BIOMETRICS 50, 1042–1053
December 1994

Exact Permutational Tests for Group Sequential Clinical Trials

Cyrus R. Mehta,[1,2] Nitin Patel,[1,2] Pralay Senchaudhuri,[1,2] and Anastasios Tsiatis[1]

[1]Department of Biostatistics, Harvard School of Public Health

and

[2]Cytel Software Corporation, 675 Massachusetts Avenue, Cambridge, Massachusetts 02139, U.S.A.

SUMMARY

An efficient numerical algorithm is developed for computing stopping boundaries for group sequential clinical trials. Patients arrive in sequence, and are randomized to one of two treatments. The data are monitored at interim time points, with a fresh block of patients entering the study from one monitoring point to the next. The stopping boundaries are derived from the exact joint permutational distribution of the linear rank statistics observed across all the monitoring times. Specifically, the algorithm yields the exact boundary generating function,

$$\Pr(W_1 < b_1,\ W_2 < b_2,\ \ldots,\ W_{i-1} < b_{i-1},\ W_i = w_i),$$

where W_j is the linear rank statistic at the jth interim time point. The distribution theory is based on assigning ranks after pooling all the patients who have entered the study, and then permuting the patients to the two treatments independently within each block of newly arrived patients. The methods are applicable for an arbitrary number of monitoring times, which need not be specified at the start of the study. The data may be continuous or categorical, and censored or uncensored. The randomization rule for treatment allocation can be adaptive. The algorithm is especially useful during the early stages of a clinical trial, when very little data have been gathered, and stopping boundaries are based on the extreme tails of the relevant boundary generating function. In that case the corresponding large-sample theory is not very reliable. To illustrate the techniques we present a group sequential analysis of a recently completed study by the Eastern Cooperative Oncology Group.

1. Introduction

Suppose patients enter a two-arm randomized clinical trial in sequence. The objective is to determine whether one treatment is better than the other with respect to some response criterion. The study is designed to accrue patients until some predetermined maximum limit is reached. We wish to build in criteria for early stopping by computing exact permutational repeated significance tests on the steadily accruing data. We shall follow the group sequential framework of Pocock (1977). This means that the data are monitored at interim time points, rather than continuously, as the study progresses. Given the administrative complexity of modern clinical trials, this is a realistic assumption. Our methods are flexible as regards the frequency of the data monitoring. It is not necessary to specify in advance the maximum number of looks at the data, or the time interval between looks. These parameters can be determined adaptively as the study progresses through the "use function" approach of Lan and DeMets (1983).

Each time the data are monitored, treatment effectiveness is summarized in terms of a linear rank statistic, derived by pooling all the patients accrued so far, and ranking their responses. Exact distribution theory is developed by permuting these ranks between the two treatment arms. The permutations are enumerated in independent patient blocks, where each block contains the new patients who entered the study since the last time the data were monitored. Stopping boundaries for the linear rank statistics are then derived. It has been noted by Puri and Sen (1971), Schoenfeld and Tsiatis (1987), and O'Brien and Fleming (1987) that rank tests depending only on within-block ranks

Key words: Categorical data; Early stopping; Error spending functions; Exact tests; Group-sequential methods; Interim monitoring; Lan–DeMets procedure; O'Brien–Fleming boundary; Repeated significance tests; Sequential Wilcoxon test; Stopping rules.

can have reduced power. For this reason we pool all the patients, thereby utilizing interblock information, before assigning ranks. However, once the ranks have been assigned, the permutations are carried out independently in each block. This ensures protection against time trends. Schoenfeld and Tsiatis (1987) suggest that, in the absence of time trend, such tests, based on pooled ranks and intrablock permutations, are asymptotically efficient.

The main result in this paper is an efficient numerical algorithm to solve the permutation problem. In a group sequential setting, the permutational distribution from which stopping boundaries may be derived is multivariate, a formidable obstacle to exact inference. Generalizations of the network algorithms of Mehta, Patel, and Tsiatis (1984), and Mehta, Patel, and Wei (1988) were used to compute the relevant multivariate probabilities. The algorithm applies to continuous as well as ordered categorical outcomes. The methods are illustrated by an example of a clinical trial recently completed by the Eastern Cooperative Oncology Group.

Related to this paper is the work of Pawitan and Hallstrom (1990), and Lin, Wei, and DeMets (1991). Pawitan and Hallstrom reported using the joint permutation distribution of rank statistics to derive exact monitoring boundaries in the Cardiac Arrhythmia Suppression Trial, but do not describe their numerical algorithm. Lin et al. used a network algorithm to compute exact stopping boundaries for group sequential clinical trials with binary outcomes.

2. Mathematical Formulation

Assume that a randomized two-arm clinical trial has just been activated. Patients enter the study in sequence and are assigned to either treatment A or treatment B according to some (possibly restricted) randomization rule. The response variable is some quickly observable measure of treatment effectiveness such as tumor regression, or drug toxicity. (Extensions of this methodology to a delayed response like survival will be discussed in the last section of this paper.) The study has been planned so that at most t patients will be admitted. However, the data are monitored at interim time points and the study may be stopped before its accrual goals are met if one treatment appears to be significantly better than the other. The problem is to obtain criteria for early stopping which limit the Type I error rate to some prespecified level of significance. This problem is formulated more precisely below.

Data monitoring occurs at interim time points that may be regularly spaced or sporadic, at the convenience of the investigator. At each look, j, we accumulate a fresh block of t_j patients, n_j on treatment A, and $t_j - n_j$ on treatment B. Thus by the end of the ith look, the data may be represented in the form of i $2 \times t_j$ contingency tables, $j = 1, 2, \ldots, i$, as shown below:

Data at first look

	Patient response				
Treatment	z_{11}	z_{21}	$\cdots$	$z_{t_1 1}$	Total
A	x_{11}	x_{21}	$\cdots$	$x_{t_1 1}$	n_1
B	$1 - x_{11}$	$1 - x_{21}$	$\cdots$	$1 - x_{t_1 1}$	$t_1 - n_1$
Total	1	1	$\cdots$	1	t_1

Data at second look

	Patient response				
Treatment	z_{12}	z_{22}	$\cdots$	$z_{t_2 2}$	Total
A	x_{12}	x_{22}	$\cdots$	$x_{t_2 2}$	n_2
B	$1 - x_{12}$	$1 - x_{22}$	$\cdots$	$1 - x_{t_2 2}$	$t_2 - n_2$
Total	1	1	$\cdots$	1	t_2

$\vdots$

Data at ith *look*

	Patient response				
Treatment	z_{1i}	z_{2i}	$\cdots$	$z_{t_i i}$	Total
A	x_{1i}	x_{2i}	$\cdots$	$x_{t_i i}$	n_i
B	$1 - x_{1i}$	$1 - x_{2i}$	$\cdots$	$1 - x_{t_i i}$	$t_i - n_i$
Total	1	1	$\cdots$	1	t_i

In the above tables z_{lj} is the response observed for the lth patient in the jth block of data, and x_{lj} is 1 if that patient was assigned to treatment A, 0 otherwise. Let

$$\mathbf{x}_j \equiv (x_{1j}, x_{2j}, \ldots, x_{t_j j})'$$

denote the $(t_j \times 1)$ vector of treatment assignments for all the t_j patients in the jth block of data, and let

$$\mathbf{z}_j \equiv (z_{1j}, z_{2j}, \ldots, z_{t_j j})'$$

denote a $(t_j \times 1)$ vector of responses for these t_j patients. This notation does not preclude tied or ordered categorical responses, since the components of the $\mathbf{z}_j$ vector need not be distinct. In fact, the example in Section 4 deals with ordered categorical toxicity data.

We will base our inference on repeated linear rank tests which are designed to preserve some overall significance level. To do this we first convert all the $t_1 + t_2 + \cdots + t_i$ responses of the form $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_i$, that have been observed by the end of the ith look into corresponding ranks $\mathbf{r}_1^{(i)}, \mathbf{r}_2^{(i)}, \ldots, \mathbf{r}_i^{(i)}$, where

$$\mathbf{r}_j^{(i)} \equiv (r_{1j}^{(i)}, r_{2j}^{(i)}, \ldots, r_{t_j j}^{(i)})'$$

and $r_{lj}^{(i)}$ is the rank of the lth patient in the jth block of patients at the end of the ith look. The superscript (i) is necessary because the ranks are assigned among all the patients that have accrued by the ith look, and might be updated when the $(i + 1)$th block of patients accrue. For example, $r_{11}^{(1)}$ is the rank of the first patient among the first t_1 patients, whereas $r_{11}^{(2)}$ is the rank of that same patient among the first $t_1 + t_2$ patients. Note that $\mathbf{r}_j^{(i)}$ is undefined if $i < j$. We define, for all $j \leq i$,

$$\mathbf{R}_j^{(i)} \equiv (\mathbf{r}'^{(i)}_1, \mathbf{r}'^{(i)}_2, \ldots, \mathbf{r}'^{(i)}_j).$$

Now suppose we want to test the null hypothesis

$$H_0: \quad \text{treatments A and B are equivalent,}$$

against the one-sided alternative hypothesis

$$H_1: \quad \text{treatment A is better than treatment B.}$$

In order to test H_0 we compute a linear rank statistic each time we look at the data, using the current set of ranks. By the time of the ith look, a set of treatment allocations

$$\mathbf{X}_i \equiv (\mathbf{x}_1', \mathbf{x}_2', \ldots, \mathbf{x}_i')',$$

and a set of ranks, $\mathbf{R}_i^{(i)}$, are available so that we can compute the linear rank statistic

$$w_i = \mathbf{R}_i^{(i)} \mathbf{X}_i. \tag{2.1}$$

Let

$$\Gamma_i = \left\{ \mathbf{x}_i: \sum_{l=1}^{n_i} x_{li} = n_i, x_{li} \in \{0, 1\} \text{ for all } l \right\}$$

and form the product set

$$\Theta_i = \Gamma_1 \times \Gamma_2 \times \cdots \times \Gamma_i.$$

The permutation distribution of the random variable W_i is derived by fixing the responses, $\mathbf{r}_j^{(i)}$, at their observed values and considering all possible permutations of treatment allocations

$$\mathbf{X}_i \in \Theta_i.$$

Each such permutation of the treatment allocations yields a realization of W_i and has a probability that depends on the randomization scheme used to allocate patients to the two treatments. Under complete randomization the probability of each permutation is

$$\left[\prod_{j=1}^{i} \binom{t_j}{n_j} \right]^{-1}. \tag{2.2}$$

Equation (2.2) could be suitably modified as discussed in Mehta et al. (1988) if a restricted randomization scheme were used in place of complete randomization.

We summarize the accumulating data in terms of $w_1, w_2, \ldots, w_i$. The problem to be addressed is how to obtain stopping boundaries $b_1, b_2, \ldots, b_i$ so that if

$$w_1 < b_1, \quad w_2 < b_2, \quad \ldots, \quad w_{i-1} < b_{i-1}, \quad w_i \geq b_i,$$

we stop the study; otherwise we continue to the $(i + 1)$th look. The stopping boundaries must be selected in such a way that the above early stopping procedure preserves the prescribed Type I error rate, α, say. The first step is to decide how much of the total available Type I error rate to "use up" or "spend" during each repeated significance test. Slud and Wei (1982) proposed that the total error, α, be partitioned a priori into k quantities, $\alpha_1, \alpha_2, \ldots, \alpha_k$, where k is the maximum number of looks and $\Sigma_{j=1}^{k} \alpha_j = \alpha$. They would then compute stopping boundaries so that the probability of stopping at the ith look was α_i, and hence the probability of stopping by the kth look was $\Sigma_{j=1}^{k} \alpha_j$. Lan and DeMets (1983) suggested that the significance level be partitioned according to a function of the proportion of total information obtained. Such a function is called a "use function" or "error spending function." It is a monotone increasing function $\alpha(p)$, $0 \leq p \leq 1$, with $\alpha(0) = 0$ and $\alpha(1) = \alpha$. The value p denotes the proportion of the total information that has so far accumulated in the clinical trial. In the present context this information is equivalent to the proportion of the total sample size, t, that has accrued so far. In the Lan and DeMets (1983) approach, the probability of stopping at the ith look need not be specified a priori as in Slud and Wei (1982), but rather is equal to $\alpha(p_i) - \alpha(p_{i-1})$, where

$$p_i = (t_1 + t_2 + \cdots + t_i)/t,$$

and $\alpha(p_i)$ is the probability of stopping by the ith look. Another advantage of this method is that one need not specify k, the maximum number of interim looks, although one would have to specify t, the maximum amount of information.

To be specific, we look at the data for the first time after t_1 patients have accrued. At this stage we are now allowed to use up $\alpha(p_1)$ of the total available Type I error rate. To do so we stop the study by rejecting H_0 if the observed $w_1 \geq b_1$, where the stopping boundary, b_1, at the first look must satisfy

$$\Pr(W_1 \geq b_1) = q_1 \leq \alpha(p_1)$$

and, due to the discreteness of the distribution of W_1, q_1 is as close as we can possibly get to $\alpha(p_1)$ without exceeding $\alpha(p_1)$. If $w_1 < b_1$, we do not stop at the first interim test point. Then the data are monitored a second time after the next t_2 patients enter the study. At this stage we are allowed to use up $\alpha(p_2)$ of the total error. We stop the study, rejecting H_0, if the observed $w_2 \geq b_2$, where the second stopping boundary, b_2, is such that

$$q_1 + \Pr(W_1 < b_1, W_2 \geq b_2) = q_2 \leq \alpha(p_2),$$

and again, due to discreteness, q_2 is as close as we can get to $\alpha(p_2)$ without exceeding it. In general, the ith stopping boundary, b_i, is obtained by solving

$$q_{i-1} + \Pr(W_1 < b_1, \ldots, W_{i-1} < b_{i-1}, W_i \geq b_i) = q_i \leq \alpha(p_i), \tag{2.3}$$

where q_i is as close as we can get to $\alpha(p_i)$ without exceeding it.

If we do not stop the study at any of the interim looks, we will have to stop it at the terminal look, k, say, after all t of the subjects that we planned to accrue have entered the study. We will reject H_0 if the observed $w_k \geq b_k$, where b_k is such that

$$q_{k-1} + \Pr(W_1 < b_1, \ldots, W_{k-1} < b_{k-1}, W_k \geq b_k) = q_k \leq \alpha, \tag{2.4}$$

and q_k is as close as we can get to α without exceeding it. If the observed $w_k < b_k$, we will accept H_0.

The above procedure guarantees an overall significance level of at most α since the probability of crossing one of the boundaries under the null hypothesis is

$$q_1 + (q_2 - q_1) + \cdots + (q_k - q_{k-1}) = q_k \leq \alpha.$$

Notice that we have provided full flexibility in the choice of k and the monitoring points $p_1, p_2, \ldots, p_k$. These quantities need not be prespecified. They may be selected adaptively as the trial

progresses, at the convenience of the trial administrator. However, to achieve this flexibility we must specify a use function, $\alpha(p)$. Following Lan and DeMets (1983), many use functions have been proposed; see also Kim and DeMets (1987). Two popular use functions, corresponding respectively to the Pocock and O'Brien–Fleming stopping boundaries, are

$$\alpha(p) = \alpha \ln(1 + ep - p) \tag{2.5}$$

and

$$\alpha(p) = 2 - 2\Phi\left\{\frac{\Phi^{-1}(1 - \alpha/2)}{\sqrt{p}}\right\}, \tag{2.6}$$

where $\Phi(\cdot)$ is the right tail of the standard normal density function. These use functions were chosen to capture the spirit of the Pocock and O'Brien–Fleming stopping boundaries. If large-sample rather than exact distribution theory were applied, the Pocock use function would correspond to stopping the trial when the standardized test statistic exceeded some constant. Similarly, the O'Brien–Fleming use function would correspond to stopping the trial when the unstandardized accumulating statistic, which behaves like Brownian motion, exceeded some constant. The numerical example in Section 4 computes stopping boundaries based on the O'Brien–Fleming use function.

Exact joint permutational probabilities for correlated random variables, such as those appearing in (2.3), are ordinarily intractable. The next section describes a numerical algorithm for obtaining them, and the corresponding stopping boundaries, efficiently.

3. Network Algorithm

Suppose we are currently at the ith look, having previously computed the stopping boundaries b_1, $b_2, \ldots, b_{i-1}$, and used up q_{i-1} of the prescribed significance level. In order to compute b_i we need the *boundary generating function*

$$f_i(w_i) = \Pr(W_1 < b_1, \ldots, W_{i-1} < b_{i-1}, W_i = w_i)$$

for all values of w_i at which $f_i(w_i) > 0$. It is then possible to find the largest b_i that satisfies equation (2.3). The function f_i and its domain are obtained by stagewise processing of a series of i networks. These networks are defined next.

3.1 *Network Representation of Reference Sets*

When we look at the data for the ith time, we have i reference sets, $\Gamma_1, \Gamma_2, \ldots, \Gamma_i$. It is convenient to represent each of them as a network of nodes and arcs. The network representation for Γ_j is formed in $t_j + 1$ stages labelled $0, 1, \ldots, t_j$. At stage l it consists of a set of nodes of the form (l, s_{lj}), where node (a_j, b_j) implies that of the first a_j subjects within the jth group, exactly b_j have been assigned to treatment A. The network begins with a single node $(0, 0)$. For $l = 0, 1, \ldots, t_j - 1$, each node (l, s_{lj}) generates successor nodes, $(l + 1, s_{l+1,j})$, say, in the range

$$\max(s_{lj}, n_j - t_j + l + 1) \leq s_{l+1,j} \leq \min(n_j, s_{lj} + 1). \tag{3.7}$$

As a consequence of (3.7), all these sequences of nodes will automatically end with a single terminal node (t_j, n_j) at stage t_j. Each node is connected to its successors by arcs. A path through the network is a sequence of connected arcs of the form

$$(0, 0) \rightarrow (1, s_{1j}) \rightarrow \cdots \rightarrow (t_j, n_j).$$

Each path corresponds to a unique $\mathbf{x}_j \in \Gamma_j$, where $x_{lj} = s_{lj} - s_{l-1,j}$. Hereafter we use $s_{lj} - s_{l-1,j}$ and x_{lj} interchangeably, recognizing the implicit relationship

$$s_{lj} = x_{1j} + x_{2j} + \cdots + x_{lj}.$$

Let the length of the arc connecting node $(l - 1, s_{l-1,j})$ to node (l, s_{lj}) be the $(i - j + 1) \times 1$ vector $\mathbf{a}_{lj}x_{lj}$, where

$$\mathbf{a}_{lj} \equiv (r_{lj}^{(j)}, r_{lj}^{(j+1)}, \ldots, r_{lj}^{(i)})'.$$

The length of any sequence of connected arcs is defined to be the sum of lengths of the individual arcs in the sequence. Thus the length of any path, $\mathbf{x}_j \in \Gamma_j$, is the $(i - j + 1) \times 1$ vector

$$\sum_{l=1}^{t_j} \mathbf{a}_{lj} x_{lj}. \tag{3.8}$$

Although we have defined the above network abstractly, one can think of Γ_j as the set of all possible ways that the jth block of patients could be assigned to the two treatments, conditional on there being exactly n_j assignments to treatment A and $t_j - n_j$ assignments to treatment B. Each path, $\mathbf{x}_j \in \Gamma_j$, represents one way of assigning the treatments to the jth block of patients. The $(i - j + 1) \times 1$ path length (3.8) then represents the contribution of that path to the test statistics $w_j, w_{j+1}, \ldots, w_i$. The first element of this vector is the contribution made by the jth block of patients to w_j, the next element is the contribution of the same block of patients to w_{j+1}, and so on.

One may interpret the set Θ_i as an extended network in which the smaller networks $\Gamma_1, \Gamma_2, \ldots, \Gamma_i$, are positioned one after another in series. A path through the extended network corresponds to a unique $\mathbf{X}_i \in \Theta_i$.

3.2 *Forward Processing of the Networks at Look i*

The i networks that were constructed at look i are processed, one by one, in such a way that by the time the terminal node (t_i, n_i) of the ith network is reached, we obtain the boundary generating function, $f_i(w_i)$, and the corresponding boundary value, b_i. It suffices to specify the algorithm for processing Γ_j, the jth of the i networks. By applying this algorithm repeatedly at $j = 1, 2, \ldots, i$, we can process all the i networks.

Suppose that we have already processed the first $(j - 1)$ networks and have thereby enumerated all the paths

$$\Theta_{j-1}^* = \{\mathbf{X}_{j-1} \in \Theta_{j-1}: w_1 < b_1, w_2 < b_2, \ldots, w_{j-1} < b_{j-1}\}. \tag{3.9}$$

For each $\mathbf{X}_{j-1} \in \Theta_{j-1}^*$ there corresponds a (not necessarily unique) value of the $(i - j + 1) \times 1$ vector

$$\mathbf{u} = (\mathbf{R}_{j-1}^{(j)} \mathbf{X}_{j-1}, \mathbf{R}_{j-1}^{(j+1)} \mathbf{X}_{j-1}, \ldots, \mathbf{R}_{j-1}^{(i)} \mathbf{X}_{j-1})'. \tag{3.10}$$

Now define

$$c(\mathbf{u}) = \left\{ \begin{matrix} \text{the count of the number of distinct paths,} \\ \mathbf{X}_{j-1} \in \Theta_{j-1}^*, \text{ yielding the same } \mathbf{u}. \end{matrix} \right\} \tag{3.11}$$

The algorithm for processing the jth network begins at node $(0, 0)$ of Γ_j, and assumes that the set of records

$$\Omega_j(0, 0) = \{(\mathbf{u}, c(\mathbf{u})): \mathbf{X}_{j-1} \in \Theta_{j-1}^*\}$$

were obtained in the course of processing the previous $j - 1$ networks. It then carries these records through the nodes of Γ_j, stage by stage, in such a way that we eventually end up at its terminal node, (t_j, n_j), with an updated set of records, $\Omega_j(t_j, n_j)$, containing all the information needed to process the $(j + 1)$th network, or, if $j = i$, to extract the boundary generating function $f_i(w_i)$.

Processing the intermediate stages of each network. We define the algorithm for processing Γ_j recursively. Suppose we have reached stage l of this network and have obtained a set of records, $\Omega_j(l, s_{lj})$, at each of the nodes of stage l. The following five-step algorithm is used to update these records and thereby move forward to stage $l + 1$, for $l = 0, 2, \ldots, t_j$.

Step 1: Select a record $(\mathbf{u}, c(\mathbf{u})) \in \Omega_j(l, s_{lj})$.

Step 2: Transmit a copy of this record to each successor node $(l + 1, s_{l+1,j})$, where the successors are identified by (3.7).

Step 3: At each successor node, $(l + 1, s_{l+1,j})$, transform the transmitted record to $(\mathbf{u}_0, c_0)$, where $\mathbf{u}_0 = \mathbf{u} + \mathbf{a}_{l+1,j} x_{l+1,j}$ and $c_0 = c(\mathbf{u})$.

Step 4: Insert $(\mathbf{u}_0, c_0)$ into $\Omega_j(l + 1, s_{l+1,j})$ as follows:

1. If there already exists a record $(\mathbf{u}, c(\mathbf{u})) \in \Omega_j(l + 1, s_{l+1,j})$ such that $\mathbf{u} = \mathbf{u}_0$, then merge the two records by replacing $(\mathbf{u}, c(\mathbf{u}))$ with $(\mathbf{u}, c(\mathbf{u}) + c_0) \in \Omega_j(l + 1, s_{l+1,j})$.
2. If no record currently in $\Omega_j(l + 1, s_{l+1,j})$ has $\mathbf{u} = \mathbf{u}_0$, then augment $\Omega_j(l + 1, s_{l+1,j})$ by adding $(\mathbf{u}_0, c_0)$ to it, as a new record.

The technique of hashing (Sedgewick, 1983, p. 201) is used to search for matches and either merge or augment records in $\Omega_j(l + 1, s_{l+1,j})$. This ensures an optimum trade-off between efficient use of available memory and fast search.

Step 5: Return to step 1.

The above five-step algorithm continues until every record in $\Omega_j(l, s_{lj})$ has been processed. Then another node at stage l is selected, and all its records are processed in accordance with the above five steps. When all nodes at stage l have been exhausted, repeat steps 1–5 for stage $l + 1$.

Starting with $\Omega_j(0, 0)$ and moving through stages $0, 1, \ldots, t_j - 1$ by repeatedly carrying out steps 1–5, we process the entire Γ_j network, ending up at its terminal node with the set of records $\Omega_j(t_j, n_j)$. Note that each record, $(\mathbf{u}, c(\mathbf{u})) \in \Omega_j(t_j, n_j)$, has been transformed by its passage through the Γ_j network, and no longer satisfies equations (3.10) and (3.11). Instead,

$$\mathbf{u} = (\mathbf{R}_j^{(j)}\mathbf{X}_j, \mathbf{R}_j^{(j+1)}\mathbf{X}_j, \ldots, \mathbf{R}_j^{(i)}\mathbf{X}_j)', \tag{3.12}$$

where

$$\mathbf{X}_j \equiv \begin{pmatrix} \mathbf{X}_{j-1} \\ \mathbf{x}_j \end{pmatrix}$$

is such that $\mathbf{X}_j \in \Theta_j$, $\mathbf{X}_{j-1} \in \Theta_{j-1}^*$, and

$$c(\mathbf{u}) = \left\{ \begin{array}{l} \text{the count of the number of distinct paths, } \mathbf{X}_j \in \Theta_j, \\ \text{that give rise to the same } \mathbf{u}\text{, while preserving the} \\ \text{boundary condition, } \mathbf{X}_{j-1} \in \Theta_{j-1}^*. \end{array} \right\}. \tag{3.13}$$

Also, by (2.1), the first element of each record, $\mathbf{R}_j^{(j)}\mathbf{X}_j$, is a permutational value of the jth linear rank statistic, w_j.

Processing the last stage of each network. Some final processing of the records, $\Omega(t_j, n_j)$, is required at the terminal node, (t_j, n_j), of Γ_j. The nature of this processing depends on whether $j < i$ or $j = i$. The two cases are discussed separately below.

Case $j < i$: In this case the set of records, $\Omega_j(t_j, n_j)$, must be suitably updated so as to form the starting set of records, $\Omega_{j+1}(0, 0)$, for the processing of Γ_{j+1}. There are two parts to this update. First, eliminate those records of $\Omega_j(t_j, n_j)$ in which the first element, $\mathbf{R}_j^{(j)}\mathbf{X}_j \geq b_j$. It is clear that such records cannot contribute to the boundary generating function since $\mathbf{R}_j^{(j)}\mathbf{X}_j = w_j$ and the sample space of $f_i(w_i)$ excludes the event $w_j \geq b_j$. Once this is accomplished, the first element of each record in the reduced set serves no further purpose and may be discarded, thereby decreasing the dimensionality of that record by 1. This reduced set of records, each of lower dimensionality than the original records in $\Omega_j(t_j, n_j)$, constitutes the starting set of records for processing Γ_{j+1}. These records are of the form

$$\Omega_{j+1}(0, 0) = \{(\mathbf{u}, c(\mathbf{u})): \mathbf{X}_j \in \Theta_j^*\},$$

where Θ_j^*, $\mathbf{u}$, and $c(\mathbf{u})$ are defined by equations (3.9), (3.10), and (3.11), respectively, with j being replaced throughout by $j + 1$. The same algorithm can now be invoked for processing Γ_{j+1}.

Case $j = i$: In this case the records in $\Omega_i(t_i, n_i)$ are automatically of the form $(w_i, c(w_i))$, and $c(w_i)$ is the boundary generating function, $f_i(w_i)$, up to a normalizing constant. The normalizing constant is the count of all possible paths through the sequence of i networks, which is easily seen to be the reciprocal of (2.2). One may now use (2.3) to compute b_i. This completes the forward processing of the networks at the ith look.

Initiating the network algorithm. Finally, to complete the recursive description of the network algorithm, we need to specify the starting set of records at node (0, 0) of Γ_1. We start with the singleton set, $\Omega_1(0, 0) = \{(\mathbf{0}, 1)\}$, where the $\mathbf{0}$ is an $(i \times 1)$ vector of zeros, and its count is $c(\mathbf{0}) = 1$. We then move through all the i networks in sequence as described above.

3.3 *Early Elimination of Records*

Consider the record $(\mathbf{u}_0, \mathbf{c}_0)$ just prior to its insertion into $\Omega_j(l + 1, s_{l+1,j})$ at step 4 of the five-step algorithm. It would be advantageous to eliminate this record instead of carrying it along since that would reduce the number of records to be processed in the future. The question is, can this record be dropped without affecting the final boundary generating function, $f_i(w_i)$? From the previous section's discussion of how to process the last stage of each network, it is clear that if the record is carried along, its updated version, $(\mathbf{u}, c(\mathbf{u})) \in \Omega_g(t_g, n_g)$, will be dropped provided

$$\mathbf{R}_g^{(g)}\mathbf{X}_g \geq b_g, \tag{3.14}$$

for *any* $g = j, j + 1, \ldots, i - 1$. We now show how to determine in advance whether (3.14) holds.

Let $\Gamma_j(l, s_{lj})$ be the subset of all the paths of the Γ_j network that originate at node (l, s_{lj}) and terminate at node (t_j, n_j). (In this notation $\Gamma_j \equiv \Gamma_j(0, 0)$.) Now define, for $g = j, j + 1, \ldots, i$,

$$SP_j^{(g)}(l, s_{lj}) = \min\{r_{lj}^{(g)}x_{lj} + r_{l+1,j}^{(g)}x_{l+1,j} + \cdots + r_{t_j j}^{(g)}x_{t_j j}\}, \tag{3.15}$$

where the minimum is taken over all the paths in $\Gamma_j(l, s_{lj})$. Finally denote the components of the $(i - j + 1) \times 1$ vector, $\mathbf{u}_0$, created at step 3 of the five-step algorithm, by

$$\mathbf{u}_0 \equiv (u_0^{(j)}, u_0^{(j+1)}, \ldots, u_0^{(i)}).$$

We now have the following theorem.

Theorem 1. The record, $(\mathbf{u}_0, \mathbf{c}_0)$, created at step 3 of the five-step algorithm can be dropped without affecting $f_i(w_i)$ provided, for at least one $g \in \{j, j + 1, \ldots, i - 1\}$,

$$u_0^{(g)} + SP_j^{(g)}(l + 1, s_{l+1,j}) + SP_{j+1}^{(g)}(0, 0) + \cdots + SP_g^{(g)}(0, 0) \geq b_g. \tag{3.16}$$

The proof is straightforward. If (3.16) holds for any g, every path $X_g \in \Theta_g$ that passes through the node $(l + 1, s_{l+1,j})$ must be such that $\mathbf{R}_g^{(g)}\mathbf{X}_g \geq b_g$. Thus these paths cannot belong to Θ_g^* and cannot, in consequence, belong to the sample space of $f_i(w_i)$.

In order to use Theorem 1, we need to compute (3.15). This can be done very easily by use of the following theorem.

Theorem 2. The minimum of a weighted sum, $\Sigma_{i=1}^t a_i x_i$, subject to $x_i \in \{0, 1\}$, and $\Sigma_{i=1}^t x_i = n$, is $a_{(1)} + a_{(2)} + \cdots + a_{(n)}$, where $a_{(1)} \leq a_{(2)} \leq \cdots \leq a_{(t)}$ are the order statistics of the a's.

The proof is by contradiction.

3.4 *Why the Network Algorithm Is Efficient*

The network algorithm just described is considerably more efficient than simply enumerating all the $\mathbf{X}_i \in \Theta_i$ exhaustively. There are three reasons for this efficiency gain: clubbing, record elimination, and direct derivation of the boundary generating function.

Clubbing. Clubbing is the merging of two records, as described at step 4.1 of the five-step algorithm for updating records. The more often records merge with other records, the fewer records remain for processing at later stages. We refer to the merging of two records as *clubbing*. If there were no clubbing whatsoever, the number of records to be processed would grow exponentially and the algorithm would soon become infeasible. The extent of the clubbing depends on the choice of the $r_{lj}^{(i)}$ scores. Considerable clubbing occurs with Wilcoxon scores because these scores can be integerized by doubling. One can show that the number of operations required to process the entire network is $O(t_1^3 \times t_2^3 \times \cdots \times t_i^3)$. Thus even without the additional efficiency gains described below, the algorithm is polynomial in the number of observations, for group sequential Wilcoxon tests. On the other hand, the log-rank scores, being sums of reciprocals, cannot be easily integerized. Therefore clubbing occurs less frequently and the algorithm is no longer polynomial. One way to make the algorithm polynomial is to truncate the log-rank scores to a fixed number of decimal digits (say 2 or 3). The error introduced by such truncation can be corrected most efficiently by Monte Carlo sampling techniques. This idea has been researched in the doctoral dissertation of one of the authors. [See Senchaudhuri, Mehta, and Patel (1995).]

Record elimination. Efficiency gains are possible by eliminating records entirely, provided they satisfy the *SP* condition described in the previous section. Since records have a tendency to propagate and multiply, this type of early record elimination is very advantageous. This is especially true when the scores are unequally spaced and do not result in much clubbing.

Direct derivation of the boundary generating function. The network algorithm circumvents the need to generate the full multivariate distribution of $(W_1, W_2, \ldots, W_i)$ and then extract the boundary generating function, $f_i(w_i)$, from it. Generating a full multivariate distribution is a very computationally-intensive problem, and would consume excessive amounts of computer memory. Notice however, from (2.1) and (3.10), that only the partial distribution of $(W_j, W_{j+1}, \ldots, W_i)$ is being carried through the Γ_j network. When we move to Γ_{j+1}, the W_j component is dropped since we have used it by then to satisfy the condition, $W_j < b_j$. Thus we never have to carry along a full multivariate distribution in i dimensions.

4. Application to a Clinical Trial

Protocol EST 2289, conducted by the Eastern Cooperative Oncology Group (ECOG) in 1985–1986, was a study of 4-Deoxydoxorubicin versus Acivicin in patients with primary liver cancer. (See Cnaan and Mullin, Technical Report 57, Dana-Farber Cancer Institute, Boston, 1989.) In all, 75 patients were admitted to the trial. The study showed, among other things, that 4-Deoxydoxorubicin produces significantly more hematologic toxicity than Acivicin. Although this study was not designed for early stopping based on excessive toxicity, we will use it as an illustrative example of our group sequential methods. Suppose we wish to limit the Type I error rate to .05, and plan to inspect the data at interim time points, stopping the study early if it appears that 4-Deoxydoxorubicin has excessive hematologic toxicity relative to Acivicin. A one-sided sequential test is appropriate because 4-Deoxydoxorubicin was expected to be myelosuppressive.

The data were monitored at three interim time points, after 30, 43, and 57 patients had entered the study, respectively. An interim report was produced at each of the three interim points. The data on hematologic toxicity shown below are extracted from those three reports:

Data at first look

	Patient toxicity				
Treatment	Acceptable	Severe	Life-threat	Lethal	Total
4-Deoxy.	6	7	1	0	14
Acivicin	15	1	0	0	16
Total	21	8	1	0	30

Additional data at second look

	Patient toxicity				
Treatment	Acceptable	Severe	Life-threat	Lethal	Total
4-Deoxy.	2	5	0	0	7
Acivicin	6	0	0	0	6
Total	8	5	0	0	13

Additional data at third look

	Patient toxicity				
Treatment	Acceptable	Severe	Life-threat	Lethal	Total
4-Deoxy.	6	1	0	1	8
Acivicin	6	0	0	0	6
Total	12	1	0	1	14

A final report was produced after all 75 patients had entered the study. The combined hematologic toxicities of all 75 patients are tabulated below:

Final count of hematologic toxicity

	Patient toxicity				
Treatment	Acceptable	Severe	Life-threat	Lethal	Total
4-Deoxy.	22	13	3	1	39
Acivicin	34	2	0	0	36
Total	56	15	3	1	75

It is appropriate to analyze these data with the Wilcoxon rank sum test (using midranks to accommodate the ties). An analysis of the final toxicity data by StatXact (1989) reveals that 4-Deoxydoxorubicin is significantly more toxic than Acivicin (P-value = .0014). It would be interesting to see whether an exact group sequential approach would have picked up this effect at one of the interim looks. Suppose we are interested in limiting the overall Type I error rate to .05

but wish to spend that error over four looks, in accordance with the O'Brien–Fleming use function (2.6). The error available for spending, the error actually spent, and the stopping boundary at each look are tabulated below. The exact computations in this table consumed less than one CPU minute on a Gateway 2000 80486/33C personal computer.

Look i	Information p_i	Error available $\alpha(p_i)$	Error actually spent (q_i)		Stopping boundaries (b_i)	
			Exact	Asymptotic	Exact boundaries	Asymptotic boundaries
1	30/75	.0019	.00014	.0031	289.0	272.6
2	43/75	.0093	.0091	.0104	546.0	542.0
3	57/75	.0240	.0203	.0212	947.5	938.9
4	75/75	.0500	.0392	.0389	1,611	1,606

In the above table, the asymptotic stopping boundaries were derived from the large-sample joint distribution of $\{W_1, W_2, \ldots, W_i\}$. This distribution is multivariate normal with

$$\operatorname{cov}(W_i, W_j) = \sum_{k=1}^{\min(i,j)} \mathbf{r}'_k(i)\operatorname{var}(\mathbf{x}_k)\mathbf{r}_k^{(j)}, \tag{4.17}$$

where the variance matrix of $\mathbf{x}_k$ is easily obtained from standard multivariate hypergeometric theory. The error available for spending at each look, $\alpha(p_i)$, is computed from the O'Brien–Fleming use function (2.6). The error actually spent at each look, q_i, is obtained by substituting the appropriate stopping boundaries into equation (2.3). Due to the discreteness of the test statistic, the error actually spent does not equal the error available for spending. However the exact method is conservative and guarantees that the error actually spent will never exceed the error available for spending. The asymptotic method provides no such assurance, and indeed at the very first look the error actually spent (.0031) is 50% greater than the error available for spending (.0019). This translates into a substantially smaller asymptotic stopping boundary (272.6) compared to the exact stopping boundary (289). Now the Wilcoxon rank sum statistic actually observed at the first look is

$$w_1 = 11 \times 6 + 25.5 \times 7 + 30 \times 1 = 274.5.$$

Since this value exceeds 272.6, the study would be prematurely terminated at the first look, if the decision were to be based on the asymptotic rather than the exact stopping boundary. This disregards the spirit of the O'Brien–Fleming strategy, under which one is required to be extremely conservative about stopping during the early stages of a trial.

At the second look the error spent by the asymptotic method again exceeds the error available for spending. But this time the discrepancy between the asymptotic and exact errors spent is less. Also, the Wilcoxon rank sum statistic observed at the second look is $w_2 = 595$, exceeding the exact stopping boundary of 546. Thus the study would have been terminated at the second look under sequential monitoring.

The discrepancy between the errors spent by the asymptotic and exact methods diminishes further at subsequent looks because one is no longer operating so far out in the tails of the boundary generating function. In fact, by the end of the study, the two methods have spent almost the same amount of error.

5. Miscellaneous Remarks

The network algorithm is polynomial in the number of observations, as long as the rank scores are truncated to a fixed number of decimal digits. Note however that the algorithm is exponential in the number of looks. It is interesting also to observe that the memory requirements of the algorithm display similar characteristics. Fortunately the cost of memory has plummeted as newer and more powerful personal computers come on the market.

If the randomization rule for treatment allocation is adaptive, the above large-sample theory does not apply, but the exact algorithm can still be used, by suitably modifying equation (2.2).

The exact algorithm is particularly useful at the early looks, when very little data have been gathered and stopping boundaries are based on the extreme tails of the boundary generating function. In this case asymptotic theory might not be very accurate. We have just seen in the previous example that the asymptotic stopping boundaries led to inappropriate early stopping at the very first look.

The exact algorithm can be readily adapted to two-sided tests, since the output from each look is the entire boundary generating function.

The exact algorithm was described in the context of an instantaneous response variable. However, the same algorithm goes through for time-to-failure data with possible censoring. Now the original data consist of pairs of observations (a survival time and a censoring indicator) for each subject. These bivariate raw data scores can be readily converted into log-rank scores, generalized Wilcoxon scores, or other univariate rank scores that incorporate the censoring, using the methods discussed in Prentice (1978), or Prentice and Marek (1979). Thereafter the permutational problem is the same. A further modification is necessary for computing the available information at a given look. The information is needed as input to the use functions (2.5) and (2.6). For instantaneous response the information is defined as the proportion of the total sample size that has accrued so far. However, for censored survival data the information is related to the number of failure events (deaths). Thus, letting d_i denote the number of deaths at the ith look, the information is specified by

$$p_i = \sum_{j=1}^{i} d_j/d,$$

where d is the maximum number of deaths needed for the trial to stop.

Acknowledgements

The authors thank David Harrington for kindly making his Pascal program "ONESIDE" available for performing the multivariate normal integrations in Section 4. They also thank two referees for their careful reading of the manuscript, and for their useful suggestion that the exact and asymptotic stopping boundaries be compared. This research was supported in part by Grants CA-33019 and CA-51962 from the National Cancer Institute, and Grant AI-31789 from NIAID.

Résumé

Nous présentons un algorithme numérique permettant de déterminer les règles d'arrêt dans le cas d'essais cliniques séquentiels. On suppose que les patients entrent les uns à la suite des autres dans l'essai, où l'un des deux traitements leur est alloué de manière randomisée. Les données sont évaluées à des temps intermédiaires, temps entre lesquels s'ajoutent à chaque fois une série supplémentaire de nouveaux patients. Les valeurs limites liées aux règles d'arrêt sont obtenues à partir de la distribution conjointe exacte des permutations de la statistique de rang observée sur l'ensemble de ces temps intermédiaires. Plus précisément l'algorithme donne la fonction exacte générant ces valeurs limites,

$$\Pr(W_1 < b_1,\ W_2 < b_2,\ \ldots,\ W_{i-1} < b_{i-1},\ W_i = w_i),$$

où W_j est la statistique de rang au temps intermédiaire j. La distribution théorique de cette statistique s'obtient en attribuant des rangs à tous les patients déjà inclus dans l'essai, puis en permutant leur appartenance aux deux groupes de traitement, au sein de chaque série de patients comprise entre deux temps intermédiaires. Cette méthode peut s'appliquer à un nombre arbitraire de temps intermédiaires non nécessairement défini préalablement à l'essai, à des données continues ou qualitatives, censurées ou non, à une randomisation adaptative ou non. Cet algorithme est particulièrement intéressant au tout début d'un essai clinique, lorsque peu de données sont encore disponibles et que les valeurs limites sont obtenues à partir des queues de distribution, situation dans laquelle la théorie asymptotique est peu fiable. A titre d'illustration nous présentons l'analyse séquentielle d'un essai récemment effectué par l'Eastern Cooperative Oncology Group.

References

Kim, K. K. and DeMets, D. L. (1987). Design and analysis of group sequential tests based on the Type I error spending rate function. *Biometrika* **74,** 149–154.

Lan, G. and DeMets, D. L. (1983). Discrete sequential boundaries for clinical trials. *Biometrika* **70,** 659–663.

Lin, D. Y., Wei, L. J., and DeMets, D. L. (1991). Exact statistical inference for group sequential trials. *Biometrics* **47,** 1399–1408.

Mehta, C. R., Patel, N. R., and Tsiatis, A. A. (1984). Exact significance testing with ordered categorical data. *Biometrics* **40,** 819–825.

Mehta, C. R., Patel, N. R., and Wei, L. J. (1988). Constructing exact significance tests with restricted randomization rules. *Biometrika* **75,** 295–302.

O'Brien, P. C. and Fleming, T. R. (1979). A multiple testing procedure for clinical trials. *Biometrics* **35,** 549–556.

O'Brien, P. C. and Fleming, T. R. (1987). A paired Prentice–Wilcoxon test for censored survival data. *Biometrics* **43,** 169–180.

Pawitan, Y. and Hallstrom, A. (1990). Statistical interim monitoring of the cardiac arrhythmia suppression trial. *Statistics in Medicine* **9,** 1081–1090.

Pocock, S. J. (1977). Group sequential methods in the design and analysis of clinical trials. *Biometrika* **64,** 191–199.

Prentice, R. L. (1978). Linear rank tests with right-censored data. *Biometrika* **65,** 167–179.

Prentice, R. L. and Marek, P. (1979). A qualitative discrepancy between censored data rank tests. *Biometrics* **35,** 861–867.

Puri, M. L. and Sen, P. K. (1971). *Nonparametric Methods in Multivariate Analysis*. New York: Wiley.

Schoenfeld, D. A. and Tsiatis, A. A. (1987). A modified log rank test for highly stratified data. *Biometrika* **74,** 167–175.

Sedgewick, R. (1983). *Algorithms*. Reading, Massachusetts: Addison-Wesley.

Senchaudhuri, P., Mehta, C. R., and Patel, N. R. (1995). Estimating exact *p*-values by the method of control variates, or Monte Carlo rescue. *Journal of the American Statistical Association* **90,** in press.

Slud, E. and Wei, L. J. (1982). Two-sample repeated significance tests based on the modified Wilcoxon statistic. *Journal of the American Statistical Association* **77,** 862–868.

StatXact (1989). *Statistical Software for Exact Nonparametric Inference.* Cambridge, Massachusetts: Cytel Software Corporation.

Received August 1991; *revised January* 1993; *accepted May* 1993.