on consecutive successes and consecutive failures, total successes and consecutive failures, and total successes and total failures [16], [13]. The length of start-up demonstration tests with these different setups can be considered as a sooner waiting time, if we allow the number of occurrences of the patterns (in this case either runs or individual successes and failures) to vary.

In this paper the more general problem of competing patterns is considered. Competing patterns refer to groups of compound patterns that compete to occur pattern-specific numbers of times. The setting for this study is an underlying sequence of multistate trials with an arbitrary order of Markovian dependence.

The waiting time distribution for this more general problem is computed through finite Markov chain imbedding [6]. This provides a theoretically rigorous environment whilst also enabling the development of easy computational algorithms for practical implementation. In addition, the framework allows for easy derivation of the limiting probabilities of absorption in one of the various states corresponding to a particular pattern being the first to occur its prescribed number of times.

The paper is organised as follows. In the next section, more formal definitions of the various concepts that are of importance in the paper are given. Section 3 contains a description of the steps used to compute the waiting time distribution and limiting absorption probabilities. Section 4 contains numerical examples, and the final section is a summary.

2. Definitions and preliminaries

Let $X_{-m+1}, X_{-m+2}, \dots, X_0, X_1, X_2, \dots$ be a multistate m th-order Markovian sequence, with state space $S_X = \{b_1, \dots, b_s\}$, $s \geq 2$, for the individual observations. An m th-order Markovian sequence is one for which

$$\begin{aligned} P(X_n = x_n \mid X_{-m+1} = x_{-m+1}, \dots, X_0 = x_0, \dots, X_{n-1} = x_{n-1}) \\ = P(X_n = x_n \mid X_{n-m} = x_{n-m}, \dots, X_{n-1} = x_{n-1}). \end{aligned}$$

Associated with the sequence are the initial probabilities

$$\pi_0(x_{-m+1}, \dots, x_0) = P(X_{-m+1} = x_{-m+1}, \dots, X_0 = x_0),$$

and time-invariant transition probabilities

$$p(x_t \mid x_{t-m}, \dots, x_{t-1}) = P(X_t = x_t \mid X_{t-m} = x_{t-m}, \dots, X_{t-1} = x_{t-1}).$$

2.1. Notation for patterns

Definition 2.1. A *simple pattern* Λ_i refers to a specified sequence of k_i symbols $b_{i_1}, \dots, b_{i_{k_i}}$, where the symbols in the pattern are allowed to be repeated.

Definition 2.2. A *compound pattern* Λ is the union of simple patterns, i.e. $\Lambda = \bigcup_{i=1}^{\eta} \{\Lambda_i\}$, where η is fixed, the simple patterns Λ_i are of lengths k_i , $i = 1, \dots, \eta$, and $\Lambda_a \cup \Lambda_b$ denotes the occurrence of either pattern Λ_a or pattern Λ_b .

Definition 2.3. Let $\Lambda^{(1)}, \dots, \Lambda^{(c)}$, $c \geq 1$, be a system of c compound patterns, and let r_j denote the number of occurrences of compound pattern $\Lambda^{(j)}$ that leads to termination of the experiment. The patterns $\Lambda^{(1)}, \dots, \Lambda^{(c)}$ are called *competing patterns*. The assumption is made that no two competing compound patterns in the system are identical.

If $C_j(n)$, $j = 1, \dots, c$, denotes the event that, by time n , the compound pattern $\Lambda^{(j)}$ has occurred r_j times, then the probability of the occurrence of one of the events $C_j(n)$, the waiting time for the competing patterns, is given by

$$P\left(\bigcup_{j=1}^c C_j(n)\right), \quad n \in \mathbb{N}. \tag{2.1}$$

If $c = 1$ then (2.1) reduces to the waiting time for the r_1 th occurrence of the compound pattern $\Lambda^{(1)}$. If $r_i = 1$ for all i then (2.1) reduces to the special case of the waiting time to the first occurrence of the compound pattern $\bigcup_{j=1}^c \{\Lambda^{(j)}\}$. If, in addition, each of the competing patterns consists of just one simple pattern, then the waiting time distribution corresponds to the *sooner waiting time distribution* considered in the literature discussed in the introduction. Next it is shown how various types of start-up demonstration tests arise as special cases of competing patterns.

For the types of start-up demonstration test that are described next, we take $S_X = \{0, 1\}$, representing failure/success of the individual start-ups, and there are $c = 2$ competing simple patterns. Four different tests are derived by varying the simple patterns $\Lambda^{(1)}$ and $\Lambda^{(2)}$, and the number of occurrences r_1 and r_2 , in the following manner. For a test with rejection of the unit if d total failures occur before k consecutive successes and acceptance if the k consecutive successes occur first, take

$$\Lambda^{(1)} = \overbrace{1 \cdots 1}^k, \quad r_1 = 1, \quad \Lambda^{(2)} = 0, \quad r_2 = d.$$

For a test with acceptance based on k consecutive successes and rejection based on d consecutive failures, take

$$\Lambda^{(1)} = \overbrace{1 \cdots 1}^k, \quad \Lambda^{(2)} = \overbrace{0 \cdots 0}^d, \quad r_1 = r_2 = 1.$$

The other two tests are obtained by taking $\Lambda^{(1)} = 1$ with $r_1 = k$ (total successes) and either

$$\Lambda^{(2)} = \overbrace{0 \cdots 0}^d, \quad r_2 = 1 \quad (\text{consecutive failures}),$$

or

$$\Lambda^{(2)} = 0, \quad r_2 = d \quad (\text{total failures}).$$

Other start-up demonstration tests may be derived as well; for example, if two rejection criteria are used in addition to an acceptance criterion, there will be $c = 3$ competing patterns.

In the next two subsections, the methods used to count patterns (Subsection 2.2) and the concepts of ending and finishing blocks (Subsection 2.3) are discussed.

2.2. Methods of counting

Two distinct methods of counting patterns are used in the paper, though much of the implementation will not change dramatically between the two cases. The first method is that of *non-overlapping counting*. In this case, when a pattern occurs the counting re-starts from that point, and any partially completed pattern cannot be finished. Non-overlapping counting can be for the entire system of competing patterns, or it can be restricted to within compound patterns (where counting only re-starts for simple patterns within the same compound pattern as the one that has just occurred). The second case is that of *overlapping counting*, where partially completed patterns can be finished at any time, regardless of whether another pattern has been completed after the partially completed pattern starts but before it is completed.

TABLE 1.

Competing pattern	SWNO	WPNO	Overlapping
$\Lambda^{(1)}$	$\Lambda_a \cup \Lambda_d$	$\Lambda_a \cup \Lambda_d$	$\Lambda_a \cup \Lambda_b \cup \Lambda_d$
$\Lambda^{(2)}$	Λ_d	$\Lambda_c \cup \Lambda_d$	$\Lambda_c \cup \Lambda_d$

Example 2.1. Let $\Lambda_a = 11111$, $\Lambda_b = 1011$, and $\Lambda_c = 00$ and let the competing patterns under consideration be $\Lambda^{(1)} = \Lambda_a$ and $\Lambda^{(2)} = \Lambda_b \cup \Lambda_c$. For the realisation $R_1 = 011011111101100$, of length $n = 15$, the use of system-wide non-overlapping (SWNO) counting yields two occurrences of Λ_b and one of Λ_c (and thus three occurrences of $\Lambda^{(2)}$), but no occurrences of $\Lambda^{(1)}$. Though there are six consecutive 1s in R_1 , only four of them occur after the first completion of 1011 (which re-starts counting), and thus Λ_a does not occur. With within-pattern non-overlapping (WPNO) counting, in addition to the occurrences mentioned above, Λ_a occurs once, since now the first occurrence of Λ_b does not cause a re-starting of counting for the partially completed pattern Λ_a , since they are in different competing patterns. Finally, with overlapping counting, $\Lambda^{(1)}$ occurs twice (the two overlapping occurrences of Λ_a), in addition to the three occurrences of $\Lambda^{(2)}$.

Remark 2.1. If SWNO counting is used, then any simple pattern wholly containing another will be deemed impossible to occur. However, with WPNO counting, this will only be the case when the simple patterns are within the same compound pattern. This is because, with non-overlapping counting, the occurrence of the smaller pattern re-starts the counting process, and thus the larger one will never be observed. On the other hand, with overlapping counting, simple patterns will be allowed to contain any smaller simple pattern within the system. The case where a simple pattern is part of two different compound patterns will not be eliminated using either non-overlapping or overlapping counting.

Example 2.2. To illustrate these concepts, if the simple patterns $\Lambda_a = 10$, $\Lambda_b = 101$, $\Lambda_c = 1010$, and $\Lambda_d = 1111$ make up the two competing patterns $\Lambda^{(1)} = \Lambda_a \cup \Lambda_b \cup \Lambda_d$ and $\Lambda^{(2)} = \Lambda_c \cup \Lambda_d$, the counting restrictions will reduce the effective competing patterns under evaluation in the manner shown in Table 1.

2.3. Ending and finishing blocks

Definition 2.4. *Ending blocks* [5] of a simple pattern $\Lambda_i = \{b_{i_1}, \dots, b_{i_{k_i}}\}$ are subpatterns of the form $\{b_{i_1}, \dots, b_{i_q}\}$, where q can be any of the integers $1, \dots, k_i - 1$. The set of ending blocks of a compound pattern is the union of the sets of ending blocks of the simple patterns of which it is comprised, along with the symbol \emptyset to indicate that none of the other ending blocks are currently active, if necessary. The set of ending blocks of the system of competing patterns is the union of the ending blocks of the competing patterns themselves, along with the symbol \emptyset .

Definition 2.5. *Finishing blocks* [14] of the simple pattern $\Lambda_i = \{b_{i_1}, \dots, b_{i_{k_i}}\}$ are subpatterns of the form $\{b_{i_\zeta}, \dots, b_{i_{k_i}}\}$, where ζ can be any of the integers $1, \dots, k_i$ (and thus Λ_i is a finishing block of itself). The finishing blocks of a compound pattern or of all of the competing patterns in the system are formed by taking the union of the finishing blocks of their respective components.

Remark 2.2. Note that, whereas ending blocks always start at the beginning of a simple pattern but may end at any point before its last symbol, finishing blocks may start at any point but always end with the last symbol.

3. Computation of waiting time distributions for competing compound patterns

This section contains details on the generation of the Markov chain, $\{Y_t\}_{t=0}^n$, that was used to compute the waiting time distribution for competing patterns. The Markov chain is such that

$$P(C_j(n)) = P(Y_n \in A_j), \quad j = 1, \dots, c, \tag{3.1}$$

where each A_j is an absorbing state corresponding to the occurrence of the event $C_j(n)$. Also,

$$P\left(\bigcup_{j=1}^c C_j(n)\right) = P(Y_n \in A), \tag{3.2}$$

where $A = \bigcup_{j=1}^c A_j$. For notational purposes, the union includes simultaneous occurrences of patterns, which are not included in (3.1). Basic properties of Markov chains are then used to compute the desired probabilities. After determining the state space S_Y of $\{Y_t\}_{t=0}^n$ and describing its generation, the method of generating the associated transition probability matrix T_Y is given. Finally, formulae for computing the waiting time distribution and limiting absorption probabilities are given.

3.1. State space S_Y

In [14], where the distribution of the time until the r th occurrence of a compound pattern was computed, vector triplets were used to represent the transient states of S_Y . In that paper, the three vector components were an m -tuple of values (x_{t-m+1}, \dots, x_t) , an ending block, and the number of simple pattern occurrences up to the present time. In this paper, because there are c competing patterns rather than one, the transient states must carry additional information. This generalisation is described next.

The vector state representation will in general be of length $2c + 1$. As before, one of these vector components gives the last m observations from the underlying sequence $\{X_t\}$. This m -tuple is necessary for Markov chain formation because the sequence $\{X_t\}$ is m th-order Markovian by assumption. The other c component pairs give ending block information and the number of pattern occurrences to date for each of the c competing patterns. The information on the number of occurrences of pattern j gives progress towards the r_j occurrences needed for termination of the experiment. The ending block information shows the progress towards the completion of the simple patterns. Note that, under SWNO counting, although c ending blocks may be used, the relevant ending block is the same for all competing patterns in the system, and thus in this case only $c + 2$ vector components are needed.

To automate the generation of the transient states of S_Y , the m -tuple is allowed to take each of the s^m possible values and, for each j , the number of pattern occurrences are given values $0, 1, \dots, r_j - 1$. The various possible values of the ending blocks also need to be generated. This is discussed next.

3.1.1. *Determining ending block information for transient states.* With non-overlapping counting, either system wide or within pattern, the current ending block(s) will depend on the relevant finishing blocks. In the case of overlapping counting, ending block formation is more straightforward. The competing patterns $\Lambda^{(1)} = \Lambda_a$ and $\Lambda^{(2)} = \Lambda_b \cup \Lambda_c$, with $\Lambda_a = 11111$, $\Lambda_b = 1011$, and $\Lambda_c = 00$, are used to illuminate the discussion. The set of ending blocks for the complete system is

$$E = \{1, 11, 111, 1111, 10, 101, 0, \emptyset\},$$

with ending blocks for the individual competing patterns $E_1 = \{1, 11, 111, 1111, \emptyset\}$ and $E_2 = \{1, 10, 101, 0, \emptyset\}$. The sets of finishing blocks are $F_1 = \{1, 11, 111, 1111, 11111\}$ and $F_2 = \{1, 11, 011, 1011, 0, 00\}$, and the set of finishing blocks F for the complete system is $F = F_1 \cup F_2$.

With non-overlapping counting, simple patterns may be completed in the middle or at the end of an m -tuple (x_{t-m+1}, \dots, x_t) , re-starting the counting process and modifying the ending block associated with the m -tuple. To account for this, for each m -tuple, first the ending block obtained by examining it in its entirety is recorded. Then the location after each symbol is examined to determine if a pattern could end there, i.e. if the segment of the m -tuple up to that point is a finishing block. If so, the effective ending block is determined from the remainder of the m -tuple. As stated previously, with SWNO counting, there is only one ending block. With WPNO counting, each compound pattern will have an ending block whose determination is based on the finishing blocks of its simple patterns. The ending blocks for overlapping counting are obtained from the entire m -tuple and, hence, finishing blocks are not needed in this case.

As an example, assume that $m = 3$ and consider the competing patterns $\Lambda^{(1)}$ and $\Lambda^{(2)}$ and the triple 000. For SWNO counting, there will be one ending block, coming from E . For WPNO counting, there will be two ending blocks, one from E_1 and one from E_2 . Since there are no 0s in the simple pattern Λ_a , all ending blocks of $\Lambda^{(1)}$ must be \emptyset . In examining the complete triple, a pattern ends after the second 0, and with SWNO counting, counting re-starts and the 0 at the end is its only ending block. With WPNO counting, this 0 is the ending block associated with pattern $\Lambda^{(2)}$.

The location after the first symbol is now checked to determine if it is a finishing block; and indeed it is, for $0 \in F_2$. If a pattern ends after the first 0, the ending block determined from 00, the remainder of the triple, is \emptyset under both system-wide and within-pattern overlapping counting. The location after the second symbol of the triple is certainly a finishing block, as a pattern ends there. The situation is just as when we examined the entire sequence, and thus the ending blocks have already been included in our set of possible state vectors. Now, in checking the location after the third symbol of 000, we find that the pattern 00 ends there, but the ending block is again \emptyset , which has previously been included. If overlapping counting is used, only the entire triple need be checked. Here the ending block associated with $\Lambda^{(2)}$ will be 0, to indicate that one more 0 gives another overlapping pattern occurrence.

Consider now as an example another triple, 111. The ending blocks associated with the complete triple are 111 for $\Lambda^{(1)}$ and 1 for $\Lambda^{(2)}$, for both overlapping and WPNO counting (with 111 being the only ending block for SWNO counting). In checking for finishing blocks at the middle locations or end of the triple, it is observed that a pattern could end after its first, second, or third 1. If a pattern ends after the first 1, for both types of non-overlapping counting (in particular, for compound pattern $\Lambda^{(1)}$ if WPNO counting is used), then the effective ending block is 11; if a pattern ends after the second 1, then the effective ending block is 1; and if a pattern ends after the third 1, then the effective ending block is \emptyset . For $\Lambda^{(2)}$ and WPNO counting or overlapping counting, the effective ending block is always 1. Also added to the state space are ending blocks of lengths greater than m with their corresponding m -tuple. Thus, 1111 is added as an ending block for $\Lambda^{(1)}$ with WPNO counting or overlapping counting, and for the whole system with SWNO counting.

3.1.2. Absorbing states. For each competing pattern, an absorbing state is included in S_Y to indicate that a corresponding compound pattern has occurred the required number of times. As a result, absorption probabilities for each particular competing pattern may be computed. Absorbing states are also added to indicate that more than one of the events $C_j(n)$ have occurred

simultaneously. In general, if $c \geq 2$ then the number of possible combinations of indices for simultaneous absorption by two or more competing patterns is

$$\binom{c}{2} + \binom{c}{3} + \dots + \binom{c}{c} = 2^c - (1 + c).$$

Adding c , for the number of absorbing states corresponding to the individual competing patterns, we find that there are a maximum of $2^c - 1$ absorbing states in S_Y . Note that it may be impossible for certain of the compound patterns to cause simultaneous absorption.

3.1.3. Initialisation states. For times $t < m$, the chain $\{Y_t\}$ may be in states that were not covered by the methods given previously. These states are added to S_Y . The most obvious of these states is the state of Y_0 , where the m -tuple is (x_{-m+1}, \dots, x_0) , every competing pattern has occurred zero times, and all ending blocks are equal to \emptyset . From this starting point, for $t = 1, \dots, m - 1$, by determining the possible destinations from each of the initialisation states already in S_Y , new states are added. The added states are deleted after time $t = m$, as they will no longer be needed in the state matrix.

3.2. Determination of T_Y

With states set up as described above, $\{Y_t\}_{t=0}^n$ is a Markov chain. To obtain its transition probability matrix T_Y for transitions from Y_t to Y_{t+1} , the possible destinations from each state of S_Y are determined, and the appropriate transition probability $p(x_{t+1}|x_{t-m+1}, \dots, x_t)$ is assigned to the transition. Note that the matrix T_Y depends on the competing patterns, the order of dependence m , the method of counting patterns, and the transition probabilities for the underlying sequence $\{X_t\}$.

3.3. Computation of the waiting time distribution

The competing pattern experiment ends with compound pattern $\Lambda^{(j)}$ having occurred r_j times if and only if the Markov chain $\{Y_t\}$ is absorbed in the corresponding absorbing state. Thus, (3.1) and (3.2) hold. By (3.1) and basic properties of Markov chains, the waiting time for r_j occurrences of compound pattern $\Lambda^{(j)}$ is given by

$$P(C_j(n)) = \psi_0 T_Y^n W_j(A_j), \quad j = 1, \dots, c. \tag{3.3}$$

Here ψ_0 is a $1 \times \text{card}(S_Y)$ row vector holding the probabilities for the initial states of Y_0 . The probability for the state with m -tuple (x_{-m+1}, \dots, x_0) , zero(es) for the number of pattern occurrences, and \emptyset as the ending block(s) is given by $\pi_0(x_{-m+1}, \dots, x_0)$; all other initial probabilities are zero. Also, $W_j(A_j)$ is a $\text{card}(S_Y) \times 1$ column vector with a 1 in the position corresponding to the absorbing state for compound pattern $\Lambda^{(j)}$ and 0s elsewhere. By (3.2), probabilities for the waiting time until one of the competing patterns occurs its prescribed number of times may be computed using

$$P\left(\bigcup_{j=1}^c C_j(n)\right) = \psi_0 T_Y^n W(A), \tag{3.4}$$

where $W(A)$ is a $\text{card}(S_Y) \times 1$ column vector with 1s at its end, corresponding to the positions of the absorbing states of A , and with 0s elsewhere.

3.4. Limiting absorption probabilities

Standard results on absorption probabilities of Markov chains [15, pp. 102–116] give the limiting probability that any particular competing pattern is the first to occur its specified

number of times. The limiting probability is needed when there is a positive probability that the experiment has not ended by any given finite time. First, partition the transition probability matrix T_Y as follows:

$$T_Y = \begin{pmatrix} Q & R \\ 0 & I_{2^c-1} \end{pmatrix}.$$

Here Q is the $(\text{card}(S_Y) - 2^c + 1) \times (\text{card}(S_Y) - 2^c + 1)$ transition probability matrix for transitions strictly among transient states, R is a $(\text{card}(S_Y) - 2^c + 1) \times (2^c - 1)$ matrix of probabilities for transitions from transient to absorbing states, 0 is the $(2^c - 1) \times (\text{card}(S_Y) - 2^c + 1)$ zero matrix, and I_κ denotes the $\kappa \times \kappa$ identity matrix. Let $\tilde{\psi}_0$ be a $1 \times (\text{card}(S_Y) - 2^c + 1)$ subvector of the initial vector ψ_0 , such that $\tilde{\psi}_0$ holds the initial probabilities for the transient states. Also, let α_n be the $1 \times (2^c - 1)$ row vector holding the probabilities that Y_n lies in one of the individual absorbing states (including those states that indicate simultaneous pattern occurrence). The limiting absorption probabilities are then found to be

$$\lim_{n \rightarrow \infty} \alpha_n = \tilde{\psi}_0 (I_{\text{card}(S_Y) - 2^c + 1} - Q)^{-1} R.$$

4. Examples

A MATLAB[®] program (available from the authors upon request) was written to implement the calculations given in the last section. The program generates the states and the transition probability matrix, and then calculates the waiting time distributions and the limiting absorption probabilities. The following are two examples of output from this program.

Example 4.1. The first example is a start-up demonstration test with two rejection criteria. The equipment is accepted if ten consecutive successes occur before either of the rejection criteria: ten total failures or the occurrence of three consecutive failures twice. In this example the underlying sequence is Markovian of order $m = 3$, with state space $S_X = \{0, 1\}$. There are $c = 3$ competing patterns,

$$\Lambda^{(1)} = \overbrace{1 \cdots 1}^{10}, \quad \Lambda^{(2)} = 0, \quad \Lambda^{(3)} = 000,$$

with $r_1 = 1$, $r_2 = 10$, and $r_3 = 2$. The transition probabilities and initial distribution were chosen so that the underlying sequence is a stationary third-order Markovian sequence, using an alternative parametrisation as described in [11]. These transition and initial probabilities are listed in Table 2.

TABLE 2: Transition and initial probabilities used in Example 4.1, for the Markovian sequences shown.

Sequence	$p(1 \cdot, \cdot, \cdot)$	$\pi_0(\cdot, \cdot, \cdot)$
1, 1, 1	$\frac{24}{25}$	$\frac{1197}{2000}$
0, 1, 1	$\frac{19}{25}$	$\frac{63}{2000}$
1, 0, 1	$\frac{76}{125}$	$\frac{7}{400}$
0, 0, 1	$\frac{149}{375}$	$\frac{21}{400}$
1, 1, 0	$\frac{4}{25}$	$\frac{63}{2000}$
0, 1, 0	$\frac{89}{275}$	$\frac{77}{2000}$
1, 0, 0	$\frac{284}{375}$	$\frac{21}{400}$
0, 0, 0	$\frac{637}{8875}$	$\frac{71}{400}$

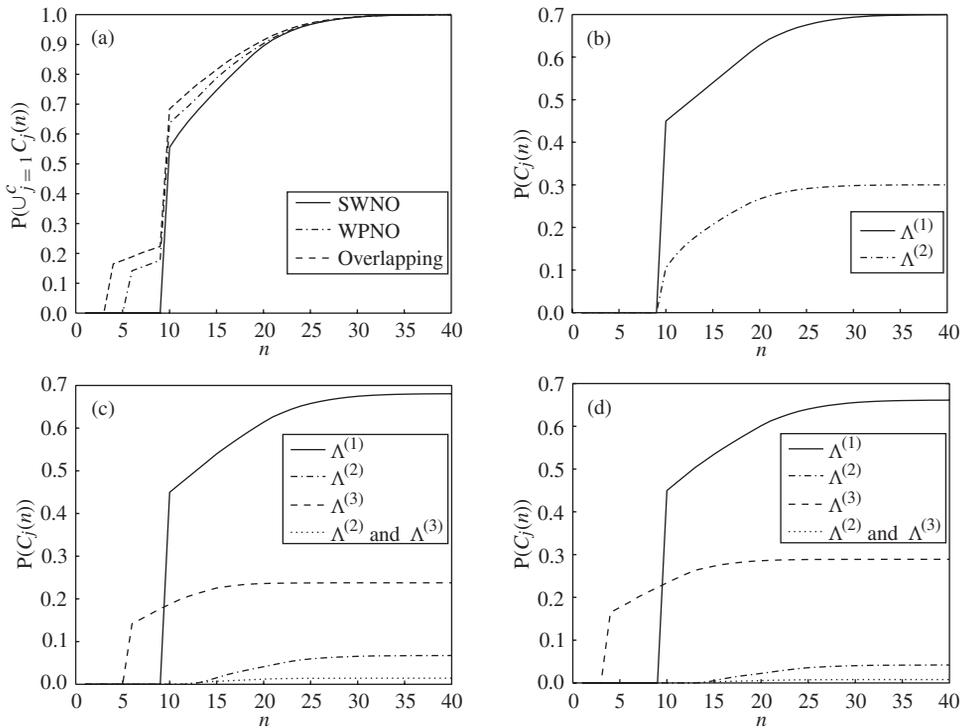


FIGURE 1: Waiting time probabilities of the competing patterns used in Example 4.1: (a) union of patterns, (b) SWNO, (c) WPNO, (d) overlapping. The fourth lines in plots (c) and (d) correspond to the absorbing state due to $\Lambda^{(2)}$ and $\Lambda^{(3)}$ occurring at the same time. This is an example of a start-up demonstration test requiring ten consecutive starts with two rejection criteria: ten total failures or two occurrences of three consecutive failures.

With SWNO counting, 178 states were required (including the initialisation states), 392 states were needed for WPNO counting, while, for overlapping counting, $\text{card}(S_Y) = 297$. The test can last no longer than $n = 100$ start-ups, but, as can be seen in the graphs of Figure 1, by time $n = 40$ the probability that one of the competing patterns has occurred its required number of times is very close to 1, regardless of the counting method. Also apparent from Figure 1 and Table 3 (which contains limiting absorption probabilities for the competing patterns) is that the probabilities of absorption into the absorbing states vary according to the counting method. In this example, the test could end with the simultaneous occurrence of ten total failures and two occurrences of three failures, but not with $\Lambda^{(1)}$ occurring simultaneously with any of the other patterns. Note that, since the test is guaranteed to end by time $n = 100$, the limiting absorption probabilities are the same as the absorption probabilities at that time. The computing time was approximately four seconds on a personal computer with 1 Gb of RAM and an Intel® Pentium® 4 processor.

Note that in typical formulations of start-up demonstration tests, such as in the references given in Section 1, the test has no ‘practice’ start-ups, and the initial distribution is based on the observations X_1, \dots, X_m instead of X_{-m+1}, \dots, X_0 . To handle this formulation, we would need to redefine the initial distribution $\pi_0(\cdot)$ in an appropriate manner. Probabilities for times $t = m + 1, \dots, n$ could then be computed by replacing T_Y^n by T_Y^{n-m} in (3.3) and (3.4), with

TABLE 3: Limiting absorption probabilities for Example 4.1, for the three different counting methods.

Pattern	Counting		
	SWNO	WPNO	Overlapping
$\Lambda^{(1)}$	0.7	0.681	0.661
$\Lambda^{(2)}$	0.3	0.0673	0.042
$\Lambda^{(3)}$		0.238	0.289
$\Lambda^{(2)}$ and $\Lambda^{(3)}$		0.014	0.008

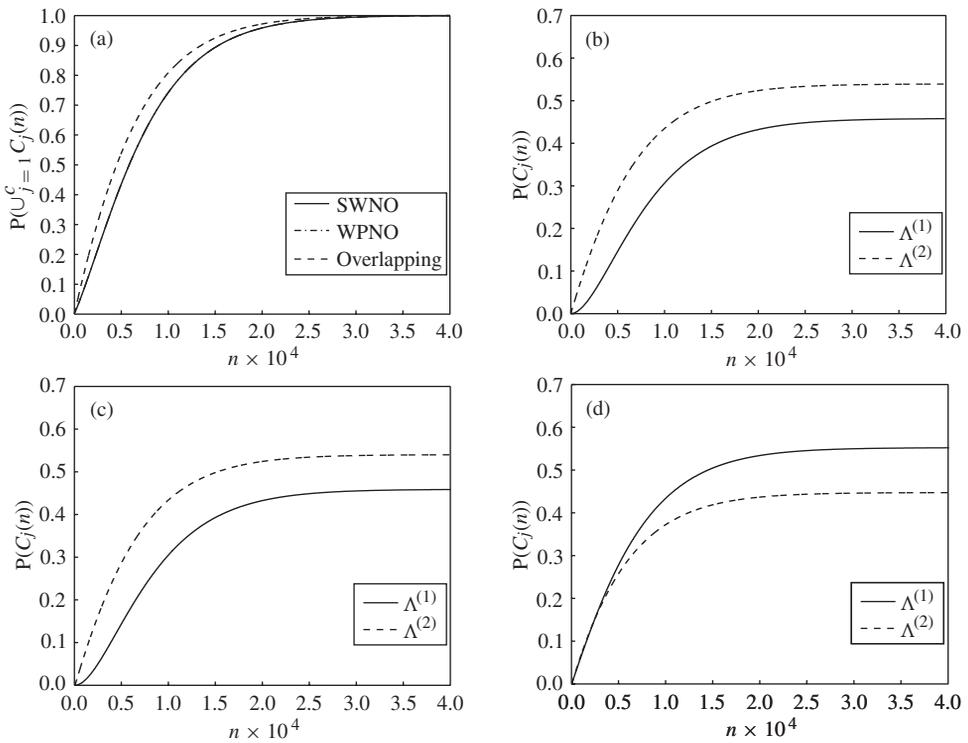


FIGURE 2: Waiting time probabilities of the patterns used in Example 4.2: (a) union of patterns, (b) SWNO, (c) WPNO, (d) overlapping. Altering the type of counting changes which competing pattern is more likely to cause absorption.

TABLE 4: Limiting absorption probabilities for Example 4.2, for the three different counting methods.

Sequence	Counting		
	SWNO	WPNO	Overlapping
$\Lambda^{(1)}$	0.459 37	0.459 36	0.551 07
$\Lambda^{(2)}$	0.540 63	0.540 64	0.448 93

the initial states of Y_m and the corresponding initial probabilities respectively determined by the m -tuple (x_1, \dots, x_m) and $\pi_0(x_1, \dots, x_m)$. Absorption probabilities at times $t < m$ could also be computed from $\pi_0(x_1, \dots, x_m)$. In the case of a stationary sequence, however, the probabilities derived from each formulation are the same.

Example 4.2. The second example illustrates that the method may be used in cases of larger state spaces and longer patterns. Here we take $\{X_t\}$ to be a fourth-order Markovian sequence with state space $S_X = \{A, C, G, T\}$. Randomly selected transition probabilities are used, meaning that uniform(0,1) variates are used, with the restriction that the probabilities sum to 1. The initial distribution for the sequence $\{X_t\}$ is the discrete uniform distribution over the $4^4 = 256$ possible initial states. For this example, approximately 750 states are needed for all three methods of counting. The competing patterns are

$$\begin{aligned} \Lambda^{(1)} &= ACGTACGT \cup AAAAAAAAA, & r_1 &= 2, \\ \Lambda^{(2)} &= TTTTTTTT, & r_2 &= 1. \end{aligned}$$

Figure 2 depicts the waiting time distributions for the two competing patterns, and the limiting absorption probabilities are listed in Table 4. One thing of interest to note is that the counting method does alter the competing pattern most likely to cause absorption. With SWNO and WPNO counting, $\Lambda^{(2)}$ is the sequence more likely to cause absorption, whilst with overlapping counting, the situation is reversed and $\Lambda^{(1)}$ is more likely to do so. While the probabilities under SWNO and WPNO counting appear to be remarkably similar, it can be seen from the absorption probabilities that they are not identical.

The computation time for the second example was approximately 600 seconds for $n = 40\,000$ trials (on the same hardware).

5. Summary

Competing patterns are compound patterns that ‘compete’ to be the first to occur specified numbers of times. Special cases of waiting times for the occurrence of competing patterns include the sooner waiting time and the time to the conclusion of a start-up demonstration test. If the number of competing patterns is $c = 1$, previous results [14] on the waiting time to the r_1 th occurrence of a compound pattern are obtained. Using finite Markov chain imbedding, waiting time distributions for competing patterns were computed. The algorithm used also allows the computation of the probability that the experiment ends due to the specified number of occurrences of specific patterns, both in finite time and in the limit.

References

- [1] AKI, S. AND HIRANO, K. (1999). Sooner and later waiting time problems for runs in Markov dependent bivariate trials. *Ann. Inst. Statist. Math.* **51**, 17–29.
- [2] BALAKRISHNAN, N. AND CHAN, P. S. (2000). Start-up demonstration tests with rejection of units upon observing d failures. *Ann. Inst. Statist. Math.* **52**, 184–196.
- [3] BALASUBRAMANIAN, K., VIVEROS, R. AND BALAKRISHNAN, N. (1993). Sooner and later waiting time problems for Markovian Bernoulli trials. *Statist. Prob. Lett.* **18**, 153–161.
- [4] EBNESHahrashoob, M. AND SOBEL, M. (1990). Sooner and later waiting time problems for Bernoulli trials: frequency and run quotas. *Statist. Prob. Lett.* **9**, 5–11.
- [5] FU, J. C. (1996). Distribution theory of runs and patterns associated with a sequence of multi-state trials. *Statistica Sinica* **6**, 957–974.
- [6] FU, J. C. AND KOUTRAS, M. V. (1994). Distribution theory of runs: a Markov chain approach. *J. Amer. Statist. Assoc.* **89**, 1050–1058.
- [7] HAHN, G. J. AND GAGE, J. B. (1983). Evaluation of a start-up demonstration test. *J. Quality Tech.* **15**, 103–105.

- [8] HAN, Q. AND HIRANO, K. (2003). Sooner and later waiting time problems for patterns in Markov dependent trials. *J. Appl. Prob.* **40**, 73–86.
- [9] KOLEV, N. AND MINKOVA, L. (1999). Quotas on runs of successes and failures in a multi-state Markov chain. *Commun. Statist. Theory Meth.* **28**, 2235–2248.
- [10] KOLEV, N. AND MINKOVA, L. (1999). Run and frequency quotas in a multi-state Markov chain. *Commun. Statist. Theory Meth.* **28**, 2223–2233.
- [11] MARTIN, D. E. K. (2000). On the distribution of the number of successes in fourth- or lower-order Markovian trials. *Comput. Operat. Res.* **27**, 93–109.
- [12] MARTIN, D. E. K. (2004). Markovian start-up demonstration tests with rejection of units upon observing d failures. *Europ. J. Operat. Res.* **155**, 474–486.
- [13] MARTIN, D. E. K. (2005). Markovian start-up demonstration tests under various scenarios. Submitted.
- [14] MARTIN, D. E. K. AND ASTON, J. A. D. (2005). Waiting time distribution of the r -th occurrence of a compound pattern in higher-order Markovian sequences. Tech. Rep. 2005-03, Institute of Statistical Science, Academia Sinica.
- [15] RESNICK, S. (1992). *Adventures in Stochastic Processes*. Birkhäuser, Boston, MA.
- [16] SMITH, M. L. AND GRIFFITH, W. S. (2003). The analysis and comparison of start-up demonstration tests. Tech. Rep. 391, Department of Statistics, University of Kentucky.
- [17] VIVEROS, R. AND BALAKRISHNAN, N. (1993). Statistical inference from start-up demonstration test data. *J. Quality Tech.* **22**, 119–130.