COMBINATORIAL ANALYSIS AND COMPUTERS

MARSHALL HALL, JR. AND D. E. KNUTH, California Institute of Technology

1. Introduction. The "logical decision-making" characteristics of computers enable them to attack many problems which are not basically numerical. Combinatorial constructions and searches are applications of this kind in which computers have been very successful. The two principal ways in which computers have been used in these problems are: 1) generation of a sequence of combinatorial patterns as part of a larger problem, and 2) constructions and searches for which hand calculations are unfeasible.

Since the size of combinatorial problems grows very rapidly, we can usually expect the computer to do only about one case larger than can be done by hand. For example, the problem of constructing a finite plane of order n is equivalent to choosing, in a restricted way, $(n-1)^2$ of the $n!$ permutations on n letters; thus, changing n to $n+1$ will make the problem many orders of magnitude larger.

2. Combinatorial sequences. Many computer applications call for the generation of combinatorial sequences such as the set of all permutations on n elements, the set of all combinations of m things taken n at a time, and so on.

The simplest combinatorial sequence is the set of all ordered n-tuples $(x_1, x_2, \cdots, x_n)$ for which $0 \leqq x_i < m$, $1 \leqq i \leqq n$. This set can, of course, be generated by repeatedly adding 1 in the m-ary number system; in applications, however, it is usually more useful if the n-tuples are generated in such a way that only one x_i changes at each step [5, 7]. The simplest method for doing this is the following: At the kth step, $1 \leqq k < m^n$, add 1 to x_{n-i}(modulo m) if m^i is the highest power of m dividing k. Whenever k is a multiple of m, it is not difficult to prove that this will be the digit immediately left of the rightmost nonzero digit, assuming $(0, 0, \cdots, 0)$ was the starting n-tuple. The case $m=2$ gives rise to the so-called Gray binary numbers which have found many engineering applications, and the sequence is the well-known procedure for solving the classical Chinese Ring Puzzle [2].

The problem of generating all permutations has stimulated much interest [1, 27, 40]. A method which is at least 150 years old [6], and which is still being rediscovered fairly often, is the following: Permutations can be generated in lexicographic order if we (1) obtain the successor of $(x_1, x_2, \cdots, x_n)$ by finding r, s such that

$$x_r < x_{r+1} \geqq x_{r+2} \geqq \cdots \geqq x_n, \qquad x_{r+s} > x_r \geqq x_{r+s+1},$$

then (2) interchange x_r and x_{r+s}; and (3) replace $(x_{r+1}, x_{r+2}, \cdots, x_n)$ by $(x_n, \cdots, x_{r+2}, x_{r+1})$. This method is valid even when the elements permuted are not distinct.

As above, however, it is usually advantageous to generate the permutations in such a way that successive permutations are "near" each other, i.e., only two elements have been interchanged. The first method of this kind was due to Wells [45], but an improved algorithm was later discovered almost simultaneously by Trotter [42], and by Johnson [22]. In the latter method, all permutations of n distinct objects are obtained by the successive interchange of two adjacent elements, by using essentially the following recursive rule: Let $(y_1, y_2, \cdots, y_{n-1})$ be the kth permutation of the sequence for all permutations on $(1, 2, \cdots, n-1)$; then the $(kn+1)$-st, $(kn+2)$-nd, $\cdots$, $(k+1)n$-th permutations on $(1, 2, \cdots, n)$ are

$$(n, y_1, y_2, \cdots, y_{n-1}), (y_1, n, y_2, \cdots, y_{n-1}),$$
$$(y_1, y_2, n, \cdots, y_{n-1}), \cdots, (y_1, y_2, \cdots, y_{n-1}, n)$$

if k is even, and the same sequence in reverse order if k is odd. For example, here is the set of all 24 permutations on 4 elements obtained by repeated interchanges of adjacent elements:

1234	3124	2314
1243	3142	2341
1423	3412	2431
4123	4312	4231
4132	4321	4213
1432	3421	2413
1342	3241	2143
1324	3214	2134

This technique is the fastest method for generating the complete set of permutations on a computer.

Similarly, an algorithm is known for successively choosing all combinations of m things, n at a time, by changing the choice of only one object each time. Algorithms for obtaining all partitions of an integer n are given in [38]; the partitions of a set are produced by the algorithm in [19]; and many similar techniques are known.

3. Backtrack. The basic method [9, 43] used for most combinatorial searches has been christened "backtrack" technique by D. H. Lehmer. A great many combinatorial problems can be stated in the form, "Find all vectors $(x_1, x_2, \cdots, x_n)$ which satisfy p_n," where $x_1, x_2, \cdots, x_n$ are to be chosen from some finite set of m distinct objects, and p_n is some property. The combinatorial sequences of the preceding section all fit into this framework; for example, if we are to generate all *permutations* of distinct objects, the property p_n states simply that $x_i \neq x_j$ if $i \neq j$. If we wish to generate all *combinations* of the numbers $(1, 2, \cdots, m)$ taken n at a time, p_n is the property

$$1 \leqq x_1 < x_2 < \cdots < x_n \leqq m.$$

The backtrack method consists of defining properties p_k for $1 \leqq k \leqq n$ in such a way that whenever $(x_1, x_2, \cdots, x_n)$ satisfies p_n, then $(x_1, \cdots, x_k)$ necessarily satisfies p_k. The computer is programmed in a straightforward manner to consider only those partial solutions $(x_1, \cdots, x_k)$ which satisfy p_k; if p_k is not satisfied, the m^{n-k} vectors $(x_1, \cdots, x_k, x_{k+1}, \cdots, x_n)$ are not examined by the program. If the properties p_k can be chosen in an efficient way, comparatively few cases are considered.

Although the backtrack approach frequently gives satisfactory results, there are equally many cases on record for which it has failed; in fact, it is not uncommon (see [**31**]) to find situations in which thousands of centuries would be required for the algorithm to be completed! For this reason, it is desirable to have some convenient means for estimating the computational time before putting the algorithm onto a computer. One can proceed by playing the following little game: Let a_1 be the number of (x_1) which satisfy p_1; then let y_1 be one such solution, chosen at random with probability $1/a_1$. Similarly let a_k be the number of x_k for which $(y_1, \cdots, y_{k-1}, x_k)$ satisfies p_k; randomly choose y_k to be one such x_k, with probability $1/a_k$. If $a_k = 0$ or $k = n$, the game terminates. It is not difficult to verify that the expected value of

$$a_1 + a_1a_2 + a_1a_2a_3 + \cdots + a_1a_2 \cdots a_n$$

will be precisely the number of cases which are examined by the backtrack algorithm, and therefore by playing this game several times one obtains a good idea of the size of this number. When the number is greater than, say, 10 million, the backtrack method is probably unfeasible with contemporary equipment.

This random process can be combined with backtracking to give "random" solutions $(x_1, x_2, \cdots, x_n)$ to the combinatorial problem (see [**30**]). The solutions, however, are not truly random in general, i.e., they are not necessarily obtained with equal probability. For example, of the 10×10 latin squares shown in Fig. 1, square B [**31, 32**] would be obtained with probability 9.0×10^{-27}, assuming the first row and first column are fixed; square A, which certainly appears to be "less random," is actually produced with the much higher probability $9^{-1}8^{-3}7^{-5}6^{-7}5^{-6}4^{-4}3^{-2} = 1.3 \times 10^{-21}$.

A	*B*	*C*
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9 0	1 8 3 2 5 4 7 6 9 0	9 6 5 4 7 8 3 0 1 2
2 3 4 5 6 7 8 9 0 1	2 9 5 6 3 0 8 4 7 1	1 3 8 5 2 6 0 9 4 7
3 4 5 6 7 8 9 0 1 2	3 7 0 9 8 6 1 5 2 4	7 8 4 0 5 3 2 1 9 6
4 5 6 7 8 9 0 1 2 3	4 6 7 5 2 9 0 8 1 3	5 2 9 6 3 7 8 4 0 1
5 6 7 8 9 0 1 2 3 4	5 0 9 4 7 8 3 1 6 2	3 9 6 2 0 1 4 5 7 8
6 7 8 9 0 1 2 3 4 5	6 5 4 7 1 3 2 9 0 8	8 4 0 1 6 9 7 2 5 3
7 8 9 0 1 2 3 4 5 6	7 4 1 8 0 2 9 3 5 6	6 7 3 9 1 0 5 8 2 4
8 9 0 1 2 3 4 5 6 7	8 3 6 0 9 1 5 2 4 7	2 0 1 7 8 4 9 6 3 5
9 0 1 2 3 4 5 6 7 8	9 2 8 1 6 7 4 0 3 5	4 5 7 8 9 2 1 3 6 0

FIG. 1. Three latin squares of order 10. Squares B and C are orthogonal.

The backtrack algorithm usually produces its results in lexicographic order, and this fact can often be utilized to shorten the computational time greatly by eliminating isomorphic solutions. If it can be established that any completion of the partial solution $(x_1, x_2, \cdots, x_k)$ must be isomorphic to a solution $(y_1, y_2, \cdots, y_n)$ with $(y_1, \cdots, y_k) < (x_1, \cdots, x_k)$ in the lexicographic order, we may omit consideration of $(x_1, x_2, \cdots, x_k)$. (This principle is to be incorporated into the property p_k.) This and other methods for rejecting isomorphic solutions are discussed by Swift [**39**]. A technique for possibly speeding up backtrack solutions by preliminary identification of objects has been suggested by Tompkins [**41**].

Backtrack has been used in most of the applications discussed later in this paper; it has also been used to determine all 80 Steiner triple systems of order fifteen [**13**], to construct certain codes [**20, 46**], to discover an Hadamard matrix of order 92 [**8**], to study the four-color problem [**47**], and to do problems with a more "recreational" flavor such as the fitting together of geometric shapes [**36**]. The techniques of dynamic programming [**3**] have proved useful as an alternative to backtrack in some combinatorial applications.

4. Latin Squares. Parker [**32**] has discovered an ingenious way to apply the backtrack method for determining latin squares orthogonal to a given one. A straightforward method would be to successively fill each cell of the potential orthogonal mate. Parker's method, however, proceeds in two steps: He first applies backtrack to obtain the set of all *transversals* of the given square; a transversal is a set of cells with exactly one in each row, one in each column, and one containing each symbol. The problem of finding an orthogonal mate to the square is equivalent to finding a set of disjoint transversals to cover the square, so the second stage is to use backtrack again in order to find such coverings by transversals. With 10×10 latin squares, this reduces the number of cases to be considered from approximately 10^{17} to approximately 5×10^5, a factor of some 200 billion! The total time required, when the problem is split into two parts in this way, may be thought of as the sum of the time for two separate steps rather than the product.

Parker used his program to show that square C of Fig. 1 is the only latin square orthogonal to square B. On the other hand, square A is known to have no transversals, hence no orthogonal mate, by theorems proved in [**12**]. Parker's program has been applied to nearly one hundred 10×10 latin squares and the majority of these have possessed orthogonal mates. This is quite remarkable, since Euler's conjecture that no such squares exist had been believed for so many years. Yet no triple of mutually orthogonal squares has ever been found after much computer searching, and more theoretical advances will be necessary before this problem can be reduced to a size computers can handle.

A set of five mutually orthogonal 12×12 latin squares has been found by computer [**4, 21**]. Starting with the multiplication table of a group which has the elements $x_1, x_2, \cdots, x_n$, the square formed by permuting the rows so that the first column has the form $y_1, y_2, \cdots, y_n$ will be orthogonal if and only if

$x_1y_1^{-1}, x_2y_2^{-1}, \cdots, x_ny_n^{-1}$ is a permutation of the group elements. We may assume y_1 is the identity. If we start with the Abelian group having elements $x_i(0 \leqq x < 6,\ 0 \leqq i < 2)$, with product $x_iy_j = (x+y) \bmod 6_{(i+j) \bmod 2}$, the following first columns will yield five mutually orthogonal latin squares:

0_0	1_0	2_0	3_0	4_0	5_0	0_1	1_1	2_1	3_1	4_1	5_1
0_0	2_0	1_0	4_1	3_1	5_1	3_0	5_0	4_0	1_1	0_1	2_1
0_0	4_1	4_0	2_1	2_0	0_1	3_1	1_0	1_1	5_0	5_1	3_0
0_0	5_0	3_1	2_0	4_1	1_1	2_1	4_0	5_1	1_0	3_0	0_1
0_0	0_1	5_0	1_1	5_1	4_0	2_0	2_1	4_1	3_0	1_0	3_1

It has been shown that no set of six mutually orthogonal latin squares of this kind exist. Similarly, if we start with the non-Abelian group having elements x_i (x a permutation on 3 letters, $0 \leqq i < 2$), and multiplication defined by multiplying the permutations and adding the subscripts (mod 2), we can get three mutually orthogonal squares:

$$(1)_0\ (12)_0\ (13)_0(23)_0(123)_0(132)_0\ \ (1)_1\ (12)_1\ (13)_1\ (23)_1(123)_1(132)_1$$

$$(1)_0(123)_1\ (12)_0(12)_1\ (23)_1\ (13)_0\ (13)_1\ (23)_0(123)_0(132)_1\ \ (1)_1(132)_0$$

$$(1)_0(132)_1(123)_0(13)_1\ (23)_0\ (23)_1(132)_0(123)_1\ (13)_0\ (12)_1\ (12)_0\ \ (1)_1$$

Starting with the alternating group on 4 elements, there are 3840 pairs of orthogonal squares but no three mutually orthogonal. From this data it appears that as the group gets less Abelian, less orthogonal squares are present, although no theoretical grounds for such behavior are known at this time.

5. Projective planes and symmetric block designs. More computer studies have been made relating to finite projective geometries than to any other branch of combinatorial analysis. A survey of this work appears in [**16**] and we mention only the principal results here.

It is easy to show by hand computation that the projective planes of orders 2, 3, 4, and 5 are unique. The Bruck-Ryser theorem shows that there is no plane of order 6. There is a unique plane of order 7; this result was first established by Norton [**29**], being incidental to his listing of the 147 nonisomorphic latin squares of order 7. Norton found only 146; an omission was found by Sade [**34**] who confirms the completeness of the list of 147. A more direct proof of the uniqueness of the plane of order 7 is given by Hall [**10**] and Pierce [**33**].

In a plane of order 8, the multiplication table of nonzero elements forms a 7×7 latin square. Using the list of 147 such squares and a few further simplifications, a computer was programmed to help establish the uniqueness of the plane of order 8 [**11**]; this proof is a good example of the effective interplay of machine and hand methods. There are four known projective planes of order 9, and work is under way to determine whether these are in fact the complete set. This problem appears to lie at the borderline of computational feasibility with respect to an exhaustive search for planes of a small order. Since planes exist for every order which is a prime or prime power, the question as to the existence of any plane of nonprime-power order is of considerable interest, and the smallest order of this kind permitted by the Bruck-Ryser theorem is order 10. A search is under way for a plane of order 10 with a nontrivial collineation, but even this is a large undertaking. A plane of order 10 is equivalent to a set of nine mutually orthogonal 10×10 latin squares, but as stated above not even three such squares are known.

If we can find residues $b_1, b_2, \cdots, b_k$(modulo v) where every difference $d \not\equiv 0$ (mod v) has exactly λ representations $d \equiv b_i - b_j$ (modulo v) we call the b's a difference set modulo v. The relation $k(k-1) = \lambda(v-1)$ must necessarily hold. If $b_1, \cdots, b_k$ are a difference set modulo v, then the "blocks" B_j: $b_1+j, b_2+j, \cdots, b_k+j$(mod v) $j=0, \cdots, v-1$ form a symmetric block design. In the special case that $\lambda=1$, $k=n+1$, $v=n^2+n+1$, the block design is a finite projective plane of order n, and the blocks are the lines of the plane. For example, the following difference sets may be used to construct the planes of small orders:

$$n = 2: \quad 1, 2, 4 \ (\text{mod } 7)$$

$$n = 3: \quad 0, 1, 3, 9 \ (\text{mod } 13)$$

$$n = 4: \quad 3, 6, 7, 12, 14 \ (\text{mod } 21).$$

By a theorem of one of the authors [**14**], under certain conditions we may assume that the difference set $b_1, \cdots, b_k$ is fixed by "multipliers," where a multiplier is a prime p dividing $k-\lambda$. If p is such a multiplier then $pb_1, \cdots, pb_k$(mod v) are the same as $b_1, \cdots, b_k$ in some order. This fact considerably simplifies a computer search for difference sets, and those of small order are discussed in [**14, 17**]. In particular, the SWAC computer was used to show that all difference sets with $\lambda=13$, $k=40$, $v=121$ are isomorphic to one of the following four:

1, 3, 4, 7, 9, 11, 12, 13, 21, 25, 27, 33, 34, 36, 39, 44, 55, 63, 64, 67, 68, 70, 71, 75, 80, 81, 82, 83, 85, 89, 92, 99, 102, 103, 104, 108, 109, 115, 117, 119.

1, 3, 4, 5, 9, 12, 13, 14, 15, 16, 17, 22, 23, 27, 32, 34, 36, 39, 42, 45, 46, 48, 51, 64, 66, 69, 71, 77, 81, 82, 85, 86, 88, 92, 96, 102, 108, 109, 110, 117.

1, 3, 4, 7, 8, 9, 12, 21, 24, 25, 26, 27, 34, 36, 40, 43, 49, 63, 64, 68, 70, 71, 72, 75, 78, 81, 82, 83, 89, 92, 94, 95, 97, 102, 104, 108, 112, 113, 118, 120.

1, 3, 4, 5, 7, 9, 12, 14, 15, 17, 21, 27, 32, 36, 38, 42, 45, 46, 51, 53, 58, 63, 67, 68, 76, 79, 80, 81, 82, 83, 96, 100, 103, 106, 107, 108, 114, 115, 116, 119.

Parker has recently shown that these four lead to nonisomorphic designs, by having a computer determine the intersection patterns of three blocks at a time.

When $\lambda=1$, the difference set will always give a projective plane, and all primes p dividing $n=k-\lambda$ must be multipliers. Singer [**37**] proved that all finite Desarguesian planes can be constructed by using difference sets; Mann and Evans [**28**], by hand computation, have shown that for $n \leqq 1600$ no projective planes can be constructed from a difference set unless n is the power of a prime.

Other searches have been bases on the algebraic properties of the planes. Killgrove [**23**] showed that no plane of order 9 has a cyclic additive group, and it was determined in [15] that the only planes with an elementary Abelian additive group are the four known ones. Therefore addition in a further plane of order 9, if there is one, cannot be a group.

Kleinfeld [24] found all Veblen-Wedderburn systems of order 16 which have a left nucleus of order 4. These systems lead to four planes: the Desarguesian plane, the Hall plane, and two new planes coordinatized by semifields (nonassociative division rings). The two latter planes inspired constructions of several new types of projective planes [18, 25, 35].

Non-Desarguesian planes of all orders p^n, where p is prime, $n>1$, were known except for the cases 2^p for prime p. As mentioned above, there are no non-Desarguesian planes of order 2^2 or 2^3, and so there was interest in the next smallest case 2^5. A search was undertaken for all semifields of order 32, independently by Walker [44] and one of the authors, and the complete set of 2502 nonisomorphic systems was found. These yield five non-Desarguesian planes whose structure was also determined by computer [25]. Fortunately, it was possible to observe a pattern in one of the semifields, and this led to a construction [26] of semifields of all orders 2^{mn} where $mn>3$ and $n>1$ is odd. In this way the solution for all remaining cases 2^p was obtained as a byproduct of a computer examination of the smallest case; such results are the main goals of combinatorial computer explorations.

The preparation of this paper was supported, in part, by NSF grant number GP-212.

References

1. Algorithm index, Commun. ACM, 7 (1964) 147.

2. W. W. Rouse Ball, Mathematical Recreations and Essays, rev. by H. S. M. Coxeter, Macmillan, New York, 1947, 305–310.

3. R. Bellman, Dynamic programming treatment of the traveling salesman problem, J. ACM, 9 (1962) 61–63.

4. R. C. Bose, I. M. Chakravarti, and D. E. Knuth, On methods of constructing sets of mutually orthogonal latin squares using a computer, Technometrics, 2 (1960) 507–516; *ibid.*, 3 (1961) 111–117.

5. Martin Cohen, Affine m-ary Gray codes, Information and Control, 6 (1963) 70–78.

6. L. J. Fischer and K. Chr. Krause, Lehrbuch der Combinationslehre und der Arithmetik, Dresden, 1812.

7. Ivan Flores, Reflected number systems, IRE Trans., EC-5 (1956) 79–82.

8. S. W. Golomb and L. D. Baumert, The search for Hadamard matrices, this MONTHLY, 70 (1963) 12–17.

9. ———, Backtrack programming, submitted for publication.

10. Marshall Hall, Jr., Uniqueness of the projective plane with 57 points, Proc. Amer. Math. Soc., 4 (1953) 912–916; correction *ibid.*, 5 (1954) 994–997.

11. Marshall Hall, Jr., J. D. Swift, and R. J. Walker, Uniqueness of the projective plane of order eight, Math. Tables Aids Comput., 10 (1956) 186–194.

12. Marshall Hall, Jr. and L. J. Paige, Complete mappings of finite groups, Pacific J. Math., 5 (1955) 541–549.

13. Marshall Hall, Jr. and J. D. Swift, Determination of Steiner triple systems of order 15, Math. Tables Aids Comput., 9 (1955) 146–152.

14. Marshall Hall, Jr., A survey of difference sets, Proc. Amer. Math. Soc., 7 (1956) 975–986.

15. Marshall Hall, Jr., J. D. Swift, and R. Killgrove, On projective planes of order nine, Math. Tables Aids Comput., 13 (1959) 233–246.

16. Marshall Hall, Jr., Numerical analysis of finite geometries, IBM J. Res. Dev. (to appear).

17. Harry S. Hayashi, Computer investigation of difference sets, submitted for publication.

18. D. R. Hughes and Erwin Kleinfeld, Semi-nuclear extensions of Galois fields, Amer. J. Math., 82 (1960) 389–392.

19. G. Hutchinson, Partitioning algorithms for finite sets, Commun. ACM, 6 (1963) 613–614.

20. B. H. Jiggs, Recent results in comma-free codes, Canad. J. Math., 15 (1963) 178–187.

21. Diane M. Johnson, A. L. Dulmage, and N. S. Mendelsohn, Orthomorphisms of groups and orthogonal latin squares, I, Canad. J. Math., 13 (1961) 356–372.

22. Selmer M. Johnson, Generation of permutations by adjacent transposition, Math. Comp., 17 (1963) 282–285.

23. R. Killgrove, A note on the nonexistence of certain projective planes of order nine, Math. Comput., 14 (1960) 70–71.

24. E. Kleinfeld, Techniques for enumerating Veblen-Wedderburn systems, Jour. ACM, 7 (1960) 330–337.

25. Donald E. Knuth, Finite semifields and projective planes, to appear in J. of Algrebra.

26. ———, A class of projective planes, to appear in Trans. Amer. Math. Soc.

27. D. H. Lehmer, Teaching combinatorial tricks to a computer, AMS Proc. Symp. Appl. Math., 10 (1960) 179–193.

28. H. B. Mann and T. A. Evans, On simple difference sets, Sankhya, 11 (1951) 357–364.

29. H. W. Norton, The 7×7 squares, Ann. Eugenics, 9 (1939) 269–307.

30. R. T. Ostrowski and K. D. Van Duren, On a theorem of Mann on latin squares, Math. Comp., 15 (1961) 293–295.

31. L. J. Paige and C. B. Tompkins, The size of the 10×10 orthogonal latin square problem, AMS Proc. Symp. Appl. Math., 10 (1960) 71–84.

32. E. T. Parker, Computer investigation of orthogonal latin squares of order ten, AMS Proc. Symp. Appl. Math., 15 (1962) 73–82.

33. W. A. Pierce, The impossibility of Fano's configuration in a projective plane with eight points per line, Proc. Amer. Math. Soc., 4 (1953) 908–912.

34. A. Sade, An omission in Norton's list of 7×7 squares, Ann. Math. Statist., 22 (1951) 306–307.

35. Reuben Sandler, Autotopism groups of some finite nonassociative algebras, Amer. J. Math., 84 (1962) 239–264.

36. Scott, Programming a combinatorial puzzle, Dept. Elec. Eng. Princeton U., 10 June 1958.

37. J. Singer, A theorem in finite projective geometry and some applications to number theory, Trans. Amer. Math. Soc., 43 (1938) 377–385.

38. F. Stockmal, Algorithms 95 and 114, Commun. ACM, 5 (1962) 344, 434.

39. J. D. Swift, Isomorph rejection in exhaustive search techniques, AMS Proc. Symp. Math., 10 (1960) 195–200.

40. C. B. Tompkins, Machine attacks on problems whose variables are permutations, AMS Proc. Symp. Appl. Math., 6 (1956) 195–212.

41. ———, Methods of successive restrictions in computational problems involving discrete variables, AMS Proc. Symp. Appl. Math., 15 (1962) 95–106.

42. H. F. Trotter, Algorithm 115, Commun. ACM, 5 (1962) 434–435.

43. R. J. Walker, An enumerative technique for a class of combinatorial problems, AMS Proc. Symp. Appl. Math., 10 (1960) 91–94.

44. ———, Determination of division algebras with 32 elements, AMS Proc. Symp. Appl. Math., 15 (1962) 83–85.

45. Mark B. Wells, Generation of permutations by transposition, Math. Comp., 15 (1961) 192–195.

46. Wozencraft and Reissen, Sequential Decoding, MIT Press, 1961.

47. H. Yamabe and D. Pope, A computational approach to the four-color problem, Math. Comp., 15 (1961) 250–253.