plexity.

uting,

In:

r and

, pp. 85 -

in NQL.

45.

ON DETERMINANTS, MATCHINGS, AND RANDOM ALGORITHMS

by L. Lovász*

1. Introduction

In the attempts to classify problems according to their algorithmic complexity, probably most attention has been given to those problems on the borderline of "polynomial" and "exponential" complexity. These are the problems in the class NP, and can be described as follows: given an input X, we are looking for another structure Y which "fits" X; the fact that Y "fits" X can be checked in no more time than $|X|^{const}$ (this in particular contains that Y is not larger than $|X|^{const}$). There is a trivial exponential time algorithm to solve such problems (looking at all possible Y's). In some cases, highly non-trivial algorithms are available which solve the problem in polynomial time; the class of problems in NP solvable in polynomial time is denoted by P.

The conjecture that $P \neq NP$ is one of the most outstanding open problems of contemporary mathematics.

If we have a problem in NP, about which we suspect that it might be in P, it may be very difficult to find an algorithm right away. Therefore it is of great importance to find classes of problems which are between P and NP; then to show that a given problem is in such a class is a realistic first step toward the complete solution.

One rather well-known relaxation of P is the class of well-characterized problems. If a problem is in NP then there does not seem to be any reason for its negation also being in NP (of course, negative results like this cannot be proved at the present stage of our knowledge). If the negation of the problem in NP can also be formulated as a problem in NP then we say that the problem is well-characterized. We denote the class of these problems by Δ. So $P \subseteq \Delta \subseteq NP$. The second inclusion is quite certainly strict. There is not any well-founded conjecture about the first, since problems which are shown to be in Δ (theorems of this kind are often among the most beautiful and deep results in combinatorics do tend to eventually find their "complete solution (i.e., a polynomial-bounded algorithm). Another relaxation of a polynomial-bounded algorithm is an algorithm which includes random steps and may make errors, but with small probability only. The main purpose of this paper is to call attention to the class of problems solvable in this sense. We shall denote this class by RP; exact definition will be given later. It will turn out that even

---

* Bolyai Institute, Jozsef Attila University, H-6720 Szeged, Hungary; this paper was written while the author was visiting the University of Waterloo and the Massachusetts Institute of Technology.

if a problem is in P, algorithms involving random steps may solve it much faster than deterministic algorithms.

In chapter 1, we discuss the matching problem, and a recent generalization called the matroid matching problem, from the point of view of their belonging to Δ, RP and P. We hope this will also illuminate the notion of these classes. In Chapter 2, we state a further generalization of these problems, define RP precisely, and point out its relationship with the so-called "probabilistic method". I am indebted to Jack Edmonds and Gary Miller for stimulating discussions on these problems.

2. Matching and Matroid Matching

Given a graph G, a set of edges of G is called a matching if no two of them have a vertex in common. A matching is perfect if it covers all the vertices.

The matching problem, in its simplest version, calls for the decision whether or not a given graph G has a perfect matching. Tutte [14] gave a necessary and sufficient condition for the existance of a perfect matching. Edmonds [3] found a polynomial-bounded algorithm to solve the matching problem.

Tutte's original proof used methods of linear algebra, which has since then been simplified so that now several purely combinatorical proofs of this fundamental theorem are available. However, his approach contains an idea which will be important for the purposes of this paper. This can be formulated as a necessary and sufficient condition.

THEOREM 1. Let G be a simple graph. Orient its edges arbitrarily. For each edge $e \in E(G)$, let x_e be an indeterminate. Form the matrix $B = (B_{ij})$ where

$$B_{ij} = \begin{cases} x_e & \text{if } e = (i, j) \\ -x_e & \text{if } e = (j, i) \\ 0 & \text{otherwise} \end{cases}$$

Then G has a perfect matching iff det B is not identically 0 in the variables x_e.

For sake of comparison, let us quote Tutte's main theorem:

THEOREM 2. A graph G has a perfect matching iff for every set $X \subseteq V(a)$, the graph G-X has at most $|X|$ odd connected components.

Let us compare the logical structure of the two conditions. To illuminate the ideas involved, suppose you are writing a book on graph theory and having introduced the notion of a perfect matching you want to put in two figures, illustrating graphs with and without perfect matchings. In the figure depicting a graph with a perfect matching, you could draw heavy lines for the edges of a perfect matching and so the reader will be immediately convinced that this graph has a perfect matching. You would like to give a one-line reasoning showing that the other example has no perfect matching. The definition of a perfect matching does not suggest anything of this sort.

However, if you apply Theorem 2 you can encircle the vertices of a set X violating the condition and just write that "the deletion of the 17 encircled points results in 19 odd components."

In more usual language, this says that Theorem 2 provides a good characterization of graphs with a perfect matching, or that the property of having a perfect matching, as well as its negation are in NP.

Note that the condition in Theorem 1 does not yield a good characterization, since to prove that there is no perfect matching in the graph, i.e., that det B is identically 0, we have to evaluate det B at infinitely many places. (The usual way to check if a polynomial is identically 0, i.e., to expand it into a sum of mononomials, is not promising here: the expansion of the determinant may lead to exponentially many terms. Determinants are easily evaluated if their entries are numbers, but not if they are functions).

On the other hand, the condition given by Theorem 1 does have some virtues. Note that det B is a polynomial in the variables x_e. Therefore if it does not vanish identically, then the set of m-tuples (x_e) $(m = |E(G)|)$ for which it vanishes is of measure 0. So if we generate an m-tuple $(\bar{x}_e)$ at random (say $\bar{x}_e \in [0,1]$ with uniform distribution, independently of each other) and evaluate det $B(..\bar{x}_e...)$, then with probability 1 we get 0 only if det B is identically 0. More precisely, if det B is identically 0 then, of course, we get 0. If det B is not identically 0 then the probability that we get 0 is 0.

In practice, we cannot generate a random real number and we cannot perform computations with infinite decimals. But we can pick a number N and choose integers $\bar{x}_e \in \{1,...,N\}$ at random. Then computing det $B(x_e)$ we are in the following situation. If det B is identically 0, then of course our result is 0. If det B is not identically 0, then the probability of obtaining 0 is very small if N is moderately large. In fact, results of Zippel [17] imply that the probability of error is less than m/N.

So Theorem 1 provides an algorithm to decide whether or not a perfect matching exists which runs in polynomial time, and gives the right answer with probability arbitrarily close to 1. For practical reasons, such an algorithm is as good as a really deterministic one!

Suppose now that we have selected random integers x_1, computed det B and found that it is non-zero. How can we actually find a perfect matching? We can answer this question after the proof of Theorem 3.

In spite of the fact that the condition in Theorem 1 yields an algorithm where the probability of error is negligible, it is important from the theoretical point of view that an efficient algorithm <u>always</u> solving this problem does exist. (Edmonds [3]). This algorithm gives more insight into the structure of the problem, among others it provides a proof of Theorem 2, and also applied to various extentions, most notably the weighted case.

Also, there is a slight inaccuracy in the interpretation of our probabilistic consideration. Let us assume that the input data, i.e., the graphs G have some distribution; let p be the probability of the event A that G has a perfect matching. In general, it is very difficult to know anything about p, or even to make reasonable assumptions.

We can design now an algorithm (with random steps) which answers "no" if G has no perfect matching, answers yes or no if it has but the probability that it answers no even though the n paits of vectors do exist is q. Let E denote the event that our algorithm concludes "yes". Then $P(A) = p$, $P(\overline{E}|A) = q$, $E \subseteq A$. Hence, by simple computation

$$P(A|\overline{E}) = \frac{pq}{1 - p + pq} .$$

So if we have generated a random $x_1,\ldots,x_m$, computed det B and found that it is 0, the probability of det B being not identically 0 is not q but $pq/(1-p+pq)$. Since we do not know p, we do not know how small q has to be to make this probability small.

A generalization of the matching problem, called the matchoid problem (Edmonds, Jenkyns [6]) and the matroid parity problem (Lawler [9]), is the following. Let $(a_1, b_1),\ldots, (a_m, b_m)$ be disjoint pairs of elements of a matroid. Are there n pairs among them whose union is independent?

This problem is exponentially difficult for general matroids (Korte [7], Lovász [11]) but is solvable polynomially if the matroid involved is representable over a field (Lovász [11]). For the purposes of this paper, we assume that $a_1,\ldots,a_m$, b_1, $\ldots,b_m$ are real vectors. We also make the less restrictive assumption that $a_1,\ldots,a_m$, $b_1,\ldots,b_m \in R^{2n}$.

First we formulate a condition analogous to Theorem 1. Let $a = (d_1,\ldots,d_n)^T$ and $b = (\beta_1,\ldots,\beta_n)^T$ be two real vectors. We define their wedge product $a \wedge b$ as the skew symmetric n x n matrix

$$(a \wedge b)_{ij} = \alpha_i\beta_j - \alpha_j\beta_i .$$

THEOREM 3. Let $a_i, b_i \in R^{2n}$ $(1 \leq i \leq m)$. Then there exist n pairs (a_i, b_i) whose union is a basis iff

$$\det (x_1(a_1 \wedge b_1) + \ldots + x_m(a_m \wedge b_m)) \qquad (1)$$

is not identically 0 in the variables $x_1,\ldots, x_m$.

For sake of comparison, let us quote the necessary and sufficient condition given in [10]:

THEOREM 4. Let $a_i, b_i \in R^{2m}$ $(1 \leq i \leq m)$. Then there exist n pairs (a_i, b_i) whose union is a basis iff for every linear mapping $A: R^{2n} \to R^{2n}$ and every partition $\{1, \ldots, n\} = I_1 \cup \ldots \cup I_k$, the following is satisfied:

$$\sum_{j=1}^{k} \left\lfloor \frac{1}{2} \dim \langle Aa_i, Ab_i : i \in I_j \rangle \right\rfloor \geq r(A).$$

It is clear that Theorems 3 and 4 have the same logical structure as Theorems 1 and 2. Again, the problem arises that supposing we have found integers $x_1, \ldots, x_m$ such that (1) is not 0, can we construct n pairs (a_i, b_i) whose union is linearly independent? This question will be answered in the affirmative after the proof of Theorem 3.

There is a "proper" polynomial-bounded algorithm to solve this problem [11]. This algorithm is, however, very complicated and certainly not suitable for practical use in its present form.

Proof of Theorem 3.

I. Suppose that e.g., the vectors $a_1, \ldots, a_n, b_1, \ldots, b_n$ are linearly independent. Choose $x_1 = \ldots x_n = 1$, $x_{n+1} = \ldots = x_m = 0$. Then

$$\det \sum_{i=1}^{m} x_i (a_i \wedge b_i) = \det \sum_{i=1}^{n} (a_i \wedge b_i) \neq 0,$$

so (1) is not identically 0.

II. Suppose that (1) is not identically 0. Denote the matrix $\sum x_i (a_i \wedge b_i)$ by $B = B(x_1, \ldots, x_m)$. It is well known that

$$\det B = (\mathrm{pf}\ B)^2$$

where pf B is the pfaffian of B.

We claim first that pf B is linear in each of the variables x_i. Consider e.g., i = 1. Without loss of generality we may assume that $a_1 = (1, 0, \ldots, 0)^T$ and $b_1 = (0, 1, 0, \ldots, 0)^T$. Then

$$B = \begin{vmatrix} 0 & x_1 & & \\ -x_1 & 0 & & 0 \\ & & & \\ & 0 & & 0 \end{vmatrix} + B' ,$$

where B' does not depend on x_1. Hence the fact that pf B is linear in x_1 follows by the definition of pfaffians.

Consider a monomial term in the polynomial pf B with non-zero coefficient. This is the product of distinct variables x_i. Without loss of generality we may assume that it is $x_1 \ldots x_n$. So if we substitute $x_1 = \ldots = x_n = 1$, $x_{n+1} = \ldots = x_m = 0$, the value of pf B will be non-zero. Hence det $B \neq 0$. We claim that this implies that $a_1,\ldots,a_n,b_1,\ldots b_n$ are linearly independent. Suppose not, then they are all contained in a $(2n-1)$-dimensional subspace; without loss of generality we may assume that their last coordinates are 0. But then the last row of B is 0, so det $B = 0$, a contradiction.

Now we are also able to answer the question: supposing we have found integers $\bar{x}_1,\ldots,\bar{x}_m$ such that det $B(\bar{x}_1,\ldots,\bar{x}_m) \neq 0$, how can we select n pairs $(a_i b_i)$ whose union is linearly independent? Let, say, $\bar{x}_1,\ldots,\bar{x}_p \neq 0$, $\bar{x}_{p+1} = \ldots = \bar{x}_m = 0$. By the argument in the previous proof it follows that we must have $p \geq n$ and if $p = n$ then $a_1,\ldots,a_n$, $b_1,\ldots,b_n$ are linearly independent. So suppose that $p > n$. We show that we can replace one of $\bar{x}_1,\ldots,\bar{x}_p$ by 0 and still have a non-zero determinant. For set

$$\alpha_i = \text{pf } B(\bar{x}_1,\ldots,\bar{x}_{i-1}, 0, \bar{x}_{i+1},\ldots\bar{x}_p, 0, \ldots, 0)\,(1 \leq i \leq p).$$

Then, using the fact that each term in the polynomial expansion of pf $B(x_1,\ldots,x_m)$ is the product of n distinct variables, we get

$$\sum_{i=1}^{p} \alpha_i = (p-n) \text{ pf } B(x_1,\ldots,x_m) \neq 0$$

and so at least one α_i is non-zero. Since each α_i is easily computable, we can find this non-zero α_i by computing at most m determinants.

3. Determinants and Other Generalizations

Another version of the idea of Theorems 1 and 3 occur in the paper [4] of Edmonds:

THEOREM 5. Let G be a simple bipartite graph with bipartition $V(G) = U \cup W$, $U = \{u_1,\ldots,u_n\}$, $W = \{w_1,\ldots,w_n\}$, and $E(G) = \{e_1,\ldots,e_m\}$. Let $x_1,\ldots,x_m$ be indeterminates and define

$$a_{ij} = \begin{cases} x_k \text{ if } (u_i, v_j) = e_k, \\ 0, \text{ if } (u_i, v_j) \notin E(G), \end{cases}$$

and let $A = A(x) = (a_{ij})$. Then

$$\max \{r(A(x)): \ x \in R^m\} \qquad (2)$$

is the maximum number of edges of a matching in G.

Again, we may note that the maximum in (2) is attained if the x_i are algebraically independent transcendentals (although this is of little help from the point of view of computation). Theorem 5 can be applied in deriving König's Theorem.

Edmonds proposed the following general problem. Let $l_{ij}(x_1,\ldots,x_k)$ $(1 \leq i \leq n, 1 \leq j \leq m)$ be linear forms with integral coefficients, and set

$$L = L(x_1,\ldots,x_k) = (l_{ij}(x_1,\ldots,x_k)).$$

Define

$$r_o(L) = \max \{r(L(x_1,\ldots,x_k): \ x_i \in R\} .$$

Problem: Compute $r_o(L)$.

Theoretically, we can determine this number by substituting algebraically independent transcendentals for $x_1,\ldots,x_k$, and compute the rank of the resulting matrix. In practice, we may generate random numbers $x_1,\ldots,x_k$, substitute them for $x_1,\ldots,x_k$, and compute the rank of the resulting matrix; the probability of error can be made arbitrarily small. However, no algorithm is known to efficiently compute this maximum rank.

We note that the problem is equivalent to the special case in which we only want to know if $r_o(L) = m$, the number of columns. For suppose we can do this efficiently. Let L be any matrix of linear forms. We select $r_o(L)$ columns of L by the following procedure: Suppose rows $w_1,\ldots,w_k$ have been selected. For each of the other columns, apply the hypothesized algorithm to check if $r_o(w_{ij},\ldots,w_k,w) = k + 1$. If you find an w for which this holds, we label it w_{k+1}. If not, then $k = r_o(L)$. Next note that we can further reduce the problem to the special case when $n = m$. We may clearly assume that $n \geq m$ since other trivially $r_o(L) \neq n$. Now append n-m new columns, whose entries are distinct new variables. It is clear that the resulting matrix L' has $r_o(L') = n$ iff $r_o(L) = m$. If L is a square matrix then $r_o(L) = n$ is equivalent to saying that det L is not identically 0. Theorems 1, 3 and (after appropriate reductions) Theorem 5 represent matrices L for which the question if det L is identically 0 can be decided efficiently. Whether or not such an algorithm exists for general L, remains open. We conclude with a further example where a very important problem is reduced to finding $r_o(L)$ for an appropriate matrix L of linear forms, and it is still not completely solved (Yemini and Cohen [16]). For another connection between rigidity and matroid matching, see [12].

Let G be a graph on at least d + 1 points. Let us place the vertices of G

"in general position" in the d-dimensional euclidian space (say choose their coordinates algebraically independent). Consider the edges as rigid bars. Will the resulting structure be rigid?

Because of the "general" position of the points, the answer to this question depends on the combinatorial structure of G only.

If d = 1, the question is obvious: the structure is rigid iff G is connected. If d = 2, then one can design an efficient algorithm to solve the problem using a theorem of Laman [8] and the matroid partitioning algorithm of Edmonds [2]. For $d \geq 3$ the problem is unsettled.

It is not difficult to see that the rigidity of the structure is equivalent to the following. Let, for each vertex $v \in V(a)$, $x_v = (x_{1v}, \ldots, x_{dv})$ be a d-tuple of variables. Consider the set of equations

$$(x_u - x_v)(u - v) = 0 \qquad (3)$$

for every edge $(u,v) \in E(a)$ (u,v are considered to be points in R^d). Then the structure is rigid iff the solution space of this system of equations has dimension $\binom{d+1}{2}$. [The vector x_u can be considered as the velocity of vertex u at some motion of G. Equation (3) expresses that the edge (u,v) is not compressed or stretched. There are $\binom{d+1}{2}$ independent rigid motions, so these are always solutions]. Now the matrix of (3) looks like $[A_1 \ldots A_d]$ where

$A_i = (a^i_{ue})$ $u \in V(G)$, $e \in E(G)$ is defined by

$$a^i_{ue} = \begin{cases} u_i & \text{if } e = (u, v) \\ 0 & \text{otherwise} \end{cases}$$

Since by the "general position" assumption the coordinates u_i is algebraically independent, the determination of the rank of this matrix is a special case of Edmonds' problem.

A related problem is the following. Let the polynomial $f(x_1, \ldots x_n)$ be presented as the result of N additions and multiplications, starting with the variables. We want to know if f is identically 0. Again, we can substitute random numbers for the variables and our previous remarks apply. Valiant [15] showed that such a polynomial is always representable as a determinant of size at most N + 2, every entry of which is a variable or a constant. So the polynomial-problem is a special case of the determinant problem (note that the determinant is not expressible by a polynomial number of additions and multiplications, so the converse is not true).

Edmonds' problem is an important special case of a general class of problems

called RP (random polynomial). Let the input data be coded in form of a 01-sequence X, and let us have the task to compute a property P(X) of X (i.e., P(X) = 1 if X has this property and p(X) = 0 otherwise). We say that the problem is in RP if there exists a polynomial f and a polynomial-bounded algorithm which computes for each instance X of the problem and for each 01-sequence Y of length f ($|X|$) a value $v(X,Y) \in \{0,1\}$ such that

(a) if P(X) = 0, then v(X,Y) = 0 for every Y;
(b) if P(X) = 1, then v(X,Y) = 1 for at least half of all sequences Y.

Note that the class NP could be defined by replacing "at least half of all sequences" by "at least one sequence" in (f). Hence $RP \subseteq NP$. Obviously, $P \subseteq RP$. It is not known whether equality holds at either place, but probably not.

Note that if we have a problem in RP we can "solve it" polynomially as follows: Generate a random 01-sequence Y and compute v(X,Y). The probability that $v(X,Y) \neq P(X)$ is at most 1/2. By repeating this k times the probability of

$$\max \{v(X,Y_1),\ldots,v(X,Y_k)\} \neq P(X)$$

will be less than 2^{-k}, so even for relatively small k the probability of an error is negligible.

The study of problems in RP has just begun (see [1, 13, 17]), motivated mainly by the problem of primality testing. Let us conclude by pointing out a connection to another very vivid area of combinatorical applications of probability theory.

Let A be a property of 01-sequences which is in NP. Suppose, moreover, that

$$\#\{X: \quad |X| = n, \quad X \in A\} = o(2^n) .$$

Construct, in polynomial time, a 01-sequence X of length n, not in A. By assumption choosing an X at random is good with probability tending to 1. If $X \in A$ means that the choice of variables x_i encoded by X is a root of (1) for some graph which has a perfect matching, algorithmic production of an X not having property A(X) would yield another polynomial-bounded algorithm for the matching problem.

But this problem has another interesting special case. Let A be the property of a graph A that it contains a clique or an independent set of more than $\frac{1}{2}\log_2|V(G)|$ vertices.

A graph not in A is an example showing that the well-known Ramsey Theorem is sharp up to a constant factor. Clearly $A \in NP$ and it is known that almost all graphs do not have property A. The construction of such a graph, however, has resisted

attempts by many. There is, in fact, a variety of combinatorial existence results which have simple proofs by random choice but no constructive proofs (see Erdös - Spencer [5]).

REFERENCES

1. L. Adleman, Two theorems on random polynomial time, 19th Annual Symp. on Foundations of Computer Science, IEEE, 1978, 75-83.

2. J. Edmonds, Minimum partition of a matroid into independent subsets, J. Res. Nat. Bur. Stand. 69B (1965) 67-72.

3. J. Edmonds, Paths, trees and flowers, Can. Journal of Math. 14(1965), 449-467.

4. J. Edmonds, Systems of distinct representatives and linear Algebra, J. Res. Rat. Bur. Stand. 71B(1967) 241-245.

5. P. Erdös, J. Spencer, Probabilistic Methods in Combinatorics, Publ. House Hungarian Acad. Sci, 1973.

6. T.A. Jenkyns, Matchoids: A generalization of matchings and matroids, Ph.D. Thesis, Univ. of Waterloo, 1974.

7. B. Korte, private communication.

8. G. Laman, On graphs and rigidity of plane skeletal structures, J. Eng. Math., 4(1970) 331-340.

9. E. Lawler, Combinatorial Optimization: Networks and Matroids; Holt, Rinehart and Winston, 1976.

10. L. Lovasz, Selecting independent lines from a family of lines in a space, Acta. Sci. Math. Univ. Szeged (to appear).

11. L. Lovasz, The matroid matching problem, Proc. Donf. on Algebraic Graph Theory, Szeged, 1978, to appear.

12. L. Lovasz, Matroid matching and some applications, submitted to J.C.T.

13. C. Rackoff, Relativized questions involving probabilistic algorithms, Proc. 10th Annual Symp. on Theory of Computing, 1978, 338-342.

14. W.T. Tutte, The factorization of linear graphs, J. London Math. Soc. 22(1947) 107-111.

15. L.A. Valiant, Completeness classes in algebra, Conf. Rec. 11th Annual ACM Symp. on Th. of Computing, 1979, 249-261.

16. Y. Yemini, D. Cohen, On some algorithmic aspects of structural rigidity, preprint.

17. R. Zippel, Probabilistic algorithms for sparse polynomials, to appear in Eurosam Proc. 1979.

Necessar

We devel
the theo
for c.f.
is allwa
the prod
free gro
by the r
$\hat{U}(G)$ is
change t
Using th
of Fox a
presenta
are inva
ideals a
fore the
In [H2]
culus on
this cal
mapping

with Z(
modul an
d is lin

e and c
In H2 we
gets inv
$Z(X^*)$ su
In the c
studied

1) Fachbe
Saarla