Calculating the Distribution of a Linear Combination of Uniform Order Statistics

Morganna Carmem Diniz • Edmundo de Souza e Silva • H. Richard Gail
Faculdades IBMEC/RJ, Av. Rio Branco 108, Rio de Janeiro - RJ, Brazil
CS Dept./COPPE, Federal University of Rio de Janeiro, CxP 68511, 21945-970, Brazil
IBM T.J. Watson Research Center, P.O. Box 704, Yorktown Heights, New York 10598, USA
morganna@ibmecrj.br • edmundo@land.ufrj.br • rgail@us.ibm.com

The calculation of the distribution of a linear combination of order statistics from random variables that are uniformly distributed is considered. A simple recursion to compute this distribution is presented that, unlike previous methods, is numerically stable and efficient. As such, this should be the algorithm of choice when the linear combination distribution needs to be obtained.

(*Reliability: Availability; Probability: Stochastic Model Applications; Probability: Markov Processes; Probability: Distributions; Statistics; Order Statistics; Performability*)

1. Introduction

The distribution of a linear combination of order statistics from a set of random variables is a measure of interest in many problems. Examples include asymptotic theory (Chernoff et al. 1967, Eicker and Puri 1976), estimation and hypothesis testing (D'Agostino and Stephens 1986, David 1981), and, more recently for the case of uniform random variables, performability modeling and analysis (de Souza e Silva and Gail 1989, 1998, Qureshi and Sanders 1994). Calculating the distribution of a linear combination of random variables is not an easy task, even in the uniform case.

Let $a_1, \ldots, a_n$ be non-negative real numbers, and let $U_{(1)}, \ldots, U_{(n)}$ be the order statistics from a set of n independent and identically distributed random variables uniform on $(0, 1)$. The random variable $G = \sum_{j=1}^{n} a_j U_{(j)}$ is a linear combination of the order statistics $U_{(j)}$. Dempster and Kleyle (1968) obtained an expression for the distribution of G when all order statistics are present in the sum, that is, when $a_j > 0, j = 1, \ldots, n$. Weisberg (1971) generalized the results of Dempster and Kleyle and derived a formula for the distribution of a linear combination of *selected* order statistics from a uniform distribution, that is, for the case when some of the a_j are zero. From this formula Weisberg developed a recursive algorithm to calculate the distribution of G.

Matsunawa (1985) considered the probability density function for a linear combination of selected order statistics from a uniform distribution. He showed that G can be represented in terms of a ratio of linear combinations of mutually independent gamma random variables. Furthermore, he showed that the pdf of G is the same as that of a mixture of scaled beta distributions. Normal approximations to the exact distribution are also investigated in Matsunawa (1985). Ramallingam (1989) made a slight correction to the pdf formula of Matsunawa (1985) and suggested a method of computing expressions for the coefficients of the pdf formula using symbolic manipulation. This technique is also useful for calculating the moments of G.

One problem with using the approach of Matsunawa for computation is the complexity involved in evaluating the coefficients from the expression given in Matsunawa (1985) (see, for instance, equation 3.8 in that paper). Furthermore, although the formula for the distribution of G and the corresponding recursive procedure given by Weisberg are relatively simple, the

0899-1499/01/0000/0001$5.00
1526-5528 electronic ISSN

resulting calculations are unfortunately susceptible to numerical problems. In fact, the procedure may lead to the subtraction of large numbers that are close in modulus, with a subsequent loss of precision or an overflow/underflow condition. The above problems make it difficult to compute the distribution of G with known techniques.

The calculation of the distribution of a linear combination of uniform order statistics is also of interest in the area of *performability* modeling and analysis. Consider a Markov process for which reward rates are associated with the states of the process, i.e., when the system is in a certain state it earns a reward at the rate corresponding to that state. The distribution of the total reward accumulated during a finite observation period, which is also known as the performability distribution (Meyer 1980), was obtained by de Souza e Silva and Gail (1989, 1998). The methodology that they proposed was based on the observation that, after first using the uniformization technique (Grassmann 1977a, b) to transform the original Markov process, the performance measure of interest can be calculated using the distribution of the linear combination G. Although the algorithm presented in this paper cannot be used directly in computing the distribution of accumulated reward, it is shown in Diniz (2000) that a slightly more expensive recursion for the distribution of G can lead to an efficient and stable algorithm for calculating the total reward using the methodology of de Souza e Silva and Gail (1989, 1998).

The purpose of this paper is to derive a new, simple recursion for calculating the distribution of G that is also numerically robust, since it involves only the addition and multiplication of terms that are probabilities. In Section 2 relevant background material is provided, while in Section 3 the main results that lead to the new algorithm we develop are presented. Section 4 includes simple examples that illustrate numerical problems that may occur when using recursions from previous work, and it also includes comparisons of results from these recursions with those obtained from our algorithm. Finally in Section 5 we summarize the paper and discuss additional work generated from the results presented here.

2. Background Material

In this section we present necessary background material and introduce the notation that will be used throughout the paper. We begin by presenting a brief review of linear combinations of order statistics.

Let $U_1(t), \ldots, U_n(t)$ be independent and identically distributed random variables uniform on $(0, t)$, and let $U_{(j)}(t)$ be the jth smallest value from these random variables with the convention that $U_{(0)}(t) = 0$ and $U_{(n+1)}(t) = t$. Then $U_{(1)}(t), \ldots, U_{(n)}(t)$ are the order statistics of $U_1(t), \ldots, U_n(t)$. Since the random variable $U_{(j)}(t)$ associated with the interval $(0, t)$ has the same distribution as $tU_{(j)}(1)$ associated with the interval $(0, 1)$, without loss of generality we may consider the case with $t = 1$. In the remainder of the paper we use the simplifying notation $U_{(j)} \stackrel{\Delta}{=} U_{(j)}(1)$. Figure 1 shows an example for $n = 5$.

Consider the linear combination of order statistics

$$G = \sum_{j=1}^{n} a_j U_{(j)}, \tag{1}$$

where each $a_j \geq 0$ is a real number. The interval lengths $Y_j = U_{(j)} - U_{(j-1)}$, $j = 1, \ldots, n+1$, are exchangeable random variables (Ross 1983). Substituting the formula $U_{(j)} = \sum_{i=1}^{j} Y_i$ into (1) and interchanging summations in the resulting expression yields

$$G = \sum_{j=1}^{n+1} d_j Y_j, \tag{2}$$

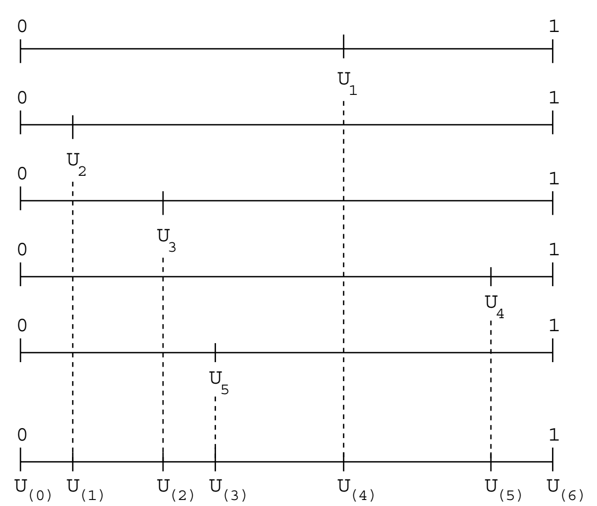

Figure 1 A Set of Uniform Order Statistics on $(0, 1)$

where we have defined $d_j = a_j + \cdots + a_n$, $j = 1, \ldots, n$, and also set $d_{n+1} = 0$.

From the definition of d_j, we see that $d_1 \geq \cdots \geq d_{n+1} = 0$ and $d_{j+1} = d_j$ when $a_j = 0$ for $j = 1, \ldots, n$. Let $\mathscr{C} = \{c_1, \ldots, c_S\}$ be the set of distinct values among $d_1, \ldots, d_{n+1}$ (thus $1 < S \leq n+1$). Then $c_1 > \cdots > c_S = 0$, and so we have $c_1 \geq G \geq c_S$ since $\sum_{j=1}^{n+1} Y_j = 1$. Define $\mathbf{k} = \langle k_1, \ldots, k_S \rangle$ where k_l is the number of interval lengths Y_j associated with c_l, $l = 1, \ldots, S$. Note that $\|\mathbf{k}\| \stackrel{\Delta}{=} k_1 + \cdots + k_S = n+1 \geq 1$, since $n \geq 0$.

A solution to the problem of determining the distribution of a linear combination of uniform order statistics was found by Weisberg (1971). As a consequence of Weisberg's result it was shown in de Souza e Silva and Gail (1998) that the distribution of $G = G(\mathbf{k})$ is given by

$$P[G(\mathbf{k}) \leq r] = \sum_{l:c_l \leq r} \frac{f_l^{(k_l-1)}(c_l, r, \mathbf{k})}{(k_l - 1)!} \tag{3}$$

$$P[G(\mathbf{k}) > r] = \sum_{l:c_l > r} \frac{f_l^{(k_l-1)}(c_l, r, \mathbf{k})}{(k_l - 1)!}, \tag{4}$$

where $f_l^{(k_l-1)}(c_l, r, \mathbf{k}), l = 1, \ldots, S,$ is the $(k_l - 1)$st derivative of

$$f_l(x, r, \mathbf{k}) = \frac{(x-r)^n}{\prod_{j \neq l}(x - c_j)^{k_j}} \tag{5}$$

evaluated at $x = c_l \in \mathscr{C}$.

In Weisberg (1971) a recursive formula was developed to calculate the derivatives of f_l, namely,

$$f_l^{(q)}(x, r, \mathbf{k}) = \sum_{j=0}^{q-1} \binom{q-1}{j} f_l^{(j)}(x, r, \mathbf{k}) g_l^{(q-1-j)}(x, r, \mathbf{k}), \tag{6}$$

where

$$g_l^{(j)}(x, r, \mathbf{k}) = (-1)^j j! \left[\frac{n}{(x-r)^{j+1}} - \sum_{i \neq l} \frac{c_i}{(x - c_i)^{j+1}} \right]. \tag{7}$$

The presence of positive and negative terms in the summation above can lead to numerical problems when using this recursion.

Remark. The general case of arbitrary real a_j can be reduced to the case for which $a_j \geq 0$ by a reordering and shifting, as suggested in Dempster and Kleyle (1968). Specifically, first reorder the d_j of equation (2) in descending order, that is, let σ be a permutation of $\{1, \ldots, n+1\}$ such that $d_{\sigma(1)} \geq \cdots \geq d_{\sigma(n+1)}$. Note that $G = \sum_{j=1}^{n+1} d_{\sigma(j)} Y_{\sigma(j)}$. Since the Y_j are exchangeable, the random variable $H \stackrel{\Delta}{=} \sum_{j=1}^{n+1} d_{\sigma(j)} Y_j$ has the same distribution as G. Using the definition of Y_j, we have

$$H = \sum_{j=1}^{n} (d_{\sigma(j)} - d_{\sigma(j+1)}) U_{(j)} + d_{\sigma(n+1)} U_{(n+1)}.$$

Since $U_{(n+1)} = 1$, we see that $G^* \stackrel{\Delta}{=} H - d_{\sigma(n+1)}$ is a linear combination of $U_{(1)}, \ldots, U_{(n)}$ with non-negative coefficients. Thus we have $P[G \leq r] = P[H \leq r] = P[G^* \leq r - d_{\sigma(n+1)}]$.

3. The New Algorithm

In this section we present a new efficient algorithm for calculating the distribution of a linear combination of uniform order statistics. We then discuss the issues related to its implementation and its computational requirements.

3.1. Recursions for f_l

The form (5) of the functions f_l will be exploited to obtain a stable recursion for $P[G(\mathbf{k}) > r]$.

Definition 1. For $\mathbf{k} = \langle k_1, \ldots, k_S \rangle$ where $\|\mathbf{k}\| = n+1$, define

$$h(x, r, \mathbf{k}) = \frac{(x-r)^n}{\prod_{l=1}^{S} (x - c_l)^{k_l}}. \tag{8}$$

Lemma 1. *For $i \neq j$, $k_i > 0$, $k_j > 0$, we have*

$$\left(\frac{c_i - r}{c_i - c_j} \right) h(x, r, \mathbf{k} - \mathbf{1}_j) + \left(\frac{r - c_j}{c_i - c_j} \right) h(x, r, \mathbf{k} - \mathbf{1}_i) = h(x, r, \mathbf{k}). \tag{9}$$

Proof. A computation shows that

$$\begin{aligned} &\left(\frac{c_i - r}{c_i - c_j} \right) h(x, r, \mathbf{k} - \mathbf{1}_j) + \left(\frac{r - c_j}{c_i - c_j} \right) h(x, r, \mathbf{k} - \mathbf{1}_i) \\ &\quad = \left(\frac{c_i - r}{c_i - c_j} \right) h(x, r, \mathbf{k}) \frac{x - c_j}{x - r} \\ &\qquad + \left(1 - \frac{c_i - r}{c_i - c_j} \right) h(x, r, \mathbf{k}) \frac{x - c_i}{x - r} \end{aligned}$$

$$= \frac{h(x,r,\mathbf{k})}{x-r} \times \left\{(x-c_i) + \left(\frac{c_i-r}{c_i-c_j}\right)[(x-c_j)-(x-c_i)]\right\}$$
$$= h(x,r,\mathbf{k}),$$

and so (9) holds. □

For $l=1,\ldots,S$, Weisberg's function f_l is simply $f_l(x,r,\mathbf{k}) = (x-c_l)^{k_l} h(x,r,\mathbf{k})$. This motivates the following definition.

Definition 2. For $l=1,\ldots,S$, and integer m, let

$$h_l(x,r,\mathbf{k},m) = (x-c_l)^m h(x,r,\mathbf{k}). \tag{10}$$

Note that $f_l(x,r,\mathbf{k}) = h_l(x,r,\mathbf{k},k_l)$.

Lemma 2. *For $i \neq j$, $k_i > 0$, $k_j > 0$, and integers $q \geq 0$ and m, we have*

$$\left(\frac{c_i-r}{c_i-c_j}\right) h_l^{(q)}(x,r,\mathbf{k}-\mathbf{1}_j,m) + \left(\frac{r-c_j}{c_i-c_j}\right) h_l^{(q)}(x,r,\mathbf{k}-\mathbf{1}_i,m) = h_l^{(q)}(x,r,\mathbf{k},m). \tag{11}$$

Proof. Multiplying equation (9) by $(x-c_l)^m$, we obtain (11) for the case $q=0$. Differentiating this result q times gives (11) for general q. □

Dividing the terms of (11) by $q!$ yields a recursion that, at first glance, seems to apply to the calculation of the distribution of G in equations (3)–(4). However, such a recursion would involve setting m equal to the lth entry of the vectors $\mathbf{k}, \mathbf{k}-\mathbf{1}_i, \mathbf{k}-\mathbf{1}_j$, in the three terms of (11), and these values are not necessarily the same constant. But this potential problem vanishes when the functions are evaluated at the point $x=c_l$, as is the case in (3)–(4).

Lemma 3. *For $m \geq k_l$, $q \geq 0$, we have*

$$\frac{h_l^{(q+1)}(c_l,r,\mathbf{k},m+1)}{(q+1)!} = \frac{h_l^{(q)}(c_l,r,\mathbf{k},m)}{q!}. \tag{12}$$

Proof. First note that

$$h_l(x,r,\mathbf{k},m+1) = (x-c_l) h_l(x,r,\mathbf{k},m).$$

Differentiating repeatedly yields

$$h_l^{(q+1)}(x,r,\mathbf{k},m+1) = (x-c_l) h_l^{(q+1)}(x,r,\mathbf{k},m) + (q+1) h_l^{(q)}(x,r,\mathbf{k},m), \tag{13}$$

where recall that

$$h_l(x,r,\mathbf{k},m) = (x-c_l)^m h(x,r,\mathbf{k}) = (x-c_l)^{m-k_l} \frac{(x-r)^n}{\prod_{j\neq l}(x-c_j)^{k_j}}.$$

Now $h_l^{(q)}(c_l,r,\mathbf{k},m)$ is finite for all q, since $m-k_l \geq 0$ and $c_j \neq c_l$ for $j \neq l$. Therefore, evaluating (13) at $x=c_l$ and dividing the result by $(q+1)!$ gives (12). □

Definition 3. For $l=1,\ldots,S$, let

$$\Theta_l(r,\mathbf{k}) = \begin{cases} \dfrac{f_l^{(k_l-1)}(c_l,r,\mathbf{k})}{(k_l-1)!} & \text{if } k_l > 0 \\ 0 & \text{if } k_l = 0. \end{cases} \tag{14}$$

In the above definition recall that $f_l(c_l,r,\mathbf{k}) = h_l(c_l,r,\mathbf{k},k_l)$.

Theorem 1. *For $i \neq j$, $k_i > 0$, $k_j > 0$, $l=1,\ldots,S$, we have*

$$\left(\frac{c_i-r}{c_i-c_j}\right)\Theta_l(r,\mathbf{k}-\mathbf{1}_j) + \left(\frac{r-c_j}{c_i-c_j}\right)\Theta_l(r,\mathbf{k}-\mathbf{1}_i) = \Theta_l(r,\mathbf{k}). \tag{15}$$

Proof. First suppose that $l \neq i$, $l \neq j$. In this case $(\mathbf{k}-\mathbf{1}_j)_l = (\mathbf{k}-\mathbf{1}_i)_l = k_l$. If $k_l = 0$, then all Θ_l terms in (15) are zero and the result holds. Otherwise, (15) follows by evaluating (11) at $x=c_l$ and setting $q=k_l-1$, $m=k_l$.

Next assume that $l=i$. In this case $k_l > 0$, and $(\mathbf{k}-\mathbf{1}_j)_l = k_l$, $(\mathbf{k}-\mathbf{1}_i)_l = k_l - 1$. First suppose that $k_l = 1$. Then $h_l(c_l,r,\mathbf{k}-\mathbf{1}_i,1) = 0$, since $h(c_l,r,\mathbf{k}-\mathbf{1}_i) = h(c_l,r,\mathbf{k}-\mathbf{1}_l)$ is finite. Thus from (11) with $m=1$ and $q=0$ we obtain

$$\left(\frac{c_i-r}{c_i-c_j}\right) f_l(r,\mathbf{k}-\mathbf{1}_j) = f_l(r,\mathbf{k}). \tag{16}$$

Now $(\mathbf{k}-\mathbf{1}_i)_l = 0$ when $k_l = 1$, so $\Theta_l(r,\mathbf{k}-\mathbf{1}_i) = 0$. Thus (16) is really (15) in this case. Next suppose that $k_l > 1$. Then from (12), we have

$$\frac{h_l^{(k_l-1)}(c_l,r,\mathbf{k}-\mathbf{1}_i,k_l)}{(k_l-1)!} = \frac{h_l^{(k_l-2)}(c_l,r,\mathbf{k}-\mathbf{1}_i,k_l-1)}{(k_l-2)!} = \Theta_l(r,\mathbf{k}-\mathbf{1}_i). \tag{17}$$

Evaluate (11) at $x=c_l$ and set $q=k_l-1$, $m=k_l$. Then, using (17) on the second term on the left hand side yields (15).

Finally assume that $l = j$. Then we have $(\mathbf{k}-\mathbf{1}_j)_l = k_l - 1$ and $(\mathbf{k}-\mathbf{1}_i)_l = k_l$. The proof proceeds in the same manner as that for the case $l = i$ above. □

We have shown how to calculate $\Theta_l(r, \mathbf{k})$ recursively on $\|\mathbf{k}\|$ for vectors with at least two nonzero entries. Since $\|\mathbf{k}\| \geq 1$, we need only determine $\Theta_l(r, \mathbf{k})$ for those vectors $\mathbf{k}$ with exactly one nonzero entry (i.e., $\mathbf{k} = k_j \mathbf{1}_j$ for some j) to complete the recursion. Noting that an empty product is equal to 1, from (5) and (14) the initial conditions are (for $j = 1, \ldots, S$, $l = 1, \ldots, S$)

$$\Theta_l(r, k_j \mathbf{1}_j) = \begin{cases} 1 & j = l \\ 0 & j \neq l. \end{cases} \tag{18}$$

Similar recursions can be obtained for arbitrary sums of the Θ_l as follows.

Definition 4. For $\|\mathbf{k}\| = n+1$, $\mathcal{S} \subset \{1, \ldots, S\}$, let

$$\Gamma(r, \mathbf{k}, \mathcal{S}) = \sum_{l \in \mathcal{S}} \Theta_l(r, \mathbf{k}). \tag{19}$$

Theorem 2. *For $i \neq j$, $k_i > 0$, $k_j > 0$, $\mathcal{S} \subset \{1, \ldots, S\}$, we have*

$$\left(\frac{c_i - r}{c_i - c_j}\right) \Gamma(r, \mathbf{k}-\mathbf{1}_j, \mathcal{S}) + \left(\frac{r - c_j}{c_i - c_j}\right) \Gamma(r, \mathbf{k}-\mathbf{1}_i, \mathcal{S}) = \Gamma(r, \mathbf{k}, \mathcal{S}). \tag{20}$$

Proof. Use Theorem 1 for each individual Θ_l and sum the results over $l \in \mathcal{S}$. □

The initial conditions in this case are (for $j = 1, \ldots, S$)

$$\Gamma(r, k_j \mathbf{1}_j, \mathcal{S}) = \begin{cases} 1 & \text{if } j \in \mathcal{S} \\ 0 & \text{if } j \notin \mathcal{S}. \end{cases} \tag{21}$$

3.2. A Stable Recursion for the Distribution of G

From the characterization (3)–(4) of the distribution of G, certain values of the subset $\mathcal{S}$ in (19) yield recursions for $P[G(\mathbf{k}) \leq r]$ and $P[G(\mathbf{k}) > r]$ using Theorem 2. In particular, two subsets of interest are

$$\mathcal{G} \stackrel{\Delta}{=} \{l : c_l > r\}, \qquad \mathcal{L} \stackrel{\Delta}{=} \{l : c_l \leq r\}. \tag{22}$$

Definition 5. For $\|\mathbf{k}\| = n+1$, let

$$\Theta(r, \mathbf{k}) = \Gamma(r, \mathbf{k}, \mathcal{G}) = \sum_{l: c_l > r} \Theta_l(r, \mathbf{k}), \tag{23}$$

$$\Omega(r, \mathbf{k}) = \Gamma(r, \mathbf{k}, \mathcal{L}) = \sum_{l: c_l \leq r} \Theta_l(r, \mathbf{k}). \tag{24}$$

Note from (3)–(4) and (14) that

$$\Theta(r, \mathbf{k}) = \sum_{l: c_l > r} \frac{f_l^{k_l - 1}(c_l, r, \mathbf{k})}{(k_l - 1)!} = P[G(\mathbf{k}) > r], \tag{25}$$

$$\Omega(r, \mathbf{k}) = \sum_{l: c_l \leq r} \frac{f_l^{k_l - 1}(c_l, r, \mathbf{k})}{(k_l - 1)!} = P[G(\mathbf{k}) \leq r]. \tag{26}$$

Therefore, recursions for Θ or Ω yield algorithms for calculating the distribution for a linear combination of uniform order statistics.

Recall that $c_1 > \cdots > c_S = 0$ and $c_1 \geq G \geq c_S$. If $c_S > r$ then $P[G(\mathbf{k}) > r] = 1$, while if $c_1 \leq r$ then $P[G(\mathbf{k}) \leq r] = 1$. Thus, to avoid trivial cases we assume that $c_1 > r \geq c_S$, that is, we assume that $\mathcal{G}$ and $\mathcal{L}$ are nonempty sets. When applying the recursion of Theorem 2 to Θ or Ω to calculate the distribution of G, a judicious choice of the indices i and j should be made at each step in order to ensure that the resulting algorithm is numerically stable. Suppose $\|\mathbf{k}\| = n+1$ and $k_i > 0$, $k_j > 0$ for some $i \in \mathcal{G}$, $j \in \mathcal{L}$. An important observation about using this choice in Theorem 2 is that the coefficients satisfy

$$0 \leq \frac{c_i - r}{c_i - c_j} \leq 1, \quad 0 \leq \frac{r - c_j}{c_i - c_j} \leq 1, \quad \frac{c_i - r}{c_i - c_j} + \frac{r - c_j}{c_i - c_j} = 1,$$

because $c_i > r \geq c_j$. That is, with such a choice at each step the algorithm only deals with sums and products of real numbers between 0 and 1, and so it is numerically stable.

Definition 6. For $\|\mathbf{k}\| = n+1$, $\mathcal{S} \subset \{1, \ldots, S\}$, let $\mathbf{k}_{\mathcal{S}}$ be the vector with entries

$$(\mathbf{k}_{\mathcal{S}})_l = \begin{cases} k_l & \text{if } l \in \mathcal{S} \\ 0 & \text{if } l \notin \mathcal{S}. \end{cases} \tag{27}$$

Note that $\|\mathbf{k}_{\mathcal{S}}\| = \sum_{l \in \mathcal{S}} k_l$.

Since $\mathcal{G}$ and $\mathcal{L}$ partition the set $\{1, \ldots, S\}$, we have $\mathbf{k}_{\mathcal{G}} + \mathbf{k}_{\mathcal{L}} = \mathbf{k}$ for any vector $\mathbf{k}$. The stable recursions for Θ or Ω based on Theorem 2 handle the case of $\mathbf{k}$ for which both $\|\mathbf{k}_{\mathcal{G}}\| > 0$ and $\|\mathbf{k}_{\mathcal{L}}\| > 0$. The initial conditions for these recursions involve the case for which either $\|\mathbf{k}_{\mathcal{G}}\| = 0$ or $\|\mathbf{k}_{\mathcal{L}}\| = 0$. Note that $\mathbf{k}_{\mathcal{G}}$ and $\mathbf{k}_{\mathcal{L}}$ cannot both be zero vectors, since $\|\mathbf{k}_{\mathcal{G}}\| + \|\mathbf{k}_{\mathcal{L}}\| = \|\mathbf{k}\| = n+1 \geq 1$.

Theorem 3. *The following give numerically stable recursions for calculating $P[G(\mathbf{k}) > r]$ in part* (i) *and $P[G(\mathbf{k}) \leq r]$ in part* (ii).

(i) *For* $\|\mathbf{k}_{\mathscr{G}}\| > 0$, $\|\mathbf{k}_{\mathscr{L}}\| > 0$, *choose* $i \in \mathscr{G}$, $j \in \mathscr{L}$ *such that both* $k_i > 0$, $k_j > 0$. *We have*

$$\left(\frac{c_i - r}{c_i - c_j}\right)\Theta(r, \mathbf{k} - \mathbf{1}_j) + \left(\frac{r - c_j}{c_i - c_j}\right)\Theta(r, \mathbf{k} - \mathbf{1}_i) = \Theta(r, \mathbf{k}). \tag{28}$$

The initial conditions are: *For either* $\|\mathbf{k}_{\mathscr{G}}\| = 0$ *or* $\|\mathbf{k}_{\mathscr{L}}\| = 0$ (*but not both*)

$$\Theta(r, \mathbf{k}) = \begin{cases} 0 & \text{if } \|\mathbf{k}_{\mathscr{G}}\| = 0 \\ 1 & \text{if } \|\mathbf{k}_{\mathscr{L}}\| = 0. \end{cases} \tag{29}$$

(ii) *For* $\|\mathbf{k}_{\mathscr{G}}\| > 0$, $\|\mathbf{k}_{\mathscr{L}}\| > 0$, *choose* $i \in \mathscr{G}$, $j \in \mathscr{L}$ *such that both* $k_i > 0$, $k_j > 0$. *We have*

$$\left(\frac{c_i - r}{c_i - c_j}\right)\Omega(r, \mathbf{k} - \mathbf{1}_j) + \left(\frac{r - c_j}{c_i - c_j}\right)\Omega(r, \mathbf{k} - \mathbf{1}_i) = \Omega(r, \mathbf{k}). \tag{30}$$

The initial conditions are: *For either* $\|\mathbf{k}_{\mathscr{G}}\| = 0$ *or* $\|\mathbf{k}_{\mathscr{L}}\| = 0$ (*but not both*)

$$\Omega(r, \mathbf{k}) = \begin{cases} 1 & \text{if } \|\mathbf{k}_{\mathscr{G}}\| = 0 \\ 0 & \text{if } \|\mathbf{k}_{\mathscr{L}}\| = 0. \end{cases} \tag{31}$$

Note that the recursions for Θ and Ω have the same form, but the initial conditions for these two measures are the opposite of one another. Although one could theoretically compute the distribution of G using the recursion for the complementary distribution and then calculating $1 - \Theta$, such a procedure should be avoided since the subtraction of similar quantities could lead to numerical problems. A direct application of the recursion for Ω in Theorem 3 should be used instead. A similar comment applies to computing the complementary distribution, which should always be done directly using the recursion for Θ.

Remark. Instead of using a convex combination of just two Θ, it may be useful to consider sums with more terms. To this end, we prove the following lemma.

Lemma 4. *Let* $\mathscr{T} \subset \{1, \ldots, S\}$, *and suppose* $k_l > 0$ *for all* $l \in \mathscr{T}$. *Suppose there are* α_l, $l \in \mathscr{T}$, *such that*: (i) $\sum_{l\in\mathscr{T}} \alpha_l c_l = r$; *and* (ii) $\sum_{l\in\mathscr{T}} \alpha_l = 1$. *Then*

$$\sum_{l\in\mathscr{T}} \alpha_l h(x, r, \mathbf{k} - \mathbf{1}_l) = h(x, r, \mathbf{k}).$$

Proof. We have

$$\begin{aligned} \sum_{l\in\mathscr{T}} \alpha_l h(x, r, \mathbf{k} - \mathbf{1}_l) &= \sum_{l\in\mathscr{T}} \alpha_l h(x, r, \mathbf{k}) \frac{x - c_l}{x - r} \\ &= \frac{h(x, r, \mathbf{k})}{x - r} \sum_{l\in\mathscr{T}} \alpha_l (x - c_l) \\ &= h(x, r, \mathbf{k}), \end{aligned}$$

where the last equality follows from (i) and (ii). □

Note that when $\mathscr{T} = \{i, j\}$, we have the previous result. In this case, $\alpha_i c_i + \alpha_j c_j = r$ and $\alpha_i + \alpha_j = 1$. The only solution of these two equations is

$$\alpha_j = \frac{c_i - r}{c_i - c_j}, \qquad \alpha_i = 1 - \frac{c_i - r}{c_i - c_j} = \frac{r - c_j}{c_i - c_j}.$$

3.3. Computational Requirements

Suppose $\mathbf{k}^*$ is a given vector, $r \geq 0$, and consider the recursion for $P[G(\mathbf{k}^*) > r]$ described in part (i) of Theorem 3. To compute $\Theta(r, \mathbf{k}^*)$, note that only the values of $\Theta(r, \mathbf{k}^* - \mathbf{1}_i)$ and $\Theta(r, \mathbf{k}^* - \mathbf{1}_j)$ need to be stored, where $c_i > r \geq c_j$, $k_i^* > 0$ and $k_j^* > 0$. For notational convenience, set $\|\mathbf{k}_{\mathscr{G}}^*\| = M$ and $\|\mathbf{k}_{\mathscr{L}}^*\| = N$.

As an example, when $\mathbf{k}^* = \langle 2, 1, 1, 1\rangle$ and $c_1 > c_2 > r \geq c_3 > c_4$, the recursive scheme is illustrated in Figure 2. In particular, in this case $M = 3$ and $N = 2$. Each vector $\mathbf{k}$ in the figure represents a linear combination of order statistics, and the iteration parameters are $n_g = n_g(\mathbf{k}) = \|\mathbf{k}_{\mathscr{G}}\|$ and $n_l = n_l(\mathbf{k}) = \|\mathbf{k}_{\mathscr{L}}\|$. Each vector also has a corresponding Θ value. From the initial conditions for Θ, we see that all vectors in the first row have Θ value 0, while all vectors in the first column have Θ value 1.

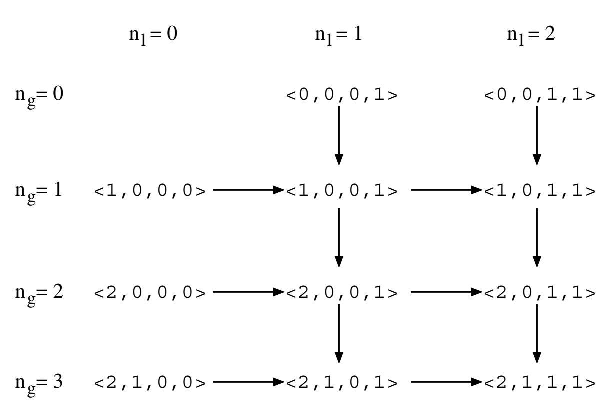

Figure 2 **Recursion for** $\mathbf{k}^* = \langle 2, 1, 1, 1\rangle$

For general $\mathbf{k}^*$, the recursion proceeds either by column or by row depending on the relative values of M and N. Thus only one vector of dimension $M+1$ or $N+1$ is necessary to implement the algorithm, and so it has $O(\min\{M, N\}+1)$ storage requirements. Furthermore, in terms of number of operations, to calculate a value $\Theta(r, \mathbf{k})$ requires two multiplications. We suggest that the recursion should be carried out for n_g increasing from 1 to M for a given value of n_l when $M \leq N$, while it should be carried out for n_l increasing from 1 to N for a given value of n_g when $N \leq M$ (as in Figure 2). Since in either case we clearly have a total of $O(2MN)$ multiplications, the above suggestion is based on storage considerations.

A direct implementation of the algorithm of Weisberg (1971) requires $O(2M+2N)$ amount of storage and $O(S_g[(\sum_{j=1}^{S}(k_j^*)^2)/2+[(M+N)/2]+S(M+N)])$ number of operations, where S is the cardinality of $\mathscr{C}$ and S_g is the cardinality of $\mathscr{G}$.

Let us compare the method of Weisberg with the algorithm introduced in this paper for a symmetric vector $\mathbf{k}^* = \langle k, k, \ldots, k\rangle$. If half of the c_j lie above r, then $S_g = S/2$ and so $M = N = Sk/2$. In this case a direct implementation of the algorithm in Weisberg (1971) has $O(S^2k^2/4+S^2k/4+S^3k/2)$ number of operations and $O(2Sk)$ storage requirements, while the algorithm introduced in this paper has $O(S^2k^2/2)$ number of operations and $O(Sk/2+1)$ storage requirements. If only one c_j lies above r, then $S_g = 1$ and so $M = k$ and $N = (S-1)k$. In this case the method of Weisberg has $O(Sk^2/2+Sk/2+S^2k)$ number of operations and $O(2Sk)$ storage requirements, while the new algorithm has $O(2(S-1)k^2)$ number of operations and only $O(k+1)$ storage requirements.

Although the algorithm introduced in this paper always requires less storage than the method of Weisberg, it often requires a greater number of operations. However, this new algorithm only involves multiplication and addition of real numbers between 0 and 1, and as such does not have numerical problems. On the other hand, as shown in the examples below, the algorithm in Weisberg (1971) can be numerically unstable and produce incorrect results.

Table 1 $P[G(\mathbf{k}^*) > r]$ **for** $\mathscr{C} = \{10, 7, 5, 3, 0\}$ **and** $\mathbf{k}^* = \langle 10, 15, 10, 10, 6\rangle$

r	Weisberg	New Algorithm
1.5	7541.31249802409	1
2.0	15.8842468261718	1
2.5	1.33160880208015	0.999999999997047
3.0	1.02277084691741	0.99999999560917
3.5	1.0000717345141	0.999998459166096
4.0	0.999835132474087	0.999832806426978
4.5	0.993687361245975	0.993687357886866
5.0	0.914963241542523	0.914963241541414
5.5	0.585808192636392	0.58580819263635
6.0	0.166874001328608	0.166874001328606

4. Examples

In order to evaluate the accuracy of the new algorithm, we first consider the example that was presented in (Weisberg 1971). In this example $G = 3U_{(10)}+2U_{(25)}+2U_{(35)}+3U_{(45)}$ with $n=50$, and so $\mathscr{C} = \{10, 7, 5, 3, 0\}$ and $\mathbf{k}^* = \langle 10, 15, 10, 10, 6\rangle$. Table 1 lists the results obtained from the two algorithms and shows that they agree for large r. However, the algorithm of Weisberg can become numerically unstable in some cases, since the subtraction of numbers close to each other can lead to overflow problems. This can be observed for small r, with better behavior as r increases.

Table 2 shows another example, this time with input parameters $\mathscr{C} = \{10, 9.5, 5, 4, 3, 0\}$ and $\mathbf{k}^* = \langle 10, 15, 10, 10, 10, 6\rangle$ (i.e., $G = .5U_{(10)}+4.5U_{(25)}+U_{(35)}+U_{(45)}+3U_{(55)}$ with $n=60$). The $\mathscr{C}$ parameters have been chosen to be closer together, which should produce larger intermediate values in the calculations with the Weisberg algorithm than in the previous example.

Table 2 $P[G(\mathbf{k}^*) > r]$ **for** $\mathscr{C} = \{10, 9.5, 5, 4, 3, 0\}$ **and** $\mathbf{k}^* = \langle 10, 15, 10, 10, 10, 6\rangle$

r	Weisberg	New Algorithm
2	−1.88894659314788e+22	1
3	−9.22337203685478e+18	0.999999999993862
4	−1.40737488355328e+15	0.99999617120683
5	23622320128	0.985405358720513
6	2048	0.448627055861267
7	0.00628662109375	0.00631140143294818
8	1.35861439487517e−07	1.35861439586292e−07
9	1.04466981693112e−18	1.04466981693112e−18

Table 3 $P[G(\mathbf{k}^*) > r]$ **for** $\mathscr{C} = \{10, 9.5, 5, 4, 3, 0\}$ **and** $\mathbf{k}^* = \langle 50, 75, 50, 50, 50, 30\rangle$

r	Weisberg	New Algorithm
2	−3.39928315402731e+185	1
3	−2.35872651551346e+167	1
4	−3.49635837635391e+146	1
5	−2.00124365551735e+121	0.999999355320442
6	−7.0023111686498e+89	0.384244123695832
7	−3.28837868399453e+49	1.20833201712547e−08
8	−4.36557456851006e−11	6.05151575310456e−31
9	1.08174101937241e−85	1.0817410193724e−85

This causes numerical problems to be more likely, as is apparent from the table.

The results obtained for the algorithms when n is increased (the number of intervals is multiplied by 5) are shown in Table 3.

5. Summary

We have obtained a new recursion for calculating the distribution of a linear combination of selected uniform order statistics. The recursion is surprisingly simple, involves only sums and products of probabilities and is numerically robust. The results of this paper can be immediately applied when one needs only to calculate the linear combination distribution. The approach developed here also leads to an additional recursion for this distribution, which has similar numerical properties but is more computationally intensive (see Diniz 2000). However, it has been shown that the recursion from Diniz (2000) can be combined with the methodology of de Souza e Silva and Gail (1989) to obtain a new simple algorithm for calculating the distribution of cumulative rate reward over a finite observation period.

Acknowledgments

Morganna Carmem Diniz was supported in part by fellowships from CNPq and CAPES. Edmundo de Souza e Silva was supported in part by grants from CNPq/ProTeM, CNPq, PRONEX and FAPERJ.

References

Chernoff, H., J. L. Gastwirth, M. V. Johns. 1967. Asymptotic distribution of linear combinations of functions of order statistics with applications to estimations. *Ann. Math. Statist.* **38** 52–72.

D'Agostino, R. B., M. A. Stephens. 1986. *Goodness-of-Fit Techniques*. Marcel Dekker.

David, H. A. 1981. *Order Statistics*, 2nd Ed. John Wiley & Sons, New York.

de Souza e Silva, E., H. R. Gail. 1989. Calculating availability and performability measures of repairable computer systems using randomization. *Journal of the ACM* **36** 171–193.

de Souza e Silva, E., H. R. Gail. 1998. An algorithm to calculate transient distributions of cumulative rate and impulse based reward. *Communications in Statistics—Stochastic Models* **14** 509–536.

Dempster, A., R. Kleyle. 1968. Distributions determined by cutting a simplex with hyperplanes. *Annals Math. Stat.* **39** 1473–1478.

Diniz, M. C. 2000. Solution techniques for models arising from multimedia networks (in Portuguese). Ph.D. Thesis—Federal University of Rio de Janeiro—COPPE/Sistemas, Brazil.

Eicker, F., M. L. Puri. 1976. On linear combinations of order statistics. S. Ikeda, T. Hayakawa, H. Hudimoto, M. Okamoto, M. Siotani, S. Yamamoto, eds. *Essays in Probability and Statistics*. Shinko Tsusho Co. Ltd., Tokyo, Japan. 433–449.

Grassmann, W. 1977a. Transient solutions in Markovian queueing systems. *Computers & Operations Research* **4** 47–53.

Grassmann, W. 1977b. Transient solutions in Markovian queues. *European Journal of Operational Research* **1** 396–402.

Matsunawa, T. 1985. The exact and aproximate distributions of linear combinations of selected order statistics from a uniform distribution. *Ann. Inst. Statist. Math.* **37** 1–16.

Meyer, J. 1980. On evaluating the performability of degradable computing systems. *IEEE Trans. on Computers* **C-29** 720–731.

Qureshi, M., W. Sanders. 1994. Reward model solution methods with impulse and rate rewards: An algorithm and numerical results. *Performance Evaluation* **20** 413–436.

Ramallingam, T. 1989. Symbolic computing the exact distributions of L-statistics from a uniform distribution. *Ann. Inst. Statist. Math.* **41** 677–681.

Ross, S. 1983. *Stochastic Processes*. John Wiley & Sons, New York.

Weisberg, H. 1971. The distribution of linear combinations of order statistics from the uniform distribution. *Annals Math. Stat.* **42** 704–709.

Accepted by Edward P.C. Kao; received August 2001; revised October 2001; accepted October 2001.