A MCMC-type simple probabilistic approach for determining optimal progressive censoring schemes

Ritwik Bhattacharyaa and Narayanaswamy Balakrishnanb

aDepartment of Industrial Engineering, School of Engineering and Sciences, Instituto Tecnologico y de Estudios Superiores de Monterrey, Monterrey, México; bDepartment of Mathematics and Statistics, McMaster University, Hamilton, Canada

ABSTRACT
We present here a simple probabilistic approach for determining an optimal progressive censoring scheme by defining a probability structure on the set of feasible solutions. Given an initial solution, the new updated solution is computed within the probabilistic structure. This approach will be especially useful when the cardinality of the set of feasible solutions is large. The validation of the proposed approach is demonstrated by comparing the optimal scheme so determined with the one obtained by exhaustive numerical search.

ARTICLE HISTORY
Received 11 May 2021
Accepted 19 March 2022

KEYWORDS
Life-testing experiment; Near-optimal solution; Optimal progressive censoring; Quantile; Variance measure; Cost criterion

1. Introduction
Let us first consider the Type-II progressive censoring scheme introduced by Cohen (1963, 1966). Suppose \( n \) test units are placed on a life-testing experiment. A pre-fixed integer \( m \), representing the number of failures to be observed, is chosen before the experiment starts. At the time of first failure, denoted by \( X_1 : m : n \), \( R_1 \) units are randomly removed from the remaining \( n - 1 \) live units. At the time of second failure, denoted by \( X_2 : m : n \), \( R_2 \) units are randomly removed from the remaining \( n - R_1 - 2 \) live units, and so on. Finally, at the time of \( m \)th failure, denoted by \( X_m : m : n \), all the remaining units \( n - R_1 - \cdots - R_{m-1} - m \) (say, \( R_m \)) are removed from the test, resulting in termination of the experiment. Therefore, the integers \( R_1, \ldots, R_m \) satisfy the identity

\[
R_1 + \cdots + R_m = n - m.
\]

The quantities \( X_i : m : n, i = 1, 2, \ldots, m, \) are called progressively Type-II censored ordered statistics; see Balakrishnan and Cramer (2014). A Type-II progressive censoring scheme is characterized by the parameters \( n, m \) and \( (R_1, \ldots, R_m) \). Note that, for given \( n \) and \( m \), there are many choices of \( R_i \)'s, and let us denote the set of all those choices by

\[
CS(n, m) = \left\{ \mathcal{R} = (R_1, \ldots, R_m) \in \mathbb{N}^m \mid \sum_{i=1}^{m} R_i = n - m, \mathcal{N} = \{0, 1, \ldots, n - m\} \right\}.
\]

The cardinality of the set \( CS(n, m) \) is \( \binom{n-1}{m-1} \) which becomes quite large even for moderate values of \( n \) and \( m \). For instance, when \( (n, m) = (30, 10) \), the cardinality is in fact 300,45,015. Therefore, our interest is then in finding the optimal Type-II progressive censoring scheme under
some suitable optimality criterion function $\psi : \text{CS}(n, m) \to [0, \infty)$. The problem of determining optimal progressive censoring scheme has been discussed considerably in the literature; see, Balakrishnan and Cramer (2014) for relevant details.

Balakrishnan and Aggarwala (2000) were the first to discuss optimal progressive censoring scheme by taking $\psi$ to be the variance of the best linear unbiased estimators (BLUEs) of the model parameters and computed optimal scheme by minimizing it under normal and extreme value distributions. Due to the computational complexity, they presented the optimal schemes only up to $n = 50$ and $m = 3$ (see Balakrishnan and Aggarwala 2000, pages 197-204). Using the same optimality criterion, Burkschat, Cramer, and Kamps (2006, 2007) computed optimal censoring schemes for generalized Pareto distribution. Some more choices of $\psi$, such as total time on test, expected test duration and variance of the test time were considered by Burkschat (2008). Interestingly, in all these articles, optimal schemes mostly emerged as one-step progressive censoring, that is, censoring occurs only at one failure time (mathematically, exactly one $R_i$ is positive). Balakrishnan et al. (2008) and Ng, Chan, and Balakrishnan (2004) also introduced three choices of $\psi$ based on the asymptotic variance-covariance matrix and Fisher information, and computed optimal censoring schemes under Weibull lifetime model. Pradhan and Kundu (2009, 2013) proposed criterion based on asymptotic variance of estimated $p$th quantile of the underlying lifetime distribution and obtained the optimal censoring schemes for generalized exponential and Birnbaum-Saunders distributions. Dahmen, Burkschat, and Cramer (2012) derived a computational formula for the Fisher information matrix and, thence, studied A- and D-optimal criteria for Weibull, Lomax, normal and extreme value distributions. Bhattacharya, Pradhan, and Dewanj (2016) introduced a meta-heuristic algorithm, called variable neighborhood search (VNS), for the determination of optimal censoring schemes. They showed that the VNS algorithm performs sufficiently well for moderate to large values of $n$ and $m$. Recently, Bhattacharya (2020) introduced the notion of compound optimal design strategy under progressive censoring in a multi-criteria setup.

In this work, we propose a method based on a probabilistic structure through which optimal censoring schemes can be computed. The proposed algorithm works in principle similar to the Markov Chain Monte Carlo (MCMC) technique; but, it is not a MCMC method as there is no target distribution. The algorithm is described in the subsequent sections along with some numerical results. For illustrative purpose, we consider a variance measure and a cost measure as optimality criterion. In fact, any optimality criteria can be utilized in the proposed method.

The rest of the paper is organized as follows. The main algorithm based on the probability structure is presented in Sec. 2. In Sec. 3, the algorithmic steps under three distributions, namely, multinomial, uniform and multivariate hypergeometric distributions, are discussed. Sec. 4 presents some numerical results for the proposed approach and some comparative results. A real-life data set is analyzed in Sec. 5 to illustrate the proposed algorithm, and finally, some concluding remarks are made in Sec. 6.

2. Algorithm based on probability structure

Suppose the members of the set $\text{CS}(n, m)$ can be generated through a known probability distribution on $\mathcal{R}$, say, $\pi_\mathcal{R}(\cdot)$. This is analogous to the proposal density in MCMC technique. So, in analogy to target density in MCMC, we shall construct a monotone function $f_{\mathcal{R} \mid \psi(\mathcal{R})}(\cdot)$ of criterion $\psi(\mathcal{R})$. Depending on the minimization (or maximization) of the criterion function $\psi(\mathcal{R})$, the choice of $f_{\mathcal{R} \mid \psi(\mathcal{R})}(\cdot)$ will be monotone decreasing (or increasing) function. Then, the algorithm will simulate censoring schemes $\mathcal{R}$ from $\pi_\mathcal{R}(\cdot)$ and the optimal scheme will be selected based on the following probability rule. Let $\mathcal{R}_{\text{old}}$ be a candidate censoring scheme simulated from $\pi_\mathcal{R}(\cdot)$. A new candidate censoring scheme $\mathcal{R}_{\text{new}}$, obtained from the old candidate $\mathcal{R}_{\text{old}}$, will then be accepted with probability
\[ x = \min \left\{ 1, \frac{f_{R|R_{\psi}(R)}(R_{\text{new}})}{f_{R|R_{\psi}(R)}(R_{\text{old}})} \times \pi_{R_{\psi}(R)}(R_{\text{old}}) \right\}. \]

Because of the monotonic structure of \( f_{R|R_{\psi}(R)}(\cdot) \), the algorithm will always try to accept better candidate censoring schemes with probability \( x \). Consequently, starting with an initial simulated candidate \( R_{\text{old}} \) from \( \pi_{R}(\cdot) \), the algorithm will tend to reach optimal censoring scheme after a number of simulation runs. It is important to note that this is not a MCMC because \( f_{R|R_{\psi}(R)}(\cdot) \) is no longer a probability structure. In particular, we choose

\[ f_{R|R_{\psi}(R)}(\psi(R)|R) = e^{-\psi(R)} \]

for computational implementation purpose. Also, for the choice of \( \pi_{R}(\cdot) \), we choose multinomial, uniform and multivariate hypergeometric distributions.

Regarding the choices of optimality criterion \( \psi(R) \), several options have been discussed in the literature, and a list of these options can be found in Bhattacharya (2020). In this work, we consider two optimality criteria. The first one is based on a variance measure given by

\[ \psi(R) = \int_0^1 \text{Var}[\ln \hat{X}_s] \, ds, \]  

(1)

where \( \hat{X}_s \) represents the maximum likelihood estimate of \( X_s \), the \( s \)-th quantile of the underlying lifetime distribution. The aggregate variance of the log-quantile estimates over all quantile points gets summarized under this measure. Moreover, the criterion is scale-invariant meaning that the optimal scheme computed by optimizing the criterion does not change if the time scale gets changed. This is a desirable property in an optimizing criterion used in life-testing experiments (see Bhattacharya, Pradhan, and Dewanji 2016). The second criterion is based on a cost measure given by

\[ \psi(R) = C_o + C_f m + C_t E[X_{m:m,n}], \]

where \( C_t \) is the cost per unit of failure, \( C_f \) is the cost per unit of duration of testing, and \( C_o \) is an overall fixed cost that does not depend on the design parameters of the test. This criterion is also scale-invariant.

A Weibull lifetime model with distribution function of the form

\[ F(x; \beta, k) = 1 - e^{-(kx)^\beta}, \quad x > 0, \]

where \( \beta > 0 \) and \( k > 0 \) are the shape and scale parameters, respectively, is considered for illustrative purpose. The required expressions of \( \psi(R) \) under Weibull model are presented in the Appendix.

### 3. Different choices for \( \pi_{R}(\cdot) \)

As mentioned in the last section, \( \pi_{R}(\cdot) \) needs to be specified with some known probability distribution, and then the censoring schemes will be simulated from this distribution. A complicated probability structure may increase the computational complexity of the algorithm. Here, we choose three known probability distributions for \( \pi_{R}(\cdot) \) and then assess the corresponding numerical results.
3.1. Multinomial distribution

Let us assume a multinomial distribution with probability mass function

\[ \pi_{\mathcal{R}}(R_1, \ldots, R_m; p_1, \ldots, p_m) = \frac{(n - m)!}{R_1! \cdots R_m!} p_{1}^{R_{1}} \cdots p_{m}^{R_{m}}, \tag{2} \]

where \((n - m)\) and \(p_i > 0, i = 1, \ldots, m,\) are the parameters; see Johnson, Kotz, and Balakrishnan (1997) for various properties of this distribution. The simulation of a candidate from (2) is straightforward and any candidate generated from (2) automatically would satisfy the censoring constraint imposed by the progressive censoring, namely, \(R_1 + \cdots + R_m = n - m.\) Now, to run the algorithm, we adopt the following two steps:

Step 1 (Simulating \(\mathcal{R}_0\)) An initial candidate \(\mathcal{R}_0\) can be simulated in the following manner. Generate \(m\) random numbers \(u_i, i = 1, \ldots, m,\) from a uniform distribution \(U[0, 1],\) and set

\[ p_i = \frac{u_i}{\sum_{i=1}^{m} u_i}. \]

Then, \(\mathcal{R}_0\) can be directly simulated from the multinomial distribution with parameters \(n - m\) and \((p_1, p_2, \ldots, p_m);\)

Step 2 (Simulating \(\mathcal{R}_\ell\) from \(\mathcal{R}_{\ell-1}\) for \(\ell = 1, \ldots, \ell_{\text{max}}\)) Now, given \(\mathcal{R}_{\ell-1},\) a new \(\mathcal{R}_\ell\) can be randomly generated from \(\mathcal{R}_{\ell-1}\) in the following manner. We randomly replace full/partial components of \(\mathcal{R}_{\ell-1}\) with observations generated from a multinomial distribution. First, randomly choose \(m_1 (0 < m_1 \leq m)\) positions of \(\mathcal{R}_{\ell-1}.\) Let \((R'_{1}, \ldots, R'_{m_1})\) and \((p_{R'_{1}}, \ldots, p_{R'_{m_1}})\) denote the corresponding positions and probability values in \(\mathcal{R}_{\ell-1},\) respectively. Now, we recalculate the probabilities for these \(m_1\) positions as

\[ q_j = \frac{p_{R'_{j}}}{\sum_{j=1}^{m_1} p_{R'_{j}}}, \]

for \(j = 1, \ldots, m_1.\) We then simulate \(m_1\) values from the multinomial distribution with parameters \(\sum_{j=1}^{m_1} R'_j\) and \((q_1, \ldots, q_{m_1}),\) and concatenate with \(\mathcal{R}_{\ell-1}\).

Following Step 2, a new candidate will be accepted with probability

\[ \min \left\{ 1, \frac{f_{\mathcal{R}|\psi(\mathcal{R})} (\psi(\mathcal{R}_\ell)|\mathcal{R}_\ell) \times \pi_{\mathcal{R}} (\mathcal{R}_{\ell-1}; p_1, p_2, \ldots, p_m)}{f_{\mathcal{R}|\psi(\mathcal{R})} (\psi(\mathcal{R}_{\ell-1})|\mathcal{R}_{\ell-1}) \times \pi_{\mathcal{R}} (\mathcal{R}_{\ell}; p_1, p_2, \ldots, p_m)} \right\}. \]

At each time of acceptance of a candidate, a new updated optimal solution is obtained, and we would therefore expect to obtain the optimal solution after performing a sufficiently large number of simulation runs, say \(\ell_{\text{max}}.\)

3.2. Uniform distribution

Let us consider a discrete uniform distribution to construct \(\mathcal{R}_0, \mathcal{R}_\ell\) for \(\ell = 1, \ldots, \ell_{\text{max}}\) and \(\pi_{\mathcal{R}}(\cdot)\) in the following manner. First, generate a random number between 0 and \(n - m\) and set it as \(R_1.\) Then, generate another number between 0 and \(n - m - R_1\) and set it as \(R_2.\) Continue the process until all \(m\) positions are filled up. Thus, \(\mathcal{R}_0\) gets generated and

\[ \pi_{\mathcal{R}}(\mathcal{R}_0) = \frac{1}{(n - m)(n - m - R_1) \cdots (n - m - R_1 - \cdots - R_{m-1})}. \]

Next, we generate a new \(\mathcal{R}_\ell\) from \(\mathcal{R}_{\ell-1}\) as follows. Randomly choose a position in \(\mathcal{R}_{\ell-1}\) and let the corresponding value at that position be \(r_i.\) Then, first generate a number randomly between 0
and \(n - m - r_\ell\) and set it as \(R_1\) in \(R_\ell\); generate another number between 0 and \(n - m - R_1\) and set it as \(R_2\) in \(R_\ell\); and continue this process until all \(m\) positions are filled up. Thus, we can generate \(R_\ell\) and

\[
\pi_R(R_\ell) = \frac{1}{(n - m - r_\ell)(n - m - R_1) \cdots (n - m - R_1 - \cdots - R_{m-1})}.
\]

Thence, a new candidate will be accepted with probability

\[
\min \left\{ 1, \frac{f_{R|R_{\psi}(R)}(\psi(R_\ell)|R_{\ell-1}) \times \pi_R(R_{\ell-1})}{f_{R|R_{\psi}(R)}(\psi(R_{\ell-1})|R_{\ell-1}) \times \pi_R(R_{\ell-1})} \right\}.
\]

### 3.3. Multivariate hypergeometric distribution

Let us assume a multivariate hypergeometric distribution with probability mass function

\[
\pi_R(R_1, \ldots, R_m; M, (M_1, \ldots, M_m), R) = \frac{\left( \begin{array}{c} M_1 \\ R_1 \\ \vdots \\ M_m \\ R_m \\ \vdots \\ M_R \\ R_m \\ \vdots \\ M_m \\ R_m \end{array} \right)}{\left( \begin{array}{c} M \\ R \\ \vdots \\ M \end{array} \right)},
\]

where \(M, (M_1, \ldots, M_m)\) and \(R\) are the parameters with \(M_j = n - m\), for \(j = 1, \ldots, m\), \(M = \sum_{j=1}^{m} M_j = m(n - m)\) and \(R = \sum_{j=1}^{m} R_j = n - m\); see Johnson, Kotz, and Balakrishnan (1997) for properties of this distribution. One may readily note that any candidate simulated from (3) would automatically satisfy the constraint imposed by the progressive censoring. For given \(n\) and \(m\), simulation of one candidate as \(R_0\) is straightforward. Next, we use the following steps to generate \(R_\ell\) from \(R_{\ell-1}\) for \(\ell = 1, \ldots, \ell_{\text{max}}\). First, randomly choose \(m_1(0 < m_1 \leq m)\) positions of \(R_{\ell-1}\), and let \((R'_1, \ldots, R'_{m_1})\) denote the corresponding positions. Next, simulate a candidate of length \(m_1\) from a multivariate hypergeometric distribution with parameters \(M_j = \sum_{j=1}^{m_1} R'_j\), \(M = \sum_{j=1}^{m} M_j\) and \(R = \sum_{j=1}^{m_1} R'_j\), and concatenate with \(R_{\ell-1}\). Then, proceeding as before, a new candidate will be accepted with probability

\[
\min \left\{ 1, \frac{f_{R|R_{\psi}(R)}(\psi(R_\ell)|R_{\ell-1}) \times \pi_R(R_{\ell-1}; M, (M_1, \ldots, M_m), R)}{f_{R|R_{\psi}(R)}(\psi(R_{\ell-1})|R_{\ell-1}) \times \pi_R(R_{\ell-1}; M, (M_1, \ldots, M_m), R)} \right\}.
\]

### 4. Numerical results

We shall use Weibull distribution with parameters \((k, \beta) = (1, 0.5), (1, 1)\) and \((1, 2)\) as the underlying lifetime model. We evaluate the performance of the proposed algorithm by comparing it with the optimal solution obtained through exhaustive search. Due to the large cardinality of the set \(CS(n, m)\), exhaustive search is only possible for small value of \(n\) and \(m\). For this reason, we set \((n, m) = (10, 5), (15, 5), (20, 5)\) and compared the optimal solutions by a relative efficiency measure defined as

\[
\text{R.eff}_1 = \frac{\text{The value } \psi(R^*) \text{ corresponding to optimal scheme } R^* \text{ in exhaustive search}}{\text{The value } \psi(R^*) \text{ corresponding to optimal scheme } R^* \text{ in proposed approach}}.
\]

The optimal schemes and the corresponding relative efficiencies so determined are presented in Table 1. These results show that even though the proposed approach does not achieve the true optimal scheme \(R^*\), but the corresponding relative efficiencies are very close to 1. For this reason, we refer to the obtained solution \(R^*\) as near-optimal censoring scheme. In addition, we have reported two quantities \(n_{\text{it}}\) and \(n_{\text{ac}}\) representing the number of simulation runs and accepted
Table 1. Comparison between the optimal censoring schemes obtained through exhaustive search and the proposed algorithm under the criterion $\psi(R) = \int_0^R \text{Var} \ln X \, ds$ in the case of Weibull lifetime model.

| Parameters $\beta, k$ | Exhustive search $\begin{bmatrix} n & m & \mathcal{R}^* \end{bmatrix}$ | $\psi(\mathcal{R}^*)$ | $\begin{bmatrix} n_{it} & n_{ac} \end{bmatrix}$ | Uniform $\begin{bmatrix} n_{it} & n_{ac} \end{bmatrix}$ | Multinomial $\begin{bmatrix} R^* \end{bmatrix}$ | $\text{Reff}_1$ |
|-----------------------|------------------------------------------------|----------------|---------------------|---------------------|---------------------|----------|
| $(0.5, 1)$            | $\begin{bmatrix} 10 & 5 \end{bmatrix}$       | $\begin{bmatrix} 0, 4, 1, 0, 0 \end{bmatrix}$ | $2.4261$            | $\begin{bmatrix} 500 & 500 \end{bmatrix}$ | $(0, 4, 0, 0, 1)$  | $0.9995$ |
|                       |                                                 |                 |                     | $\begin{bmatrix} 1000 & 1000 \end{bmatrix}$ | $(0, 4, 0, 0, 1)$  | $0.9995$ |
|                       |                                                 |                 |                     | $\begin{bmatrix} 10000 & 8902 \end{bmatrix}$ | $(0, 4, 0, 0, 1)$  | $0.9995$ |
|                       |                                                 |                 |                     | $\begin{bmatrix} 50000 & 18936 \end{bmatrix}$ | $(0, 4, 0, 0, 1)$  | $0.9995$ |
|                       |                                                 |                 |                     | $\begin{bmatrix} 100000 & 99952 \end{bmatrix}$ | $(0, 4, 0, 0, 1)$  | $0.9995$ |
| $(0.5, 1)$            | $\begin{bmatrix} 10 & 5 \end{bmatrix}$       | $\begin{bmatrix} 0, 4, 1, 0, 0 \end{bmatrix}$ | $2.4261$            | $\begin{bmatrix} 500 & 412 \end{bmatrix}$  | $(0, 4, 0, 0, 1)$  | $0.9990$ |
|                       |                                                 |                 |                     | $\begin{bmatrix} 1000 & 420 \end{bmatrix}$  | $(0, 4, 0, 0, 1)$  | $0.9990$ |
|                       |                                                 |                 |                     | $\begin{bmatrix} 10000 & 9508 \end{bmatrix}$ | $(0, 4, 0, 0, 1)$  | $0.9990$ |
|                       |                                                 |                 |                     | $\begin{bmatrix} 50000 & 47487 \end{bmatrix}$ | $(0, 4, 0, 0, 1)$  | $0.9990$ |
|                       |                                                 |                 |                     | $\begin{bmatrix} 100000 & 49685 \end{bmatrix}$ | $(0, 4, 0, 0, 1)$  | $0.9990$ |
| $(1, 1)$              | $\begin{bmatrix} 15 & 5 \end{bmatrix}$       | $\begin{bmatrix} 0, 10, 0, 0, 0 \end{bmatrix}$ | $0.4983$            | $\begin{bmatrix} 500 & 310 \end{bmatrix}$  | $(0, 0, 5, 5, 0)$  | $0.9996$ |
|                       |                                                 |                 |                     | $\begin{bmatrix} 1000 & 508 \end{bmatrix}$  | $(0, 7, 2, 0, 1)$  | $0.9996$ |
|                       |                                                 |                 |                     | $\begin{bmatrix} 10000 & 5438 \end{bmatrix}$ | $(0, 8, 1, 1, 0)$  | $0.9996$ |
|                       |                                                 |                 |                     | $\begin{bmatrix} 50000 & 49507 \end{bmatrix}$ | $(0, 9, 0, 0, 1)$  | $0.9996$ |
|                       |                                                 |                 |                     | $\begin{bmatrix} 100000 & 26463 \end{bmatrix}$ | $(0, 0, 6, 3, 6)$  | $0.9998$ |
| $(1, 1)$              | $\begin{bmatrix} 15 & 5 \end{bmatrix}$       | $\begin{bmatrix} 0, 10, 0, 0, 0 \end{bmatrix}$ | $0.4983$            | $\begin{bmatrix} 500 & 310 \end{bmatrix}$  | $(0, 0, 5, 5, 0)$  | $0.9996$ |
|                       |                                                 |                 |                     | $\begin{bmatrix} 1000 & 508 \end{bmatrix}$  | $(0, 7, 2, 0, 1)$  | $0.9996$ |
|                       |                                                 |                 |                     | $\begin{bmatrix} 10000 & 5438 \end{bmatrix}$ | $(0, 8, 1, 1, 0)$  | $0.9996$ |
|                       |                                                 |                 |                     | $\begin{bmatrix} 50000 & 49507 \end{bmatrix}$ | $(0, 9, 0, 0, 1)$  | $0.9996$ |
|                       |                                                 |                 |                     | $\begin{bmatrix} 100000 & 26463 \end{bmatrix}$ | $(0, 0, 6, 3, 6)$  | $0.9998$ |
| $(2, 1)$              | $\begin{bmatrix} 20 & 5 \end{bmatrix}$       | $\begin{bmatrix} 0, 15, 0, 0, 0 \end{bmatrix}$ | $0.1113$            | $\begin{bmatrix} 500 & 244 \end{bmatrix}$  | $(0, 0, 3, 10, 2)$ | $0.9998$ |
|                       |                                                 |                 |                     | $\begin{bmatrix} 1000 & 414 \end{bmatrix}$  | $(0, 0, 3, 10, 2)$ | $0.9998$ |
|                       |                                                 |                 |                     | $\begin{bmatrix} 10000 & 4799 \end{bmatrix}$ | $(0, 9, 6, 0, 0)$  | $0.9998$ |
|                       |                                                 |                 |                     | $\begin{bmatrix} 50000 & 29089 \end{bmatrix}$ | $(0, 18, 0, 0, 1)$ | $0.9998$ |
|                       |                                                 |                 |                     | $\begin{bmatrix} 100000 & 26463 \end{bmatrix}$ | $(0, 0, 6, 3, 6)$  | $0.9998$ |
| $(2, 1)$              | $\begin{bmatrix} 20 & 5 \end{bmatrix}$       | $\begin{bmatrix} 0, 15, 0, 0, 0 \end{bmatrix}$ | $0.1113$            | $\begin{bmatrix} 500 & 440 \end{bmatrix}$  | $(0, 14, 0, 0, 1)$ | $0.9998$ |
|                       |                                                 |                 |                     | $\begin{bmatrix} 1000 & 987 \end{bmatrix}$  | $(0, 14, 0, 0, 1)$ | $0.9998$ |
|                       |                                                 |                 |                     | $\begin{bmatrix} 10000 & 8431 \end{bmatrix}$ | $(0, 14, 0, 0, 1)$ | $0.9998$ |

(continued)
candidates, respectively. The corresponding results show the performance of multinomial distribution to be better than the uniform and multivariate hypergeometric distributions in that multinomial needs less number of accepted candidates than the other two distributions to reach near-optimal scheme with the same relative efficiency when the cardinality of the set CS(n, m) is large.

As pointed out in Bhattacharya, Pradhan, and Dewanji (2016), the VNS algorithm provides near-optimal censoring schemes for large values of n and m. Therefore, we also compare the performance of the proposed approach with the VNS algorithm, and the corresponding results are reported in Table 2. We use the notation $R^*$ to denote the near-optimal scheme obtained through the VNS algorithm, and we define a relative efficiency measure as

$$R_{eff_2} = \frac{\text{The value of } \int_0^1 \text{Var} \left[ \ln X_s \right] ds \text{ corresponding to optimal scheme } R^* \text{ in VNS algorithm}}{\text{The value of } \int_0^1 \text{Var} \left[ \ln X_s \right] ds \text{ corresponding to optimal scheme } R^* \text{ in proposed approach}}.$$  

The results in Table 2 show that, even though the near-optimal schemes are different, the corresponding $R_{eff_2}$ are always 1 indicating that the proposed approach performs very well as compared to the VNS algorithm. While reporting the optimal schemes, we use the notation $a^b$ to denote that $a$ is repeated $b$ times. For example, the scheme $(0^5, 20, 0^4)$ for the case $(n, m) = (30, 10)$ under VNS algorithm in Table 2 refers to the scheme $(0, 0, 0, 0, 20, 0, 0, 0, 0)$. In a similar vein, the relative efficiency under cost criterion measure is defined as

$$R_{eff_3} = \frac{\text{The value of } C_0 + C_l m + C_i E[X_{nc:m:n}] \text{ corresponding to optimal scheme } R^* \text{ in VNS algorithm}}{\text{The value of } C_0 + C_l m + C_i E[X_{m:m:n}] \text{ corresponding to optimal scheme } R^* \text{ in proposed approach}}.$$  

Although the corresponding numerical results are not reported here, we found in our comparative study that the efficiencies $R_{eff_3}$ are all once again very close to one.

### 5. Data analysis

In this section, a real-life data set is analyzed to demonstrate the applicability of the proposed methodology. We took the data on log-times to break down of an insulating fluid in an accelerated test conducted at various test voltages (see Nelson 1982, Table 6.1, p.228). Using the censoring scheme $R^* = (0, 0, 3, 0, 3, 0, 5)$, Viveros and Balakrishnan (1994) presented a progressively censored sample of size $m = 8$ that had been randomly generated from $n = 19$

| Parameters | Exhaustive search | Uniform |
|------------|-------------------|---------|
| ($b, k$)   | ($n, m$)           | $R^*$   | $\psi(R^*)$ | $n_t$ | $n_{ac}$ | $R^*$ | $R_{eff_1}$ |
| (2, 1)     | (20, 5)            | (0, 15, 0, 0, 0) | 0.1113       | 500   | 246     | (4, 2, 0, 2, 7) | 0.9998 |
|            |                    |         |              | 1000  | 581     | (0, 1, 6, 2, 6) | 0.9998 |
|            |                    |         |              | 10000 | 5387    | (0, 1, 4, 8, 2) | 0.9998 |
|            |                    |         |              | 50000 | 26835   | (0, 3, 3, 6, 3) | 0.9998 |
|            |                    |         |              | 100000| 46463   | (0, 0, 6, 3, 6) | 0.9998 |

| Parameters | Exhaustive search | Multivariate hypergeometric |
|------------|-------------------|-----------------------------|
| ($b, k$)   | ($n, m$)           | $R^*$ | $\psi(R^*)$ | $n_t$ | $n_{ac}$ | $R^*$ | $R_{eff_1}$ |
| (2, 1)     | (20, 5)            | (0, 15, 0, 0, 0) | 0.1113       | 500   | 246     | (4, 2, 0, 2, 7) | 0.9998 |
|            |                    |         |              | 1000  | 581     | (0, 1, 6, 2, 6) | 0.9998 |
|            |                    |         |              | 10000 | 5387    | (0, 1, 4, 8, 2) | 0.9998 |
|            |                    |         |              | 50000 | 26835   | (0, 3, 3, 6, 3) | 0.9998 |
|            |                    |         |              | 100000| 46463   | (0, 0, 6, 3, 6) | 0.9998 |
observations recorded at 34 kilovolts in Nelson’s Table 6.1. These observations are reported in Table 3. Assuming an extreme value distribution, Viveros and Balakrishnan (1994) estimated the location and scale parameters as 2.222 and 1.026, respectively. Thence, the estimates of the corresponding shape and scale parameters for the Weibull distribution are found to be $\hat{\beta} = 1/1.026 = 0.974$ and $\hat{k} = e^{-2.222} = 0.108$, respectively. Now, assuming $\hat{\beta} = 0.974$ and $\hat{k} = 0.108$ as the true
values of \( \beta \) and \( k \), respectively, optimal progressive censoring schemes are determined with known values of \((n, m)\) as (19, 8) for three different choices of \( \pi_R(\cdot) \). We considered the variance measure defined in (1) and computed the \( R_{\text{eff}}^2 \) values. The results, presented in Table 4, show that the \( R_{\text{eff}}^2 \) values are always 1 although the optimal schemes are different, as expected.

6. Conclusions

In this work, we have used a probability structure to develop a simple algorithm for the determination of optimal progressive censoring scheme. We have used three probability distributions for \( \pi_R(\cdot) \) for illustrative purpose. The comparisons made with a direct exhaustive search method and the VNS algorithm demonstrate the efficiency and good performance of the proposed method.

Appendix

Variance and cost measures under weibull model

To derive explicit expressions for the two criteria used here, we first need the density of \( X_{i;m,n} \), given by the Kamps-Cramer representation (see Balakrishnan 2007; Balakrishnan and Cramer 2014), as

\[
f_{X_{i;m,n}}(x; \theta) = \sigma_{r-1} \sum_{p=1}^{i} a_p \gamma_i (1 - F(x; \theta))^\gamma_i - 1 f(x; \theta), \quad \text{for } i = 1, \ldots, m,
\]

where \( \gamma_i = m - r + 1 + \sum_{j=r}^{m} p_j \), for \( r = 1, \ldots, m, \sigma_{r-1} = \prod_{i=1}^{r-1} \gamma_i, \quad a_{i,r} = \prod_{j=1}^{r-1} \frac{1}{\gamma_j} \) for \( 1 \leq i \leq r \leq m \) with \( a_{1,1} = 1 \), and \( f(x; \theta) \) is the probability density function of \( X \) with \( \theta = (k, \beta) \). From the above density function of \( X_{i;m,n} \), \( E[X_{i;m,n}] \) can be found to be

\[
E[X_{i;m,n}] = \frac{1}{k} \Gamma \left( 1 + \frac{1}{\beta} \right) \sigma_{m-1} \sum_{p=1}^{i} a_p \gamma_i \frac{1}{\gamma_i^2}.
\]

Finally, to derive an expression of the variance measure, we need the expression of Fisher information matrix about \( \theta \) given by (see Bhattacharya, Pradhan, and Dewanji 2016)

\[
\mathcal{I}(\theta) = \begin{bmatrix}
\mathcal{I}_{11}(\theta) & \mathcal{I}_{12}(\theta) \\
\mathcal{I}_{21}(\theta) & \mathcal{I}_{22}(\theta)
\end{bmatrix},
\]

where

\[
\mathcal{I}_{11}(\theta) = \frac{1}{\beta^2} \sum_{i=1}^{m} \sum_{j=1}^{i} \sigma_{j-1} \frac{a_{i,j}}{\gamma_j} \int_0^\infty \left( 1 + \ln \left( \frac{z}{\gamma_j} \right) \right)^2 e^{-z} dz,
\]

\[
\mathcal{I}_{22}(\theta) = \left( \frac{\beta^2}{k} \right) \sum_{i=1}^{m} \sum_{j=1}^{i} \sigma_{j-1} \frac{a_{i,j}}{\gamma_j},
\]

\[
\mathcal{I}_{12}(\theta) = \mathcal{I}_{21}(\theta) = \frac{1}{k} \sum_{i=1}^{m} \sum_{j=1}^{i} \sigma_{j-1} \frac{a_{i,j}}{\gamma_j} \int_0^\infty \left( 1 + \ln \left( \frac{z}{\gamma_j} \right) \right) e^{-z} dz.
\]

It is difficult to find the exact expression of \( \int_0^s \text{Var}[\ln X_i] ds \). However, by using delta method, the asymptotic expression of \( \text{Var}[\ln X_i] \) can be found to be \( \text{Var}[\ln X_i] = (\mathcal{I}^{11}(\theta)/\beta^2) (g(s))^2 + (2 \mathcal{I}^{12}(\theta)/\beta^2 k) g(s) + (2 \mathcal{I}^{22}(\theta)/k^2) \), where \( g(s) = \ln \{ - \ln (1 - s) \} \), and \( \mathcal{I}^{11}(\theta), \mathcal{I}^{12}(\theta) \) and \( \mathcal{I}^{22}(\theta) \) are the elements of the inverse of the Fisher information matrix given by

\[
\mathcal{I}^{-1}(\theta) = \begin{bmatrix}
\mathcal{I}^{-11}(\theta) & \mathcal{I}^{-12}(\theta) \\
\mathcal{I}^{-21}(\theta) & \mathcal{I}^{-22}(\theta)
\end{bmatrix}.
\]

Therefore, an asymptotic expression of \( \int_0^1 \text{Var}[\ln X_i] ds \) can be obtained readily as

\[
\frac{\mathcal{I}^{11}(\theta)}{\beta^4} \int_0^1 (g(s))^2 ds + \frac{2 \mathcal{I}^{12}(\theta)/\beta^2 k}{g(s)} \int_0^1 g(s) ds + \frac{\mathcal{I}^{22}(\theta)}{k^2}.
\]
Acknowledgments

The authors express their sincere thanks to two anonymous referees and an associate editor for their critical comments and valuable suggestions which have resulted in an improvement over the earlier version of the manuscript. The second author thanks the National Sciences and Engineering Research Council of Canada for supporting this research through an Individual Discovery Grant RGPIN-2020-06733.

ORCID

Ritwik Bhattacharya http://orcid.org/0000-0003-4732-7994

References

Balakrishnan, N. 2007. Progressive censoring methodology: An appraisal. TEST 16 (2):211–96. with discussions). doi:10.1007/s11749-007-0061-y.
Balakrishnan, N, and R. Aggarwala. 2000. Progressive censoring: Theory, methods, and applications. Boston, MA: Birkhäuser.
Balakrishnan, N., M. Burkschat, E. Cramer, and G. Hofmann. 2008. Fisher information based progressive censoring plans. Computational Statistics & Data Analysis 53 (2):366–80. doi:10.1016/j.csda.2008.07.058.
Balakrishnan, N, and E. Cramer. 2014. The art of progressive censoring: Application to reliability and quality. Boston, MA: Birkhäuser.
Bhattacharya, R. 2020. Implementation of compound optimal design strategy in censored life-testing experiment. TEST 29 (4):1029–50. doi:10.1007/s11749-019-00699-7.
Bhattacharya, R., B. Pradhan, and A. Dewanji. 2016. On optimum life-testing plans under Type-II progressive censoring scheme using variable neighborhood search algorithm. TEST 25 (2):309–30. doi:10.1007/s11749-015-0449-z.
Burkschat, M. 2008. On optimality of extremal schemes in progressive Type II censoring. Journal of Statistical Planning and Inference 138 (6):1647–59. doi:10.1016/j.jspi.2007.05.042.
Burkschat, M., E. Cramer, and U. Kamps. 2006. On optimal schemes in progressive censoring. Statistics & Probability Letters 76 (10):1032–6. doi:10.1016/j.spl.2005.12.011.
Burkschat, M., E. Cramer, and U. Kamps. 2007. Optimality criteria and optimal schemes in progressive censoring. Communications in Statistics - Theory and Methods 36 (7):1419–31. doi:10.1080/03610920601077196.
Cohen, A. C. 1963. Progressively censored samples in life testing. Technometrics 5 (3):327–9. doi:10.1080/00949655.1963.10490102.
Cohen, A. C. 1966. Life testing and early failure. Technometrics 8 (3):539–49. doi:10.2307/1266699.
Dahmen, K., M. Burkschat, and E. Cramer. 2012. A- and D- optimal progressive Type-II censoring designs based on fisher information. Journal of Statistical Computation and Simulation 82 (6):879–905. doi:10.1080/00949655.2011.560118.
Johnson, N. L. S. Kotz, and N. Balakrishnan. 1997. Discrete multivariate distributions. New York: John Wiley & Sons.
Nelson, W. 1982. Applied life data analysis. New York: John Wiley & Sons.
Ng, H. K. T., P. S. Chan, and N. Balakrishnan. 2004. Optimal progressive censoring plans for the Weibull distribution. Technometrics 46 (4):470–81. doi:10.1198/004017004000000482.
Pradhan, B, and D. Kundu. 2009. On progressively censored generalized exponential distribution. TEST 18 (3):497–515. doi:10.1007/s11749-008-0110-1.
Pradhan, B, and D. Kundu. 2013. Inference and optimal censoring schemes for progressively censored Birnbaum-Saunders distribution. Journal of Statistical Planning and Inference 143 (6):1098–108. doi:10.1016/j.jspi.2012.11.007.
Viveros, R, and N. Balakrishnan. 1994. Interval estimation of parameters of life from progressively censored data. Technometrics 36 (1):84–91. doi:10.1080/00401706.1994.10485403.