A Network Algorithm for Performing Fisher's Exact Test in $r \times c$ Contingency Tables

CYRUS R. MEHTA and NITIN R. PATEL*

An exact test of significance of the hypothesis that the row and column effects are independent in an $r \times c$ contingency table can be executed in principle by generalizing Fisher's exact treatment of the 2×2 contingency table. Each table in a conditional reference set of $r \times c$ tables with fixed marginal sums is assigned a generalized hypergeometric probability. The significance level is then computed by summing the probabilities of all tables that are no larger (on the probability scale) than the observed table. However, the computational effort required to generate all $r \times c$ contingency tables with fixed marginal sums severely limits the use of Fisher's exact test. A novel technique that considerably extends the bounds of computational feasibility of the exact test is proposed here. The problem is transformed into one of identifying all paths through a directed acyclic network that equal or exceed a fixed length. Some interesting new optimization theorems are developed in the process. The numerical results reveal that for sparse contingency tables Fisher's exact test and Pearson's χ^2 test frequently lead to contradictory inferences concerning row and column independence.

KEY WORDS: Fisher's exact test; $r \times c$ contingency tables; Conditional reference set; Network algorithm; Contingency tables; Implicit enumeration; Permutation distribution; Exact tests.

1. INTRODUCTION

Fisher's exact treatment of the 2×2 contingency table readily generalizes to an exact test of row and column independence in $r \times c$ contingency tables. One would in general prefer to report the exact p value associated with an observed $r \times c$ table especially when the entries in each cell are small. (See, for example, Cochran 1954). It is more common, however, to report the tail area of the χ^2 distribution based on Pearson's X^2 statistic, because of the formidable computational effort that is needed to perform the exact test of significance. One may get some idea of the complexity of this numerical problem from the work of Gail and Mantel (1977).

In a recent paper, Mehta and Patel (1980) developed an algorithm for computing the exact significance level in $2 \times c$ contingency tables. Subsequently, Pagano and Halvorsen (1981) provided an algorithm for the more general $r \times c$ problem. The present article provides an alternative algorithm for the exact treatment of $r \times c$ contingency tables which considerably extends the bounds of computational feasibility relative to the algorithm provided by Pagano and Halvorsen. Many $r \times c$ tables that are computationally feasible with our algorithm, but infeasible without it, are still small enough to justify an exact computation rather than a χ^2 approximation.

The network approach that forms the basis of our algorithm can be adapted to many other exact permutation tests.

2. FORMULATION AS A NETWORK PROBLEM

Given a $r \times c$ contingency table, $\mathbf{X}$, let x_{ij} denote the entry in row i and column j. Let $R_i = \sum_{j=1}^{c} x_{ij}$ be the sum of all entries in row i and let $C_j = \sum_{i=1}^{r} x_{ij}$ be the sum of all entries in column j. Throughout we will assume that the x_{ij}'s and their various partial sums are nonnegative integers. Denote by $\mathcal{T}$ the reference set of all possible $r \times c$ contingency tables with the same marginal totals as $\mathbf{X}$. Thus

$$\mathcal{T} = \left\{ \mathbf{Y}: \mathbf{Y} \text{ is } r \times c, \sum_{j=1}^{c} y_{ij} = R_i, \quad \sum_{i=1}^{r} y_{ij} = C_j, \right\}.$$

Under the null hypothesis of row and column independence the probability of observing any $\mathbf{Y} \in \mathcal{T}$ can be expressed as a product of multinomial coefficients

$$P(\mathbf{Y}) = \left(\prod_{j=1}^{c} \frac{C_j!}{y_{1j}!\, y_{2j}! \cdots y_{rj}!} \right) \Big/ \frac{T!}{R_1!\, R_2! \cdots R_r!},$$

where $T = \sum_{i=1}^{r} R_i$. For later use we define $D = T!/R_1!\, R_2! \cdots R_r!$.

The exact significance level or p value associated with the observed table $\mathbf{X}$ is defined as the sum of the probabilities of all the tables in $\mathcal{T}$ that are no more likely than

* Cyrus R. Mehta is Assistant Professor of Biostatistics, Harvard School of Public Health, and Dana Farber Cancer Institute, 44 Binney Street, Boston, MA 02115. Nitin R. Patel is Professor, Indian Institute of Management, Vastrapur, Ahmedabad 380-015, India. The authors thank Professor Marvin Zelen for initially stimulating their interest in these problems. The availability of the DASH library, developed by the DASH Software Development Group, Division of Biostatistics and Epidemiology, Dana Farber Cancer Institute, Boston, made it possible for them to access the algorithm of Pagano and Halvorsen. The authors also thank Mrs. Rita Rama Rao for progamming assistance. This investigation was supported in part by Grants CA-33019 and CA-23415 from the National Cancer Institute.

June 1983, Volume 78, Number 382
Theory and Methods Section

X. Specifically,

$$p = \sum_{Y \in \mathcal{S}} P(\mathbf{Y}),$$

where $\mathcal{S} = \{\mathbf{Y}: \mathbf{Y} \in \mathcal{T} \text{ and } P(\mathbf{Y}) \leq P(\mathbf{X})\}$.

The problem is to evaluate p. Our approach is to construct a directed acycle network of nodes and arcs such that the set of all distinct paths from the initial node to the terminal node is isomorphic with the set $\mathcal{T}$. Moreover, by construction, the length of the path that corresponds to the table $\mathbf{Y} \in \mathcal{T}$ equals $D \cdot P(\mathbf{Y})$. The original problem is now equivalent to identifying and summing the lengths of all paths which are no longer than $D \cdot P(\mathbf{X})$.

2.1 Construction of the Network

We construct the network in $c + 1$ stages labeled successively $c, c - 1, \ldots, 0$. At any stage k there exist a set of nodes each labeled by a unique vector $(k, R_{1k}, \ldots, R_{rk})$. Arcs emanate from each node at stage k and every arc is directed to exactly one node at stage $k - 1$. The network is defined recursively by specifying all the nodes of the form $(k - 1, R_{1,k-1}, \ldots, R_{r,k-1})$ that succeed the node $(k, R_{1k}, \ldots, R_{rk})$ and are connected to it by arcs. The range of $R_{i,k-1}$, $i = 1, 2, \ldots, r$ for these successor nodes is given by

$$\max\left(0, R_{ik} - C_k + \sum_{l=1}^{i-1} (R_{lk} - R_{l,k-1})\right) \leq R_{i,k-1} \leq \min\left(R_{ik}, S_{k-1} - \sum_{l=1}^{i-1} R_{l,k-1}\right), \quad (2.1)$$

where $S_j = \sum_{l=1}^{j} C_l$ and we follow the convention that a summation is null if its lower limit exceeds its upper limit. There is only one node at stage c, the initial node, which is labeled $(c, R_{1c}, \ldots, R_{rc})$ where $R_{ic} \equiv R_i$, $i = 1, 2, \ldots, r$. By applying the recursion (2.1) successively at stages $c, c - 1, \ldots, 1$, we obtain exactly one node at stage 0, labeled $(0, 0, \ldots, 0)$. This is the terminal node.

The length of an arc from node $(k, R_{1k}, \ldots, R_{rk})$ to node $(k - 1, R_{1,k-1}, \ldots, R_{r,k-1})$ is equal to

$$C_k!/R_{1k} - R_{1,k-1})! \cdots (R_{rk} - R_{r,k-1})!.$$

A complete path through the network is defined as a succession of connected arcs directed from the initial node to the terminal node. The length of any path is the product of the arc lengths that constitute the path.

One can readily verify that for a network constructed as defined above each path of the form

$$(c, R_{1c}, \ldots, R_{rc}) \to (c - 1, R_{1,c-1}, \ldots, R_{r,c-1}) \to \cdots \to (0, 0, \ldots, 0),$$

corresponds to an $r \times c$ contingency table $\mathbf{Y} \in \mathcal{T}$ where $y_{ij} = R_{ij} - R_{i,j-1}$, $i = 1, 2, \ldots, r$, $j = 1, 2, \ldots, c$. Moreover, the length of the path equals $D \cdot P(\mathbf{Y})$. Suppose, for example, that $\mathcal{T}$ is the set of all 3×3 contingency tables with $R_1 = 2$, $R_2 = 1$, $R_3 = 6$ and $C_1 = 3$, $C_2 = 3$, $C_3 = 3$. The paths of the network in Figure 1 are in one-to-one correspondence with the tables in $\mathcal{T}$.

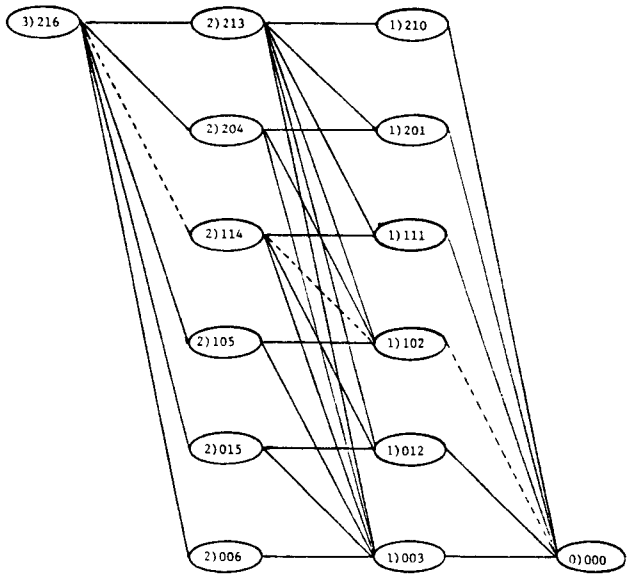

Figure 1. Network Representation for All the 3 × 3 Contingency Tables with $R_1 = 2$, $R_2 = 1$, $R_3 = 6$ and $C_1 = 3$, $C_2 = 3$, $C_3 = 3$

2.2 The Network Algorithm

Suppose that **X** is the observed $r \times c$ table. In network terms our goal is to identify and sum all paths whose lengths do not exceed $D \cdot P(\mathbf{X})$. If we systematically enumerate each path through the network, compute its length and sum all those path lengths that do not exceed $D \cdot P(\mathbf{X})$, we are in effect examining each $r \times c$ table in $\mathcal{T}$ individually. This is usually computationally infeasible. The advantage of the network representation is that it circumvents the need to explicitly enumerate each path. To see this, we define the following two items:

$\text{SP}(k, R_{1k}, \ldots, R_{rk})$ = the length of the shortest subpath from node $(k, R_{1k}, \ldots, R_{rk})$ to node $(0, 0, \ldots, 0)$.

$\text{LP}(k, R_{1k}, \ldots, R_{rk})$ = the length of the longest subpath from node $(k, R_{1k}, \ldots, R_{rk})$ to node $(0, 0, \ldots, 0)$.

Let $\mathcal{Q}$ denote all the distinct paths that share a common subpath $(c, R_{1c}, \ldots, R_{rc}) \to (c - 1, R_{1,c-1}, \ldots, R_{r,c-1} \to \cdots \to (k, R_{1k}, \ldots, R_{rk})$ through stages $c, c - 1, \ldots, k$ of the network. Let

$$\text{PAST} = \prod_{j=k+1}^{c} \{C_j!/(R_{1j} - R_{1,j-1})! \cdots (R_{rj} - R_{r,j-1})!\}$$

denote the length of the common subpath up to node $(k, R_{1k}, \ldots, R_{rk})$. Although the paths all diverge at this node, we can enumerate them implicitly if either one of the following two conditions holds:

$$\text{PAST} \cdot \text{LP}(k, R_{1k}, \ldots, R_{rk}) \leq D \cdot P(\mathbf{X}), \quad (2.$$

or

$$\text{PAST} \cdot \text{SP}(k, R_{1k}, \ldots, R_{rk}) > D \cdot P(\mathbf{X}). \quad (2.3)$$

If (2.2) is satisfied we know immediately that every path in $\mathfrak{D}$ must be no greater than $D \cdot P(\mathbf{X})$. Hence the lengths of all these paths contribute to the p value. But one can show (see the Appendix) that the sum of all the path lengths in $\mathfrak{D}$ equals

$$\text{PAST} \cdot (S_k!/R_{1k}! \cdots R_{rk}!) \quad (2.4)$$

so that explicit enumeration of all these paths is unnecessary. If (2.3) is satisfied we know immediately that every path in $\mathfrak{D}$ exceeds $D \cdot P(\mathbf{X})$. Hence none of these paths can contribute to the p value, and they are dropped from all further consideration. Explicit enumeration of the paths in $\mathfrak{D}$ is once again unnecessary. If neither (2.2) nor (2.3) is satisfied then of course we cannot implicitly enumerate all the paths in $\mathfrak{D}$. In that case we extend the common subpath to a node $(k-1, R_{1,k-1}, \ldots, R_{r,k-1})$ at stage $k-1$ in accordance with (2.1) and proceed to verify (2.2) and (2.3) in the same manner as before.

The only remaining problem is to compute SP and LP at each node. One approach is to use dynamic programming in a single backward pass through the network. This method was advocated by Mehta and Patel (1980) for $2 \times c$ contingency tables. For the general $r \times c$ problem, there may be too many nodes in the network, and this approach could soon become computationally infeasible. In the next section, we have exploited the special structure of this problem to obtain a closed form upper bound for LP and to show how a lower bound for SP can be easily computed by solving only a few triangular systems of linear equations. These bounds can be substituted in equations (2.2) and (2.3). They converge to the true values of LP and SP respectively if the column sums C_j are constant for all j.

3. THEOREMS FOR COMPUTING THE LONGEST AND SHORTEST PATHS FROM ANY NODE TO THE TERMINAL NODE

Let us assume initially that $C_j = N$ for all j. We can compute $\text{LP}(k, R_{1k}, \ldots, R_{rk})$ with the help of the following theorem.

Theorem 1. The optimal objective function value (OFV) for the problem

$$\text{P1:} \quad \text{maximize} \prod_{j=1}^{k} \left(\frac{N!}{y_{1j}! \cdots y_{rj}!} \right) \quad (3.1)$$

subject to

$$\sum_{j=1}^{k} y_{ij} = R_{ik}, \quad i = 1, 2, \ldots, r, \quad (3.2)$$

$$\sum_{i=1}^{r} y_{ij} = N, \quad j = 1, 2, \ldots, k, \quad (3.3)$$

$$y_{ij} \geq 0 \quad \text{and integer for all } i, j, \quad (3.4)$$

is given by

$$\left(\frac{N!}{d_1! \cdots d_r!} \right)^k \Big/ (d_1 + 1)^{h_1} (d_2 + 1)^{h_2} \cdots (d_r + 1)^{h_r}, \quad (3.5)$$

where $d_i = [R_{ik}/k]$ ($[x]$ denoting the largest integer less than or equal to x) and $h_i = R_{ik} - kd_i$.

Proof. Let $\mathbf{A} = \{a_{ij}, i = 1, 2, \ldots, r, j = 1, 2, \ldots, k\}$ be an optimal solution to P1. We first show that $|a_{ip} - a_{iq}| \leq 1$ for all i and $p \neq q$. If the contrary is true, we can find some i_o, p_o, q_o such that $a_{i_o p_o} \geq a_{i_o q_o} + 2$. It then follows from (3.3) that there exists some $t_o \neq i_o$ such that $a_{t_o p_o} + 1 \leq a_{t_o q_o}$. Consider the matrix $\mathbf{A}'$ where

$$a_{i_o p_o}' = a_{i_o p_o} - 1, \; a_{i_o q_o}' = a_{i_o q_o} + 1,$$

$$a_{t_o p_o}' = a_{t_o p_o} + 1, \; a_{t_o q_o}' = a_{t_o q_o} - 1$$

and $a_{ij}' = a_{ij}$ for all other i, j values. Clearly $\mathbf{A}'$ satisfies (3.2), (3.3), and (3.4). Let OF be the value of (3.1) under $\mathbf{A}$ and OF′ be the value of (3.1) under $\mathbf{A}'$. Then

$$\text{OF}'/\text{OF} = (a_{i_o p_o})(a_{t_o q_o})/(a_{i_o q_o} + 1)(a_{t_o p_o} + 1) > 1,$$

and $\mathbf{A}$ cannot be an optimal solution to P1. This conclusion contradicts the initial hypothesis; therefore, $|a_{ip} - a_{iq}| \leq 1$ for all i. It follows from (3.2) that exactly h_i entries in row i of $\mathbf{A}$ are equal to $d_i + 1$, the remaining entries being equal to d_i. Substituting these values into (3.1) and simplifying yields (3.5).

Expression (3.5) provides a closed form rapidly computable formula for LP from any node. Although no comparable closed form expression for SP is available, one can nevertheless compute it fairly easily. The next two theorems show how this is done. Before we develop the theorems we need the following definition.

Definition. Let $\mathbf{A}$ be any $r \times k$ matrix with entries a_{ij} in row i and column j. The matrix $\mathbf{A}$ is said to contain a *cycle* if there exist p distinct nonzero entries $a_{i_1 j_1}, a_{i_2 j_2}, \ldots, a_{i_p j_p}$ such that p is even and

$$j_1 = j_2, i_2 = i_3, j_3 = j_4, i_4 = i_5, \ldots, i_p = i_1.$$

The presence of a cycle in $\mathbf{A}$ implies that it is possible to start from a nonzero cell (i_1, j_1), and, by alternately moving vertically and horizontally to other nonzero cells, return to the starting cell.

Theorem 2. Let $\mathbf{A} = \{a_{ij}, i = 1, 2, \ldots, r, j = 1, 2, \ldots, k\}$ be the solution to the following problem.

P2: minimize (3.1) subject to (3.2), (3.3), and (3.4).

Then the nonzero entries of $\mathbf{A}$ cannot form a cycle.

Proof. Suppose that $\mathbf{A}$ minimizes (3.1) subject to (3.2), (3.3) and (3.4). Let $a_{i_1 j_1}, a_{i_2 j_1}, a_{i_2 j_2}, \ldots, a_{i_1 j_{p-1}}, a_{i_1 j_1}$ be a cycle of nonzero entries in $\mathbf{A}$. Consider solutions $\mathbf{A}'$ and $\mathbf{A}''$ to (3.2), (3.3) and (3.4) in which $a_{ij}' = a_{ij}'' = a_{ij}$ for all i, j not on the cycle but

$$a_{i_1 j_1}' = a_{i_1 j_1} + 1, \; a_{i_2 j_1}' = a_{i_2 j_1} - 1, \; a_{i_2 j_2}' = a_{i_2 j_2} + 1,$$

$$\ldots, \; a_{i_1 j_{p-1}}' = a_{i_1 j_{p-1}} - 1,$$

and

$$a_{i_1 j_1}'' = a_{i_1 j_1} - 1,\ a_{i_2 j_1}'' = a_{i_2 j_1} + 1,\ a_{i_2 j_2}'' = a_{i_2 j_2} - 1,$$

$$\ldots,\ a_{i_1 j_{p-1}}'' = a_{i_1 j_{p-1}} + 1.$$

Let OF, OF′, and OF″ be the values of (3.1) corresponding to **A**, **A**′, and **A**″. Then

$$\frac{OF'\ OF''}{OF^2} = \frac{a_{i_1 j_1}}{a_{i_1 j_1}+1} \cdot \frac{a_{i_2 j_1}}{a_{i_2 j_1}+1} \cdots \frac{a_{i_1 j_{p-1}}}{a_{i_1 j_{p-1}}+1} < 1.$$

So either OF′ < OF or < OF″ < OF and A cannot minimize (3.1). This contradiction proves the theorem. ...

It is well known (Hadley 1962, pp. 288-290) that since **A** contains no cycles it must be one of the basic feasible solutions of the constraints (3.2) and (3.3). This characterization enables us to minimize (3.1) by systematically examining all possible basic feasible solutions of (3.2) and (3.3). The next theorem can drastically reduce the number of basic feasible solutions to be examined.

Theorem 3. If $R_{1k} \geq N$, the minimum OFV of problem P2 is equal to the minimum OFV of the following problem:

$$\text{P3:}\quad \text{minimize} \prod_{j=1}^{k-1} \left(\frac{N!}{y_{1j}! \cdots y_{rj}!} \right), \tag{3.6}$$

subject to (3.4) and

$$\sum_{j=1}^{k-1} y_{ij} = R_{ik},\ i = 2, 3, \ldots, r, \tag{3.7}$$

$$\sum_{i=1}^{r} y_{ij} = N,\ j = 1, 2, \ldots, k-1, \tag{3.8}$$

$$\sum_{j=1}^{k-1} y_{1j} = R_{1k} - N. \tag{3.9}$$

Proof. Let $\{b_{ij},\ i = 1, 2, \ldots, r, j = 1, 2, \ldots, k-1\}$ be an optimal solution to P3. Let **B** be the augmented set $\{b_{ij},\ i = 1, 2, \ldots, r, j = 1, 2, \ldots, k\}$ in which $b_{1k} = N$ and $b_{ik} = 0,\ i \neq 1$. Then **B** is feasible for P2 and yields the same OFV as the minimum OFV for P3. This proves that the minimum OFV for P2 ≤ the minimum OFV for P3.

Next let $\mathbf{A} = \{a_{ij},\ i = 1, 2, \ldots, r, j = 1, 2, \ldots, k\}$ be an optimal solution to P2. We will presently show that $a_{1j_o} = N$ for some value of j_o. By relabeling columns so that $j_o = k$ and by invoking the constraint (3.3), we see that $a_{1k} = N$ and $a_{ik} = 0,\ i \neq 1$. Therefore, the restricted set $\{a_{ij},\ i = 1, 2, \ldots, r, j = 1, 2, \ldots, k-1\}$ is a feasible solution for P3 and yields the same OFV as the minimum OFV for P2. This proves that the minimum OFV for P3 ≤ the minimum OFV for P2.

To complete the proof of the theorem it only remains to show that there exists some j_o such that $a_{1j_o} = N$. Suppose that no such j_o exists. We can, without loss of generality, let

$$a_{11} \geq a_{12} \geq \cdots a_{1p} > 0 \quad \text{and} \quad a_{1,p+1} = a_{1,p+2} = \cdots = a_{1k} = 0,$$

where $1 \leq p \leq k$. Since $a_{11} < N$, there must be some i_o such that $a_{i_o 1} > 0$. Without loss of generality let $i_o = 2$. By Theorem 2, it follows that $a_{2j} = 0, j = 2, 3, \ldots, p$, since otherwise there would be a cycle in **A**. Also, since $a_{11} + a_{12} + \cdots + a_{1p} \geq N$ and $a_{11} + a_{21} \leq N$, there must be a q, $2 \leq q \leq p$, such that $a_{12} + a_{13} + \cdots + a_{1,q-1} < a_{21}$ and $a_{12} + a_{13} + \cdots + a_{1q} \geq a_{21}$. Let $a_{12} + a_{13} + \cdots + a_{1q} - a_{21} = \alpha \geq 0$. Now construct a feasible solution $\mathbf{A}' = \{a_{ij}',\ i = 1, 2, \ldots, r, j = 1, 2, \ldots, k\}$ in which

$$a_{11}' = a_{11} + a_{21},\ a_{12}' = 0,\ a_{13}' = 0, \ldots, a_{1,q-1}' = 0,$$

$$a_{1q}' = \alpha,$$

$$a_{21}' = 0,\ a_{22}' = a_{12},\ a_{23}' = a_{13}, \ldots, a_{2,q-1}' = a_{1,q-1},$$

$$a_{2q}' = a_{1q} - \alpha,$$

and $a_{ij}' = a_{ij}$ for all other i, j.

If OF is the value of (3.1) under **A** and OF′ is the value of (3.1) under **A**′, the ratio of OFV's is, upon simplification,

$$\frac{OF}{OF'} = \binom{a_{11} + a_{21}}{a_{11}} \Big/ \binom{a_{1q}}{\alpha}.$$

Considering the two possibilities $a_{11} \leq a_{21}$ and $a_{11} > a_{21}$ separately, one can show that OF > OF′. Thus $a_{1j} < N$ for all j is impossible for an optimal solution to P2.

This theorem is useful provided $k \geq r$. Notice that if $k \geq r$, at least one row, i_o say, has $R_{i_o k} \geq N$ since

$$\sum_{i=1}^{r} R_{ik}/r = k\,N/r \geq N.$$

We have, without any loss of generality, relabeled row i_o as row 1 in Theorem 3. The optimal solution to P3 is easier to obtain than the optimal solution to P2 since there are r fewer variables in P3.

We may apply Theorem 3 repeatedly, eliminating r variables from the resulting optimization problem each time, until there are no longer any row totals whose values exceed N. This leads to the following corollary.

Corollary. The minimum OFV of the problem P2 is equal to the minimum OFV of the problem

$$\text{P4:}\quad \text{minimize} \prod_{j=1}^{k'} \left(\frac{N!}{y_{1j}! \cdots y_{rj}!} \right) \tag{3.10}$$

subject to (3.4) and

$$\sum_{j=1}^{k'} y_{ij} = R_{ik} - \left[\frac{R_{ik}}{N} \right] \cdot N,\ i = 1, 2, \ldots, r, \tag{3.11}$$

$$\sum_{i=1}^{r} a_{ij} = N,\quad j = 1, 2, \ldots, k', \tag{3.12}$$

where

$$k' = k - \sum_{i=1}^{r} \left[\frac{R_{ik}}{N} \right].$$

We have reduced the problem of computing SP to one of enumerating all the basic feasible solutions of the constraints (3.11) and (3.12), and selecting one which minimizes (3.10). For $r \times c$ tables where $r \leq 5$ and c is arbitrary we can show that there cannot be more than 150 basic feasible solutions for the constraints (3.11) and (3.12), at any node in the network. Moreover, each basic feasible solution is easily obtained by solving a triangular system of linear equations.

Theorems 1, 2, and 3 enable us to compute LP and SP, provided the column totals are all equal to N. If this restriction is removed we can still use these theorems to obtain a lower bound for SP and an upper bound for LP.

Theorem 4. Let SP and LP be the minimum and maximum values, respectively, of

$$\mathrm{OF} = \prod_{j=1}^{k} (C_j!/y_{1j}! \cdots y_{rj}!) \tag{3.13}$$

where the y_{ij}'s are constrained by (3.4) and

$$\sum_{i=1}^{r} y_{ij} = C_j, j = 1, 2, \ldots, k, \tag{3.14}$$

$$\sum_{j=1}^{k} y_{ij} = R_{ik}, i = 1, 2, \ldots, r. \tag{3.15}$$

Let SP′ and LP′ be the minimum and maximum values, respectively, of

$$\mathrm{OF}' = \prod_{j=1}^{k} (N!/z_{1j}! \cdots z_{rj}! z_{r+1,j}!) \tag{3.16}$$

where the z_{ij}'s are constrained by (3.4) and

$$\sum_{i=1}^{r+1} z_{ij} = N, j = 1, 2, \ldots, k, \tag{3.17}$$

$$\sum_{j=1}^{k} z_{ij} = R_{ik}, i = 1, 2, \ldots, r, \tag{3.18}$$

$$\sum_{j=1}^{k} z_{r+1,j} = kN - \sum_{i=1}^{r} R_{ik}, \tag{3.19}$$

and $N = \max(C_1, C_2, \ldots, C_k)$. Then

$$\mathrm{SP} \geq \mathrm{SP}' \Big/ \prod_{j=1}^{k} \binom{N}{C_j} \tag{3.20}$$

and

$$\mathrm{LP} \leq \mathrm{LP}' \Big/ \prod_{j=1}^{k} \binom{N}{C_j}. \tag{3.21}$$

Proof. Let $\mathbf{A} = \{a_{ij}, i = 1, 2, \ldots, r, j = 1, 2, \ldots, k\}$ be a minimizing solution of (3.13) subject to (3.4), (3.14) and (3.15). Then the extension $\{a_{ij}, i = 1, 2, \ldots, r+1, j = 1, 2, \ldots, k\}$ where $a_{r+1,j} = N - C_j$ for all j satisfies the constraints (3.4), (3.17), (3.18), (3.19). The value of (3.16) with this solution simplifies to

$$\mathrm{SP} \prod_{j=1}^{k} \binom{N}{C_j}$$

Table 1. Computational Experience with the Network Algorithm on Problems of Varying Size

Problems	Contingency Table	p Value		CPU Time (minutes)		# of Tables in the Reference Set[b]
		Exact	χ^2-approx.	Network	Pagano/Halvorsen	
1	1 1 1 0 0 0 1 3 3 4 4 4 4 4 4 4 1 1	.0680	.0605	.02	.49	40,500
2	2 0 1 2 6 1 3 1 1 1 1 0 3 1 0 1 2 1 2 0	.0911	.0932	.24	35.67	1.1×10^6
3	2 0 1 2 6 5 1 3 1 1 1 2 1 0 3 1 0 0 1 2 1 2 0 0	.0454	.0666	1.15	Infeasible[a]	68×10^6
4	1 1 1 0 0 0 1 2 4 4 4 4 5 5 5 6 5 0 1 1 1 0 0 0 1 2 4	.0354	.0935	5.34	Infeasible[a]	624×10^6
5	1 2 2 1 1 0 2 0 0 2 3 0 0 1 1 1 2 7 1 1 2 0 0 0 0 1 1 1 1 0	.0258	.0771	3.04	Infeasible[a]	1.6×10^9
6	1 2 2 1 1 0 1 2 0 0 2 3 0 0 0 1 1 1 2 7 3 1 1 2 0 0 0 1 0 1 1 1 1 0 0	.0393	.1213	14.09	Infeasible[a]	64×10^9

[a] Failed to compute the exact p value within 180 CPU minutes.
[b] Estimated by the method of Gail and Mantel (1977).

so that

$$SP' \leq SP \prod_{j=1}^{k} \binom{N}{C_j}.$$

The proof for LP′ is similar.

Thus we can compute SP′ and LP′ with the help of Theorems 1, 2, and 3 and we can always bound SP and LP when the C_j's are not equal. The computations in the next section demonstrate that these bounds perform satisfactorily.

4. COMPUTATIONAL EXPERIENCE

Exact p values were computed by the Network algorithm for problems of varying size. The results are displayed in Table 1. Where possible the CPU minutes used by Network have been compared with the corresponding CPU minutes used by the algorithm of Pagano and Halvorsen (1981). Both algorithms have been programmed on a DEC-2060 computer. The algorithm developed by Pagano and Halvorsen is most appropriate for these comparisons because their approach already yields substantial improvements over previously published approaches to this problem. Table 1 shows that Network considerably extends the bounds of computational feasibility for exact tests of significance in $r \times c$ contingency tables. Sparse matrices with greater than 20 cells are infeasible using the algorithm of Pagano and Halvorsen (because the CPU times exceed 180 minutes) but are easily evaluated by Network. The last column of Table 1 provides an estimate of the number of tables in the reference set $\mathcal{T}$ corresponding to each problem. This estimate was computed by a technique due to Gail and Mantel (1977) and it explains why explicit enumeration soon becomes computationally infeasible. Note also that for many of these problems the exact and approximate p value would lead to quite different inferences about row and column independence.

Table 1 does not provide a good estimate of the CPU ratio for the two algorithms because most of the problems considered were too large to be handled by the Pagano-Halvorsen algorithm. Therefore, a number of smaller problems were also considered. Table 2 shows that the Network algorithm performs uniformly better for all these problems and the CPU ratio increases dramatically with the size of the problem.

Table 2. A Comparison of the Network and the Pagano/Halvorsen Algorithms for a Variety of r× c Contingency Tables

Size of r× c Table	CPU Seconds Network	CPU Seconds Pagano/Halvorsen	CPU Ratio
2 × 4	.32	.82	2.56
2 × 4	.27	.73	2.70
3 × 3	.32	1.05	3.28
3 × 3	.31	.81	2.61
2 × 5	.40	1.73	4.33
3 × 4	.36	1.25	3.47
3 × 4	.89	3.84	4.31
3 × 5	2.38	34.02	14.29
4 × 5	2.31	41.12	17.80
4 × 5	.35	4.39	12.54
4 × 5	6.10	1263.40	207.11
5 × 5	1.30	87.45	67.27
5 × 5	2.04	815.36	399.69
5 × 5	14.22	4910.74	345.34

5. DISCUSSION

We have demonstrated that an important but rather difficult computational problem in hypothesis testing can be successfully approached if it is reformulated in terms of enumerating paths through a network. The network has interesting structural properties that lead to a closed-form upper bound for LP, the longest path from any node to the terminal node. The corresponding lower bound for SP, the shortest path, is easily evaluated as a basic feasible solution to a set of constraints that arise more commonly in the transportation problem of linear programming.

The computational results displayed in Table 1 (especially problem 6) show that for sparse contingency tables, fairly large differences can exist between the p values generated by Fisher's exact test and by Pearson's χ^2 test. This was previously demonstrated for $r \times 1$ contingency tables with small expectations (Chapman 1976). Inferences based on the χ^2 test might, therefore, be misleading whereas absolute reliance can be placed on the significance level generated by Fisher's exact test. This does not imply that Fisher's is the only procedure yielding an exact p value. There are several alternative methods. In general, we may define a discrepancy measure $d: \mathcal{T} \to \mathcal{R}$ as a function that assigns a real number to each contingency table in the reference set $\mathcal{T}$. If $\mathbf{X}$ is the observed table, an exact test is defined by $p = \sum_{Y \in \mathcal{S}} P(\mathbf{Y})$ where $\mathcal{S} = \{\mathbf{Y}: \mathbf{Y} \in \mathcal{T} \text{ and } d(\mathbf{Y}) \geq d(\mathbf{X})\}$. For Fisher's exact test $d(\mathbf{X}) = 1/P(\mathbf{X})$. Two other commonly used discrepancy measures are Pearson's X^2 statistic and the likelihood ratio statistic, $-2 \log R$. (See, for example, Chapman 1976). For 2×2 contingency tables, the randomized version of Fisher's procedure yields a uniformly most powerful unbiased test (Lehmann 1959). However, for higher dimensional tables, no such optimal property has been demonstrated so that any of the above three discrepancy measures appears reasonable. The network algorithm adapts itself to all three discrepancy measures. In particular, equations (2.2), (2.3), and (2.4) are unchanged but different optimization problems must be solved in order to bound LP and SP for each discrepancy measure. Alternatively, dynamic programming in a single backward pass through the network is usually an efficient way to compute LP and SP exactly, regardless of the discrepancy measure. This technique will be feasible provided the number of nodes in the network is kept within reasonable bounds.

Finally, as the cell expectations become large, all three exact tests converge to the χ^2 test. Bishop, Fienberg, and Holland (1975, Ch. 14) showed this for Pearson's X^2 and the likelihood ratio, $-2 \log R$. The convergence of Fisher's exact test to the χ^2 can be demonstrated by gener-

alizing Van der Waerden's (1957, Sec. 49.7) results. Traditionally, Cochran's (1954) guidelines have been used to decide whether the χ^2 test will yield a reliable p value. It is suspected, however, that Cochran's guidelines are conservative (see, for example, Yarnold 1970). Therefore, it would be instructive to make an empirical comparison of the χ^2 and various exact tests along the lines of Haber (1980). Computational feasibility is no longer the major obstacle to such an investigation.

It is worth pointing out that the network algorithm can be applied to a variety of exact permutation tests. The basic approach of constructing a network of nodes and arcs and testing conditions of the form of (2.2), (2.3) at each node does not depend in any way on the special structure of the $r \times c$ contingency table. Analogous networks can be constructed for several problems of which the following are a representative subset: exact tests of hypothesis concerning the relative risk in k 2×2 contingency tables (Zelen 1971); exact permutation distributions for the Wilcoxon statistic and the logrank statistic; distribution free tests for one- and two-way layouts. These and other related problems will be investigated in future papers.

Many important and interesting details of implementation have not been discussed here in the interests of presenting only the main features of our algorithm. The following notable omissions will be discussed elsewhere:

1. the ability to trade off accuracy against CPU time by specifying in advance the number of decimals to which the p value should be correct,
2. the ability to collapse into a single node all nodes at stage k for which $(R_{1k}, R_{2k}, \ldots, R_{rk})$ are permutations of each other,
3. the use of hashing with linear probing (Knuth 1973) for the identification, storage, and retrieval of nodes that are permutations of each other,
4. the use of spanning trees (Murty 1976, pp. 289–310) to enumerate all basic feasible solutions,
5. the ability to collapse rows of the contingency table so as to limit the number of basic feasible solutions to at most 150 at each node.

APPENDIX

We wish to show that the sum of all path lengths from node $(k, R_{1k}, \ldots, R_{rk})$ to node $(0, 0, \ldots, 0)$ is given by $(C_1 + C_2 + \cdots + C_k)!/R_{1k}!\,R_{2k}! \cdots R_{rk}!$. To prove this result we use induction on k. For $k = 1$, this is clearly true. Suppose that the result is true for $k - 1$. We must then show that it is true for k. We can express the sum of all path lengths from $(k, R_{1k}, \ldots, R_{rk})$ to $(0, 0, \ldots, 0)$ as

$$\sum\left[\frac{C_k!}{y_{1k}!y_{2k}! \cdots y_{rk}!}\right] \times \left[\begin{array}{l}\text{Sum of all path lengths from} \\ (k-1, R_{1k} - y_{1k}, \ldots, R_{rk} - y_{rk}) \\ \text{to } (0, 0, \ldots, 0)\end{array}\right],$$

where we are summing over all $y_{1k} + y_{2k} + \cdots + y_{rk} = C_k$. By the induction hypothesis this expression is equal to

$$\sum\left[\frac{C_k!}{y_{1k}!y_{2k}! \cdots y_{rk}!}\right] \times \left[\frac{(C_1 + C_2 + \cdots + C_k)!}{(R_{1k} - y_{1k})!(R_{2k} - y_{2k})! \cdots (R_{rk} - y_{rk})!}\right]$$

$$= \frac{(C_1 + C_2 + \cdots + C_k)!}{R_{1k}!R_{2k}! \cdots R_{rk}!} \times \left\{\sum\binom{R_{1k}}{y_{1k}}\binom{R_{2k}}{y_{2k}} \cdots \binom{R_{rk}}{y_{rk}} \Big/ \binom{C_1 + C_2 + \cdots + C_k}{C_k}\right\}.$$

Notice that

$$\sum\binom{R_{1k}}{y_{1k}}\binom{R_{2k}}{y_{2k}} \cdots \binom{R_{rk}}{y_{rk}} = \binom{R_{1k} + R_{2k} + \cdots + R_{rk}}{C_k}.$$

To see this, compare the coefficient of t^{C_k} on both sides of the following identity:

$$(1 + t)^{R_{1k}}(1 + t)^{R_{2k}} \cdots (1 + t)^{R_{rk}} = (1 + t)^{R_{1k} + R_{2k} + \cdots + R_{rk}}.$$

Moreover, this network is so constructed that $R_{1k} + R_{2k} + \cdots + R_{rk} = C_1 + C_2 + \cdots C_k$. Therefore,

$$\sum\binom{R_{1k}}{y_{1k}}\binom{R_{2k}}{y_{2k}} \cdots \binom{R_{rk}}{y_{rk}} \Big/ \binom{C_1 + C_2 + \cdots C_k}{C_k} = 1,$$

and the desired result holds for k.

[*Received July 1981. Revised March 1982.*]

REFERENCES

BISHOP, Y.M.M., FIENBERG, S.E., and HOLLAND, P.W. (1975), *Discrete Multivariate Analysis: Theory and Practice*, M.I.T. Press.

CHAPMAN, J.W. (1976), "A Comparison of X^2, -2 Log R, and Multinomial Probability Criteria for Significance Tests when Expected Frequencies are Small," *Journal of the American Statistical Association*, 71, 854–863.

COCHRAN, W.G. (1954), "Some Methods for Strengthening the Common χ^2 Tests," *Biometrics*, 10, 417–451.

GAIL, M., and MANTEL, N. (1977), "Counting the Number of Contingency Tables with Fixed Margins," *Journal of the American Statistical Association*, 72, 859–862.

HABER, M. (1980), "A Comparison of Some Continuity Corrections for the Chi-Squared Test on 2×2 Tables," *Journal of the American Statistical Association*, 75, 510–515.

HADLEY, G. (1962), *Linear Programming*, Reading, Mass.: Addison-Wesley.

KNUTH, D. E. (1973), *The Art of Computer Programming, Vol. 3: Sorting and Searching*, Reading, Mass.: Addison-Wesley.

LEHMANN, E.L. (1959), *Testing Statistical Hypotheses*, New York: John Wiley.

MEHTA, C.R., and PATEL, N.R. (1980), "A Network Algorithm for the Exact Treatment of the $2 \times k$ Contingency Table," *Communications in Statistics*, Ser. B9(6), 649–664.

MURTY, K.G. (1976), *Linear and Combinatorial Programming*, New York: John Wiley.

PAGANO, M., and HALVORSEN, K. (1981), "An Algorithm for Finding the Exact Significance Levels of $r \times c$ Contingency Tables," *Journal of the American Statistical Association*, 76, 931–934.

VAN DER WAERDEN, B. (1957), *Mathematische Statistik*, Heidelberg: Springer Verlag.

YARNOLD, J.K. (1970), "The Minimum Expectation in χ^2 Goodness of Fit Tests and the Accuracy of Approximations for the Null Distributions," *Journal of the American Statistical Association*, 65, 864–886.

ZELEN, M. (1971), "The Analysis of Several 2×2 Contingency Tables," *Biometrika*, 58, 129–137.