December, 1966

OPTIMAL GROUP TESTING

Milton Sobel*

Technical Report No. 83

University of Minnesota

Minneapolis, Minnesota

*This research was supported in part by the National Science Foundation under Grant No. GP-3813 at the University of Minnesota, and in part by the Office of Naval Research under contract NONR-225(53)(NR-042-002) at Stanford University.

OPTIMAL GROUP TESTING

By

Milton Sobel
University of Minnesota

1. Introduction

The problem of group-testing is concerned with the classification of each of a finite number N of given units into one of two distinct categories which we call satisfactory and unsatisfactory (or simply, good and defective). The characteristic feature of the testing procedure is that any number $x(1 \leq x \leq N)$ of units can be tested simultaneously with two possible results: (i) either all x are good or (ii) at least one of the x is defective; in the latter case it is not known which ones or how many of the x units are defective. The model considered is that the N units are realizations of N independent and identically distributed Bernoulli chance variables with common, known probability q of a unit being good and $p = 1-q$ of a unit being defective. The problem is to devise a scheme which minimizes the expected number of tests required to classify each of the N units as good or defective; the scheme can be sequential in the sense that the present test can depend on the results of previous tests.

Different procedures have already been considered for this problem with $N < \infty$ by Dorfman [1], Sterrett [12], Sobel and Groll [7], Sobel [8], and Finucan [2]. Several procedures were considered in [7] and [8] but special emphasis in [7] was given to a procedure called R_1 and in [8] to a procedure called R_0. It was pointed out in [7] that the procedure R_1 has some optimal properties but that for q close to unity it is not optimal. In [8], the procedure R_0 was developed and it follows from the construction that it is as good or better than R_1 for all values of $q(0 < q < 1)$. In this paper we show that the procedure R_0 has additional optimal properties but that it also is not optimal. Necessary properties for an optimal procedure are used in this paper to develop another procedure R_{00} which is as good as or better than R_0 (and hence also as good as or better than R_1) for all values of q.

The procedure R_{00} coincides with both R_1 and R_0 for $q \leq (\sqrt{5}-1)/2 = .618...$ and tests units one at-a-time for any finite N; in this range of q, this simple rule is known to be optimal [13]. In constructing R_{00} a special effort was made for each N (see Tables A1 and A2) to make it optimal for q arbitrarily close to one; it is conjectured that R_{00} is optimal for all q and all integers N of the form $3 \cdot 2^{b-2} \leq N \leq 2^b$ for some integer $b \geq 0$.

The procedure R_{00} is explicitly worked out for $N = 1$ through 8 for all values of q and for N larger than 8 it is defined implicitly by recursion formulas. In addition two lower bounds, each of which holds for any group-testing procedure, are developed in [7] and [8] so that meaningful numerical comparisons of the expected numbers of tests $E\{T|R\}$, for $R = R_0, R_1, R_{00}$ and the lower bounds can be made.

Based on the fact that R_{00} is either optimal or very close to optimal, we define the percentage efficiency of procedures R_1 and R_0 by

$$(1.1) \qquad PE(R_i|q) = \frac{E\{T|R_{00}\}}{E\{T|R_i\}} 100 \qquad (i = 0, 1).$$

We find that the efficiencies of R_1 and R_0 are quite good and better than those based on using either lower bound in the numerator of (1.1). The numerical computations (see Table 1) for $N \leq 8$ indicate that R_1 and R_0 have a minimum efficiency over all q of at least 95% and 99.9%, respectively. From the fact that $E\{T|R\}$ increases with N for each of the above procedures and gets close to the lower bounds (see Table 2), there is good reason to suspect that these lower bounds on the efficiency hold for all values of N as well as for all values of q.

Two closely related problems to the one treated here are (i) the case of finite N and unknown q treated in [10], and (ii) the case of known q and infinite N treated in [8]; the latter case will also be considered in this paper. A somewhat different problem is obtained by assuming that the number of defectives D is known at the onset; for this hypergeometric problem optimal solutions are obtained for $D = 1$ in [6] and [14] and for $D \geq 2$ in [11].

TABLE 1

Comparison of the Expected Number of Tests for Procedures R_{00}, R_0, R_1 for selected values of n and q.

		q = .75		q = .90		q = .95		q = .99	
		E(T)	Efficiency*	E(T)	Efficiency*	E(T)	Efficiency*	E(T)	Efficiency*
n=3	R_{00}	2.51563		1.62700		1.30713		1.06030	
	R_0	2.51563	100.00	1.62700	100.00	1.30700	100.00	1.06030	100.00
	R_1	2.51563	100.00	1.66100	97.95	1.34013	97.54	1.06960	99.13
n=4	R_{00}	3.33203		2.01700		1.50463		1.10020	
	R_0	3.33203	100.00	2.01890	99.91	1.50487	99.98	1.10020	100.00
	R_1	3.33203	100.00	2.05100	98.34	1.53763	97.85	1.10950	99.16
n=5	R_{00}	4.15723		2.44868		1.71354		1.14059	
	R_0	4.15723	100.00	2.44887	99.99	1.71356	100.00	1.14059	100.00
	R_1	4.15723	100.00	2.48951	98.36	1.77122	96.74	1.15881	98.43
n=6	R_{00}	4.97583		2.90873		1.95500		1.18236	
	R_0	4.97583	100.00	2.91046	99.94	1.95536	99.98	1.18244	99.99
	R_1	4.97583	100.00	2.94341	98.82	2.00921	97.30	1.20831	97.85
n=7	R_{00}	5.79938		3.37973		2.19085		1.23177	
	R_0	5.79938	100.00	3.38146	99.95	2.19088	100.00	1.23177	100.00
	R_1	5.79938	100.00	3.41441	98.98	2.25184	97.29	1.25802	97.91
n=8	R_{00}	6.61922		3.86572		2.43862		1.28167	
	R_0	6.61922	100.00	3.86620	99.99	2.43904	99.98	1.28167	100.00
	R_1	6.61922	100.00	3.90441	99.01	2.49934	97.57	1.30792	97.99

*Efficiency is defined in equation (1.1).

Table 2A

Expected Number of Tests for Procedures R_1, R_0 and R_{00} and Lower Bounds for any Group Testing Procedures Starting with a Binomial Set of Size N.

$q = .90$

Initial Number of Units N	Expected Number of Tests under Procedure R_1 $H(N;q,R_1)$	Procedure R_0 $H(N;q,R_0)$	Procedure R_{00} $H(N;q,R_{00})$	Huffman Lower Bound# $HB(N,q)$	Information Lower Bound* $IB(N,q)$
1	1.000	1.000	1.000	1.000	0.469
2	1.290	1.290	1.290	1.290	0.938
3	1.661	1.627	1.627	1.598	1.407
4	2.051	2.019	2.017	1.970	1.876
5	2.490	2.449	2.449	2.401	2.345
6	2.943	2.910	2.909	2.825	2.814
7	3.414	3.381	3.380	3.320	3.283
8	3.904	3.866	3.866	3.806	3.752
9	4.395	4.356	4.356	4.275	4.221
10	4.872	4.834	4.833	4.767	4.690
11	5.327	5.288	5.287	5.206	5.159
12	5.790	5.755	5.753	5.640	5.628
13	6.261	6.226	6.224	6.128	6.097
14	6.732	6.697	6.696	6.607	6.566
15	7.213	7.176	7.175	7.085	7.035
16	7.695	7.657	7.656	7.547	7.504
17	8.161	8.123	8.123	8.012	7.973
18	8.629	8.593	8.592	8.458	8.442
19	9.100	9.064	9.063	8.940	8.911
20	9.572	9.536	9.535	9.420	9.380
21	10.044	10.008	10.007	9.891	9.849
22	10.520	10.483	10.483	10.357	10.318
23	10.996	10.959	10.958	10.812	10.787
24	11.466	11.430	11.429	11.274	11.256
25	11.937	11.901	11.900	11.752	11.725
30	14.301	14.265	14.264	14.091	14.070
35	16.661	16.625	16.624	16.492	16.415
40	19.024	18.988	18.987	18.790	18.760
45	21.387	21.351	21.350	21.139	21.105
50	23.750	23.714	23.713	23.482	23.450
60	28.475	28.439	28.438	28.168	28.139
70	33.200	33.164	33.163	32.860	32.830
80	37.925	37.889	37.888	37.549	37.519
90	42.650	42.614	42.613	42.238	42.309
100	47.375	47.339	47.338	46.928	46.899

See References [7] and [4].

* $IB(N,q) = -N[p \log_2 p + q \log_2 q]$ (See reference [7]).

Table 2B

Expected Number of Tests for Procedure R_1, R_0 and R_{00} and lower Bounds for any Group Testing Procedures Starting with a Binomial Set of Size N.

q = .95

Initial Number of Units N	Expected Number of Tests under Procedure R_1 $H(N;q,R_1)$	Procedure R_0 $H(N;q,R_0)$	Procedure R_{00} $H(N;q,R_{00})$	Huffman Lower Bound HB(N,q)	Information Lower Bound* IB(N,q)
1	1.000	1.000	1.000	1.000	0.286
2	1.148	1.148	1.148	1.148	0.573
3	1.340	1.307	1.307	1.300	0.859
4	1.538	1.505	1.505	1.469	1.146
5	1.771	1.714	1.714	1.681	1.432
6	2.009	1.955	1.955	1.897	1.718
7	2.252	2.191	2.191	2.126	2.005
8	2.499	2.439	2.439	2.390	2.291
9	2.767	2.696	2.696	2.657	2.578
10	3.039	2.977	2.977	2.920	2.864
11	3.315	3.251	3.251	3.177	3.150
12	3.594	3.533	3.533	3.449	3.437
13	3.878	3.817	3.816	3.742	3.723
14	4.166	4.104	4.103	4.042	4.010
15	4.458	4.392	4.391	4.336	4.296
16	4.753	4.684	4.684	4.626	4.582
17	5.051	4.982	4.982	4.913	4.869
18	5.348	5.285	5.285	5.197	5.155
19	5.648	5.585	5.584	5.494	5.442
20	5.940	5.872	5.872	5.807	5.728
21	6.220	6.151	6.151	6.077	6.014
22	6.499	6.435	6.435	6.345	6.301
23	6.780	6.716	6.716	6.609	6.587
24	7.064	7.000	7.000	6.892	6.874
25	7.348	7.285	7.284	7.183	7.160
30	8.791	8.724	8.724	8.634	8.592
35	10.240	10.175	10.175	10.050	10.024
40	11.671	11.607	11.607	11.491	11.456
45	13.116	13.050	13.050	12.920	12.888
50	14.555	14.490	14.490	14.350	14.320
60	17.438	17.372	17.372	17.213	17.184
70	20.316	20.251	20.251	20.078	20.048
80	23.197	23.132	23.132	22.943	22.912
90	26.078	26.013	26.013	25.807	25.776
100	28.959	28.894	28.894	28.670	28.640

*$IB(N,q) = -N[p \log_2 p + q \log_2 q]$.

Table 2C

Expected Number of Tests for Procedure R_1, R_0 and R_{00} and Lower Bounds for any Group Testing Procedure Starting with a Binomial Set of Size N.

q = .99

Initial Number of Units N	Expected Number of Tests under Procedure R_1 $H(N;q,R_1)$	Procedure R_0 $H(N;q,R_0)$	Procedure R_{00} $H(N;q,R_{00})$	Huffman Lower Bound HB(N,q)	Information* Lower Bound IB(N,q)
1	1.000	1.000	1.000	1.000	0.081
2	1.030	1.030	1.030	1.030	0.162
3	1.070	1.060	1.060	1.060	0.242
4	1.110	1.100	1.100	1.091	0.323
5	1.159	1.141	1.141	1.131	0.404
6	1.208	1.182	1.182	1.172	0.485
7	1.258	1.232	1.232	1.213	0.566
8	1.308	1.282	1.282	1.257	0.646
9	1.366	1.332	1.332	1.309	0.727
10	1.425	1.384	1.384	1.362	0.808
11	1.484	1.437	1.437	1.414	0.889
12	1.543	1.492	1.492	1.467	0.969
13	1.603	1.550	1.550	1.521	1.050
14	1.662	1.609	1.609	1.574	1.131
15	1.722	1.668	1.668	1.628	1.212
16	1.782	1.728	1.728	1.691	1.293
17	1.849	1.788	1.788	1.755	1.373
18	1.916	1.850	1.850	1.820	1.454
19	1.984	1.913	1.913	1.882	1.535
20	2.051	1.977	1.977	1.946	1.616
21	2.119	2.043	2.042	2.010	1.697
22	2.187	2.110	2.109	2.073	1.777
23	2.255	2.177	2.176	2.137	1.858
24	2.324	2.244	2.244	2.201	1.939
25	2.392	2.312	2.311	2.264	2.020
30	2.738	2.654	2.654	2.605	2.424
35	3.101	3.007	3.007	2.972	2.828
40	3.478	3.384	3.384	3.338	3.232
45	3.859	3.768	3.767	3.701	3.636
50	4.243	4.154	4.154	4.077	4.040
60	5.026	4.936	4.936	4.882	4.848
70	5.830	5.735	5.734	5.681	5.656
80	6.647	6.557	6.557	6.486	6.463
90	7.477	7.387	7.387	7.314	7.271
100	8.320	8.227	8.227	8.147	8.079

*$IB(N,q) = - N[p \log_2 p + q \log_2 q]$.

2. Group-Testing and the Huffman Code

In this section we point out that every group-test can be interpreted as a code but the converse is not true. In particular, it is shown that for $N \geq 3$ no Huffman encoding scheme [4] for the complete binomial problem leads to a group-testing code if q is close to zero or one. In fact, it is only for values of q in some small interval centered at $q = 1/2$ (the length of which approaches zero as N tends to infinity) that the Huffman Code corresponds to a group-test code; the associated group-test is the one that tests units one at a time.

To show that every group-test gives rise to a binary code we simply write down 2^N rows of zeros and ones, each row corresponding to a possible sequence of experimental outcomes with (say) 0 representing a test that passes and 1 representing a test that fails. For a group-test the number of rows, corresponding to the possible states of nature, must be a power of 2 (in fact 2^N) and this already shows that not all binary codes can be interpreted as group-test codes. In fact, a code must satisfy several conditions to be a group-test code; a detailed discussion of some necessary conditions is given in appendix I to this paper.

For example, the code

Code	State of Nature	Probability
0	S S	q^2
10	S U	pq
110	U S	pq
111	U U	p^2

Figure 2.1 A Group-Testing Code

corresponds to a group-test with $N = 2$ units, in which the first test is on both units and subsequent tests are on one unit each. Clearly there are exactly two possible codes for group-testing with 2 units; in the other code each word has 2 digits and all units are tested one at a time.

We define the cost of the code to be the expected number of tests which is exactly the same as the expected number of digits per row (or per word in coding theory terminology). In the above example this is easily seen to be $E\{T\} = 3-q-q^2$.

The Huffman encoding for our problem with N units starts with a complete listing of 2^N probabilities

$$q^N,\ pq^{N-1},\ pq^{N-1},\ldots,p^2q^{N-2},\ldots,p^{N-1}q,\ p^N \tag{2.1}$$

and proceeds by the well known algorithm [4] of combining the two smallest probabilities, replacing them by their sum, reordering and repeating the procedure until only one number (equal to one) is left. If we replace each of the probabilities by the states of nature they represent and then follow the Huffman algorithm in reverse, i.e., separate the states of nature into disjoint sets instead of combining them, then it is known [4] that the resulting search procedure is optimal. Hence if the Huffman encoding corresponds to a group-test it would have to be the optimal group-test procedure. We shall be particularly interested in values of q close to one, but not equal to one, since this is the region in which the maximum saving in expected number of tests is obtained by group-testing, and since the optimal result for small values of q is known [13]. It is therefore desirable to prove the lemma

Lemma 1: For $N \geq 3$ and q close to one (or zero), no Huffman encoding for the complete binomial problem is a group-test code.

Proof: For q sufficiently close to one, p is close to zero and the last $2^N - (N+1)$ probabilities in (2.1) combine in the Huffman algorithm to form their sum (say ϵ) before any of the first $N+1$ probabilities enter into any sums. Similarly for q sufficiently close to one we obtain the sum $Nv + \epsilon$, where $v = pq^{N-1}$, before $V = q^N$ enters into any sum. Hence for q close to one we obtain the typical scheme, using $N = 4$ as an illustration,

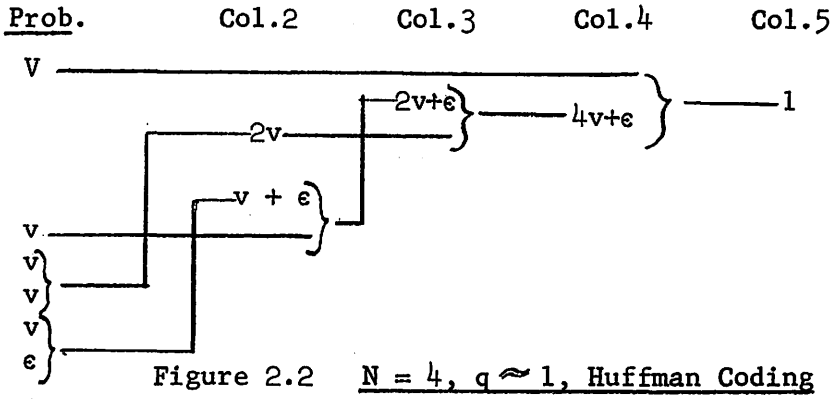


Figure 2.2 N = 4, q ≈ 1, Huffman Coding

in which at least two v's are summed before combining either with any part of ϵ. This means (looking at the algorithm in reverse) that in a final test we distinguish two states of nature such as SSUS and SUSS, i.e., for the 2nd and 3rd units we distinguished SU from US on the last test. This implies that the possibility UU for these units was previously eliminated, but there is no group-testing method of eliminating UU without having classified each of the units in question. Hence we have a contradiction.

The lemma also holds for q close to zero since the Huffman cost is symmetrical and approaches one as q approaches zero, while the best group-testing code clearly has a cost of N for small q. This completes the proof of the lemma.

Although the Huffman encoding for $N \geq 3$ and q close to one is not a group-test code it is still possible that some group-test attains the same minimal cost as the Huffman encoding. We now show that no group-testing code attains the Huffman cost for $N \geq 3$ and q close to one. Thus the Huffman cost becomes a strict (i.e., unachievable) lower bound for the expected number of tests in a group-test on N units for $N \geq 3$ and q close to one.

In the following theorem we use the fact that with any exhaustive binary code containing W words we can associate an arborescence with W pendant vertices (or endpoints) and $W-1$ branching vertices (or continuation points); the latter including the root (or starting point). (See appendix I for explanation of terminology.) If probabilities p_i $(i = 1, 2, \ldots, W)$ are assigned to the words (or pendant vertices) then we define the probability of any vertex V to be equal to the sum of all the probabilities of the pendant vertices that can be reached from V; in particular the root then always has probability one. The sum of the probabilities over all branching vertices is equal to the cost of the code. This follows from the fact that the number of digits in any word is the number of times that the probability associated with that word is included in the various branching point sums of the code. (See Picard [5] for another

proof of this result.) Furthermore if, as in figure 2.2, we replace the original (atomic) probabilities by the probabilities of any collection of disjoint, exhaustive subsets of pendant vertices (or words) then the analogous result also holds for the subproblem of finding out in which of these subsets the true state of nature lies.

Theorem 1: For $N \geq 3$ and q close to one (or zero) no group-testing code can achieve the cost of the Huffman encoding.

Proof: The result for q close to zero is clear from the remark in lemma 1 and we need only consider q close to one. We again use $N = 4$ to illustrate certain ideas but it should be noted that the argument holds for any $N \geq 4$; the case $N = 3$ is treated as a special case.

Consider for $N = 4$ the following partition of the 16 states of nature into 7 subsets: {SSSS}, {SSSU}, {SSUS}, {SUSS}, {USSS}, {SSUU} and {the remaining ten possibilities} with probabilities denoted by $q^4 = N$, $pq^3 = v$, v, v, v, $\epsilon_1 = p^2q^2$ and ϵ_2, respectively; let $\epsilon = \epsilon_1 + \epsilon_2$. [More generally, ϵ_1 and ϵ_2 correspond to the events "The first two units are both satisfactory and at least 2 units are unsatisfactory" and "at least two units are unsatisfactory, at least one being among the first two units".]

We refer to the problem of finding out in which of these 7 subsets the true state of nature lies as a subproblem and to the corresponding code (see figure 2.2) as a subcode. The cost C_1 associated with the subcode in figure 2.2 is clearly

$$C_1 = 1 + 9v + 3\epsilon. \tag{2.2}$$

If we let $C^*(S_\epsilon|R)$ denote the conditional cost of finding the true state of nature given that it is in the subset S_ϵ, which has probability ϵ, and that some continuation procedure R is used, then the total cost $C(R)$ is given by

$$C(R) = C_1 + \epsilon C^*(S_\epsilon|R) = C_1 + C(S_\epsilon|R); \tag{2.3}$$

here $C(S_\epsilon|R) = \epsilon\, C^*(S_\epsilon|R)$ is the result obtained if we use the same algorithm as above but start with probabilities summing to ϵ, instead of one. If the Huffman procedure has a strictly smaller value of C_1 than any group-testing

code then it is clear (from the fact that the Huffman procedure is optimal) that the total cost must also be strictly smaller than any group-testing code. In other words it suffices to show that the Huffman cost is unachievable in the subproblem.

We now consider three basic modifications, at least one of which is needed to change a Huffman encoding to a group-testing code, and we shall show that each of them increases the cost C_1 of the subcode.

Modification 1: This modification first tests all N units and, if that fails, we then test a subset containing fewer than $N - 1$ units. We use $N = 4$ to illustrate the ideas.

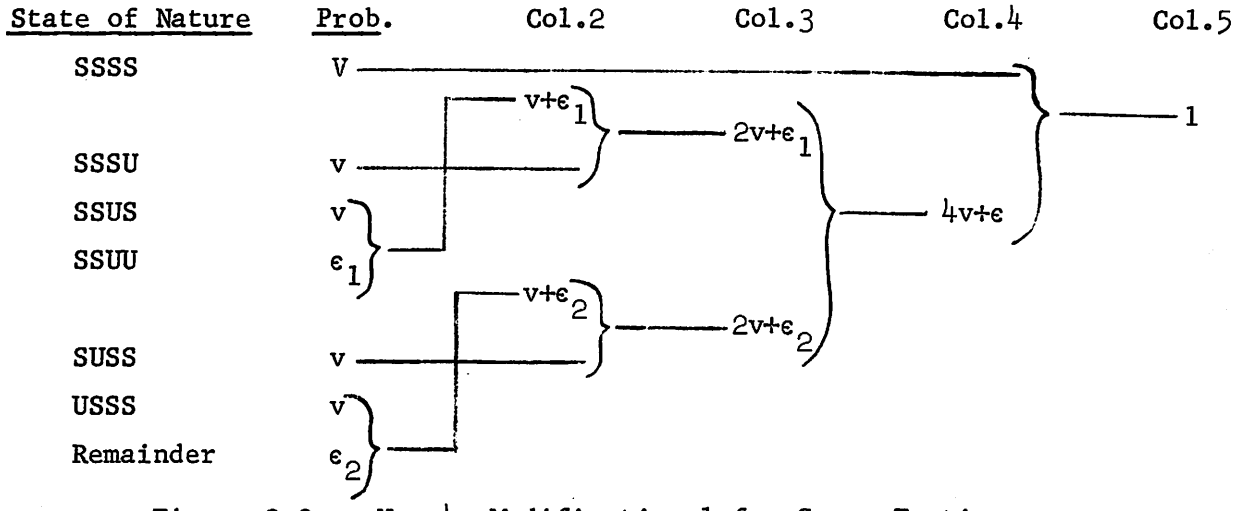


Figure 2.3 N = 4, Modification 1 for Group-Testing

The cost of the subcode in figure 2.3 is $1 + 10v + 3\epsilon$ and since ϵ_1 and ϵ_2 both contain p^2 it follows that the total cost of any code that uses this subcode is $1 + 10v + \mathcal{O}(p^2)$ as $q \to 1$. Similarly, using (2.2) and (2.3), any code that uses the subcode in figure 2.2 has a total cost of $1 + 9v + \mathcal{O}(p^2)$. Hence for $N = 4$ and q close to one the cost of the Modification 1 is strictly greater than that of the Huffman encoding.

For higher values of N, we could form subsets of size $3 = 2^2-1$ containing two v's as in figure 2.3 or of size $7 = 2^3-1$ containing three v's (for example, SSSSU, SSSUS, SSUSS, SSSUU, SSUSU, SSUUS, SSSSS, with the first three being v's) or etc. For any such grouping at least two v's have to be separately combined with ϵ's and this introduces at least one extra v into the total cost.

[Later we give explicit formulas for the coefficients of v in the Huffman encoding and in Modification 1 for group-testing; these provide another proof that the former is less than the latter for $N \geq 4$.] It follows that for $N \geq 4$ and q close to one, the code of Modification 1 is strictly greater than that of the Huffman encoding.

Modification 2: This modification first tests all N units and, if that fails, it then tests separately for each of the N states of nature with probability v. We again use $N = 4$ to illustrate the ideas.

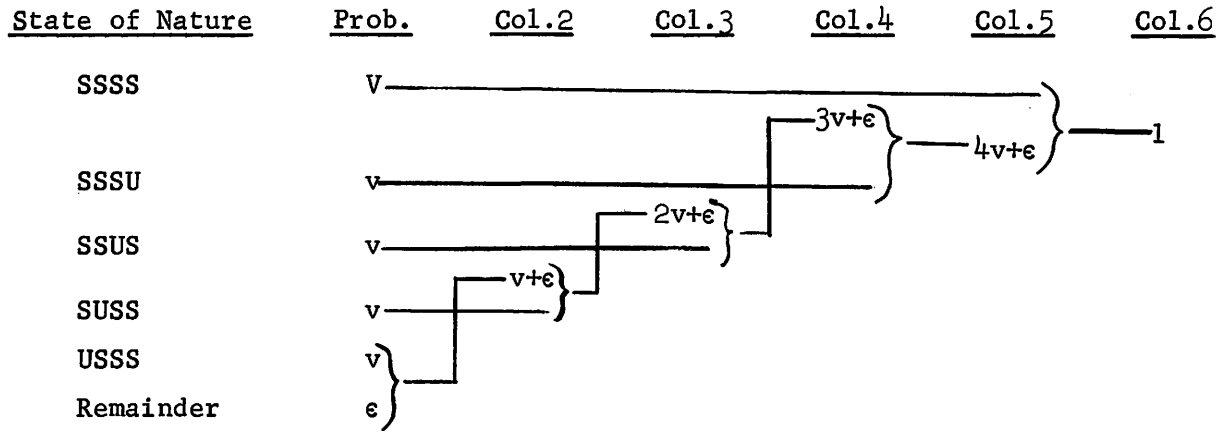


Figure 2.4: N = 4, Modification 2 for Group-testing

The cost of the subcode in Figure 4 is $1 + 10v + 4\epsilon$ which, for $N = 4$, is even greater than that obtained for Modification 1 for all values of q, and the resulting partition is actually coarser since all 11 states making up the probability ϵ are combined here as one. For $N > 4$ we can disregard the ϵ since the coefficient of v in Modification 2 will be larger than that in Modification 1. This follows from the fact that for any integer w with $2 \leq w \leq N-2$ we can get a better result than $\binom{N+1}{2}$ for the coefficient of v by simply breaking up N into two subsets of size w and $N-w$ and using Modification 2 within each subset. The result is better since it is easily seen that the inequality

$$\binom{N+1}{2} \geq N + \binom{w+1}{2} + \binom{N-w+1}{2} \tag{2.4}$$

holds for any $N \geq 4$ and this inequality is strict for $N \geq 5$. Hence we can

disregard Modification 2 for $N \geq 4$; we shall return to this later to discuss the case $N = 3$.

Modification 3: In this modification we combine V and v before combining either with any other probability. This adds an extra q^N to the cost and makes this modification inferior to both of the other two modifications for q close to one. We therefore disregard this possibility in the subsequent search for an optimal procedure for q close to one, i.e., the optimal procedure must start by testing all N units if q is close to one.

Consider any modification of a Huffman encoding to form a group-testing code. It must involve one or more of these three basic modifications, i.e., to avoid adding two v's in Column 2, we must either form subgroups as in Modification 1, or add the v's one at a time to ϵ as in Modification 2, or add at least one v to V as in Modification 3. In each case the cost is strictly greater than the Huffman cost for $N \geq 4$ and q close to one.

For $N = 3$, it is easy to verify directly that for the subproblem considered above (with three v's) the cost of the subcode using the Huffman encoding is $1 + 6v + 2\epsilon$ which is strictly less than the result $1 + 6v + 3\epsilon$ obtained by using Modification 2. Moreover for $N = 3$, Modification 2 gives a result that is strictly less than that obtained by using Modification 1. (In addition, the latter has no group-testing interpretation for subsets containing one or two v's.) Modification 3 can be disregarded for the same reason as before. This completes the proof of theorem 1.

Remark 1: For $N = 2$ and q close to one the Huffman encoding is identical with Modification 2; the resulting code is given in Figure 2.1.

Remark 2: The above comparisons of three modifications gives us considerable information about the structure of optimal group-testing procedures for q close to one: For any N we start by testing all N units. If in the course of experimentation we reach a similar situation in which there are only n units unclassified ($n \leq N$) and the a posteriori knowledge about them consists only

of the (binomial) assumption made at the outset then we again test all n units; we call this an H(n)-situation (see[7]). We now consider what to do next if the n units do not pass, i.e., in the so-called G(n,n)-situation with $n \le N$ (see [7]). For $N = 3$, we found above that Modification 2 is strictly better than Modification 1 (with three v's and two ϵ's) and the former is the optimal group-testing procedure for $n = 3$ and q close to one. This procedure for the G(3,3)-situation is to keep trying different pairs of units until one pair passes or all three pairs fail, whichever comes sooner. If one pair passes we are through; otherwise we then test one unit at a time until one unit passes or all fail, whichever comes sooner. Since Modification 2 is inferior to Modification 1 for $n \ge 4$ it follows that, for q close to one, Modification 2 yields an optimal procedure for $n = 3$ only. The procedure R_0 defined in [8] as well as R_{00} defined in this paper both have this property for $n = 3$, but R_1 does not.

Remark 3: Although the Huffman cost is not attained for q close to one, it would be useful to have an explicit expression for it since it is a lower bound for any group-testing procedure (see Table A6). For any N we define the integer r_i by the inequalities

$$2^{r_i} \le \binom{N}{i} < 2^{r_i+1} . \tag{2.5}$$

The cost of the Huffman encoding of the subproblem consisting of $N + 2$ events with probabilities $v_0 = q^N$, $v_1 = pq^{N-1}$ (repeated N times) and $\epsilon_1 = 1 - v_0 - Nv_1$ for q close to one is given (proof omitted) by

$$\begin{aligned} \text{Cost} &= 1 + [(r_1+2)N - 2^{r_1+1} + 1]v_1 + (r_1+1)\epsilon_1 \\ &= q^N + [(r_1+3)N - 2^{r_1+1} + 1]v_1 + \mathcal{O}(p^2) \end{aligned} \tag{2.6}$$

and the latter result also holds for the complete binomial problem. If we apply the Huffman algorithm to the subproblem consisting of $\binom{N}{i} + 1$ events with probabilities $v_i^* = v_i/\epsilon_{i-1}$ (repeated $\binom{N}{i}$ times) and $\epsilon_i^* = \epsilon_i/\epsilon_{i-1}$ where

$v_i = p^i q^{N-i}$ and $\epsilon_i = \sum_{j=i+1}^{N} \binom{N}{j} p^j q^{N-j}$ then we obtain for $i = 2, 3, \ldots, N-1$ and q close to one

$$(2.7) \qquad (\epsilon_{i-1})\ \text{Cost} = [(r_i+2)\binom{N}{i} - 2^{r_i+1} + 1]v_i + (r_i+1)\epsilon_i.$$

The left side of (2.7) is the probability that there are at least i defectives present multiplied by the conditional cost of determining which of the above $\binom{N}{i} + 1$ events is the true state of nature given that there are at least i defectives present. Summing the results in (2.6) and in (2.7) for each i, we find that the total cost of the complete binomial problem for q close to one and any N is

$$(2.8) \qquad \begin{aligned} \text{Cost} &= 1 + \sum_{i=1}^{N-1} [(r_i+2)\binom{N}{i} - 2^{r_i+1} + 1]\, p^i q^{N-i} + \sum_{i=1}^{N-1} (r_i+1)\epsilon_i \\ &= 1 - p^N + \sum_{i=0}^{N} p^i q^{N-i} [\binom{N}{i} \sum_{\alpha=0}^{i} (r_\alpha+1) - 2^{r_i+1} + 1]. \end{aligned}$$

For example, for $N = 2$ this gives $1 + 3pq + 2p^2 = 3 - q - q^2$ which is the result for $(\sqrt{5}-1)/2 < q < 1$ in each of the procedures R_1, R_0 and R_{00}.

3. Optimal Modifications for the G(m,n)-Situation for q Close to One

In section 2 we applied the Huffman type analysis to the H(n)-situation and the G(n,n)-situation; in this section we apply a similar analysis to the G(m,n)-situation (see [7]). In a G(m,n)-situation there are n units still unclassified and included among these is a subset of m units $(2 \le m \le n)$ which is known to contain at least one defective unit. This partition of the n units into 2 subsets of size m and $n-m$ is in direct correspondence with the partition of the n events with probability v in Modification 1 (see Figure 2.3 where $n = N = 4$ and $m = n - m = 2$) into two subsets with m of the v's in one subset and $n - m$ of the v's in the other. The details are similar to what we had before with modification 1 and 2 with the following minor changes:

1. The event "all good" with probability q^N is omitted.
2. The value of v is pq^{n-1} where $n \le N$, but the number of v's is only m; we replace v by $v^* = v/(1 - q^m)$.
3. The value of ϵ is $1 - q^m - mv$; we replace ϵ by $\epsilon^* = \epsilon/(1 - q^m)$ and ϵ_i by $\epsilon_i^* = \epsilon_i/(1 - q^m)$ $(i = 1, 2)$.

Then v^*, ϵ_1^*, ϵ_2^* are the conditional probabilities of the $m + 2$ events in our subproblem analogous to those of Modification 1, and v^*, ϵ^* are the conditional probabilities corresponding to those of Modification 2; we use the same terminology "Modifications 1 and 2" and as before with the above 3 changes and with $\epsilon^* = \epsilon_1^* + \epsilon_2^*$ in Modification 1. The cost of the subproblem for Modification 1 for $m = 4 \le n$ is $1 + 6v^* + 2\epsilon^*$ and the contribution to the total cost for the complete binomial problem is the product

$$(3.1) \qquad (1 - q^4)\,\text{Cost} = 1 - q^4 + 6v + 2\epsilon = 10pq^{n-1} + 6(2n - 5)p^2q^{n-2} + \dots .$$

The corresponding cost for Modification 2 for $m = 4 \le n$ is $1 + 6v^* + 3\epsilon^*$ and the contribution to the total cost for the complete binomial problem is the product

$$(3.2) \qquad (1 - q^4)\,\text{Cost} = 1 - q^4 + 6v + 3\epsilon = 10pq^{n-1} + 8(2n - 5)p^2q^{n-2} + \dots ;$$

the last expression in (3.1) and (3.2) in powers p^iq^{n-i} for increasing i is not needed here but is useful for later discussions. Hence for $m = 4$ and q close to one, Modification 1 gives a better result at a lower cost than Modification 2. Moreover we find for $n \ge m \ge 5$, as in the proof of theorem 1, that by using Modification 1 with a single subset of two v^*'s and ϵ_1^* we replace $(m-1)v^*$ by $3v^*$ in Modification 2, and hence Modification 1 is strictly better than Modification 2 for $n \ge m \ge 5$ and q close to one. On the other hand, for $m = 2 < n$ and $m = 3 < n$ it is easy to verify that Modification 2 is better than Modification 1. In fact with the given definition of ϵ^*, we are forced to use Modification 2 for $m = 2$ and it follows as in

lemma 1 that Modification 1 has no group-testing interpretation for $m = 3$.)

The above results show for $n \geq m$ and q close to 1 that Modification 1 is strictly better for $m \geq 4$ and Modification 2 is strictly better for $m = 2$ and 3. The use of Modification 2 for $m = 2$ and 3 does imply the nature of the next m tests (see Tables A1 and A2) for q close to one. However we should not assume that a procedure with Modification 1 and with the best leading coefficient is necessarily a unique group-testing procedure (even for the very next test). We now describe a class of procedures $R^{(j)}$ $(j = 0, 1, \ldots, n-4)$ all corresponding to Modification 1; it follows from the results of Section 4 that any procedure with Modification 1 and with the lowest leading coefficient must be in this class. We again use $m = 4$ to illustrate the ideas; in this case optimal means that it corresponds to Modification 1 and the leading coefficient in the cost expression is 10.

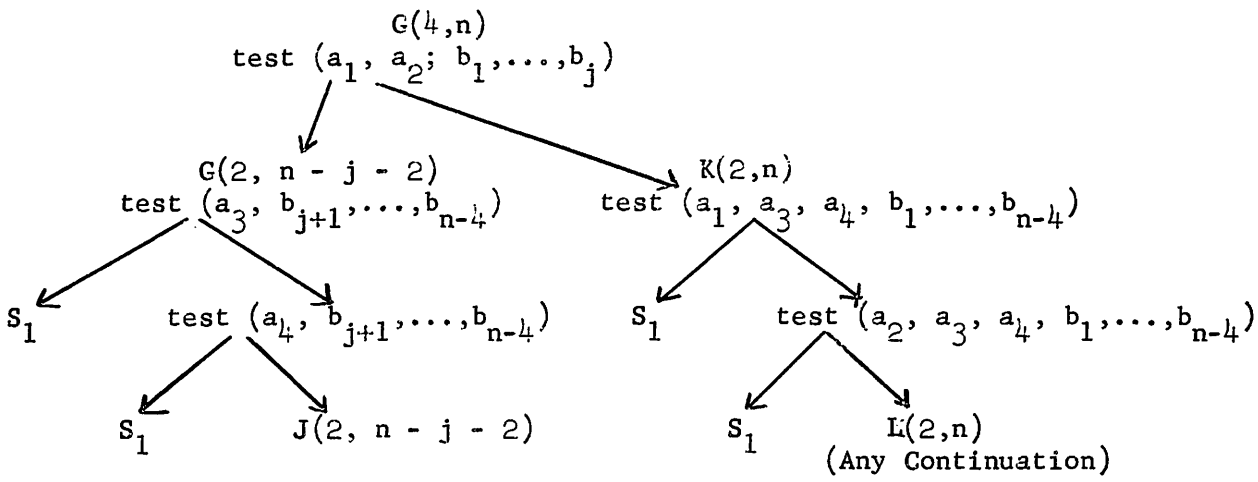


Figure 3.1 A Class of Procedures $R^{(j)}$ (Modification 1) for $n \geq m = 4$.

Here S_1 denotes a terminal stopping point with exactly one defective. For $j = 0$ we replace $K(2,n)$ by $G(2,n)$ after the first test fails and we replace $L(2,n)$ by $J(2,n)$ after 3 successive failures. For each $j \geq 0$ we get the same leading term, $10v$, in (3.1); hence we will have to make a deeper analysis of the cost to show that $j = 0$ gives the best results; this will be done later in Sections 5 and 6.

First we obtain several results on the basis of the following temporary

Assumption 1: In a $G(m,n)$ situation with $n \geq m \geq 4$ the next test-group of size $x = x(m,n)$ is taken only from the defective set.

In an $H(n)$ or $G(m,n)$ situation with $n \geq m \geq 4$ the size x of the next test-group is determined by the basic recursion formulas

$$(3.3) \qquad H(n) = 1 + \min_{1 \leq x \leq n} [q^x H(n-x) + pG^*(x,n)],$$

$$(3.4) \qquad G^*(m,n) = 1 + q + \ldots + q^{m-1} + \min_{1 \leq x \leq m-1} [q^x G^*(m-x,n-x) + G^*(x,n)]$$

and the boundary conditions $H(0) = 0$ and $G^*(1,n) = H(n-1)$ for $n = 1, 2, \ldots$; these formulas are derived in [7] and are used there (for $n \geq m \geq 2$) to define procedure R_1 which is optimal for all small values of q. $H(n)$ is the expected number of tests starting with a binomial or H-situation and $pG^*(m,n)/(1 - q^m) = G(m,n)$ is the expected number of tests starting with a $G(m,n)$ situation. Under procedure R_{00} the results for $H(n)$ and $G(m,n)$ are different than those obtained for procedure R_1 only because we derive special subroutines (and corresponding recursion formulas) for $m = 2$ and $m = 3$. These special subroutines are obtained in Section 7.

On the basis of assumption 1 we now prove that for $m \geq 4$ and any q the integer solution $x = x(m,n)$ of (3.4) depends only on m; this makes it much easier to describe the procedure R_{00}. It is interesting that this result does not depend on the definitions of R_{00} for the $G(2,n)$ and $G(3,n)$ situations.

Theorem 2: Under assumption 1 for a $G(m,n)$ situation with $n \geq m \geq 4$ and q close to one the optimal number $x = x(m)$ of units for the next test does not depend on n.

Proof: Let $F(m,n)$ with $n \geq m \geq 4$ denote the expected number of tests necessary to reach a $G(m,n)$ situation with $m = 1$, 2 or 3; we will show that $F(m,n) = F(m)$ depends only on m. For convenience we set

(3.5) $$F^*(m,n) = (1 - q^m)\, F(m,n)/p.$$

For $m \geq 4$ we have (using assumption 1)

(3.6) $$G^*(m,n) = F^*(m,n) + \sum_{i=1}^{I} q^{b_i} G^*(a_i, n-b_i)$$

where $a_i = 1$, 2, or 3, I is the number of paths leading to a $G(m,n')$ situation with $m = 1$, 2, or 3 and b_i is the number of units proved good along the i^{th} path $(i = 1, 2, \ldots, I)$; for $m = 1$, 2, or 3 equation (3.6) is an identity and yields no reduction. Moreover, if c_i is the number of tests needed along the i^{th} path, then $F^*(m,n)$ in (3.6) is a polynomial of the form

(3.7) $$F^*(m,n) = p \sum_{i=1}^{I} c_i q^{b_i} (1 - q^{a_i}).$$

To show that $F^*(m,n)$ does not depend on n we consider the best continuations assuming that $x = j$ yields the minimum in (3.4); the number of paths $I = I(j)$ can depend on j. For some integer I_1 $(0 \leq I_1 \leq I)$ we can then write

(3.8) $$G^*(m-j, n-j) = F^*(m-j, n-j) + \sum_{i=1}^{I_1} q^{b_i - j} G^*(a_i, n-b_i)$$

(3.9) $$G^*(j,n) = F^*(j,n) + \sum_{i=I_1+1}^{I} q^{b_i} G^*(a_i, n-b_i)$$

where the a_i and b_i are the same as in (3.6) and (3.7). Then, substituting (3.6), (3.8) and (3.9) in (3.4), we find that the three summations cancel for any j giving the result

(3.10) $$F^*(m,n) = 1 + q + \ldots + q^{m-1} + \min_{1 \leq j \leq m-1} [q^j F^*(m-j, n-j) + F^*(j,n)],$$

the same equation as for $G^*(m,n)$. We now use induction of m and the fact that $F^*(1,n) = F^*(2,n) = F^*(3,n) = 0$, independent of n. By the induction hypothesis the r.h.s. of (3.10) does not depend on n and hence the l.h.s. also does not; hence we can write $F^*(m)$ for $F^*(m,n)$. Since the integer (say, $j = x$) that minimizes the r.h.s. of (3.4) is the same as the integer that

minimizes the r.h.s. of (3.10) it now follows that x (the size of the next test for a $G(m,n)$ situation with $n \geq m \geq 4$) also does not depend on n; we write $x = x(m)$. This proves theorem 2 under assumption 1; the problem of removing this assumption is studied in Section 6.

Remark: For q close to one and $m \geq 4$ we can disregard the possibility that $a_i = 1$ in the above proof; then we need only consider paths leading to $G(2,n')$ and $G(3,n')$ situations to get the conclusion for q close to one.

4. Derivation of the Optimal Leading Coefficient for $G^*(m,n)$ with $n \geq m \geq 4$.

In general group-testing situation we have a problem of optimal encoding subject to special restriction. We have a number M of states of nature with equal probability $v = p^s q^{N-s}$ where s,N are fixed integers with $0 \leq s < N$. All the other states have probability of the form $\epsilon = p^t q^{N-t}$ with $t > s$ and we shall also use ϵ for the sum of such probabilities. One restriction is that we can never reduce the set of possible true states of nature to exactly two of the M states above without including another state with probability $\epsilon = p^{s+1} q^{N-s-1}$. This means that no two of the v's can be added at the outset, i.e., in forming Column 2. It follows that for any partition of the problem into two categories, if a category has two or more of the M states then, it must also have at least one state with probability ϵ.

We consider the subproblem consisting of M states with probability v and (say) Q states with probability $\epsilon_1, \epsilon_2, \ldots, \epsilon_Q$, which need not be equal. We can assume that the sum of these ϵ's and Mv is unity since, if it is not, we merely divide each by the sum and in computing the contribution to the total cost, the denominator cancels.

Let $h(m)$ denote the coefficient of v for the best group-test encoding of this subproblem. Let the integers y and $M - y$ denote the number of the M-states assigned to each of two categories by the first partition of the M states. For $M \geq 4$ and the best group-test we have

$$h(M) = M + \min_{1 \leq y \leq M-1} [h(y) + h(M-y)], \tag{4.1}$$

and the optimal value of y is the one that minimizes the right hand side of (4.1); the choice $y = 0$ need not be considered. Equation (4.1), together with the boundary conditions

$$h(1) = 0, \quad h(2) = 3, \quad h(3) = 6, \tag{4.2}$$

define $h(M)$ for every positive integer M: we could also use (4.1) with $M \geq 3$ and omit the last boundary condition.

We now produce an explicity function satisfying (4.1) and (4.2) and show exactly which set of integers $y = y(M)$ satisfies (4.1) for each M; in particular, the integer (or integers) closest to $M/2$ are always included in this set. This limits severely the possible encodings that have to be searched to find the optimal one.

For any $M \geq 1$, let the integer $b = b(M) \geq 0$ be defined by

$$3 \cdot 2^{b-2} \leq M < 3 \cdot 2^{b-1} \tag{4.3}$$

so that b is the exponent of the power of 2 that is closest to M, when that power is unique. Let $h_1(M)$ be defined so that $h_1(1) = 0$ and for $M \geq 2$

$$h_1(M) = M(b + 2) - 3 \cdot 2^{b-1}. \tag{4.4}$$

Lemma 2: The function $h(M)$ in (4.1) and (4.2) is identical with $h_1(M)$ given in (4.4). For $M \geq 4$ the set of y-values that minimize the right hand side of (4.1) includes the integer (or integers) closest to $M/2$.

Proof: Suppose first that $h_1(y) + h_1(M - y)$ is minimized by taking y as close as possible to $M/2$. Then for even $M \geq 4$ we put $h_1(M/2)$ for $h(M/2)$ in the right hand side of (4.1) and obtain for $y = M/2$

$$M + 2[\tfrac{M}{2}(b+1) - 3 \cdot 2^{b-2}] = M(b+2) - 3 \cdot 2^{b-1} \tag{4.5}$$

which agrees with (4.4) of (4.1). Similarly for odd $M > 4$ and $M \neq 3 \cdot 2^{b-1}-1$ we obtain for $y = (M-1)/2$

$$M + [(\tfrac{M-1}{2})(b+1) - 3 \cdot 2^{b-2}] + [(\tfrac{M+1}{2})(b+1) - 3 \cdot 2^{b-2} = M(b+2)-3\cdot 2^{b-1}, \tag{4.6}$$

and for $M = 3 \cdot 2^{b-1}-1$ we obtain

$$(4.7) \qquad M + [(\tfrac{M-1}{2})(b+1) - 3 \cdot 2^{b-2}] + [(\tfrac{M+1}{2})(b+2) - 3 \cdot 2^{b-1}]$$

$$= M(b+2) - 3 \cdot 2^{b-1};$$

which agree with (4.4). Thus $h_1(M)$ satisfies (4.1). It is easily verified that $h_1(2) = h(2) = 3$ and $h_1(3) = h(3) = 6$ and, by definition, $h_1(1) = h(1) = 0$. Hence if the above supposition is true, then $h_1(M) = h(M)$ for all $M \geq 1$.

We now show that for any $M \geq 4$ the value (or values) of y closest to $M/2$ minimize

$$(4.8) \qquad h(y; M) = h_1(y) + h_1(M - y) \qquad (1 \leq y \leq M-1)$$

where $h_1(y)$ is given by (4.4). Clearly, for any M, $h(y; M)$ is symmetric about $y = M/2$. If we can show for any $M \geq 4$ that $h(y; M)$ is convex on the domain of integers y $(1 \leq y \leq M-1)$, then it follows that the minimum is achieved at an integer y which is closest to $M/2$. For this convexity it is sufficient to show by (4.8) that $h_1(y)$ is convex for $y \geq 2$. For the latter result it is sufficient to show that for any 3 consecutive integers $(y-1, y, y+1)$ with $y \geq 2$ the second difference of $h_1(y)$ is nonnegative, i.e., $\Delta^2 h_1(y) = h_1(y+1) - 2h_1(y) + h_1(y-1) \geq 0$. For $y \geq 3$, three disjoint and exhaustive cases arise according as

i) all three integers have the same b-value, or

ii) only the two smallest integers have the same b-value, or

iii) only the two largest integers have the same b-value.

If we take $y = 3 \cdot 2^{b-2} + x$ for M in (4.4), then for case i) we have a linear function in x, since b is constant, and hence $\Delta^2 h_1(y) = 0$. For case ii) we set $y + 1 = 3 \cdot 2^{b-2}$ and obtain

$$(4.9) \qquad \Delta^2 h_1(y) = [3 \cdot 2^{b-2}(b+2) - 3 \cdot 2^{b-1}] + [(3 \cdot 2^{b-2}-2)(b+1) - 3 \cdot 2^{b-2}]$$

$$- 2[(3 \cdot 2^{b-2}-1)(b+1) - 3 \cdot 2^{b-2}] = 0.$$

For case iii) we set $y = 3 \cdot 2^{b-2}$ and obtain

$$(4.10) \qquad \Delta^2 h_1(y) = [(3 \cdot 2^{b-2}+1)(b+2) - 3 \cdot 2^{b-1}] + [(3 \cdot 2^{b-2}-1)(b+1) - 3 \cdot 2^{b-2}]$$

$$- 2[3 \cdot 2^{b-2}(b+2) - 3 \cdot 2^{b-1}] = 1.$$

Finally, for $y = 2$ we see (from the values of $h_1(y)$ for $y = 1, 2,$ or 3 given above) that $\Delta^2 h_1(2) = 0$. This proves the convexity of $h_1(y)$ and hence $h_1(y)$ assumes its minimum at the integer (or integers) closest to $M/2$.

To complete the proof we use induction and assume that $h(x) = h_1(x)$ for all integers $x < M$. Then by (4.1)

$$(4.11) \qquad h(M) - M = \min_{1 \le y \le M-1} [h(y) + h(M-y)] = \min_{1 \le y \le M-1} [h_1(y) + h_1(M-y)]$$

$$= h_1(M) - M.$$

Thus $h(M) = h_1(M)$ and since $h(1) = h_1(1) = 0$, the lemma is proved. The above lemma with different boundary conditions (4.2) also appears in [11].

Lemma 2 gives a value of y that necessarily minimizes $h(y; M)$ in (4.1) but it is also useful to know all the values of y that minimize $h(y; M)$. The answer is given in

Lemma 3: For any $M \ge 4$, with b defined by (4.3), an integer y will minimize the r.h.s. of (4.1) if and only if y and $M-y$ are both in the closed interval bounded by $3 \cdot 2^{b-3}$ and $3 \cdot 2^{b-2}$.

Proof: Since $h(y) = h_1(y)$ is symmetric about $y = M/2$, it is sufficient to prove the result for $y \le M/2$ or $y \le M-y$. We now consider different values for y and $M - y$.

$$(4.12) \quad \text{Case 1:} \qquad 3 \cdot 2^{b-3} \le y \le M - y < 3 \cdot 2^{b-2}.$$

Then $b(y) = b(M - y) = b-1$ and to check the equality in (4.1) we use (4.4) and obtain for the r.h.s. of (4.1)

$$(4.13) \qquad M + h(y; M) = M + [y(b+1) - 3 \cdot 2^{b-2}] + [(M-y)(b+1) - 3 \cdot 2^{b-2}]$$

$$= M(b+2) - 3 \cdot 2^{b-1} = h(M).$$

Thus all y values satisfying (4.12) will yield the minimum $h(M)$ in (4.1).

(4.14) Case 2: $3 \cdot 2^{b-3} \le y < 3 \cdot 2^{b-2} \le M - y.$

Then $b(M - y) = b = b(y) + 1$ since $M - y \le M$ and we obtain as above

$$(4.15) \qquad M + h(y; M) = M + [y(b+1) - 3 \cdot 2^{b-2}] + [(M - y)(b+2) - 3 \cdot 2^{b-1}].$$

$$= M(b+2) - 3 \cdot 2^{b-1} + [M - y - 3 \cdot 2^{b-2}].$$

Since the last quantity in brackets is nonnegative, we get the minimum value for $h(m)$ only when $M - y = 3 \cdot 2^{b-2}$.

(4.16) Case 3: $y < 3 \cdot 2^{b-3} \le M - y < 3 \cdot 2^{b-2}.$

Then $b(y) < b(M - y) = b-1$ and the same calculation gives

$$(4.17) \qquad M + h(y; M) = M + [y(b(y) + 2) - 3 \cdot 2^{b(y)-1}] + [(M-y)(b+1)-3 \cdot 2^{b-2}]$$

$$= h(M) + [3 \cdot 2^{b-2} - 3 \cdot 2^{b(y)-1} - y(b-1-b(y))]$$

$$> h(M) + 3 \cdot 2^{b(y)-1}[2^x - (x+1)]$$

where we have replaced y using the second inequality of (4.3) and let $x = b-1-b(y) \ge 1$. Since $2^x \ge x+1$ for $x = 1, 2, \ldots$ it follows that the strict inequality holds in (4.17). This proves that no value of y satisfying (4.16) attains the minimum $h(M)$.

(4.18) Case 4: $0 < y < 3 \cdot 2^{b-3}, \; 3 \cdot 2^{b-2} \le M - y.$

Then $b(y) + 1 < b = b(M - y)$ and the same calculation gives

$$(4.19) \qquad M + h(y; M) = M + [y(b(y) + 2) - 3 \cdot 2^{b(y)-1}]$$

$$+ [(M-y)(b+2) - 3 \cdot 2^{b-1}]$$

$$= h(M) + [(M-y) - y(b-1-b(y)) - 3 \cdot 2^{b(y)-1}]$$

$$> h(M) + [3 \cdot 2^{b-2} - 3 \cdot 2^{b(y)-1}(b-b(y))]$$

$$\geq h(M) + 3 \cdot 2^{b(y)-1}[2^x - (x+1)]$$

and the same argument as in Case 3 shows that no value of y satisfying (4.18) attains the minimum $h(M)$. Since these four cases are exhaustive, lemma 3 is proved.

Remark: The result of lemma 3 is useful not only to delimit the possible values for the first test-group size x under Assumption 1, but also in the more general framework in which this assumption is dropped. Starting with a $G(m,n)$ situation, let x_A and x_B denote the number of units in the first test-group from the defective set and the binomial set, respectively; let the total size of the first test be $x = x_A + x_B$. Lemmas 2 and 3 tell us how to break up the m states of nature each with probability pq^{n-1} so as to obtain the smallest value $h(m)$ of the leading coefficient in the cost expression (for q close to 1). It follows that the results of lemmas 2 and 3 can also be applied to x_A in the more general framework, e.g., both x_A and $m - x_A$ have to lie in the closed interval $[3 \cdot 2^{b-3}, 3 \cdot 2^{b-2}]$ where b is defined by (4.3) with $M = m$. Also $x_A = x_A(m)$ depends only on m and we are interested as in Figure 3.1 to consider primarily the same values for x_A that we find to be best for x under Assumption 1.

5. Optimal Test Size for the $G(m,n)$ situation with $m \geq 4$ and q Close to One.

In this section we show how to apply the results of Section 4 to find the optimal test size $x(m)$ for any $G(m,n)$ situation with $m \geq 4$. A simple explicit formula for the optimal $x(m)$ is given together with explicit formulas for the first two coefficients, $h(m)$ and $g(m,n)$, in an asymptotic expression for the expected number of tests starting from a $G(m,n)$ situation with $m \geq 4$, i.e., for the coefficients of pq^{n-1} and p^2q^{n-2} in $pG^*(m,n)$ when the expected number of tests is expressed as a linear combination of the quantities p^iq^{n-i} $(i = 1,\ldots,n)$.

Under Assumption 1 the next test-group in a $G(m,n)$ situation with $n \geq m \geq 4$ is taken from the defective set of size m and hence $x(m) < m$. For q close

to one, our first concern is about the m states of nature with probability pq^{n-1} so that $s = 1$, $N = n$ and $M = m$. By lemma 3 we know that $x = x(m)$ must be such that both x and $m - x$ are in the closed interval $[3 \cdot 2^{b-3}, 3 \cdot 2^{b-2}]$, where b is defined by (4.3). If m is of the form $3 \cdot 2^{b-2}$ then it follows from lemma 3 that $x = 3 \cdot 2^{b-3} = m - x$, so that the result for $x(m)$ is then unique. In other cases there are several values of x all yielding the same leading coefficient $h(m)$. The appropriate criterion for choosing the optimal x (for q close to one) is to select from all those yielding the same first coefficient $h(m)$ the one that yields the smallest second coefficient $g(m,n)$. Of course, if the latter did not give a unique result we would go to the third coefficient, fourth coefficient, etc; computation of the first and second coefficients usually turn out to be sufficient to give a unique optimal value of $x = x(m)$.

Before giving an explicit formula for the optimal $x(m)$ for q close to one, we need some more lemmas.

Lemma 4: The coefficients of p^2q^{n-2} in $(1-q^2)G(2,n)$ and in $(1-q^3)G(3,n)$ are, respectively,

$$g(2,n) = h(2n-3) + 2(2n-3), \tag{5.1}$$

$$g(3,n) = h(3n-6) + 3(3n-6). \tag{5.2}$$

Proof: For the $G(2,n)$ situation with q close to one we test $(a_1, b_1, \ldots, b_{n-2})$ where the a's are units from the defective set and the b's are units from the binomial set. If that fails we test $(a_2, b_1, \ldots, b_{n-2})$ and if they both fail we have a $J(2,n)$ situation with $2n-3$ possible states of nature, each having exactly 2 defectives and probability p^2q^{n-2}/s, where $s = 1-q^2$. Letting $\varepsilon = (2n-3)p^2q^{n-2}/s$ the Huffman Analysis takes the form

States	Probabilities	Col. 2	Col. 3
(a_1 defective, all others good)	pq^{n-1}/s	.....	$(\varepsilon + 2pq^{n-1})/s$
(a_2 defective, all others good)	pq^{n-1}/s	$(\varepsilon + pq^{n-1})/s$	
(all remaining states)	ε/s		

Figure 5.1 Mixing Routine for $G(2,n)$ Situation under Procedure R_{00}.

The cost associated with reaching the ε-states is $(2\varepsilon + 3pq^{n-1})/s$. If we add to this the cost $h(2n-3)p^2q^{n-2}/s$ which is the future cost of determining which of these $2n-3$ states is the true state of nature and multiply by s to get $pG^*(2,n) = (1-q^2)G(2,n)$, then we obtain the contribution to the cost in the form

$$(5.3) \qquad (1-q^2)\ \text{Cost} = h(2)pq^{n-1} + [h(2n-3) + 2(2n-3)]p^2q^{n-2} + \mathcal{O}(p^3),$$

where $h(2) = 3$. Similarly, the corresponding result for $m = 3$ is

$$(5.4) \qquad (1-q^2)\ \text{Cost} = h(3)pq^{n-1} + [h(3n-6) + 3(3n-6)]p^2q^{n-2} + \mathcal{O}(p^3).$$

If we write $1-q^m$ in (3.4) as $(q+p)^{n-m}\,[(q+p)^m - q^m]$ and take the coefficient of p^2q^{n-2} after multiplying both sides of (3.4) by p, then we obtain for $m \geq 4$

$$(5.5) \qquad g(m,n) = \underset{2\leq x\leq a-1}{\text{Min}}\ [g(x,n) + g(m-x,\ n-x)] + \binom{n}{2} - \binom{n-m}{2}.$$

We shall not attempt to solve (5.5) directly as was done for (4.1) in lemmas 2 and 3 but after some more lemmas we do give explicit solutions for x and g(m,n), respectively, in theorems 3 and 4 below.

Let I_b for $b \geq 2$ denote the set of integers in the closed interval $[3 \cdot 2^{b-2},\ 3 \cdot 2^{b-1}]$ and let $I_\alpha (\alpha = 0,1)$ denote the integers in the closed intervals [0,1] and [1,3], respectively. Then the slope (or difference) of the function h(x) increases (when and) only when x is at the right end point of one of these intervals.

Lemma 5: If h(x) is a convex function (defined for integers or real numbers) and the vectors $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n)$ are such that

$$(5.6) \qquad x_1 \geq x_2 \geq \ldots \geq x_n;\ y_1 \geq y_2 \geq \ldots \geq y_n \text{ and}$$

$$(5.7) \qquad \sum_{i=1}^{j} x_i \leq \sum_{i=1}^{j} y_i \qquad j = 1,2,\ldots, n$$

with equality in (5.7) for $j = n$, then

$$(5.8) \qquad \sum_{i=1}^{n} h(x_i) \le \sum_{i=1}^{n} h(y_i).$$

For the $h(x)$ of section 4 with the vectors x and y not identical, strict inequality holds in (5.8) unless all the 2n components of x and y are in the same I_b.

Proof: The main result (5.8) is proved in Theorem 108 of Hardy, Littlewood and Polya [2] (see also Theorem 250) and is used extensively in the sequel.

For the second part of the lemma we first consider n-vectors of the form $x^0 = (x_1, x_2, \ldots, x_2)$ and $y^0 = (y_1, y_2, \ldots, y_2)$ with non-negative components. Set $d = n-1 \ge 1$ and hold $y_1 - dx_2 = x_1 - dy_2 = c$ fixed. We assume $x_2 \ge 1$ since otherwise $x_2 = y_2 = 0$ and the two vectors are identical. Applying the method of lemma 2 above to the function

$$(5.9) \qquad H(x_2) = h(c + dx_2) - dh(x_2),$$

we find that if x_2 and $c + dx_2$ are in the same I_b then $H(x_2) = H(x_2|b,b)$ has the constant value $c(b+2) + (d-1)3 \cdot 2^{b-1}$. If $b(x_2) = b \ge 1$ and $b(c+dx_2) = b + a \ge b + 1$ then we obtain for $H(x_2|b+a,b)$ a strictly increasing (in x_2 for fixed a and b) function

$$(5.10) \quad H(x_2|b+a,b) = c(b+2) + 3d \cdot 2^{b-1} + [a(c + dx_2) - 3 \cdot 2^{a+b-1}] \ge H(x_2|b,b).$$

This inequality is easily shown separately for $a = 1$ and $a \ge 2$ and equality holds only if $a = 1$ and $c + dx_2 = 3 \cdot 2^{b-1}$; i.e., strict inequality holds when and only when x_2 and $c + dx_2$ are not in the same I_b. If $x_2 \ge y_2$ (or $y_1 = c + dx_2 \ge c + dy_2 = x_1$), it follows that

$$(5.11) \qquad H(x_2) \ge H(y_2) = h(c + dy_2) - dh(y_2)$$

and strict inequality holds if and only if y_2 and $c + dx_2$ (and the other two arguments) are not in the same I_b. Rewriting our result, we have

$$(5.12) \qquad h(y_1) + (n-1)h(y_2) \ge h(x_1) + (n-1)h(x_2)$$

and if the vectors x^0 and y^0 are not identical then strict inequality holds in (5.12) if and only if all four arguments are not in the same I_b. It is easily noted that the two conditions $x_2 > y_2$ and $y_1 + dy_2 = x_1 + dx_2$ imply the

other inequalities. In particular, with $d = 1$ we obtain the desired result for $n = 2$; we need the general result for the induction below.

For general $n \geq 3$ we assume $y_1 > x_1$ since otherwise $y_1 = x_1$ and we can use the induction hypothesis. If $y_{n-1} > y_n$ we set $y'_{n-1} = y_{n-1} - \epsilon$, $y'_n = y_n + \epsilon$, and $y'_i = y_i (i = 1,2,\ldots, n-2)$, keeping the x-values fixed. Increasing $\epsilon \geq 0$ until some inequality becomes an equality, we either first get $y'_n = x_n$ in which case we have one less argument and can use the induction hypothesis or we first get $y'_{n-1} = y'_n$. If $y_{n-1} = y_n$ at the outset then we go immediately to the next step. In the next step we set $y''_{n-2} = y'_{n-2} - 2\epsilon$, $y''_j = y'_j - \epsilon$ for $j = n-1, n$ and $y''_i = y'_i$ $(i = 1,2,\ldots, n-3)$ and, using the induction argument, the only case that remains is the one in which the vector y has (at least) 3 equal, smallest components. We continue thus until we obtain a vector $y^0 = (y_1^0, y_2^0, \ldots, y_2^0)$ with $n - 1$ equal components. For each of these steps we can use our result in (5.12) to show that

$$(5.13) \qquad \sum_{i=1}^{n} h(y_i) \geq \sum_{i=1}^{n} h(y'_i) \geq \ldots \geq \sum_{i=1}^{n} h(y_i^0)$$

with strict inequality between the two extreme sums if and only if the original $y_i (i = 2,3,\ldots, n)$ are not all in the same I_b. Similarly, if $x_{n-1} < x_n$ we set $x'_{n-1} = x_{n-1} + \epsilon$, $x'_n = x_n - \epsilon$ and $x'_i = x_i (i = 1,2,\ldots, n-2)$. Applying the same methods as above we obtain a vector x^0 with the last $n-1$ components equal and using (5.12) we have

$$(5.14) \qquad \sum_{i=1}^{n} h(x_i) \leq \sum_{i=1}^{n} h(x'_i) \leq \ldots \leq \sum_{i=1}^{n} h(x_i^0),$$

where strict inequality holds between the two extreme sums if and only if the original $x_i (i = 2,3,\ldots, n)$ are not all in the same I_b. Comparing the last sums in (5.13) and (5.14) we note that by (5.12)

$$(5.15) \qquad \sum_{i=1}^{n} h(y_i^0) \geq \sum_{i=1}^{n} h(x_i^0)$$

with strict inequality if and only if the four arguments are not in the same

I_b. Combining those results after (5.13), (5.14) and (5.15) with 2 different vectors x,y, we obtain (5.8) with strict inequality if and only if the original $2n$ components are not all in the same I_b.

We need a final remark for the induction part of the proof. If, in reducing the problem from one with n components to one with $n-1$ components (and using induction), we change from a state in which all components are not in a common I_b to one in which they are, then the strict inequality shows up in a step corresponding to (5.13) or (5.14) and we use the weak inequality (5.8) for the induction step; if all arguments remain in their respective I_b's then the strict inequality shows up in the induction step.

Since lemma 5 was shown to hold for $n = 2$, the induction proof is now complete.

Define $r = r(m)$ by the inequalities

$$2^r \le m < 2^{r+1}. \tag{5.16}$$

Theorem 3: Under Assumption 1 the optimal test size $x = x(m)$ for a $G(m,n)$ situation with $n \ge m \ge 4$ and q close to one is given by

$$x = \begin{cases} 2^{r-1} & \text{for } 4 \cdot 2^{r-2} \le m < 5 \cdot 2^{r-2} \\ m - 3 \cdot 2^{r-2} & \text{for } 5 \cdot 2^{r-2} \le m < 7 \cdot 2^{r-2} \\ 2^r & \text{for } 7 \cdot 2^{r-2} \le m < 8 \cdot 2^{r-2} \end{cases} \tag{5.17}$$

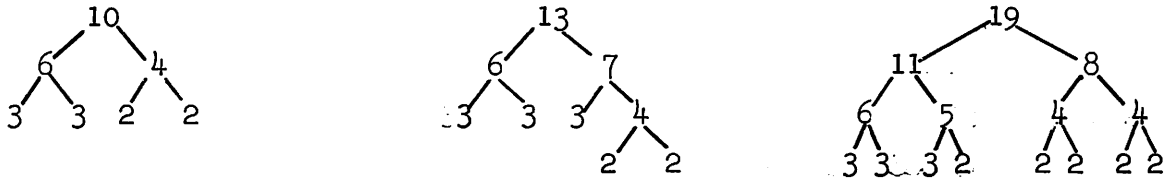
where r is defined by (5.16).

Remarks: We note that $x(m)$ increases by ones as m increases from $5 \cdot 2^{r-2}$ to $7 \cdot 2^{r-2}$ and remains fixed otherwise; in particular, it is a nondecreasing function of m. We also note that either x is a power of 2 (and then it remains a power of 2 at all later steps) or x is congruent to m modulo 3. For q close to one we start by testing all n units and if it fails we get a $G(n,n)$ situation. For any $G(m,n)$ situation reached after this if m is not a power of 2 then the integers m and n must be congruent modulo 3. It follows that the only $G(3,n)$ situations that can arise for q close to one are those in which n is a multiple of 3. This same phenomenon arises

for q close to 1 under Procedure R_0 defined in [8]. (See footnote to Table A7 in this paper.)

It should also be noted that for q close to one the expression for x in (5.17) does not agree with the results for $x = x(m)$ to be used for q close to one under Procedure R_1 in [7]. However it does agree with the results for q close to one under Procedure R_0 in [8] although the dividing points are not the same; this agreement has not been rigorously proved.

Proof of Theorem 3: Consider the triangular pattern (or arborescence) induced if we use (5.17) to define x and the recursion (3.4) to define our Procedure R_{00} for $m \geq 4$. For examples, if $m = 10$, 13 and 19, respectively, the arborescences are



m = 10 m = 13 m = 19

Figure 5.2 Breakdown of the m-value for $m = 10$, 13 and 19.

We note that the first argument is broken down until we reach a first argument equal to 2 or 3. We have arbitrarily put $x(m)$ corresponding to a test that fails below and to the right and $m - x(m)$ corresponding to a test that passes below and to the left in the above arborescences. We note that with the definition of $x(m)$ in (5.17) we have the following properties:

i) The two's all come out on the same level, say b.

ii) The three's all come out on the same level, say r.

iii) If $2^r < m < 3 \cdot 2^{r-1}$ then $b = r$ and all the two's fall to the right of the three's.

iv) If $3 \cdot 2^{r-1} < m < 2^{r+1}$ then $b = r+1$ and again the two's are on the right.

v) If $m = 3 \cdot 2^{r-1}$ we obtain only three's and if $m = 2^r$ we obtain only two's.

To see these properties we note from (5.17) that if m is in the closed interval $[3 \cdot 2^{b-2}, 3 \cdot 2^{b-1}]$ then x_1 and $m - x_1$ are both in the closed interval $[3 \cdot 2^{b-3}, 3 \cdot 2^{b-4}]$; we now strengthen this result.

If $3 \cdot 2^{b-2} \le m \le 2^b$ then by (5.17) we have for x_1

$$3 \cdot 2^{b-3} \le x_1 \le 2^{b-1} \quad \text{and} \quad 3 \cdot 2^{b-3} \le m-x_1 \le 2^{b-1} \tag{5.18}$$

and similar results hold for x_i and $m-x_i$ on the i^{th} level ($i \ge 1$). By (5.17) if any number in the arborescence is of the form 2^r or $3 \cdot 2^r$ then it gives rise only to numbers of the same form as itself. It follows that all the four's in the triangle (corresponding to $b = 3$ in (5.18)) have to line up only with any three's that arise and that no five's can arise in this case; thus $b = r+1$ as in property (iv).

If $2^b \le m \le 3 \cdot 2^{b-1}$ then by (5.17) we have for x_1

$$2^{b-1} \le x_1 \le 3 \cdot 2^{b-2} \text{ and } 2^{b-1} \le m-x_1 \le 3 \cdot 2^{b-2} \tag{5.19}$$

and similar results hold for x_i and $m-x_i$ on the i^{th} level ($i \ge 1$). It follows in this case that all the four's in the arborescence (corresponding to $b = 3$ in (5.19)) have to line up with any five's and six's that arise; thus the two's line up with the three's and $b = r$ as stated in property (iii).

To prove the rest of the 5 properties we note that by (5.17) either x_1 is a power of 2 or $m - x$ is 3 times a power of 2. In the first case the integer x is repeatedly halved so that we end up with two's on the right and if $m-x$ is not of the form 3 times a power of 2 then we repeat the argument with $m-x$ replacing m. In the second case $m-x$ is repeatedly halved giving us three's on the left and if x is not a power of 2 we repeat the argument with x replacing m. Since x and $m-x$ are both less than m and the properties hold for $m = 4$, the induction is completed.

We shall refer to the above arborescences as "optimal triangles." Another property they have (which we do not prove) is that (i) if any number is of the form $w = 2^t$ then all the numbers to the right of it (on the same level)

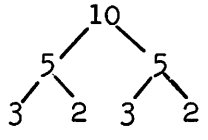
are equal to w and (ii) if any number is of the form $w = 3 \cdot 2^t$ then all the numbers to the left of it (on the same level) are equal to w.

It is quite clear that the specification of arborescences for each m (which give consistent instructions) is equivalent to a rule or a formula for $x = x(m)$. The two's and three's in each arborescence are pendant vertices and m is called the root; their relative position determines the cost of the procedure. We wish to show that the "optimal triangles" induced by (5.17) yield the lowest cost. For this purpose we consider any arbitrary rule which does not mix in the $G(m,n)$ situation for $m \geq 4$ and which yields a different configuration of pendant vertices (different from the optimal triangles above) in its arborescence for any root m. We wish to show that by means of a set of basic changes, each of which reduces (or does not increase) the cost, we can get from the arbitrary pattern to the optimal triangle. Consider the four changes in the following list.

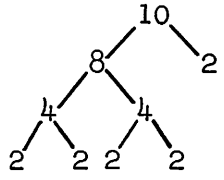
Figure 5.3 Arborescence Changes to Reduce Cost.

1. $(2,3) \rightarrow (3,2)$
2. $(4,3) \rightarrow (3,4)$
3. $(2,x) \rightarrow (3,x-1)$ and $(x,2) \rightarrow (x-1,3)$ for $x \geq 4$
4. $(3,x) \rightarrow (4,x-1)$ and $(x,3) \rightarrow (x-1,4)$ for $x \geq 5$.

The arrow points to the lower cost in each case. We shall prove some of these results (not all of them) but first we illustrate these changes. Suppose that with $m = 10$ we start by taking $x(10) = 5$ in one example (and $x(10) = 2$ in another example) and then use (5.17) in all later steps.



Example 1



Example 2

By change 1 in Figure 5.3 we can replace the middle pair (2,3) in example 1 by (3,2) thus giving the "optimal triangle." By change 3 we replace (8,2) by

(7,3) and then we use change 4 to replace it by (6,4) in example 2. This causes changes in the other numbers but not in the root $m = 10$. Clearly, the pair (6,4) gives the "optimal triangle."

To prove that change 1 in Fig. 5.3 reduces the cost we consider any pair (3,2). The contribution to the second term of the cost, using lemma 4 and starting from a $G(5,n)$ situation with $n_1 \geq 5$, is

$$(5.20) \qquad C_R = g(2,n_1) + g(3,n_1-2) + 5n_1 - 15$$
$$= h(2n_1 - 3) + h(3n_1 - 12) + 18n_1 - 57;$$

the same result also holds if the 2 and 3 stem from different predecessors in the arborescence. If we interchange the 2 and 3 the result depends on whether or not the 2 and 3 stem from the same number ($m = 5$) or from different numbers. In the first case, using lemma 4, we obtain

$$(5.21) \qquad C_L = g(3,n_1) + g(2,n_1 - 3) + 5n_1 - 15$$
$$= h(2n_1 - 9) + h(3n_1 - 6) + 18n_1 - 51.$$

To show that C_L is strictly larger than C_R we drop the difference 6 in the last two terms and it is sufficient to show that for $n_1 \geq 5$

$$(5.22) \qquad h(2n_1 - 3) + h(3n_1 - 12) \leq h(2n_1 - 9) + h(3n_1 - 6).$$

This follows from lemma 5 and the convexity of $h(x)$. In the second case we compare $(y,3;\ 2,x)$ and $(y,2;\ 3,x)$ where x,y are integers ≥ 2; the notation is clear from Fig. 5.3. For $(y,3;\ 2,x)$ the total contribution to the second term of the cost starting with a $G(x + y + 5,\ n_2)$ situations with $4 \leq x+y \leq n_2 - 5$, is

$$(5.23) \qquad C_R = g(x,\ n_2) + g(2,\ n_2 - x) + g(3,\ n_2 - x - 2) + g(y,\ n_2 - x - 5)$$
$$+ (x + y + 5)(2n_2 - x - y - 6)$$

and for $(y,2;\ 3,x)$ we obtain the comparable quantity

$$(5.24) \qquad C_L = g(x,\ n_2) + g(3,\ n_2 - x) + g(2,\ n_2 - x - 3) + g(y,\ n_2 - x - 5)$$
$$+ (x + y + 5)(2n_2 - x - y - 6).$$

Hence to show that C_L is strictly larger than C_R we use lemma 4, drop the "extra" 6 that arises for C_L and it is sufficient to show that for $n_2 \geq x + 3$

$$(5.25) \qquad h(2n_2 - 2x - 3) + h(3n_2 - 3x - 12) \leq h(2n_2 - 2x - 9) + h(3n_2 - 3x - 6).$$

This again follows from lemma 5 and the convexity of $h(x)$.

For change 2 in Fig. 5.3 we again examine the contribution to the second term of the cost. For (3,4) we obtain, starting with a $G(7,n_3)$ situation with $n_3 \geq 7$,

$$(5.26) \qquad \begin{aligned} C_R &= g(2, n_3) + g(2, n_3 - 2) + g(3, n_3 - 4) + 11n_3 - 38 \\ &= h(3n_3 - 18) + h(2n_3 - 3) + h(2n_3 - 7) + 28n_3 - 112; \end{aligned}$$

the result is the same if the 4 and 3 stem from different predecessors. If we interchange 3 and 4 the result depends on whether the 3 and 4 stem from the same number ($m = 7$) or from different numbers. In the first case we obtain

$$(5.27) \qquad \begin{aligned} C_L &= g(3, n_3) + g(2, n_3 - 3) + g(2, n_3 - 5) + 11n_3 - 50 \\ &= h(3n_3 - 6) + h(2n_3 - 9) + h(2n_3 - 13) + 28n_3 - 112. \end{aligned}$$

Unfortunately the last 2 terms are the same in (5.26) and (5.27) and (5.8) only gives the weak inequality $C_R \leq C_L$. The last part of lemma 5 still allows possibilities of equality for $n_3 = 31, \ldots, 34, 55, \ldots, 66$, etc.

However, we remark that these will not lead to any inconsistency under the proposed rule since if we start with an H-situation and use (5.17) for q close to 1 then a $G(7,n)$ situation can arise only if $n = 2(7) - 1 = 13$ or $2(13) - 1 = 25$ or etc. and these are always of the form $3 \cdot 2^s + 1$ with $s \geq 1$. Hence we have for the first (A_1) and the third (A_3) arguments in (5.27) with $n_3 = 3 \cdot 2^s + 1$,

$$(5.28) \qquad A_1 = 9 \cdot 2^s - 3 = 3 \cdot 2^{s+1} + 3(2^s - 1) > 3 \cdot 2^{s+1} > 3 \cdot 2^{s+1} - 11 = A_3.$$

Strict inequality now follows from the second part of lemma 5, i.e., $C_R < C_L$.

In the second case we compare $(y,3;\ 4,x)$ with $(y,4;\ 3,x)$ where x,y are integers ≥ 2. The only difficult case is $y = 4$ and $x = 3$ where (5.8) only gives the weak inequality. As above we note that for q close to one a $G(14,n)$ situation arises only for $n = 2(14) - 2 = 26$ or $2(26) - 2 = 50$ or etc. and these are always of the form $3 \cdot 2^s + 2$ with $s \geq 2$. Hence we have for the largest $A_1 = 3n_3 - 6$ and the smallest argument $A_6 = 2n_3 - 31$ with $n_3 = 3 \cdot 2^s + 2$,

$$(5.29) \qquad A_1 = 9 \cdot 2^s = 3 \cdot 2^{s+1} + 3 \cdot 2^s > 3 \cdot 2^{s+1} > 3 \cdot 2^{s+1} - 27 = A_6.$$

Thus the strict inequality $C_R < C_L$ follows from the second part of lemma 5.

For the change $(2,x) \to (3,x-1)$ with $x \geq 4$ it is sufficient to look at the contribution to the first coefficient. For $(2,x)$ we obtain $x + 5 + h(x)$ and for $(3,x - 1)$ we obtain $x + 8 + h(x - 1)$ and it is sufficient to show that $h(x) - h(x - 1) > 3$ for $x \geq 4$. This follows from the facts that $\Delta h(x)$ is increasing (proved in lemma 5) and $h(4) - h(3) = 4$. The same argument applies to the change $(x,2) \to (x - 1,\ 3)$ and also to the cases in which the integers 2 and x stem from different predecessors in the arborescence.

The same proof also holds for the change $(3,x) \to (4,\ x - 1)$ with $x \geq 5$ (the case $x = 4$ which only gives equality was discussed above) and the change $(x,3) \to (x - 1,\ 4)$ for $x \geq 5$.

We have shown that the definition of $x(m)$ by (5.17) for q close to one induces a particular type of arborescence with 5 properties. Any other definition which gives a different pattern yields an increase in the first or second coefficient of the total cost starting from an H-situation. In this sense the definition $x(m)$ in (5.17) is optimal for q close to one; this completes the proof of Theorem 3.

One consequence of (5.17) which will be useful is

Corollary 3.1: Under Procedure R_{00} for $m \geq 4$ and q close to one

$$(5.30) \qquad \frac{2}{5} \leq \frac{x}{m} \leq \frac{4}{7}$$

and the upper bound holds with (x,m) replaced by (m,n), respectively, if we start with an H-situation.

Proof: It is easy to see from (5.17) that $(m-x)/m$ has a maximum when $m = 5 \cdot 2^{r-1}$ and a minimum when $m = 7 \cdot 2^{r-1}$, and these values are $3/5$ and $3/7$, respectively. The result (5.30) is then immediate. If we start with an H-situation with q close to one then we test all n units and if it fails we have a $G(n,n)$ situation. Hence $x/n \leq 4/7$ and, since the x becomes a future m-value, it is clear that the same inequality holds for m/n; this proves the corollary.

If we need to prove that a result holds for a $G(7,n)$ situation with $n \neq 7$ then by this corollary we need only show it for $n \geq 7(7)/4$ or for $n \geq 13$; this illustrates the use we will make of the corollary.

Theorem 4: The coefficient $g(m,n)$ of p^2q^{n-2} in $(1 - q^m)G(m,n) = pG^*(m,n)$, which together with the definition of $x = x(m)$ in (5.17) constitutes a solution of (5.5), is given by

$$(5.31) \qquad g(m,n) = g^*(m,n) + (r+2)[mn - \binom{m+1}{2}] + m^*(2m^* + 1 - 2n)]$$

where r is defined by (5.16), $m^* = \text{Max}\,[3 \cdot 2^{r-1} - m, 0]$,

$$(5.32) \qquad g^*(m,n) = \sum_{i=1}^{d} h(3n - 3m - 6 + 9i) + \sum_{i=1}^{(m-3d)/2} h(2n + 1 - 4i),$$

$d(m) = d$ is the distance from m to the nearest power of 2, and $h(x)$ is defined in (4.4).

For example, if $m = n = 7$ then $r = 2$, $x(7) = 4$, $d = 1$, $m^* = 0$, and the coefficient of p^2q^5 is

$$(5.33) \qquad g(7,7) = h(11) + h(7) + h(3) + 84 = 156.$$

If we took 3 units for the first test instead of 4 and then use (5.17) afterwards then we obtain

$$(5.34) \qquad g(3,7) + g(4,4) + 21 = h(15) + h(5) + h(1) + 84 = 164,$$

where (5.31) is used to obtain $g(4,4)$.

The coefficient of pq^6 is 23 for both cases.

Proof of Theorem 4:

We use the recursion formula (5.5) and the definition of $x(m)$ in (5.17) in this proof by induction. By Lemma 4, the result (5.31) holds for $m = 2$ and $m = 3$. Assuming it holds when the first argument is less than m, we substitute (5.31) in both places on the r.h.s. of (5.5) and show that it also holds for m. We consider four disjoint and exhaustive cases according as $j \cdot 2^{r-2} \le m < (j+1) \cdot 2^{r-2}$ for $j = 4, 5, 6$ or 7 with arbitrary $r \ge 2$.

Using (5.17) we note that in Case 1 we have $2^{r-1} = x \le m/2 \le m - x < 3 \cdot 2^{r-2}$ so that in terms of $r = r(m)$ we have $r(x) = r(m - x) = r-1$. From (5.31) and the r.h.s. of (5.5) we obtain by straightforward (but tedious) algebra

$$(5.35) \quad \sum_{i=1}^{2^{r-2}} h(2n + 1 - 4i) + (r+1)[n2^{r-1} - \binom{2^{r-1}+1}{2}] + 2^{r-2}(2^{r-1} + 1 - 2n)$$

$$+ \sum_{i=1}^{5\cdot 2^{r-2}-m} h(2n+1 - 4i - 2^r) + \sum_{i=1}^{m-2^r} h(3n - 3m - 6 + 9i) + mn - \binom{m+1}{2}$$

$$+ (r+1)[m-2^{r-1})(n-2^{r-1}) - \binom{m+1-2^{r-1}}{2})] + (5 \cdot 2^{r-2} - m)(7 \cdot 2^{r-1} + 1 - 2m - 2n)$$

$$= \sum_{i=1}^{m-2^r} h(3n - 3m - 6 + 9i) + \sum_{i=1}^{3\cdot 2^{r-1}-m} h(2n + 1 - 4i)$$

$$+ (r+2)[mn - \binom{m+1}{2}] + m^*(2m^* + 1 - 2n),$$

which agrees with the result for $g(m,n)$ in (5.31). We omit the algebra for the remaining three cases since the computation is similar to that above. This completes the proof of the theorem.

Corollary 4.1: If we define $g^*(m,n)$ by (5.31) then the recursion (5.5) becomes

$$(5.36) \qquad g^*(m,n) = \min_{2 \le x \le m-1} [g^*(x,n) + g^*(m-x, n-x)]$$

and the solution is given by (5.17) and (5.32).

Proof: It suffices to consider the same four cases as in Theorem 4. For example, for Case 3 we have $6 \cdot 2^{r-2} \le m < 7 \cdot 2^{r-2}$ so that $m^*(m) = m^*(x) = m^*(m-x) = 0$ and $r(x) = r(m-x) = r(m) - 1$. Substituting (5.31) into the r.h.s. of (5.5) gives for any $r \ge 2$ and any $x \ge 2$

$$(5.37) \qquad \begin{aligned} &g^*(x,n) + (r+1)[xn - \tbinom{x+1}{2}] + g^*(m-x, n-x) + (r+1)[(m-x)(n-x) - \tbinom{m-x+1}{2}] \\ &\quad + mn - \tbinom{m+1}{2} = g^*(x,n) + g^*(m-x, n-x) + mn - \tbinom{m+1}{2} = g^*(m,n). \end{aligned}$$

A similar computation holds for the other 3 cases and this proves the corollary.

6. On the Problem of Removing Assumption 1.

We would like to show in this section that other modification 1 procedures that (unlike Procedure R_{00}) mix units from the defective and binomial set in a $G(m,n)$ situation with $m \ge 4$ are not better than the proposed Procedure R_{00}. A complete definition of the latter is given in Section 8, but we will only use (5.17) and its consequences.

We define $m \ge 2$ (and also n) to be of type 1 if for some $b \ge 2$

$$(6.1) \qquad 3 \cdot 2^{b-2} \le m \le 2^b$$

and to be of type 2 otherwise; thus, $m = 5$ is the smallest m-value of type 2. We define a procedure to be A2 optimal, or asymptotically $(q \to 1)$ 2^d degree optimal, if its cost has the smallest first coefficient and, among procedures with the same smallest first coefficient, its cost has the smallest second coefficient. In this section we will prove by induction that if we start with n units, where n is of type 1, then Procedure R_{00} is A2 optimal; the main idea here is to show that we can remove assumption 1 if n is of type 1. We also show with counterexamples that if n is of type 2 then this result does not hold; thus Procedure R_{00} is not optimal if q is close to 1 and n is of type 2.

Let $x = x_{00}(m)$ denote the integer obtained from (5.17). We shall refer to the long chain from the upper left to the lower right in our schematized diagrams (see e.g. Figure 6.1) as the main diagonal. To explain the proof and also start the induction we consider the special cases $m = 4, 6, 7, 8$; the cases $m = 4$ and $m = 7$ are treated in more detail.

We start with a $G(4,n)$ situation with $4 < n$ and consider the Procedure R_M which starts by testing $(2 + t)$ units, i.e., two from the defective set of size 4 and t $(0 \le t \le n-4)$ from the binomial set; for $t > 0$ it is a mixing procedure and for $t = 0$ it reduces to R_{00}. The scheme is given in Figure 3.1. The first coefficient is $h(4) = 10$ for any $t \ge 0$ and the second coefficient for R_M is

$$h(2n + 2t - 3) + h(2n - 2t - 7) + 12n - 30 \tag{6.2}$$

and the same result holds for R_{00} with $t = 0$. By Lemma 5

$$h(2n + 2t - 3) + h(2n - 2t - 7) \ge h(2n - 3) + h(2n - 7) \tag{6.3}$$

and this shows that no R_M is better (with respect to the first two coefficients) than R_{00}.

For a $G(6,n)$ situation with $6 < n$ we start with $(3 + t)$ units in the first test and obtain $h(6) = 18$ for the first coefficient for any $t \ge 0$ and for the second coefficient under R_M we obtain

$$C_2(R_M) = h(3n + 3t - 6) + h(3n - 3t - 15) + 24n - 84. \tag{6.4}$$

Lemma 5 again proves that this is a minimum for $t = 0$.

For a $G(7,n)$ situation with $7 < n$ we start by testing $(4 + s)$ units. If it fails we then test $(2 + t)$ units where the s and t binomials can have c units in common. The scheme is given in Figure 6.1.

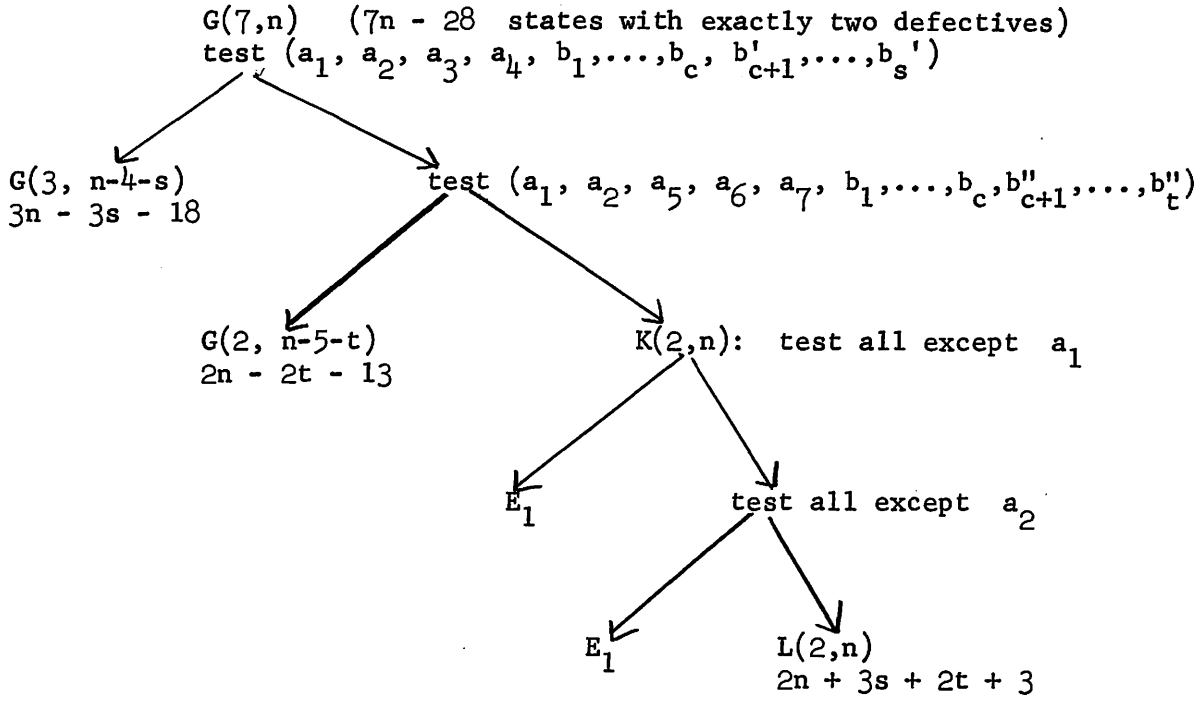


Figure 6.1: Procedure R_M for $m = 7 \leq n$.

The first coefficient for R_M is $h(7) = 23$ for any $t \geq 0$. The second coefficient of the cost of R_M is given by

$$C_2(R_M) = h(A_1) + h(A_2) + h(A_3) + 28n - 112 \tag{6.5}$$

where $A_1 = 2n + 3s + 2t + 3$, $A_2 = 3n - 3s - 18$ and $A_3 = 2n - 2t - 13$. The same result holds for R_{00} with coefficients A_1', A_2', A_3' obtained from those above by setting $s = t = 0$.

We have already noted that for $m = 7$ and q close to one the possible values of n are of the form $3 \cdot 2^b + 1$ with $b \geq 2$. For $b = 2$ and $n = 13$ we have $A_1' > A_2' > A_3'$ and $A_1 > \max(A_2, A_3)$. If $A_2 \geq A_3$, then we have $A_1 \geq A_1'$, $A_1 + A_2 \geq A_1' + A_2'$ and $A_1 + A_2 + A_3 = A_1' + A_2' + A_3'$ so that Lemma 5 can be used to show that $C_2(R_M) \geq C_2(R_{00})$. If $A_3 > A_2$ then $A_1 + A_3 > A_1 + A_2 \geq A_1' + A_2'$ and Lemma 5 again applies.

Suppose now that $b \geq 3$ so that $n \geq 25$ and $A_2' > A_1' > A_3'$. Although we know that $A_1 > A_3$ we have to consider two cases according as $A_2 \geq A_1 > A_3$ or $A_1 \geq \max(A_2, A_3)$.

Case 1: $A_2 \geq A_1$ or $n \geq 6s + 2t + 21$

We note that $A_1' = 3 \cdot 2^{b+1} + 5 \in I_{b+1}$, $A_2' = 3 \cdot 2^{b+1} + 3 \cdot 2^b - 15 \in I_{b+1}$ and we wish to show that $A_1 \in I_{b+1}$ and $A_2 \in I_{b+1}$. Using the fact that $n \geq 6s + 2t + 21$ we have $A_2 = 3 \cdot 2^{b+1} + 3 \cdot 2^b - 15 - 3s \geq 3 \cdot 2^{b+1} + 3 \cdot 2^{b-1} - 5 + t$ and since $b \geq 3$ and $A_2 \leq A_2'$ we have $A_2 \in I_{b+1}$. Since $A_1 \leq A_2$ and $A_1 = 3 \cdot 2^{b+1} + 5 + 3s + 2t$ we also have $A_1 \in I_{b+1}$. It is also clear that $A_2^* = A_2 + 3s \in I_{b+1}$ and $A_1^* = A_1 - 3s \in I_{b+1}$ and hence by Lemma 5

$$h(A_1) + h(A_2) = h(A_1^*) + h(A_2^*). \tag{6.6}$$

Now $A_2^* = A_2'$, $A_2^* + A_1^* = A_1 + A_2 \geq A_2' + A_1'$ and $A_2^* + A_1^* + A_3^* = A_2 + A_1 + A_3$ so that $C_2(R_M) \geq C_2(R_{00})$ since by Lemma 5

$$\sum_{i=1}^{3} h(A_i) = h(A_1^*) + h(A_2^*) + h(A_3^*) \geq h(A_1') + h(A_2') + h(A_3'). \tag{6.7}$$

Case 2: $A_1 > \text{Max}\ (A_2, A_3)$

If $A_1 \geq A_2'$ then we can apply Lemma 5 since $A_1 + A_2 \geq A_2' + A_1'$ and $A_1 + A_2 + A_3 = A_1' + A_2' + A_3'$. Otherwise $A_2' \geq A_1$ or $n \geq 3s + 2t + 21$. As in Case 1 we have $A_1' \in I_{b+1}$ and $A_2' \in I_{b+1}$. Then $A_2 = 3 \cdot 2^{b+1} +$ $+ 3 \cdot 2^b - 15 - 3s \geq 3 \cdot 2^{b+1} + 2t + 5 \in I_{b+1}$ and $A_1 = 3 \cdot 2^{b+1} + 5 + 3s + 2t$ $\leq 3 \cdot 2^{b+1} + 3 \cdot 2^b - 16 \in I_{b+1}$. Hence the same argument as in Case 1 applies. This completes the proof for $m = 7$.

For $G(8,n)$ with $8 < n$ the same result $C_2(R_M) \geq C_2(R_{00})$ holds. Although we now get four h-functions, none of the above difficulties arise and we omit the proof.

We note that in each of the above cases ($m = 4, 6, 7, 8$) the size of the first group-test has two components and the first component (or the number of units taken from the defective set) is $x(m)$ given by (5.17). This value of $x(m)$ is used for two reasons. First, we can only use values which minimize the first coefficient of the cost and $x(m)$ accomplishes this. Second, if we used different values, say $x'(m)$, and denote the resulting

mixing procedure by R_M', then we could use the above analysis to find another non-mixing procedure, say R', that is as good or better than R_M'. It then follows from the results of Section 5 that Procedure R_{00}, which is the best of the non-mixing procedures, must be at least as good as R'. For example, suppose that for the case $m = 7$ treated above we started with $x'(7) = 3$ units from the defective set. Then the three arguments of the h functions are $3n + 4s - 6$, $2n - 2s - 9$, $2n - 2s - 13$ and it is easily seen that $C_2(R_M') \geq C_2(R')$. Later we briefly indicate a generalized version of this argument to any m value of type 1.

Consider now any $m_o \geq 12$ of type 2. The cases $m_o = 2^r$ and $m_o = 3 \cdot 2^{b-2}$ present little difficulty although the latter have to be treated separately. We first assume that $m_o = 3 \cdot 2^{b-2} + c$ with $b \geq 4$ and $c \geq 1$.

Theorem 5: If we compare Procedure R_{00} with the best mixing Procedure R_M that has the same first coefficient as Procedure R_{00}, then, starting with a G(m,n) situation with $m < n$ and q close to 1,

(6.8) $\quad C_2(R_{00}) \leq C_2(R_M)$ if m is of type 1.

Proof: The proof is by induction on m; suppose we start with m_o of type 1. We use the notation $x_1 = x(m_o)$, $x_2 = x(x_1)$, $x_3 = x(x_2)$, etc., and let t_i denote the number of binomial units in the i^{th} test on the main diagonal. Omitting the breakdown of all G(m,n) situations with $m < m_o$ and also the steps from K(2,n) to L(2,n), the mixing procedure R_M is schematized in Figure 6.2.

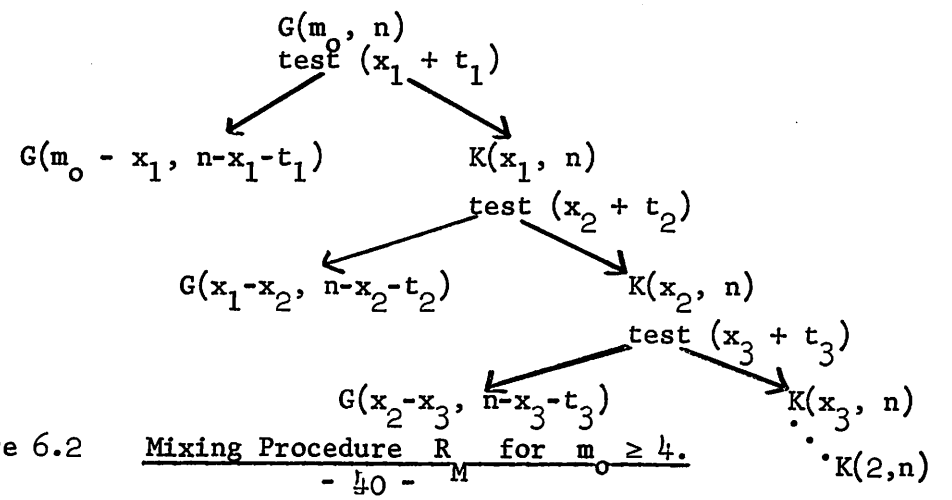


Figure 6.2 Mixing Procedure R_M for $m_o \geq 4$.

By the induction hypothesis we do not have to consider mixing in any of the $G(m,n')$ situations with $m < m_o$ in Figure 6.2. If we now break down all these $G(m,n')$ using (5.17) then we obtain $R = 2^b - m_o$ pendant vertices of the form $G(3,n')$, $B-1 = 2(m_o - 3 \cdot 2^{b-2}) - 1$ pendant vertices of the form $G(2,n')$ and one of the form $K(2,n)$. The values of n' for the first R pendant vertices are $n - t_1 - (m_o-3),\ldots,n - t_\alpha - 4c$ where $1 \le \alpha \le b-1$ and the corresponding numbers of states of nature with exactly two defectives are $N_1 = 3n - 3t_1 - 3m_o + 3,\ldots,N_R = 3n - 3t_\alpha - 12c - 6$.

If $7 \cdot 2^{b-3} \le m \le 2^b$ then the corresponding number of states for $K(2,n)$ is

$$N_0 = 2n - 3 + 3ct_1 + 3\sum_{j=2}^{\alpha} 2^{b-1-j}t_j + 2\sum_{j=\alpha+1}^{b-1} 2^{b-1-j}t_j. \tag{6.9}$$

Here α is the smallest integer such that $2^{b-2-\alpha} \le c$; since $c \ge 1$ and $b \ge 3$ we have $1 \le \alpha \le b-2$. If $N_0 \ge N_i$ $(i=1,2,\ldots,R)$ then we have no difficulty in showing that $C_2(R_M) \ge C_2(R_{00})$. Assuming that $N_0 \le N_j$ for some j we wish to show that N_0 and N_j belong to the same I-interval. If $m_o = 3 \cdot 2^{b-2} + c$ for $0 < c < 2^{b-2}$ and q is close to one, then we can assume that $n = 3 \cdot 2^s + c$ for some $s \ge b-1$ and the same c. We can assume that $t_1 \ge 1$, since if $t_1 = 0$ we could again invoke the induction hypothesis. For $c \ge 1$ and $t_1 \ge 1$ we have $N_0 \le N_R$ and

$$N_0 \ge 3 \cdot 2^{s+1} + [2c - 3 + 3ct_1] + 3\sum_{j=2}^{\alpha} 2^{b-1-j}t_j \ge 3 \cdot 2^{s+1}, \tag{6.10}$$

$$N_R = 3 \cdot 2^{s+1} + 3 \cdot 2^s - 9c - 3t_\alpha - 6 \ge 3 \cdot 2^{s+1} + [2c-3+3ct_1] \ge 3 \cdot 2^{s+1}. \tag{6.11}$$

It follows from these inequalities that N_0 and N_R both belong to I_{s+1}; it also follows that $N_R^* = N_R + 3t_\alpha$ and $N_0^* = N_0 - 3t_\alpha$ both belong to I_{s+1}. It follows that if $N_0 \le N_j$ for any j $(j = 1, 2,\ldots,R)$ then $N_0 \le N_j \le N_R$ and N_j is also in I_{s+1}. Moreover if N_j contains the term $3t_\beta$ then $N_j^* = N_j + 3t_\beta$, $N_0^* = N_0 - 3t_\beta$ and $N_0^{**} = N_0 - 3t_\alpha - 3t_\beta \le N_0^*$ are all in I_{s+1}.

It follows that

$$(6.12)\qquad h(N_0) + h(N_R) + h(N_j) = h(N_0^*) + h(N_R^*) + h(N_j)$$

$$= h(N_0^{**}) + h(N_R^*) + h(N_j^*)$$

$$\geq h(N_0') + h(N_R') + h(N_j'),$$

where N_j' are the corresponding N-values for the non-mixing Procedure R_{00}. Continuing in this manner and letting N_{R+i} $(i = 1, 2,\ldots,B-1)$ denote the N-values for the $G(2,n')$ pendant vertices, we easily obtain with the help of Lemma 5

$$(6.13)\qquad C_2(R_M) = \sum_{j=0}^{R+B-1} h(N_j) + (b+1)[mn - \binom{m+1}{2}]$$

$$\geq \sum_{j=0}^{R+B-1} h(N_j') + (b+1)[mn - \binom{m+1}{2}] = C_2(R_{00}).$$

If $3 \cdot 2^{b-2} < m < 7 \cdot 2^{b-3}$ and $b \geq 4$ then the only change in (6.9) is that the coefficients of the t_i are rotated so that the coefficients of $t_1, t_2,\ldots$ are $3 \cdot 2^{b-3}$, $3 \cdot 2^{b-4},\ldots,$ and $3c$ is the coefficient of t_γ where $\gamma = 2^{b-3} - c + 1$; the same changes take place in the N'-values. Since $b \geq 4$, $c \geq 1$ and $t_1 \geq 1$ we again have $2c - 3 + 3 \cdot 2^{b-3} > 0$ and the same proof holds.

If $m = 3 \cdot 2^{b-2}$ then we terminate the main diagonal in Figure 6.2 with $K(3,n)$ and all the pendant vertices are of the form $G(3,n')$. Then $N_0 > N_i$ for all i $(i = 1, 2,\ldots,b-1)$ and the proof that $C_2(R_M) \geq C_2(R_{00})$ is then straightforward; we omit the details.

To complete the proof of Theorem 5 we remark that if the x-values are not taken from (5.17) but are subject only to the condition that the first coefficient of the cost is $h(m_o)$ then all of the five properties of the arborescence noted in Section 5 hold except that the order of the two's and three's is arbitrary. Essentially the same proof as above shows that for any

mixing Procedure R_M' the corresponding non-mixing Procedure R' is at least as good. Since R_{00} is the best of the non-mixing procedures it follows that $C_2(R') \geq C_2(R_{00})$; this completes the proof of Theorem 5.

To show that the condition that m be of type 1 is necessary, we now give a mixing procedure, say R_M^o, for $m = 5$ that improves on R_{00} for q close to one. To be specific we take $n = 11$ and show the scheme in Figure 6.3.

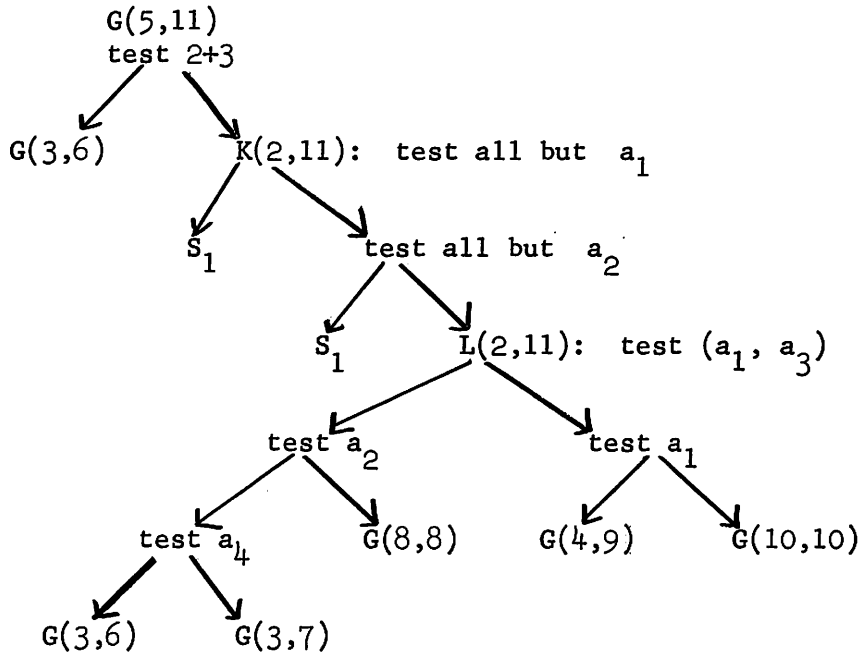


Figure 6.3 Mixing Routine R_M^o for $m = 5$.

The first coefficient of the cost for R_M^o is easily seen to be $h(5) = 14$. The number of states of nature with exactly two defectives for each of the six pendant vertices is 12, 3, 3, 8, 4 and 10 which sum to $55 - \binom{6}{2} = 40$. The second coefficient of the cost for R_M^o is

$$C_2(R_M^o) = h(12) + 2h(3) + h(8) + h(4) + h(10) + 48 + 2(18) + 40 + 20 + 50 = 330; \tag{6.14}$$

the corresponding second coefficient of the cost for R_{00} is

$$C_2(R_{00}) = h(21) + h(19) + 4(21) + 3(19) = 102 + 90 + 141 = 333. \tag{6.15}$$

It should be noted that R_M^o is not necessarily the best mixing procedure for $G(5,11)$ but it has been shown to be better than R_{00} for q sufficiently close to one.

The results of this section are used as a basis for conjecturing that R_{00} is an optimal procedure if we start with a number of units N which is of type 1.

The explicit formulas for $G_{00}^*(5,11)$ and $G_M^*(5,11)$ under Procedures R_{00} and the above mixing Procedure R_M^o, respectively, are given as polynomials in increasing powers of q by the coefficients

$$(6.16) \qquad G_{00}^*(5,11) = [43, 6, -3, -1, -6, -2, -12, -5, -16, 21, -11],$$

$$(6.17) \qquad G_M^*(5,11) = [44, 6, -3, -2, -8, -2, -8, 3, -30, 15, -1].$$

Using the basic recursion formulas given in (8.1) and (8.2) below we can then write for q close to one

$$(6.18) \qquad H_{00}(11) = 1 + pG_{00}^*(11,11) = 2 - q^{11} + pq^5 G_{00}^*(6,6) + pG_{00}^*(5,11)$$

$$= [45, -37, -9, 2, -5, 23, -21, -2, -10, 27, -20, 8],$$

$$(6.19) \qquad H_M(11) = [46, -38, -9, 1, -6, 25, -17, 2, -32, 35, -4, -2].$$

An examination of the difference Δ of (6.19) and (6.18) yields

$$(6.20) \qquad \Delta = (1-q)^2(1 + q + q^2 - 2q^4 - 2q^5 + 2q^6 + 10q^7 - 4q^8 - 10q^9),$$

which is of interest only in the interval $.975 < q < 1.000$. The value of Δ throughout this interval is negative and at least $-.001$; this gives some idea of the kind of improvement that can be expected by such mixing procedures.

7. Analysis of the G(2,n) and G(3,n) Situations for q Close to One.

We have already seen that in a $G(2,n)$ situation the optimal procedure for q close to one is to first test all units except a_1 and if that fails we test all units except a_2. If either test passes we are finished; if both fail we refer to the resulting situation as a $J(2,n)$ situation. For q close to one the relation between the expected number of tests $G(2,n)$ for the $G(2,n)$ situation and $J(2,n)$ for the $J(2,n)$ situation is

$$(7.1) \qquad pG^*(2,n) = 3pq^{n-1} + [p^2 + 2pq(1 - q^{n-2})]\,[2 + J(2,n)]$$

where $G^*(2,n)$ is defined in (3.4) (see also Table A2). This can also be written for q close to one as

$$(7.2) \qquad J^*(2,n) = G^*(2,n) + q^{n-1} - 2(1 + q) = pJ^{**}(2,n).$$

where $J^*(2,n)$ and $J^{**}(2,n)$ are defined in terms of $J(2,n)$ by

$$(7.3) \qquad J^*(2,n) = [p + 2q(1 - q^{n-2})]\,J(2,n) = pJ^{**}(2,n).$$

We denote the two units that are known to contain a defective by $a_i (i = 1,2)$ and the remaining units by $b_i (i = 1,2,\ldots,n-2)$.

In a $J(2,n)$ situation the number of states with maximal probability when q is close to one is $2n-3$. One possible continuation at this point is to test a_1 alone and this leads to a $G(n-2,n-2)$ or a $G(n-1,n-1)$ situation according as a_1 is good or bad. For both of these we have an optimal way of proceeding since the test on a_1 cuts down the M-set with $2n-3$ states into $n-2$ and $n-1$ states. It is easily seen that no other test that includes a_1 or a_2 will be as good as testing a_1 alone (or equivalently a_2 alone); we refer to such a test as being of type 1.

However, we must also consider tests that include only units from the set $b_1, b_2, \ldots, b_{n-2}$, i.e., that exclude a_1 and a_2. The same argument that yielded the result for $x(m)$ in (5.17) also shows that the number of units y from this set should be such that the $2n-3$ states are partitioned (as closely

as possible) into $x = x(2n-3)$ and $2n-3-x(2n-3)$ states. If we take y units, then the number of states corresponding to a successful test is $2(n-2-y) + 1$ and setting this equal to $2n-3-x$ gives $y = x/2$; we refer to such a test as being of type 2. An optimal test must be of type 1 or 2 and we are now interested in comparing these two types.

We will show by induction that if n is of the form $2^s + 2$ then a particular test of type 2 is better than any type 1 test.

Theorem 6: If we have a $J(2,n)$ situation where n is of the form $2^s + 2$ for some integer $s \geq 1$ then for q close to one there is a procedure of type 2 that starts by testing

$$(7.4) \qquad y = \frac{x(2^{s+1}+1)}{2} = 2^{s-1}$$

units, excluding a_1 and a_2, which is better than any procedure of type 1.

Proof: We assume that for $n_1 = 2^t + 2$ (with $t < s$) we can find a procedure of type 2 for the $J(2,n_1)$ situation which is superior to the type 1 procedure. We show this later for $t = 1$.

The optimal procedure of type 2 for the $J(2,n)$ situation has the form

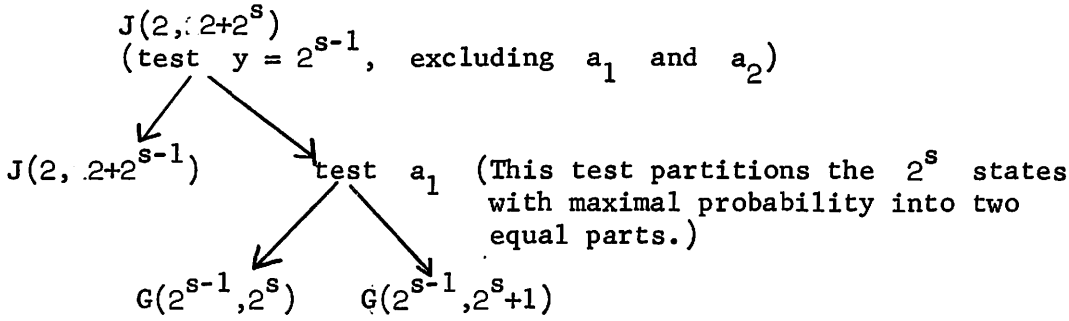


Figure 7.1 Optimal Test of Type 2 for $J(2,n)$ Situation.

Here the left arrows denote a successful test and the right arrows denote an unsuccessful test. For q close to one, the optimal procedure of type 1 has the form

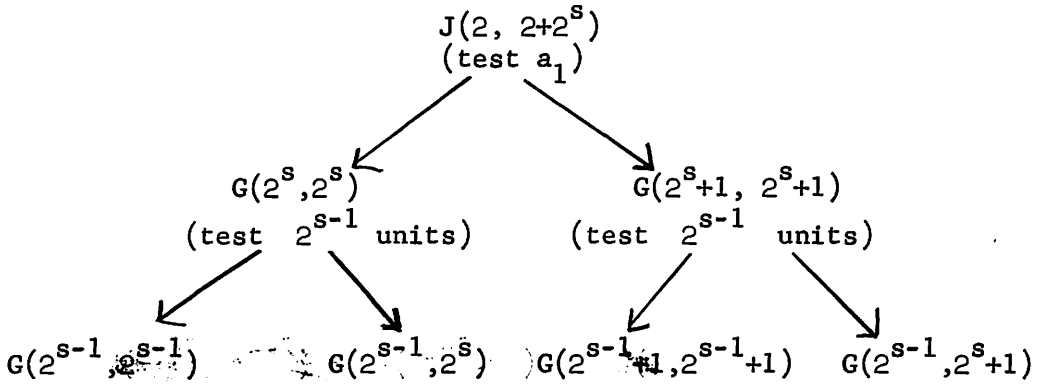


Figure 7.2 Optimal Test of Type 1 for J(2,n) Situation.

If the event $J(2, 2+2^{s-1})$ in Figure 7.1 is analyzed one step further by testing a_1 then we obtain the two results $G(2^{s-1},2^{s-1})$ and $G(2^{s-1}+1,2^{s-1}+1)$ and this makes the last line of Figure 7.1 identical with that of Figure 7.2, but by the induction hypothesis we can do better than this. Hence to complete the proof we need only show that for $s = 1$ we can do better by testing b_1 than by testing a_1. In fact we obtain for $s = 1$ from the procedure in Figure 7.1 and using Table A4

$$(7.5) \qquad J^*(2,4) = \{2p^2q^2 + p(1 - q^2)[1 + G(2,3)] + pq(1 - q^2)[2 + G(2,2)\}/p$$

$$= 14pq^2 + 19p^2q + 6p^3$$

and by testing a_1 we obtain for the procedure in Figure 7.2 with $s = 1$

$$(7.6) \qquad \{pq(1 - q^2)[1 + G(2,2)] + p(1 - q^3)[1 + G(3,3)]\}/p$$

$$= 14pq^2 + 21p^2q + 7p^3,$$

which is uniformly larger; this completes the proof of theorem 6.

It can also be shown by a similar induction argument that if $x(2n-3)$ is a power of 2 (say 2^s) and n is not of the form $2^s + 2$ then the best procedure of type 1 is equivalent to the best procedure of type 2. In fact the best procedure of type 2 for q close to one has the form

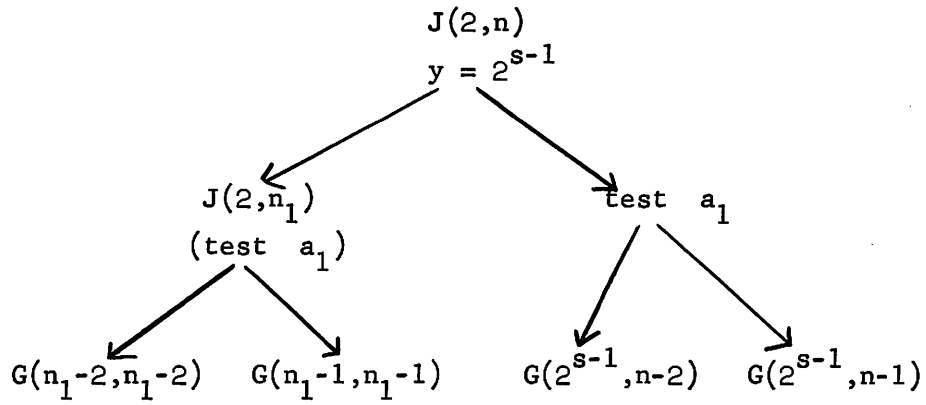


Figure 7.3 Optimal Test of Type 2 for J(2,n) Situation.

Here $n_1 = n - y = n - 2^{s-1}$ is not of the form 2^t+2 if n is not of that form since if $n = 2^s + 2 + \theta$ with $\theta \neq 0$ then by (5.17.) we find that $-2^{s-3} \leq \theta \leq 2^{s-2}$ and thus $n_1 = 2 + 2^{s-1} + \theta$ is not of the form $2^t + 2$. Hence by the induction hypothesis the best procedure for $J(2,n_1)$ is to start by testing a_1. We have used the hypothesis that $x(2n-3)$ is a power of 2 to insure that testing a_1 is optimal if the first test in Figure 7.3 fails. Hence the best procedure of type 2 has the form in Figure 7.3.

Since $x(2n-3) = 2^s$ and $2n-3$ is odd, it follows from the definition of $x(m)$ in (5.17.) that $x(2n-4)$ and $x(2n-2)$ are also equal to 2^s and hence $x(n-2)$ and $x(n-1)$ are both equal to 2^{s-1}. Hence the corresponding procedure of type 1 for q close to one has the form

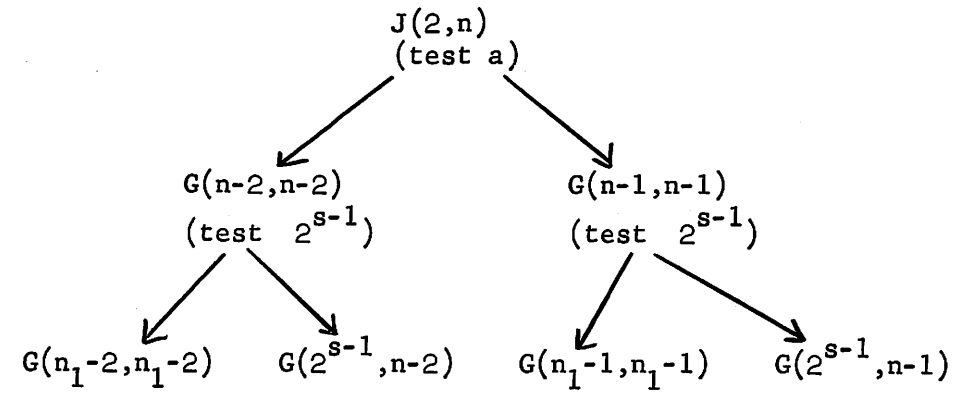


Figure 7.4 Optimal Test of Type 1 for J(2,n) Situation.

where $n_1 = n - 2^{s-1}$ as above. We note that this is equivalent with the procedure of type 2 in Figure 7.3 since we have merely interchanged the order

of two tests. To complete this proof we must show that for $J(2,3)$ and $J(2,5)$ the best procedures of type 1 and type 2 are identical. For $J(2,3)$ we get complete symmetry and the result is clear. The result for $J(2,5)$ can also be thrown back on the result for $J(2,3)$, and this completes the proof.

In the remaining cases where $x(2n-3)$ is not a power of 2, it has been found empirically that the procedure of type 1 is as good or better than any of type 2 but this has not been proved to be optimal and must be regarded as a conjecture.

In the case of the $G(3,n)$ and $J(3,n)$ situations for $n \geq 3$, we have given in Table A2 general patterns for $n = 2^s + 1$, $n = 2^s + 2$ and $2^s+3 \leq n \leq 2^{s+1}$.

It is known that in the $G(3,n)$ situation with $n \geq 3$ the best procedure for q close to one tests all except a_3 and if that fails all except a_2 and if that fails all except a_1. If any one of these passes, the test is clearly over and if they all fail, we are then in a $J(3,n)$ situation. For q close to one the relation between the expected number of test $G(3,n)$ for the $G(3,n)$ situation and $J(3,n)$ for the $J(3,n)$ situation is

$$(7.7) \qquad pG^*(3,n) = 6pq^{n-1} + [p^3 + 3p^2q + 3pq^2(1-q^{n-3})][3 + J(3,n)]$$

where $G^*(3,n)$ is defined in (3.4) (see also Table A3). This can also be written for q close to one as

$$(7.8) \qquad J^*(3,n) = G^*(3,n) - 3(1+q+q^2-q^{n-1}) = pJ^{**}(3,n)$$

where $J^*(3,n)$ and $J^{**}(3,n)$ are defined in terms of $J(3,n)$ by

$$(7.9) \qquad J^*(3,n) = [p^2 + 3pq + 3q^2(1-q^{n-3})]\, J(3,n) = pJ^{**}(3,n).$$

To give some indication why the procedures in Table A2 are optimal for the $J(3,n)$ situation when $n = 2^s + 2$ for some integer $s \geq 2$, we first note that the number of maximal probability states for q close to one is

$M = 3n-6 = 3\cdot 2^s$. In this case we have already noted by considering the coefficient of q^n that if there are group tests that partition this M-set exactly in half then the optimal procedure must be among them. We now show that for $J(3,n)$ with $n = 2^s+2$ and $s \geq 2$ there is a unique first group-test that accomplishes this partition into "half and half" and hence it must be the best possible first test.

Denote the 3 units that are known to contain a defective by a_i $(i = 1,2,3,)$ and remaining units by b_i $(i = 1,2,\ldots, n-3)$. We can easily rule out the possibility that the next test contains 2 of the 3 units (a_1, a_2, a_3) since the set corresponding to a successful test contains at most 2^s-1 states, which is less than $3\cdot 2^{s-1}$. If the next test contains 1 of these 3 units (say a_1) and say j of the units b_i where $j \geq 0$, then it is easily verified that the set corresponding to a successful test contains $2n-5-2j$ states, which is odd and hence not equal to $3\cdot 2^{s-1}$ for $s \geq 2$. If the next test contains none of the units a_i and j of the units b_i then the set corresponding to a successful test contains $3n-6-3j$ states and setting this equal to $3\cdot 2^{s-1}$, we obtain the unique solution $j = 2^{s-1}$.

If the first test passes then we obtain a similar situation, namely $J(3,2^{s-1}+2)$, and the argument is more involved since the optimal continuation is not unique. In the next test we can either test $(b_1, b_2,\ldots, b_w)$ where $w = 2^{s-2}$ or $(a_1, b_1, b_2,\ldots, b_t)$ where $t = 2^{s-3}$. In fact, if we choose the first alternative, then a_1 can be brought into the $2r^{th}$ test for any integer r provided $s \geq 2r+1$. The optimal continuation for any of these choices is determined by the fact that the maximal probability set has size $3\cdot 2^j$ for some integer $j \geq 1$ and the optimal continuation is a group-test that splits this set exactly in half. It can be verified (the proof is omitted) that all these procedures are equivalent in that they yield the same expected number of tests (See Table A2). The lack of uniqueness here for the optimal procedure is analogous to the lack of uniqueness observed for the $J(2,n)$-situation when n is of the form 2^s+2.

8. Definition of Procedure R_{00}

We now use the previous results to define the procedure R_{00} for all values of q. As in the case of procedures R_0 and R_1, the procedure R_{00} is defined recursively. We use $H(n)$ to denote $E\{T|H(n),q,R_{00}\}$ and $G^*(m,n)$ with $n \geq m$ to denote $(1+q+\ldots+q^{m-1})G(m,n)$ where $G(m,n) = E\{T|G(m,n),q,R_{00}\}$.

In the $H(n)$ situation with $n \geq 1$ we test $x = x(n)$ units determined by

$$(8.1) \qquad H(n) = 1 + \min_{1\leq x\leq n} [q^x H(n-x) + pG^*(x,n)].$$

In the $G(m,n)$ situation with $n \geq m \geq 4$ we test $x = x(m)$ units from the defective set (without mixing) where $x(m)$ is determined by

$$(8.2) \qquad G^*(m,n) = (1+q+\ldots+q^{m-1}) + \min_{1\leq x\leq m-1} [q^x G^*(m-x,n-x) + G^*(x,n)].$$

The derivations of (8.1) and (8.2) are given in [7]. The value of $x = x(m,q)$ does not depend on n (see theorem 2 above); for any $m \geq 4$ the value of x is given by (5.17) for q sufficiently close to one.

Corresponding to the value of the r.h.s. of (8.2) for $x = 1$, we define for $n \geq m = 2$ and $n \geq m = 3$, respectively,

$$(8.3) \qquad G_1(2,n) = 1+q+qH(n-2) + H(n-1),$$

$$(8.4) \qquad G_1(3,n) = 1 + q + q^2 + qG^*(2,n-1) + H(n-1).$$

For $m = 2,3$ and $n \geq m$ we shall use $G'(m,n)$ to denote (for all q) the polynomial expression that holds for $G^*(m,n)$ when q is close to one. According to Procedure R_{00} when $n \geq m = 2$ or 3 we either test one unit from the defective set of size m (as in procedure R_1) or we use the appropriate subroutine given in Tables A1 and A2. Hence we have for $m = 2,3$ and any $n \geq m$

$$(8.5) \qquad G^*(m,n) = \min[G_1(m,n),\ G'(m,n)].$$

It should be noted that $G^*(2,2) \equiv G_1(2,2) \equiv G'(2,2)$ since in this case

we can only test one unit from the defective set. Using $G'(m,n)$, we can also rewrite (7.2) and (7.8), respectively, so that they hold for all values of q, namely

$$(8.6)\quad J^*(2,n) = G'(2,n) - (2+2q-q^{n-1}) \qquad (n\geq 2),$$

$$(8.7)\quad J^*(3,n) = G'(3,n) - 3(1+q+q^2-q^{n-1}) \qquad (n\geq 3).$$

In particular, we note from (7.3) and (7.9) that $J^*(2,3) = J^*(3,3)$ for all q; this holds because the $J(2,3)$- and $J(3,3)$-situations are identical.

Putting (8.6) and (8.7) in (8.5) we obtain for $m = 2,3$ and $n \geq m$

$$(8.8)\quad G^*(2,n) = \text{Min}[G_1(2,n),\ J^*(2,n) + 2 + 2q - q^{n-1}],$$

$$(8.9)\quad G^*(3,n) = \text{Min}[G_1(3,n),\ J^*(3,n) + 3(1 + q + q^2 - q^{n-1})]$$

and the only remaining problem is to use the procedures in Tables A1 and A2 to find recursion formulae for $J^*(2,n)$ and $J^*(3,n)$ for higher values of n.

For $m = 2$ we note that $J^*(2,2)\equiv 0$ for all q and using the procedures in Table A1 we obtain the following recursion formulas. For $n = 3$ or 4

$$(8.10)\quad J^*(2,n) = 1 + q - 2q^{n-1} + qJ^*(2,n-1) + pG^*(2,n-1).$$

If $n \geq 5$ and not of the form 2^s+2 for any integer s

$$(8.11)\quad J^*(2,n) = 1 + q - 2q^{n-1} + pqG^*(n-2,n-2) + pG^*(n-1,n-1).$$

If $n = 2^s+2$ for some integer $s \geq 2$, letting $z = 2^{s-1} \geq 2$,

$$(8.12)\quad J^*(2,n) = (2-q^z)(1+q) - 2q^{n-1} + q^zJ^*(2,z+2) + pqG^*(z,n-2) + pG^*(z,n-1).$$

For $m = 3$ we have noted above that $J^*(3,3)\equiv J^*(2,3)$ given in (8.10) and using the procedures in Table A2 we further obtain the following recursion formulas. For $n = 4$ or 5

$$(8.13)\quad J^*(3,n) = 1 + q + q^2 - 3q^{n-1} + qJ^*(2,n-1) + pG^*(n-1,n-1).$$

If $n = 2^s + 2$ for some integer $s \geq 2$, letting $z = 2^{s-1}$,

$$(8.14)\quad J^*(3,n) = (1+q+q^2)[s-(s-1)q^z]-3q^{n-1}+ q^zJ^*(3,z+2) + p\sum_{i=1}^{z} q^{i-1}G^*(3,n-i).$$

For $n = 7, 8$ or 9

$$(8.15)\quad J^*(3,n) = 3 + 3q - 2q^3 - 4q^{n-1} + q^2J^*(2,n-2) + pq^2G^*(n-3,n-3) + pqG^*(2,n-2) + pG^*(2,n-1).$$

If $2^s + 3 \le n \le 2^{s+1}$ for some integer $s \ge 3$, letting $w = 2^{s-2}$,

$$(8.16)\quad J^*(3,n) = 3 + 4q + 4q^2 - 3q^{w+1} - 4q^{w+2} - 4q^{n-1} + q^{w+1}J^*(2,n-1-w) + pq^{w+2}G^*(n-w-3,n-w-3) + pq^2G^*(w,n-3) + pqG^*(w,n-2) + pG^*(w+2,n-1).$$

If $n = 2^s+1$ for some integer $s \ge 4$, letting $w = 2^{s-2}$,

$$(8.17)\quad J^*(3,n) = 3 + 4q + 4q^2 - 4q^w - 3q^{w+1} - 4q^{n-1} + q^wJ^*(2,n-w) + pq^wG^*(3w,3w) + pq^2G^*(w-1,n-3) + pqG^*(w-1,n-2) + pG^*(w,n-1).$$

The boundary conditions for $G^*(m,n)$ and $H^*(m,n)$ are the same as for $G(m,n)$ and $H(n)$ in [7], namely, for all q

$$(8.18)\quad \begin{aligned} H(0) &= 0 \\ G^*(1,n) &= H(n-1) \qquad n = 1,2,\ldots. \end{aligned}$$

This completes the definition of procedure R_{00}.

It is worth remarking that the expressions for $J^*(2,n)$ and $J^*(3,n)$ either have no $J^*(m,n)$ on the right side or can be iterated to remove $J^*(m,n)$ from the right side. Hence it is possible to write all the equations in terms of $H_1(n)$ and $G^*(m,n)$ only, without any $J^*(m,n)$ at all. For example, we can use (7.2) to replace (8.12) by

$$(8.19)\quad J^*(2,n) = (2-3q^z)(1+q) - q^{n-1} + q^zG^*(2,z+2) + pqG^*(z,n-z) + pG^*(z,n-1)$$

and we can substitute this in the right side of (8.8) for $n = 2^s+2$ and $z = 2^{s-1}$. This may be useful for computation but the result does not appear to be simpler; such expressions were given in a preliminary report [9] on this paper.

We note that in (8.1) either the argument is reduced or we obtain a $G^*(x,n)$-term. For each $G^*(m,n)$ by (8.2) either the second argument is

reduced or the first one is. The same is true in all the above formulas. Hence the testing procedure must come to a conclusion after a finite number of tests.

Explicit polynomial expressions for $H(n)$ and $G^*(m,n)$ for $n \leq 8$ and all values of q can be found in Table A3 and A4, respectively; the polynomial for $G^*(m,n)$ defined after (3.4) then yields an explicit expression for the corresponding values of $G(m,n)$. Polynomial expressions for $J^{**}(2,n)$ and $J^{**}(3,n)$ for $n \leq 8$ and q close to one (or p close to zero) are given in Tables A6 and A7; these are given in powers of p.

9. Concluding Remarks and Conjectures

The above procedure R_{00} has all the properties that have been shown in this paper to be necessary for a procedure to be optimal. As mentioned in Section 7, several of the subroutines for the $J(3,n)$ situation, schematized in Table A3, have not been proved to be optimal. In addition it has been implicitly assumed that if we were to "combine" (see (8.5)) the recursion formulas for Procedure R_1, which is optimal for small values of q, with a special procedure which is optimal for q-values asymptotically close to 1 then the resulting procedure would be optimal for all values of q; this has not been proved. However it is conjectured that the procedure R_{00} is optimal for all values of q if N is of type 1 (see Section 6).

The procedure R_0 defined in [8] was introduced in this paper mainly for purposes of comparison (see Table 1). However it was pointed out in [8] that the procedure R_0 (as well as R_1) has the "first come, first served" property, i.e., if the unit u_i stands before the unit u_j in a given (original) ordering then u_j will not be classified before u_i. This property is destroyed for procedure R_{00}, e.g., in the special subroutine for $J(2,4)$ in Table A1. It is conjectured that Procedure R_0 is the best procedure retaining this property.

It has already been noted that procedure R_1 is the best non-mixing

procedure; this follows from the derivation of the recursion formulas (3.3) and (3.4). It was noted in [7] that if the units are not labeled (so that their individual past history is lost after each test) then the mixing subroutines of both procedures R_0 and R_{00} are no longer possible. It is conjectured that Procedure R_1 is optimal under this restriction.

Some further general remarks and conjectures about Procedures R_1 and R_0 are given in [7] and [8].

10. Acknowledgement

The author wishes to thank Professor George Woodworth and Miss Phyllis Groll both of Stanford University for helpful discussions in connection with this paper. He also wishes to acknowledge the assistance of Miss Phyllis Groll and Mrs. Elaine Frankowski of the University of Minnesota for assistance in the preparation of all the tables in this paper.

Table A1

Special Subroutines for Procedure R_{00} in the G(2,n) and J(2,n) Situations for $n \geq 2$ and q Sufficiently Close to One.#

G(2,2):	$T(a_1) \nearrow^{S} T(a_2) \nearrow^{S} S$	(Same as R_0)
G(2,n): $n \geq 2$	$T(a_1,b_1,\ldots,b_{n-2}) \nearrow^{S} T(a_2,b_1,\ldots,b_{n-2}) \nearrow^{S} J(2,n)$	(Same as R_0)

J(2,3):	$T(a_1) \nearrow^{S} G(2,2)$	(Same as R_0)
J(2,4):	$T(b_1) \nearrow^{J(2,3)} G(2,3)$	

If $n \geq 5$ and $n \neq 2^r+2$ for any positive integer r then the scheme is

J(2,n):	$T(a_1) \nearrow^{G(n-2,n-2)} G(n-1,n-1)$	(Same as R_0)

If $n = 2^r+2$ for some integer $r \geq 2$ and we let $t = 2^{r-1}$ then the scheme is

J(2,n): $T(b_1,\ldots,b_t) \nearrow^{J(2,t+2)} T(a_1) \nearrow G(t,n-2)$, $T(a_1) \rightarrow G(t,n-1)$

#The symbol T(x,y) indicates that the procedure R_{00} calls for a test on x and y. A slanted arrow corresponds to a successful test; a horizontal arrow corresponds to an unsuccessful test. The symbol S indicates terminal stopping point; the symbols G(m,n) and J(m,n) indicate situations as explained in the text.

Table A2

Special Subroutines[1] for Procedure R_{00} in the G(3,n) and J(3,n)

Situations for $n \geq 3$ and q Sufficiently[2] Close to One.[3]

G(3,n): $n \geq 3$	$T(a_1,a_2,b_1,\ldots,b_{n-3}) \xrightarrow{S} T(a_1,a_3,b_1,\ldots,b_{n-3}) \xrightarrow{S} T(a_2,a_3,b_1,\ldots,b_{n-3}) \xrightarrow{S} J(3,n)$	(Same as R_0)
J(3,3) = J(2,3):	$T(a_1) \xrightarrow{S} G(2,2)$	(Same as R_0)
For n = 4 and 5 J(3,n):	$T(a_1) \xrightarrow{J(2,n-1)} G(n-1,n-1)$	(Same as R_0)
J(3,6):	$T(b_1,b_2) \xrightarrow{J(3,4)} T(b_1) \xrightarrow{G(3,4)} G(3,5)$	
For n = 7,8 and 9 J(3,n):	$T(a_1,b_1) \xrightarrow{J(2,n-2)} T(a_2,a_3) \xrightarrow{G(n-3,n-3)} T(a_1) \xrightarrow{G(2,n-2)} G(2,n-1)$	

If $n = 2^r+2$ for any integer $r \geq 3$ and we let t for 2^{r-3}, then the scheme is

J(3,n): $T(b_1,\ldots,b_{4t}) \xrightarrow{J(3,4t+2)} T(b_1,\ldots,b_{2t}) \rightarrow T(b_{2t+1},\ldots,b_{3t}) \rightarrow T(b_1,\ldots,b_t) \ldots\ldots$

$T(b_{4t-1}) \rightarrow G(3,4t+2)$, $G(3,4t+3)$

$T(b_3) \rightarrow G(3,n-4)$, $G(3,n-3)$

$T(b_1) \rightarrow G(3,n-2)$, $G(3,n-1)$

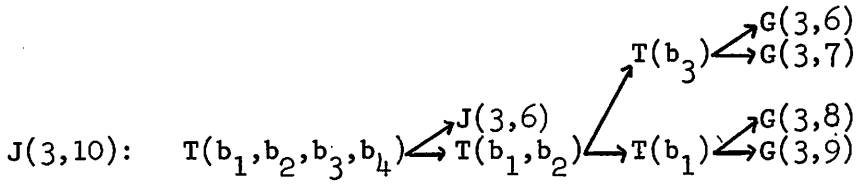
[1]The situations J(2,3), J(3,3) and J(3,4) are symmetrical and hence the test of any one unit gives the same results.

[2]See Tables A4 and A5 for the explicit q intervals.

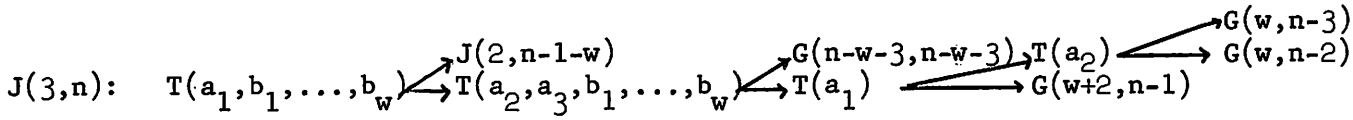
[3]See footnote to Table A1 for explanation of symbols.

Table A2 cont.

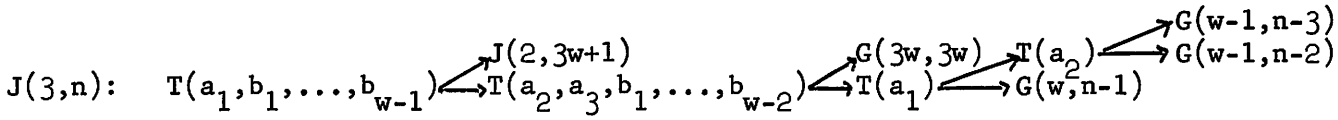
For example, we obtain for[§] n = 10 (i.e., r = 3)



For $2^r+3 \leq n \leq 2^{r+1}$ with r an integer ≥ 3 (writing w for 2^{r-2})



For $n = 2^r+1$ with r an integer ≥ 4 (writing w for 2^{r-2})



[§]The J(3,7), J(3,8) and J(3,10) situations arise only from a $J(3,2^r+2)$ situation with even $r \geq 4$ and only by passing the first test (see the appropriate scheme above); trying different starting n values and using (5.2), we find that they require at least n = 22 units in the initial H-situation.

Table A3

Polynomial Expression for $H^*(n) = E\{T|H(n), q, R_{00}\}$

(The integer shown is the coefficient of the power of q at the top of the column and the terms are then added to form $H^*(n)$)

n	$x^{\#}$	1	q	q^2	q^3	q^4	q^5	q^6	q^7	q^8	q-interval
2	1	2									.000 to .618
	2	3	- 1	- 1							.618 to 1.000
3	1	3									.000 to .618
	2	5	- 3	- 1	1						.618 to .707
	3	5	- 2	- 1	-1						.707 to .843
	3	7	- 3	- 6	3						.843 to 1.000
4	1	4									.000 to .618
	2	7	- 5	0	1	-1					.618 to .707
	2	7	- 4	- 1	-1	1					.707 to .786
	4	8	- 4	- 2	-1						.786 to .843
	4	10	- 5	- 7	3						.843 to 1.000
5	1	5									.000 to .618
	2	9	- 7	1	0	-1	1				.618 to .707
	2	9	- 6	0	-1	1	-1				.707 to .755
	3	9	- 5	- 1	-2	1					.755 to .786
	3	10	- 6	- 2	-1	0	1				.786 to .817
	5	11	- 7	- 2	0	0	-1				.817 to .843
	5	13	- 8	- 7	4	0	-1				.843 to .891
	5	14	- 8	- 8	4	-3	2				.891 to 1.000
6	1	6									.000 to .618
	2	11	- 9	2	-1	0	1	-1			.618 to .707
	2	11	- 8	1	-2	1	-1	1			.707 to .755
	3	11	- 6	- 2	-2	2	0	-1			.755 to .786
	3	12	- 7	- 3	-1	1	1	-1			.786 to .817
	3	13	- 9	- 2	0	0	-1	1			.817 to .843
	3	15	-10	- 7	4	0	-1	1			.843 to .844
	6	16	-11	- 6	4	- 1	-1				.844 to .891
	6	17	-12	- 7	5	- 4	5	-3			.891 to .914
	6	19	-14	-10	13	-11	4				.914 to .948
	6	20	-11	- 9	1	-10	12	-2			.948 to 1.000

$^{\#}$The entry x indicates that the next test is on x units.

Table A3 cont.

n	x	l	q	q^2	q^3	q^4	q^5	q^6	q^7	q^8	q-interval
7	1	7									.000 to .618
	2	13	-11	3	-2	1	0	-1	1		.618 to .707
	2	13	-10	2	-3	2	-1	1	-1		.707 to .755
	3	13	- 7	-3	-2	2	0	-1	1		.755 to .786
	3	14	- 8	-4	0	1	0	-1			.786 to .817
	3	15	-10	-3	1	0	-2	1			.817 to .819
	4	16	-11	-3	0	0	-1	1			.819 to .843
	4	18	-12	-8	4	0	-1	1			.843 to .844
	4	19	-14	-6	3	-1	0	0	1		.844 to .869
	7	19	-13	-6	3	-1	-1				.869 to .891
	7	20	-14	-7	4	-4	5	-3			.891 to .915
	7	22	-16	-10	12	-11	4				.915 to .939
	7	24	-19	-8	15	-21	12	-5	3		.939 to .948
	7	25	-17	-10	2	-8	19	-15	5		.948 to 1.000
8	1	8									.000 to .618
	2	15	-13	4	-3	2	-1	0	1	-1	.618 to .707
	2	15	-12	3	-4	3	-2	1	-1	1	.707 to .755
	3	15	- 8	-4	-2	3	0	-2	1		.755 to .786
	3	16	- 9	-5	0	1	0	-1	0	1	.786 to .812
	4	17	-10	-5	0	1	0	-1			.812 to .817
	4	18	-12	-4	1	0	-2	1			.817 to .819
	4	19	-14	-3	0	1	-1				.819 to .843
	4	21	-15	-8	4	1	-1				.843 to .844
	4	22	-17	-6	3	0	0	-1	1		.844 to .869
	4	22	-16	-7	3	0	-1	0	0	1	.869 to .885
	8	22	-15	-7	3	-1	-1	1	0	-1	.885 to .891
	8	23	-16	-8	5	-4	4	-2	-3	2	.891 to .914
	8	25	-18	-13	15	-8	-5	8	-2	-1	.914 to .915
	8	26	-19	-12	14	-10	-2	7	-4	1	.915 to .939
	8	28	-24	-7	15	-23	16	-6	4	-2	.939 to .940
	8	29	-24	-8	15	-23	16	-6	1	1	.940 to .948
	8	30	-22	-10	2	-10	23	-16	3	1	.948 to 1.000

Table A4

Polynomial Expression for $G^*(m,n) = (\frac{1-q^m}{1-q})E\{T|G(m,n), q, R_{00}\}$ for pairs (m,n) with $n \leq 8$ that can arise when using Procedure R_{00}.

m	n	x#	1	q	q^2	q^3	q^4	q^5	q^6	q^7	q^8	q-interval
2	2	1	2	1								.000 to 1.000
2	3	1+0	3	2								.000 to .618
		1+0	4	1	- 1							.618 to .768
		1+1	5	2	- 4							.768 to 1.000
3	3	1	3	3	-2							.000 to .618
		1	4	2	1							.618 to .843
		2	6	3	- 3							.843 to 1.000
2	4	1+0	4	3								.000 to .618
		1+0	6	1	- 2							.618 to .707
		1+0	6	2	- 2	-2						.707 to .843
		1+2	8	3	- 6	-2						.843 to 1.000
3	4	1+0	4	4	3							.000 to .618
		1+0	6	2	1							.618 to .707
		1+0	6	3	1	-2						.707 to .768
		1+0	6	4	2	-5						.768 to .843
		1+0	8	3	- 3	-1						.843 to .907
		2+1	10	4	- 2	-6						.907 to 1.000
4	4	1	4	4	4	3						.000 to .618
		1	6	2	2	3						.618 to .707
		1	6	3	2	1						.707 to .755
		2	7	3	1							.755 to .843
		2	9	4	- 3							.843 to 1.000
2	5	1+0	5	4								.000 to .618
		1+0	8	1	- 3							.618 to .707
		1+0	8	2	- 3	-2						.707 to .786
		1+0	9	2	- 4	-2	-1					.786 to .843
		1+0	11	3	-10	-3	3					.843 to .891
		1+3	12	4	-10	-3						.891 to 1.000
3	5	1+0	5	5	4							.000 to .618
		1+0	8	2	2	-1	-1					.618 to .707
		1+0	8	3	2	-3	-1					.707 to .786
		1+0	9	3	1	-3	-2					.786 to .843
		1+0	11	4	- 3	-3	-2					.843 to .934
		2+2	13	5	-12	-3	-7					.934 to 1.000

#The entry $x=x_1+x_2$ indicates that the next test is on x_1+x_2 units, x_1 from the set known to contain at least one defective, i.e., the a's, and x_2 from the remaining binomial set, i.e., the b's. If m=n then only 1 integer is shown in the x column.

Table A4 cont.

m	n	x	l	q	q^2	q^3	q^4	q^5	q^6	q^7	q^8	q-interval
		1	5	5	5	5	4					.000 to .618
		1	8	2	3	4	3					.618 to .682
		2	9	2	2	3	2					.682 to .707
5	5	2	9	3	2	1	2					.707 to .786
		2	10	3	1	1	1					.786 to .843
		2	12	4	-3	1	1					.843 to .891
		2	13	5	-3	1	-2					.891 to 1.000
		1+0	6	5								.000 to .618
		1+0	10	1	-4							.618 to .707
		1+0	10	2	-4	-2						.707 to .755
		1+0	10	3	-5	-3	0	1				.755 to .786
		1+0	11	3	-6	-3	-1	1				.786 to .817
		1+0	12	2	-6	-2	-1	-1				.817 to .843
		1+0	14	3	-12	-3	3	-1				.843 to .891
		1+0	15	3	-13	-3	0	2				.891 to .913
		1+4	16	4	-14	-2	1	-2				.913 to 1.000
		1+0	6	6	5							.000 to .618
		1+0	10	2	3	-3	-1	1				.618 to .707
		1+0	10	3	3	-4	-1	-1				.707 to .755
		1+0	10	4	2	-5	-1					.755 to .786
3	6	1+0	11	4	1	-5	-2					.786 to .817
		1+0	12	3	1	-4	-2	-2				.817 to .843
		1+0	14	4	-3	-6	-3	2				.843 to .891
		1+0	15	5	-3	-6	-6	2				.891 to .949
		2+3	18	7	-2	-7	-14	4				.949 to 1.000
		1	6	6	6	6	6	5				.000 to .618
		1	10	2	4	4	4	5				.618 to .652
		2	11	2	3	3	3	4				.652 to .707
		2	11	3	3	2	3	2				.707 to .755
		2	11	4	3	1	2	2				.755 to .786
6	6	2	12	4	2	1	1	2				.786 to .817
		2	13	3	2	2	1					.817 to .843
		2	15	4	-2	2	1					.843 to .891
		2	16	4	-3	2	-2	3				.891 to .914
		2	18	4	-6	7	-4					.914 to .948
		3	19	8	-1	0	-10	2				.948 to 1.000

Table A4 cont.

m	n	x	l	q	q^2	q^3	q^4	q^5	q^6	q^7	q^8	q-interval
2	7	1+0	7	6								.000 to .618
		1+0	12	1	-5							.618 to .707
		1+0	12	-2	-5	-2						.707 to .755
		1+0	12	4	-7	-3	0	1	-1			.755 to .786
		1+0	13	4	-9	-3	0	1				.786 to .817
		1+0	14	3	-9	-2	0	-1				.817 to .843
		1+0	16	4	-15	-3	4	-1				.843 to .844
		1+0	17	3	-14	-3	3	-1	-1			.844 to .891
		1+0	18	3	-15	-3	0	2	-1			.891 to .914
		1+0	20	1	-18	5	-7	1	2			.914 to .929
		1+5	21	2	-18	5	-7	1	-1			.929 to .948
		1+5	22	5	-17	-7	-6	9	-3			.948 to 1.000
3	7	1+0	7	7	6							.000 to .618
		1+0	12	2	4	-5	0	1	-1			.618 to .707
		1+0	12	3	4	-6	-1	-1	1			.707 to .755
		1+0	12	5	2	-7	-1					.755 to .786
		1+0	13	5	1	-7	-2					.786 to .817
		1+0	14	4	1	-6	-2	-2				.817 to .843
		1+0	16	5	-3	-8	-3	2				.843 to .844
		1+0	17	4	-2	-8	-4	2	-1			.844 to .891
		1+0	18	4	-3	-8	-7	5	-1			.891 to .914
		1+0	20	4	-6	-3	-9	2	-1			.914 to .948
		1+0	21	7	-5	-15	-8	10	-3			.948 to .958
		1+4	22	6	-4	-7	-7	0	-4			.958 to 1.000
4	7	1+0	7	7	7	6						.000 to .618
		1+0	12	2	5	3	-3					.618 to .707
		1+0	12	3	5	2	-3	-2				.707 to .755
		2+0	13	5	2	0	-3	-1				.755 to .786
		2+0	14	5	1	0	-4	-1	-1			.786 to .817
		2+0	15	4	1	1	-4	-3	-1			.817 to .843
		2+0	17	5	-3	1	-6	-4	3			.843 to .844
		2+0	18	4	-2	1	-7	-4	2			.844 to .891
		2+0	19	4	-2	2	-10	-1	-1			.891 to .914
		2+0	21	2	-5	10	-17	-2	2			.914 to .929
		2+0	22	3	-5	10	-17	-2	-1			.929 to .948
		2+0	23	6	-4	-2	-16	6	-3			.948 to 1.000

Table A4 cont.

m	n	x	l	q	q^2	q^3	q^4	q^5	q^6	q^7	q^8	q-interval
		1	7	7	7	7	7	7	6			.000 to .618
		1	12	2	5	4	5	6	5			.618 to .637
		2	13	2	4	3	4	5	4			.637 to .682
		2	13	2	5	3	3	4	3			.682 to .707
		2	13	3	5	2	3	2	3			.707 to .755
		2	13	5	3	1	3	3	2			.755 to .786
		2	14	5	2	1	2	3	2			.786 to .817
7	7	2	15	4	2	2	2	1	2			.817 to .843
		2	17	5	-2	2	2	1	2			.843 to .844
		2	18	4	-1	2	1	1	1			.844 to .857
		3	18	5	-1	2	1					.857 to .891
		3	19	5	-2	2	-2	3				.891 to .914
		3	21	5	-5	7	-4					.914 to .939
		4	23	4	-4	11	-10	2	-3			.939 to .948
		4	24	7	-3	-1	-9	10	-5			.948 to 1.000
		1+0	8	7								.000 to .618
		1+0	14	1	-6							.618 to .707
		1+0	14	2	-6	-2						.707 to .755
		1+0	14	5	-9	-4	0	2	-1			.755 to .786
		1+0	15	5	-11	-3	0	1	0	-1		.786 to .817
		1+0	16	4	-12	-1	0	-2	0	1		.817 to .819
		1+0	17	3	-12	-2	0	-1	0	1		.819 to .843
2	8	1+0	19	4	-18	-3	4	-1	0	1		.843 to .844
		1+0	20	3	-17	-3	3	-1	-1	1		.844 to .869
		1+0	20	4	-17	-3	3	-2	-1			.869 to .891
		1+0	21	4	-19	-3	1	1	2	-3		.891 to .914
		1+0	23	4	-24	2	2	-7	4			.914 to .939
		1+0	25	1	-22	5	-8	1	-1	3		.939 to .940
		1+7	26	2	-22	5	-8	1	-1			.940 to .948
		1+7	27	5	-21	-7	-7	9	-3			.948 to 1.000
		1+0	8	8	7							.000 to .618
		1+0	14	2	5	-7	1	0	-1	1		.618 to .707
		1+0	14	3	5	-8	0	-1	1	-1		.707 to .755
		1+0	14	6	2	-9	-1					.755 to .786
		1+0	15	6	1	-9	-2					.786 to .817
		1+0	16	5	1	-8	-2	-2				.817 to .819
		1+0	17	4	1	-9	-2	-1				.819 to .843
		1+0	19	5	-3	-11	-3	3				.843 to .844
3	8	1+0	20	4	-2	-11	-4	3	-1			.844 to .869
		1+0	20	5	-2	-11	-4	2	-1	-1		.869 to .891
		1+0	21	5	-3	-11	-7	5	-1	-1		.891 to .914
		1+0	23	5	-8	-6	-6	-3	1	2		.914 to .929
		1+0	23	6	-7	-6	-6	-3	1	-1		.929 to .939
		1+0	25	3	-5	-3	-16	5	-4	2		.939 to .948
		1+0	26	6	-4	-15	-15	13	-6	2		.948 to .964
		2+5	28	5	-4	-16	-9	20	-17	-1		.964 to 1.000

Table A4 cont.

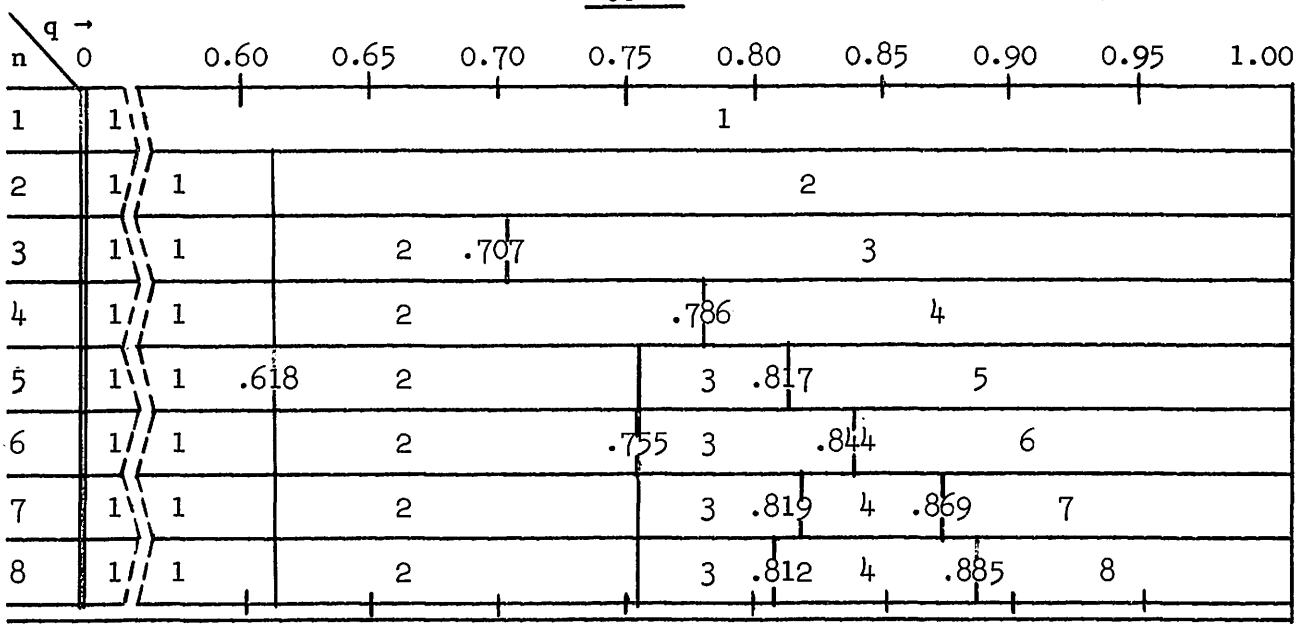
m	n	x	l	q	q^2	q^3	q^4	q^5	q^6	q^7	q^8	q-interval
		1+0	8	8	8	7						.000 to .618
		1+0	14	2	6	3	-4					.618 to .707
		1+0	14	3	6	2	-4	-2				.707 to .755
		2+0	15	6	2	0	-5	-1	-1	1		.755 to .786
		2+0	16	6	1	1	-6	-2	-1			.786 to .817
		2+0	17	5	1	2	-6	-4	-1			.817 to .819
		2+0	18	4	1	1	-6	-3	-1			.819 to .843
4	8	2+0	20	5	-3	1	-8	-4	3			.843 to .844
		2+0	21	4	-2	1	-9	-4	2			.844 to .869
		2+0	21	5	-2	1	-9	-5	2	-1		.869 to .891
		2+0	22	5	-3	1	-12	-2	2	-1		.891 to .913
		2+0	24	5	-7	7	-12	-9	5	-2		.913 to .939
		2+0	26	2	-5	10	-22	-1	0	1		.939 to .940
		2+0	27	3	-5	10	-22	-1	0	-2		.940 to .948
		2+0	28	6	-4	-2	-21	7	-2	-2		.948 to 1.000
		1	8	8	8	8	8	8	8	7		.000 to .618
		1	14	2	6	4	6	6	6	7		.618 to .629
		2	15	2	5	3	5	5	5	6		.629 to .652
		2	15	2	6	3	4	4	4	5		.652 to .707
		2	15	3	6	2	4	3	4	3		.707 to .755
		2	15	6	3	1	4	4	2	3		.755 to .786
		2	16	6	2	2	3	3	2	2		.786 to .817
		2	17	5	2	3	3	1	2	2		.817 to .819
		3	18	5	2	2	2	1	2	2		.819 to .843
8	8	3	20	6	-2	2	2	1	2	2		.843 to .844
		3	21	5	-1	2	1	1	1	2		.844 to .869
		3	21	6	-1	2	1	0	1	1		.869 to .891
		3	22	6	-2	3	-1	3	1	-2		.891 to .914
		3	24	6	-7	8	0	-5	3	1		.914 to .915
		4	25	6	-6	8	-2	-4	3	-1		.915 to .939
		4	27	3	-4	11	-12	4	-2	2		.939 to .940
		4	28	4	-4	11	-12	4	-2	-1		.940 to .948
		4	29	7	-3	-1	-11	12	-4	-1		.948 to 1.000

Table A5

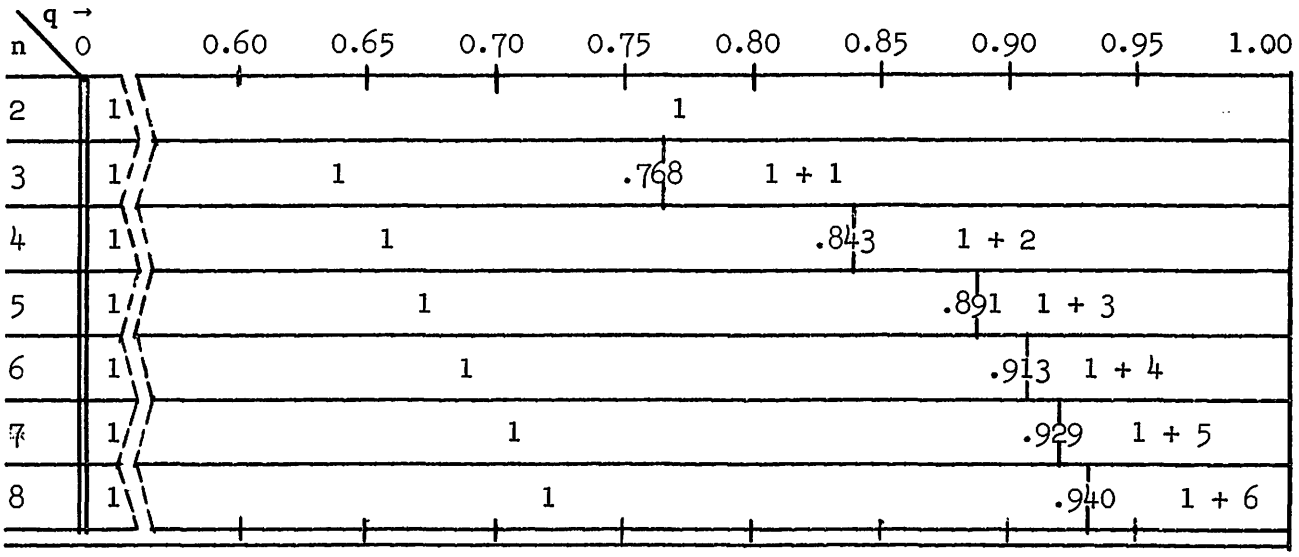
Diagram Showing the Number of Units to be Taken in any H-situation or any G-situation for $n = 1$ through 8 and $m \leq n$ under Procedure R_{00}.

(Those G-situations which will never arise if we start with an H-situation are omitted from the diagram.)

$H_{00}(n)$



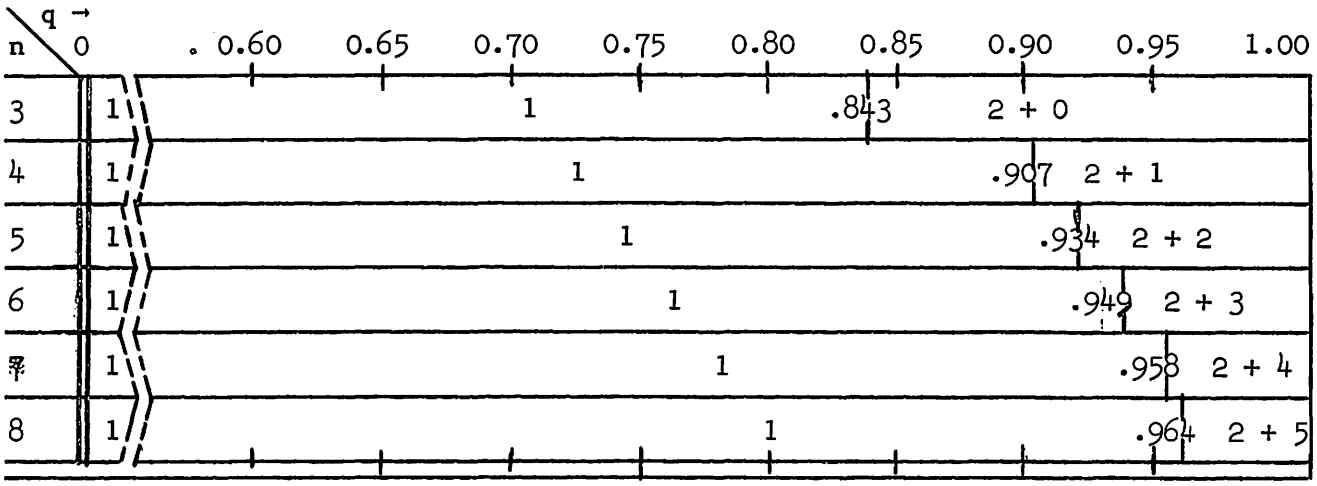
$G_{00}(2,n)$



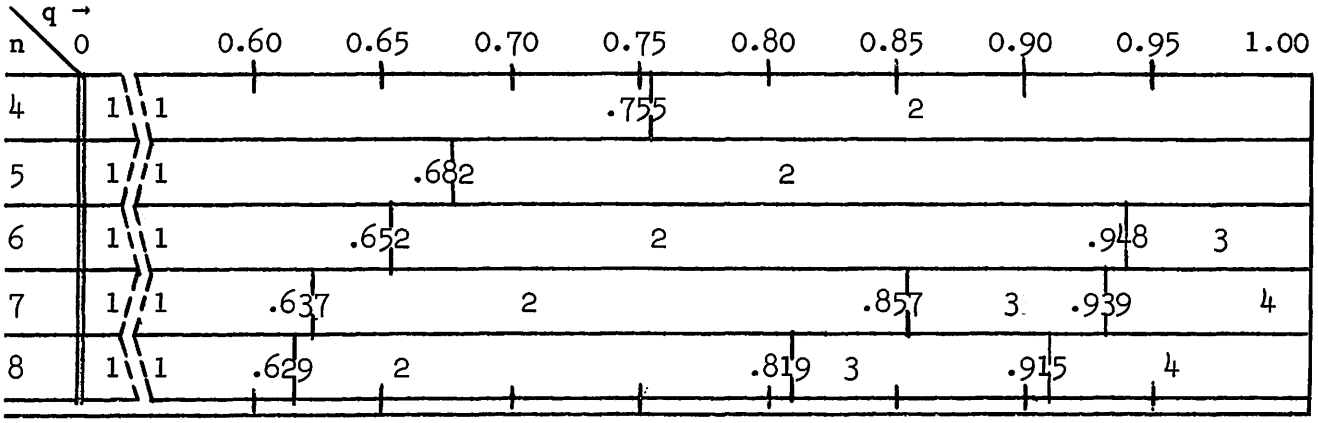
q = Probability of a Good Unit

Table A5 cont.

$G_{00}(3,n)$



$G_{00}(m,n)$ for[1] $n \geq m \geq 4$



[1] By Theorem 2 this part of the rule R_{00} does not depend on n.

Table A6

The Polynomials $J^{**}(2,n)$ and the First Test-Group in the $J(2,n)$ Situation under Procedures R_{00} and R_0 for $n \le 10$ and p Close to Zero[#]

$J^{**}(2,n|R_{00})$

n	Test	1	p	p^2	p^3	p^4	p^5	p^6	p^7	p^8	Range[$]
3	(a_1)	6	-3								$p \le .232$
4	(b_1)	14	-9	1							$p \le .157$
5	(a_1)	23	-13	-1	1						$p \le .109$
6	(b_1,b_2)	33	-24	8	-4	1					$p \le .087$
7	(a_1)	43	-14	-19	9	3	-2				$p \le .052$
8	(a_1)	54	-18	-30	28	-12	4	-1			$p \le .052$
9	(a_1)	66	-44	26	-28	-5	31	-20	4		$p \le .052$
10	(b_1,b_2,b_3,b_4)	78	-64	75	-119	100	-46	15	-5	1	$p \le .048$

$J^{**}(2,n|R_0)$

n	Test	1	p	p^2	p^3	p^4	p^5	p^6	p^7	p^8	Range[$]
3	(a_1)	6	-3								$p \le .232$
4	(a_1)	14	-7								$p \le .149$
5	(a_1)	23	-13	1							$p \le .109$
6	(a_1)	33	-22	5	-1						$p \le .086$
7	(a_1)	43	-13	-38	50	-28	6				$p \le .051$
8	(a_1)	54	-17	-49	69	-43	12	-1			$p \le .051$
9	(a_1)	66	-44	32	-81	116	-87	33	-5		$p \le .051$
10	(a_1)	78	-62	68	-136	166	-114	41	-6	0	$p \le .048$

[#] Multiplying each coefficient by the power of p above it and summing the row gives

$$J^{**}(2,n) = J^{*}(2,n)/p = (1 + q - 2q^{n-1})\, J(2,n)/p.$$

[$] Each entry is one minus the last dividing point for the corresponding $G(2,n)$ function.

Table A7

The Polynomials $J^{**}(3,n)$ and the First Test-Group in the $J(3,n)$ Situation under Procedures R_{00} and R_0 for $n \leq 10$ and p Close to Zero[#]

$J^{**}(3,n|R_{00})$

n	Test	1	p	p^2	p^3	p^4	p^5	p^6	p^7	p^8	Range[$]
3	(a_1)	6	-3								$p \leq .157$
4	(a_1)	18	-14	3							$p \leq .093$
5	(a_1)	33	-38	19	-4						$p \leq .066$
6	(b_1,b_2)	48	-40	-7	21	-7					$p \leq .051$
7	(a_1,b_1)	66	-85	55	-22	6	-1				$p \leq .042$
8	(a_1,b_1)	84	-122	122	-94	40	-3	-2			$p \leq .036$
9	(a_1,b_1)	102	-127	92	-86	95	-69	28	-5		$p \leq .031$
10	(b_1,b_2,b_3,b_4)	120	-93	-207	690	-997	840	-417	113	-13	$p \leq .027$

$J^{**}(3,n|R_0)$

n	Test	1	p	p^2	p^3	p^4	p^5	p^6	p^7	p^8	Range[$]
3	(a_1)	6	- 3								$p \leq .157$
§4	(a_1)	18	-14	3							$p \leq .093$
§5	(a_1)	33	-36	18	- 4						$p \leq .066$
6	(a_1)	49	-59	32	- 7						$p \leq .051$
§7	(a_1)	66	-66	27	16	- 9	2				$p \leq .041$
§8	(a_1)	84	-100	46	3	-13	6	- 1			$p \leq .035$
9	(a_1)	103	-137	108	-87	78	-51	19	- 3		$p \leq .031$
§10	(a_1)	123	-196	273	-408	472	-357	167	-44	5	$p \leq .027$

[#] Multiplying each coefficient by the power of p above it and summing the row gives

$$J^{**}(3,n) = J^{*}(3,n)/p = (1 + q + q^2 - 3q^{n-1})\, J(3,n)/p.$$

[$] Each entry is one minus the last dividing point for the corresponding $G(3,n)$ function.

§ Under procedure R_0, if we start with an H-situation, the only $J(3,n)$-situations that can arise are those for which n is an integer multiple of 3.

Appendix

Necessary Properties of Group-Test Codes

Corresponding to the fact that each test has exactly two possible outcomes we consider only binary codes in this discussion. Any test is allowed in group-testing provided only that we do not carry out a test whose answer we know in advance. This also implies that any inference that can be made about the units will be made as soon as possible.

The rows of a code will be called code words. It will be convenient to assume throughout that the code words are listed in lexicographic (or dictionary) ordering, although some properties (such as properties 1 and 2 below) can be stated without reference to any ordering.

Any code which has a group-testing interpretation will be called a group-testing code (GT code). The class of GT codes is a proper subset of the exhaustive codes (see property 2 below) and it should not be assumed that a GT code can have only one group-testing interpretation. Such questions of uniqueness will be discussed later.

The following properties are necessary properties of a GT code. Although it is conjectured that they are also sufficient, no attempt has been made to prove the sufficiency. In addition to stating and proving that these properties hold for a GT code, we show by examples how they can be used to prove that certain exhaustive codes are not GT codes.

Property 1: Every GT code has the prefix property, i.e., no code word W in a GT code can be the prefix of any other word in that code. Such codes are also called instantaneous.

Proof: A group-test must either stop or continue after the sequence of tests indicated by W and cannot do both; hence property 1 follows.

Property 2: If W_j $(j = 1, 2, \ldots, J)$ are the J words of a GT code and $L(W_j)$ denotes the length of W_j then

$$\sum_{j=1}^{J} 2^{-L(W_j)} = 1. \tag{A1}$$

For an arbitrary code satisfying property 1 is it well known [ ] that the weak inequality $\leq$ holds in (A1). Codes satisfying (A1) may be regarded as efficient codes and in the presence of property 1 have been called exhaustive codes by some authors [ ].

Proof: Assume that the strict inequality holds in (A1) so that the code is not exhaustive. Then we can add at least one code word to our list without destroying property 1 and still have the weak inequality $\leq$ in (A1). This means that there is a junction point in our code tree which emits only one line segment instead of the usual two. This in turn means that we carried out a test for which we knew the answer in advance. Since this is not allowed, it could not be a GT code.

We shall use the symbol $W = (V,0^{\alpha})$ to denote a code word with the prefix V of length $v \geq 0$ (consisting of zeros and ones) followed by α zeros. A corollary of property 2 is

Corollary 2.1: Any code word of the form $(V,0)$ is followed by the code word $(V,1,0^{\alpha})$ for some integer $\alpha \geq 0$.

Proof: If the next word has the prefix $(V,0)$ then property 1 is destroyed. If it has the prefix $(V,1)$ and is not followed only by zeros then we can insert at least one "missing" code word and property 2 is destroyed. If it does not have the prefix V and is in the correct ordering then $(V,1)$ can be used as a missing code word and property 2 is again negated.

Property 3: In a GT code any code word W of the form $(V,1)$ has as an immediate predecessor the code word $(V,0)$. This means that the last group test was on a single unit with no associated inference.

Proof: From properties 1 and 2 it follows that the predecessor of W has to have the form $W' = (V,0,V')$ where V' contains only ones (but we do not need this for our proof). By the nature of group-testing, the last test can result in failure (i.e., in a one) only if it is a test on a single unit with no inference and there are no other unclassified units left. Then we have to terminate after this test regardless of its result. Hence the code word $(V,0)$ must be present

in the code and in the lexicographic ordering it must be the immediate predecessor.

As corollaries of property 3 we have the following properties:

Corollary 3.1: Two successive code words of a GT code cannot both end in a one.

Proof: Corollary 3.1 is an immediate consequence of property 3.

Before giving the next corollary it is instructive to define a branch B of a GT code as a succession of code words $W_{i+1}, W_{i+2}, \ldots, W_j$ (in lexicographic order) such that W_α ends in 0 for $i < \alpha < j$ and W_i, W_j both end in 1. The length $L(B)$ of the branch B is the number of code words in B, so that $L(B) = j$. Then the code (or tree) is a collection of branches and we can think of the code words in a branch as "twigs" on that branch, and finally of the digits in a particular code word (or twig) as leaves on that twig.

Corollary 3.2: The twigs in any branch are increasing in length except for the last two that have the same length and differ only in the last leaf (i.e., digit).

Proof: Corollary 3.2 is a simple consequence of property 3 and corollary 2.1.

(In going from a word ending in one to a word ending in zero, the length can increase, decrease or stay the same, i.e., all three possibilities can occur.)

Property 4: For any GT code the number n_j of code words with prefix 0^j for $j \geq 0$ is zero or a power of 2, the numbers $\{n_j\}$ forming a nonincreasing sequence. For any j for which $n_j > 0$, if we isolate these code words and drop the prefix 0^j then we again obtain a GT code. For $j = 0$ this says that the total number J of code words must be a power of 2; in fact, $J = 2^N$ where N is the number of units being classified. For $j = 1$, if exactly half of the code words start with a zero then the first test is on a single unit; hence if we isolate the code words starting with 1 and drop the prefix 1 then we again obtain a GT code.

Proof: If r units are passed by the first j tests as satisfactory then there are $N-r$ units left to classify and these must lead to exactly 2^{N-r} endpoints, which is the same as the number of code words. The first test is on a single unit if and only if half of the code words start with a zero and then if we drop the common first digit the remainder is a group test on $N-1$ units.

Corollary 4.1: The maximum number of digits in any code word of a GT is $J-1$, where $J = 2^N$ is the number of code words.

Proof: Since $J = 2^N$, the maximal number of tests in any one code word is the number of non-empty subsets of the N units, i.e., $2^N - 1 = J - 1$.

Equality in corollary 4.1 can only hold if all the digits, except possibly the last, are ones. A stronger form of this result will be given in property 6 below.

Property 5: If $(V,0^\alpha)$ and $(V,1^\beta)$ are both code words of the same GT code, then $\alpha \leq \beta$. If $\alpha = \beta$ then the number of units classified in the last α tests equals α and there is no inference associated with these tests. In particular, if the length v of V is zero then $\alpha \leq N \leq \beta$.

Proof: Since the prefix V is in common, there is a common situation for both code words after the tests and the results corresponding to the digity of V. At this common situation the number of unclassified units must be the same. A sequence of α zeros classifies at least α units and a sequence of β ones classifies at most β units. Hence we obtain a contradiction unless $\alpha \leq \beta$, and if $\alpha = \beta$ the common value must be the number of units classified in these α tests If there was any inference associated with these tests then the α zeros would classify at least $\alpha + 1$ units and we could not have $\alpha = \beta$. If the length v of V is zero then the number of unclassified units is N at the outset and the same argument tells us that $\alpha \leq N \leq \beta$.

The following lemma is a generalization of property 3 above. We again assume that V is an arbitrary vector of zeros and ones with length $v \geq 0$.

Lemma 6.1: If a GT code contains all $\cdot 2^j$ code words with the common prefix V and length $v + j$ then the last j tests are on one unit each and there is no inference associated with these j tests.

Proof: We use induction on j; the results hold for $j = 1$ by property 3 above. Since the GT code contains both $(V,0^j)$ and $(V,1^j)$ then by property 5 the last j tests associated with these two words are on one unit each without any inference.

It follows that the last j digits of all the 2^j code words with prefix V, and in particular of $(V,1^j)$, are used to classify exactly j units. Suppose the $v + 1^{st}$ tests are on one unit each and hence the code word $(V,1^j)$ can classify at most $j-1$ units after the prefix V; this is a contradiction of the above statement that j units are classified after the prefix V. Thus the $v + 1^{st}$ test must be on one unit. If the $v + 1^{st}$ test carried any inference with it then by the induction hypothesis the code word $(V,0,1^{j-1})$ has to be shorter than $(V,1^j)$ or else the latter code word would leave at least 1 unit unclassified. Since both words are in the code and they have the same length, we have a contradiction. Hence there cannot be any inference associated with this $v + 1^{st}$ tests, and using the induction hypothesis there cannot be any inference in any of the last j tests; this proves lemma 6.1.

Lemma 6.2: In a GT code if the code word $W' = (V,0,1^{\alpha})$ is followed by $W'' = (V,1,0^{\beta})$ where the common prefix V contains d ones ($\alpha \geq 0$, $d \geq 0$) and $\beta \geq \alpha + d + 1 = \delta$ (say) then there must be an inference associated with at least one of the last δ zeros in W''.

Proof: If the $v + 1^{st}$ test is on a single unit then W' classifies at most δ units after the prefix V, and W'' classifies at least $\beta + 1$ units after the prefix V. Since they must classify the same number of units, $\beta + 1 \leq \delta$ and this contradicts our hypothesis. Hence the $v + 1^{st}$ test must be on u units, where $u \geq 2$. Since W'' has only zeros after the $v + 1^{st}$ test, it must have an additional inference among the last β tests. If this inference is not associated with one of the last δ zeros then the number of units classified by W'' after the prefix V is at least $u + \delta$. The number of units classified by W' after the prefix V is at most $u + \delta - 1$. Since they must classify the same number of units, we have a contradiction which proves lemma 6.2.

Corollary 6.1: If $W' = (0^{\gamma})$ for $\gamma \geq 1$ is followed by $W'' = (0^{\gamma-1},1,0^{\beta})$ where $\beta > 0$, then there must be a single inference associated with the last zero in W''.

Proof: Set $\alpha = d = 0$ above.

Property 6: In a GT code the code word $W' = (V,0,1^{\alpha})$ with prefix V containing d ones ($\alpha \geq 0$, $d \geq 0$) cannot be followed by the complete set of 2^{δ} code words of the form $(V,1,0^{\beta-\delta}0^{\delta}),\ldots,(V,1,0^{\beta-\delta}1^{\delta})$, all having the same length $v + 1 + \beta$, if $\beta \geq \delta = \alpha + d + 1$.

Proof: By lemma 6.2 there must be an inference associated with at least one of the last δ zeros of $(V,1,0^{\beta})$. However by lemma 6.1 (using $V^* = (V,1,0^{\beta-\delta})$ as a new prefix) there is no inference associated with the last δ tests. This contradiction proves property 6.

Illustrations: Four examples of codes for $N = 3$ and $v = 0$ that are ruled out as GT codes by property 6:

$\alpha = 0;\ \beta = 2$	$\alpha = 0;\ \beta = 2$	$\alpha = 0;\ \beta = 3$	$\alpha = 1;\ \beta = 3$
0	0	0	00
100	100	1000	01
101	101	1001	1000
110	1100	1010	1001
11100	1101	1011	1010
11101	1110	110	1011
11110	11110	1110	110
11111	11111	1111	111

These examples are not ruled out by any of the previous properties of a GT code.

For any code with $J = 2^N$ code words let $d = d(W)$ denote the number of ones in W and let

$$(A2) \qquad h(d;N) = \begin{cases} 0 & \text{if } d = 0 \\ 1 & \text{if } 0 < d \leq 2^{N-1} = 2^N - 2^{N-1} \\ \cdots\cdots & \cdots\cdots \\ j & \text{if } 2^{N-j+1}(2^{j-1}-1) < d \leq 2^{N-j}(2^j-1) \\ \cdots\cdots & \cdots\cdots \\ N & \text{if } 2^N-2 < d \leq 2^N-1 \quad (\text{i.e., if } d = 2^N-1) \\ \infty & \text{if } d > 2^N-1. \end{cases}$$

Then $h(d;N)$ is the minimal number of defectives that must be present under W if d different tests fail. Let $g = g(W) \geq 0$ denote the number of zeros in W, so that $N^* = N - g$ is an upper bound to the number of defectives under W. Then

$h(d;N^*) \geq h(d;N)$ is a better lower bound to the number of defectives present under W since $h(d;N)$ is a nondecreasing function of N.

Property 7: For any code word W in a GT code with $J = 2^N$

(A3) $\qquad h(d;N^*) \leq N^*.$

Proof: Since the left side of (A3) is a lower bound to the number of defective units and the right side is an upper bound, the result follows.

Illustration: The following two codes for $N = 3$ are ruled out by property 7:

00	0
01	1000
100	10010
1010	10011
10110	1010
10111	1011
110	110
111	111

They are not ruled out by any of the previous properties of a GT code. It is interesting to note that the weaker form of property 7 which states that

(A4) $\qquad h(d;N) \leq N^*$

rules out the second code above but not the first. In particular, the 5^{th} word of the first code gives $N^* = 1$ and $h(3;1) = 2$.

REFERENCES

[1] Dorfman, Robert, (1943), "The Detection of Defective Members of Populations," Ann. of Math. Stat., 14, pp. 436-440.

[2] Finucan, H. M., (1964), "The blood testing problem," Applied Statistics, 13, pp. 43-50.

[3] Hardy, G. H., Littlewood, J. E. and Polya, G., (1959), Inequalities, Cambridge at the University Press.

[4] Huffman, David A., (1952), "A Method for the Construction of Minimum Redundancy Codes," Proc. I. R. E., 40, p. 1098.

[5] Picard, Claude, (1965), Theorie des Questionnaires, Gauthiers-Villars, Paris.

[6] Sandelius, Martin, (1961), "On an Optimal Search Procedure," Amer. Math. Monthly, 68, No. 2, pp. 133-134.

[7] Sobel, Milton and Groll, Phyllis A., (1959), "Group Testing to Eliminate Efficiently all Defectives in a Binomial Sample," Bell System Tech. Jour., 38, pp. 1179-1252.

[8] Sobel, Milton, (1960), "Group Testing to Classify Efficiently All Units in a Binomial Sample," an article in Information and Decision Processes, edited by Machol, R. E., McGraw Hill Book Co.

[9] Sobel, Milton, (1964), "Optimal Group Testing," Technical Report No. 72, Stanford University.

[10] Sobel, Milton and Groll, Phyllis, (1966), "Binomial Group Testing With an Unknown Proportion of Defectives," Technometrics, 8, No. 4.

[11] Sobel, Milton, (1967), "Binomial and Hypergeometric Group Testing," accepted for publication in Studia Scientiarum Mathematicarum Hungarica.

[12] Sterrett, Andrew, (1957), "On the Detection of Defective Members of Large Populations," Ann. Math. Stat., 28, No. 4.

[13] Ungar, Peter, (1960), "The Cut-Off Point for Group-Testing," Comm. Pure Appl. Math., 13, pp. 49-54.

[14] Zimmerman, S., (1959), "An Optimal Search Procedure," Amer. Math. Monthly, 66.