THE DEPENDENCE GRAPH FOR BASES IN MATROIDS

by

Stein Krogdahl

STAN-CS-75-495
MAY 1975

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY

Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
US Department of Commerce
Springfield, VA. 22151

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER STAN-CS-75-495	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) THE DEPENDENCE GRAPH FOR BASES IN MATROIDS		5. TYPE OF REPORT & PERIOD COVERED technical, May 1975
		6. PERFORMING ORG. REPORT NUMBER STAN-CS-75-495
7. AUTHOR(s) S. Krogdahl		8. CONTRACT OR GRANT NUMBER(s) NR 044-402 N00014-75-A-0362-0001
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer Science Department Stanford University Stanford, Ca. 94305		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS ARPA/IPT, Attn: Stephen D. Crocker 1400 Wilson Blvd., Arlington, Va. 22209		12. REPORT DATE May 1975
		13. NUMBER OF PAGES 32
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) ONR Representative; Philip Surra Durand Aeronautics Bldg., Rm. 165 Stanford University Stanford, Ca. 94305		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Releasable without limitations on dissemination

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

bases in matroids, bipartite graphs, matroid connectivity, algorithms, matroid dependence.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

see reverse.

block 20

This paper discusses a certain graph, called the "dependence graph" ("the DPG"), that can be defined naturally for a given independent set in a matroid. We are mainly concerned with the DPG of bases, and we study what the DPG of a base tells about the matroid. We show that there is a nice connection between the DPG and duality, and between the DPG and connectivity for matroids. This last fact leads to an algorithm for determining the connected components of a matroid and also to one for computing a circuit containing two given distinct elements in the same such component. A simple algorithm using depth-first search is given for solving this last problem for graphic matroids.

The Dependence Graph for Bases in Matroids

Stein Krogdahl

Abstract

This paper discusses a certain graph, called the "dependence graph" ("the DPG"), that can be defined naturally for a given independent set in a matroid. We are mainly concerned with the DPG of bases, and we study what the DPG of a base tells about the matroid. We show that there is a nice connection between the DPG and duality, and between the DPG and connectivity for matroids. This last fact leads to an algorithm for determining the connected components of a matroid and also to one for computing a circuit containing two given distinct elements in the same such component. A simple algorithm using depth-first search is given for solving this last problem for graphic matroids.

Keywords: bases in matroids, bipartite graphs, matroid connectivity algorithms, matroid dependence.

This research was supported by The Norwegian Research Council for Science and the Humanities. Printing costs were supported in part by National Science Foundation grant GJ 36473X and by the Office of Naval Research contract NR 044-402. Reproduction in whole or in part is permitted for any purpose of the United States Government.

1. Introduct:

This paper discusses a certain bipartite graph that can naturally be defined for an independent set in a matroid. This graph is here called the "dependence graph" of the independent set, but it occurs in [6] under the name "simple border graph".

The dependence graph of an independent set exposes to a certain extent how this set is located within its "environment", the set it spans. This graph is important in Lawler's matroid intersection algorithm [7], and its properties make up the fundamentals for the combinatorial proof given for the algorithm in [6]. Lawler has also conjectured that the "matroid parity problem" is solvable in polynomial time, and dependence-graphs may well turn out to be important also here. (The "matroid parity problem" is to find the greatest set of pairs constituting an independent set in a matroid where the elements are partitioned into pairs.)

An interesting property of the dependence graph of any base of a matroid is that it very nicely reflects the structure of connectivity in the matroid. This leads to a simple algorithm for finding the connected components of a matroid, that is described in Section 6.

As the title of this paper indicates, we shall mainly be concerned with the dependence graph of bases. This is, however, not a very strong restriction, as any result obtained for this special case directly applies to any independent set considered as a base of its span.

The dependence graph of a base is closely related to Whitney's concept of a "strict fundamental set of circuits" in [10], and Lemma 10 in Section 7 is more or less a translation of one of his theorems for such circuit-sets into the language of this paper. The proof, however, is different.

2. Basic Concepts

In the following we will consider the basic properties of matroids as known. However, to settle the terminology we give a brief survey of some definitions and theorems from this theory below. A nice introduction to matroid theory is given in Whitney's original paper [9].

Throughout the paper we will take the freedom of writing e instead of $\{e\}$ when this is obvious from the context. The cardinality of a set A will be denoted $|A|$.

A matroid is defined on a finite set E by a family of subsets of E, called the "independent" subsets of E, that obey the following axioms:

(i) $\emptyset$ is independent;

(ii) any subset of an independent set is independent;

(iii) for any set $A \subseteq E$, all maximal independent subsets of A have the same cardinality.

The common cardinality mentioned in (iii) is called the "rank" of A, written " $r(A)$ ".

A set which is not independent is said to be "dependent". The minimal dependent sets are called "circuits". No circuit is properly contained in another, and if C_1 and C_2 are circuits such that $e \in C_1 \cap C_2$ and $e_1 \in C_1 - C_2$, then there is a circuit in $C_1 \cup C_2 - e$ containing e_1.

For all $A \subseteq E$ the maximal set S such that $A \subseteq S \subseteq E$ and $r(A) = r(S)$ is well defined, and this set is called the "span" of A, written " $sp(A)$ ". The elements in $sp(A)-A$ are exactly those $e \in E-A$ such that there is a circuit in $A \cup e$ containing e. If $I \subseteq E$ is independent and $e \in sp(I)-I$, then $I \cup e$ contains a unique circuit, which we shall denote " $C(e,I)$ ".

A maximal independent set is called a "base". All bases have the same cardinality $r(E)$, and if B_1 and B_2 are different bases, then for each $e_1 \in B_1-B_2$ there is an $e_2 \in B_2-B_1$ such that $B_1 \cup e_2-e_1$ is also a base.

A matroid is obviously determined if its set of bases is given. It turns out that the set of base-complements (in E) for a matroid M form the base-set of another matroid, which is called the "dual" matroid of M.

We also need some elementary graph-theory and we use the following terminology. A graph is a finite set of "nodes", together with a set of "arcs", each being an unordered pair of distinct nodes, called its "endnodes". A subset P of the nodes of a graph G is said to be a "partitioning set of G" if each arc of G has one endnode in P and one outside. If a graph has a partitioning set P then it is said to be "bipartite", and then the set of nodes outside P also forms a partitioning set.

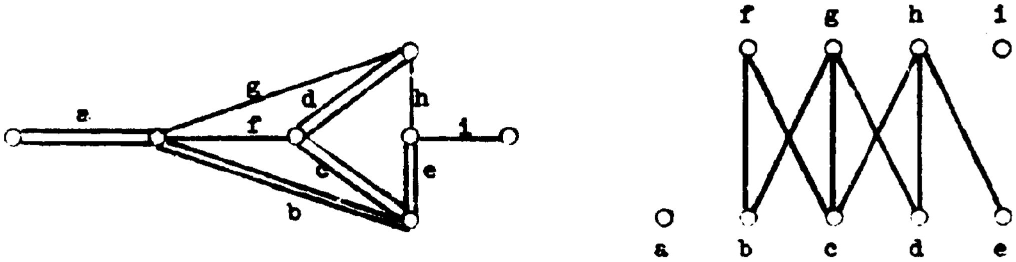
A "matching" in a graph is a subset L of its arcs such that each node occurs as an endnode of at most one arc of L. A set N of nodes is said to be "covered" by a matching L if each node in N occurs as an endnode of an arc in L. In Section 4 we will use the following well-known theorem due to P. Hall (cf. [2]):

Theorem. If G is a bipartite graph and P is a partitioning set of G , then there is a matching in G covering P if and only if for each $P' \subseteq P$ the set Q' of nodes reachable by following an arc from P' , is such that $|P'| \leq |Q'|$. ⊐

3. Definition of the Dependence Graph.

Let M be a matroid over a set E , and assume that $I \subseteq E$ is independent. The "dependence-graph of I " (written "the DPG of I ", or only " DPG(I) ") is defined as the following bipartite graph G . The nodes of G are (in one-to-one correspondence with) the elements of E , and I is a partitioning set of G . There is an arc in G between $e_1 \in I$ and $e_2 \in E-I$ if and only if $e_2 \in sp(I)$ and $e_1 \in C(e_2,I)$.

When we draw a dependence graph we will usually have the nodes from I at the bottom. As an example let M be the graphic matroid defined on the arc set of the graph below, and let I = {a,b,c,d,e} , which is marked by double lines.



A graphic matroid and the DPG of {a,b,c,d,e} .

4. The DPG of a base, and what it tells about other bases

The following rather obvious lemma shows that the DPG of a base could have been defined fully within the framework of bases.

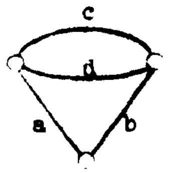
Lemma 1. If B is a base of a matroid and $e_1 \in B$ and $e_2 \in E-B$, then there is an arc between e_1 and e_2 in DPG(B) if and only if $B \cup e_2 - e_1$ is also a base.

Proof. If there is an arc between e_1 and e_2 in DPG(B), then we will destroy the only circuit in $B \cup e_2$ by removing e_1. Thus $B \cup e_2 - e_1$ is independent, and therefore also a base.

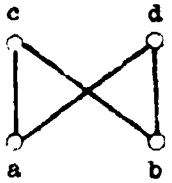
Conversely, if there is no arc between e_1 and e_2 then $B \cup e_2 - e_1$ will contain the circuit $C(e_2, B)$ and can therefore not be a base. □

Lemma 1 tells us that the DPG of a base B describes, and is described by, the set of bases that differ from B in exactly one element. That this in general is not enough to determine all bases (and thus the full structure) of the matroid is demonstrated by the following example.

We define two matroids on the set $\{a,b,c,d\}$. M_1 is the graphic matroid of the graph pictured below, and M_2 is the matroid where all sets of cardinality less than or equal to 2 are independent. In both M_1 and M_2, the set $\{a,b\}$ is a base and the DPG of $\{a,b\}$ is the one pictured below in both matroids. However, M_1 and M_2 are not equal as $\{c,d\}$ is a base in M_2 but not in M_1.



The graph defining M_1 .



The DPG of $\{a,b\}$ in M_1 and M_2 .

We may obtain a feeling for how "little" the DPG of a base says about the matroid by observing that the number of different DPGs on n nodes grows not faster than $O(2^{n^2})$. However, D. Knuth has shown in [5] that the number of essentially different matroids over a set with n elements is as big as $2^{2^{n-\frac{3}{2}\log_2 n+O(\log\log n)}}$, which grows considerably faster.

In spite of these facts it turns out that the DPG of a base points out a much larger class of sets that must be bases than those covered by Lemma 1, and exactly how large this class is, is described in Lemma 6. However, the proof of the fact that any set not covered by this lemma is "unreliable as a base" is for convenience postponed to Section 7.

We will also (and first) describe those sets that under no circumstances can be bases, by giving a condition that all bases must obey. This condition will immediately be shown to be as strong as possible.

In the following lemmas, matchings in the DPG of a base will be important. If L is a matching in the DPG of a base B , then we will denote the set of its endnodes outside B as " $OUT(L)$ " and the set of those inside B as " $IN(L)$ ". The set $B \cup OUT(B) - IN(B)$ will be denoted $L(B)$. Obviously $|OUT(L)| = |IN(L)|$ and $|L(B)| = |B|$.

We start out with the condition which all bases must obey, and we first prove a slightly more general result than we need.

Lemma 2. Let B be a base and I an independent set of a matroid. Then there is a matching L in $DPG(B)$ such that $OUT(L) = I-B$ and $IN(L) \subseteq B-I$.

Proof. By P. Hall's theorem it is enough to show that, for any $I' \subseteq I-B$, the set J' of nodes in $B-I$ that are reachable from I' by arcs in $DPG(B)$ will satisfy $|I'| \leq |J'|$.

For each I' we have

$$J' = (B-I) \cap \left(\bigcup_{e \in I'} C(e,B) \right) .$$

This implies that $I' \cup (B \cap I) \subseteq sp(J' \cup (B \cap I))$. Since both $I' \cup (B \cap I)$ and $J' \cup (B \cap I)$ are independent we must have $|I' \cup (B \cap I)| \leq |J' \cup (B \cap I)|$. This implies $|I'| \leq |J'|$, which is what we wanted. □

As all bases have the same cardinality, we immediately obtain this lemma (which was probably first proved by Magnanti in [8]):

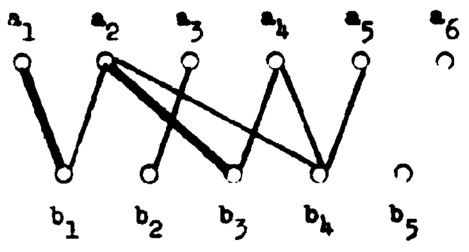
Lemma 3. Let B and B' be two bases of a matroid. Then there is a matching L in $DPG(B)$ such that $OUT(L) = B'-B$ and $IN(L) = B-B'$.

To show that this condition is the strongest possible, we prove the lemma below. Note that the following lemma also proves that for any bipartite graph G with a designated partitioning set B , there is a well-defined base-richest matroid over the nodes of G , such that B is a base of this matroid and G is the DPG of B .

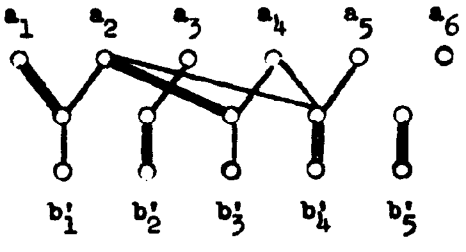
Lemma 4. Let G be a bipartite graph and let B be one of its partitioning sets. Then the set $\mathcal{B} = \{L(B) \mid L \text{ is a matching in } G\}$ forms the set of bases of a matroid over the node-set of G . (Here we allow L to be empty, so that $B \in \mathcal{B}$.)

Proof. We could here prove directly that the axioms for bases are true. However, the constructions needed have been done once and for all in a more general setting by Edmonds and Fulkerson in [1]. Here they prove that if H is a graph and E is a subset of its nodes, then the subsets of E that can be covered by a matching in H will form the independent sets of a matroid on E .

The appropriate H for our case is obtained from G by adding an arc with endnode b' from each node $b \in B$. Further we let E be all nodes in this graph except the old b-nodes, letting b' be their new representatives. That the Edmonds/Fulkerson construction on this graph yields the matroid in the lemma is now fairly easy to see, and the details are left to the reader. The following illustration may clarify the construction:



G



H

The matching indicated in G shows that $\{a_1, a_2, b_2, b_4, b_5\}$ is in $\mathcal{B}$. The matching indicated in H shows how the corresponding set $\{a_1, a_2, b'_2, b'_4, b'_5\}$ occurs as a base of the matroid described above. □

We now turn to the problem of characterising those sets that the DPG of a base B points out as bases. Let B' be a set such that $|B'| = |B|$. By Lemma 3 we know that if B' has any chance of being a base, there must be a matching L in $DPG(B)$ such that $B' = L(B)$. It turns out that the only time we can really conclude that B' is a base is when there is only one matching L such that $L(B) = B'$.

A matching L which is such that no other matching has exactly the same IN- and OUT- set will be called "clean". However, since we want to use this condition in different forms, we will first define cleanness in a rather obscure way, and then prove that this is equivalent to the above condition, and also to a third form which will turn out to be perhaps the most useful one.

Let L be a matching in the DPG of a base B. We will say that L is "clean" if every submatching L' of L is such that there exists a node in $IN(L')$ whose only arc to nodes in $OUT(L')$ is the one in L'. A simple cycle in $DPG(B)$ which is such that exactly each second arc is in L is called an "L-alternating cycle". (In [6] this is called a "main cycle induced by L".) We prove the following lemma.

<u>Lemma 5</u>. Let L be a matching in the DPG of some base B of a matroid. Then the following three statements are equivalent:

(a) L is clean.

(b) There is no matching L' in $DPG(B)$ such that $IN(L) = IN(L')$, $OUT(L) = OUT(L')$ and $L \neq L'$.

(c) The DPG of B has no L-alternating cycle.

Proof.

(a) $\Rightarrow$ (b). By the definition of cleanness we can see that if IN(L) and OUT(L) are given, then the process of choosing arc by arc a matching that covers exactly these sets can only be done in one way, since there is always an arc in the remaining set that has to be chosen.

(b) $\Rightarrow$ (c). Suppose there were an L-alternating cycle in DPG(B), with arc-set C. Then $(L \cup C) - (L \cap C)$ would be another matching obeying all the requirements of L' in (b).

(c) $\Rightarrow$ (a). Suppose L is not clean. Then there is a submatching L' of L such that each node in IN(L') has arcs to at least two nodes in OUT(L'). Start at any node in OUT(L') and follow the arc in L' from here to the "corresponding" node in IN(L'). Then take any other arc to another node in OUT(L'), and repeat the process again. By the finiteness of L' we must eventually come back to a node in OUT(L') which we have seen before, and then an L-alternating cycle is formed. □

We are now ready to prove the result claimed above.

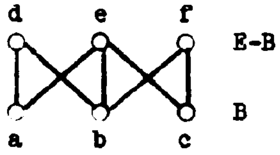
Lemma 6. If L is a clean matching in the DPG of a base B of some matroid, then L(B) is a base.

Proof. By the original definition of cleanness we can pick an arc $a \in L$ with endnodes $e_1 \in IN(L)$ and $e_2 \in OUT(L)$ such that there is no arc from e_1 to any node in $OUT(L)-e_2$. This means that e_1 does not occur in the circuit $C(e,I)$ for any node $e \in OUT(L)-e_2$.

By Lemma 1 the set $B' = B \cup e_2 - e_1$ is also a base, and since the circuits mentioned above are not touched by this interchange, the arcs from nodes in $OUT(L)-e_2$ are the same in DPG(B') as in DPG(B) . Thus the matching L-a reoccurs in DPG(B') and it is clean here as it was in DPG(B) . Thus by an inductive argument on the size of L , we can conclude that L(B) is a base. □

Not surprisingly, it is also true that if L is a clean matching, then L will reoccur "upside down" in DPG(L(B)) , but we leave the proof of this to the interested reader. (A proof occurs in [6].) This is not generally true if L is not clean, even if L(B) is a base.

The remaining question now is whether Lemma 6 covers all sets that must be bases. Unfortunately the set $\mathcal{B} = \{L(B) \mid L$ is a clean matching in DPG(B)$\}$ does not generally form the base-set of a matroid. This can for example be seen by studying the following bipartite graph:



Here $B_1 = \{a,d,e\}$ and $B_2 = \{c,e,f\}$ are both in $\mathcal{B}$. However, to conform to the base-axioms, there would have to be an element in $B_2-B_1 = \{c,f\}$ that could replace a in B_1 . But neither $\{c,d,e\}$ nor $\{f,d,e\}$ are in $\mathcal{B}$, showing that $\mathcal{B}$ is not the base set of a matroid.

Thus the "base-poorest" matroid is not nicely well-defined as is the base-richest. The proof that Lemma 6 still is the best possible (in a little weaker sense) is postponed to Section 7, where the necessary tools are developed.

5. Duality and the DPG of a Base

The relation between the DPG of a base and duality is the following simple one.

Lemma 7. If B is a base for a matroid M on E , then the DPG of the base E-B in the dual matroid M* is the same as the DPG of B in M , considered "upside down".

Proof. This is a direct consequence of Lemma 1 and the fact that the bases of M* are exactly the complements of the bases of M .

The relation between the DPG and duality will be investigated further in Section 7.

6. Connectivity

In this section we will prove a simple relation between connectivity in matroids, and the usual graph-connectivity in the DPG of some base in the matroid. This relation will naturally point out a simple algorithm for computing the connected components of a matroid.

By using Whitney's original definition of connectivity in matroids (in terms of rank-relations) the relation we shall prove would be rather evident. We will, however, give a presentation in terms of circuits, which will also yield an algorithm for finding a common circuit of two elements if they lie in the same connected component.

We will say that two elements of a matroid are "connected" if there exists a circuit containing them both. This relation is obviously symmetric, and we will define it to be reflexive. That is, an element that occurs in no circuit is connected to itself and nothing else. That it also is transitive follows from this lemma:

Lemma 8. In a matraid let C_1 and C_2 be circuits such that $C_1 \cap C_2 \neq \emptyset$, and let e_1 and e_2 be elements such that $e_1 \in C_1 - C_2$ and $e_2 \in C_2 - C_1$. Then there is a circuit in $C_1 \cup C_2$ containing both e_1 and e_2.

Proof. The proof is by induction on $|C_1 \cup C_2|$: the lemma holds vacuously when $|C_1 \cup C_2| < 3$.

For the inductive step choose an element $e \in C_1 \cap C_2$, and a circuit C_3 in $C_1 \cup C_2 - e$ containing e_1. If $e_2 \in C_3$ then we are done. If not, we can use the induction hypothesis on C_3 and C_2, except when $C_1 - C_2 \subseteq C_3$. In this case we pick a circuit C_4 in $C_1 \cup C_2 - e$ containing e_2. Then $C_4 \cap C_3 \supseteq C_4 - C_2 \neq \emptyset$, and as C_3 and C_4 both avoid e we have $|C_3 \cup C_4| < |C_1 \cup C_2|$. Thus we can use the induction hypothesis again. □

Thus the relation of being connected is an equivalence-relation, and the equivalence classes with respect to this relation are usually called the "connected components" of the matroid.

Note that the connected components of a graphic matroid are identical to the 2-connected components of the arcs in the underlying graph. However, the connected components of a graph as used below are the equivalence-classes of nodes with respect to being connected by a single path. To avoid confusion we will call this last type of connectivity in graphs "G-connectivity", while connectivity in matroids is called "M-connectivity". The corresponding connected components will be called "G-components" and "M-components" respectively.

Lemma 9. If B is a base of a matroid, then the G-components of DPG(B) are identical to the M-components of the matroid.

<u>Proof</u>. Suppose e and e', $e \neq e'$, are in the same G-component of $DFG(B)$. Then we can obviously find a sequence $e_0, e_1, \ldots, e_n$ from $E-B$ such that $e \in C(e_0,B)$, $e' \in C(e_n,B)$ and $C(e_{i-1},B) \cap C(e_i,B) \neq \emptyset$ for $i = 1,2,\ldots,n$. Thus e and e' are in the same M-component.

To obtain the lemma we must also prove that there are no circuits in the matroid containing elements from two or more G-components. Therefore suppose there exist such circuits, and choose one of them, C_1, such that $|C_1-B|$ is minimal. Obviously $|C_1-B| \geq 1$. Suppose C_1 has elements from the G-components of $DFG(B)$, K_1 and K_2, such that C_1 has an element e_1 in K_1-B. We know that $C_1 \neq C(e_1,B)$ since all elements of $C(e_1,B)$ are in K_1. Choose $e_2 \in C_1 \cap K_2$, and choose a circuit C_2 in $C_1 \cup C(e_1,B)-e_1$ containing e_2. Obviously then $|C_2-B| < |C_1-B|$. However, C_2 must also have an element in $C(e_1,B)-C_1 \subseteq K_1$, a contradiction. □

Note that through Lemma 7 this lemma gives a nice demonstration of the well known fact that the M-components of a matroid and its dual are the same.

Now suppose a matroid is given over E such that there is a polynomial time algorithm for deciding whether a given set is independent or not, taking $|E|$ as the "size" of the problem. Then, given any dependent set, we can find one of its circuits by scanning through its elements once, pushing out those that do not make the remaining set independent. This gives a simple polynomial time algorithm for finding $C(e,B)$ if e and B are given, and for finding the DFG of B.

We can now give an algorithm for computing the M-components of a matroid based on Lemma 9. The algorithm needs a data-structure that keeps the elements of E divided into disjoint subsets, and it needs an operation " MERGE(a,b) ", $a,b \in E$, that will unite the subsets containing a and b if they are in different subsets, and otherwise do nothing. There are very efficient data structures and algorithms available for this problem (see [4], page 354).

When we start the algorithm, we have each element of E in its own subset. We look at each element of E once in any order, and as we proceed we build up a base for the matroid in a set B (which initially is empty), by adding to B each element e we meet that makes $B \cup e$ independent. On the other hand, when we meet an element e such that $B \cup e$ is dependent we compute $C(e,B)$, and perform MERGE(e,e') for each $e' \in C(e,B)-e$.

That the partition of E yields the M-components of the matroid, when all elements have been treated, is a direct consequence of Lemma 9.

We will now show that the proof of Lemma 8 gives a polynomial time algorithm for finding a circuit containing two given distinct elements e_1 and e_2 whenever they are in the same M-component. To decide if they are, we first use the above algorithm to find the M-components, and if e_1 and e_2 are in the same component, it is also easy to construct a "chain" of circuits $C_0,C_1,\ldots,C_n$ such that $e_1 \in C_0$ and $e_2 \in C_n$, and $C_{i-1} \cap C_i \neq \emptyset$, $i = 1,2,\ldots,n$. If possible, we also "shortcut" this chain until further shortcuts are impossible.

The last step is now to "perform" Lemma 8 repeatedly on neighboring C_i's to "shrink" them into one circuit without breaking the chain, until only one circuit is left.

This obviously solves the problem in polynomial time, if Lemma 8 is "performable" in polynomial time. To see that it is, we first describe how to find a circuit in a set $A \subseteq E$ containing one given element $e \epsilon A$. First build a maximal independent set I for $A-e$ in the same way as we built the base B in the last algorithm. If $I \cup e$ is dependent, we compute $C(e,I)$ and use this; if $I \cup e$ is independent, there is no circuit of the type we want.

With this construction in mind it is easy to see that the proof of Lemma 8 directly yields a polynomial-time algorithm for finding the circuit that the lemma itself asserts the existence. The details are left to the reader.

Simplifications of these algorithms in the case of graphic matroids are discussed in Section 9.

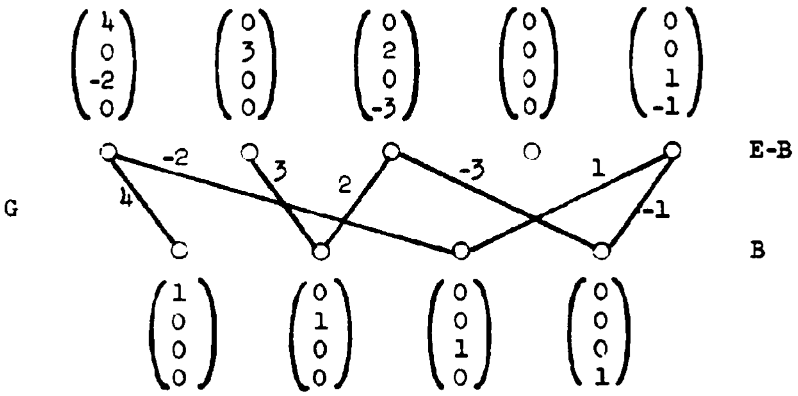
7. Matroids Induced from Vector-spaces

Suppose V is a vector-space over some field F , and let E be a finite subset of V . If we define a subset of E to be independent if and only if it is linearly independent in V , then basic theorems from linear algebra tell us that these define a matroid on E . Let us call this the "induced matroid" of E .

If we fix some independent r-tuple $B = (b_1,b_2,\ldots,b_r)$ from V that spans E then each element $e \epsilon E$ can naturally be represented as an element e' in F^r . We will consider the elements of F^r as columns, and these columns also form a vector-space in their own right. In this vector-space the elements $E' = \{e' \mid e \epsilon E\}$ will induce the same matroid as E did in V , quite independent of the choice of the set B .

If G is a bipartite graph, we define an "F-labeling of G" to be an assignment of non-zero values from F to the arcs of G .

Let E be the node-set of G , let $B \subseteq E$ be a partitioning set of G and suppose an F-labeling of G is given. We can then in a natural way associate elements of F^r , $r = |B|$, to each node in G such that B is a base of the matroid induced by these elements, and such that the DPG of B in this matroid is G . We first arrange the nodes of B in some order $b_1, b_2, \ldots, b_r$, and to node b_i we associate the column b_i' with a " 1 " in position " i ", and with zeroes elsewhere. To each node $a \in E\text{-}B$ we associate the column $a' = \sum_{i=1}^{r} q_{ai} b_i'$, where q_{ai} is the label of the arc between a and b_i if it exists, else it is zero. As an example, look at this picture (F are the real numbers).



We leave it to the reader to verify that the DPG of B in the induced matroid is indeed equal to G . We further notice that any set of r independent vectors from any vector-space over F used as b_i' , $i = 1, \ldots, r$, (computing the " a' " 's by the same formula) would induce the same matroid over E .

Conversely, let E be a finite set of vectors from a vector-space over F , and let B be a base for the induced matroid M . Since then every element of $E\text{-}B$ is a unique linear expression in elements from B , we see that the elements of E define a natural F-labeling of the DPG of B , such that the matroid M will reoccur by using the construction above on this labeling of the DPG of B .

Thus, for any F-labeled bipartite graph G and for any partitioning set B of G there is a natural F-induced matroid on the nodes of G such that B is a base and $G = \mathrm{DPG}(B)$. It is also possible to construct any matroid induced by elements of a vector-space over F in this way (but there are generally many labelings inducing the same matroid).

As an F-labeling completely determines a matroid when G and B are given, we should be able to characterize the rest of this matroid directly from the labeling. The following lemma should then come as no surprise.

Lemma 10. Let M be the matroid defined over the nodes of the bipartite graph G by a given labeling from F and a given partitioning set B , and let A be a set of nodes such that $|A| = |B|$. Then A is a base for M if and only if $\det(Q) \neq 0$, where Q is the quadratic matrix in F formed by letting $i \in A-B$ index its columns and $j \in B-A$ its rows and defining the element q_{ij} of Q as the label of the edge between i and j if it exists, and otherwise zero.

Proof. Let us associate with each node of G the "natural" element from F^r , $r = |B|$, and let us do the ordering of B that was necessary for this such that the elements of $B-A$ comes first. Let us further order the columns associated with elements of A such that those associated with elements in $A-B$ come first and such that the rest follow in the same order as $B \cap A$ was ordered above. The rows of A will then constitute this matrix:

$$\begin{array}{c|c|c|}
 & A\text{-}B & A\cap B \\ \hline
B\text{-}A & Q & 0 \\ \hline
B\cap A & Z & \begin{matrix}1 & & & 0\\ & 1 & & \\ & & \ddots & \\ 0 & & & 1\end{matrix} \\ \hline
\end{array}$$

Here Q is the matrix in the theorem and Z can contain anything. We know that the rows of this matrix are independent (which is equivalent to A being a basc in M) if and only if its determinant is not zero. From the special structure of the matrix it follows that this is true if and only if $\det(Q) \neq 0$.

It may be instructive to observe how Lemmas 3 and 6 could be proved for this type of matroids from Lemma 10.

We notice that if A is a base and we want to find DPG(A) with its labeling, then we must invert the matrix pictured above. We also notice with interest that the condition in Lemma 10 is invariant with respect to replacing B by E-B , since $\det(Q) = \det(Q^T)$. This immediately leads to the following lemma.

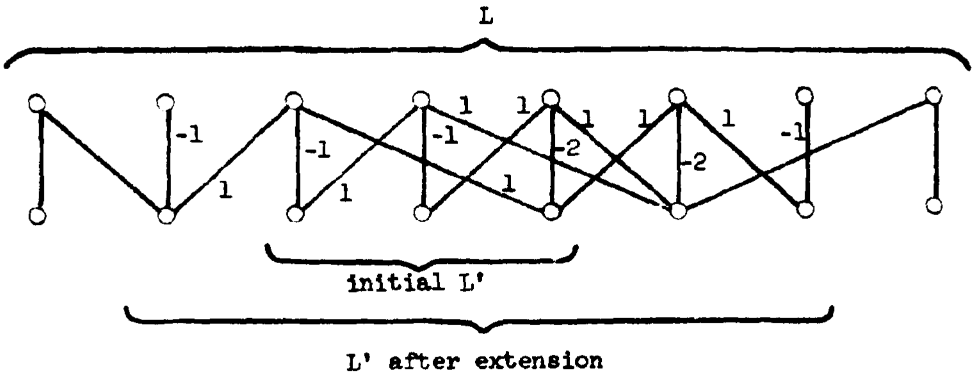
Lemma 11. Let G be a bipartite graph with a labeling from a field F , and let B be a partitioning set of G . Then the matroid defined over the set E of nodes in G with respect to B is the dual of the matroid defined on E with respect to E-B (by the same labeling).

We conclude this section by proving the earlier announced fact that Lemma 6 is best possible. This will follow from the lemma:

Lemma 12. Let G be a bipartite graph with a partitioning set B and a matching L that is not clean. Then there is a matroid over the nodes of G such that B is a base, $DPG(B) = G$, and such that $L(B)$ is not a base.

Proof. We will construct a labeling on G from the real numbers such that the matrix Q defined "between" the two sets $IN(L)$ and $OUT(L)$ (indexing the rows and columns respectively) as in Lemma 10 is such that $\det(Q) = 0$. This will be done by making a (nonempty) subset of the columns of Q sum to zero.

Since L is not clean we know by Lemma 5 that G contains an L-alternating cycle, and let L' be the subset of L that occurs in one such cycle. We note that from each node in $IN(L')$ there go at least two arcs to nodes in $OUT(L')$. This is as we want it, but we also want L' to be such that all nodes in $IN(L)$ that are reachable (by an arc) from $OUT(L')$ are also in $IN(L')$. To obtain this we extend L' by repeating the following operation until it no longer has any effect: Let X be the set of nodes in $IN(L)$ that are reachable from $OUT(L')$, and extend L' such that $IN(L') = X$ (and such that L' is still a submatching of L). When this stops, as it must by the finiteness of L, then L' obviously fulfills both the requirements we gave above (see illustration below). We now label each arc between $IN(L')$ and $OUT(L')$ with a nonzero number such that for each node in $IN(L')$ the labels of the arcs to nodes in $OUT(L')$ will sum to zero. (This is possible since there are at least two such arcs from each node in IN(L') .) We can now complete the labeling in any way we want, and we observe that the sum of the columns of Q associated with the nodes in OUT(L') will sum to zero. ⊐



An example of the construction described in the text above.

8. Binary Matroids

A matroid is said to be binary if it is induced by a finite set of vectors from a vector-space over GF(2) (the field of integers modulo 2). As Whitney proved in [9], such a matroid is fully determined by the DPG of some base. In our setting this is evident, since there is only one labeling from GF(2) of a given bipartite graph. Thus we can give a necessary and sufficient condition for a matching L in the DPG of some base B of such a matroid to be such that L(B) is a base.

<u>Lemma 13</u>. Let B be a base in a binary matroid and let L be a matching in DPG(B) . Then L(B) is a base if and only if for every nonempty subset $\subsetneq$ OUT(L) there is a node in IN(L) from which there is an odd number of arcs into I .

Proof. We prove that the negations of the two statements are equivalent. First suppose there is a set $X \subseteq OUT(L)$ such that from all elements in $IN(L)$ the number of arcs to X is even. It is then evident that the sum (modulo 2) of the columns in Q (as defined in Lemma 10, letting $A = L(B)$) corresponding to this set, is zero. Thus $\det(Q) = 0$, and $L(B)$ is not a base.

Conversely, suppose that $L(B)$ is not a base, which by Lemma 10 implies $\det(Q) = 0$. Then there must be some subset of the columns of Q such that their sum is zero, since 1 is the only nonzero constant. It is then easy to verify that the set $X \subseteq OUT(L)$ corresponding to this set of columns must be such that all nodes in $IN(L)$ has an even number of arcs into X . □

That the condition in Lemma 13 is also invariant with respect to turning the graph "upside down" is not quite transparent, but it must be true by Lemmas 10 and 11. This can be considered as a theorem in graph-theory whose proof relies on the fact that $\det(Q) = \det(Q^T)$ for all square matrices over $GF(2)$.

9. Graphic Matroids

"Graphic matroids" are binary matroids induced by a set of columns from $GF(2)^r$, each having exactly two " 1 "s and $(r-2)$ " 0 "s. We usually then identify each "row" with a node in a graph, and each column with an arc between the two nodes where it has its " 1 "s. Obviously this graph (or "multigraph", since multiple arcs may occur) fully determines this matroid (since the order of the rows is irrelevant), and for each graph there is such a matroid. However, different graphs may correspond to the same matroid.

It is well known that the circuits of such a matroid correspond to (the arcs in) the simple cycles of the graph, which means that an arc-set is independent if and only if it contains no cycle. Also, the connected components of a graphic matroid correspond to the 2-connected components of the graph.

Suppose a graphic matroid is given by a corresponding graph, and suppose we want to find the DFG of some base (which we are free to choose) of this matroid (as for example in the M-component-algorithm treated in Section 6). The best way to do this will generally be to use "depth-first search" ("DFS"), where you search along arcs and always complete the search from the latest found nodes before you go back to search from an earlier found one. (See [9].) It is then natural to keep the nodes on the path in the search-tree from the start-node to the current one, in a stack with the start node at the bottom. Then, whenever you meet an arc e which forms a cycle with the arcs picked for the base B (that is, makes $B \cup e$ dependent) then the "other" endnode of e will always be on the stack, and the rest of the arcs in the cycle formed (that is, $C(e,B)-e$) will be exactly those on the path formed by the nodes on the stack above (and included) the other endnode of e .

This makes the construction of a base and its DFG very simple, and if we study what further simplifications can be done with the M-component-algorithm given in Section 6 using this construction, we see that Hopcroft and Tarjan's DFS-algorithm given in [9] for finding the 2-connected components of a graph comes out rather naturally.

We shall also see that we can modify this algorithm so that it finds a common cycle of two given arcs A and B $(A \neq B)$ from the same 2-connected component. This version of the algorithm can also easily determine if A and B are in the same 2-connected component, but for the time being, let us assume that they are. In the following description we will use the same terminology as Tarjan uses in [9].

To force the search to produce a spanning tree that is good for our purpose, we use the following simple deviation from random choice. We start out the search in an endnode of one of our arcs, say A , and we choose arc A as the first one to follow. Let us call the start-node " r " and the other endnode of A " s ". Later, whenever we come to a new node, we first check if it is an endnode of B , and if so, we choose to follow B first. We will call the node from where we first see B " t " and the other endnode of B we call " u ".

Considerations in [9] then tell us that we will find B before we backtrack along A , which means that A is the first arc on the path from the root r to node t in the spanning tree. We also know that B will be included in the spanning tree.

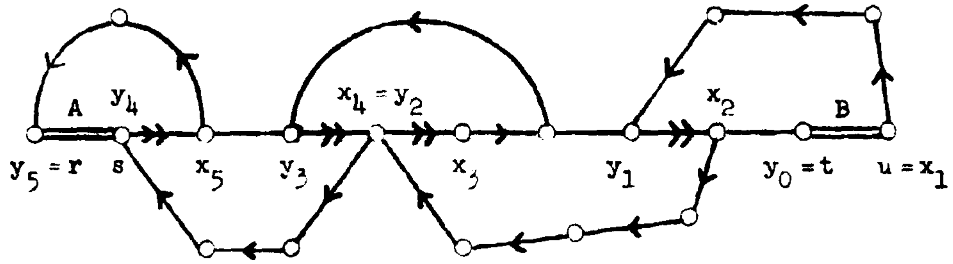
In this version we need no stack except the one that keeps track of the nodes in the tree between the root and the current node. When we see arc B for the first time (in node t) we set up a link-chain from the rootnode r through the spanning tree to node t by using the contents of this stack. For this purpose we have a pointer-field ABLINK in each node, which is now set to point towards t for all nodes on the stack. Let us call the sequence of nodes from node r through the spanning tree to node u (both included) the "AB-chain".

The nodes also have the integer-fields "NUMBER" and "LOWPT", and we fill these exactly as Tarjan does in [9] as the search proceeds. That is, in NUMBER we put consecutive growing numbers as we see the nodes for the first time, and in LOWPT of a node x we record the lowest NUMBER-field found in a node reachable from x by going further out in the spanning tree zero or more steps, and then following one frond. In our version we also keep track of the path to the node numbered LOWPT, by including and maintaining a pointer-field "LOWLINK" that points out the "direction" we followed to find this node.

Tarjan has proved in [9] that for any node x on the AB-chain (except r and s) the node y numbered with LOWPT of x also will lie on the AB-chain, y will be nearer the root r than x , and there will be at least one node between x and y on the AB-chain. This leads to the following obscure strategy for finding a path P (with no arcs, but perhaps some nodes, used twice) from node u to the root r .

We start out from node u by first remembering its LOWPT, and then following LOWLINKs until we find the node with this NUMBER. Then we test if we are in node r , and if we are not, we follow ABLINK one step, notice the LOWPT-value here, and start out following LOWLINKs again repeating the process. We stop when we find the root r .

Let us say $x_1(= u), x_2, \ldots, x_n$ is the sequence of nodes where we noticed LOWPT and started following LOWLINKs, and let $y_1, y_2, \ldots, y_n(= r)$ be the corresponding sequence of nodes where these searches succeeded. (We show below that these sequences really make progress towards the root of the tree so that r is eventually found.) This process can for example lead to something like this:



In this picture the LOWLINKs and the ABLINKs we have followed are marked with single and double arrows respectively. From earlier comments we know that each y_i , $i = 1,2,\ldots,n$ will lie nearer r than y_{i-1} (taking $y_0 = t$). This also implies that when we start following LOWLINKs from an x_i , $i = 2,3,\ldots,n$, we must "leave" the AB-chain at the latest at y_{i-2} , for if we followed it to x_{i-1} we would automatically be led back to y_{i-1} , which we are not.

Thus r must eventually be reached, and it is easy to see that if Q is the set of arcs connecting the AB-chain (including A and B) then $(P \cup Q) - (P \cap Q)$ is a simple cycle containing A and B .

We finally observe that if A and B are not in the same 2-connected component, then this algorithm will either backtrack through A without finding B , or the process of constructing P will go into a loop. Both these situations are easy to detect.

We end this paper by posing an apparently unsolved problem. We know that a binary matroid is determined by the DPG of one of its bases, and we may ask for an algorithm that determines if a certain bipartite graph G , with a given partitioning set B , is the DPG of a graphic matroid, and if so builds a graph representing this matroid.

Obviously we can treat each connected component of G alone, and it is not difficult to see that the problem is equivalent to this: Given a set of arcs A and a family D of subsets of A . If possible, put the arcs of A together to a tree such that each set in D constitutes a path in the tree. This is an easily stated combinatorial problem that may be interesting in its own right. If we know that the dual matroid is graphic and we know a representing graph (that is, if the problem is positively answered and solved for the graph G above, with respect to the complement of B) then we know we can solve this problem, as it then becomes equivalent to determine if a graph is planar. For this problem a very efficient algorithm exists, see [3].

Added in Proofreading

It turns out that W. T. Tutte has treated the above problem in the following two papers:

- "An algorithm for determining whether a given binary matroid is graphic," Proceedings of the AMS, 11 (1960), 905-917.
- "From matrices to graphs," Canadian Journal of Math., 16 (1964), 108-127.

It also turns out that an algorithm for finding the M-components of a matroid which is essentially equal to the one described on pages 14 and 15 here, is given in W. H. Cunningham's Ph.D. thesis, "A combinatorial decomposition theory," University of Waterloo, 1974, page 5.16.

References

[1] J. Edmonds and D. R. Fulkerson, "Transversals and matroid partitions," *J. Res. Nat. Bur. Standards Sec. B* 69 (1965), 147-153.

[2] F. Harary, *Graph Theory*, Addison-Wesley, Massachusetts (1969).

[3] J. Hopcroft and R. Tarjan, "Efficient Planarity Testing," *Journal of the ACM*, 21 (1974), 549-568.

[4] D. E. Knuth, *Fundamental Algorithms*, vol. 1, *The Art of Computer Programming*, Addison-Wesley, (1968).

[5] D. E. Knuth, "The asymptotic number of geometries," *Journal of Combinatorial Theory*, 16 (1974), 398-400.

[6] S. Krogdahl, "A combinatorial base for some optimal matroid intersection algorithms," Stanford Computer Science Department Report STAN-CS-74-468.

[7] E. L. Lawler, "Optimal matroid intersections," *Combinatorial Structures and their Applications*, Proceedings of the Calgary International Conference, Gordon and Breach, New York (1970), 233.

[8] T. L. Magnanti, "Independent systems and combinatorial optimization," Ph.D. thesis, Stanford University, March 1972, p. 24.

[9] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM Journal on Computing*, 1 (1972), 146-160.

[10] H. Whitney, "On the abstract properties of linear dependence," *American J. Math.*, 57 (1935), 509-533.