MACHINE ATTACKS ON PROBLEMS WHOSE VARIABLES ARE PERMUTATIONS

BY

C. TOMPKINS

1. Introduction. This paper was prepared in order to show by example how modern high-speed electronic digital computers may be applied to problems whose variables are discrete. It contains a more or less journalistic account of the application of the National Bureau of Standards Western Automatic Computer (SWAC) to problems whose variables are permutations. Several problems of this type have been attacked computationally on SWAC during the last few years. They seem reasonably typical of problems in which operations other than the standard arithmetic operations of addition and multiplication on real numbers play an important or dominant role; substitutions according to a group multiplication table may be typical of these operations.

The speed with which modern machines such as SWAC can carry out arithmetic operations (SWAC requires $6.4 \cdot 10^{-5}$ sec. to draw two numbers from its memory, add them, store the result and determine what to do next) makes large exhaustive searches feasible and attractive. Explicitly, it is not conceivable that a person would try a hand search through all permutations on 10 marks to arrive at a solution of a problem, for there are 3,628,800 of these permutations. If such a problem is to be solved by hand, some method of rejecting large classes of permutations so that the search is reduced to a reasonable size is imperative. However, if the nature of the problem is such that SWAC can generate each permutation and test it as a potential solution to the problem, using, say, 200 additions (or an equivalent amount of computing time) in the generation and testing of each permutation, then the time required for the search through all permutations is about thirteen hours; in this case the expenditure of much energy in reducing the size of the search becomes unattractive.

However, SWAC, and most of the other famous computers, were designed for application to problems which involve only the usual arithmetic operations on real numbers, and there are real difficulties and inefficiencies in applying them to discrete-variable problems of the type described here. These difficulties are largely associated with the rejection of isomorphic or equivalent trial solutions. Only the most modest progress has been made on this front. This limitation of progress is partly due to the lack of versatile comparison and substitution devices and facilities in the machines, and it is partly due to the difficulties inherent in expressing criteria for rejecting isomorphs in terms of elementary operations suitable for any machine which might be built now.

If discrete-variable problems continue to increase in economic importance

(*e.g.*, several places in industry may furnish "assignment problems" [1][1] of the size of the problem mentioned above and of nonlinear or traveling-salesman type, and these may presently require exhaustive search for solution and may well become economically important), it may be expected that enhanced comparison and substitution facilities will be developed for some machines. No attempt will be made here to examine engineering aspects of this development. What may be discernible here will be a few techniques which may be applicable to the most elementary problems involving permutations as variables, and from these there may be some gain to other workers in formulating more subtle techniques of analysis which may lead to rejection criteria suitable for high-speed automatic computation.

For mathematical problems, one other application of modern high-speed digital computing equipment should be mentioned here even though this application is not explicitly stressed in the main body of the paper. This application is the potential use of the coded command sequence for the machine in the complete logical exposition of a result involving lengthy calculation.

The coded command sequence provides a feasible means of complete logical exposition of lengthy and heretofore not reasonably describable calculations, and it thus provides a means whereby users of these results can verify their logical accuracy. This sequence of commands is stored within the machine's memory, and because of the limited capacity of this memory (4,352 numbers, each containing 36 binary digits and a sign in SWAC) the sequences are necessarily short. Information is packed densely into such sequences in the form of instructions to the machine which cause it to modify the coded commands for reuse as the calculation proceeds. Thus the coded command sequence furnishes a fairly efficient and a completely stated description of the calculation. This is a contribution to mathematics and to mathematical exposition even in cases where the use of the machine leads to no saving in calculation time (which may happen if complicated rejection criteria not easily coded for the machine are applicable to reduce the hand-computing time). Lengthy and laborious calculations such as those performed by Tarry [2] in proving that no pair of orthogonal latin squares of order 6 can exist or those performed by White, Cole, and Cummings [3] in listing the essentially different systems of Steiner triples of order 15, have not heretofore been feasibly publishable; the authors of such papers have been in the uncomfortable position of having to ask the public to accept on faith the accuracy and, to some extent, the logic of their calculation. It now becomes possible to use the short coded command sequence to present this logic with reasonably modest demands on the publisher and the reader and thus to permit the reader to estimate the probability of accurate calculation and to facilitate confirming recalculation by other workers if this becomes desirable.

[1] It may be noted here that the bibliography is intended to provide easily available expansion of some of the concepts which arise in this paper; there has been no effort to list the earliest or the most important works or to give a complete listing.

2. A convenient class of problems. In the next section a method will be developed for the systematic generation of permutations in a convenient order. The order in which permutations are generated by this method is illustrated in Table I (to be read down the successive columns), in which the first 24 permutations on five marks are exhibited, some more than once. The feature to be noticed is that the permutations are arranged in an order which groups all permutations whose first marks are the same; specifically, the permutations shown in Table I are those whose first mark is 1, all those in any column have the same second mark, and those in the top, middle, or bottom third of a column have the same third mark. The starred permutations are duplicates of permutations which have appeared earlier, and they are to be ignored (their presence in the table will be explained in the next section).

TABLE I

12345	15234	14523	13452
12354	15243	14532	13425
12345*	15234*	14523*	13452*
12534	15423	14352	13245
12543	15432	14325	13254
12534*	15423*	14352*	13245*
12453	15342	14235	13524
12435	15324	14253	13542
12453*	15342*	14235*	13524*
12345**	15234**	14523**	13452**

Several problems will be outlined in which this arrangement of permutations may be exploited to reject whole blocks of consecutively listed permutations and thus to reduce the length of search required. The earlier of these problems will be drawn from classical problems of pure mathematics, and some later ones are of interest in econometrics. The final two paragraphs summarize the section.

By way of notation, the permutations listed will be considered to be the images of the marks 1, 2, 3, $\cdots$, n for some value of n ($n = 5$ in Table I) under a permutation transformation; the image of the mark i will be denoted p_i.

A simple problem which illustrates the technique to be used is the following: How many permutations on the five marks 1, 2, 3, 4, 5 have the property that $p_i - i \equiv p_j - j$ (modulo 5) implies that $i = j$?

In solving this problem, it is possible first to remove most of the potential trials as isomorphs of the ones which will be made based on the permutations listed in Table I. It is clear that any permutation satisfying the requirement of the problem must take exactly one mark into itself, and it is not hard to show that each such permutation is equivalent (by subtraction of a constant number from both the argument mark and the image mark) uniquely to one which takes the mark 1 into itself; *i.e.*, there are five times as many permutations fulfilling the requirements as the number which can be found among the permutations of Table I, each of which transforms the mark 1 into itself.

(*e.g.*, several places in industry may furnish "assignment problems" [1][1] of the size of the problem mentioned above and of nonlinear or traveling-salesman type, and these may presently require exhaustive search for solution and may well become economically important), it may be expected that enhanced comparison and substitution facilities will be developed for some machines. No attempt will be made here to examine engineering aspects of this development. What may be discernible here will be a few techniques which may be applicable to the most elementary problems involving permutations as variables, and from these there may be some gain to other workers in formulating more subtle techniques of analysis which may lead to rejection criteria suitable for high-speed automatic computation.

For mathematical problems, one other application of modern high-speed digital computing equipment should be mentioned here even though this application is not explicitly stressed in the main body of the paper. This application is the potential use of the coded command sequence for the machine in the complete logical exposition of a result involving lengthy calculation.

The coded command sequence provides a feasible means of complete logical exposition of lengthy and heretofore not reasonably describable calculations, and it thus provides a means whereby users of these results can verify their logical accuracy. This sequence of commands is stored within the machine's memory, and because of the limited capacity of this memory (4,352 numbers, each containing 36 binary digits and a sign in SWAC) the sequences are necessarily short. Information is packed densely into such sequences in the form of instructions to the machine which cause it to modify the coded commands for reuse as the calculation proceeds. Thus the coded command sequence furnishes a fairly efficient and a completely stated description of the calculation. This is a contribution to mathematics and to mathematical exposition even in cases where the use of the machine leads to no saving in calculation time (which may happen if complicated rejection criteria not easily coded for the machine are applicable to reduce the hand-computing time). Lengthy and laborious calculations such as those performed by Tarry [2] in proving that no pair of orthogonal latin squares of order 6 can exist or those performed by White, Cole, and Cummings [3] in listing the essentially different systems of Steiner triples of order 15, have not heretofore been feasibly publishable; the authors of such papers have been in the uncomfortable position of having to ask the public to accept on faith the accuracy and, to some extent, the logic of their calculation. It now becomes possible to use the short coded command sequence to present this logic with reasonably modest demands on the publisher and the reader and thus to permit the reader to estimate the probability of accurate calculation and to facilitate confirming recalculation by other workers if this becomes desirable.

[1] It may be noted here that the bibliography is intended to provide easily available expansion of some of the concepts which arise in this paper; there has been no effort to list the earliest or the most important works or to give a complete listing.

2. A convenient class of problems. In the next section a method will be developed for the systematic generation of permutations in a convenient order. The order in which permutations are generated by this method is illustrated in Table I (to be read down the successive columns), in which the first 24 permutations on five marks are exhibited, some more than once. The feature to be noticed is that the permutations are arranged in an order which groups all permutations whose first marks are the same; specifically, the permutations shown in Table I are those whose first mark is 1, all those in any column have the same second mark, and those in the top, middle, or bottom third of a column have the same third mark. The starred permutations are duplicates of permutations which have appeared earlier, and they are to be ignored (their presence in the table will be explained in the next section).

TABLE I

12345	15234	14523	13452
12354	15243	14532	13425
12345*	15234*	14523*	13452*
12534	15423	14352	13245
12543	15432	14325	13254
12534*	15423*	14352*	13245*
12453	15342	14235	13524
12435	15324	14253	13542
12453*	15342*	14235*	13524*
12345**	15234**	14523**	13452**

Several problems will be outlined in which this arrangement of permutations may be exploited to reject whole blocks of consecutively listed permutations and thus to reduce the length of search required. The earlier of these problems will be drawn from classical problems of pure mathematics, and some later ones are of interest in econometrics. The final two paragraphs summarize the section.

By way of notation, the permutations listed will be considered to be the images of the marks 1, 2, 3, $\cdots$, n for some value of n ($n = 5$ in Table I) under a permutation transformation; the image of the mark i will be denoted p_i.

A simple problem which illustrates the technique to be used is the following: How many permutations on the five marks 1, 2, 3, 4, 5 have the property that $p_i - i \equiv p_j - j$ (modulo 5) implies that $i = j$?

In solving this problem, it is possible first to remove most of the potential trials as isomorphs of the ones which will be made based on the permutations listed in Table I. It is clear that any permutation satisfying the requirement of the problem must take exactly one mark into itself, and it is not hard to show that each such permutation is equivalent (by subtraction of a constant number from both the argument mark and the image mark) uniquely to one which takes the mark 1 into itself; *i.e.*, there are five times as many permutations fulfilling the requirements as the number which can be found among the permutations of Table I, each of which transforms the mark 1 into itself.

Now, two subtractions suffice to show not only that the first permutation of the set in Table I is not the set sought but that no permutation in the first column of Table I is in this set. Specifically, the first two marks in the permutation cannot be 12, for each of the differences generated by this pair is $0 = 1 - 1 = 2 - 2$. Since the permutations are arranged so that all the permutations beginning 12 are together in the first column, this whole column may be rejected. Beginning with the second column the first three (not counting the starred permutations) may be rejected, but only on the basis of the fourth mark (reading and computing from the left); and the fourth permutation, 15432, is one of the set sought.

All the rest of the permutations (only two in number) of the second column may be rejected in a block because the beginning 153 yields two differences of 0. Similarly, the first two permutations of the third column may be rejected because of their beginning 145, with two differences of 2, the next two may be rejected because of their beginning 143, with two differences of 0, the fifth is rejected after four marks, with two differences of 4, and the sixth, 14253, is one of the permutations sought.

In the fourth column, the first two permutations are rejected because the beginning 134 gives two differences of 1, the third and fourth are rejected each on the basis of the first four marks, the fifth, 13524, is one of the permutations sought, and the last is rejected on the basis of the first four marks.

Thus, there are seen to be 15 permutations in the set sought. This calculation has been programmed for SWAC for permutations on larger numbers of marks, and rejections occur in larger blocks usually. Hand calculations (which may later be verified by SWAC calculations in the spirit of the last paragraphs of Sec. 1 above) indicate that there are 133 such permutations on seven marks; SWAC has shown that there are 2,025 such permutations on nine marks and punched the complete set of these on cards in about twenty minutes; SWAC has shown that there are 37,851 such permutations on eleven marks, requiring about an hour for the complete count. These calculations on SWAC were studied by J. Dean Swift and the author in an effort to gain some information concerning types of implied solutions and rejections which might lend themselves to SWAC coding. However, the importance here is only as an illustration of a method of systematic search suitable for machine computation.

Other problems to which this method is applicable are those which involve a function of a permutation, which function is at least partially estimated by the first few marks in the permutation. On this basis, rejection of blocks of permutations is undertaken. In the problem above, the functional value may be taken as 0 if there are no repeated differences and as 1 if there are repeated differences; this function may be computed for the first few marks in the permutation, and the permutation is known to be unacceptable if the function becomes positive for any subset of marks. This knowledge leads to the possibility of rejection in blocks.

The method has been used with no substantial change by Frank Meek in a systematic search for a pair of orthogonal latin squares of order 10. A latin square of order n is an ordered set of n permutations on the marks $1, 2, \cdots, n$ with the property that no mark has the same image under any two different permutations of the set. A latin rectangle is a set of no more than n permutations with this property of nonrepetition of image. Two latin rectangles are orthogonal if they are equal in size and if no repetitions occur among the ordered pair of marks obtained by designating one rectangle as the first and the other as the second and by writing pairs of marks which are images of the same argument under permutations in the same positions in the two rectangles; that is, if the image of the mark j under the ith permutation in one square is p_{ij} and in the other square is q_{ij}, then $p_{\alpha\beta} = p_{ij}$ and $q_{\alpha\beta} = q_{ij}$ imply that $\alpha = i$ and $\beta = j$. Euler conjectured that there can be no pair of orthogonal latin squares of order $4k + 2$ for integral k, and Tarry [2] proved that there is no pair of orthogonal latin squares of order 6. Other information concerning orthogonal latin squares is available in Mann [4].

The calculation involves a function (the number of repeated pairs) for latin rectangles and a function (the number of repeated images of the same argument) for arrays of permutations. The problem is then stated as one in which 20 permutations on 10 marks each occur as variables. The second function is used to reject permutations which do not lead to latin rectangles (of increasing size) as parts of the first square sought, and both functions are used to reject permutations which do not lead to latin rectangles as parts of the second square sought and to reject permutations which introduce nonorthogonal elements. The formulation of the problem in these terms is straightforward, and nothing would be gained by going into detail about it here.

This systematic search for orthogonal latin squares has found none after several hours' search. It is clear that no considerable fraction of the number of permutations which may plausibly lead to solutions can be tried in a reasonable length of time (such as the expected lifetime of a researcher or his machine). As the problem is now being attacked, only the most elementary isomorphs are being rejected along with any rejected permutation, and it is obvious that many trials are being made which could be avoided if the machine were able to consider more involved criteria of rejection. The difficulty of coding these criteria springs mainly from difficulties involved in stating them formally, and it is to be expected that these difficulties will be alleviated to some extent as experience is built up. It might be worth noting that the machine quickly builds up a pair of orthogonal rectangles each with five permutations and that these remain unchanged for several minutes; a considerable but small fraction of time is spent in enlarging these to rectangles with six permutations each, and orthogonal pairs with seven permutations each are found comparatively rarely—once every few minutes. This represents a large gain over the number of permutations which would be required if the partial-rejection scheme were not employed, and the rejection of blocks of permutations increases the factor

considerably. However, the problem cannot be considered to be adequately formulated, and methods for more efficient rejection of isomorphs must be devised.

It might be noted that the problem of finding a pair of orthogonal latin squares of order 10 is closely related to the problem of the existence of finite projective planes with 11 points on each line; both problems are of some interest in connection with the design of experiments (again see Mann [4]).

This same method of search was introduced earlier by J. J. Wolf and the author in a systematic search for permutation solutions to a set of equations

$$a_t = X^{-1}R_tZS_tXb_t, \qquad (t = 1, 2, \cdots, m),$$

where for each t, a_t and b_t are known marks, X and Z are unknown permutations, and R_t and S_t are known permutations. The index t and the coefficients a_t, b_t, R_t, and S_t all had ranges sufficient to give considerable redundancy to the equations. The process yielded some information concerning a problem arising in connection with research for the Logistics Branch of the Office of Naval Research. Its only interest here is in connection with an attempt to introduce continuous variables to the problem—an attempt which will be mentioned later.

This same process was used by Erwin Kleinfeld in a study of Veblen-Wedderburn systems of order 16. These systems are sets of 16 elements with operations of addition and multiplication. They form a commutative group with respect to addition, the element 0 satisfies $a \cdot 0 = 0 \cdot a = 0$ with respect to multiplication for all a, multiplication from the right is distributive with respect to addition, unique left and right quotients exist for denominators different from 0, and the equation

$$x \cdot r = x \cdot s + t$$

is solvable uniquely for x for any combination r, s, t for which $r \neq s$.

Veblen-Wedderburn systems exist for all orders which are integral powers of primes; they too are related to finite projective planes. They were introduced in [5].

The procedure used by Kleinfeld involved systematic trial of all possible multiplication tables. He found 1,240 systems. The time for this search was about thirty minutes on SWAC.

Laverne Rickard and J. D. Swift have used the same technique in a search for Steiner triple systems. Much advice in this has been received from Marshall Hall, Jr. A Steiner triple system of order m is a set of $m(m-1)/6$ trigraphs, each digit of which is one of the marks from 1 to m; the trigraphs must be so chosen that every pair of marks occurs in exactly one trigraph of the system.

White, Cole, and Cummings [3] reported that there are 80 essentially different systems of order 15. Straightforward search on SWAC rejecting only the most obvious isomorphs has led to a depressingly copious output.

Swift has reduced the output drastically by means of a second systematic search for isomorphic solutions which would have been uncovered earlier, but this search is a considerable strain on the machine, and it is not complete. This problem seems to offer a fruitful field for study concerning problems of coding and solving problems whose variables are permutations, for here the machine seems unable to get much further than ingenious workers have been able to get by hand. A real attack on the problem of complete enumeration of all essentially different Steiner triple systems of order 19 would be interesting both from the standpoint of machine design and from information which would be furnished by the output.

This same method may be tried on the assignment problem, which is of interest in econometrics. This is a problem which has recently had wide discussion in connection with the optimal assignment of resources (the assignment of personnel to jobs, for example). Its earliest satisfactory solution may have been by Egerváry [6], whose paper was recently discovered and translated by Harold Kuhn. The problem may be stated arithmetically as finding a matrix with maximal trace among all matrices which may be constructed by permuting the columns of a given square matrix. Benjamin Handy, on the suggestion of D. H. Lehmer and with advice from T. S. Motzkin [1], coded this problem for SWAC; he used exhaustive search including rejection of blocks of permutations when the first few elements of the trace led to a hopelessly low contribution. The method worked for a problem whose matrix had 12 rows and 12 columns and was composed of random three-digit numbers. The solution in this case took three hours. Some restrictions which had been imposed concerning the types of problems to which the code should be applicable led to some inefficiencies; however, the simplex method of G. B. Dantzig [7] and various other methods of solution of this problem seem greatly superior to this method of exhaustive search; work on this problem is summarized in this volume by Motzkin [1].

The simplex method seems at first glance to be a continuous method, but more careful examination shows it to use the continuous variables introduced only for purposes of exposition and proof; the actual variation is discrete from one permutation to the next (in the assignment problem), and it offers remarkable efficiency in rejection of permutations which cannot be solutions. A method proposed by Gleyzal and one which has been proposed by Kuhn, based on the work of Egerváry in [6], also seem to give efficient methods of rejection of permutations. All these methods, however, seem to depend upon the possibility of restatement of the problem as a continuous linear problem.

Nonlinear assignment problems (which, in econometric terms, may arise when there is interference or cooperation between activities assigned to neighboring territories—competition for the transportation system, for example) seem not to lend themselves to this type of computation. Except for a few cases which seem to occur significantly less frequently than would be hoped, exhaustive search on the basis used by Handy seems to be the only method

guaranteed to be effective in their solution; the cases which are known to be solvable on a continuous basis are enumerated by Harold Kuhn and A. W. Tucker [8].

A traveling-salesman problem is in some respects similar to the assignment problem. It seems definitely more difficult, however. In its most commonly stated version, it demands the shortest connected path which leads through each of several points (say the capitals of the 48 states of the United States and Washington, D.C.). The problem would be equivalent to the assignment problem if no demand for a connected path were made.

This problem is important in various ways; one application is the assignment of an optimal order to a cycle of operations in which something from each operation (tooling, for example) may be salvaged to be applied to the next operation. (This application was brought to the author's attention by Prof. M. M. Flood, who had received the problem as a real one faced by one of his students in an industrial situation.) Again, the function is one of the type to which the systematic search through permutations is applicable with rejection in blocks. The function to be maximized is the salvage value summed from each pair of successive operations; if this function is diminished by the least possible salvage value following each individual operation, the efficiency of the exhaustive search with rejection of blocks of permutations may be good.

An interesting solution of the traveling-salesman problem for the 48 state capitals and Washington, D.C., has been reported by Dantzig, Fulkerson, and Johnson [9].

This section has been devoted to problems in which the variables are permutations. No detailed solutions have been given except in one problem of great simplicity and slight interest. The remarkable feature is the considerable variety of problems which may be stated in a particular form—problems in which a function whose argument is a permutation and whose value is a real number is to be maximized and in which enough information concerning this function may be implied by the first few marks of the permutation to remove all permutations with these first few marks from consideration in the problem. Problems of this type may be attacked by exhaustive search with rejection of large blocks of permutations under a scheme of systematic generation of permutations outlined above; this scheme will be described in more detail in the next section.

It must be noted, however, that this is not a completely satisfactory scheme for solution of such problems. In a few important cases (such as the assignment problem) more efficient machine methods have been devised. In more problems listed above, the strong desirability or necessity of developing machines and methods to a point where more effective rejection of hopeless candidates or of solutions isomorphic to solutions already found is obvious.

3. Systematic machine generation of permutations. Two methods of systematic generation of permutations will be described. Each is suitable for machine application; one specialization of the second is a means of generating

permutations in an order assumed to be available and noted to be convenient for one class of problem in Sec. 2.

Both schemes depend on a *signature* s. For permutations on n marks this signature will be a number of n digits. Each digit of the signature has a different radix; the radices vary from 1 (a formal digit which can have only 0 as a value) through n. The order of the permutations generated will be the order of increasing signature. It should be noted here that the rule of counting, upon which all arithmetic is based, is easily stated for numbers of variable radix [10]. To pass from one number to the next higher, increase by 1 the value of the least significant digit which has not attained its highest possible value in the first number, set all less significant digits to the value 0, and leave all more significant digits at their former value.

The sense of the notation used here will be recalled. A permutation on n marks will always be considered as a transformation of the n numbers 1, 2, $\cdots$, n into the numbers $p_1, p_2, \cdots, p_n$, respectively. The set of values p_i of *images* is the same as the set of values of *arguments* i.

The first method of systematic generation of permutations is based upon a simple method of assigning permutations to signatures. Let d_r be the digit with radix r in the signature s and let j be a value such that $p_j = r$; then d_r is the number of values of i such that $i > j$ and $p_i < r$. In particular, if the permutation is written as a set of numbers $p_1p_2p_3 \cdots p_n$, then d_r is the number of marks in this set less than r and appearing to the right of r. For example, the permutation 13254 on five marks has signature (written with digits of higher radix in more significant positions to the left of digits of lower radix) 10100.

The proof that each signature determines a permutation uniquely and that no two signatures determine the same permutation is obvious.

This scheme of generation of permutations has been used on SWAC; however, it is more conveniently applied when the signature is generated through some process and the permutation is then to be found than it is in exhaustive searches. One example of such a process is a Monte Carlo process in which the signature is generated in a quasi-random way in order to lead to a quasi-random permutation. The task of converting a signature to a permutation is an easy one. D. H. Lehmer has used this method of attack on a linear assignment problem; it was not so effective here as some other methods which have been mentioned above, but it might well become an effective means of attack on some problems of the nonlinear assignment type if augmented by some improvement technique—in a way which will be suggested in Sec. 5.

This first scheme of systematic generation of permutations is credited to Marshall Hall, Jr.; he is not known to have published it.

The second method of generation of permutations is inductive in nature. It will be described by means of processes which generate a permutation $P(s_2)$ corresponding to a signature s_2 if a permutation $P(s_1)$ corresponding to a signature s_1 is known and if s_2 may be generated from s_1 by increasing by 1

(modulo its radix r_0) some digit d_{r_0} of s_1 which has the property that no less significant digit has a positive value. It is clear that all signatures can be generated by the process used in passing from s_1 to s_2—by a counting process based on increasing the least significant digit by 1 modulo its radix and carrying. Thus, the process to be described will be seen to define inductively a correspondence between signatures and permutations.

This method of generation was possibly first used by Lowell J. Paige and the author on some SWAC calculations.

It will be noted that the permutations listed in Table I are in an order which would be created by one specialization of this process—the starred permutations being those which are created in the course of an operation which generates a carry to a more significant digit of the signature.

This scheme has been used in several different versions; all of them will be described under one general method which depends upon the choice of a permutation with images i_r of the arguments r which assigns a value of the argument i (of the permutation with images p_i) to each value of the radix r. In the most usual version (applied in Sec. 2), $r_i = n - i + 1$, so that the last argument is assigned to radix 1, the next to last to radix 2, etc.

It is convenient to introduce one other symbol q_r to represent the image of the radix r under the product of the permutations which take r into i and i into p: $q_r = p_{i_r}$.

The whole process is now described in the following statements. The signatures s_1 and s_2 are equal in all digits except one, whose radix is r_0; the digit of radix r_0 in s_2 has value 1, higher (modulo r_0) than the digit of radix r_0 in s_1; all digits whose significance in the signature is less than that of the digit of radix r_0 have value 0 in both s_1 and s_2. For $r > r_0$, the mark q_r in the permutation $P(s_2)$ corresponding to signature s_2 is equal to the mark q_r in the permutation $P(s_1)$ corresponding to signature s_1; q_{r_0} in $P(s_2)$ is equal to q_1 in $P(s_1)$; for $r < r_0$, q_r in $P(s_2)$ is equal to q_{r+1} in $P(s_1)$.

The effect of this rule is to assign a set I_r of r values of the argument i to each value of the radix r; I_r is the set of values i_r, corresponding to values of $r' \leqq r$. If the permutation is written as a list of n values p_i in order of increasing i, then the transformation from the permutation corresponding to s_1 to the one corresponding to s_2 (for s_2 and s_1 related as described above) simply permutes the values in the r_0 positions of the permutation corresponding to the set I_{r_0}, and the permutation is one which has a cycle length r_0. In particular, if the digits of the signature are significant in order of increasing radix, and if the basic permutation of r into i is one which reverses the order of the marks (as described above), the permutation carrying $P(s_1)$ into $P(s_2)$ is a cyclic permutation on the last r_0 marks, moving each mark of the set, except the last, one place to the right.

The length of the cycle of these elementary permutations leads to an easy proof, which will not be presented in detail here, that the signature uniquely determines a permutation if the permutation corresponding to any one signature is specified (frequently the permutation represented by marks in natural order is assigned to the signature all of whose digits are 0), and the obvious fact that q_n depends only on the value of the digit of radix n in the signature leads easily to a proof that different signatures correspond to different permutations, the proof being based on induction on the value of n.

With reference to Table I, it is pointed out that if every permutation is to be generated (and the possibility of rejection of blocks is not to be exploited as in Sec. 2), then the number of useless starred permutations is minimized by making the same assignment of $i_r = n - r + 1$ as was suggested above but by assigning significance to the digits of the signature in order of decreasing radix. Also, it should be noted that the permutations corresponding to cyclic permutations on the one element q_1 are not shown in Table I; however, they are usually generated on a computer simply as a means of decreasing coding complexity.

No applications are known in which other permutations i_r or in which other assignments of significance to digits of the signature have been exploited to reject classes of permutations or to avoid needless consideration of isomorphs of permutations already considered in a problem; however, in principle such applications are possible, and coding systematic generation of permutations based on such assignments will not be particularly complex if such applications are found.

4. Embedding of permutations in continuous spaces. Many attempts have been made to study problems involving permutations by continuous-gradient methods. These studies have required that some continuous variables be used to replace the permutations and that the functions of permutations occurring in the original problem be assigned a meaning in the extended problem involving the continuous variables.

The most frequently used approach of this kind involves doubly stochastic matrices. This approach depends upon the representation of permutations by matrices whose elements take on values of 0 and 1 only, and the extension of this set of matrices to a set whose elements are continuously variable between 0 and 1.

Specifically, it is easily verified that the permutation $p_1p_2 \cdots p_n$ (in the notation used above) may be represented by the matrix whose elements are s_{ij}, where

$$s_{ij} = 1 \quad (\text{if } i = p_j),$$

and

$$s_{ij} = 0 \quad (\text{if } i \neq p_j).$$

The operation of transforming the mark i into the mark p_i is accomplished by usual multiplication of a columnar vector by this matrix (the matrix simply permutes the elements of the vector), the mark i being represented by a vector t_j whose elements (components) are all 0 except $t_i = 1$. The product of two permutations (still considered as transformations) is represented by the product of the matrices which represent them.

A doubly stochastic matrix is a matrix with elements s_{ij} which are nonnegative and which satisfy the relations

$$\Sigma_j s_{ij} = \Sigma_i s_{ij} = 1.$$

The elements of doubly stochastic matrices may clearly be made to vary continuously; in fact, the set of doubly stochastic matrices may be considered to be a convex set of points of dimension $(n-1)^2$—exactly $2n-1$ of the $2n$ restrictions above are independent—in a Cartesian coordinate space of n^2 dimensions.

The permutation matrices are clearly doubly stochastic. They are the only vertices of the convex set described above. A simple proof of this statement is due to G. B. Dantzig. First note easily that no vertex of a set of points defined (as the above set is defined) by a set of N linear equalities and by the restriction that all coordinates be nonnegative can contain more than N positive coordinates. For doubly stochastic matrices, $N = 2n-1$, so that there must be one row with at most one nonzero element. In order to satisfy the equality pertaining to this row, this nonzero element must exist and have value 1. There can be no other nonzero element on its column (for the equality pertaining to this column would then be violated), and the matrix of $(n-1)$ rows and columns which results from deleting this row and this column must also be a doubly stochastic matrix. This argument furnishes an inductive proof.

The common operations involving permutations are easily rewritten in terms of doubly stochastic matrices. Two problems will be considered briefly here. The first is the linear assignment problem. If the original matrix is a_{ij} (the problem being one of permuting the columns of this matrix to maximize the trace) the assignment problem may be stated as one of maximizing the function

$$J = \Sigma_{ij} a_{ij} s_{ij}$$

among all permutation matrices s_{ij}. Since the function J is linear in the variables s_{ij}, it will attain its maximum value at a vertex of a convex polytope, such as the set of doubly stochastic matrices, and therefore the set of doubly stochastic matrices may be taken as the set of variables s_{ij} and a solution found which corresponds to a permutation. The common solutions to the assignment problem depend upon this formulation of the problem in their usual proofs, but there is no computational use of continuous variation in these common procedures.

A slightly more subtle approach leads to a formulation of a problem almost equivalent to the problem of solving the equation $a_t = X^{-1}R_tZS_tXb_t$, discussed in Sec. 2. The solution outlined there involved exhaustive search through the permutations X, seeking one which implies nothing which contradicts an assumption that Z is also a permutation. In the continuous approach, doubly

stochastic matrices are substituted for the permutation X, and the stochastic vectors

$$u_t = R_t^{-1}Xa_t,$$
$$v_t = R_t^{-1}Xb_t,$$

are written, where X and R_t^{-1} represent matrices (the matrix R_t representing the permutation R_t, and the matrix X being doubly stochastic), and a_t and b_t are vectors representing the marks a_t and b_t, respectively.

Now, if X is a permutation matrix, the square matrix which is obtained by matrix multiplication of v_t transposed by u_t,

$$W_t = u_t v_t^T,$$

will have one element whose value is 1 and all other elements with value 0; furthermore, if X is a permutation matrix representing a solution to the problem, the nonzero element will fall in one of n places in W_t; this is one of the positions at which the matrix representing the permutation Z implied by this choice of X must have a nonzero element.

Thus, it is clear that a correct permutation X, when represented by its corresponding doubly stochastic matrix, will imply that the nonzero elements of the matrix

$$W = \Sigma_t W_t$$

are at most n in number. It is easy to prove that the sum of the elements of the matrix W is exactly equal to the number of equations presented for solution; it is plausible that any incorrect assumption for the permutation X would tend to randomize the distribution of nonzero contributions to W from the various matrices W_t which are summed to make W; and it follows that the original problem should be solvable by maximizing some function which measures the roughness of the distribution of elements in W. Such a function is

$$J = \Sigma_{ij} w_{ij}^2,$$

where w_{ij} is used to denote the element in the ith row and jth column of the matrix W.

Machine attacks on problems using more or less this approach have been made by A. E. Roberts, Lynn Wilson, Ben Handy, Lowell J. Paige, and the author; in many of these there has been cooperation and advice from Marshall Hall, Jr., A. M. Gleason, H. H. Campaigne, R. A. Leibler, and others. Generally speaking, they have seemed less effective than more straightforward computational attacks.

The problem of minimizing J above furnishes good examples of difficulties which are encountered. The number of variables is high if n is large, so that the calculation of a gradient to use in maximizing the continuous function J is extensive, and in the nonlinear problem outlined it is probable that every permutation matrix furnishes a local maximum (so that the space of doubly

stochastic matrices should possibly be replaced by some other space or the function should be replaced by some other function); these local maxima are exceedingly troublesome in the attacks which have been tried.

These difficulties may be increased in problems of increasing complexity. For example, latin squares may be represented as vertices of a convex region in a Cartesian space of n^3 dimensions, and two orthogonal latin squares may be represented as vertices of a convex region in Cartesian space of n^4 dimensions; H. H. Campaigne has formulated such representations using equalities similar to those used for the representations of permutations as vertices of the region of doubly stochastic matrices. However, the region described by Campaigne has vertices which are not representative of latin squares and orthogonal latin squares, respectively, and hence it is not clear that introduction of these regions facilitates the solution even of linear problems.

5. The problem of economic computing. The continuous embedding approaches sketched in Sec. 4 were invented mainly in an attempt to provide arithmetical operations which are compatible with the machines being exploited. These machines are presently built around a large arithmetic unit, capable of high-speed multiplication of a pair of numbers of full length. Any arithmetic operation on one of the currently available general-purpose machines utilizes this arithmetic unit and its attached memory unit. The problem is one of devising a method of attack on the problem at hand which will utilize the arithmetic unit of the machine most effectively. It is possible to conceive of machinery which is better suited to problems whose variables are permutations and to other problems of discrete variables, but until this machinery is constructed, mathematicians with these problems must look to methods involving multiplication and other arithmetic processes of comparable complexity before the machines available can be used effectively. Thus the problem is one of exploiting complicated arithmetic operations (such as full-sized multiplication) in a way which is productive of results to an extent comparable with the complexity of the operation.

However, some steps toward defining the problems have been made by Stanley Rothman, Dan Teichroew, E. W. Barankin, and others. Some of these ideas will be paraphrased here.

Most computational schemes which are presently employed consist of steps from the following list:

(i) Choose values of the independent variables as tentative candidates for solution;

(ii) Examine these chosen values as to their contribution of knowledge concerning the problem;

(iii) Improve the values chosen by one of possibly several improvement processes available.

As an example, consider the problem of maximizing the function $J = \Sigma_{ij} w_{ij}^2$ introduced in Sec. 4. Here the independent variables are the elements of a doubly stochastic matrix. There are several methods of procedure. First,

the choice of tentative candidates for solution may be restricted to permutation matrices. In this case the examination procedure might consist of a straightforward examination of the function to determine whether the presently considered permutation yields a higher value than any previously considered permutation. In this approach there may be no method available to improve the tentative candidate for solution.

However, there is always a problem of setting up a method for choosing these tentative candidates for solution. In Sec. 2 it was noted that the permutations could be examined systematically with some rejection of large blocks which seem to yield too low values for J, and it is also possible to consider random sampling. On the whole, the systematic method is better for attaining the maximum of J, but random sampling or some other method of generation which permits more radical variations between one tentative solution and the next would be expected to lead to earlier solutions approximating (with high tolerance) the maximum value. D. H. Lehmer has, for instance, attacked the assignment problem on the basis of random sampling and attained fairly good solutions in times much shorter than is required for full sampling with rejection of blocks of permutations.

Another scheme, which has also been tried computationally, is to choose a doubly stochastic matrix with no zero elements as an initial candidate. If it is improved, it is to be improved by variation along the positive gradient of J (in the space of doubly stochastic matrices) until a zero element is produced. If this new matrix is to be improved, it may be improved by using it as the matrix of coefficients of an assignment problem whose permutation solution is the improvement sought (this solution is in some sense the permutation which most closely approximates the doubly stochastic matrix used initially).

This second scheme is attractive to the extent that "the highest mountain has the biggest base," and the initial choice would be one of trying to land on the base of the highest peak. If this is successful, the gradient method would lead to the peak without intervention of the solution of the assignment problem, but computationally the gradient becomes increasingly hard to compute when the inequalities involved (*viz.*, that the variables be nonnegative) become locally restrictive. For this reason, it is plausible that a solution of the linear assignment problem is the best way to proceed, and an assumption that this will lead to the permutation which would be reached through the gradient process is also plausible.

In the first attack outlined here, then, the question of economical calculation is one of how much a complete solution is worth compared with an incomplete solution. The method of choosing tentative candidates for solution depends upon the answer to this question and upon the estimated distribution of values of the function to be maximized.

The second attack leads to more involved questions. First is the question of how to choose the candidate for solution. Second comes the question of whether to reject a choice (presumably because of a low value attained at it by

the function to be maximized) or whether to try to improve this choice through a gradient approach. If the choice is improved, the question of whether to abandon the candidate at that point or to improve it by solving the assignment problem arises. Finally comes the question of when to stop the calculation.

A rational solution to the problem depends upon the following estimates:

(i) A value function which assigns a utility value to a point depending upon the value the function J attains at that point;

(ii) Cost functions, expressed in the same units as the value function above, stating the cost of carrying out each step of the calculation—the initial choice, the evaluation of the function J, the gradient improvement, the assignment-problem improvement;

(iii) Probability functions concerning the probability of attaining various values of J through choice, gradient improvement, improvement through solution of the assignment problem.

It is certain that in most problems this information, especially the information in (iii), will not be available in any completely reliable form. Also in mathematical problems the function demanded in (i) is likely to be missing. However, some comparison of methods is possible under estimates of these functions even for mathematical problems; for problems from econometrics the information demanded above has a more tangible significance, and it is probable that the estimates of the best method of a continuing calculation should be made. The formulation of these estimates is simple formally; it involves nothing more than writing the expected net gain from continuing the calculation at any point. A reasonable rule would be to continue if the value reached at that point exceeds a critical value which depends upon the best tentative solution previously discovered.

The point to be made here is that the experiments sketched in Sec. 4 provide some slight information concerning the problem of economical calculation but that the simplest rational description of the problem is so complex that arguments must be presented on the empirical basis of arguments concerning the art of calculation rather than the science of calculation.

Bibliography

1. Theodore S. Motzkin, *The assignment problem* (in this volume).

2. M. G. Tarry, *Le problème des* 36 *officiers*, C. R. Assoc. Française Avancement Sci. vol. 29 (1900) pp. 170–203.

3. H. S. White, F. N. Cole, and Louise D. Cummings, *Complete classification of the triad systems on fifteen elements*, Nat. Acad. Sci. Mem. vol. 14 (1919) part 2.

4. H. B. Mann, *Analysis and design of experiments*, Dover Publications, Inc., New York, 1949, pp. 76–129.

5. Oswald Veblen and J. H. M. Wedderburn, *Non-Desarguesian and non-Pascalian geometries*, Trans. Amer. Math. Soc. vol. 8 (1907) pp. 379–388.

6. J. Egerváry, *Matrixok kombinatorius tulajdonságairól*, Mat. Fiz. Lapok vol. 38 (1931) pp. 16–27 (translated by H. W. Kuhn, *On combinatorial properties of matrices*, Logistics Papers (George Washington University) issue 11 (1955), paper 4, pp. 1–11).

7. George B. Dantzig, *Maximization of a linear function of variables subject to linear inequalities* in *Proceedings of a conference on activity analysis*, Tjalling C. Koopmans, ed., John Wiley & Sons, Inc., New York, 1951, pp. 339–347. Later developments by the author have been published in various places not readily accessible to all readers, including reports of The RAND Corporation, Santa Monica, Calif.

8. H. W. Kuhn and A. W. Tucker, *Non-linear programming*, in *Proceedings of the second Berkeley symposium on mathematical statistics and probability*, University of California Press, Berkeley, Calif., 1951, pp. 481–492.

9. G. Dantzig, R. Fulkerson, and S. Johnson, *Solution of a large scale traveling salesman problem*, The RAND Corp. Rep. P-510 (Apr. 12, 1954).

10. Staff of Engineering Research Associates, Inc., *High-speed computing devices*, McGraw-Hill Book Company, Inc., New York, 1950, chap. 6.

National Bureau of Standards, Institute for Numerical Analysis,
Los Angeles, Calif.