A Simple Procedure for Computing Performance Characteristics of Truncated Sequential Tests with Exponential Lifetimes

Peichuen Kao
School of Business Administration
Temple University
Philadelphia, PA 19122

Edward P. C. Kao and
Jack M. Mogg
Quantitative Management Science
University of Houston
Houston, TX 77004

This paper studies truncated sequential tests with exponential lifetimes. In the paper, we present a new procedure for computing the operating characteristic curve and the average time to terminate the test. The procedure differs from those proposed by Aroian, and by Woodall and Kurkjian. It directly extends to more complex boundaries. A comparative example is given at the end.

KEY WORDS
Truncated sequential tests
Exponential lifetimes
Operating characteristics
Average termination times

I. INTRODUCTION
Truncated sequential life tests involving the exponential distribution, \( f(t) = \lambda \exp(-\lambda t) \), have been treated extensively by Aroian and his coworkers (for example, see [1], [2], [3], [4], and additional references given in a recent paper [5]). In fact, these studies form the basis for MIL-STD-781C [7]. In such a reliability test, we test the hypothesis \( \lambda = \lambda_0 \) against the alternative \( \lambda = \lambda_1 \) under a specified producer's risk \( \alpha \) and consumer's risk \( \beta \). For a given set of values \( \lambda_0, \lambda_1, \alpha, \) and \( \beta, \) the acceptance line is given by \( t = sh_s + h_0 \) and the rejection line is given by \( t = sh_s - h_1, \) where \( t \) is the test time, and \( s \) represents the number of failures. Parameters \( h_0, h_1, \) and \( h_s \) are given in [2] and [4], and defined as follows: \( h_0 = -\ln(\alpha/1-\alpha)/(\lambda_1-\lambda_0), \) \( h_1 = J \ln(\lambda_1/\lambda_0)/(\lambda_1-\lambda_0), \) and \( h_s = \ln(\lambda_1/\lambda_0)/(\lambda_1-\lambda_0), \) where \( J \) represents the intercept of the rejection line at the \( s \)-axis on the \((s,t)\) plane (cf., Figure 1). The approach for finding \( J \) is presented in [1, pp. 8–17]. The sequential test is truncated after some pre-selected \( T_0 \) time units or \( s_0 \) failures have been observed. Thus if a decision has not been reached earlier, we accept \( \lambda = \lambda_0 \) if \( T_0 \) is observed before \( s_0 \) failures, otherwise we accept \( \lambda = \lambda_1 \). An example illustrating these boundaries is shown in Figure 1.

Two principal performance characteristics of a sequential life test are the operating characteristic (OC) curve and the average termination time (ATT) function. Procedures for generating the OC and ATT functions have been proposed by Aroian and his coworkers [2], [3], and [4]. The computations essentially require convolving the probabilities associated with each path leading to test termination. Woodall and Kurkjian [8] propose an alternative approach for obtaining the OC curve. In this paper, we propose a rather simple procedure for computing the OC and ATT functions. The procedure amounts to a direct application of the results obtained by Durbin [6] in the context of boundary-crossing problems for the Brownian motion and Poisson processes.

Received April 1977; revised July 1978
2. THE PROCEDURE

For simplicity, in Figure 1 we call the upper boundary \( A(t) \) and the lower boundary \( B(t) \). Let \( N(t) \) be the Poisson process denoting the number of failures which have occurred by time \( t \). We say \( N(t) \) upcrosses a boundary if the crossing is made from below and downcrosses a boundary if the crossing is made from the left. For the two sample paths shown in Figure 1, \( a_1 \) and \( a_2 \) correspond to points of upcrossing, and \( b_1 \) and \( b_2 \) correspond to points of downcrossing. (In a sequential life test upcrossing \( a_1 \) and downcrossing \( b_1 \) after test termination are not permitted. Their inclusions only facilitate the exposition.)

Define \( u_i, v_i \) by \( A(u_i) = i, i = p, p + 1, \ldots \), and \( B(v_i) = j, j = 0, 1, \ldots \), where \( p = [A(0)] + 1 \) and \([x]\) denotes the integral part of \( x \). It is important to note that downcrosses can occur only at \( t = u_i \) and \( v_i \), whereas upcrosses can occur at any point in time.

Let \( A_i, B_i \) denote the probability that \( N(t) \) downcrosses \( A_i(t) \), \( B_i(t) \) at \( t = u_i, v_i \), not having previously downcrossed either boundary. Define \( r_p = \max (j | v_j < u_k) \), and \( p(l) = e^{-\lambda l}/k! \). Conditioning on the first downcrossing, we can write the following two sets of recursive equations

\[
p(k | \lambda u_k) = A_k + \sum_{i=p}^{k-1} A_i p(k - i | \lambda(u_k - u_i))
+ \sum_{j=0}^{k-1} B_j p(k - j | \lambda(u_k - v_j))
k = p, p + 1, \ldots \tag{1}
\]

\[
p(k | \lambda v_k) = B_k + \sum_{i=0}^{u_k} A_i p(k - i | \lambda(v_k - u_i))
+ \sum_{j=0}^{k-1} B_j p(k - j | \lambda(v_k - v_j))
k = 0, 1, 2, \ldots \tag{2}
\]

(Here, we adopt the usual convention about empty summations, i.e., for any function \( f \), we define \( \sum_{a}^{b} f = 0 \) if \( a > b \).) In computation, we first note that \( B_0 = \exp(-\lambda u_0) \). Using (2), we find recursively \( B_0, B_1, \ldots, B_{p-1} \). Alternating between (1) and (2), we can then compute \((A_p, B_p), (A_{p+1}, B_{p+1}), \) and so on. For our purpose, we can terminate the prescribed computation at \( k = \max (s_p - 1, [B(T_p)]) \).

The above results refer to downcrossing only. To include the upcrossings, we first define \( p_{lt} \) as the probability that \( N(t) \) crosses either boundary and \( N(t) = k \), where \( B(t) < k < A(t) \). Let \( p' = [A(t)] \) and \( q' = [B(t)] \). Then using the Markov property of \( N(t) \), we can write

\[
p_{lt} = A_k + \sum_{i=p}^{k-1} A_i p(k - i | \lambda(t - u_i))
+ \sum_{j=0}^{k-1} B_j p(k - j | \lambda(t - v_j)) \quad \text{if } t > v_k, \tag{3}
\]

where the second sum is zero for \( 0 < t \leq v_k \). In the above calculation, we have included all the probabilities associated with the upcrossings. This is due to the fact that by restricting \( B(t) < k < A(t) \) each upcrossing will for certain be followed by a downcrossing. Therefore conditioning on all the possible
first downcrossings will take care of all eventualities. The probability that \( N(t) \) has not crossed either boundary by time \( t \) and \( N(t) = k \) has not crossed either boundary by time \( t \) and \( N(t) = k \) is thus given by \( p(k|\lambda t) - \rho_{kt} \), where \( B(t) < k < A(t) \). Let \( T \) be the time at which \( N(\cdot) \) crosses either boundary for the first time. Then it is clear that for \( \nu_0 < t \leq T_0 \)

\[
P(T > t) = \sum_{k=0}^{t} [p(k|\lambda t) - \rho_{kt}] .
\]

(4)

For \( 0 < t < \nu_0 \), we must add \( p(0|\lambda t) \) to the right-hand-side of (4) (note that \( \rho_{0t} = 0 \) if \( 0 < t \leq \nu_0 \)). The average termination time can thence be found from

\[
\text{ATT}(\lambda) = E(T) = \int_{0}^{T_0} P(T > t) \, dt .
\]

(5)

To compute the OC curve, we note that

\[
PA(\lambda) = \text{Prob} \{ \text{accepting the material} | \lambda \} = \sum_{j=0}^{l} B_j + \text{ATT}(\lambda) = \int_{0}^{\nu_0} P(T > t) \, dt
\]

(6)

where \( l = [B(T_0)] \).

3. AN EXAMPLE

Consider the numerical example given in [2, pp. 18–20]. There they specify that \( \lambda_0 = 1/167, \lambda_1 = 4.5/167 \) (both in failures per hour), \( \alpha = 0.1, \) and \( \beta = 0.2. \) These requirements call for acceptance line \( t = 71.76658 + 71.76658 \), rejection line \( t = 71.76658 - 71.09858, s_0 = 3, \) and \( T_0 = 215.30. \) The test termination boundaries are depicted in Figure 2. They also give \( \nu_0 = 71.77, \nu_1 = 143.53, \nu_2 = 215.30, u_0 = 0.67, u_2 = 72.43, \) and \( \nu_2 = 144.20. \)

We first illustrate the use of the aforementioned procedure in finding the OC and ATT for the case when \( \lambda = 1/167. \) We start with \( B_0 = \exp(-71.77/167) = 0.6507 \) and note that \( p = 1. \) For \( k = 1, \)

\[
p(1|0.0040) = A_1, \text{ yields } A_1 = 0.0040, \text{ and } p(1|0.8595) = B_1 + A_1p(0|0.8554) + B_1p(1|0.4297) \text{ yields } B_1 = 0.1803.
\]

For \( k = 2, \) we have \( \nu_0 = \max\{|v_0 < u_0 = 72.43\} = 0, \text{ } p(2|0.4337) = A_2 + A_2p(1|0.4297) + B_2p(2|0.0040) \text{ yields } A_2 = 0.0599, \text{ and } p(2|1.2892) = B_2 + A_2p(1|1.2852) + A_2p(0|0.8555) + B_2p(2|0.8595) + B_2p(1|0.4298) \text{ yields } B_2 = 0.0499. \) For \( k = 3, \) we have \( \nu_0 = r_0 = 0, \text{ and } p(3|0.8635) = A_3 + A_3p(2|0.8595) + A_3p(1|0.4298) + B_3p(3|0.4337) + B_3p(2|0.0040) \text{ yields } A_3 = 0.0221. \)

The computation involving equations (3), (4), and (5) on a computer requires the digitization of the test time \( t. \) Here we choose a one-hour interval as an approximation. The results of the above calculations yield \( \text{ATT}(\lambda = 1/167) = 92.87. \) For the OC curve, we note that \( P(T > 215) = 0, \) and consequently \( PA(\lambda = 1/167) = \sum_{j=0}^{l} B_j = 0.8809. \) For comparison, we now apply the methods proposed by Aroian, and Woodall and Kurkjian to the same problem.

The Aroian Method [2]

In Figure 2, we see that \( u_0 - v_0 = u_0 - u_0 = c \) and \( v_0 = u_0 - u_0 = u_2 - u_0 = \lambda. \) Let \( g(t,s) \) denote the probability of ever reaching point \((t,s)\) from the origin \((0,0)\) in a sequential test. Straightforward probabilistic arguments yield (the darkened line with arrows in the figure indicates the sequence of computations):

\[
g(0,0) = 1
\]

\[
g(u_0,0) = g(0,0)p(0|\lambda u_0) = 0.996
\]

\[
g(v_0,0) = g(v_0,0)p(0|\lambda X) = 0.6507
\]

\[
g(u_0,1) = g(u_0,0)p(1|\lambda X) = 0.2770
\]

\[
g(u_1,1) = g(u_0,1)p(1|\lambda c) = 0.2759
\]

and similarly, \( g(v_0,1) = 0.1803, g(v_1,2) = 0.0767, g(u_0,2) = 0.0764. \) For the OC curve,

\[
g(u_0,3) = 0.0499.
\]

For the OC curve,

\[
\text{denotes points in } S
\]

\[
\text{denotes points in } R
\]

**FIGURE 2. Test termination boundaries for the example.**
we have \( P_A(\lambda = 1/167) = \sum_{i=0}^\infty g(t_i, i) = 0.8809 \).

To compute the ATT, we use the six rectangular blocks shown in Figure 2. For each block, we use the lower left corner point to index it. The set of the six such corner points is denoted by \( S = \{(0,0),(u_0,0),(u_1,1),(u_2,1),(u_3,2),(u_4,2)\} \). For simplicity, we define \( P(k|\lambda) = \sum_{i=0}^\infty p(i|\lambda) \) and \( R = \{(u_0,0),(u_1,1),(u_2,2)\} \). Let \( T_{k,s} \) denote the expected time at which testing terminates in Block \((k,s)\). Exploiting well-known results on order statistics yields

\[ T_{(0,0)} = \sum_{k=1}^\infty [u_k/(k + 1)]p(k|\lambda u_k) = (1/\lambda)P(2|\lambda u_k) = 0.0013 \]

\[ T_{(1,0)} = \sum_{k=1}^\infty [u_k + (2X/(k + 1))]p(k|\lambda X) = u_k P(2|\lambda X) + (2/\lambda)P(3|\lambda X) = 3.1784 \]

\[ T_{(0,1)} = \sum_{k=1}^\infty [u_k + (c/(k + 1))]p(k|\lambda c) = u_k P(1|\lambda c) + (1/\lambda)P(2|\lambda c) = 0.2887 \]

\[ T_{(1,1)} = \sum_{k=1}^\infty [u_k + (2X/(k + 1))]p(k|\lambda X) = u_k P(2|\lambda X) + (2/\lambda)P(3|\lambda X) = 8.1002, \]

and similarly, \( T_{(0,2)} = 0.5760, T_{(1,2)} = 61.9559 \). The ATT at \( \lambda = 1/167 \) is then given by

\[ \sum_{(t,i)\in S} g(t)T_{(t,i)} + \sum_{(t,i)\in R} u_i g(t) = 93.4867. \]

In this example, we note that if two failures occur at any time, the point will fall in the reject region (this enables us to compute the exact ATT). When the distance between the rejection line and acceptance line is large, the amount of computational effort will clearly multiply due to the increase in the number of elements in \( S \).

The Woodall and Kurkjian Method [8]

For this particular example, Equation (12) given in [8, p. 1407] reduces to \( P_A(\lambda = 1/167) = \sum_{i=0}^2 S_i \exp(-\lambda u_i) \). Using Equations (9a) and (9c) given in [8, p. 1406], the \( S_i \) are found recursively as follows: \( S_0 = 1, S_i = \lambda(u_0, u_i)S_0 - 0.4257, \) and \( S_2 = \sum_{i=1}^2 S_i \). Consequently, we find that \( P_A(\lambda = 1/167) = 0.8809 \). As mentioned earlier, this procedure does not extend to the solution of the ATT.

4. CONCLUSIONS

Truncated sequential life tests are of significant importance in large-scale real world applications. The procedure described in this paper for computing the OC and ATT functions requires considerably less computational effort than the original Aroian method. Like the other two procedures, the present method produces exact OC curves. The degree of accuracy in the ATT functions so constructed depends on the extent of discretization (a problem shared by the Aroian method when the boundaries do not satisfy \( 1 < h_s + h_r < 2 \)).

REFERENCES