

Matroid Optimization and Algorithms

Robert E. Bixby

and

William H. Cunningham

June, 1990

TR90-15

MATROID OPTIMIZATION AND ALGORITHMS

by

Robert E. Bixby
Rice University

and

William H. Cunningham
Carleton University

Contents

1. Introduction
2. Matroid Optimization
3. Applications of Matroid Intersection
4. Submodular Functions and Polymatroids
5. Submodular Flows and other General Models
6. Matroid Connectivity Algorithms
7. Recognition of Representability
8. Matroid Flows and Linear Programming

1. INTRODUCTION

This chapter considers matroid theory from a constructive and algorithmic viewpoint. A substantial part of the developments in this direction have been motivated by optimization. Matroid theory has led to a unification of fundamental ideas of combinatorial optimization as well as to the solution of significant open problems in the subject. In addition to its influence on this larger subject, matroid optimization is itself a beautiful part of matroid theory.

The most basic optimizational property of matroids is that for any subset every maximal independent set contained in it is maximum. Alternatively, a trivial algorithm maximizes any $\{0, 1\}$ -valued weight function over the independent sets. Most of matroid optimization consists of attempts to solve successive generalizations of this problem. In one direction it is generalized to the problem of finding a largest common independent set of two matroids: the matroid intersection problem. This problem includes the matching problem for bipartite graphs, and several other combinatorial problems. In Edmonds' solution of it and the equivalent matroid partition problem, he introduced the notions of good characterization (intimately related to the NP class of problems) and matroid (oracle) algorithm.

A second direction of generalization is to maximizing any weight function over the independent sets. Here a greedy algorithm also works: Consider the elements in order of non-increasing weight, at each step accepting an element if its weight is positive and it is independent with the set of previously-accepted elements. A most significant step was Edmonds' recognition of a polyhedral interpretation of this fact. He used the fact that the greedy algorithm optimizes any linear function over the convex hull of characteristic vectors of independent sets to establish a linear-inequality description of that polyhedron. He then showed that a greedy algorithm also works for the much larger class of polymatroids: polyhedra that are defined from functions that, like matroid rank functions, are submodular. This polyhedral approach led to the solution of the weighted matroid intersection problem and extensive further generalizations, culminating in the optimal submodular flow problem. Other important related problems are that of minimizing an arbitrary submodular function and a polymatroid generalization of nonbipartite matching. Complete solutions of these problems remain to be discovered, but substantial progress has been made.

There is a second aspect of matroid theory, other than the optimizational one, for which a constructive approach is desirable. This might be called the structural aspect. We seek constructive answers to such fundamental questions as the connectivity, graphicness, or linear representability of a given matroid. It is pleasant to realize how many of the classical structural results of Tutte and Seymour are essentially constructive in nature. In fact, the most important structural result to date, Seymour's characterization of regular

matroids, yields an algorithm to recognize regularity by reducing that problem to recognizing graphicness and connectivity, two problems to which Tutte contributed substantially. On the other hand many structural questions can be proved to be unsolvable by matroid algorithms. Indeed, recognizing regularity turns out to be essentially the only question on linear representability that is solvable.

We use the matroid terminology introduced in the Handbook chapter of Welsh. The following additional notation is used. For a subset A of the set S , we use \overline{A} to denote $S \setminus A$. Where J is an independent set of matroid $M = (S, \mathcal{I})$, and $e \in \overline{J}$ with $J \cup \{e\} \notin \mathcal{I}$, we use $C(J, e)$ to denote the unique circuit (*fundamental circuit*) contained in $J \cup \{e\}$. Use of this notation will imply that $J \cup \{e\} \notin \mathcal{I}$.

2. MATROID OPTIMIZATION

Two of the most natural optimization problems for a matroid $M = (S, \mathcal{I})$ with weight vector $c \in \mathbf{R}^S$ are to find an independent set of maximum weight, and to find a circuit of minimum weight. These generalize standard optimization problems on graphs. We show in this section that the classical greedy algorithm solves the independent set problem. While this problem is easy, it leads to very useful polyhedral methods. Considering the greedy algorithm requires the discussion of efficiency of matroid algorithms. It turns out that, with respect to the resulting notion of algorithmic solvability, the circuit problem above is intractable.

The Optimal Independent Set Problem

If we are asked to find a maximum *cardinality* independent set, we know from the independent set axioms (Chapter Welsh) that any maximal independent set is a solution. Hence the following trivial algorithm works: Where $S = \{e_1, e_2, \dots, e_n\}$, start with $J = \emptyset$ and treat the e_i sequentially, adding e_i to J if and only if $J \cup \{e_i\}$ is independent. The *maximum weight independent set problem*

$$(2.1) \quad \max(c(J) : J \in \mathcal{I})$$

includes the above problem as a special case (take each $c_i = 1$). Moreover, (2.1) is solved by making two simple modifications to the above algorithm. First, we treat the elements of S in order of non-increasing weight, and second we do not add to J any negative-weight elements. The resulting method is the *greedy algorithm* (GA).

Greedy Algorithm for the Maximum-Weight Independent Set Problem

Order $S = \{e_1, e_2, \dots, e_n\}$ so that $c_{e_1} \geq c_{e_2} \geq \dots \geq c_{e_m} \geq 0 \geq c_{e_{m+1}} \geq \dots \geq c_{e_n}$

$J := \emptyset$

For $i = 1$ to m do

 If $J \cup \{e_i\} \in \mathcal{I}$ then $J := J \cup \{e_i\}$

(2.2) Theorem. For any matroid $M = (S, \mathcal{I})$ and any $c \in \mathbf{R}^S$, GA solves (2.1).

Proof. Suppose that $J = \{j_1, \dots, j_k\}$ is found by GA, but $Q = \{q_1, \dots, q_\ell\}$ has larger weight. Assume that the j_i are in the order in which GA added them, and that $c_{q_1} \geq c_{q_2} \geq \dots \geq c_{q_\ell}$. There is a least index i such that $c_{q_i} > c_{j_i}$ or $c_{q_i} > 0$ and $i > k$. Then $\{j_1, \dots, j_{i-1}\}$ is a basis of $A = \{j_1, \dots, j_{i-1}, q_1, \dots, q_i\}$, for otherwise GA would choose one of q_1, \dots, q_i as j_i . But $\{q_1, \dots, q_i\}$ is a larger independent subset of A , a contradiction.

□

A pair (S, \mathcal{I}) satisfying only that $\mathcal{I} \neq \emptyset$ and $A \subseteq B \in \mathcal{I}$ implies $A \in \mathcal{I}$, is sometimes called an *independence system*. Both the problem (2.1) and the greedy algorithm can be

stated for any independence system, and one wonders whether other independence systems are similarly nice. However, the matroid axioms say that (S, \mathcal{I}) is a matroid if and only if GA solves (2.1) for every $c \in \{0, 1\}^S$. In view of this observation, Theorem (2.2) can be restated as follows.

(2.3) Theorem. An independence system (S, \mathcal{I}) is a matroid if and only if GA solves (2.1) for every $c \in \mathbf{R}^S$.

A second proof of (2.2), almost as short and much more useful, leads us to the polyhedral method.

Proof. (of 2.2) Let \bar{x} be the characteristic vector of the set J produced by GA , and let x be the characteristic vector of any independent set J' . Then $c(J') = \sum c_{e_i} x_{e_i} = c \cdot x$. Let $T_i = \{e_1, \dots, e_i\}$ for $0 \leq i \leq n$. Notice that $\bar{x}(T_i) \geq x(T_i)$ for $1 \leq i \leq m$, because $J \cap T_i$ is a maximal independent subset of T_i . Then

$$\begin{aligned} c \cdot x &= \sum_{i=1}^m c_{e_i} x_{e_i} + \sum_{i=m+1}^n c_{e_i} x_{e_i} \\ &= \sum_{i=1}^m c_{e_i} (x(T_i) - x(T_{i-1})) + \sum_{i=m+1}^n c_{e_i} x_{e_i} \\ &= \sum_{i=1}^{m-1} (c_{e_i} - c_{e_{i+1}}) x(T_i) + c_m x(T_m) + \sum_{i=m+1}^n c_{e_i} x_{e_i} \\ &\leq \sum_{i=1}^{m-1} (c_{e_i} - c_{e_{i+1}}) \bar{x}(T_i) + c_m \bar{x}(T_m) + \sum_{i=m+1}^n c_{e_i} \bar{x}_{e_i}. \end{aligned}$$

But the last line is $c \cdot \bar{x}$, since the inequality holds with equality for $x = \bar{x}$. \square

Notice that the only properties of x, \bar{x} used in the second proof of (2.2) were that $x \geq 0$ and $x(T_i) \leq \bar{x}(T_i) (= r(T_i))$, $1 \leq i \leq m$. So GA actually solves the following linear programming problem, since $\bar{x}(T_i) = r(T_i)$ implies $x(T_i) \leq \bar{x}(T_i)$:

$$\begin{aligned} &\text{maximize } c \cdot x \\ &\text{subject to} \\ &x(A) \leq r(A), \quad A \subseteq S; \\ &x \geq 0. \end{aligned} \tag{2.4}$$

This observation implies the following *Matroid Polytope Theorem* of Edmonds (1970).

(2.5) Theorem. For any matroid $M = (S, \mathcal{I})$, the extreme points of the polytope $P(M) = \{x \in \mathbf{R}_+^S : x(A) \leq r(A) \text{ for all } A \subseteq S\}$ are precisely the characteristic vectors of independent sets of M .

Proof. It is easy to see that the characteristic vector of any independent set is an extreme point of $P(M)$. Now let x' be an extreme point of $P(M)$. Then there is $c \in \mathbf{R}^S$ such that x' is the unique optimal solution of $\max(c \cdot x : x \in P(M))$. Applying GA to M and this c , we obtain \bar{x} , the characteristic vector of an independent set, and \bar{x} solves the same linear programming problem. Hence $x' = \bar{x}$, as required. \square

The dual linear program to (2.4) is:

$$\begin{aligned}
 (2.6) \quad & \text{minimize } \sum (r(A)y_A : A \subseteq S) \\
 & \text{subject to} \\
 & \sum (y_A : j \in A) \geq c_j, \quad \text{for } j \in S; \\
 & y_A \geq 0, \quad \text{for } A \subseteq S.
 \end{aligned}$$

Analysis of the second proof of theorem (2.2) shows that the following formula, sometimes called the *dual greedy algorithm*, gives an optimal solution y' to (2.6):

$$\begin{aligned}
 y'_{T_i} &= c_{e_i} - c_{e_{i+1}}, & 1 \leq i < m; \\
 y'_{T_m} &= c_{e_m}; \\
 y'_A &= 0 \text{ for all other } A \subseteq S.
 \end{aligned}$$

It follows that (2.6) has an optimal solution that is integer-valued whenever c is integer-valued, that is, that the constraint system of (2.4) is *totally dual integral*. (This important notion is developed in Chapter Schrijver. One fundamental fact that we do mention, is that if $Ax \leq b$ is totally dual integral, b is integer-valued, and $\max(cx : Ax \leq b)$ has an optimal solution, then it has one that is integer-valued.) One can also use y' and the linear programming optimality conditions to prove the following converse to (2.2). (The point here is that there can be many choices of the ordering of S , because of equal weights or zero weights.)

(2.7) Theorem. $J \in \mathcal{I}$ has maximum c -weight if and only if it can be found by GA .

Efficient Matroid Algorithms

How efficient is GA ? Because sorting can be done in polynomial time, therefore GA runs in polynomial time if and only if there is a polynomial-time algorithm to answer the question: “Is $J \in \mathcal{I}$?” (In fact, we need to answer at most n such questions.) It is usual for matroids to be represented, for purposes of algorithms, by such an (independence-testing) oracle. An abstract matroid algorithm is *good* if both the number of calls to the oracle and the amount of additional computation is bounded by a polynomial in n and the size of any additional input (such as the weights). Hence GA is a good matroid algorithm. For concrete classes, such as matroids represented by matrices, for which there exists a

subroutine for independence-testing that is polynomial-time in the usual sense (that is, relative to input size), a good matroid algorithm does yield a polynomial-time algorithm.

Having defined what is meant by “good matroid algorithm” and having described an important example, we digress briefly on some related issues. First, let us explain why the oracle representation is necessary. A natural alternative would be to have a general encoding for matroids, as we do, say, for graphs, and to measure algorithm efficiency relative to the encoding-size of the input matroid. The difficulty is that the number of matroids on an n -element set (see Chapter Welsh) is so large that any general encoding scheme would require space exponential in n , which would affect the meaning of polynomial time.

Since we are not using the usual model of computation, the theory of NP-completeness does not play the same role, and it is natural to ask whether there is an analogous theory of difficult matroid problems. We call a matroid problem *intractable* if there is no good matroid algorithm solving the problem. In contrast to the situation in ordinary complexity theory, we can *prove* that certain natural problems are intractable. To illustrate a typical proof method, we consider the problem of finding the girth (minimum size of a circuit) of a matroid. Let $|S| = n = 2m$, let M be the uniform matroid of rank m on S , and, for each $A \subseteq S$ with $|A| = m$, let M_A denote the matroid obtained from M by making A a circuit. If an algorithm concludes that the girth of M is $m + 1$ after making fewer than $\binom{2m}{m}$ oracle calls, then there is some A such that the same (incorrect) conclusion would have been reached if the input had been M_A . (Notice that the argument uses the fact that all calls other than A receive the same answer for both M and M_A .) Hence a correct algorithm requires at least $\binom{2m}{m}$ calls, which is not bounded by a polynomial in n , and so computing the girth is intractable. (We mention that it is an open problem whether there is a polynomial-time algorithm to compute the girth of the matroid of a given binary matrix; on the other hand, finding a smallest circuit containing a fixed element in such a matroid is known to be NP-hard.) Other examples of intractable problems are described in later sections. Proofs can be more intricate than this example, but use similar “adversary” arguments.

Finally, we discuss why independence is an appropriate choice of oracle on which to base a theory of matroid algorithms. Consider the following alternatives: (i) “What is $r(B)$?”; (ii) “Is B a circuit?”; (iii) “What is the minimum size of a circuit contained in B ?” It is easy to see that (i) is equivalent to the independence-testing oracle, since there is a good matroid algorithm for each with respect to the other. On the other hand (ii) is weaker than independence-testing: $|B| + 1$ calls to the independence-testing oracle will be enough to determine whether B is a circuit, but there can be no polynomial-time

simulation of independence-testing by an oracle for (ii). (To see this, use an adversary approach, considering matroids on S having at most one circuit.) So (ii) would lead to a theory of matroid algorithms with respect to which independence-testing is intractable — in contrast to what the standard classes of matroids would lead us to expect. On the other hand, we have seen that (iii) would lead to a theory in which the existence of a good matroid algorithm would not imply the existence of a polynomial-time algorithm for the corresponding matrix problem.

3. MATROID INTERSECTION

We are given matroids M_1, M_2 on the same set S . We want to find a maximum weight (or maximum cardinality) common independent set. Obviously this problem generalizes the optimal independent set problem (2.1), taking $M_1 = M_2$. Some examples of applications are the following:

- (3.1) Finding a maximum weight matching in a bipartite graph;
- (3.2) Finding a maximum weight branching (forest in which each node has indegree at most 1) in a digraph;
- (3.3) Finding a maximum weight subgraph that is the union of k forests in a graph;
- (3.4) Given bases B_1, B_2 of a matroid M and $X_1 \subseteq B_1$, finding $X_2 \subseteq B_2$ such that $(B_1 \setminus X_1) \cup X_2, (B_2 \setminus X_2) \cup X_1$ are both bases.

Problem (3.1) is a direct special case of weighted matroid intersection. Where $\{V_1, V_2\}$ is a bipartition of $G = (V, E)$, take $S = E$ and $\mathcal{I}_i = \{J \subseteq S: \text{each } v \in V_i \text{ is incident with at most one element of } J\}$. Problem (3.2) is also a special case; (3.3) and (3.4) will be treated later. We begin the discussion with the maximum cardinality case.

Matroid Intersection Theorem

It is obvious, for any $J \in \mathcal{I}_1 \cap \mathcal{I}_2$ and any $A \subseteq S$, that

$$|J| = |J \cap A| + |J \cap \overline{A}| \leq r_1(A) + r_2(\overline{A}).$$

So if we can find J and A satisfying this relationship with equality, then we know that this J is maximum. In fact, such a pair (J, A) always exists; this is the content of the Matroid Intersection Theorem of Edmonds (1970). We remark that applying this theorem to maximum cardinality bipartite matching as above immediately yields König's Theorem (Chapter Pulleyblank). While the theorem follows from the algorithm described later, we first give a short non-constructive proof.

(3.5) Theorem. For matroids M_1, M_2 on S ,

$$\max(|J| : J \in \mathcal{I}_1 \cap \mathcal{I}_2) = \min(r_1(A) + r_2(\overline{A}) : A \subseteq S).$$

Proof. The proof is by induction on $|S|$. Let k be the minimum of $r_1(A) + r_2(\overline{A})$, and choose $e \in S$ with $\{e\} \in \mathcal{I}_1 \cap \mathcal{I}_2$. (If none exists, $k = 0$, and we are finished.) If the minimum of $r_1(A) + r_2(S' \setminus A)$ over subsets A of $S' = S \setminus \{e\}$ is k , then we are finished, by induction. If M'_i denotes $M_i / \{e\}$ and the minimum of $r'_1(A_1) + r'_2(A_2)$ over subsets A_1, A_2 of S' is at least $k - 1$, then induction gives a common independent set of M'_1, M'_2

of size $k - 1$; adding e to it gives the desired J . We conclude that, if there is no common independent set of size k , then there exist subsets A, B of S' such that

$$r_1(A) + r_2(S' \setminus A) \leq k - 1$$

and

$$r_1(B \cup \{e\}) - 1 + r_2((S' \setminus B) \cup \{e\}) - 1 \leq k - 2.$$

Adding and applying submodularity, we have

$$r_1(A \cup B \cup \{e\}) + r_1(A \cap B) + r_2(S \setminus (A \cap B)) + r_2(S \setminus (A \cup B \cup \{e\})) \leq 2k - 1;$$

it follows that the sum of the middle two terms, or the sum of the other two terms, is at most $k - 1$, a contradiction. (Notice that this proof gives no hint of how to find the desired subsets J and A efficiently.)

The Matroid Intersection Algorithm

The Matroid Intersection Algorithm (MIA) uses an augmenting path approach that generalizes a common method for bipartite matching. It maintains $J \in \mathcal{I}_1 \cap \mathcal{I}_2$, at each step either finding a larger such J , or finding A giving equality with J in the min-max formula, proving that J is maximum. Given $J \in \mathcal{I}_1 \cap \mathcal{I}_2$, an auxiliary digraph $G = G(M_1, M_2, J)$ having node-set $S \cup \{s, t\}$ is constructed. It contains:

- an arc (e, t) for every $e \in S \setminus J$ such that $J \cup \{e\} \in \mathcal{I}_1$;
- an arc (s, e) for every $e \in S \setminus J$ such that $J \cup \{e\} \in \mathcal{I}_2$;
- an arc (e, f) for every $e \in S \setminus J, f \in J$ such that $f \in C_1(J, e)$;
- an arc (f, e) for every $e \in S \setminus J, f \in J$ such that $f \in C_2(J, e)$.

The following “augmenting path” theorem is the basis for the algorithm.

- (3.6) Theorem** (a) If there exists an (s, t) -dipath in G , then J is not maximum; in fact, if $s, e_1, f_1, \dots, e_k, f_k, e_{k+1}, t$ is a chordless (s, t) -dipath, then $J' = (J \cup \{e_1, \dots, e_{k+1}\}) \setminus \{f_1, \dots, f_k\} \in \mathcal{I}_1 \cap \mathcal{I}_2$.
- (b) If there exists no (s, t) -dipath in G , then J is maximum; in fact, if $A \subseteq S$ and $\delta^+(A \cup \{s\}) = \emptyset$, then $|J| = r_1(A) + r_2(\overline{A})$.

We remark that, while the restriction to chordless dipaths is not necessary for the special case of bipartite matching, it is needed in general. The proof of (b) in (3.6) is easy. Consider $e \in A \setminus J$. Since (e, t) is not an arc, $J \cup \{e\}$ contains an M_1 -circuit C . Since there is no arc (e, f) with $f \in S \setminus A$, we have $C \subseteq (A \cap J) \cup \{e\}$. Therefore, $J \cap A$ M_1 -spans A . Similarly, $J \cap \overline{A}$ M_2 -spans \overline{A} . Hence

$$|J| = |J \cap A| + |J \cap \overline{A}| = r_1(A) + r_2(\overline{A}).$$

We can deduce (a) of (3.6) from the following result, whose proof is a straightforward induction. (Recall that $\sigma(A)$ denotes the closure of the subset A .)

(3.7) Lemma. Let $M = (S, \mathcal{I})$ be a matroid, let $J \in \mathcal{I}$ and let $x_1, y_1, \dots, x_k, y_k$ be a sequence of distinct elements of S such that

- (a) $x_i \notin J$, $y_i \in J$ for $1 \leq i \leq k$;
- (b) $y_i \in C(J, x_i)$ for $1 \leq i \leq k$;
- (c) $y_i \notin C(J, x_j)$ for $1 \leq i < j \leq k$.

Then where $J' = (J \cup \{x_1, \dots, x_k\}) \setminus \{y_1, \dots, y_k\}$, $J' \in \mathcal{I}$ and $\sigma(J') = \sigma(J)$.

Efficiency of the Matroid Intersection Algorithm

Since MIA will terminate after at most $n = |S|$ augmentations, since the dipath or set A as in (3.6) can be found with standard methods, and since each auxiliary digraph can be constructed with $O(n^2)$ independence tests, therefore MIA is a good matroid algorithm. Here we mention the complexity for some refinements and special cases. Just as MIA generalizes a basic bipartite matching algorithm, more efficient versions of MIA generalize some of the ideas used to speed up matching and network flow algorithms. (See Chapter Frank.) The most important refinement is a natural one: At each step augment J using a shortest (s, t) -dipath of the auxiliary digraph. Of course, any such dipath will automatically be chordless. The next result implies that a large proportion of the resulting augmentations will be on very short augmenting paths. It is from Cunningham (1986), and part (a) is also due to Gabow and Stallmann (1985).

(3.8) Theorem. If MIA using shortest augmenting paths is applied to matroids M_1, M_2 on S , $|S| = n$, then:

- (a) The length of a shortest augmenting path never decreases, and the number of different lengths is $O(\sqrt{n})$;
- (b) The sum of lengths of augmenting paths used is $O(n \log n)$.

It follows from (a) of (3.6) that the work of the algorithm can be divided into $O(\sqrt{n})$ stages; during each stage all augmenting paths have the same length. It is possible to find and perform all of the augmentations of a stage more efficiently than if the auxiliary digraph were reconstructed after each stage. This leads to a version of MIA (Cunningham (1986)) that requires $O(n^{2.5})$ independence tests rather than the $O(n^3)$ of the basic algorithm. How close is this bound to being best possible? One of the few results on this theme is that the greedy algorithm for finding a basis of a single matroid is optimal; it requires exactly n independence tests, and an easy argument shows that no algorithm uses fewer. For matroid intersection nothing more is known, that is, it is an open problem to find a nonlinear lower bound on the number of independence tests required.

For concrete classes of matroids, one can usually obtain better bounds than arise from simply multiplying the number of oracle calls by the oracle complexity. For example, for matroids arising from two matrices each having at most n rows, part (b) of (3.8) can be used to show that the complexity of MIA is $O(n^3 \log n)$, assuming that arithmetic operations are counted as single steps. As another example, Gabow and Stallmann (1985) have given an $O(p^{2.5})$ time bound for MIA on the cycle matroids of two graphs, each having at most p nodes.

Matroid Partitioning

Many of the applications of matroid intersection are most easily derived through the theory of matroid partitioning. In fact this theory is equivalent to that for matroid intersection and actually was discovered earlier by Edmonds.

The matroid partitioning problem is, given matroids $M_i = (S, \mathcal{I}_i)$, $1 \leq i \leq k$, to find a maximum cardinality subset $J \subseteq S$ that is *partitionable*, that is, $J = \cup(J_i : 1 \leq i \leq k)$ where $J_i \in \mathcal{I}_i$, $1 \leq i \leq k$. Obviously, we may assume that the J_i are disjoint. Moreover, the assumption that all M_i have underlying set S is made only for convenience. The main result of the theory is the following Matroid Partition Theorem.

(3.9) Theorem. Let J be a maximal partitionable subset with respect to $M_i = (S, \mathcal{I}_i)$, $1 \leq i \leq k$. Then $|J| = \min_{A \subseteq S} \left(\sum_{i=1}^k r_i(A) + |\overline{A}| \right)$.

As usual we can observe that for any such J and A , we have $|J| = |J \setminus A| + |J \cap A|$

$$\begin{aligned} &\leq |S \setminus A| + \sum |J_i \cap A| \\ &\leq |\overline{A}| + \sum r_i(A). \end{aligned}$$

Notice that (3.9) is stronger than a simple max-min equality; it implies that every maximal partitionable subset is maximum. From this observation, one easily obtains the following important consequence.

(3.10) Theorem. The subsets of S partitionable with respect to the M_i , form the independent sets of a matroid. Its rank function r is given by $r(B) = \min(|B \setminus A| + \sum r_i(A) : A \subseteq B)$.

The matroid partitioning problem is reduced to a matroid intersection problem as follows. (This construction, and the reverse one described later, are due to Edmonds (1970).) Make k disjoint copies S_1, S_2, \dots, S_k of S , and imagine M_i as being defined on S_i rather than S . Let N_a be the direct sum of the M_i and let N_b be the matroid on $S' = \cup S_i$ in which a set is independent if and only if it contains at most one copy of e

for each $e \in S$. It is easy to see that there is a correspondence between partitionable sets with respect to M_1, \dots, M_k and common independent sets of N_a, N_b . It is also easy to see that a set $B \subseteq S'$ that minimizes $r_a(B) + r_b(S' \setminus B)$ can be chosen to consist of all the copies of elements of A , for some $A \subseteq S$. It follows that the maximum size of a partitionable set is $\min(|\overline{A}| + \sum r_i(A) : A \subseteq S)$. Moreover, every maximal partitionable set has this cardinality, since whenever a copy of e is deleted from the common independent set of N_a, N_b by the intersection algorithm, it is replaced by another copy of e , so that no element is ever deleted from the partitionable set. Now (3.10) follows from the fact that the same argument could be applied to maximal partitionable subsets of an arbitrary subset B .

There is a neater description of the partitioning algorithm, obtained by identifying all copies of each element e of S in the auxiliary digraph for the intersection algorithm. The resulting digraph has node-set $S \cup \{s, t\}$ and has

- an arc (s, e) for each $e \in S \setminus J$;
- an arc (e, t) for each $e \in S$ such that $J_i \cup \{e\} \in \mathcal{I}_i$ for some i ;
- an arc (e, f) for each $e, f \in S$ such that $f \in C_i(J_i, e)$ for some i .

At termination of the algorithm, any set $A \subseteq S$ such that $\delta^+(A \cup \{s\}) = \emptyset$ has the property that $J_i \cap A$ M_i -spans A , and so $|J| = |\overline{A}| + \sum r_i(A)$. It is worthwhile also to observe that *every* set A that minimizes $|\overline{A}| + \sum r_i(A)$ must have this property and so must satisfy $\delta^+(A \cup \{s\}) = \emptyset$.

Now recall problem (3.3) at the beginning of the section. By (3.10) the feasible solutions form the independent sets of a matroid. Hence (3.3) can be solved by the greedy algorithm, with independence tests requiring applications of the matroid partition algorithm to k copies of the cycle matroid of the graph. The Matroid Partition Theorem applied to this example yields standard graph results of Nash-Williams and Tutte on the existence of disjoint spanning trees and the covering of edges by forests. (See Chapter Frank.) There are also beautiful combinatorial applications in transversal theory; see Chapter Welsh and Mirsky (1971).

Finally, let us describe Edmonds' reduction of intersection to partitioning. The proof is easy.

(3.11) Theorem. Let B be a basis of M_2^* . Extend B to a maximal partitionable set B' with respect to M_1 and M_2^* . Then $B' \setminus B$ is a maximum cardinality common independent set of M_1 and M_2 .

Basis Exchange

Recall problem (3.4). If X_2 has the required properties, then $B_2 \setminus X_2$, X_2 provides a partitioning of B_2 with respect to the matroids $M_1 = M/X_1$ and $M_2 = M/(B_1 \setminus X_1)$. Thus X_2 can be found with the Matroid Partition Algorithm. Moreover, applying the Matroid Partition Theorem one gets, after a short calculation, the following result of Greene.

(3.12) Theorem. Let B_1, B_2 be bases of a matroid M and let $X_1 \subseteq B_1$. Then there exists $X_2 \subseteq B_2$ such that $(B_1 \setminus X_1) \cup X_2$ and $(B_2 \setminus X_2) \cup X_1$ are also bases.

Solution of the Shannon Game

The Shannon game, proposed by Shannon and generalized to matroids and solved by Lehman (1964), is a game played on a matroid with a single distinguished element e . (See also Chapter Guy.) The two players, Short and Cut, alternately choose elements of $S \setminus \{e\}$, with elements chosen by Cut deleted from M and elements chosen by Short contracted. Short's (Cut's) objective is to reach a minor in which e is a loop (coloop).

A game (M, e) is called *short* (*cut*) if there is a winning strategy for Short (Cut) playing second (and hence also playing first). The game is *neutral* if it is neither cut nor short, that is, if the first player, whether Cut or Short, has a winning strategy. It is easy to see that (M, e) is a cut game if and only if (M^*, e) is short, and that (M, e) is neutral if and only if (M, e) is not short and (M', e) is short, where M' is obtained from M by adding an element parallel to e . Hence it is enough to characterize short games.

(3.13) Theorem. (M, e) is short if and only if there exist disjoint independent sets I_1, I_2 of M such that $e \in \sigma(I_1) = \sigma(I_2)$ and $e \notin I_1 \cup I_2$.

The “if” part of (3.13) can be proved by checking that the following strategy, applied iteratively, works. If Cut plays $f \in I_1$ then Short plays $e \in I_2$, where (with respect to the current minor just before f is deleted) $(I_1 \cup \{e\}) \setminus \{f\}$ is independent, and similarly for $f \in I_2$. (If Cut does not play an element of $I_1 \cup I_2$, then this only makes life easier for Short.) The “only if” part can be proved (this is harder) by showing that I_1, I_2 exist for one of $(M, e), (M^*, e), (M', e)$.

The condition of (3.13) is easily recognized by the partitioning algorithm applied with $M_1 = M_2 = M$. (In fact, Lehman's work was one of the motivations for Edmonds' development of matroid partitioning.) We know that the minimizers of g , where $g(A) = |S \setminus A| + 2r(A)$, are precisely the sets $A \subseteq S$ such that, in the auxiliary digraph at termination of the algorithm, $\delta^+(A \cup \{s\}) = \emptyset$. There is a unique smallest such A (easily found); call it A' . If $e \in A'$, then the minimum of g remains the same when e is deleted, so there is a maximum partitionable set $J = J_1 \cup J_2$ with $e \notin J$. Then $e \in A' = \sigma(J_1 \cap A') = \sigma(J_2 \cap A')$, so $J_1 \cap A', J_2 \cap A'$ are the required sets. On the other hand, we claim that if such sets I_1, I_2 exist, then necessarily $e \in A'$.

For suppose we start the partition algorithm with $J_1 = I_1$, $J_2 = I_2$. Then in the auxiliary digraph, we have $\delta^+(\sigma(I_1)) = \emptyset$, so no augmenting path can use any element of $\sigma(I_1)$ and so $I_1 \subseteq J_1$, $I_2 \subseteq J_2$, $e \notin J_1 \cup J_2$ will be maintained throughout execution of the algorithm. At termination $\delta^+(A' \cup \{s\}) = \emptyset$, and so $e \in A'$, as required. Finally, we point out that the sets I_1, I_2 , can be found by applying the algorithm to $M \setminus e, M \setminus e$ after first checking that $e \in A'$.

Further analysis of the game and interesting extensions due to Edmonds, Bruno and Weinberg, and the authors can be found in Hamidoune and Las Vergnas (1986).

Weighted Matroid Intersection

Just as $r_1(A) + r_2(\bar{A})$ provides an upper bound for the size of a common independent set J , where $c \in \mathbf{R}^S$ we can define a simple upper bound for $c(J)$. Namely let (c^1, c^2) be a “weight-splitting”, that is, $c^1 + c^2 = c$. Then

$$(3.14) \quad c(J) = c^1(J) + c^2(J) \leq \max_{J_1 \in \mathcal{I}_1} c^1(J_1) + \max_{J_2 \in \mathcal{I}_2} c^2(J_2).$$

Hence, if we find $J \in \mathcal{I}_1 \cap \mathcal{I}_2$ and a weight-splitting (c^1, c^2) such that equality holds in (3.14), we know that J has maximum weight. In fact, this is always possible.

(3.15) Theorem. For matroids M_1, M_2 on S and $c \in \mathbf{R}^S$, there exists a weight-splitting (c^1, c^2) and a set J such that J has maximum c^i -weight among independent sets of M_i , for $i = 1$ and 2 .

We shall also see that, if c is integer-valued, then there is an integral weight-splitting in (3.15). Actually, (3.15) is equivalent to the *Matroid Intersection Polytope Theorem* of Edmonds (1970). (Notice that the converse of (3.16) is trivially true.)

(3.16) Theorem. For matroids M_1, M_2 on S , every extreme point of $P(M_1) \cap P(M_2)$ is the characteristic vector of a common independent set.

Proof. (of equivalence of (3.15) and (3.16)). Suppose that (3.15) holds and let \bar{x} be an extreme point of $P(M_1) \cap P(M_2)$. Choose $c \in \mathbf{R}^S$ such that \bar{x} is the unique optimal solution of $\max(c \cdot x : x \in P(M_1) \cap P(M_2))$. Then by (3.15) we have $J \in \mathcal{I}_1 \cap \mathcal{I}_2$ and (c^1, c^2) such that $c(J) = c^1(J) + c^2(J) \geq c^1 \cdot \bar{x} + c^2 \cdot \bar{x} = c \cdot \bar{x}$. It follows that \bar{x} is the incidence vector of J .

Now suppose that every extreme point of $P(M_1) \cap P(M_2) = \{x \in \mathbf{R}_+^S : x(A) \leq r_1(A), x(A) \leq r_2(A), \text{ for } A \subseteq S\}$ is the characteristic vector of a common independent set. Given $c \in \mathbf{R}^S$, let J be a maximum weight common independent set, and let \bar{x} be its characteristic vector. Let (y^1, y^2) be an optimal solution of the linear program dual to $\max(c \cdot x : x \in P(M_1) \cap P(M_2))$. Then the optimality conditions imply that $|J \cap A| = r_i(A)$

whenever $y^i(A) > 0$. For $i = 1$ and 2 and $j \in S$, let $c_j^i = \sum(y^i(A) : j \in A)$; then $c \leq c^1 + c^2$. Thus \bar{x}, y^i satisfy the optimality conditions for $\max(c^i \cdot x : x \in P(M_i))$ and its dual, so J is c^i -optimal in M_i , for $i = 1$ and 2 . Moreover, $c(J) = c^1(J) + c^2(J)$, so $c_j^1 + c_j^2 > c_j$ implies $j \notin J$, so c_j^2 can be lowered to $c_j - c_j^1$ without affecting the c^2 -optimality of J . \square

Although Edmonds (1970) used the idea of weight-splitting in a non-constructive proof of (3.16), it was Frank (1981) who showed how to use the optimality conditions based on (3.15) to simplify the weighted matroid intersection algorithms of Edmonds (1979) and Lawler (1975). We describe here an algorithm based essentially on Frank's, but with an additional simplification.

The basic idea is to generalize the unweighted MIA by using a weight-splitting to assign costs to the arcs of the auxiliary digraph. Where c_0^i denotes $\max(c_e^i : e \notin J, J \cup \{e\} \in \mathcal{I}_i)$, the arc costs w_{uv} are defined by:

$$\begin{aligned} w_{et} &= c_0^1 - c_e^1; \\ w_{se} &= c_0^2 - c_e^2; \\ w_{ef} &= -c_e^1 + c_f^1; \\ w_{fe} &= -c_e^2 + c_f^2. \end{aligned}$$

We shall require that $(c^1, c^2), J$ satisfy the properties $c_0^2 = 0$, and $w_{uv} \geq 0$ for each arc uv . If in addition, we have $c_0^1 \leq 0$, then the conditions of (3.15) are satisfied (by (2.7)) and we are finished. We can begin with $c^1 = c$, $c^2 = 0$, $J = \emptyset$. (Notice that, if we augment on an (s, t) dipath P to obtain J' from J , then the cost of P is $c_0^1 + c_0^2 + c(J) - c(J')$. This motivates choosing P to have least cost. In fact, solving a least-cost dipath problem gives a way to update the weight-splitting too. This observation makes possible the following simpler presentation of Frank's algorithm. We remark that Lawler also used a shortest path calculation, but without weight-splitting.)

Iteration of Weighted MIA

If $c_0^1 \leq 0$, stop;

Form $G = G(M_1, M_2, J, c^1, c^2)$;

Compute a dipath from s to t of least cost d_v for each $v \in S \cup \{s, t\}$;

For each $v \in S$, let $\alpha_v = \min(d_v, d_t, c_0^1)$, and replace c_v^1 by $c_v^1 - \alpha_v$, c_v^2 by $c_v^2 + \alpha_v$;

If $c_0^1 \leq 0$, stop;

Augment J on a zero-cost (s, t) dipath having as few edges as possible.

Notice that the resulting algorithm has essentially the same complexity as its unweighted version, since the non-negative-cost shortest path calculation can be done in

time $O(n^2)$ (see Chapter Frank). We outline a proof of validity of the algorithm. There are three things to check: (i) That the change in (c^1, c^2) preserves the properties required of it; (ii) That J remains common independent after an augmentation; (iii) That an augmentation does not violate the properties required of (c^1, c^2) . It is straightforward to check (i), using the fact that the d_v satisfy $d_u + w_{uv} \geq d_v$. Notice that (ii) is not obvious, since the dipath may have chords (but not zero-cost ones). One actually shows that the subgraph induced by the zero-cost arcs is $G(M'_1, M'_2, J)$ for new matroids M'_1, M'_2 for which J is common independent, and every common independent set is also independent in both M_1 and M_2 . Then the result follows from the validity of the unweighted MIA. To define M'_1 , let $p_1 > p_2 > \dots > p_k$ be the distinct values of c_e^1 that are greater than c_0^1 , let T_i denote $\{e \in S : c_e^1 \geq p_i\}$, and let $M'_1 = (M_1|T_1) \oplus (M_1/T_1)|(T_2 \setminus T_1) \oplus \dots \oplus (M_1/T_k)$. For (iii), it is easy to show that the change in J preserves the property that $c_0^2 = 0$; to show that it also preserves $w \geq 0$, we use the fact that the latter condition is equivalent to J being c^i -optimal of its cardinality in \mathcal{I}_i , $i = 1$ and 2 . Consider $i = 1$, and let e_{m+1} be the second-last node of the augmenting path P . Then $c_{m+1}^1 = c_0^1$, so $J \cup \{e_{m+1}\}$ is c^1 -optimal of cardinality $|J| + 1$ in \mathcal{I}_1 . But $c^1(J') = c^1(J \cup \{e_{m+1}\})$, since each element of J on P has the same c^1 -weight as the preceding node on P , and we are done.

A good deal of recent work has been done on faster implementations of a MIA for special classes of matroids. See Gabow and Tarjan (1984), Brezovec, Cornuéjols, and Glover (1988), and Gabow and Xu (1989).

4. SUBMODULAR FUNCTIONS AND POLYMATROIDS

Let f be a function defined on subsets of S , with values in \mathbf{R} ; f is *submodular* if $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$ for all $A, B \subseteq S$.

Some examples:

(4.1) Let M be a matroid on S , and let f be the rank function of M .

(4.2) Let $G = (V, E)$ be a digraph, let $s \in V$, let $S = V \setminus \{s\}$, and let $f(A) = |\delta^-(A)|$. (Specifying s is not necessary here, but will be useful later.)

Given a set function f on S we let $P(f)$ denote the polyhedron $\{x \in \mathbf{R}_+^S : x(A) \leq f(A) \text{ for all } A \subseteq S\}$. A *polymatroid* is a polyhedron of the form $P(f)$ where f is submodular and non-negative. It is said to be *integral* if f is integral. It follows from (2.5) that the polymatroid defined by f of (4.1) is the convex hull of characteristic vectors of independent sets of M . In example (4.2), if there exists in G a family of edge-disjoint directed paths, each beginning at s and ending in S , and we let x_v denote the number of dipaths ending at v , then it is easy to see that $x \in P(f)$. It is a consequence of a form of Menger's Theorem that every integer-valued element of $P(f)$ arises in this way.

The first result ((4.3) below) shows that our polymatroids do satisfy the central condition of Edmonds' original geometric definition: For any $u \in \mathbf{R}_+^S$, all maximal (with respect to component-wise order) vectors $x \in P$ with $x \leq u$, have the same component-sum. Another consequence of this result is a construction for matroids: The $\{0, 1\}$ -valued vectors in an integral polymatroid are the characteristic vectors of the independent sets of a matroid, and (4.3) gives a formula for its rank function. In particular, the fact that r of Theorem (3.10) is a matroid rank function follows from applying this to $f = \Sigma r_i$.

(4.3) Theorem. Let f be submodular on S , let $u \in \mathbf{R}_+^S$, and let x be any maximal vector satisfying $x \leq u, x \in P(f)$. Then $x(S) = \min_{A \subseteq S} (f(A) + u(\bar{A}))$. Moreover, if f and u are integer-valued and x is required to be integer-valued, the conclusion is still satisfied.

Proof. First we observe that, for any $A \subseteq S$ and any x (maximal or not), we have $x(S) = x(A) + x(\bar{A}) \leq f(A) + u(\bar{A})$. Therefore, it will be enough to prove that there is some A for which equality holds. Obviously, for each $j \in S$, if $x_j \neq u_j$, then by the maximality of x , there is a set $A_j \subseteq S$ with $j \in A_j$ such that $x(A_j) = f(A_j)$. Call such a set *x-tight*, or just *tight*. A fundamental fact is:

(4.4) Claim. The intersection and union of tight sets are also tight.

Proof of claim. If A, B are tight, we have

$$\begin{aligned}
x(A \cup B) + x(A \cap B) &\leq f(A \cup B) + f(A \cap B) \\
&\leq f(A) + f(B) \\
&= x(A) + x(B) \\
&= x(A \cup B) + x(A \cap B),
\end{aligned}$$

so $x(A \cup B) = f(A \cup B)$ and $x(A \cap B) = f(A \cap B)$. \square

Now if we choose A to be the union of the A_j , then A is tight and $x(\bar{A}) = u(\bar{A})$, so we are finished. The same proof applies to the integer-restricted version. \square

It is a consequence of (4.3) that a greedy algorithm maximizes $x(S)$ over $P(f)$ (or more generally over $\{x : x \in P(f), x \leq u\}$). The algorithm begins with $x = 0$, and for each $j \in S$ increases x_j as much as possible subject to the restriction that $x \in P(f)$. Just as in the special case of matroid polytopes, we generalize to arbitrary weight-vectors by treating the elements in order of non-increasing weight.

Polymatroid Greedy Algorithm (PGA)

Order $S = \{e_1, \dots, e_n\}$ so that $c_{e_1} \geq \dots \geq c_{e_m} \geq 0 \geq c_{e_{m+1}} \geq \dots \geq c_{e_n}$

$x := 0$

For $i = 1$ to m do

 Choose x_{e_i} as large as possible so that $x \in P(f)$.

(4.5) Theorem. For any non-negative submodular function f on S and any $c \in \mathbf{R}^S$, PGA optimizes $c \cdot x$ over $P(f)$. Moreover, if f is integer-valued, the output of PGA is integer-valued.

Where $T_i = \{e_1, \dots, e_i\}$ and \bar{x} is the output of PGA, one can apply (4.3) to deduce that $\bar{x}(T_i) = \max(x(T_i) : x \in P(f))$. With this observation, (4.5) can be proved in the same way as (2.2). Other results for matroid polytopes immediately generalize. These include the fact that the extreme points of $P(f)$ are precisely the vectors that can be the output of PGA, the dual greedy algorithm, and the total dual integrality of the linear system for $P(f)$. One also can prove a converse of (4.5) similar to (2.3), characterizing polymatroids as the compact subsets of \mathbf{R}_+^S that are closed below and for which a greedy algorithm always works.

Call a function f a *polymatroid function* if f is submodular, normalized ($f(\emptyset) = 0$), and monotone ($f(A) \geq f(B)$ if $A \supseteq B$). For an arbitrary submodular function f , the function f' defined by $f'(A) = \min(f(B) : B \supseteq A)$, is obviously monotone, and is easily

seen to be submodular. Notice that, for the f of example (4.2), $f(A)$ is the maximum number of edge-disjoint directed paths beginning at s and ending in A .

More generally, applying (4.3) with $u_j = 0, j \notin A$ and u_j large, $j \in A$, yields

$$(4.6) \quad f'(A) = \max(x(A) : x \in P(f)), \text{ for } A \neq \emptyset.$$

It follows that $P(f) \subseteq P(f')$. But $f' \leq f$ by definition, so $P(f) = P(f')$. Notice also that if f is a polymatroid function, then $f = f'$ and PGA reduces to a formula (which works in the slightly more general case when f is non-negative and monotone):

$$(4.7) \quad \bar{x}_{e_i} = \begin{cases} f(T_i), & i = 1; \\ f(T_i) - f(T_{i-1}), & 2 \leq i \leq m; \\ 0, & i > m. \end{cases}$$

It is easy to derive from these observations the following result of Edmonds.

(4.8) Theorem. Every polymatroid is determined by a (unique) polymatroid function.

Submodular Function Minimization

We have seen from (4.7) that when f is non-negative and monotone, the greedy algorithm is especially simple. Its efficiency depends on the ease with which we can obtain function values, and the size (number of digits) of the values. There is no difficulty with the first of these, since we assume that the function is given via an evaluation oracle. On the other hand, the maximum size of function values must, like $n = |S|$, be treated as a measure of input size. (We must do the same for the element weights c_j .) With these ground rules, the greedy algorithm can be regarded as a polynomial-time oracle algorithm, when f is monotone.

On the other hand, if no additional assumption on f is made, computing component e of \bar{x} in the greedy algorithm requires minimizing $f(A) - \bar{x}(A)$ over subsets A containing e , a problem easily seen to be equivalent to that of finding the minimum of an arbitrary submodular function. This latter problem is fundamental; it includes as special cases both the minimum cut problem and, by (3.9), the problem of finding the maximum size of a partitionable set. Using some of the above results, we shall show that the minimum of a submodular function can be well-characterized. (The obverse problem of maximizing a submodular function, on the other hand, includes NP-hard special cases, and can be easily proved intractable in the oracle context.)

It is useful to reduce the general problem of submodular function minimization to that of minimizing a function of the form $f(A) - u(A)$, where f is a polymatroid function and $u \in \mathbf{R}_+^S$. Let g be a submodular function on S and let $u_j = g(S \setminus \{j\}) - g(S)$, $j \in S$. If $u_j < 0$, it is easy to see that no minimizer of g will include j , so the problem could

be restricted to subsets of $S \setminus \{j\}$. Hence we may assume that $u \geq 0$. The function f defined by $f(A) = g(A) + u(A) - g(\emptyset)$ is easily shown to be a polymatroid function. Hence minimizing g is equivalent to minimizing $f(A) + u(\overline{A})$, since this differs from g by the constant $u(S) - g(\emptyset)$. Then (4.3) characterizes the minimum. That this is a useful characterization is not completely obvious, since the maximizing x must be certifiably in $P(f)$. But x can be expressed as a convex combination of at most $n + 1$ extreme points of $P(f)$, by a standard result in polyhedral theory, and these can be generated by PGA, since f is a polymatroid function.

In fact, a minimization algorithm can be based on the above ideas: Maintain $x \in P(f)$ with $x \leq u$ explicitly as a convex combination of extreme points of $P(f)$, and at each step either find A giving equality in (4.3), or find a new x with $x(S)$ larger. This combinatorial approach was first developed (Cunningham (1984)) for the special case in which f is a matroid rank function. The resulting algorithm, which can be viewed as a generalization of the matroid partition algorithm, runs in polynomial time. For general f , a finite algorithm occurs in Bixby, Cunningham, and Topkis (1985), and it was modified to run in “pseudo-polynomial” time (Cunningham (1985)).

Grötschel, Lovász, and Schrijver (1981) *did* find a polynomial-time algorithm for submodular function minimization. It is based on the equivalence, via the ellipsoid method, of the optimization and separation problems for polyhedra. (See Chapter Schrijver.) Since we wish to minimize $f(A) - u(A)$, it is enough to be able to determine, given $K \in \mathbf{R}$, either a set $A \subseteq S$ for which $f(A) - u(A) < K$, or the information that no such A exists. (For then one could search over K for a sufficiently small K for which A does exist.) The function f_K defined by $f_K(B) = f(B) - K$ is submodular and monotone, and such A exists if and only if $u \notin P(f_K)$. But this is the separation problem for $P(f_K)$, and since f_K is monotone, the optimization problem for $P(f_K)$ is solvable, and we are done. The resulting algorithm, while theoretically acceptable, is not computationally useful. An important open question is the existence of a polynomial-time *combinatorial* minimization algorithm.

Polymatroid Intersection

The matroid intersection theorem (3.5) is a special case of a result of Edmonds (1970) on polymatroids.

(4.9) Theorem. Let f_1, f_2 be polymatroid functions on S . Then

$$\max(x(S) : x \in P(f_1) \cap P(f_2)) = \min_{A \subseteq S} (f_1(A) + f_2(S \setminus A)).$$

Moreover, if f_1, f_2 are integer-valued, then the maximizing x can be chosen to be integer-valued.

We remark that, if f_1, f_2 are not required to be monotone, then by monotonization arguments the same result holds, except that the right-hand side becomes $\min(f_1(A) + f_2(B) : A \cup B = S)$. The inductive proof of (3.5) outlined in Section 2 generalizes to a proof of the integral version of (4.9). (The induction is now on $\Sigma(\min(f_1(\{j\}), f_2(\{j\})) : j \in S)$, and the appropriate analogue of contracting $e \in S$ is to form the function f_i^e by $f_i^e(A) = \min(f_i(A), f_i(A \cup \{e\}) - 1)$. The non-integral version can be deduced from the integral version in a straightforward way. Later we shall see other proofs and algorithmic aspects of (generalizations of) (4.9).

The following sandwich theorem of Frank (1982) is a useful and attractive restatement of (4.9). Its resemblance to classical results on separation of convex and concave functions is evident; we shall see that the relationship is more than an analogy. A set function f is *supermodular* if $-f$ is submodular, and is *modular* if it is both sub- and supermodular. It is easy to see that a function m is modular on subsets of S if and only if $m(A) = x(A) + k$ for some $x \in \mathbf{R}^S, k \in \mathbf{R}$.

(4.10) Theorem. Let g, h be defined on subsets of S such that g is supermodular, h is submodular, and $g \leq h$. Then there exists a modular function m satisfying $g \leq m \leq h$. Moreover, if f and g are integer-valued, then m may be chosen integer-valued.

To derive (4.10) from (4.9) one proceeds as follows. Add a constant to g and h to make $g(\emptyset) = 0$. Lower $h(\emptyset)$ to 0. Raise $g(S)$ to $h(S)$. Add a function p of the form $p(A) = M|A|$ to make f and g monotone. Now take $f_1 = h$, define f_2 by $f_2(A) = g(S) - g(S \setminus A)$, apply (4.9) to find $x \in P(f_1) \cap P(f_2)$ with $x(S) = g(S)$, and define m by $m(A) = x(A)$. A similar construction allows the derivation of (4.9) from (4.10).

Optimization over the Intersection of Polymatroids

The problem of optimizing a linear function over the intersection of two polymatroids may be stated as a linear program:

$$\begin{aligned}
 (4.11) \quad & \text{maximize } c \cdot x \\
 & \text{subject to} \\
 & x(A) \leq f_1(A), \quad A \subseteq S; \\
 & x(A) \leq f_2(A), \quad A \subseteq S; \\
 & x_j \geq 0, \quad j \in S.
 \end{aligned}$$

The dual linear program is

$$\begin{aligned}
 (4.12) \quad & \text{minimize } \Sigma(f_1(A)y_A^1 + f_2(A)y_A^2 : A \subseteq S) \\
 & \text{subject to}
 \end{aligned}$$

$$\begin{aligned}\Sigma(y_A^1 + y_A^2 : j \in A \subseteq S) &\geq c_j, \quad j \in S; \\ y_A^1, y_A^2 &\geq 0, \quad A \subseteq S.\end{aligned}$$

The main result (Edmonds (1970)) on this topic may be stated as follows.

(4.13) Theorem. If f_1, f_2 are integer-valued, then (4.11) has an optimal solution that is integer-valued. If c is integer-valued, then (4.12) has an optimal solution that is integer-valued.

A number of important results are consequences of (4.13). For example, taking f_1, f_2 to be matroid rank functions, we can conclude that the intersection of two matroid polyhedra is a polyhedron with $\{0, 1\}$ -valued extreme points, and thus derive the Matroid Intersection Polyhedron Theorem. A second consequence is the Polymatroid Intersection Theorem (4.9), obtained by taking each $c_j = 1$ and observing that y^1, y^2 can be required to take a very special form.

A proof of (4.13) based on the theory of total dual integrality and total unimodularity can be found in Chapter Schrijver. In the next section we treat a generalization of (4.13).

It is worthwhile to identify the results that extend to the intersection of three (or more) polymatroids. It is possible to optimize any linear function over the intersection of three polymatroids, using the ellipsoid method, since the separation problem can be solved efficiently. However, the integrality theorem (4.13) does not generalize, and optimizing over the integral vectors in three polymatroids, even over common independent sets of three matroids, is difficult. Although we are not aware of a proof that this problem is unsolvable in the oracle context, it is well known to contain NP-hard problems.

Some Extensions

It is frequently useful in applications to relax some of the assumptions in the definition of polymatroid. A first such variant is to drop non-negativity, considering $Q(f) = \{x \in \mathbf{R}^S : x(A) \leq f(A) \text{ for all } A \subseteq S\}$. For such *submodular polyhedra* (4.3) still holds with the same proof. In addition the greedy algorithm works (for any $c \in \mathbf{R}_+^S$) with the same proof. Moreover, we do not need monotonicity for the formula (4.7) to be correct.

This greedy algorithm for $Q(f)$ motivates the definition (Lovász (1983)) of an extension of a set function. For $c \in \mathbf{R}_+^S$ let $\hat{f}(c)$ denote $\max(c \cdot x : x \in Q(f))$. It is easy to see (essentially from the proof of the greedy algorithm) that if f is submodular, $\hat{f}(c)$ can also be calculated as follows: c can be expressed (uniquely) as $\sum_{i=1}^k \lambda_i \chi^{T_i}$ for $\lambda_i > 0$ and $T_i \subseteq S$ with $T_1 \supset T_2 \supset \dots \supset T_k$. Then $\hat{f}(c) = \sum_{i=1}^k \lambda_i f(T_i)$. This can be taken as the definition of

the extension of any set function, submodular or not, to a function on \mathbf{R}_+^S . Now we can make the connection between submodularity and convexity more explicit.

(4.14) Theorem. f is submodular on S if and only if \hat{f} is convex on \mathbf{R}_+^S .

This result, from Lovász (1983), has a straightforward proof. When combined with a standard result on separation of convex and concave functions, it implies the first part of Frank's Theorem (4.10).

Another useful extension is to allow a function to take value ∞ on some of the subsets of S . With this extension essentially all of the previous results still obtain, with obvious exceptions caused by unboundedness of $P(f)$. (It is true that if $P(f)$ is bounded and non-empty, then it is a polymatroid.) This idea is often combined with another important extension, namely, requiring the submodular inequality to hold only for certain pairs of sets. We say that f is *intersecting (crossing) submodular* if $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$ whenever $A \cap B \neq \emptyset$ ($A \cap B \neq \emptyset$ and $A \cup B \neq S$). (When considering such weaker notions, we sometimes refer to ordinary submodular functions as *fully* submodular). Edmonds (1970) considered intersecting submodular functions and proved extensions of most of our earlier results. For example, the next result generalizes (4.3); the same idea underlies the proof.

(4.15) Theorem. (4.3) is true if f is intersecting submodular, with $\min_{A \subseteq S} (f(A) + u(\bar{A}))$ replaced by $\min (\Sigma f(A_i) + u(\overline{\cup A_i}) : \emptyset \neq A_i \subseteq S, A_i \text{ pairwise disjoint})$.

As for (4.3), we can conclude from (4.15) that if f is integer-valued and intersecting submodular, then the $\{0, 1\}$ -valued vectors in $P(f)$ correspond to the independent sets of a matroid. A classical example of this construction results in the forest matroid of a graph $G(V, E)$. Here we take $S = E$ and $f(A) = |V(A)| - 1$ for $A \neq \emptyset$, with $f(\emptyset) = 0$.

The following consequence of (4.15) is also useful.

(4.16) Theorem. If f is intersecting submodular on S and f' is defined by $f'(A) = \min(\Sigma f(A_i) : A = \cup A_i, \emptyset \neq A_i \subseteq S, A_i \text{ pairwise disjoint})$, then f' is submodular on S , and $Q(f') = Q(f)$.

There is also a construction that produces an intersecting submodular function beginning with a crossing submodular function; it is from Frank (1982) and also implicitly Fujishige (1984).

(4.17) Theorem. If f is crossing submodular on S with $f(S)$ finite, and f' is defined by $f'(A) = \min(\Sigma f(A_i) : \bar{A} = \cup \bar{A}_i, A_i \subseteq S, \bar{A}_i \text{ pairwise disjoint})$, then f' is intersecting submodular; moreover, $Q(f') \cap \{x \in \mathbf{R}^S : x(S) = f(S)\} = Q(f) \cap \{x \in \mathbf{R}^S : x(S) = f(S)\}$.

Notice the essential difference between (4.16) and (4.17). If f is crossing submodular, it need not be true that $Q(f) = Q(f')$; in fact, $Q(f)$ need not be a submodular polyhedron. It is the *base polyhedron* $B(f) = Q(f) \cap \{x \in \mathbf{R}^S : x(S) = f(S)\}$ that is preserved. However, we can still construct a matroid from a crossing submodular function (Frank and Tardos (1984)).

(4.18) Theorem. Let f be integer-valued and crossing submodular on S and let $k \in \mathbf{Z}_+$. Then $\{B \subseteq S : |B| = k, \chi^B \in Q(f)\}$, if non-empty, is the basis family of a matroid.

5. SUBMODULAR FLOWS AND OTHER GENERAL MODELS

In this section we describe several more general models. We give considerable attention to submodular flows, a generalization of polymatroid intersection. In particular, we describe the basic ideas behind solution algorithms. (This has not been done for polymatroid intersection.) We also report on the polymatroid matching problem, a submodular model that includes graph matching as a special case. Finally, we describe two of the many additional generalizations of matroids, Δ -matroids and greedoids.

Submodular Flows: Models and Applications

The optimal submodular flow problem is a generalization of the problem of optimizing over the intersection of two polymatroids, that keeps the important integrality and algorithmic properties of the latter problem. In addition, it contains several other fundamental problems. There are a number of closely related models, introduced under other names, and many of the important contributions have been made in these differing contexts. (Schrijver (1984) explains the connections among these models.) However, we define just two of the models, and state the main results in the language of one of them.

First we describe the *polymatroidal network flow* model of Lawler and Martel (1982); it was introduced independently by Hassin (1982). Let $G = (V, E)$ be a digraph, let s, t be distinct elements of V , and for each $v \in V$ let f_v, g_v be polymatroid functions on $\delta^-(v), \delta^+(v)$. A *feasible flow* is a vector $x = (x_j : j \in E)$ satisfying

$$\begin{aligned} x(\delta^-(v)) - x(\delta^+(v)) &= 0, \text{ for } v \in V \setminus \{s, t\}; \\ x(A) &\leq f_v(A), \text{ for all } v \in V, \text{ all } A \subseteq \delta^-(v); \\ x(A) &\leq g_v(A), \text{ for all } v \in V, \text{ all } A \subseteq \delta^+(v); \\ x_j &\geq 0, \text{ for all } j \in E. \end{aligned}$$

If $u \in \mathbf{R}_+^E$ and we take $f_v(A) = u(A)$, $g_v(A) = u(A)$, then the feasible flows are the feasible flows of an ordinary (single-source, single-sink) flow network. If we take $V = \{s, t\}$ and allow g_s, f_t to be arbitrary polymatroid functions, the feasible flows are the elements of $P(g_s) \cap P(f_t)$.

The model for which we shall present the main results is one introduced by Edmonds and Giles (1977). Let $G = (V, E)$ be a directed graph, let b be a crossing submodular function on V , and let $\ell, u, c \in \mathbf{R}^E$. We allow values of u and b to be ∞ , and we allow values of ℓ to be $-\infty$. The *optimal submodular flow problem* is

$$\begin{aligned} (5.1) \quad & \text{maximize } \Sigma(c_j x_j : j \in E) \\ & \text{subject to} \\ & x(\delta^-(A)) - x(\delta^+(A)) \leq b(A), \quad A \subseteq V; \\ & \ell_j \leq x_j \leq u_j, \quad j \in E. \end{aligned}$$

We shall use the term *feasible flow* to refer to a vector x satisfying the constraints of (5.1). The special case where b is identically 0, is the well-known optimal circulation problem of network flow theory. Let us also show that the polymatroid intersection problem (4.11) can be cast in this form. For each $j \in S$, let j_1, j_2 be copies of j , let A_i denote $\{j_i : j \in A\}$ for any $A \subseteq S$, and let $G = (V, E)$ be defined by $V = S_1 \cup S_2$ and $E = S$ with $j = (j_1, j_2)$ for each $j \in S$. Define $\ell_j = 0$, $u_j = \infty$ for each j . Define $b(A_2) = f_1(A)$ and $b((S_1 \setminus A_1) \cup S_2) = f_2(A)$ for all $A \subseteq S$, $b(A) = \infty$ for all other $A \subseteq V$. The resulting (5.1) is exactly (4.11). We encourage the reader to check that b is crossing submodular.

The main integrality result for (5.1) is due to Edmonds and Giles.

(5.2) Theorem. If ℓ, u, b are integer-valued and (5.1) has an optimal solution, then it has one that is integer-valued. If c is integer-valued and the dual of (5.1) has an optimal solution, then it has one that is integer-valued.

The proof of (5.2) uses an idea similar to that of (4.13) in Chapter Schrijver: the dual of (5.1) has an optimal solution whose non-zero components form an optimal solution to a linear program having a totally unimodular constraint matrix. Such an optimal solution is obtained from any optimal solution y by successive “uncrossings”, that is, given sets A, B with $A \cap B, A \setminus B, B \setminus A, V \setminus (A \cup B)$ all non-empty and $y_A, y_B > 0$, decreasing y_A, y_B by $\varepsilon = \min(y_A, y_B)$ and increasing $y_{A \cup B}, y_{A \cap B}$ by ε .

As yet another illustration of the power of the submodular flow model, we show how the Lucchesi-Younger dicut-covering result (Chapter Frank) can be derived from (5.2). Given $G = (V, E)$, we put $\ell_j = 0$, $c_j = -1$, and $u_j = \infty$ for each $j \in E$. For each $A \subseteq V$ such that $\delta^-(A) = \emptyset$ and $\emptyset \neq A \neq V$ we put $b(A) = -1$, and $b(A) = \infty$ for all other A . Then it is easy to check that an optimal integer-valued feasible flow is the characteristic vector of a minimum cardinality cover of directed cuts, and an optimal integer-valued solution of the dual problem picks out a collection of arc-disjoint directed cuts. Hence the Lucchesi-Younger result follows from (5.2) and the linear programming duality theorem.

Submodular Flow Algorithms

Suppose we are given $x = (x_j : j \in E)$ and want to determine whether x is a feasible flow. Then it will be enough to be able to minimize $g(A) = b(A) - x(\delta^-(A)) + x(\delta^+(A))$ over $A \subseteq V$. Since g is a (crossing) submodular function, there exists a polynomial-time (ellipsoid) algorithm to minimize it. Hence by the equivalence of separation and optimization, there exists a polynomial-time algorithm for (5.1). The resulting algorithm uses the ellipsoid method on two different levels. The search for better algorithms has succeeded in decreasing this reliance on the ellipsoid method. We shall outline an efficient combinatorial algorithm for (5.1), assuming the availability of a subroutine for minimizing a submodular function. It follows that, if an efficient combinatorial algorithm for the latter

problem is discovered, then (5.1) is also solved in a satisfactory way. In addition, there exist instances of (5.1) for which the submodular functions arising can be minimized by efficient combinatorial algorithms. An example is the problem of re-orienting the arcs of a digraph at minimum cost so as to make it k -arc-disconnected (Frank (1982)).

Notice that x is a feasible flow if and only if $\ell \leq x \leq u$ and $Bx \in Q_0(b) = Q(b) \cap \{z \in \mathbf{R}^V : z(V) = 0\}$, where B is an appropriately defined matrix. (Recall that $Q(b) = \{x \in \mathbf{R}^S : x(A) \leq b(A), A \subseteq S\}$.) The results (4.16) and (4.17) tell us that $Q_0(b) = Q_0(b')$ for some fully submodular function b' , since there is no harm in assuming that $b(V) = 0$. Hence the feasible flows remain the same when b is replaced by b' ; this is a result of Fujishige (1984). In addition the submodular oracle that is used has the same output for b and for b' . Therefore, we may pretend that b is fully submodular even if it is not. (There are two exceptions to this statement; for purposes of this exposition we ignore them.)

Submodular Flow Algorithms: Maximum Flows and Consistent BFS

The *maximum (submodular) flow problem* is to find a feasible flow that maximizes x_f for a fixed arc $f \in E$. (Notice that it is an equivalent problem to minimize x_f , since f 's direction could be reversed.) This is a special case of (5.1), and includes as special cases the ordinary network maximum flow problem (Chapter Frank), and the problem of finding a maximum component-sum vector in the intersection of two polymatroids. (The first is easy to see; the second requires modifying the previously-described submodular flow representation of polymatroid intersection.) The shortest augmenting path technique of maximum flow theory (see Chapter Frank) generalizes to this context, but we also need an important further refinement of breadth-first search.

The algorithm for the maximum flow problem generalizes the cardinality matroid intersection algorithm, as well as the usual network maximum flow algorithm. To motivate the augmentation used, we first describe two special cases. Suppose that we find a circuit in G including f and such that $x_e < u_e$ for all arcs having the same orientation as f and $x_e > \ell_e$ for all arcs having opposite orientation to f . Then x_f could be increased by sending flow around the circuit, increasing x_e by ε for arcs of the first kind and decreasing x_e by ε for arcs of the second kind; ε must not exceed $u_e - x_e$ for any arc of the first kind, or $x_e - \ell_e$ for any arc of the second kind. Next, suppose that we have a path with the same properties, say from q to p , and we attempt to send flow along the path. Now there is an additional limitation on ε ; for the new flow to satisfy the constraints of (5.9), ε cannot exceed $\varepsilon_x(p, q)$, defined to be $\min(b(A) - x(\delta^-(A)) + x(\delta^+(A)) : p \in A, q \notin A)$. This limitation could be represented as an upper bound for the flow on a fictitious arc (p, q) . An actual augmentation in the algorithm will consist of a sequence of augmentations on paths linked into a circuit by the addition of fictitious arcs, and is best described via an auxiliary digraph.

Given a feasible flow x , let $G' = G'(G, b, \ell, u, x)$ have node-set V and:

- For each $e = (p, q) \in E$ with $x_e < u_e$, an arc (p, q) with capacity $u_e - x_e$;
- For each $e = (p, q) \in E$ with $x_e > \ell_e$, an arc (q, p) with capacity $x_e - \ell_e$;
- For each $p, q \in V$ with $\varepsilon_x(p, q) > 0$, an arc (p, q) with capacity $\varepsilon_x(p, q)$.

We call the arcs *forward*, *backward*, and *jumping*, respectively. Suppose that $f = (t, s)$. A dipath P in G' from s to t together with (t, s) yields a directed circuit C in G' . Let ε be the minimum capacity of its arcs. The augmentation corresponding to P increases x_e by ε if a forward arc of C arises from e , and decreases x_e by ε if a backward arc of C arises from e . The next lemma occurs in Frank (1985); similar results for other models can be found in Fujishige (1978), Hassin (1982), Lawler and Martel (1982), and Schönsleben (1980).

(5.3) Lemma. If P is a chordless (s, t) -dipath in G' , then the augmentation corresponding to P results in a feasible flow.

On the other hand, if there is no (s, t) -dipath in G' , then x_f is maximum, and the algorithm may terminate. To see this we observe that in this case there is a set $A \subseteq V$ with $s \in A$, $t \notin A$ such that $x_e = u_e$ for all $e \in \delta^+(A)$, $x_e = \ell_e$ for all $e \in \delta^-(A) \setminus \{f\}$, and for every $p \in A$, $q \notin A$ there is a tight set containing p and not q . (A set is *tight* if its inequality in (5.1) holds with equality.) Since we may assume that b is fully submodular, it is easy to prove that the intersection and union of tight sets is tight. It follows that A is tight. Therefore,

$$x_f = b(A) - \ell(\delta^-(A) \setminus \{f\}) + u(\delta^+(A)).$$

But obviously no feasible flow can have x_f exceeding this, so x_f is maximum. The resulting min-max theorem is the following.

(5.4) Theorem. If there is a maximum flow and b is fully submodular, then

$$\begin{aligned} \max(x_f : x \text{ a feasible flow}) = \\ \min(u_f, \min(b(A) - \ell(\delta^-(A) \setminus \{f\}) + u(\delta^+(A)) : s \in A \subseteq V \setminus \{t\})). \end{aligned}$$

There is a more general version of (5.4) for crossing submodular functions. It can be derived from (5.4) by applying (4.16) and (4.17). In addition, feasibility can be tested by applying the maximum submodular flow algorithm to a certain auxiliary problem and so the following feasibility characterization is also a consequence. Again a more general version (Frank (1984)) is available.

(5.5) Theorem. If b is fully submodular then there exists a feasible flow if and only if, for all $A \subseteq V$, $\ell(\delta^+(A)) - u(\delta^-(A)) \leq b(A)$.

Now we discuss the efficiency of the algorithm. First, we observe that if b, u, ℓ and the initial x are integer-valued, then x remains integer-valued, and the algorithm terminates (assuming there exists a maximum flow). However, we would like to have a bound on the number of augmentations that is not so dependent on the size and form of the input numbers. A similar difficulty arises in ordinary network flows, where the classical solution is to find shortest augmenting paths, found by “breadth-first search”: scanning nodes in the order in which they are labelled. The analysis of that method is based on the following facts. For a flow x , let $k(x)$ denote the length of a shortest augmenting path with respect to x , and let $E(x)$ denote the set of arcs contained in some shortest augmenting path with respect to x . Suppose that an augmentation replaces flow x by flow x' . Then:

- (a) $k(x') \geq k(x)$;
- (b) If $k(x') = k(x)$, then $E(x') \subset E(x)$.

It follows from (a) that the computation is divided into at most $n = |V|$ stages and it follows from (b) that each stage takes at most n^2 augmentations. In the more general situation of submodular flows, (a) still holds but (b) fails. However, careful examination of *how* it can fail, leads to a further refinement. In addition to scanning nodes in the order labelled, we label nodes (from a node being scanned) in an order consistent with a fixed ordering of V . The resulting path has a node-sequence that is lexicographically least among node-sequences of shortest augmenting paths. This important technique was introduced independently by Schönsleben (1980) and Lawler and Martel (1982), who used it in contexts similar to the present one. Cunningham (1984) used it in his algorithm for testing membership in matroid polyhedra. He also labelled the technique “consistent breadth-first search”, and described its essential properties in a context-free way. For the submodular flow model we are using, the following result is due to Frank (1984).

(5.6) Theorem. If consistent breadth-first search is used, the maximum flow algorithm terminates after $O(n^3)$ augmentations.

Submodular Flow Algorithms: Optimization

Recall that the weighted MIA used the following three ideas: (i) Simpler optimality conditions (using weight-splitting) than those coming from a straightforward use of linear programming duality; (ii) Use of the same auxiliary digraph as the unweighted algorithm, except that arcs were assigned costs; (iii) Use of a least-cost dipath computation to update the dual solution, followed by use of the unweighted algorithm on arcs of cost zero. These three ideas will be used in extending the work of the last section to an algorithm for the optimal submodular flow problem.

The first important idea, suggested by Frank (1982), is that an optimal dual solution for (5.1) can be represented by a vector of *potentials* π_v , $v \in V$. Given such a vector π and an arc $e = (p, q) \in E$, \bar{c}_e denotes $c_e + \pi_p - \pi_q$.

(5.7) Theorem. Suppose that x is a feasible flow and π, x satisfy

- (a) If $\bar{c}_e > 0$ then $x_e = u_e$, for $e \in E$;
- (b) If $\bar{c}_e < 0$ then $x_e = \ell_e$, for $e \in E$;
- (c) If $\pi_p > \pi_q$, then $\varepsilon_x(p, q) = 0$, for $p, q \in V$.

Then x is optimal.

This result is easily proved, by showing that linear programming optimality conditions are satisfied by x and the dual solution (y, f, g) constructed from π as follows. Let $\pi_0 < \pi_1 < \dots < \pi_k$ be the distinct values of π , and let A_i denote $\{v \in V : \pi_v \geq \pi_i\}$, $1 \leq i \leq k$. Define y_A to be $\pi_i - \pi_{i-1}$ if $A = A_i$ and to be 0 otherwise. Define f_e to be $\max(\bar{c}_e, 0)$ and g_e to be $\max(-\bar{c}_e, 0)$, $e \in E$. (The dual variable f_e corresponds to the constraint $x_e \leq u_e$; g_e corresponds to $-x_e \leq -\ell_e$.) Notice that (y, f, g) is integer-valued if π is.

The algorithm maintains a feasible flow x and a potential π satisfying (5.7c). Let $f = (t, s)$ be an arc violating (5.7a); the other case is similar. It is convenient to assume that f is the only arc violating (5.7a) or (5.7b). This can be accomplished by temporarily changing the appropriate bound of any other offending arc e to x_e . Now we form the auxiliary digraph G' of the last section and assign arc-costs as follows: A forward arc (p, q) has cost $w_{pq} = -\bar{c}_e$; a backward arc (q, p) has cost $w_{qp} = \bar{c}_e$; a jumping arc (p, q) has cost $\pi_q - \pi_p$. Now each arc of G' (except the forward arc (t, s) , which we delete) has non-negative cost, and indeed, this is equivalent to the condition that (5.7abc) is violated only by f .

Next we use an $O(n^2)$ shortest-path algorithm to find a least-cost dipath in G' from s to v for each $v \in V$; let d_v be its cost. For $v \in V$ we replace π_v by $\pi_v - \min(d_v, \bar{c}_f)$. It is quite easy to check that the arc-costs remain non-negative. Now either $\bar{c}_f = 0$, in which case we have ended f 's violation of the optimality conditions, or $\bar{c}_f > 0$, in which case there exists in G' an (s, t) -dipath consisting of arcs having weight 0. In this case we use consistent breadth-first search to find such a dipath and perform an augmentation. (It is possible that this dipath yields a directed circuit of G' having no finite capacity; in this case (5.1) is unbounded, and the algorithm terminates.) That such an augmentation preserves (5.7abc) is obvious. That it delivers a feasible flow can be proved as for the weighted MIA, by showing that it is an augmentation of the maximum submodular flow algorithm applied to a more restricted submodular flow problem; see Cunningham and Frank (1985). From this observation it follows also that there will be at most $O(n^3)$ augmentations before π

changes again. The number of potential changes can be shown to be finite, and better bounds hold when c is nice. Refinements based on scaling c lead to a polynomial bound (Cunningham and Frank (1985)) and to a bound polynomial in n alone (Fujishige, Röck, and Zimmermann (1989)) for the number of augmentations.

Polymatroid Matching

An important common generalization of graph matching and matroid intersection may be formulated as a problem on *2-polymatroids*: polymatroids whose polymatroid function f is integer-valued and satisfies $f(\{j\}) \leq 2$ for each $j \in S$. A *matching* is a set $J \subseteq S$ such that $f(J) = 2|J|$. Ordinary matching on a graph $G = (V, E)$ arises when we take $S = E$ and $f(A) = |V(A)|$ for $A \subseteq S$. Matroid intersection for matroids M_1, M_2 on S arises when we take $f = r_1 + r_2$. Historically the first common generalizations of these two important problems were equivalent models called matroid parity (Lawler), matchoids (Edmonds), and matroid matching. The latter is the case of polymatroid matching in which we are given a graph $G = (V, E)$ and a matroid M on V and take $S = E$ with $f(A) = r(V(A))$. However, the maximum matching problem, even in this special case, is intractable from the oracle viewpoint (Lovász (1980b), Korte and Jensen (1982)), and also contains NP-hard problems.

Nevertheless, a great deal of progress has been made on the 2-polymatroid matching problem, mainly due to the work of Lovász. He has described (Lovász (1980b)) general reduction techniques for computing a maximum matching. For several important special classes these lead to efficient algorithms and min-max results. Among the applications are finding a maximum cardinality forest in a 3-uniform hypergraph, and finding a maximum family of openly disjoint A -paths in a graph. (See Chapter Frank for more on the latter problem).

The most important special case is that of *linear 2-polymatroids*. Here S is a set of lines (equivalently, pairs of points) in a vector space, and $f(A)$ is $ar(\cup\{e : e \in A\})$, where ar denotes affine rank. Graph matching and matroid intersection for two matroids linearly represented over the same field are both special cases, and their min-max formulas are generalized by the next result from Lovász (1980a).

(5.8) Theorem. The maximum size of a matching in a linear 2-polymatroid is

$$\min ar(A_0) + \sum_{i=1}^k \lfloor \frac{ar(A_i)}{2} \rfloor,$$

where the minimum is over sets A_0, A_1, \dots, A_k of points of the space such that, for every $e \in S$, either $ar(A_0 \cup e) \leq ar(A_0) + 1$ or, for some i , $ar(A_i \cup e) = ar(A_i)$.

Lovász has also given a (complex) polynomial-time algorithm for linear 2-polymatroid matching. Gabow and Stallmann (1986) contains a different, much more efficient algorithm. Its running time is $O(n^4)$, where $n = |S|$, surprisingly close to the best bound known for the special case of linear matroid intersection. An outstanding question is the solvability of the corresponding weighted problem, about which little is known.

Delta-matroids and Bisubmodular Polyhedra

Many of the optimizational properties of matroids are preserved in an interesting generalization introduced under various names by Dress and Havel (1986), Bouchet (1987), and Chandrasekaran and Kabadi (1988). A *delta-matroid* is a pair (S, \mathcal{F}) where S is a finite set and \mathcal{F} is a family of subsets of S , called the *feasible* sets, satisfying the following *symmetric exchange axiom*:

$$(5.9) \quad \text{If } F_1, F_2 \in \mathcal{F} \text{ and } a \in F_1 \triangle F_2, \text{ there exists } b \in F_1 \triangle F_2 \text{ such that } F_1 \triangle \{a, b\} \in \mathcal{F}.$$

(Here \triangle denotes symmetric difference.) Matroids, defined by their basis families, are precisely the delta-matroids in which the feasible sets all have the same cardinality. They form the fundamental examples, although the independent sets of a matroid also form the feasible sets of a delta-matroid. Other interesting examples are *matching* delta-matroids (S is the node-set of a graph and $F \subseteq S$ is feasible if and only if it is the set of end-nodes of some matching), and *linear* delta-matroids (where $A_{S,S}$ is a skew-symmetric (or symmetric) matrix over a field, $F \subseteq S$ is feasible if and only if $A_{F,F}$ is non-singular).

The maximum-weight feasible set problem can be solved by a type of greedy algorithm. This algorithm needs an oracle that, for disjoint subsets A, B of S , determines whether there is a feasible set F satisfying $A \subseteq F \subseteq S \setminus B$.

Symmetric Greedy Algorithm for the Maximum-Weight Feasible Set Problem

Order $S = \{e_1, e_2, \dots, e_n\}$ so that $|c_{e_1}| \geq |c_{e_2}| \geq \dots \geq |c_{e_n}|$

For each i , let R_i denote $\{e_i, \dots, e_n\}$

$J := \emptyset$

For $i = 1$ to n do

If $c_{e_i} \geq 0$ and there exists $F \in \mathcal{F}$ with $J \cup \{e_i\} \subseteq F \subseteq J \cup R_i$ then $J := J \cup \{e_i\}$

If $c_{e_i} < 0$ and there exists no $F \in \mathcal{F}$ with $J \subseteq F \subseteq J \cup R_{i+1}$ then $J := J \cup \{e_i\}$

(5.10) Theorem. (S, \mathcal{F}) is a delta-matroid if and only if for any $c \in \mathbf{R}^S$, the symmetric greedy algorithm finds a maximum-weight feasible set.

Again the “if” part is an easy consequence of the definition. The other part can be proved by methods similar to the ones used for matroids. However, it is also possible to

derive many of the results for delta-matroids from corresponding matroid facts. One useful technique is the following. (The reader should consider the effect of choosing N to be the set of negative-weight elements of S .)

(5.11) Proposition. (S, \mathcal{F}) is a delta-matroid if and only if for every $N \subseteq S$, the maximal members of $\{F \triangle N : F \in \mathcal{F}\}$ are the bases of a matroid.

We define a rank function f for a delta-matroid by $f(A, B) = \max(|F \cap A| - |F \cap B| : F \in \mathcal{F})$ for subsets A, B of S . Clearly the characteristic vector of any feasible set, and therefore any convex combination of such vectors, satisfies the inequalities

$$(5.12) \quad x(A) - x(B) \leq f(A, B), \quad A, B \subseteq S, A \cap B = \emptyset.$$

(5.13) Theorem. The convex hull of characteristic vectors of feasible sets of a delta-matroid is precisely the set of solutions of (5.12).

As for matroids, there is one property of the rank function that is the essential one for proving such results. Let f be a function defined on ordered pairs of disjoint subsets of S . We say that f is *bisubmodular* if it satisfies

$$(5.14) \quad f(A, B) + f(A', B') \geq f(A \cap A', B \cap B') + f((A \cup A') \setminus (B \cup B'), (B \cup B') \setminus (A \cup A')).$$

Kabadi and Chandrasekaran (1990), Nakamura (1988), and Qi (1988) have shown that for any bisubmodular f , the system (5.12) is totally dual integral. In particular, this implies that if f is also integer-valued, then every extreme point of the corresponding *bisubmodular polytope*, that is, the solution set $P(f)$ of (5.12), is also integer-valued. This result easily implies Theorem 5.13. The total dual integrality of (5.12) can again be proved by a greedy algorithm for maximizing $c \cdot x$ over the bisubmodular polyhedron, a generalization of the one for delta-matroids. (A. Frank has observed that it also is a special case of (5.2)). In fact, this algorithm already appears in Dunstan and Welsh (1973). For $P \subseteq \mathbf{R}^S$, $A \subseteq S$, and $x \in \mathbf{R}^A$, we say that $x \hat{\in} P$ if there exists $y \in P$ such that $y_j = x_j$ for all $j \in A$.

Symmetric Greedy Algorithm for Bisubmodular Polyhedra

Order $S = \{e_1, e_2, \dots, e_n\}$ so that $|c_{e_1}| \geq |c_{e_2}| \geq \dots \geq |c_{e_n}|$

For each i , let R_i denote $\{e_i, \dots, e_n\}$

$J := \emptyset$

For $i = 1$ to n do

If $c_{e_i} \geq 0$ choose x_{e_i} as large as possible so that $(x_{e_1}, \dots, x_{e_i}) \hat{\in} P$

If $c_{e_i} < 0$ choose x_{e_i} as small as possible so that $(x_{e_1}, \dots, x_{e_i}) \hat{\in} P$

Dunstan and Welsh call a compact set $P \subseteq \mathbf{R}^S$ *greedy* if the above version of the greedy algorithm maximizes $c \cdot x$ over P for every $c \in \mathbf{R}^S$. To describe their characterization of greedy sets, we need to define some terms. For a point $x \in \mathbf{R}^S$ and a subset $A \subseteq S$, we denote by $x \Delta A$ the point obtained by replacing x_j by its negative for each $j \in A$. For a subset P of \mathbf{R}^S , we write $P \Delta A$ to denote $\{x \Delta A : x \in P\}$. Finally, the hereditary closure $\{y \in \mathbf{R}^S : y \leq x \text{ for some } x \in P\}$ of P is denoted by $dn(P)$. The equivalence of (a) and (c) below is due to Dunstan and Welsh; that of (a) and (b) is due to Kabadi and Chandrasekaran and Nakamura. It is an interesting puzzle to understand why the recent interesting work on delta-matroids and bisubmodular polyhedra occurred so long after the initial work of Dunstan and Welsh.

(5.15) Theorem. For a polytope $P \subseteq \mathbf{R}^S$, the following statements are equivalent:

- (a) P is greedy;
- (b) P is a bisubmodular polyhedron;
- (c) For every $A \subseteq S$, $dn(P \Delta A)$ is a submodular polyhedron.

The bad news about delta-matroids and bisubmodular polyhedra is the absence of an intersection theorem. In fact, the matroid matching problem is easily seen to be the intersection of a matroid with a matching delta-matroid, so the delta-matroid intersection problem is intractable. Frank introduced a special class of bisubmodular polyhedra that is better behaved in this respect, but still includes polymatroids and base polyhedra. A *generalized polymatroid* is a bisubmodular polyhedron determined by a bisubmodular function f satisfying $f(A, B) = g(A) - h(B)$, where g is submodular, h is supermodular, and they satisfy

$$(5.16) \quad g(A) - h(B) \leq g(A \setminus B) - h(B \setminus A), \quad A, B \subseteq S.$$

Actually, the definition of Frank is slightly more general. The paper of Frank and Tardos (1988) is an excellent reference, giving a wealth of results on generalized polymatroids and related topics. One result that helps to explain their good behaviour is that every such polyhedron can be obtained by projection from a base polyhedron. The main integrality result, generalizing the Polymatroid Intersection Theorem, may be stated as follows.

(5.17) Theorem. The union of the defining systems of two generalized polymatroids in \mathbf{R}^S is totally dual integral.

Greedoids and Independence Systems

A pair (S, \mathcal{F}) , where \mathcal{F} is a family of subsets of S containing \emptyset , is a matroid if it satisfies:

$$(5.18) \quad \text{If } A \subseteq B \in \mathcal{F}, \text{ then } A \in \mathcal{F};$$

(5.19) If $A, B \in \mathcal{F}$, $|A| > |B|$, then there is $a \in A \setminus B$ with $B \cup \{a\} \in \mathcal{F}$.

Requiring only one of (5.18), (5.19) yields two different generalizations. The first, of course, is independence systems. These objects seem to be too general to have an interesting theory. We do mention one result on the effectiveness of the greedy algorithm. For any $A \subseteq S$, let $r^+(A)$ denote $\max(|F| : A \supseteq F \in \mathcal{F})$, and let $r^-(A)$ denote $\min(|F| : A \supseteq F \in \mathcal{F}, F \text{ maximal})$. Of course, an independence system is a matroid precisely when $r^+(A) = r^-(A)$ for all $A \subseteq S$. That is, the quantity $\min\left(\frac{r^-(A)}{r^+(A)} : A \subseteq S, r^+(A) > 0\right) = p(S, \mathcal{F})$ is 1 for matroids, and $p(S, \mathcal{F})$ is a measure of how close (S, \mathcal{F}) is to being a matroid. We know from Theorem 2.2 that the maximum weight independent set problem is solved optimally by the greedy algorithm when $p(S, \mathcal{F}) = 1$. Jenkyns, and independently Hausmann and Korte, have proved more generally that the greedy algorithm works well if $p(S, \mathcal{F})$ is not too small.

(5.20) Theorem. Let (S, \mathcal{F}) be an independence system. For any $c \in \mathbf{R}^S$ the greedy algorithm delivers a feasible set of weight at least $p(S, \mathcal{F}) \max(c(F) : F \in \mathcal{F})$.

Now let us consider the objects obtained when we drop (5.18) and keep (5.19). The resulting structures are called *greedoids*. They were introduced by Korte and Lovász and, while very general, have a surprising amount of structure. (In fact, enough to justify a book; see Korte, Lovász and Schrader (1990).) Here we mention a few of the connections with optimization.

For a greedoid (S, \mathcal{F}) a *basis* of (S, \mathcal{F}) is a maximal feasible set. Given a weight function c such that $c(F) \in \mathbf{R}$ for each $F \in \mathcal{F}$, we consider the problem of finding a basis F maximizing $c(F)$. We do *not* assume that c is linear, that is, determined by $c(F) = \sum(c_j : j \in F)$ for an element-weighting $c \in \mathbf{R}^S$. We state a greedy algorithm for the maximum-weight basis problem.

Greedy Algorithm for a Greedoid

$J := \emptyset$

While there exists $e \in S \setminus J$ with $J \cup \{e\} \in \mathcal{F}$ do

Choose such e with $c(J \cup \{e\})$ maximum

$J := J \cup \{e\}$

Notice that this algorithm, for the case when c is linear and (S, \mathcal{F}) is a matroid, is equivalent to the greedy algorithm for finding a matroid basis of maximum weight. In that case the algorithm finds an optimal solution. The situation for greedoids is more complicated. We illustrate two of its aspects by considering a class of greedoids arising from a digraph $G = (V, E)$ with a fixed node $s \in V$. We take $S = E$ and take \mathcal{F} to be the arc-sets of arborescences rooted at s . Notice that (5.19) is preserved under the truncation

operation $\mathcal{F}^k = \{F \in \mathcal{F} : |F| \leq k\}$. Let $T \subseteq V \setminus \{s\}$ be a set of target nodes and assign (linear) weights of 1 to arcs having head in T and 0 to the others. By finding a maximum weight basis of (S, \mathcal{F}^k) , we can decide whether there is an s -rooted arborescence having at most k edges and including all the target nodes. This latter problem is a version of the “Steiner tree problem” and is NP-hard. So optimizing linear functions over greedoid bases is, in general, difficult.

On the other hand there are interesting *nonlinear* functions that the greedy algorithm optimizes. For an example, consider again (S, \mathcal{F}) from above. Given $d \in \mathbf{R}_+^S$ and $F \in \mathcal{F}$, we let $c(F)$ denote $\sum_{v \in V(F)} (\sum_{j \in P_v} -d_j)$ where P_v is the arc-set of the unique dipath in F from s to v . Then a basis F of (S, \mathcal{F}) maximizing $c(F)$ provides least-cost dipaths (with respect to d) from s to v for all $v \in V$, and the greedy algorithm will find such a basis. (These facts are consequences of standard results on shortest path problems; see Chapter Frank.) More generally, there is a large class of functions that the greedy algorithm optimizes over the bases of any greedoid. There is also a characterization of greedoids in terms of the functions that the greedy algorithm optimizes, and a characterization of the greedoids for which it optimizes every linear function. For these results and others, we refer the reader to Korte, Lovász and Schrader (1990).

Finally, we describe a generalization of the matroid intersection theorem to a larger class of greedoids. A *distributive supermatroid* (we do not attempt to explain the name) is a greedoid (S, \mathcal{F}) together with a partial order on S satisfying

$$(5.21) \quad \text{If } A \subseteq B \in \mathcal{F} \text{ and } A \text{ is an ideal, then } A \in \mathcal{F}.$$

Notice that every matroid satisfies (5.21), by taking the partial order to be trivial. It is a result of Tardos (1987) that, for two distributive supermatroids (S, \mathcal{F}_1) , (S, \mathcal{F}_2) on the same set *with the same partial order*, the quantity $\max(|F| : F \in \mathcal{F}_1 \cap \mathcal{F}_2)$ is well-characterized. If the partial orders are not assumed to agree, then the intersection problem can be shown to be intractable in general. We state a simple version of Tardos’ min-max theorem. This is a bit misleading since it involves quantities that are difficult to compute; her paper provides a more complicated formula that does not have this drawback. For $i = 1$ and 2 and $A \subseteq S$, let $\beta_i(A)$ denote $\max(|F \cap A| : F \in \mathcal{F}_i)$. (Of course, if (S, \mathcal{F}_i) is matroid, then β_i is just the rank function.)

(5.22) Theorem. Let (S, \mathcal{F}_1) , (S, \mathcal{F}_2) be distributive supermatroids with the same partial order. Then $\max(|F| : F \in \mathcal{F}_1 \cap \mathcal{F}_2) = \min(\beta_1(A) + \beta_2(\overline{A}) : A \subseteq S)$.

6. MATROID CONNECTIVITY ALGORITHMS

Connectivity is a fundamental structural property of matroids. Algorithmically this notion took on a central role with Seymour's work on regular matroids. The connectivity-based decompositions applied there have since been further applied and significantly extended by Seymour and Truemper.

Let M be a matroid on S with rank function r . A partition $\{S_1, S_2\}$ of S is an m -separation of M for $m \geq 1$ if

$$(6.1) \quad |S_1| \geq m \leq |S_2|, \text{ and}$$

$$(6.2) \quad r(S_1) + r(S_2) - r(S) \leq m - 1.$$

Define M to be k -connected for $k \geq 2$ if M has no m -separation for $m < k$ (Tutte (1966)); in this case we say that M has *connectivity* k . 2-connected matroids are called *connected* or *nonseparable*. It is easy to see that $\{S_1, S_2\}$ is a 1-separation of M if and only if every circuit of M is either a subset of S_1 or a subset of S_2 .

The above definition is motivated by graph connectivity. Tutte proved that the polygon-matroid $M(G)$ of a graph G is k -connected if and only if G is k -connected in the usual sense (that is, connected and remains so upon the deletion of any $k - 1$ or fewer vertices), and either $|S| \leq 2k - 1$ or G has no cycle of size less than k . To obtain an exact analogue of node k -connectivity in graphs we can replace the cardinality condition (6.1) by $r(S_1) \geq m \leq r(S_2)$. This definition, however, is not invariant under duality, whereas Tutte's definition is: It is easy to show that $r(S_1) + r(S_2) - r(S) = r^*(S_1) + r^*(S_2) - r^*(S)$ for any pair $\{S_1, S_2\}$.

The following table summarizes the best algorithms available for testing k -connectivity for various values of k . Note that for general k , the complexity does grow exponentially with $|S|$, as it inevitably must since computing the connectivity of a matroid specified by an oracle is easily shown to be intractable (see the argument used for girth in Section 2).

| k | Complexity (Oracle Calls) | Method |
|----------|--------------------------------|----------------------|
| 2 | $O(S ^2)$ | Shifting |
| 3 | $O(S ^3)$ | Bridges or Shifting |
| 4 | $O(S ^{4.5} \sqrt{\log S })$ | Shifting |
| ≥ 5 | $O(S ^{k+1})$ | Matroid Intersection |

Algorithms for Testing k -Connectivity

Partial Representations

Partial representations (Truemper (1984)) will be used in several places in this section. Let X be a basis of a matroid M . The *partial representation*, abbreviated PR, of M

with respect to X is the $\{0,1\}$ -matrix R with rows indexed on elements $x \in X$ and columns indexed on elements $y \in Y = S \setminus X$ such that the (x, y) entry is 1 if and only if $x \in C(X, y)$. If M is binary, (I, R) is an actual representing matrix for M over $GF(2)$, where I is an identity matrix of appropriate dimension.

For $X' \subseteq X$ and $Y' \subseteq Y$, let $R(X', Y')$ denote the submatrix of R with row index set X' and column index set Y' . $R(X', Y')$ is then a PR of $M \setminus (Y \setminus Y') / (X \setminus X')$ with respect to $X \setminus X'$. Define the R -rank of $R(X', Y')$ by

$$rk(R(X', Y')) = r(Y' \cup (X \setminus X')) - |X \setminus X'|,$$

where r is the usual matroid rank function for M . If $rk(R(X', Y')) = |X'| = |Y'|$, then $R(X', Y')$ is called *nonsingular*.

2-Connectivity

By definition, a matroid is disconnected (separable) if there is a partition $\{S_1, S_2\}$ of S with $S_1 \neq \emptyset \neq S_2$ and $r(S_1) + r(S_2) = r(S)$ ($r(S_1) + r(S_2) \geq r(S)$ holds by submodularity). There is an easy algorithm for testing this condition using PR's.

Let X be any basis of M and let R be a corresponding PR. R has a naturally associated bipartite graph $G(R)$ in which there is one node for each row and column of R and one edge for each nonzero entry.

(6.3) Theorem. Let M be matroid, X a basis of M , R the PR corresponding to X and $G(R)$ the associated bipartite graph. Then M is connected if and only if $G(R)$ is connected. (Thus, M is connected if and only if R has no nontrivial block decomposition.)

Proof. We prove one half of the theorem. Suppose that M is not connected, let $\{S_1, S_2\}$ be a separator, and let X be a basis of M . Define $X_i = S_i \cap X$ ($i = 1, 2$). Now $X_i \subseteq S_i$ is independent and so

$$|X| = r(S) = r(S_1) + r(S_2) \geq |X_1| + |X_2| = |X|.$$

It follows that X_i is a basis of S_i ($i = 1, 2$). Let $Y_i = S_i \setminus X_i$. We conclude that every fundamental circuit $C(X, e)$ for $e \in Y_i$ is contained in S_i . That is, the submatrices $R(X_i, Y_j)$, $i \neq j$, are identically zero. It follows that $G(R)$ is not connected. \square

Theorem (6.3) yields an algorithm for testing the connectivity of M , as follows. First construct a basis and the corresponding PR R . This step requires at most $|S| + r(S)(|S| - r(S)) \leq |S|^2$ calls to an independence oracle. Now test the connectivity of the graph $G(R)$. The time for this latter computation is dominated by that for the first, and so the total is $O(|S|^2)$ oracle calls.

The relationship introduced in the last subsection between 2-connectivity and bipartite graph connectivity is the basis for an algorithmic proof of the following result. This approach is typical of several proof techniques in the subject.

Proof. Let M and N be as in the statement of the theorem. Let X' be independent and Y' coindependent in M such that $N = M \setminus Y' / X'$. Take a PR R corresponding to a basis X such that $X' \subseteq X \subseteq S \setminus Y'$. Let $e \in S(M) \setminus S(N) = X' \cup Y'$. By duality assume $e \in Y'$. Now if column e of R has a nonzero in the rows of N , we are done: $N' = M \setminus (Y' \setminus \{e\}) / X'$ satisfies the requirements of the theorem (by (6.3)). Otherwise, we find a chordless path in the bipartite graph $G(R)$ from e to any element in N . Deleting the column elements not in this path, and contracting the row elements not in the path yields the following picture (or a slight variant), where all entries not in N and not on the path are zero:

The diagram shows a square matrix partitioned into four quadrants. The top-left quadrant is a square labeled N containing a single element 1 . The top-right quadrant is a rectangle labeled e . The bottom-left quadrant is a large rectangle. The bottom-right quadrant is a small square containing a 1 . A diagonal sequence of 1 's runs from the top-left to the bottom-right, with ellipses indicating intermediate elements.

Theorem (6.4) may be applied in the following way. Suppose that we know an excluded-minor characterization for some class of matroids and ask the question, whether one of the excluded minors, when it occurs, can be isolated to a part of the ground set of a given matroid, or must be present throughout that matroid. This question leads to the notion of 1-roundedness: We say that a subfamily \mathcal{P}' of a family \mathcal{P} of connected matroids is *1-rounded* if whenever $M \in \mathcal{P}$ has a minor N isomorphic to a member of \mathcal{P}' , then for every element $e \in S(M)$, there is minor N' of M such that N' is isomorphic to a member of \mathcal{P}' and $e \in S(N')$. An example is provided by the binary matroids. In that case we take \mathcal{P} to be the class of connected non-binary matroids, and $\mathcal{P}' = \{U_{2,4}\}$, where $U_{2,4}$ denotes a matroid on 4 elements every 2 of which form a basis. By (6.4), to prove that $\{U_{2,4}\}$ is 1-rounded, we need only prove that for every connected 1-element extension of $U_{2,4}$, the new element in the extension

is an element of a minor isomorphic to $U_{2,4}$. This result together with Tutte's theorem (Tutte (1965a)) that every non-binary matroid has a minor isomorphic to $U_{2,4}$ implies a result of Bixby: If M is a connected non-binary matroid, then for every element $e \in M$, there is a minor of M isomorphic to $U_{2,4}$ and containing e . Note that the proof of (6.4) gives an algorithm to find such a minor, given any single $U_{2,4}$ minor of M . This idea is used later in computing the special 3-separations needed for testing the regularity of general matroids not known to be binary.

To use the algorithm suggested in the proof of (6.4) we need to know N in the form $N = M \setminus Y'' / X''$, for disjoint sets X'' and Y'' . In $O(|S|)$ oracle calls we can find a maximal independent subset of X'' , extend it to a maximal independent set in $X'' \cup S(N)$ and then further extend to a basis X of M . Let $X' = X \cap (X'' \cup Y'')$ and $Y' = (X'' \cup Y'') \setminus X'$. Then X' is independent, Y' is coindependent and $N = M \setminus Y' / X'$. Now constructing the partial representation R corresponding to X and finding the chordless path described in the above proof takes time $O(|S|^2)$. Finally, computing the necessary contractions and deletions is no more than the work for computing a PR for N . Thus, the complete algorithm runs in $O(|S|^2)$ oracle calls.

3-Connectivity and Bridges

Connectivity (i.e. 2-connectivity) is relatively easy to analyze. 3-connectivity is thus the first difficult instance of k -connectivity. It is particularly important in the theory of regular matroids and in decomposition theory in general. In this section we present a special-purpose recursive algorithm for testing 3-connectivity in connected matroids. Some of the ideas in this algorithm will be used later in a graph-realization algorithm.

Define the *elementary separators* of a matroid to be the minimal nonempty sets S' such that $\{S', S \setminus S'\}$ is a 1-separation. Let M be a matroid, and let D be a cocircuit of M . The elementary separators of $M \setminus D$ are called the *bridges* of D . D is called a *separating cocircuit* if it has more than one bridge. The corresponding *D -components* are the matroids $M / (S \setminus (D \cup A))$ where A is a bridge. The *A -segments* of D , for a bridge A , are the parallel classes of $M / (S \setminus (D \cup A))$ in D .

The above quantities are easily computed using PR's. Take a PR of M such that D is a fundamental cocircuit. Then D minus some single element is a row of this matrix. Deleting this row and the incident columns gives a PR for $M \setminus D$. The bridges of D can then be computed using the connectivity algorithm given by (6.3). The corresponding D -component for a particular bridge A is obtained by deleting the columns and rows not in $D \cup A$. The computation of the A -segments is then straightforward.

One more definition is required. Say that two D -bridges A_1 and A_2 *avoid* if there are A_i -segments Y_i ($i = 1, 2$) such that $Y_1 \cup Y_2 = D$. Define the *bridge graph* of D to have vertices corresponding to the bridges of D and an edge joining two vertices if and only if they *do not* avoid.

(6.5) Theorem. (*Bixby and Cunningham (1979)*) Let M be a connected matroid

that is both simple and cosimple, and let D be a cocircuit of M . Then

- (a) M has a 2-separation $\{S_1, S_2\}$ such that $D \subseteq S_2$ if and only if the matroid obtained from some D -component by identifying parallel elements has a 2-separation;
- (b) M has a 2-separation $\{S_1, S_2\}$ such that D meets both S_1 and S_2 if and only if the bridge graph of D is not connected. \square

The above theorem suggests a recursive procedure for testing 3-connectivity, given a method for finding separating cocircuits. But this latter task is easy, because of the following lemma:

(6.6) Lemma. Let M be a simple, connected matroid, and let X be a basis of M . If every fundamental cocircuit with respect to X is nonseparating, then M is 3-connected. \square

The algorithm based on the above results proceeds as follows. First construct a PR. Check its rows to see whether any is separating. If none is, the matroid is 3-connected by (6.6); otherwise, take a separating cocircuit D and compute its D -components. Compute the corresponding bridge graph. By (6.5) we can determine whether M is 3-connected by examining the connectivity of the bridge graph and determining whether the D -components are 3-connected. Since the latter matroids are smaller than the original matroid, the suggested procedure can be applied recursively. An appropriate implementation gives a bound of $|S|^3$ oracle calls.

k -Connectivity

We describe two algorithms for testing k -connectivity for general k . The first is based on an elementary shifting idea in partial representations. For 2-connectivity, this algorithm reduces to the one given by Theorem (6.3). The second algorithm uses the cardinality matroid intersection algorithm.

A Shifting Algorithm

We need an interpretation of m -separation in PR's. Let M be a matroid, and let R be a PR of M determined by a basis X . Let $\{S_1, S_2\}$ be a partition of M and define $X_i = S_i \cap X$ and $Y_i = S_i \cap Y$ ($i = 1, 2$), where $Y = S \setminus X$. Define $R_{ij} = R(X_i, Y_j)$. This situation is depicted below:

$$(6.7) \quad \begin{array}{cc} & \begin{array}{cc} Y_1 & Y_2 \end{array} \\ \begin{array}{c} X_1 \\ X_2 \end{array} & \boxed{\begin{array}{cc} R_{11} & R_{12} \\ R_{21} & R_{22} \end{array}} \end{array}$$

PR for M

Using the above notation, an equivalent definition of m -separation in terms of PR's is obtained from the computation:

$$\begin{aligned}
rk(R_{21}) + rk(R_{12}) &= r((X \setminus X_2) \cup Y_1) - |X \setminus X_2| \\
(6.8) \qquad \qquad \qquad &+ r((X \setminus X_1) \cup Y_2) - |X \setminus X_1| \\
&= r(S_1) + r(S_2) - r(S).
\end{aligned}$$

Thus, a partition of the elements S of a matroid into two sets is an m -separation if and only if its blocks are sufficiently large and, in any partial representation R , the corresponding "off diagonal" submatrices determined by this partition have total R -rank at most $m - 1$.

We now present the shifting algorithm. It is convenient to assume that M is known to be m -connected (we could first apply the algorithm for smaller values of m). Consider a basis X and the corresponding PR R , and suppose R'_{12} and R'_{21} are submatrices of R having disjoint row and column sets and total R -rank $m - 1$ ($m \geq 1$). These matrices induce a partitioning of R , as illustrated below:

| | | |
|--------|-----------|-----------|
| | Y'_1 | Y'_2 |
| X'_1 | R'_{11} | R'_{12} |
| | | |
| X'_2 | R'_{21} | R'_{22} |

PR for M

We also assume that $|X'_1 \cup Y'_1| \geq m$. The basic routine of the shifting algorithm determines whether R'_{12}, R'_{21} can be extended to a pair R_{12}, R_{21} determining an m -separation as in (6.7). (We call R_{12}, R_{21} a *legal* extension of R'_{12}, R'_{21} .) It consists of the following two operations.

Row Shifting: Given $x \in X$ such that $rk(R(X'_2 \cup \{x\}, Y'_1)) > rk(R'_{21})$, set $X'_1 = X'_1 \cup \{x\}$.

Column Shifting: Given $y \in Y$ such that $rk(R(X'_1, Y'_2 \cup \{y\})) > rk(R'_{12})$, set $Y'_1 = Y'_1 \cup \{y\}$.

It is easy to see that these operations are valid, in the sense that any legal extension R_{12}, R_{21} of R'_{12}, R'_{21} must also be an extension of the new R'_{12}, R'_{21} . Now suppose that we repeatedly apply these operations. If $rk(R'_{12}) + rk(R'_{21})$ ever increases, then we can stop; no legal extension exists. On the other hand, if $rk(R'_{12}) + rk(R'_{21}) = m - 1$, but no shifting operation is possible, then $X_1 = X'_1, X_2 = X'_2 \cup (X \setminus X'_1), Y_1 = Y'_1, Y_2 = Y'_2 \cup (Y \setminus Y'_1)$ defines a legal extension, unless $|X'_2 \cup Y'_2| < m$, in which case no legal extension exists.

Next we explain how to use the basic routine to test M for the existence of an m -separation. For matrices R_{12}, R_{21} as in (6.7) and determining an m -separation of M , there exist square non-singular submatrices P of one and Q of the other whose total R -rank is $m - 1$. Suppose that we are given P and Q , and we want to find

R_{12}, R_{21} . Let $z \in Y$ such that z does not index a column of P or of Q . We run the basic routine twice. First we initialize R'_{12} to be P and R'_{21} to be Q with column z appended, and second we initialize R'_{12} to be Q and R'_{21} to be P with column z appended. (We are taking advantage of the fact that we may assume z indexes a column of R'_{21} .) The shifting algorithm applies this procedure for all choices of P, Q .

Complexity of Shifting

For $m = 1$, testing 2-connectivity, the shifting algorithm reduces to the algorithm given by (6.3). First, since $m - 1 = 0$, the matrices P, Q must be 0×0 matrices, implying there is only one pair to consider. There is also only one case to consider for the special element z —being appended to P is now equivalent to being appended to Q . Finally, the shifting procedure is equivalent to using a standard graph algorithm to compute the component containing z in $G(R)$.

In general, the shifting algorithm can be shown to have a complexity of $O(|S|^{2m})$ oracle calls for computing m -separations. This involves an $O(|S|^2)$ implementation of the basic routine for a given pair P, Q and the observation that there are at most $O(|S|^{2m-2})$ pairs of nonsingular matrices with total R -rank $m - 1$. A defect in this approach is that, for general m , we do not know any device to decrease the estimate of the number of pairs considered. Such a device is known for the matroid intersection approach given in the next subsection.

Cunningham suggested the following idea in the special case $m = 2$. Assume M is connected and select a spanning tree of $G(R)$. Since $m - 1 = 1$, it follows that one of the matrices R_{12}, R_{21} in (6.7) must be a zero matrix, so that the other must then contain one of the elements from the spanning tree. Using this observation, we see that only $|S| - 1$ pairs P, Q must be considered, yielding an overall bound of $O(|S|^3)$ oracle calls, the same as that derived from (6.5). For the binary case, this bound can be further improved to $O(|S|^2)$ running time using a graph decomposition algorithm of Spinrad.

Finally, an improved bound is also known for testing 4-connectivity. The details are too involved to present here, but using a graph theory lemma of Szegedy, Rajan has shown how to reduce the bound to $O(|S|^{4.5} \sqrt{\log |S|})$ oracle calls.

An Algorithm Using Matroid Intersection

Let M be a matroid on S and suppose that we wish to test whether M has an m -separation for some $m \geq 1$. To do this it suffices to test for each pair of disjoint sets U_1, U_2 , both of cardinality m , whether there exists an m -separation $\{V_1, V_2\}$ such that $U_i \subseteq V_i$. For a particular choice of U_1 and U_2 , define matroids $M_1 = M/U_1 \setminus U_2$ and $M_2 = M/U_2 \setminus U_1$. These are matroids on $S' = S \setminus (U_1 \cup U_2)$, and for any partition $\{V'_1, V'_2\}$ of S' we have

$$r_1(V'_1) + r_2(V'_2) = r(V'_1 \cup U_1) - r(U_1) + r(V'_2 \cup U_2) - r(U_2).$$

Since $r(U_1) + r(U_2)$ is a constant, minimizing this quantity over partitions $\{V'_1, V'_2\}$ of S' is equivalent to minimizing $r(V_1) + r(V_2)$ over partitions $\{V_1, V_2\}$ of S such that $U_i \subseteq V_i$ ($i = 1, 2$). Thus, using Theorem (3.5) and the matroid intersection algorithm we can determine whether a given pair $\{U_1, U_2\}$ induces an m -separation. It follows that with $O(|S|^{2m})$ applications of the matroid intersection algorithm we can test $(m + 1)$ -connectivity.

This bound can be significantly improved using the following observation. Fix some set $Q \subseteq S$ with $|Q| = m$. Fix a partition $\{Q_1, Q_2\}$ of Q . There are $O(|S|^m)$ ways to complete this partition to a pair of disjoint sets $\{U_1, U_2\}$ such that $|U_1| = |U_2| = m$. Since there are 2^m partitions of Q into two sets, and this number is a constant relative to $|S|$, we see that $O(|S|^m)$ applications of matroid intersection will do. Since matroid intersection takes times $O(|S|^{2.5})$ we obtain an overall bound of $O(|S|^{m+2.5})$ oracle calls. By taking into account the similarity of the matroid intersection instances being solved, the bound can be reduced to $O(|S|^{m+2})$ oracle calls, and in the linear case, to $O(|S|^{m+2})$ total work.

Menger's Theorem for Matroids

Tutte has generalized Menger's theorem to matroids (Tutte (1965b)). Let $M = M(G)$ for a graph G , and let S be the edge-set of G . Pick two vertices u, v of G , let P and Q be the stars of these vertices and assume that $P \cap Q = \emptyset$. Menger's theorem for graphs (see Chapter Frank) asserts that the minimum number of vertices, distinct from u and v , the deletion of which separates u and v equals the maximum number of internally node-disjoint paths joining them. If G is 2-connected, this minimum can be expressed in matroid terms as

$$\min_{P \subseteq A \subseteq S \setminus Q} r(A) + r(S \setminus A) - r(S) + 1$$

To express the maximum imagine that we have found a family of m node-disjoint paths joining u and v . Deleting all edges not on these paths, and contracting all remaining edges other than those in P and Q , yields a graph in which u and v are still joined by m paths. This minor of G corresponds to a minor M' of M , and for this minor we have $m = r'(P) + r'(Q) - r'(P \cup Q) + 1$, where r' is the rank function of M' . Now it is easy to prove that for any such minor, $r'(P) + r'(Q) - r'(P \cup Q) + 1$ is no bigger than the minimum above. Tutte proved that equality can always be achieved.

(6.9) Theorem. (Tutte) Let M be a matroid on S and let P and Q be disjoint subsets of S . Let \mathcal{M}' be the family of minors of M on the element set $P \cup Q$. Then

$$\max_{M' \in \mathcal{M}'} r'(P) + r'(Q) - r'(P \cup Q) = \min_{P \subseteq A \subseteq S \setminus Q} r(A) + r(S \setminus A) - r(S)$$

Note that the quantity on the right can be computed using the matroid intersection algorithm. This fact was the basis for the connectivity algorithm in the previous

subsection. It works by defining $M_1 = M \setminus P / Q$, $M_2 = M / P \setminus Q$ and finding a maximum cardinality set J jointly independent in M_1 and M_2 . It can then be proved that for this J , the matroid $M' = M / J \setminus (S \setminus (P \cup Q \cup J))$ achieves the maximum in (6.9). This idea, due to Edmonds, yields a proof of (6.9).

7. RECOGNITION OF REPRESENTABILITY

Graph Realization for General Matroids

A matroid M is *graphic* if there is a graph G such that $S = E(G)$ and the circuits of M are exactly the edge-sets of simple cycles in G . In this case we write $M = M(G)$. *Graph realization* (GR) is the problem, given a matroid M , to determine that M is not graphic or find a graph G such that $M = M(G)$. Seymour (1981b) first solved GR. We use here a slight variation of Seymour's result due to Truemper.

(7.1) Theorem. Let M and M' be matroids on S , and let G be a graph with edge-set S . Suppose that

- (1) M and M' are connected,
- (2) M and M' have a common basis X such that the corresponding partial representations are identical,
- (3) $M' = M(G)$, and
- (4) for every node v of G , the star of v contains a cocircuit of M .

Then $M = M'$, which implies that M is graphic.

We remark that (1) is necessary. Let $S = \{a, b, c, d, e\}$ and define two matroids M and M' on S as follows: M is the direct sum of U_4^2 and U_1^1 where $S(U_1^1) = \{e\}$, and $M' = M(G)$ where G has four vertices the stars of which are $\{a, c, d\}$, $\{b, c, d\}$, $\{a, b, e\}$ and $\{e\}$. Then the pair M, M' satisfies (2)–(4) with $X = \{a, b, e\}$, but $M \neq M'$.

(7.2) Lemma. Assume that M is a connected matroid, X is a basis, C is a circuit, and $S \setminus (X \cup C) \neq \emptyset$. Then there is an element $e \in S \setminus (X \cup C)$ such that either $M \setminus e$ is connected or e is in series with some element of X . \square

(7.3) Lemma. Assume that (M, M') satisfy (2)–(4). Let $e \in S$, and assume e is a coloop of neither M nor M' .

- (a) For $e \in S \setminus X$, $(M \setminus e, M' \setminus e)$ satisfy (2)–(4) with G replaced by $G \setminus e$.
- (b) For $e \in X$, if e is parallel to no edge in G , then $(M/e, M'/e)$ satisfy (2)–(4) with G replaced by G/e and X replaced by $X \setminus \{e\}$. \square

(7.4) Lemma. Assume that (M, M') satisfy (1)–(4).

- (a) If $e \in S \setminus X$ is parallel to some element of X (in either M or M'), then (1)–(4) hold for $(M \setminus e, M' \setminus e)$ with G replaced by $G \setminus e$.

- (b) If $e \in X$ is in series with some element of $S \setminus X$ (in either M or M'), then (1)–(4) hold for $(M/e, M'/e)$ with G replaced by G/e and X replaced by $X \setminus \{e\}$. \square

Proof of (7.1). Suppose that $M \neq M'$, and that S has been chosen minimal subject to this condition. Let C be a circuit of one of M, M' , independent in the other. Then by (7.2), (7.3a), and (7.4b), $C \supseteq S \setminus X$, and so $r(S) \geq r^*(S)$. Let D be a cocircuit of one of M, M' , coindependent in the other. Then, by the dual of (7.2), and by (7.3b) and (7.4a), $D \supseteq X$, and so $r^*(S) \geq r(S)$. It follows that $r(S) = r^*(S)$, $C = S \setminus X$ and $D = X$.

Now suppose that C is a circuit of M and that D is a cocircuit of M . Then for $e \in X$, adding e to C creates no circuit in M , because of D . Hence, by the minimality of S , $C \cup \{e\}$ is a circuit in M' . But if $|X| \geq 2$, this implies X contains a circuit of M' , since M' is binary. Moreover, $|X| \neq 1$ (otherwise $M = M'$), and so we conclude that C is a circuit of M' and D is a cocircuit of M' . But then deleting D from M' , that is, G , leaves M' connected, because of C , which implies that D is the star of a node in G . On the other hand, C is a basis of M , and so D contains no cocircuit of M , contrary to (4). This proves the theorem. \square

In the next subsection we will describe two algorithms for recognizing when a binary matroid is graphic; given such an algorithm, (7.1) yields an algorithm for GR, as follows. Let M be a matroid and assume that M is connected (if not, apply the algorithm to its connected components). Construct a partial representation R for M and determine whether the associated binary matroid $M((I, R))$ is graphic. If not, stop— M is not graphic; otherwise, taking G to be a graph with representation (I, R) , use (7.1) to test whether $M = M(G)$.

The computational complexity of this algorithm may be derived as follows. The construction of a partial representation R requires $O(r(M) : S :)$ calls to an independence oracle. To determine whether (I, R) is the binary representation of a graphic matroid, we may use the algorithm of Bixby and Wagner described in the next subsection. The work for this step is bounded by $O(z\alpha(z, r(M)))$, where $\alpha(\cdot, \cdot)$ is an inverse of the Ackermann function (see (7.8)) and z is the number of nonzero entries in R . Finally, to apply (7.1) one must check the star of each node in G to see whether it contains a cocircuit. This checking requires a total of $O(r(M) : S :)$ calls to an independence oracle. Thus, the entire algorithm uses $O(r(M) : S :)$ oracle calls, plus $O(z\alpha(z, r(M)))$ other work.

In closing we mention another result that can be used to solve GR. This result generalizes a result of Tutte for binary matroids. It uses concepts defined in the discussion preceding Theorem (6.5).

(7.5) Theorem. (*Bixby*) Let M be a matroid and let D be a cocircuit of M . Then M is graphic if and only if

- (a) the D -components of M are graphic, and
- (b) the bridge graph of D is bipartite.

This theorem suggests a recursive algorithm for GR, very similar to the first algorithm discussed in the next subsection. The bound for this algorithm is $O(r(M)z)$, where z is the number of nonzero elements in some partial representation of M .

Graph Realization for Binary Matroids

Numerous algorithms have been proposed for GR on binary matroids. We describe two. The first is based on Theorem (7.5) for the case of binary matroids. Let M be a binary matroid and let R be a partial representation of M . The steps in the algorithm are the following:

- Step 1.** If (I, R) has at most 2 ones in each column, M is graphic; otherwise, choose a column having a one in each of three rows, corresponding to cocircuits D_1, D_2 and D_3 .
- Step 2.** If each of D_1, D_2 and D_3 has just one bridge, then M is not graphic (because edges of graphs can be in at most two nonseparating cocircuits, corresponding to their end vertices); otherwise, let D be a separating cocircuit. (For definitions of terms used here, see the discussion preceding (6.5).)
- Step 3.** Compute the bridge graph of D . If it is not bipartite, M is not graphic; otherwise, apply the above steps recursively to the D -components of M . M is graphic if and only if each of these is graphic.

In (Bixby and Cunningham (1980)) it is shown that this procedure can be implemented to run in time $O(r(M)z)$ where z is the number of nonzero elements in the representation (I, R) .

As our second algorithm, we describe one (Bixby and Wagner (1988)) based on an idea of Löfgren. The algorithm has the same computational complexity as an algorithm outlined by Fujishige, which is also based on Löfgren's procedure. This complexity is the best for any known GR algorithm on binary matroids. It is conjectured in (Bixby and Wagner (1988)) to be best possible.

Let G be a 2-connected graph with edge-set E . For $E' \subseteq E$, let $V(E')$ denote the set of vertices in G incident to some edge in E' . Let $\{E_1, E_2\}$ be a partition of the edge-set of G such that $V(E_1) \cap V(E_2) = \{u, v\}$. Let G_1 be the subgraph of G induced by E_1 . Define G' to be the graph obtained by interchanging in G_1 the incidences of the vertices u and v . Then G' is said to be obtainable from G by *reversing* G_1 . In general, G'' is *2-isomorphic* to G if G'' is obtainable from G by a sequence of subgraph reversals.

(7.6) Theorem. (*Whitney*) Let G and G' be 2-connected graphs on the same edge-set. Then G and G' are 2-isomorphic if and only if they have the same matroid.

□

Let R be a partial representation of a binary matroid M . Define R to be *graphic* if M is graphic. Let R_k denote the matrix made up of the first k columns of R , where rows consisting entirely of zeros have been deleted. R is called *totally nonseparable* if R_k is nonseparable for $1 \leq k \leq S : -r(M)$. For any matrix A such that $G(A)$ is connected, it is easy to compute a permutation of the columns such that the resulting matrix is totally nonseparable.

Where C_k is the fundamental circuit defined by column k of R , define $P_k = C_k \cap \{\cup_{j < k} C_j\}$. A set of edges P of a graph G is a *hypopath* of G if P is a path in some graph 2-isomorphic to G . The following statement, provable directly from (7.6), is Löfgren's "subrearrangement theorem."

(7.7) Theorem. Let R be a totally nonseparable $\{0, 1\}$ -matrix with c columns. Assume that for some $1 \leq k \leq c$, R_k is graphic with realization G_k . Then R_{k+1} is graphic if and only if P_{k+1} is a hypopath of G_k . \square

Löfgren suggested the following procedure for testing whether R is graphic: Assume R is totally nonseparable. Clearly R_1 is graphic. Suppose there exists a graph G_k that realizes R_k . Further, suppose P_{k+1} is a hypopath of G_k . Then there exists a graph G'_k 2-isomorphic to G_k such that P_{k+1} is a path in G'_k . Add the edges of $C_{k+1} \setminus P_{k+1}$ to G'_k so that they form a path between the ends of P_{k+1} but are not incident to any other vertices of G'_k . It is straightforward to verify that the resulting graph G_{k+1} is a realization of R_{k+1} . If the above procedure breaks down at any point, it follows that R is not graphic.

To implement the above idea requires a polynomial-time method for constructing G'_k from G_k . A natural approach is to invoke some representation of G_k that "displays" all graphs 2-isomorphic to G_k . For this representation we use a graph decomposition theory developed by Tutte in which a 2-connected graph is uniquely decomposed into polygons, bonds and 3-connected graphs. Using this decomposition, we can determine in polynomial time whether a given subset of edges is a hypopath.

The complexity of the algorithm is stated in the following theorem. The function $\alpha(\cdot, \cdot)$ is an inverse of the Ackermann function, and is very slow growing, being for all practical purposes never bigger than 4.

(7.8) Theorem. (*Bixby and Wagner*) Given an $r \times c$ $\{0, 1\}$ -matrix (I, R) with z nonzero entries, there is an algorithm that runs in time $O(z\alpha(z, r))$ and uses space $O(z)$ to determine whether the binary matroid $M((I, R))$ is graphic. \square

An Application to Linear Programming

A (*linear*) *network (flow) problem* is a linear programming problem $\min\{c^T x : Nx = b, x \geq 0\}$, where N is a $\{0, \pm 1\}$ -matrix with no column having two equal nonzero entries. We call a matrix with this property a *network matrix*. These are exactly the matrices that occur as submatrices of node-arc incidence matrices of digraphs.

Two matrices A and R are *projectively equivalent* if one can be obtained from the other by elementary row operations and nonzero column scaling. A linear program $\min\{c^T x : Ax = b, x \geq 0\}$ is called a *hidden network* if its constraint matrix A is projectively equivalent to a network matrix N . In this case, given explicit knowledge of N , one can easily produce an equivalent network problem with constraint matrix N . The motivation for doing so is primarily computational: It is well established that network linear programs can be solved much more efficiently than general linear programs.

The relationship between GR and hidden networks is summarized in the following theorem. In particular, it follows from this result that any polynomial-time algorithm for GR on binary matroids implies a polynomial-time algorithm for testing whether a given linear program is a hidden network.

(7.9) Theorem. (*Bixby and Cunningham (1980)*) Let $A = (I, R)$ be a real-valued matrix, where I is an identity matrix, and let $A' = (I, R')$ where R' is obtained from R by replacing nonzero entries with 1's. Then A is projectively equivalent to a network matrix if and only if the following two conditions hold:

- (a) R' is the partial representation of a graphic matroid M ; and
- (b) where G is any graph whose matroid is M , D is any orientation of G , and N is its corresponding network matrix, A is projectively equivalent to N . \square

Recognizing Total Unimodularity

A $\{0, 1\}$ -matrix is *totally unimodular* if every square submatrix has determinant ± 1 or 0. The significance of these matrices in optimization was pointed out by A. J. Hoffman and J. B. Kruskal (see Chapter Schrijver for an extensive discussion of total unimodularity), who observed that linear programming problems $\min\{c^T x : Ax = b, x \geq 0\}$ with integral b and totally unimodular constraint matrix A have integral basic feasible solutions (a simple consequence of Cramer's Rule). It is well known that network matrices are totally unimodular, and Hoffman and Kruskal asked whether other interesting classes could be found. Seymour gave an answer to this question:

(7.10) Theorem. (*Seymour (1980)*) Let $M = M(A)$ where A is totally unimodular, and assume that M is 3-connected and has no 3-separation $\{S_1, S_2\}$ with $|S_1| \geq 4 \leq |S_2|$. Then M is either graphic, cographic (the dual of a graphic matroid) or isomorphic to R_{10} . (See Chapter Seymour for a definition of R_{10} .) \square

The above theorem is perhaps best viewed as a "decomposition" result. In Chapter Seymour, the connectivity-based notions of 1-, 2- and 3-sum are defined. In terms of these sums, (7.10) can be stated as follows: Every matroid arising from a totally unimodular matrix can be constructed using 1-, 2- and 3-sums starting with only graphic and cographic matroids and copies of R_{10} . Thus, apart from the matrices

representing R_{10} , all “indecomposable” totally unimodular matrices arise from graphs, or duals of graphs. This view suggests an algorithm for testing whether a matrix is totally unimodular. More generally, it can be used to test whether a given matroid M is regular (i.e. $M = M(A)$ where A is totally unimodular).

To see that a solution to the second problem above yields a solution to the first, suppose that $M = M(I, A)$ has been shown to be regular where A is a given real matrix with $\{0, \pm 1\}$ entries, and I is an identity matrix of appropriate dimension. Let A_k denote the submatrix of A made up of the first k columns. A_1 is clearly totally unimodular. In general, assume that A_k ($k \geq 1$) is known to be totally unimodular. Find the components of the bipartite graph $G(A_k)$, and let i' and i'' be two row indices in the same component of $G(A_k)$ and such that $a_{i',k+1} \neq 0 \neq a_{i'',k+1}$. Find a shortest path from i' to i'' in $G(A_k)$. The row and column indices of this path together with the column index $k+1$ determine a square submatrix of A_{k+1} , since the path was chosen to be shortest. If this submatrix has determinant other than ± 1 or 0 , then A_{k+1} is not totally unimodular, and so neither is A ; otherwise, after checking all such pairs (i', i'') , it follows that A_{k+1} is totally unimodular. The validity of this last assertion can be deduced by showing that if (I, A') is a totally unimodular matrix with the same zero, nonzero pattern as (I, A) ((I, A') exists because M is regular), then the success of the checking procedure for A implies that A can be scaled to A' by multiplying some rows and columns of A by -1 .

We now sketch an algorithm for determining whether a general matroid is regular, using Seymour’s theorem above. The description proceeds inductively on $: S(M) :$. Thus, we assume that a matroid M is given, specified by an independence oracle, and that we can test for regularity any matroid that has a smaller number of elements.

Obviously we can test whether M is isomorphic to R_{10} , and using the results on graph realization given earlier in this section, we can test whether M is graphic or cographic. If any of these tests is positive, then M is regular and we are done; otherwise, using results of Section 6, we test whether M has a 1-separation $\{S_1, S_2\}$. If so, let $M_i = M \setminus S_i$ ($i = 1, 2$); it is easy to see that M is regular if and only if M_1 and M_2 are regular. Hence, we may assume that M is connected. Now, test for a 2-separation $\{S_1, S_2\}$. If one exists, find a circuit C of M such that $C_i = C \cap S_i \neq \emptyset$ ($i = 1, 2$). Select $e_i \in C_i$ ($i = 1, 2$) and let $M_i = M \setminus (S_i \setminus C_i) / (C_i \setminus \{e_i\})$ ($i = 1, 2$). Then both M_1 and M_2 are smaller than M , and it can be shown that M is regular if and only if M_1 and M_2 are regular.

Finally, it remains to consider the case when M is 3-connected, neither graphic nor cographic, and not R_{10} . This case is difficult because we must, in effect, consider representing M as a 3-sum, 3-sums are defined only for binary matroids, and M is not known to be binary (moreover, if M is not regular we have no way to test whether it is binary, short of determining that it is regular, since testing whether a general matroid is binary is intractable.) The method given below for dealing with this difficulty is due to Truemper (1982).

Using the connectivity algorithm based on matroid intersection (see §7), we can easily determine whether M has a 3-separation $\{S_1, S_2\}$ such that $: S_1 : \geq 4 \leq : S_2 :$.

Suppose this is the case. Let X_2 be a basis of S_2 , and extend X_2 to a basis $X = X_1 \cup X_2$ of M , where $X_1 \cap X_2 = \emptyset$. Let $Y_i = S_i \setminus X_i$ ($i = 1, 2$). Let R be the PR of M determined by X :

$$R = \begin{bmatrix} A_1 & 0 \\ D & A_2 \end{bmatrix}$$

where $A_i = R(X_i, Y_i)$ ($i = 1, 2$) and $rk(D) = 2$. Suppose that D has the structure

$$D = \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix}$$

where J_1 and J_2 are matrices of all 1's. If D does not have this structure, we take $R' = R$, $X'_2 = X_2$ and $Y'_2 = Y_2$; otherwise, since M is 3-connected, it follows that there is a shortest path in A_2 joining some row of J_1 and some row of J_2 . A *pivot* on a nonzero element r_{xy} of R is a sequence of elementary row operations (over $\text{GF}(2)$) that remove all nonzeros in column y , except r_{xy} , followed by a resetting of all entries in this column to their original values. Performing $\text{GF}(2)$ pivots on appropriate 1's along this path in A_2 we obtain a matrix

$$R' = \begin{bmatrix} A_1 & 0 \\ D' & A'_2 \end{bmatrix}$$

where $D' = R'(X'_2, Y_1)$ and $A'_2 = R'(X'_2, Y'_2)$, and where D' has some row of all 1's and $X'_2 \cup Y'_2 = S_2$. Note that if R' is not a PR of M , then M is not binary, and hence not regular.

Now define $M_1 = M/X_1$ and $M_2 = M_2 \setminus Y'_2$. Let N_1 be the binary matroid with PR (D', A_2) , and let N_2 be the binary matroid with PR A_1 . Remove loops, coloops, series and parallel elements from N_1 and N_2 . If these elements are not loops, coloops, series and parallel in M_1 and M_2 , respectively, then M is not regular; otherwise, let M'_1 and M'_2 be the corresponding reduced versions of M_1 and M_2 , respectively. Now M_1 and M_2 are smaller than M , and Truemper has proved that they are both regular if and only if M is regular.

The above procedure clearly runs in polynomial time. We will not attempt to estimate its complexity. Truemper has given a more complicated, direct algorithm with complexity $O(|S|^3)$.

Intractable Problems

In Section 2 it was shown that computing the girth of a matroid is intractable (i.e. that there is no oracle polynomial-time algorithm for this problem), and, as noted at the beginning of Section 6, the same argument applies to show that computing the connectivity of a matroid is intractable.

In this section we have given polynomial-time algorithms for a small set of “representability” questions: Testing whether a matroid is graphic, testing whether a binary matroid is graphic and testing whether a matroid is regular. The surprising fact is

that these are essentially the *only* “interesting” representability questions for which polynomial-time algorithms are *possible*. Thus, at one extreme, testing whether there exists a field over which a given matroid is representable is intractable, as is the other extreme case of testing whether a given matroid is representable over a given field.

The most basic result along these lines was proved by Seymour (1981), who showed that testing whether a matroid is binary is intractable. His proof runs as follows. Let $S = \{x_1, \dots, x_k, y_1, \dots, y_k\}$ be a $2k$ -element set, define $Y = \{y_1, \dots, y_k\}$ and define two families of subsets \mathcal{A} and \mathcal{B} of S as follows:

$$\mathcal{A} = \{\{x_i, y_i, x_j, y_j\} \subseteq S : i < j\}$$

$$\mathcal{B} = \{Z = \{z_1, \dots, z_k\} \subseteq S : z_i = x_i \text{ or } y_i, \text{ and } |Z \cap Y| \text{ is even}\}$$

Then $\mathcal{C} = \mathcal{A} \cup \mathcal{B}$ is the family of circuits of a binary matroid M on S . Now, for each $Z \in \mathcal{B}$, let M_Z denote the nonbinary matroid that is identical to M except that Z is independent (M_Z is a matroid since Z is a circuit and hyperplane of M). The existence of the matroids M_Z implies that any algorithm proving that M is binary must have made at least $|\mathcal{B}| = 2^{k-1}$ calls to the independence oracle.

8. MATROID FLOWS AND LINEAR PROGRAMMING

Maximum Flows

Let M be a matroid on S . Fix $l \in S$, and let \mathcal{C}_l be the family of circuits of M containing l . Let A be the $\{0, 1\}$ -matrix with columns indexed on elements $e \in S \setminus \{l\}$ and rows indexed on circuits $C \in \mathcal{C}_l$, such that $a_{Ce} = 1$ if and only if $e \in C$. Let \mathcal{D}_l be the family of cocircuits of M containing l . We say that M is an l -MFMC matroid, that is, has the (integral) *max-flow min-cut property* with respect to l , if for every choice of nonnegative integral vector w defined on $S \setminus \{l\}$,

$$\min_{D \in \mathcal{D}_l} w(D \setminus \{l\}) = \max(\mathbf{1}^T y : y^T A \leq w^T, y \geq 0 \text{ and integral}),$$

where $\mathbf{1}$ is a vector of 1's.

A nonnegative vector y satisfying $y^T A \leq w^T$ is called a *flow*, or *l -flow*, and $\mathbf{1}^T y$ is its *value*. A flow of maximum value is a *maximum flow*. The special element l is called the *demand* element. Define M to be a *MFMC* matroid if it is an l -MFMC matroid with respect to every choice of demand element l . (In Chapter Seymour MFMC matroids are called *free flowing*.)

(8.1) Theorem. (*Seymour (1977)*) A matroid M is a MFMC matroid if and only if it is binary and contains no F_7^* minor. \square

This theorem can be proved using “splitter theory” (Seymour (1980)), from which it follows that a 3-connected matroid M is binary and has no F_7^* minor if and only if it is either regular or isomorphic to F_7 . This structural result gives an algorithm for computing maximum flows in MFMC matroids, and for testing whether a matroid is a MFMC matroid. We describe this algorithm for computing flows.

Suppose M is not 2-connected. Then it is easy to see that M is a MFMC matroid if and only if each of its 2-connected components is. In particular, for any choice of demand element, the computation of a maximum flow for this demand element can be restricted to the component containing it.

Assume M is 2-connected but not 3-connected. Let $\{S_1, S_2\}$ be a 2-separation of M and let C be a circuit of M such that $C_i = C \cap S_i \neq \emptyset$ ($i = 1, 2$). Select $e_i \in C_i$ ($i = 1, 2$) and let $M_i = M \setminus (S_i \setminus C_i) / (C_i \setminus \{e_i\})$ ($i = 1, 2$). Then M is a MFMC matroid if and only if M_1 and M_2 are. Assuming that M_1 and M_2 satisfy this property, we may compute maximum flows for M as follows. Select an element $l \in S$ and assume $l \in S(M_1)$. Let w be an integral vector defined on S . Restricting w to S_2 , compute a maximum e_1 -flow in M_2 . Let r be the value of this flow. Now define w' on S_1 by $w' : (S_1 \setminus \{e_2\}) = w$ and $w'(e_2) = r$. Now compute the value of a maximum l -flow on M_1 with respect to w' . It is easy to prove that this is the maximum flow value for M ; moreover, it is not difficult to construct the actual flow for M from the flows for M_1 and M_2 .

Finally, suppose that M is 3-connected. Then M is either isomorphic to F_7 or regular. Both of these properties can be checked, the latter using the algorithm given in §7. To complete the description of the algorithm for finding maximum flows in MFMC matroids, we describe a method for computing maximum flows in regular matroids. This can be done in a variety of ways. The simplest is via linear programming. Let M be regular and assume K is a totally unimodular matrix such that $M = M(K)$. Then it can be proved that the optimal value of the following linear program is the value of a maximum l -flow in M :

$$\max(x_l : Kx = 0, 0 \leq x_e \leq w_e(e \neq l))$$

Since K is totally unimodular, this LP will have an integral optimal solution x . By decomposing this x into multiples of rows of A , we obtain the desired flow vector y .

The characterization of the l -MFMC matroids, for a fixed l , is very similar to that for the MFMC matroids, but the proof is much more difficult.

(8.2) Theorem. (*Seymour (1977)*) A matroid M is an l -MFMC matroid with respect to some fixed demand element l if and only if it is binary and contains no F_7^* minor containing l . \square

Let us denote the class of binary matroids that are l -MFMC matroids with respect to some fixed l by \mathcal{M}_l . Truemper (1987) has given a polynomial-time algorithm for testing membership in \mathcal{M}_l and has shown that the maximum flow problem on this class always has an integral solution by giving an algorithm for computing such a solution. We describe Truemper's algorithm for computing maximum flows on \mathcal{M}_l . To simplify the presentation, we do not concern ourselves with finding integral flows. We also do not treat membership, although the development given below can easily be modified to do so.

The problem we will actually solve is the problem of finding shortest l -paths. An l -path is a set P of the form $C \setminus \{l\}$ where $C \in \mathcal{C}_l$. Given a real-valued "length" function d defined on S , the *length* of P is $d(P)$. P is *shortest* if its length is minimum. Now, to see that we can solve maximum flow problems by computing shortest paths, note first that to solve a given maximum flow problem, it is sufficient to solve the corresponding dual:

$$(8.3) \quad \min(w^T x : Ax \geq \mathbf{1}, x \geq 0).$$

Of course, (8.3) generally has an enormous number of constraints, and it is no more obvious how to solve it than to solve the primal. Note, however, that for a given $x \geq 0$, checking $Ax \geq \mathbf{1}$ is nothing but the problem of checking whether

$$\min_{C \in \mathcal{C}_l} x(C \setminus \{l\}) \geq 1.$$

Thus, if we can compute shortest l -paths in polynomial time, then we can check for violated inequalities in (8.3) in polynomial time, and so we can solve (8.3) in polynomial time using the ellipsoid method.

Let $M \in \mathcal{M}_l$ and consider the problem of computing a shortest l -path with respect to some nonnegative weight function w defined on S . We begin by discussing several special cases.

If M is not 3-connected, this shortest path problem on M can be solved, or, more precisely, reduced to smaller shortest path problems, using exactly the same methods we used to compute maximum flows in the non-3-connected MFMC matroids.

Suppose M is either regular or isomorphic to F_7 . The F_7 case obviously presents no difficulty. For the case when M is regular, we can use the following computation, where $M^* = M(K^*)$ and K^* is totally unimodular:

$$(8.4) \quad \max(x_l : K^*x = 0, 0 \leq x_e \leq w_e (e \neq l)).$$

Since regular matroids are MFMC matroids, the optimal value in (8.4) equals the minimum weight of a circuit containing l .

Suppose that there is a triad $\{x, y, z\}$ of M , not containing l . (A *triad* is a cocircuit of cardinality 3.) In this case we make use of the following result:

(8.5) Lemma. Suppose that $M \in \mathcal{M}_l$, $\{x, y, z\}$ is a triad of M not containing l , and $\{e, f, g\}$ is disjoint from S . Then $M' \in \mathcal{M}_l$, where M' is obtained from M by creating circuits $\{e, x, y\}$, $\{f, y, z\}$ and $\{g, z, x\}$. \square

This lemma justifies the following construction. Assign weights $w_e = w_x + w_y$, $w_f = w_y + w_z$ and $w_g = w_z + w_x$ to the elements e, f and g , respectively. Let $M'' = M' \setminus \{x, y, z\}$. Now, using any shortest l -path in M'' , we easily construct a shortest l -path in M . This procedure reduces by one the number of triads not containing l .

We are now reduced to considering the case when M is 3-connected, not isomorphic to F_7 , not regular, and contains no triad missing l . In this final case, M can be appropriately decomposed, and the shortest path problem solved on it by solving appropriate problems on the smaller matroids that result from the decomposition.

Assume that $\{S_1, S_2\}$ is a 3-separation of M with the properties that : $S_1 : \geq 4 \leq S_2$: , $l \in S_2$ and $S_2 \setminus \{l\}$ spans l . Let X_2 be a basis of $S_2 \setminus \{l\}$ and extend X_2 to a basis $X = X_1 \cup X_2$ of M , where $X_1 \cap X_2 = \emptyset$. Let $Y_i = S_i \setminus X_i$ ($i = 1, 2$). Let R be the PR determined by X (in fact, R is a *representation* of M since M is binary). Let $X'_2 \subseteq X_2$ be such that $X'_2 \cup \{l\}$ is the fundamental X -circuit determined by l , and let $X''_2 = X_2 \setminus X'_2$. Then we may display R as

| | | | |
|---------|-------|---------------------------|---------|
| | | $\overbrace{\quad}^{Y_2}$ | |
| | Y_1 | l | |
| X_1 | A_1 | $\mathbf{0}$ | |
| X'_2 | D' | $\mathbf{1}$ | A'_2 |
| X''_2 | D'' | $\mathbf{0}$ | A''_2 |

where $\mathbf{1}$ and $\mathbf{0}$ are matrices of 1's and 0's, respectively, of appropriate dimensions. Now if D' is identically 0, then there must be an element in Y_2 such that the

corresponding column of R has a 1 in both A'_2 and A''_2 ; otherwise, M is not connected. By pivoting on a 1 in the intersection of such a column with A'_2 , D' becomes nonzero. We would also like to arrange that $rk(D') = 2$. Suppose this is not the case. Let \bar{X}'_2 be the subset of elements in X'_2 determined by the nonzero rows in D' , and let \bar{X}''_2 be the subset of elements in X''_2 determined by the nonzero rows in D'' , different from those in $R(\bar{X}'_2, Y_1)$. Then there is a path in $G = G(R(X_2, Y_2))$ between some element of \bar{X}'_2 and some element of \bar{X}''_2 , for in the alternative case M is not 3-connected (where T is the vertex-set of the component of G containing \bar{X}'_2 , $\{T, S \setminus T\}$ is either a 1- or 2-separation of M if $S \setminus T \neq \emptyset$). Taking a shortest such path, and pivoting on appropriate 1's in the path, results in $rk(D') \geq 2$. Now, applying essentially the same argument to A_1 , letting any nonzero row in A_1 play the role of l , it follows that we may assume R has the following form, for appropriate elements e, f, x, y and z :

| | | | | |
|-----|-------|-------------|---|----------|
| | | $y \ z \ l$ | | |
| | A_1 | | | |
| x | | 1 | 1 | 0 |
| e | D_1 | 1 | 0 | 1 |
| f | | 0 | 1 | 1 |
| | | D_2 | | A_2 |

From the above representation for M , we construct two matroids M_1 and M_2 with PR's given by R_1 and R_2 :

$$R_1 = \begin{array}{c} x \\ e \\ f \end{array} \begin{array}{|c|c|c|} \hline & y & z & l \\ \hline A_1 & & & \\ \hline & 1 & 1 & \mathbf{0} \\ \hline D_1 & 1 & 0 & 1 \\ \hline & 0 & 1 & 1 \\ \hline \end{array} \quad \text{and} \quad R_2 = \begin{array}{c} x \\ e \\ f \end{array} \begin{array}{|c|c|c|} \hline & y & z & l \\ \hline 1 & 1 & & \mathbf{0} \\ \hline 1 & 0 & 1 & \\ \hline 0 & 1 & 1 & \\ \hline D_2 & & & A_2 \\ \hline \end{array}$$

We can now state

(8.6) Theorem. Let $M \in \mathcal{M}_l$ be 3-connected. Assume that M is not regular, not isomorphic to F_7 and has no triad missing l . Then there is a 3-separation $\{S_1, S_2\}$ of M such that $|S_1| \geq 4 \leq |S_2|$, $l \in S_2$, $S_2 \setminus \{l\}$ spans l and M_1 is either isomorphic to F_7 or is regular. \square

There are now two steps left. We must show how to compute a 3-separation of the form guaranteed by (8.6), and we must show how the resulting decomposition into M_1 and M_2 can be used to compute shortest l -paths. For the computation of the 3-separation we use the following expedient algorithm. (A much more efficient, direct algorithm is given in Tseng and Truemper (1986).) For each disjoint pair $\{U_1, U_2\}$, where $l \in U_2$ and $|U_1| = 4 = |U_2|$, we apply the matroid intersection algorithm to the pair of matroids $M' = M/U_1 \setminus U_2$ and $M'' = M/U_2 \setminus U_1$ to determine the unique minimal set $A \subseteq S' = S \setminus (U_1 \cup U_2)$ that minimizes $r_{M'}(A) + r_{M''}(S' \setminus A)$.

¹ Then $\{U_1 \cup A, S \setminus (U_1 \cup A)\}$ is a 3-separation if and only if there exists some 3-separation $\{V_1, V_2\}$ such that $U_i \subseteq V_i$ ($i = 1, 2$). Now, by the minimality of the A 's, for any 3-separation $\{S_1, S_2\}$ that satisfies the conditions of Theorem (8.6), we will find some A determining a 3-separation and such that $U_1 \cup A \subseteq S_1$. It follows that this 3-separation also satisfies the conclusion of the theorem.

We are now in a position to complete the description of a method for finding a shortest l -path in M . By the computation of the previous paragraph, there is a 3-separation giving rise to matroids M_1 and M_2 satisfying the conditions of (8.6). The method for finding shortest paths is then straightforward. Set $w_e = w_f = w_l = 0$ in M_1 . Find the shortest e -, f - and l -paths in $M_1 \setminus \{f, l\}$, $M_1 \setminus \{e, l\}$ and $M_1 \setminus \{e, f\}$, respectively, and let their corresponding lengths be d_e , d_f and d_l . (These computations are possible since M_1 is either regular or isomorphic to F_7 .) Now add new elements e' , f' and g' to M_2 so that $\{e', x, y\}$, $\{f', y, z\}$ and $\{g', z, x\}$ are circuits; then delete x, y and z , denoting the resulting matroid by M'_2 . By Lemma (8.5), $M'_2 \in \mathcal{M}_i$; moreover, M'_2 has fewer elements than M , and so we may assume (by induction on $|S|$) that an algorithm is available to compute shortest l -paths in M'_2 . Set $w_i = d_i$ for $i = e', f', g'$. It is straightforward to see that a shortest l -path in M'_2 , together with the three shortest paths constructed in M_1 can be used to build a shortest l -path in M .

Oriented Matroids and Linear Programming

Matroid theory arises as a combinatorial abstraction of properties of linear dependence. From this viewpoint, oriented matroid theory arises when attention is restricted to ordered fields. Hence, it allows interpretation and generalization of ideas of real linear algebra. We indicate here how this can be done with linear programming. It is a very attractive theory, which has also produced new insights and methods in linear programming. A good reference for this subject is the monograph of Bachem and Kern (1990).

Oriented matroids are treated in Chapter Welsh. We give a definition that is useful for our purposes, and explain the relationship with the earlier definition. A *sign vector* on S is an element of $\{0, 1, -1\}^S$. To each vector in \mathbf{R}^S we associate a sign vector in the obvious way. For any vector $x \in \mathbf{R}^S$, the *support* of x is $\{j \in S : x_j \neq 0\}$. For a subspace L of \mathbf{R}^S , a vector $x \in L$ is *elementary* if $x \neq 0$ and the support of x is minimal. The supports of elementary vectors of L are the circuits of a matroid M , and those of L^\perp , the orthogonal complement of L , are the circuits of M^* . In particular, where A is a matrix with columns indexed by S , and $L = \{x : Ax = 0\}$, M is the matroid whose independent sets correspond to linearly independent sets of columns of A . One can easily check that the sets \mathcal{F} , \mathcal{F}^* of sign vectors corresponding to elementary vectors of L, L^\perp satisfy:

(8.7) The supports of members of \mathcal{F} are the circuits of a matroid M , and those of

¹To see that we can compute A with this property, note that A here is the same as the A in Theorem (3.6b), where M' plays the role of M_1 and M'' the role of M_2 . The minimal A is just the set of vertices $v \neq s$ of G such that there exists an (s, v) -dipath.

\mathcal{F}^* are the circuits of M^* ;

- (8.8) For every circuit C of M (M^*), there are exactly two elements x, y of \mathcal{F} (\mathcal{F}^*) such that C is the support of both, and $x = -y$;
- (8.9) If $x \in \mathcal{F}$, $y \in \mathcal{F}^*$, and $x_i y_i \neq 0$ for some i , then there exists $j, k \in S$ with $x_j y_j = 1$ and $x_k y_k = -1$.

For a triple $(S, \mathcal{F}, \mathcal{F}^*)$ such that $\mathcal{F}, \mathcal{F}^*$ are sets of signed vectors on S , we take (8.7)–(8.9) as the definition of a *dual pair of oriented matroids*. It is not difficult to see that (S, \mathcal{F}) determines \mathcal{F}^* , so oriented matroids do not need to be defined in dual pairs. (See Chapter Welsh.) It is also unnecessary (and a little inconvenient) to deal with *elementary* sign vectors. We have done so because the corresponding matroidal objects (circuits) are more familiar. The price we pay is that, in applications, we often have “elementary” as a qualifier when we do not really want it. In many cases, the following result can be used to get rid of it.

(8.10) Lemma. Let L be a subspace of \mathbf{R}^S and let $x \in L$, $x \neq 0$. Then there exist elementary vectors x^1, x^2, \dots, x^k of L such that $\sum_{i=1}^k x^i = x$ and $x_j^i \neq 0$ implies $x_j^i x_j > 0$. \square

As an example, suppose that we are interested in the existence of a non-negative solution of $A'x' = b$. By taking $A = (A', -b)$, this is equivalent to the existence of a non-negative solution of $Ax = 0$ with a particular component x_e positive. By (8.10) the latter is equivalent to the existence of such a solution that is also elementary. So the following oriented matroid theorem (due independently to Bland and Las Vergnas, see Bland (1977)) gives a characterization. (Notice that both situations in (8.11) cannot occur, because of (8.9).)

(8.11) Theorem. Let $(S, \mathcal{F}, \mathcal{F}^*)$ be a dual pair of oriented matroids, and let $e \in S$. Then either there exists $x \in \mathcal{F}$ with $x \geq 0$ and $x_e = 1$, or there exists $y \in \mathcal{F}^*$ with $y \geq 0$ and $y_e = 1$. \square

Applying (8.10) and (8.11), we have that $A'x' = b$ has a non-negative solution if and only if there does not exist a vector $z \geq 0$ in the row space of $(A', -b)$ that is positive in the last component, that is, if and only if there does not exist y such that $y^T A' \geq 0$, $y^T b < 0$. This is the classical Farkas Lemma; see Chapter Schrijver.

Somewhat similar techniques allow a generalization of the duality theorem of linear programming. One form of that theorem says that if the problems $\max(c^T x' : A'x' = b, x' \geq 0)$ and $\min(y^T b : y^T A' \geq c^T)$ have feasible solutions, then they have ones for which $c^T x' = y^T b$. An equivalent statement of the latter condition is that $(y^T A' - c^T)x' = 0$. We handle the objective function $c^T x'$ by adding a new variable x_f and a new equation $x_f - c^T x' = 0$ as the last equation in the system. Thus, where the components of x' and the columns of A' are indexed by elements of $S \setminus \{e, f\}$, we

get a new problem with unknown vector $x \in \mathbf{R}^S$. So the existence of x' such that $A'x' = b$ and $x' \geq 0$ is equivalent to the existence of x such that $Ax = 0$, $x_e = 1$ and $x_j \geq 0$ for $j \in S \setminus \{f\}$. The existence of a vector y such that $y^T A' \geq c^T$ is equivalent to the existence of a vector $z = (y^T, 1)A$ in the row space of A , such that $z_j \geq 0$ for $j \in S \setminus \{e\}$ and $z_f = 1$. If x , y and z have the above properties, then the condition $(y^T A' - c^T)x' = 0$ is equivalent to $x_j z_j = 0$ for $j \in S \setminus \{e, f\}$. Hence, the following result of Lawrence generalizes the duality theorem.

(8.12) Theorem. Let $(S, \mathcal{F}, \mathcal{F}^*)$ be a dual pair of oriented matroids and let $e, f \in S$ with $e \neq f$. If there exists $x \in \mathcal{F}$ with $x_e = 1$ and $x_j \geq 0$ for $j \in S \setminus \{f\}$, and there exists $z \in \mathcal{F}^*$ with $z_f = 1$ and $z_j \geq 0$ for $j \in S \setminus \{e\}$, then there exist such x, z with $x_j z_j = 0$, $j \in S \setminus \{e, f\}$. \square

Considerable effort has gone into finding constructive proofs of results such as (8.11) and (8.12). One would like to have, for example, an oriented matroid algorithm that constructs x and z of (8.12). Bland (1977) found an extension to oriented matroids of the simplex method of linear programming, although the extension is not as complete as one might hope. (More recently other algorithms for “optimizing” in oriented matroids have been introduced. See Bachem and Kern (1990) for references.)

We briefly indicate how the simplex method can be imitated. Consider a basis B of M , such that $e \notin B$ and $f \in B$. We define a tableau $(a_{ij} : i \in B, j \in S)$ determined by B as follows. For each $i \in B$, $(a_{ij} : j \in S)$ is the vector $z \in \mathcal{F}^*$ whose support is the fundamental circuit $C(S \setminus B, i)$ of M^* , and that also satisfies $z_i = 1$. It is easy to show that for each $j \in B$, the vector x defined by $x_j = -1$, $x_k = a_{kj}$ for $k \in B$, and $x_k = 0$ for $k \in (S \setminus B) \setminus \{j\}$, is in \mathcal{F} , and its support is the fundamental circuit $C(B, j)$ of M . In particular, the $z \in \mathcal{F}^*$ determined by choosing $i = f$ above, and the $x \in \mathcal{F}$ determined by choosing $j = e$, satisfy the condition $x_j z_j = 0$ for all $j \in S \setminus \{e, f\}$ of (8.12). If this x satisfies $x_j \geq 0$, $j \in S \setminus \{f\}$, then the tableau is *primal feasible*; if this z satisfies $z_j \geq 0$, $j \in S \setminus \{e\}$, it is *dual feasible*.

A strengthening of (8.12) is that (under the same hypotheses) there exists a tableau that is both primal and dual feasible. A primal simplex method is one that begins from a primal feasible tableau and performs pivot operations, replacing B by a basis $B' = (B \cup \{j\}) \setminus \{k\}$ where $j \in S \setminus (B \cup \{e\})$ violates dual feasibility. There is a simple rule in ordinary linear programming for choosing k , but it is based on numerical comparisons, and so is unavailable in the oriented matroid setting. However, it can be proved that there does exist a choice for k , if the hypotheses of (8.12) are satisfied. A more serious difficulty is that cycling (repeating a basis) can occur in a different way from that in ordinary linear programming, making rules for finite termination considerably more complicated.

Acknowledgment. We are grateful to Kazuo Murota and András Sebő for their comments on earlier versions of this paper.

References

BACHEM, A. and W. KERN

- [1990] *Oriented Matroids from a Polyhedral Point of View*, to appear.

BIXBY, R.E.

- [1974] ℓ -matrices and a characterization of binary matroids, *Discrete Math.* 8, 139-145.
[1982] Matroids and operations research, in: *Advanced Techniques in the Practise of Operations Research*, edited by H.J. Greenberg, F.H. Murphy, and S.H. Shaw (North-Holland, New York) pp. 333-458.

BIXBY, R.E. and W.H. CUNNINGHAM

- [1979] Matroids, graphs and 3-connectivity, in: *Graph Theory and Related Topics*, edited by J.A. Bondy and U.S.R. Murty (Academic Press, N.Y.) pp. 91-103.
[1980] Converting linear programs to network problems, *Math. of Opns. Res.* 5, 321-357.

BIXBY, R.E., W.H. CUNNINGHAM and D.M. TOPKIS

- [1985] The poset of a polymatroid extreme point, *Math. of Opns. Res.* 10, 367-378.

BIXBY, R.E. and A. RAJAN

- [1988] A short proof of the Truemper-Tseng theorem on max-flow min-cut matroids, *Linear Algebra and Appl.*, 114/115, 277-292.

BIXBY, R.E. and D.K. WAGNER

- [1988] An almost linear-time algorithm for graph realization, *Math. of Opns. Res.* 13, 99-123.

BLAND, R.G.

- [1977] A combinatorial abstraction of linear programming, *J. Combinatorial Theory B* 23, 33-57.

BOUCHET, A.

- [1987] Greedy algorithm and symmetric matroids, *Math. Programming* 38, 147-159.

BREZOVEC, C., G. CORNUÉJOLS and F. GLOVER

- [1988] A matroid algorithm and its application to the solution of two optimization problems on graphs, *Math. Programming* 42, 471-488.

CHANDRASEKARAN, R., and S.N. KABADI

- [1988] Pseudomatroids, *Discrete Math.* 71, 205-217.

CUNNINGHAM, W.H.

- [1984] Testing membership in matroid polyhedra, J. Combinatorial Theory B 36, 161-188.
- [1985] On submodular function minimization, Combinatorica 5, 185-192.
- [1986] Improved bounds for matroid partition and intersection algorithms, SIAM J. Comp. 15, 948-957.

CUNNINGHAM, W.H. and A. FRANK

- [1985] A primal dual algorithm for submodular flows, Math. of Opns. Res. 10, 251-262.

DUNSTAN, F.D.J. and D.J.A. WELSH

- [1973] A greedy algorithm for solving a certain class of linear programmes, Math. Programming 5, 338-353.

EDMONDS, J.

- [1970] Submodular functions, matroids, and certain polyhedra, in: *Combinatorial Structures and their Applications*, edited by R.K. Guy, H. Hanani, N. Sauer, J. Schönheim, (Gordon and Breach, New York) pp. 69-87.
- [1979] Matroid intersection, Ann. Discrete Math. 4, 39-49.

EDMONDS, J. and R. GILES

- [1977] A min max relation for submodular functions on directed graphs, Ann. Discrete Math 1, 185-204.

FRANK, A.

- [1981] A weighted matroid intersection algorithm, J. Algorithms 2, 328-336.
- [1982] An algorithm for submodular functions on graphs, Ann. Discrete Math. 16, 97-120. and
- [1984] Finding feasible vectors of Edmonds-Giles polyhedra, J. Combinatorial Theory B 36, 221-239.

FRANK, A. and E. TARDOS

- [1984] Matroids from crossing families, in: *Finite and Infinite Sets, Eger 1981* edited by A. Hajnal, (North-Holland, Amsterdam), pp. 295-304.
- [1988] Generalized polymatroids and submodular flows, Math. Programming 42, 489-563.

FUGISHIGE, S.

- [1978] Algorithms for the independent flow problems, J. Oper. Res. Soc. Japan 21, 189-203.
- [1984] Structures of polytopes determined by submodular functions on crossing families, Math. Programming 29, 125-141.

FUGISHIGE, S., H. RÖCK and U. ZIMMERMANN

- [1989] A strongly polynomial algorithm for minimum cost submodular flow problems, Math. of Opns. Res., 14, 60-69.

GABOW, H. and M. STALLMANN

- [1985] Efficient algorithms for graphic matroid intersection and parity, extended abstract, in: *Automata, Languages and Programming* (Springer, Berlin).
- [1986] An augmenting path algorithm for the linear matroid parity problem, *Combinatorica* 6, 123-150.

GABOW, H. and R.E. TARJAN

- [1984] Efficient algorithms for a family of matroid intersection problems, *J. Algorithms* 5, 80-131.

GABOW, H. and Y. XU

- [1989] Efficient theoretic and practical algorithms for linear matroid intersection problems, Technical Report CU-CS-424-89, University of Colorado.

GRÖTSCHEL, M., L. LOVÁSZ and A. SCHRIJVER

- [1981] The ellipsoid method and its consequences in combinatorial optimization, *Combinatorica* 1, 169-197.

HASSIN, R.

- [1982] Minimum cost flow with set constraints, *Networks* 12, 1-21.

KABADI, S. and R. CHANDRASEKARAN

- [1990] On totally dual integral systems, *Discrete Applied Math.* 26, 87-104.

KORTE, B. and P. JENSEN

- [1982] Complexity of matroid property algorithms, *SIAM J. Computing* 11, 184-190.

KORTE, B., L. LOVÁSZ, and R. Schrader

- [1990] *Greedoids* (Springer, Berlin).

LAWLER, E.L. and C. MARTEL

- [1982] Finding maximal polymatroidal network flows, Math. of Opns. Res. 7, 334-347.

LAWLER, E.L.

- [1975] Matroid intersection algorithms, *Math. Programming* 9, 31-56.

LEHMAN, A.

- [1964] A solution of the Shannon switching game, *SIAM J. Appl. Math.* 12, 687-725.

LOVÁSZ, L.

- [1980a] Selecting independent lines from a family of lines in a space, *Acta Sci. Univ. Szeged* 42, 121-131.
- [1980b] Matroid matching and some applications, *J. Combinatorial Theory B* 28, 208-236.
- [1983] Submodular functions and convexity, in: *Mathematical Programming: The State of the Art*, edited by A. Bachem, M. Grötschel and B. Korte (Springer, Berlin) pp. 235-257.

MIRSKY, L.

- [1971] *Transversal Theory*, (Academic Press, New York).

QI, L.

- [1988] Directed submodularity, ditroids, and directed submodular flows, *Math. Programming* 42, 579-599.

SCHÖNSLEBEN, P

- [1980] *Ganzzahlige Polymatroid-Intersektions-Algorithmen*, thesis, ETH Zürich.

SCHRIJVER, A.

- [1978] *Matroids and Linking Systems*, (Mathematisch Centrum, Amsterdam).
- [1984] Total dual integrality from directed graphs, crossing families and sub- and supermodular functions, in: *Progress in Combinatorial Optimization*, edited by W.R. Pulleyblank (Academic Press, New York) pp. 315-362.

SEYMOUR, P.D.

- [1977a] A note on the production of matroid minors, *J. Combinatorial Theory B* 22, 289-295.
- [1977b] The matroids with the max-flow mincut property, *J. Combinatorial Theory B* 23, 189-222.
- [1980] Decomposition of regular matroids, *J. Combinatorial Theory B* 28, 305-359.
- [1981a] Matroids and multicommodity flows, *European J. Combinatorics* 2, 257-290.
- [1981b] Recognizing graphic matroids, *Combinatorica* 1, 75-78.

TARDOS, E.

- [1987] An intersection theorem for supermatroids, preprint.

TRUEMPER, K.

- [1982] On the efficiency of representability tests for matroids, *European J. Combinatorics* 3, 275-291.
- [1984] Partial matroid representations, *European J. Combinatorics* 5, 377-394.

- [1985] A decomposition theory for matroids, I: General results, J. Combinatorial Theory B 39, 43-76.
 - [1986a] A decomposition theory for matroids, III: Decomposition conditions, J. Combinatorial Theory B 41, 275-305.
 - [1986b] Max-flow min-cut matroids: Polynomial testing and polynomial algorithms for maximum flow and shortest routes, Math. of Opns, Res. 12, 72-96.
- TSENG, F.T. and K. TRUEMPER
- [1986] A decomposition of matroids with the max-flow min-cut property, Discrete Appl. Math. 15, 329-364.
- TUTTE, W.T.
- [1965a] Lectures on matroids, J. Res. Nat. Bur. Stand B 69, 1-47.
 - [1965b] Menger's theorem for matroids, J. Res. Nat. Bur. Stand. B 69, 49-53.
 - [1966] Connectivity in matroids, Canad. J. Math. 18, 1301-1324.
- WELSH, D.J.A.
- [1976] *Matroid Theory* (Academic Press, New York).