JOURNAL OF ALGORITHMS **10**, 557–567 (1989)

Maximum Matchings in General Graphs through Randomization*

MICHAEL O. RABIN

Aiken Computation Laboratory, Harvard University, and Institute of Mathematics, The Hebrew University of Jerusalem

AND

VIJAY V. VAZIRANI

Computer Science Department, Cornell University

Received December 15, 1987; accepted November 18, 1988

A new randomized algorithm for the maximum matching problem is presented. Unlike conventional matching algorithms which are combinatorial, our algorithm is algebraic and works on the Tutte matrix of the given graph. Although slower than the best known matching algorithm, our algorithm has the advantage of being conceptually simple and easy to program.

1. INTRODUCTION

We present a new randomized algorithm for the maximum matching problem. Unlike conventional matching algorithms which are based on the combinatorial approach of finding "augmenting paths" and "blossoms" in graphs (see [Ed1] for definitions), our algorithm is algebraic and works on the Tutte matrix of the given graph.

Edmonds [Ed1] gave the first polynomial time algorithm ($O(n^4)$) for this problem by giving an ingenious way of dealing with the complex manner in which blossoms get nested. Subsequently more efficient algorithms were obtained. The current best running time is $O(\sqrt{|V|}\,|E|)$ [MV], [Va]; this algorithm involves a very precise consideration of blossoms. The advantage of our approach is that it bypasses blossoms completely and therefore results in an algorithm which is conceptually simpler and is considerably easier to program, especially if a subroutine for matrix inversion is avail-

*This research was supported by NSF Grant MCS-81-21431 at Harvard University during 1983–84.

0196-6774/89 $3.00

able. On the other hand, it is less efficient, its running time is $O(M(n)n \log n \log\log n)$ bit operations, where $M(n)$ is the number of arithmetic operations required for multiplying two $n \times n$ matrices and n denotes the number of vertices in the graph. The current best bound for $M(n)$ is $O(n^{2.376})$ [CW]; however, because of a very large constant in the running time, this algorithm is not practical. Using Gaussian elimination and ordinary integer multiplication, the running time of our algorithm is $O(n^4 \log^2 n)$ bit operations.

This algorithm was programmed by Dan Winkler at Harvard University in 1984. It was found to be very quick for graphs having up to a few hundred vertices. For such sized graphs, the numbers being operated on fit into one computer word; consequently, the running time can be taken to be $O(n^4)$ word operations. Considering also the ease of programming, our algorithm may be a more practical alternative, especially for small graphs.

Central to our algorithm is a theorem of Tutte [Tu] which states that the Tutte matrix of a graph (see Section 2 for definition) is non-singular iff the graph contains a perfect matching. The first algorithmic use of this theorem was proposed by Lovasz [Lo]; he gave a randomized algorithm for the decision problem, "Does the given graph have a perfect matching?" The idea is to substitute for the variables in the Tutte matrix of the graph randomly from a polynomially large set of integers, and test the resulting matrix for non-singularity. A straightforward algorithm for finding a perfect matching follows using self-reducibility; it makes $O(n^2)$ calls to the above procedure, and has a running time of $O(M(n)n^3 \log^3 n \log\log n)$.

In this paper, we obtain a more efficient algorithm for the search problem. The improvement comes in three ways. First, instead of using determinant computations, we resort to matrix inversion. The advantage is that matrix inversion is no more expensive than a determinant computation; on the other hand, the inverse has information about all n^2 minors of the matrix. We show that $n/2$ matrix inversions suffice; each inversion gives one edge of the matching. Second, we perform computations over a finite field; for this purpose, we show that Tutte's theorem and its generalizations extend to finite fields. This also enables us to obtain an $O(M(n)\log^2 n)$ algorithm for computing the size of a maximum matching in a graph. Although not practical, this is asymptotically the best known algorithm for this task. Third, notice that the straightforward algorithm sketched above requires fresh random bits for each call. This is not only wasteful, but also makes the error probability accumulate. Instead, in our algorithm, only one random substitution suffices; the remaining inversions are performed on submatrices of this initial randomized matrix.

Lovasz generalized Tutte's theorem to show that the rank of the Tutte matrix gives the size of the maximum matching in the graph [LP]. In this paper, we present an alternative proof of this generalization using an old

theorem of Frobenius. This proof schema enables us to derive additional properties of the Tutte matrix which are used in the algorithm.

Finally, we present some preliminary results on the problem of parallelizing matching. In fact, since this work first appeared in [RV], considerable progress has been made on understanding the parallel complexity of matching. The first random NC (RNC^3) algorithm was obtained by Karp, Upfal, and Wigderson [KUW]. Subsequently, a more direct RNC^2 algorithm was found by Mulmuley, Vazirani, and Vazirani [MVV].

These algorithms and other results on the parallel complexity of matching use an algebraic approach, rather than a combinatorial one, and work on the Tutte matrix of the graph. It is not surprising that some of our ideas have been relevant in this direction. For example, the processor-efficient implementation of [KUW], given in [GP], uses matrix inversion. So does the algorithm of [MVV], where finding a maximum matching is reduced to a single matrix inversion. Also, the proof schema using Frobenius' theorem is used in [VV] to obtain a further generalization of Tutte's theorem, establishing a one-to-one correspondence between the bases of the Tutte matrix and the sets of vertices matched by maximum matchings in the graph. This is used for obtaining an RNC^2 algorithm for the two-processor scheduling problem (which has close connections with matching) [VV] and the vertex weighted matching problem [MVV].

2. Tutte's Theorem and Its Extensions

Tutte's theorem will be central to our algorithm; it gives an algebraic characterization for the existence of a perfect matching in a graph. The case of bipartite graphs is much simpler, and a similar characterization for this case was given by Edmonds [Ed2]. It should be mentioned that Tutte's theorem is based on the work of Pfaff on skew-symmetric matrices.

Definition. Given a graph $G(V, E)$, let D be its adjacency matrix, i.e.,

$$d_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \\ 0 & \text{otherwise.} \end{cases}$$

Replace the 1's in this matrix by indeterminates in the following manner: if $d_{ij} = d_{ji} = 1$, then replace these two entries by a unique indeterminate, say x_{ij}, and its negative, i.e., $-x_{ij}$, so that the entries above the diagonal get a positive sign. The resulting matrix, A, is called the *Tutte matrix* of G. Notice that A is skew-symmetric. Its determinant, $|A|$, will be a polynomial in the various indeterminates.

THEOREM (Tutte). *Let G be a graph, and let A be its Tutte matrix. Then,*

$$|A| \not\equiv 0 \Leftrightarrow \textit{there } \exists \textit{ a perfect matching in } G.$$

The following generalization of Tutte's theorem was given by Lovasz [LP]:

THEOREM (Lovasz). *Let A be the Tutte matrix of graph G, and let m be the size of a maximum matching in G. Then,* $\operatorname{rank}(A) = 2m$.

We will first present an alternative proof of this generalization. The difficult step in carrying out the proof is showing the existence of a "symmetrically located" submatrix of A of dimension $\operatorname{rank}(A)$ (see proof). There are several ways of carrying out this step; we do it by invoking a theorem of Frobenius [Ko, p. 144] stated below. This proof schema enables us to derive additional useful properties of the Tutte matrix and, more generally, of skew-symmetric matrices (see Lemmas 1, 2, and 3). It was also used for obtaining a further generalization of Tutte's and Lovasz's theorems [VV].

Notation. Let A be an $n \times n$ matrix, and α and β be subsets of $\{1, 2, \ldots, n\}$. Then, $A_{\alpha\beta}$ will denote the submatrix of A obtained by choosing the rows of A corresponding to indices in α and columns corresponding to indices in β.

THEOREM (Frobenius). *Let A be an $n \times n$ skew-symmetric matrix, and let α and β be subsets of $\{1, 2, \ldots, n\}$ such that $|\alpha| = |\beta| = \operatorname{rank}(A)$. Then,*

$$|A_{\alpha\alpha}|\,|A_{\beta\beta}| = (-1)^{|\alpha|}|A_{\alpha\beta}|^2.$$

Proof of Lovasz's Theorem. We prove the theorem in two parts:

(i) $\operatorname{rank}(A) \geq 2m$. Choose any matching of size m, and let $U \subseteq V$ be the set of vertices matched by it. Let G' be the subgraph of G induced on U; G' has a perfect matching. Since A_{UU} is the Tutte matrix of G', $|A_{UU}| \not\equiv 0$. Since $|U| = 2m$, $\operatorname{rank}(A) \geq 2m$.

(ii) $\operatorname{rank}(A) \leq 2m$. Suppose $\operatorname{rank}(A) = k$. Let $A_{\alpha\beta}$ be a non-singular submatrix of A, with $|\alpha| = |\beta| = k$. We want to show that G has a matching of size at least $k/2$. Notice that if $A_{\alpha\beta}$ is *symmetrically located*, i.e., $\alpha = \beta$, we are done (since the restriction of G to the vertices in α must have a perfect matching by Tutte's theorem).

Since $|A_{\alpha\beta}| \not\equiv 0$ and $|\alpha| = |\beta| = \operatorname{rank}(A)$, by Frobenius's theorem, $|A_{\alpha\alpha}| \not\equiv 0$. Since this is a symmetrically located submatrix of A, by the

previous remark, G has a matching of size $k/2$ (notice that k must be even, since a skew-symmetric matrix of odd dimensions is always singular). □

Tutte's theorem suggests a means for testing if the given graph has a perfect matching, namely, check if $|A|$ is non-vanishing. The difficulty is that the known methods for testing deterministically whether A is non-singular resort to computing $|A|$, which may take exponential time since A contains an unbounded number of indeterminates. Lovasz [Lo] proposed the use of randomization for getting around this difficulty: substitute for the variables in A randomly from a polynomially large set of integers; if A was non-singular, the substituted matrix will be non-singular with very high probability. Since the substituted matrix has small, i.e., $O(\log n)$ bit integral entries, its determinant can be computed in polynomial time. A formal probabilistic analysis of this algorithm is based on a lemma of Schwartz [Sc].

The above algorithm, although polynomial time, is inefficient: since $|A|$ is being computed over the rationals, the intermediate numbers in the computation may be $O(n \log^2 n)$ bits long. For the purpose of efficiency, we will substitute for the indeterminates in A randomly from a finite field $\mathbf{Z}_p$; however, in order to do this we must clear up one point.

Tutte's determinant $|A|$ is a polynomial in certain variables x_{ij} with integer coefficients. Could it happen that all these coefficients are divisible by a prime p so that mod p the polynomial is the zero polynomial even though $|A| \not\equiv 0$ over the integers? In this case the substituted matrix will be necessarily singular.

Similarly, in the case that $|A| \not\equiv 0$ (i.e., there are no perfect matchings) we want to compute rank(A). If rank$(A) = k$ then there is a $k \times k$ submatrix A' of A so that $|A'| \not\equiv 0$ as a polynomial. Once again, could it happen that $|A'|$ is zero mod p?

LEMMA 1. *If A is a Tutte matrix of graph G and $|A| \not\equiv 0$ then for any prime p, $|A| \bmod p \not\equiv 0$.*

Proof. The condition $|A| \not\equiv 0$ implies, by Tutte's theorem, that G has a perfect matching $M = \{(i_1, j_1), \ldots, (i_m, j_m)\}$, where $m = n/2$. But then $|A|$ contains the monomial

$$x_{i_1 j_1}^2 \cdot \cdots \cdot x_{i_m j_m}^2$$

with coefficient 1. Thus $|A| \bmod p \not\equiv 0$. □

LEMMA 2. *If A is the Tutte matrix of graph G and* rank$(A) = k$ *then for any prime p, there exists a $k \times k$ submatrix B of A such that $|B| \not\equiv 0$ and $|B| \bmod p \not\equiv 0$.*

Proof. Since $\text{rank}(A) = k$, there exists a $k \times k$ submatrix $A_{\alpha,\beta}$ such that $|A_{\alpha,\beta}| \not\equiv 0$. By Frobenius' theorem, also $|A_{\alpha,\alpha}| \not\equiv 0$. Since $A_{\alpha,\alpha}$ is the Tutte matrix of a subgraph of G, the proof follows from Lemma 1. □

Remark. Actually, the ideas in the proof of Tutte's theorem can be extended to show that for *every* non-singular submatrix of A' of a Tutte matrix, $|A'|$ is a polynomial in which every coefficient is a power of 2. Thus in every field Z_p, $p \geq 3$, *none* of the coefficients degenerate. Moreover, every such submatrix has a monomial with coefficient ± 1.

Finally, we will prove a property of skew-symmetric matrices that will be useful in the matching algorithm.

Notation. Let A be an $n \times n$ matrix. Denote by A_{ij} the $(n-1) \times (n-1)$ submatrix obtained from A by removing the ith row and jth column. Similarly denote by $A_{ii,jj}$ the $(n-2) \times (n-2)$ submatrix obtained from A by removing the ith row *and* column as well as the jth row and column. There is no danger of confusion with the previous use of $A_{\alpha\beta}$; the meaning will always be clear from the context.

LEMMA 3. *Let A be an $n \times n$ skew-symmetric matrix of even dimension, with entries in a field F. For $1 \leq i, j \leq n$, $i \neq j$, if $|A_{ij}| \neq 0$ then $|A_{ii,jj}| \neq 0$.*

Proof. W.l.o.g. we may assume that $i = 1$ and $j = 2$. Since $|A_{12}| \neq 0$, the $n-1$ columns of A_{12} are linearly independent. In particular, the $n-2$ columns of A_{12} numbered by $\beta = \{3, \ldots, n\}$ are linearly independent. Therefore, for some subset $\alpha \subseteq \{2, \ldots, n\}$, $|\alpha| = n-2$, the square submatrix $A_{\alpha\beta}$ is nonsingular.

Now consider the matrix A_{11}. Since this matrix is skew-symmetric and of odd dimension, it is singular. Since $A_{\alpha\beta}$ is also a submatrix of A_{11} and $|A_{\alpha\beta}| \not\equiv 0$, $\text{rank}(A_{11}) = n-2$. Therefore, by Frobenius' theorem,

$$|A_{\beta\beta}| = |A_{11,22}| \neq 0. \quad \square$$

3. THE MATCHING ALGORITHM

We first give an efficient algorithm for computing the size of a maximum matching in a graph.

THEOREM 1. *There is an $O(M(n)\log^2 n)$ randomized (Monte Carlo) algorithm for computing the size of a maximum matching in a graph $G(V, E)$.*

Proof. Let A be the Tutte matrix of graph G. If $\text{rank}(A) = k$, then by Lemma 2, A contains a $k \times k$ submatrix B, such that $|B| \bmod p \not\equiv 0$. All larger submatrices of A are singular.

Find a prime $p \geq n^4$. This can be done quickly, using a randomizing test for primality (Rabin [Ra2], Solovay and Strassen [SS]). Choose a *random substitution* S: *set of* $x_{ij}'s \to Z_p$, i.e., independently, each $S(x_{ij})$ is equally likely to be any element of Z_p. Denote by A^S the result of replacing each x_{ij} in A by $S(x_{ij})$. Using Schwartz's lemma [Sc] one can show

$$\Pr[S:|B_S| \neq 0] \geq \left(\frac{p-2}{p}\right)^{n^2} \approx e^{-2/n^2}.$$

This is a simple calculation which utilizes the fact that $\deg_{x_{ij}}|B| \leq 2$ for each variable, and there are fewer than n^2 distinct variables. Hence, with probability at least e^{-2/n^2}, i.e., very close to 1, rank (A^S) = rank(A). The theorem follows since rank(A^S) can be computed in $O(M(n)\log^2 n)$ steps (see [BH, IMH]). □

Using fast matrix multiplication, Theorem 1 gives the best known algorithm for computing the size of a maximum matching in a dense graph (the best combinatorial algorithm requires $O(n^{2.5})$ time on dense graphs [MV]). Although this is not of practical value, it provides progress on a challenging open problem (see Section 5).

The matching algorithm will be developed in three stages:

(*a*) *Finding a Perfect Matching*

DEFINITION. Let G be a graph having a perfect matching, A be its Tutte matrix, and p be any prime. A substitution S for the variables in A by elements of Z_p is a *good substitution* if $|A^S| \neq 0 \pmod p$.

Our algorithm works by inverting A^S to identify one edge of the perfect matching. The following lemma shows how.

Notation. We will denote the (i, j)th entry of matrix A by (lowercase) a_{ij}.

LEMMA 4. *Let S be a good substitution for the Tutte matrix A and let $B = (A^S)^{-1}$. There is a j, $1 < j \leq n$ such that $S(a_{1j}) \neq 0 \pmod p$ and $b_{j1} \neq 0 \pmod p$. Moreover, for each j satisfying this condition, the edge (v_1, v_j) is in some perfect matching of G.*

Proof. Since S is a good substitution, $|A^S| \neq 0 \pmod p$,

$$|A^S| = \sum_{j=1}^{n} (-1)^{1+j} S(a_{1j}) |A_{1j}^S|.$$

Therefore, there is a j, $1 < j \leq n$ (notice that $a_{11} = 0$) such that $S(a_{1j}) \neq 0 \pmod{p}$ and $|A^S_{1j}| \neq 0 \pmod{p}$; since $b_{j1} = (-1)^{1+j}|A^S_{1j}|/|A^S|$, also $b_{j1} \neq 0 \pmod{p}$. Therefore G must have the edge (v_1, v_j). By Lemma 3, $|A^S_{11jj}| \neq 0 \pmod{p}$. So $|A_{11jj}| \not\equiv 0$ over Z_p, and also over the integers. Let G_1 be the graph obtained by restricting G to $V - \{v_1, v_j\}$; A_{11jj} is its Tutte matrix. Since this matrix is non-singular, G_1 contains a perfect matching, by Tutte's theorem. Hence, (v_1, v_j) is in some perfect matching of G. □

ALGORITHM. The algorithm consists of two phases. In the first phase, a random substitution S is chosen for A, as in Theorem 1. With high probability, this will be a good substitution for A. If so, the second phase will succeed in finding a perfect matching in G. This phase consists of $n/2$ iterations; in each iteration, one edge of the matching is found.

In the first iteration, A^S is inverted, and an edge (v_1, v_j) is found as described in Lemma 4. Since $|A^S_{11jj}| \neq 0 \pmod{p}$, S is also a good substitution for the Tutte matrix, A_{11jj}, of the graph, G_1, induced on $V - \{v_1, v_j\}$. Now, by inverting A^S_{11jj}, an edge is chosen from G_1, and so on. The union of edges so chosen form a perfect matching in G.

(*b*) *Extension to Maximum Matching*

Using Theorem 1, we can compute the size of the maximum matching in the graph, say it is m. Obtain graph G' from G by adding $n - 2m$ new vertices and an edge connecting each new vertex to each original vertex. G' clearly has a perfect matching; moreover, any such perfect matching when restricted to G gives a maximum matching in G.

(*c*) *Making the Algorithm Las Vegas*

The algorithm presented above is Monte Carlo, in the sense that it has a non-zero probability of ending in an error. Using the method proposed by Lovasz (appearing in [Ka]) for obtaining Las Vegas extensions for the parallel matching algorithms, a similar extension for our algorithm also follows. We refer the reader to [Ka] for details. As stated in that paper, we will compute the set of "non-critical" vertices by n calls to the procedure of Theorem 1. Then, using the Tutte–Berge formula and the Gallai–Edmonds structure theorem, we obtain an upper bound on the size of the maximum matching in G. Because of error probability in Theorem 1, this will also be a Monte Carlo algorithm. Running the two Monte Carlo algorithms simultaneously until they agree yields a Las Vegas algorithm. When it halts, it will have the correct solution, and its *expected* running time is $O(M(n)n \log^2 n)$. (For a detailed description of these notions see [Gi] or [Ra1].)

THEOREM 2. *There is an $O(M(n)n \log^2 n)$ Las Vegas algorithm for finding a maximum matching in a graph.*

4. The Parallel Perspective

PROPOSITION 1. *There is an NC^2 algorithm for finding a perfect matching in a graph having a unique perfect matching.*

Proof. Let $G(V, E)$ have a unique perfect matching, let A be its Tutte matrix, and let $M = \{(v_{i_1}, v_{j_1}) \cdots (v_{i_m}, v_{j_m})\}$ be the perfect matching in G, where $m = n/2$. Then,

$$|A| = x_{i_1 j_1}^2 \cdots x_{i_m j_m}^2.$$

Let S be the substitution s.t. $x_{ij} = 1$ for each variable. Since $|A^S| = 1 \pmod p$, S is a good substitution. Let $B = (A^S)^{-1}$. For each i, $1 \le i \le n$, there is a unique j s.t. $(v_i, v_j) \in E$ and $G(V - \{v_i, v_j\})$ has a perfect matching. Therefore, by Lemma 4, for each i, $1 \le i \le n$, there is a unique j s.t. $S(a_{ij}) \neq 0$ and b_{ji}. Hence, by picking a substitution deterministically, and inverting A^S, we obtain the perfect matching. Since matrix inversion is in NC^2 [Cs], the result follows. Notice that for this case, the sequential algorithm runs in $O(M(n)\log^2 n)$ time. (See also Section 5). □

Notice that uniqueness is crucial in the above procedure. The importance of uniqueness in parallel computation is further explored and utilized in the parallel matching algorithm of [MVV]. They observe that in a parallel algorithm one must coordinate the processors so they seek the *same* solution in parallel. The question is, "which solution?" The obvious alternative of seeking the lexicographically largest solution is often not suitable, because this renders the problem P-complete even though an arbitrary solution can be found in parallel (e.g., maximal independent set and depth first search). The parallel complexity of finding the lexicographically largest perfect matching is as yet unresolved. The main ingredient in [MVV] is the use of randomization for isolating one solution from the possibly exponentially many. Currently, an outstanding open problem is to obtain an NC algorithm for matching, i.e., isolate one solution deterministically (see Section 5).

In the next result we show that in order to give a parallel matching algorithm, it is sufficient to restrict attention to perfect matchings, by showing that using an oracle for the perfect matching problem one can solve the maximum matching problem in NC^1, i.e., showing an NC^1-reduction (see [Co] for a formal definition).

PROPOSITION 2. *The problem of finding a maximum matching NC^1 reduces to the perfect matching problem.*

Proof. The procedure in Section 3(b), which uses Theorem 1, gives an RNC^2 reduction. Let us extend this to an NC^1 reduction. Let m be the size

of a maximum in G, and let G' be obtained from G by adding k new vertices, where the parity of k is the same as the parity of n, and there is an edge connecting each new vertex to each original vertex. Clearly, G' has a perfect matching iff $k \geq n - m$. So, using the oracle for perfect matching and doing a binary search on k, we can determine m. The reduction follows. □

5. Discussion and Open Problems

Several open problems remain. First, notice that in each iteration, the algorithm uses only one column of the inverse. It therefore seems that performing $n/2$ matrix inversions is quite wasteful; it might be possible to use some clever updating ideas to quickly obtain subsequent matrix inverses from the previous ones and aim for a running time of $O(n^3 \log^2 n)$. Even more challenging would be to attempt the open problem stated in [MVV], of finding a perfect matching in time $O(M(n)\log^k n)$, for some constant k. Although not of practical value, this would tie the complexity of matching to that of matrix multiplication. It seems that in order to achieve this running time one would need to compute over a finite field, and some of the ideas presented here might be useful. Theorem 1 and Proposition 1 provide progress towards this problem.

Currently, an outstanding open problem in parallel computation is to give a (deterministic) *NC* algorithm for matching. In an interesting result, Proposition 1 was recently generalized to graphs having polynomially many perfect matchings [GK]. Other results related to this open problem appear in [KVV].

Acknowledgments

We are thankful to Manuel Blum and Les Valiant for valuable discussions.

References

[BH] J. R. Bunch and J. E. Hopcroft, Triangular factorization and inversion by fast matrix multiplication, *Math. Comput.* **28**, 125 (1974), 231–236.

[Co] S. A. Cook, A taxonomy of problems with fast parallel algorithms, *Inform. and Control* **64** (1985), 2–22.

[CW] D. Coppersmith and S. Winograd, Matrix multiplication via arithmetic progressions, *in* "Proceedings, 19th STOC Conf., 1987," pp. 1–6.

[Ed1] J. Edmonds, Paths, trees and flowers, *Canad. J. Math.* **17** (1965), 449–467.

[Ed2] J. Edmonds, Systems of distinct representatives and linear algebra, *J. Res. Nat. Bur. Standard. Sect. B.* **71**, No. 4 (1967), 241–245.

[GP] Z. GALIL AND V. PAN, Improved processor bounds for algebraic and combinatorial problems in RNC, *in* "Proceedings, 26th FOCS, 1985," pp. 490–495.

[Gi] J. GILL, Computational complexity of probabilistic turing machines, *SIAM J. Comput.* **6** (1977), 675–695.

[GK] D. GRIGORIEV AND M. KARPINSKI, The matching problem for bipartite graphs with polynomially bounded permanents is in *NC*, *in* "Proceedings, FOCS, 1987," pp. 166–172.

[IMH] O. H. IBARRA, S. MORAN, AND R. HUI, A generalization of the fast LUP matrix decomposition algorithm and applications, *J. Algorithms* **3** (1982), 45–56.

[Ka] H. KARLOFF, A randomized parallel algorithm for the odd set cover problem, *Combinatorica* **6**, No. 4 (1986), 387–391.

[KUW] R. M. KARP, E. UPFAL, AND A. WIGDERSON, Constructing a maximum matching is in random NC, *Combinatorica* **6**, No. 1 (1986), 35–48.

[Ko] G. KOWALEWSKI, "Einfuhrung in die Determinanten Theorie," pp. 144, Leipzig Verlag von Veit & Co., 1909.

[KVV] D. KOZEN, U. V. VAZIRANI, AND V. V. VAZIRANI, NC algorithms for comparability, interval graphs, and testing for unique perfect matchings, *in* "Fifth FST & TCS Conference, 1985."

[LR] J. K. LENSTRA AND A. H. G. RINNOY KAN, Complexity of scheduling under precedence constraints, *Operations Res.* **26** (1978), 22–35.

[Lo] L. LOVASZ, On determinants, matchings, and random algorithms, *in* "Fundamentals of Computing Theory" (L. Budach, Ed.), Akademia-Verlag, Berlin, 1979.

[LP] L. LOVASZ AND M. PLUMMER, "Matching Theory," Academic Press, Budapest, Hungary, 1986.

[MV] S. MICALI AND V. V. VAZIRANI, An $O(\sqrt{|V|}\,|E|)$ algorithm for finding maximum matching in general graphs, *in* "Proceedings, FOCS, 1980," pp. 17–27.

[MVV] K. MULMULEY, U. V. VAZIRANI, AND V. V. VAZIRANI, Matching is as easy as matrix multiplication, *Combinatorica* **7**, No. 1 (1987), 105–113.

[Ra1] M. O. RABIN, Probabilistic algorithms, *in* "Algorithms and Complexity"(J. F. Traub, Ed.), Academic Press, New York (1976) pp. 21–39.

[Ra2] M. O. RABIN, Probabilistic algorithms for testing primality, *J. Number Theory* **12** (1980), 128–138.

[RV] M. O. RABIN AND V. V. VAZIRANI, "Maximum Matching in General Graphs through Randomization," Report No. TR-15-84, Center for Research in Computing Technology, Harvard University, Cambridge, MA 1984.

[Sc] J. T. SCHWARTZ, Fast probabilistic algorithms for verification of polynomial identities, *J. Assoc. Comput. Mach.* **27**, No. 4 (1980), 701–717.

[SS] R. SOLOVAY AND V. STRASSEN, A fast Monte-Carlo test for primality, *SIAM J. Comput.* **6** (1977), 84–85.

[Tu] W. T. TUTTE, The factorization of linear graphs, *J. London Math. Soc.* **22** (1947), 107–111.

[VV] U. V. VAZIRANI AND V. V. VAZIRANI, The two-processor scheduling problem is in R-NC, *SIAM J. Comput.* **18**, No. 6 (1989).

[Va] V. V. VAZIRANI, A theory of alternating paths and blossoms for proving correctness of the $O(\sqrt{|V|}\,|E|)$ general graph matching algorithm, to appear.