Discrete Mathematics 12 (1975) 341–358.

RANDOM MATROIDS*

Donald E. KNUTH
Stanford University, Stanford, Calif. 94305, USA

Received 29 August 1974

A simple combinatorial construction capable of producing an arbitrary matroid is introduced, and some of its properties are investigated. The structure of a matroid is defined one rank at a time, and when random choices are made the result might be called a random matroid. Some experimental statistics about such matroids are tabulated. If we specify the subsets of rank $<k$, the construction defines a rank function having the richest possible matroid structure on the remaining subsets, in the sense that no new relationships are introduced except those implied by the given subsets of rank $<k$. An appendix to this paper presents several computer programs for dealing with matroids over small sets.

0. Introduction

Mathematical systems called matroids were introduced and named by Whitney in 1935 (see [8]), and the associated theory became extensively developed during the ensuing decades, most notably by Tutte in the late 1950's [6,7]. Edmonds' subsequent discovery that most of the known efficient solutions to combinatorial problems can be associated with a matroid structure (cf. [2,4]) has led to considerable interest in matroids during recent years.

Matroids are abstract systems, but of course when we deal with them we usually have a more or less concrete model in mind. Much of the theory has been developed from a geometric or algebraic point of view, using the fact that a special type of matroid arises in the study of vector spaces spanned by the rows of a matrix. Other aspects of the theory have been derived using intuition from graph theory, since certain

* This research was supported in part by National Science Foundation Grant GJ-36473X and by the Office of Naval Research Contract NR-044-402.

matroids arise naturally in the study of graphs.

The purpose of this paper is to introduce another approach to the study of matroids, a viewpoint that is essentially combinatorial and constructive. The author believes that this approach may shed some new light on the theory, and that many interesting research problems are suggested by this work; but it must be confessed that the present paper contains more definitions than theorems.

The approach we shall discuss rests on a simple technique that constructs all matroids when given a (typically small) number of virtually unconstrained "enlargements" whose consequences fully define the structure. If these "enlargements" are selected at random, we don't obtain truly random matroids, since different matroids will in general be obtained with differing probabilities, but the probability distribution that arises does appear to have interesting properties.

Section 1 of this paper defines matroids and establishes the notational conventions to be used. The main construction appears in Section 2, and an example is given in Section 3. Sections 4, 5 and 6 prove that the algorithm of Section 2 is correct, complete, and well-defined. Section 7 looks at the construction from a more general point of view, and observes that it can be used to define the "free completion" of a matroid above rank k in a meaningful way. Some experimental results are reported in Section 8, and some open problems suggested by this research are listed in Section 9. The Appendix presents detailed programs which implement the construction with reasonable efficiency.

1. Definitions and notation

Matroids may be defined in many equivalent ways (e.g., by their independent sets, their circuits, their bases, their bonds, their rank function, or their closed sets), and for our purposes the definition via closed sets is most convenient. We shall therefore say that a *matroid* $\mathcal{M} = (E, \mathcal{F})$ is a (finite) set E together with a family $\mathcal{F}$ of subsets of E, satisfying the following three axioms.

(i) $E \in \mathcal{F}$;

(ii) if $A, B \in \mathcal{F}$, then $A \cap B \in \mathcal{F}$;

(iii) if $A \in \mathcal{F}$ and $a, b \in E \setminus A$, then b is a member of all sets of $\mathcal{F}$ containing $A \cup a$ if and only if a is a member of all sets of $\mathcal{F}$ containing $A \cup b$.

(Here "$A \cup a$" is shorthand for "$A \cup \{a\}$".) The elements of $\mathcal{F}$ are called the *closed* subsets of E.

If A is any subset of E, we define

$$\bar{A} = \bigcap\{B \in \mathcal{F}: B \supseteq A\}$$

as the intersection of all closed sets containing A. Axiom (i) guarantees that this intersection is nonempty, and Axiom (ii) implies that $\bar{A}$ is itself closed; consequently $\bar{A}$ is the (unique) smallest closed set containing A, and

$$\bar{\bar{A}} = \bar{A} .$$

Axiom (iii) may now be rewritten more simply:

(iii) if $A \in \mathcal{F}$ and $a, b \in E \setminus A$, then ($b \in \overline{A \cup a}$ iff $a \in \overline{A \cup b}$).

2. A general construction

Our goal is to understand the implications of the above axioms, and one way to approach them is to try to construct all such families $\mathcal{F}$ over a given set E. We can never hope to look at them all unless E is a rather small set, since there are $2^{2^{n-O(\log n)}}$ possibilities when E has n elements [3], but it may be helpful to consider algorithms which are in principle capable of constructing all matroids.

For this purpose, let us try to construct $\mathcal{F}$ by starting with small closed sets and then defining the larger ones. If A is any closed set, the gist of Axioms (i), (ii), (iii) is that the smallest closed sets properly containing A must partition the elements of $E \setminus A$. In other words, there must exist disjoint sets $B_1, ..., B_k$ such that $B_1 \cup ... \cup B_k = E \setminus A$ and such that $a \in B_j$ iff $\overline{A \cup a} = A \cup B_j$ for $1 \leqslant j \leqslant k$. For if we define a relation on the elements of $E \setminus A$ by saying

$$a \sim b \quad \text{iff } b \in \overline{A \cup a} ,$$

then Axioms (i) and (ii) imply that

$$a \sim b \quad \text{iff } \overline{A \cup b} \subseteq \overline{A \cup a}$$

and Axiom (iii) tells us that $a \sim b$ iff $b \sim a$, hence $\sim$ is an equivalence relation. The problem is to find a family of sets which defines such partitions for all closed sets A.

The following algorithm, which attempts to find the "finest" partitions consistent with these conditions, now suggests itself.

Step 1. [Initialize.] Set r to 0, and let $\mathcal{F}_0$ be $\{\emptyset\}$, the family of sets consisting of the empty set alone.

Step 2. [Generate covers.] Let $\mathcal{F}_{r+1}$ be the set of all "covers" of the sets in $\mathcal{F}_r$, i.e.,

$$\mathcal{F}_{r+1} = \{A \cup a \colon A \in \mathcal{F}_r \quad \text{and} \quad a \in E \setminus A\} .$$

Step 3. [Enlarge.] Add additional sets to $\mathcal{F}_{r+1}$, if desired, where each new set properly contains some element of $\mathcal{F}_r$. (This step is indeterminate. By making different choices we will in general produce different matroids.)

Step 4. [Superpose.] If $\mathcal{F}_{r+1}$ contains any two sets A, B whose intersection $A \cap B$ is not contained in C for any $C \in \mathcal{F}_r$, replace A and B in $\mathcal{F}_{r+1}$ by the single set $A \cup B$. Repeat this operation until $A \cap B \subseteq C$ for some $C \in \mathcal{F}_r$ whenever A and B are distinct members of $\mathcal{F}_{r+1}$. (We shall prove later that the replacements can be made in any order without affecting the final result of this step.)

Step 5. [Test for completion.] If $E \in \mathcal{F}_{r+1}$, terminate the construction. Otherwise increase r by 1 and return to Step 2.

This construction terminates because every member A of $\mathcal{F}_{r+1}$ properly contains some member of $\mathcal{F}_r$, hence A contains at least $r+1$ elements. We shall prove that the family

$$\mathcal{F} = \mathcal{F}_0 \cup \mathcal{F}_1 \cup \ldots \cup \mathcal{F}_r \cup \mathcal{F}_{r+1}$$

obtained at the conclusion of the construction defines the closed sets of a matroid, no matter what choices are made in Step 3.

3. A "random" example

Before going into any details of the proof, let us look at a concrete example in order to fix the ideas. For convenience in notation we shall use the decimal digits $\{0, 1, ..., 9\}$ as elements of the set E. Subsets of E will be written without braces or commas, so that "156" stands for the 3-element subset consisting of 1, 5 and 6; and "{156,23}" stands for a family of two subsets of E.

The construction of Section 2 begins with $\mathcal{F}_0 = \{\ \ \}$, then Step 2 tells us that $\mathcal{F}_1$ is the family $\{0,1,2,3,4,5,6,7,8,9\}$ of all singleton subsets. Let us assume that the first execution of Step 3 leaves $\mathcal{F}_1$ unchanged; consequently Step 4 will also leave $\mathcal{F}_1$ unchanged. (It turns out that any changes made to $\mathcal{F}_1$ at this point are equivalent to "shrinking" E into a smaller set. By leaving $\mathcal{F}_1$ unchanged we will be constructing a so-called "combinatorial geometry" [1], namely a matroid in which all one-point sets are closed.) Step 5 sets r to 1 and we return to Step 2.

Now Step 2 causes $\mathcal{F}_2$ to be {01, 02, ..., 89}, the family of all pairs. Let us add further sets to $\mathcal{F}_2$ in a "random" way, using the digits of

$$\pi = 3.14159\ 26535\ 89793\ 23846$$

to govern our choices. From "3,1,4" we shall add the set 134, from "1,5,9" we shall add 159, and similarly we shall add 256 and 358; since "9,7,9" involves only two digits, let us include the following digit 3, and then we shall also include 238. Thus, we have added six triples 134, 159, 256, 358, 379, 238 to $\mathcal{F}_2$; and we may as well stop here, since six is a perfect number.

Step 4 is interesting now, since it causes many of the sets in $\mathcal{F}_2$ to be merged together. Since $\mathcal{F}_1$ contains all the one-element sets, any distinct sets A, B in $\mathcal{F}_2$ which have two or more common elements are replaced by $A \cup B$. In particular, sets like 13, which are contained in the added triples, simply disappear since $13 \cap 134 = 13$ and $13 \cup 134 = 134$. Furthermore, we replace 358 and 238 by their union 2358, which in turn combines with 256 to give 23568. We are ultimately left with 30 subsets, namely

$$\mathcal{F}_2 = \{01, 02, 03, 04, 05, 06, 07, 08, 09, 12, 134, 159, 16, 17, 18, 23568, 24, 27, 29, 379, 45, 46, 47, 48, 49, 57, 67, 69, 78, 89\}\ .$$

Then we set r to 2 and return to Step 2.

Let us leave $\mathcal{F}_3$ untouched when we next reach Step 3; perhaps we don't know π to enough decimals, or we simply want to see what happens. It turns out that a great deal happens in Step 4; e.g. $235689 \cap 2379 = 239$ is not contained in any member of $\mathcal{F}_2$, so we replace 235689 and 2379 by 2356789, etc. The following 22 subsets are eventually obtained:

$$\mathcal{F}_3 = \{012, 0134, 0159, 016, 017, 018, 023568, 024, 027, 029, 0379, 045, 046, 047, 048, 049, 057, 067, 069, 078, 089, 123456789\}\ .$$

Since the last subset 123456789 has only one proper cover, namely E, we are bound to have

$$\mathcal{F}_4 = \{0123456789\}$$

regardless of what transpires in the next Step 3, so the construction will terminate with $r = 3$.

It is not hard to check that $\mathcal{F} = \mathcal{F}_0 \cup \mathcal{F}_1 \cup \mathcal{F}_2 \cup \mathcal{F}_3 \cup \mathcal{F}_4$ defines the closed sets of a matroid. (The *bases*, or maximal independent sets, of this matroid are

$$\{0123, 0124, 0125, 0126, 0127, 0128, 0129, 0135, 0136, 0137, 0138, 0139, 0145, 0146, 0147, 0148, 0149, 0156, 0157, 0158, 0167, 0168, 0169, 0178, 0179, 0189, 0234, 0237, 0239, 0245, 0246, 0247, 0248, 0249, 0257, 0259, 0267, 0269, 0278, 0279, 0289, 0345, 0346, 0347, 0348, 0349, 0357, 0359, 0367, 0369, 0378, 0389, 0456, 0457, 0458, 0459, 0467, 0468, 0469, 0478, 0479, 0489, 0567, 0569, 0578, 0579, 0589, 0678, 0679, 0689, 0789\};$$

and the *circuits*, or minimal dependent sets, are

$$\{134, 159, 235, 236, 238, 256, 258, 268, 356, 358, 368, 379, 568, 1237, 1239, 1245, 1246, 1247, 1248, 1249, 1257, 1267, 1269, 1278, 1279, 1289, 1357, 1367, 1369, 1378, 1389, 1456, 1457, 1458, 1467, 1468, 1469, 1478, 1479, 1489, 1567, 1578, 1678, 1679, 1689, 1789, 2347, 2349, 2457, 2459, 2467, 2469, 2478, 2479, 2489, 2579, 2679, 2789, 3457, 3459, 3467, 3469, 3478, 3489, 4567, 4569, 4578, 4579, 4589, 4678, 4679, 4689, 4789, 5679, 5789, 6789\}.$$

Note that our construction needed only six 3-element sets to specify the entire matroid, so this approach leads to economy in specification.)

4. Proof of correctness

Let us now prove that the family $\mathcal{F} = \mathcal{F}_0 \cup \mathcal{F}_1 \cup \ldots \cup \mathcal{F}_r \cup \mathcal{F}_{r+1}$ defined by our construction yields a matroid. We shall prove that $(E, \mathcal{F})$ is a matroid whenever $\mathcal{F} = \mathcal{F}_0 \cup \ldots \cup \mathcal{F}_{r+1}$ is a family of sets with the following properties:

(a) $E \in \mathcal{F}_{r+1}$; and $E \notin \mathcal{F}_j$ for $0 \leq j \leq r$;

(b) $\mathcal{F}_0$ consists of a single set;

(c) if $A, B \in \mathcal{F}_j$, $A \neq B$ and $j > 0$, then $A \cap B \subseteq C$ for some $C \in \mathcal{F}_{j-1}$;

(d) if $A \in \mathcal{F}_j$, $a \in E \setminus A$ and $j \leq r$, then $A \cup a \subseteq C$ for some $C \in \mathcal{F}_{j+1}$;

(e) if $A, B \in \mathcal{F}_j$ and $A \subseteq B$, then $A = B$.

Properties (a), (b), (c) are immediate from the construction; and so is property (d), since Steps 3 and 4 do not remove any sets from a family unless a larger set is substituted. Furthermore, by Steps 2 and 3 each element of $\mathcal{F}_j$ for $j > 0$ properly contains some element of $\mathcal{F}_{j-1}$. Property (e) follows by induction on j, since $A \subseteq B$ implies that $A \subseteq C$ for some $C \in \mathcal{F}_{j-1}$ according to (c), but A properly contains some $D \in \mathcal{F}_{j-1}$.

Axiom (i) holds trivially because of (a).

Given $A \subseteq E$, let q be minimal such that $A \subseteq B$ for some $B \in \mathcal{F}_q$. This B is unique; for if $A \subseteq B_1$ and $A \subseteq B_2$, then $q > 0$ by (b) and $A \subseteq B_1 \cap B_2 \subseteq C$ for some $C \in \mathcal{F}_{q-1}$ by (c). We shall say that q is the *rank* of A, and we shall define $\bar{A} = B$.

Note that $\bar{A} = A$ implies that $A \in \mathcal{F}$. Conversely if $A \in \mathcal{F}$ we have $\bar{A} = A$; for if $A \in \mathcal{F}_j$ and the rank of A is $q < j$, then properties (a) and (d) imply that A is properly contained in some $B \in \mathcal{F}_j$, contradicting (e). Hence $\bar{\bar{A}} = \bar{A}$.

Let $a \notin A$ and assume that $\overline{A \cup a} \neq \bar{A}$, where A has rank q. Then $A \cup a \not\subseteq C$ for any $C \in \mathcal{F}_j$, $0 \leq j \leq q$, by the uniqueness of $B = \bar{A}$. But we do have $\bar{A} \cup a \subseteq C$ for some $C \in \mathcal{F}_{q+1}$, by property (d); consequently

$$\overline{A \cup a} = \overline{\bar{A} \cup a} \supseteq \bar{A} .$$

We have proved that $\bar{A} \subseteq \overline{A \cup a}$ for all A and a, and by the finiteness of our universe it follows that

$$A \subseteq B \quad \text{implies} \quad \bar{A} \subseteq \bar{B}$$

for all $A, B \subseteq E$.

If $A, B \in \mathcal{F}$, we now have

$$\overline{A \cap B} \subseteq \bar{A} = A , \qquad \overline{A \cap B} \subseteq \bar{B} = B ,$$

hence

$$\overline{A \cap B} \subseteq A \cap B \subseteq \overline{A \cap B} ;$$

in other words, $A \cap B = \overline{A \cap B}$, and Axiom (ii) is established.

Let A^* be the intersection of all closed sets containing a given set A.

Clearly $A^* \subseteq \bar{A}$, since $\bar{A}$ is such a closed set, and we have proved that A^* is closed, hence $A^* = \bar{A}$; our definition of $\bar{A}$ in this section agrees with the definition in Section 1. Now Axiom (iii) follows immediately.

5. Proof of completeness

We can also show that *every* matroid is essentially obtainable by the construction in Section 2. Let $(E, \mathcal{F}')$ be a (finite) matroid, and for $A \in \mathcal{F}'$ let rank(A) be the minimal r such that A is the closure of some r-element subset of E. It is well-known that rank$(A \cup a) =$ rank$(A)+1$ whenever $a \notin \bar{A}$. Let $\mathcal{F}'_r$ be all the closed sets of rank r, for $r = 0, 1, \ldots$.

If the empty set $\emptyset$ is closed, we can prove that the algorithm in Section 2 is capable of constructing the matroid $(E, \mathcal{F}')$, with $\mathcal{F}_r = \mathcal{F}'_r$ for all r. In fact, this is true for $r = 0$, and it holds for $r+1$ if we add the elements of $\mathcal{F}'_{r+1}$ to $\mathcal{F}_{r+1}$ in Step 3. The reason is simply that each cover generated in Step 2 is contained in some unique element of $\mathcal{F}'_{r+1}$, hence Step 4 simply removes everything but $\mathcal{F}'_{r+1}$. (Of course, it is generally possible to obtain the same result with far fewer additions in Step 3; a study of the minimum number of necessary enlargements should prove to be interesting.)

If the empty set is not closed, then $\bar{\emptyset}$ is contained as "excess baggage" in every closed set, and $(E, \mathcal{F})$ is isomorphic to the matroid $(E \setminus \bar{\emptyset}$, $\{A \setminus \bar{\emptyset}\colon A \in \mathcal{F}\})$. Our construction would be capable of producing such degenerate matroids if, for example, we would change Step 1 to "Set r to -1, set $\mathcal{F}_{-1}$ empty, set $\mathcal{F}_0$ to $\{\emptyset\}$, and go to Step 3"; but no new cases of interest would be produced.

6. Commutativity

Step 4 of the algorithm in Section 2 is the keystone of our construction, and we should prove that it does not depend on the order in which reductions are made. In general, let $\mathcal{P}$ be any *order ideal* on the subsets of E (i.e., if $B \in \mathcal{P}$ and $A \subseteq B$, then $A \in \mathcal{P}$), and let $\mathcal{A}$ be any family of subsets of E; we shall consider the following operation which generalizes Step 4.

"If $\mathcal{A}$ contains any two sets A, B whose intersection $A \cap B$ is not contained in $\mathcal{P}$, replace A and B in $\mathcal{A}$ by the single set $A \cup B$. Repeat this operation until $A \cap B \in \mathcal{P}$ whenever A and B are distinct members of $\mathcal{A}$."

Let $\mathcal{A} = (A_1, A_2, ..., A_m)$ be a sequence of subsets of E, and consider the following operation $\langle i,j \rangle$:

If $A_i \neq A_j$ and $A_i \cap A_j \notin \mathcal{P}$, set A_i and A_j both equal to $A_i \cup A_j$. This operation makes two copies of the merged set $A_i \cup A_j$, so that each member of $\mathcal{A}$ retains its original position in the sequence, otherwise it is equivalent to the general operation described above. Suppose we apply such operations repeatedly, obtaining a sequence of sequences $\mathcal{A} = \mathcal{A}^0$, $\mathcal{A}^1, ..., \mathcal{A}^k$, where $\mathcal{A}^k = (A_1^k, A_2^k, ..., A_m^k)$ is *fixed* in the sense that $A_i^k \neq A_j^k$ implies $A_i^k \cap A_j^k \in \mathcal{P}$ for all i, j. Note that $A_j^k \supseteq A_j$ for all j. If we apply $\langle i,j \rangle$ operations in another order to the same initial sequence, obtaining $\mathcal{A} = \mathcal{B}^0, \mathcal{B}^1, \mathcal{B}^2, ...,$ it is easy to prove by induction on t that $\mathcal{B}^t = (B_1^t, B_2^t, ..., B_m^t)$, where $B_j^t \subseteq A_j^k$ for all j. For if $B_i^{t-1} \neq B_j^{t-1}$ and $B_i^{t-1} \cap B_j^{t-1} \notin \mathcal{P}$, we have $A_i^k \cap A_j^k \supseteq B_i^{t-1} \cap B_j^{t-1}$; hence $A_i^k \cap A_j^k \notin \mathcal{P}$, and $A_i^k = A_j^k \supseteq B_i^{t-1} \cup B_j^{t-1}$.

If $\mathcal{B}^t$ and $\mathcal{A}^k$ are both fixed, we have $B_j^t \subseteq A_j^k \subseteq B_j^t$ by symmetry. The final result is therefore independent of the order in which $\langle i,j \rangle$ operations are applied.

7. Free completion

It is well known that any matroid can be "truncated to rank k", in the sense that we eliminate all closed sets of rank $\geqslant k$ except E itself. This truncation operation is equivalent to adding E to $\mathcal{F}_{r+1}$ when $r = k-1$ in our construction.

Conversely, our construction allows us to add the richest possible structure above rank k to a given matroid, in the sense that we can find the greatest number of closed sets for ranks $> k$ in any matroid having prescribed closed sets for ranks $\leqslant k$. If we make no additions in Step 3, let us say that the family $\mathcal{F}_{r+1}$ obtained at the end of Step 4 is the *free completion* of $\mathcal{F}_r$. If $(E, \mathcal{F})$ is a given matroid of rank $> k$, its *free completion above rank k* is the matroid over E whose closed sets are the closed sets $\mathcal{F}_0 \cup ... \cup \mathcal{F}_k$ of $\mathcal{F}$ having rank $\leqslant k$, together with $\mathcal{F}'_{k+1} \cup \mathcal{F}'_{k+2} \cup ... \cup \mathcal{F}'_{r+1}$, where $\mathcal{F}'_k = \mathcal{F}_k$ and $\mathcal{F}'_{q+1}$ is the free completion of $\mathcal{F}'_q$ for $k \leqslant q \leqslant r$, and $\mathcal{F}'_{r+1} = \{E\}$. In a sense every matroid whose closed sets for ranks $\leqslant k$ are in $\mathcal{F}_0 \cup ... \cup \mathcal{F}_k$ is a "homomorphic image" of this free completion, where the homomorphism corresponds to enlargements made in Step 3.

It should be interesting to explore properties of free completion. Note that the construction of $\mathcal{F}_{r+1}$ depends only on $\mathcal{F}_r$, so that the matroid is being built up layer by layer. The same construction can be applied in general to any "clutter", i.e., to the set of maximal elements of any order ideal, in place of $\mathcal{F}_r$ in Step 2; Steps 2 and 4 then define the free completion of a clutter, whether or not the clutter can be represented as the sets of rank $\leqslant k$ in some matroid. This may lead to a generalization of matroids. However, in every case tried by the author where the order ideal is not that of a matroid, the free completion reduced trivially to $\{E\}$; perhaps such collapsing will always occur in non-matroid situations.

8. Some experiments

In an attempt to study the behavior of the algorithm when random "coarsening" is applied to the structure in Step 3, several experiments were attempted with small sets E.

The experiments were conducted as follows. Step 4 was performed immediately after Step 2, in order to shorten the list of subsets and to be sure that all consequences of the present structure were taken into account. Then a member A of $\mathcal{F}_{r+1}$ was selected at random, each being equally likely; and when A had been chosen, an element a of $E \setminus A$ was selected at random, each being equally likely. The set A was replaced by $A \cup a$ in $\mathcal{F}_{r+1}$, and Step 4 was performed again. This enlargement process was repeated a specified number of times, p_r, depending on the current rank r; p_0 was always 0, so that the first effects would appear in $\mathcal{F}_2$.

For example, our experiment based on π in Section 3 corresponds roughly to $p_1 = 6, p_2 = p_3 = 0$, on a 10-element set E. Thirty random experiments were conducted with these parameters, and in each case the resulting matroid had rank 4. Table 1 shows the number of elements in $\mathcal{F}_2$ and $\mathcal{F}_3$ after reduction, and the number of bases and circuits in the first ten resulting matroids. (The last of these has, by chance, the same statistics as the "random" matroid in Section 3.) The computation time for these ten experiments, using ALGOL W on a 360/67, was 15.6 seconds.

Table 2 shows the average values obtained for several settings of the parameters. In nearly every case the final rank was reduced by one each time an enlargement was made.

It should be interesting to develop theoretical results that account for this observed behavior.

Table 1

$\lVert \mathcal{F}_2 \rVert$	$\lVert \mathcal{F}_3 \rVert$	Bases	Circuits
23	15	48	51
32	24	76	89
23	15	48	51
32	24	76	89
23	15	48	51
31	23	74	82
27	19	62	61
23	10	63	36
23	15	48	51
30	22	71	76

Table 2
Observed mean values.

n $(p_1, p_2, \ldots)$	Trials	Bases	Circuits	$\lVert \mathcal{F}_2 \rVert$	$\lVert \mathcal{F}_3 \rVert$	$\lVert \mathcal{F}_4 \rVert$	$\lVert \mathcal{F}_5 \rVert$	$\lVert \mathcal{F}_6 \rVert$	$\lVert \mathcal{F}_7 \rVert$
10 (6, 0, 0)	30	62.1	60.8	25.9	17.8	1.0			
10 (5, 1, 0)	10	89.8	86.1	33.0	28.5	1.0			
10 (5, 2, 0)	8 [a]	109.9	159.8	32.8	1.0				
10 (5, 2, 0)	2 [b]	140.5	91.5	34.5	39.0	1.0			
10 (6, 1, 0)	20	102.8	141.5	28.9	1.0				
10 (4, 2, 0)	10	114.8	105.9	36.4	37.8	1.0			
10 (3, 3, 0)	8 [b]	114.6	112.3	38.8	41.9	1.0			
10 (3, 3, 0)	2 [c]	94.5	55.5	38.5	64.5	36.0	1.0		
10 (0, 6, 0)	5 [b]	157.8	159.0	45.0	74.0	1.0			
10 (0, 6, 0)	5 [c]	128.0	92.8	45.0	100.2	68.8	1.0		
10 (0, 1, 1, 1)	10	38.3	10.6	43.0	101.8	136.4	96.7	29.1	1.0
13 (6, 0, 0, 0, 0, 0)	3	141.7	44.0	63.7	149.7	179.7	107.0	26.3	1.0
13 (6, 2, 0, 0)	9	432.8	327.2	64.3	137.3	100.2	1.0		

[a] Averages for experiments when final rank was 3.
[b] Averages for experiments when final rank was 4.
[c] Averages for experiments when final rank was 5.

9. Open problems

A few research problems have been stated above, and they will be repeated here for emphasis.

(1) If an order ideal in the lattice of subsets of E does not correspond to the sets of rank $\leqslant r$ of any matroid, is its free completion always trivial, or do we obtain a generalization of matroid behavior?

(2) What can be said about the fewest number of enlargements needed

to completely specify a given matroid of rank r on n elements? (Computer experiments indicate that $n-r$ suitably chosen enlargements will work in nearly all the small cases, but the construction in [3] shows that considerably more enlargements are needed in general.)

(3) Can the stochastic properties of this construction be analyzed carefully enough to narrow the known bounds on the asymptotic number g_n of matroids on n elements? (It is known [3] that $\log_2 \log_2 g_n$ lies between $n - \frac{3}{2}\log_2 n$ and $n - \log_2 n$ plus terms of lower order.)

Appendix. Computer programs

The computer programs used in this study are presented here for the possible benefit of others who wish to experiment with matroids, and also for the possible interest of language designers, because there is still a relative scarcity of published algorithms dealing with manipulation of sets. The programming is in ALGOL W [9], a language chosen by the author primarily because of the excellent debugging facilities available [5]; it will not be difficult to transliterate the programs below into other languages.

The running time of the construction in the text is governed largely by the speed of Step 4, which would be extremely inefficient if programmed in a brute force manner based on the definitions. The implementation below reduces this cost substantially by combining Steps 2, 3 and 4, using a routine that maintains a list of subsets satisfying the condition at the end of Step 4 at all times, so that the basic operation is one of inserting into such a list. The inner loop of this insertion process is kept short by using a table that tells whether or not any given subset has rank $\leqslant r$.

The time and space requirements of these algorithms for manipulating random matroids grow exponentially with $n = \|E\|$, as one might expect. The program below assumes that $n \leqslant 13$, but with suitable modifications it would be possible to adapt the program so that cases as large as $n = 20$ become feasible on contemporary medium-to-large scale computers.

Sets are represented in the program by the so-called **bits** variables of ALGOL W, since these variables are subject to Boolean operations. It is also convenient at times to consider **bits** variables as binary numbers, so that they can be used as subscripts or in arithmetic operations. If v is of type **bits**, ALGOL W uses the notation *number*(v) for the corresponding integer; if u is of type **integer**, the notation *bitstring*(u) stands for the corresponding bits. Neither *number* nor *bitstring* requires any computation time on a binary computer.

The program deals with linked lists of sets, kept in two arrays S and L; $S[k]$ is the set stored at position k, and $L[k]$ is the position number of the next set in the list. The lists are linked circularly, in most cases; if h is the "head" of a list, then $S[h]$ is irrelevant, the first item of the list is in position $L[h]$, and the last item is in position k, where $L[k] = h$. An empty circular list therefore has $L[h] = h$.

The program is designed to do more than the construction in the text; it prints out the independent sets for each rank as well as the circuits of the matroid. For this purpose it is convenient to have a table which indicates the cardinality of each subset; hence the *rank* array serves double duty. If v is the bitstring representation of a set A, the table entry $rank[number(v)]$ will be set to $100 + \|A\|$ at the beginning of the computation and until the true rank of A is computed; then again $100+\|A\|$ will be used at the end of the program when the circuits are being tabulated.

With these introductory remarks, it is hoped that the comments on the program below will be sufficiently explanatory. Note that "long labels" are occasionally used as comments, to help indicate the program structure; the text of procedures has been deferred until after the main program, as a further attempt to make the program readable in one pass.

```
begin comment Exploration of "random" matroids;
   integer n; comment number of elements in universe, must be ≤ 13;
   integer mask; comment 2^n − 1, represents the set E;
   integer i, j, k; comment temporary indices;
   integer r; comment the current rank;
   integer h; comment head of the circular list of closed sets for rank r;
   integer nh; comment head of the circular list being formed for the
                       closed sets for rank r+1;
   integer avail; comment beginning of the list of available space;
   bits x; comment a set used to communicate with the insert routine;
   bits array S[0 : : 4999]; integer array L[0 : : 4999]; comment list
                                                                  memory;
   integer array rank[0 : : 8191]; comment 100 + cardinality, or assigned
                                           rank of subset;
   procedure ... (see procedure declarations below);
   read(n); comment first data entry is the matroid size;
   mask := round(2↑n − 1);
```

```
   set initial contents of rank table:
      k := 1; rank[0] := 100;
      while k ≤ mask do
      begin for i := 0 until k − 1 do rank[k+i] := rank[i] + 1;
         k := k+k;
      end;
   initialize list memory to available:
      for i := 0 until 4998 do L[i] := i+1;
      L[4999] := −1; avail := 2;
      L[1] := 0; S[1] := #0; h := 0; comment list containing the empty set;
   rank[0] := 0; r := 0;
   while rank[mask] > r do
   begin comment pass from rank r to r+1;
      create empty list:
         nh := avail; avail := L[nh]; L[nh] := nh;
      generate; comment see procedure below;
      enlarge; comment see procedure below;
      return list h to available storage:
         j := h; while L[j] ≠ h do j := L[j];
         L[j] := avail; avail := h;
      r := r+1; h := nh;
      printstatistics; comment see procedure below;
      assign rank to sets and print independent ones:
         write ("Independent sets for rank", r, ":");
         j := L[h];
         while j ≠ h do
         begin mark(number(S[j]));
            comment see procedure below;
            j := L[j];
         end;
   end;
   printcircuits; comment see procedure below;
end.
```

The procedures mentioned in this program are implemented as follows.

```
procedure generate;
begin comment insert the minimal closed sets for rank r+1 into a cir-
              cular list headed by nh (see Step 2 in the text);
   bits t, v, y; integer j; comment temporary storage;
```

```
    j := L[h]; comment prepare to go through h list;
    while j ≠ h do
    begin y := S[j]; comment closed set of rank r;
       t := bitstring(mask − number(y)); comment set complement;
       find all sets in nh list which already contain y and
          remove excess elements from t:
          k := L[nh];
          while k ≠ nh do
          begin if (S[k] ∧ y) = y then t := (t ∧ ~S[k]);
             k := L[k];
          end;
       insert y ∪ a for each a ∈ t:
          while t ≠ #0 do
          begin x := y ∨ (t ∧ ~bitstring(number(t) − 1));
             insert; comment insert x into nh, possibly enlarging x, see
                             below;
             t := t ∧ ~x;
          end;
       j := L[j];
    end;
end;

procedure insert;
begin comment insert set x into list nh, but augmenting x if necessary
              (and deleting existing entries of the list) so that no two
              entries have an intersection of rank > r;
   integer j, k;
   j := nh;
store: S[nh] := x;
loop: k := j;
continue: j := L[k];
   if rank[number(S[j] ∧ x)] ≤ r then go to loop;
   if j ≠ nh then
      begin if x = (x ∨ S[j]) then
         begin remove from list and continue:
            L[k] := L[j]; L[j] := avail; avail := j;
            go to continue;
         end else
         begin augment x and go around again:
            x := x ∨ S[j]; nh := j; go to store;
```

```
        end;
     end;
insert new item:
   j := avail; avail := L[j];
   L[j] := L[nh]; L[nh] := j; S[j] := x;
end;

procedure enlarge;
begin comment insert sets read from data cards until encountering an
              empty set;
   readon(x);
   while x ≠ #0 do
      begin insert; readon(x) end;
end;

procedure printstatistics;
begin integer j;
   write ("Closed sets for rank", r, ":");
   j := L[h];
   while j ≠ h do
      begin writeon(S[j]); j := L[j] end;
end;

procedure mark (integer value m);
begin comment m is a binary-coded subset. This procedure sets rank[m']
              := r for all subsets m' of m whose rank is not already
              ≤ r, and outputs m' if it is independent (i.e., if its rank
              equals its cardinality);
   integer t, v;
   if rank[m] > r then
      begin if rank[m] = 100 + r then writeon(bitstring(m));
         rank[m] := r;
         t := m;
         while t ≠ 0 do
         begin v := number(bitstring(t) ∧ bitstring(t − 1));
            mark (m − t + v);
            t := v;
         end;
      end;
end;
```

```
procedure printcircuits;
begin comment This procedure prints all minimal dependent sets and
               assigns rank ⩾ 100 to all dependent sets;
  write ("The circuits are:");
  k := 1;
  while k ⩽ mask do
  begin for i := 0 until k − 1 do
    if rank[k+i] = rank[i] then
      begin writeon(bitstring(k+i));
        unmark (k+i, rank[i]+101);
      end;
    k := k+k;
  end;
end;

procedure unmark (integer value m, card);
begin integer t, v;
if rank[m] < 100 then
  begin rank[m] := card; comment card is 100 plus the cardinality of m;
    t := mask − m;
    while t ≠ 0 do
    begin v := number(bitstring(t) ∧ bitstring(t − 1));
      unmark (m+t−v, card + 1);
    end;
  end;
end;
```

Further efficiency was gained in practice by sorting the closed sets so that they appear on list h in order of decreasing cardinality when the *generate* procedure is called. Thus, the statement "*sort*;" was inserted just before "*printstatistics*;". A simple radix list sort was used as follows:

```
procedure sort;
begin integer array hd[100 : : 113];
  for i := 100 until 100+n do hd[i] := −1;
  j := L[h]; L[h] := h;
  while j ≠ h do
  begin i := rank[number(S[j])];
    k := L[j]; L[j] := hd[i]; hd[i] := j; j := k;
  end;
```

```
  for i := 100 until 100+n do
  begin j := hd[i];
    if j ≥ 0 then
      begin while L[j] > 0 do j := L[j];
        L[j] := L[h]; L[h] := hd[i];
      end;
  end;
end;
```

The effect of this procedure was to reduce the number of tests on *rank* in the main *insert* loop from about 7500 to about 1700, when $n = 10$ and $p_1 = 6, p_2 = p_3 = 0$. For larger n the gain in efficiency was even more significant, since the lists are never very long when $n = 10$ and $p_1 = 6$.

Acknowledgments

I wish to thank Stein Krogdahl and Daniel Kleitman for their helpful comments on the first draft of this paper.

References

[1] H.H. Crapo and G.-C. Rota, On the Foundations of Combinatorial Theory: Combinatorial Geometries (preliminary edition) (MIT Press, Cambridge, Mass., 1970).

[2] J. Edmonds, Submodular functions, matroids, and certain polyhedra, Combinatorial Structures and their Applications, Proc. Calgary International Conference 1969 (Gordon and Breach, New York, 1970) 69–87.

[3] D.E. Knuth, The asymptotic number of geometries, J. Combin. Theory 16(B) (1974) 398–400.

[4] E.L. Lawler, Combinatorial Optimization: Networks and Matroids (Holt, Rinehart and Winston, New York, in preparation).

[5] E. Satterthwaite, Debugging tools for high level languages, Software–Practice and Experience 2 (1972) 197–217.

[6] W.T. Tutte, A homotopy theorem for matroids, I and II, Trans. Am. Math. Soc. 88 (1958) 144–174.

[7] W.T. Tutte, Matroids and graphs, Trans. Am. Math. Soc. 90 (1959) 527–552.

[8] H. Whitney, On the abstract properties of linear dependence, Am. J. Math. 57 (1935) 509–533.

[9] N. Wirth and C.A.R. Hoare, A contribution to the development of ALGOL, Comm. ACM 9 (June 1966) 413–434.