Discrete Applied Mathematics 25 (1989) 179-190
North-Holland

PFAFFIAN ORIENTATIONS, 0-1 PERMANENTS, AND EVEN CYCLES IN DIRECTED GRAPHS

Vijay V. VAZIRANI*
Computer Science Department, Cornell University, 4130 Upson Hall, Ithaca, NY 14853, USA

Milhalis YANNAKAKIS
AT&T Bell Labs, Murray Hill, NJ 07974, USA

Received 9 January 1989

The following issues in computational complexity remain imprecisely understood: The striking difference in the complexities of computing the permanent and determinant of a matrix despite their similar looking formulae, the complexity of checking if a directed graph contains an even length cycle, and the complexity of computing the number of perfect matchings in a graph using Pfaffian orientations. Via polynomial time equivalences, we show inter-relationships among these issues.

1. Introduction

The expression for computing the permanent of a matrix differs from that of the determinant only in the sign terms; yet the computational complexities of these two problems are strikingly different. The determinant of a matrix can be computed in polynomial time using the age-old Gaussian elimination; on the other hand, all attempts at computing the permanent efficiently have failed. In an early attempt, Polya [15] suggested computing the permanent of a 0-1 matrix A by reducing it to a suitable determinant. His idea was to change some of the $+1$ entries of A into -1 so that the determinant of the resulting matrix B equalled the permanent of A. However, he showed that such a transformation is not always possible. In a related result, Marcus and Minc [13] showed that for $n \geq 3$, there is no linear transformation T which reduces permanent to determinant, i.e., such that $\text{perm}(A) = \det(T(A))$ for all real matrices of order n. Valiant [19] explained the hardness of computing the permanent using modern complexity theory notions, by showing that this problem is $\#P$-complete. This holds even when the problem is restricted to 0-1 matrices.

In this paper, we first reconsider Polya's scheme and study the complexity of the following problem: Given a 0-1 matrix A, does there exist a transformed matrix B, obtained by changing some of the $+1$ entries of A into -1, so that $\text{perm}(A) = \det(B)$? We show that this problem is polynomial-time equivalent to the problem

* Work done while visiting AT&T Bell Labs during 1986-87. Supported in part by a PYI Award, with matching funds from AT&T Bell Labs.

0166-218X/89/\$3.50 © 1989, Elsevier Science Publishers B.V. (North-Holland)

of determining if a given directed graph contains an even length cycle. This later problem has algebraic connections (for example with the sign-solvability of matrices [6, 12, 18]), and is an outstanding open problem in computational graph theory [16, 17].

The problem of computing the permanent of a 0–1 matrix is the same as the problem of counting the number of perfect matchings in a bipartite graph. Let us consider the generalization of this later problem to arbitrary graphs. Clearly, this problem is also #P-complete. This problem has applications in physics (see for example [5]), especially when the given graph is planar.

In a classic result, Kasteleyn, a physicist, gave a poynomial time algorithm for computing the number of perfect matchings in planar graphs [5]. In retrospect, Kasteleyn's method may be viewed as an extension of Polya's idea to general graphs. The idea is to start with the Tutte matrix of the graph, and substitute either $+1$ or -1 for the variables so that the determinant of the resulting matrix is the square of the number of perfect matchings in the graph. Kasteleyn defined the notion of Pfaffian orientation of a graph; this orientation tells us how to do the above substitution. Once the Pfaffian orientation of a graph is found, the number of perfect matchings in it can be computed in polynomial time. Kasteleyn also showed that every planar graph has such an orientation, and gave a polynomial time algorithm for finding it.

Little [8] extended Kasteleyn's work by showing that every $K_{3,3}$-free graph has a Pfaffian orientation. It is easy to see that Little's proof yields a polynomial time algorithm for computing the number of perfect matchings in a $K_{3,3}$-free graph; moreover, this can also be done in NC [21]. Is there a larger class of graphs which have Pfaffian orientation? More importantly, a long standing open problem is to resolve the complexity of determining if a given graph has a Pfaffian orientation, or even to determine if a given orientation for a graph is Pfaffian (see [11]).

We partially resolve this mystery: We show that the problem of determining if a given graph has a Pfaffian orientation is in Co-NP; moreover, it is polynomial-time equivalent to determining if a given orientation for a graph is Pfaffian. We also show that for the case of bipartite graphs, these problems are equivalent to the even cycle problem. For the first result, we use Lovász's [10] polynomial time algorithm for computing the GF[2] rank of the set of perfect matchings of a graph. Lovász's algorithm is based on a decomposition of the graph into "bricks" and "braces". We show that the given graph has a Pfaffian orientation iff all of its bricks and braces have Pfaffian orientations. The proof is constructive; given valid orientations for the bricks and braces, we can obtain an orientation for the graph in polynomial time. Braces are bipartite graphs, so their Pfaffian orientation reduces to the even cycle problem. On the other hand, the complexity of orienting bricks remains an intriguing open problem. As a corollary we show that the number of Pfaffian orientations in a graph is either zero or a power of 2.

Lovász's work [10] draws on several important papers which study the perfect matching polytope, the linear space generated by incidence vectors of perfect matchings, and the ear and brick decompositions of matching covered graphs; and extends significantly this rich theory. The results stated above provide a natural application of this work.

We finally study the complexity of determining if the permanent of a given integer matrix is equal to its determinant. We consider several cases: When the entries are 0–1 (and generally, nonnegative), when they are arbitrary, and when the computations are done mod k, for a fixed integer k. The first case is the most interesting, and is shown to be polynomial-time equivalent to the even cycle problem.

The interface between algebra and graph theory has yielded interesting results. The algebraic connection of matching, via Tutte's theorem, has resulted in efficient algorithms, even fast parallel algorithms [4, 14]. On the other hand, despite its algebraic connections, the complexity of the even cycle problem remains unresolved. In this paper, we have established a link between evaluating 0–1 permanents (i.e., counting the number of perfect matchings in a bipartite graph) and the even cycle problem. It remains to be seen if this link yields algorithmic ideas for the even cycle problem (see Section 6 for a detailed discussion).

2. The complexity of Polya's scheme

Let A be an $n \times n$ matrix, and let σ be a permutation in the symmetric group S_n. The permanent of A is defined as follows:

$$\text{perm}(A) = \sum_{\sigma \in S_n} \text{value}(\sigma),$$

where,

$$\text{value}(\sigma) = \prod_{1 \le i \le n} A(i, \sigma(i)).$$

We will be concerned with the special case when A is a 0–1 matrix. Let $G(U, V, E)$ be the bipartite graph whose adjacency matrix is A. The bipartition is $U = \{u_1, \ldots, u_n\}$, $V = \{v_1, \ldots, v_n\}$, and the edge set is E. The permutation σ is a perfect matching in G iff for $1 \le i \le n$, $(u_i, v_{\sigma(i)}) \in E$. Hence

$$\text{value}(\sigma) = \begin{cases} 1, & \text{if } \sigma \text{ is a perfect matching in } G, \\ 0, & \text{otherwise.} \end{cases}$$

Thus perm(A) is equal to the number of perfect matchings in G.

The determinant of A is defined as follows:

$$\det(A) = \sum_{\sigma \in S_n} \text{sign}(\sigma) \cdot \text{value}(\sigma),$$

where sign(σ) is $+1$ if σ is an even permutation and -1 otherwise. Thus det(A) may fall short of perm(A) because of perfect matchings in G which are odd permutations. Polya's scheme for "remedying" this is the following: Change some of the $+1$ in A into -1 to obtain a new matrix B such that

$\forall$ perfect matching σ: $\text{sign}(\sigma) = \text{value}_B(\sigma)$.

Then clearly, $\det(B) = \text{perm}(A)$. We will study the computational complexity of:

POLYA'S PROBLEM.
-*Instance:* An $n \times n$ 0–1 matrix A.
-*Question:* Is there an $n \times n$ 0–1––1 matrix B satisfying Polya's scheme?

Let us define:

EVEN CYCLE.
-*Instance:* A directed graph G.
-*Question:* Is there an even length simple cycle in G?

Theorem 2.1. *POLYA'S PROBLEM is polynomial-time equivalent to EVEN CYCLE.*

Before giving the proof, we need to introduce another problem. Say that a directed graph is *even* if for every assignment of 0–1 weights to its edges, it contains an even weight cycle. The *weight* of a cycle is simply the sum of the weights of its edges. Seymour and Thomassen [16] show that the problem of testing if a given graph is even is polynomial-time equivalent to EVEN CYCLE. Section 3 is motivated by their theorem; however, in Section 3 we introduce a new proof method which can also be used to give a simpler proof of their theorem.

Proof. Let us first reduce POLYA'S PROBLEM to EVEN CYCLE. Let A be the $n \times n$ adjacency matrix of bipartite graph $G(U, V, E)$. Find a perfect matching M in G (if G has none, then $\text{perm}(A) = 0$). W.l.o.g. assume that this perfect matching corresponds to the identity permutation (since permuting the columns of A will not change its permanent), i.e., A has +1's on its diagonal. W.l.o.g. we may fix each diagonal entry B to +1.

Now, the signs of the remaining perfect matchings (permutations) depend on the lengths of their alternating cycles w.r.t. M. Construct a directed graph H on vertex set $X = \{x_1, \dots, x_n\}$ as follows: Corresponding to each edge (u_i, v_j) in G, there is a directed edge $(x_i \to x_j)$ in H. Notice that H will have a self-loop on each vertex, corresponding to the edges in M. Furthermore, each perfect matching in G corresponds to a *cycle cover* in H, i.e., a set of directed edges such that each vertex has one incoming and one outgoing edge. Consider the cycle cover C, corresponding to a perfect matching N, and let e be the number of even cycles in it. Then $\text{sign}(N) = (-1)^e$. Let us define $\text{sign}(C)$ to also be $(-1)^e$. We will now prove an easy though important lemma which will be useful in Section 5 as well.

Lemma 2.2. *Let H be a digraph with a self-loop on each vertex, and let A be its adjacency matrix (notice that A will have 1's on its diagonal). Then,* $\det(A) = \text{perm}(A)$ *iff H has no even cycle.*

Proof. Clearly, $\det(A) \le \mathrm{perm}(A)$; equality holds iff H has no cycle cover having negative sign. If H has no even cycles, clearly every cycle cover has positive sign and equality holds. On the other hand, if H has an even cycle, we can complete it into a cycle cover C by picking self-loops on the vertices not in this cycle. Since C has negative sign, equality does not hold. □

Thus, if H has no even cycles, the answer to Polya's problem is trivially "yes". Lemma 2.2 also indicates that we need to "repair" the even cycles in H in order to solve Polya's problem. The following lemma shows how to do this.

Lemma 2.3. *There is a matrix B satisfying Polya's scheme iff the directed graph H is not even.*

Proof. Suppose H is not even. Then there is an assignment of 0–1 weights to the edges of H so that there is no even weight cycle. The edges of H correspond to the off-diagonal entries of A. Make an entry -1 iff the corresponding edge has weight 0. Let B be the resulting matrix. We will show:

$$\forall \text{ perfect matching } \sigma\text{: } \mathrm{sign}(\sigma) = \mathrm{value}_B(\sigma).$$

First notice that even (odd) length cycles in H must have an odd (even) number of 0 weight edges. Suppose σ is even. Then σ "traces" an even number of even length cycles in H, and hence it "traces" an even number of 0 weight edges. Therefore $\mathrm{value}_B(\sigma) = +1$. Similarly, if σ is odd, it "traces" an odd number of even length cycles in H, and so $\mathrm{value}_B(\sigma) = -1$.

Next suppose there is a matrix B satisfying Polya's scheme. Assign 0–1 weights to the edges of H so that the 0 weight edges correspond to the -1 entries in B. Now, for each cycle in H, we can demonstrate a perfect matching in G that "traces" exactly this one cycle. Hence all cycles in H must have odd weight. □

Lemma 2.3 together with [16] completes the reduction from POLYA'S PROBLEM to EVEN CYCLE. The reduction in the other direction is also obvious: Given a directed graph H, construct the 0–1 matrix A with 1's on its diagonal, and the remaining 1 entries correspond to the directed edges in H. By Lemma 2.3, H is not even iff A is a "yes-instance" of POLYA'S PROBLEM. □

3. Pfaffian orientations

Definition. Say that a cycle C in graph $G(V, E)$ is *good* if it has even length and $G(V-C)$ has a perfect matching. A graph obtained by directing each edge in G is called an *orientation* of G. An even cycle in an oriented graph is *oddly oriented* if in traversing the cycle, an odd number of its edges are directed in the direction of traversal. An orientation $\vec{G}$ of a graph G is a *Pfaffian orientation* if every good cycle is oddly oriented.

It can be shown that in order to obtain a Pfaffian orientation it is sufficient to oddly orient all the alternating cycles w.r.t. *any* perfect matching in G (see [1]). The importance of Pfaffian orientation stems from the following: Let $\vec{G}$ be a Pfaffian orientation of $G(V,E)$. Let A be the (symmetric) $n\times n$ adjacency matrix of G, $|V|=n$. Obtain a matrix B from A as follows:

$$B(i,j)=\begin{cases}+1, & \text{if } (v_i\to v_j)\in\vec{G},\\ -1, & \text{if } (v_j\to v_i)\in\vec{G},\\ 0, & \text{otherwise.}\end{cases}$$

B is a skew-symmetric matrix, and $\det(B)$ will be the square of the number of perfect matchings in G. B is derived from the Tutte matrix of G. For a detailed explanation of this theory see [1,5,11].

The complexity of the following two problems is as yet unresolved:

PROBLEM 1. "Does the given graph G have a Pfaffian orientation?"

PROBLEM 2. "Is $\vec{G}$ a Pfaffian orientation for G?"

Notice that since G may have exponentially many good cycles, PROBLEM 2 is not trivially solvable.

Theorem 3.1. *PROBLEM 1 and PROBLEM 2 are polynomial-time equivalent.*

Proof. The basic idea behind Theorem 3.1 is the following principle from linear algebra: Suppose we have a set of linear equations $Ax=b$ over an arbitrary field. Suppose A is $m\times n$, with $m>n$, and $\text{rank}(A)=r$. Choose a basis for the row space of A, and denote this by A_r. Let b_r be the vector containing the corresponding components of b, and let s be *any* solution to $A_rx=b_r$. Then, $Ax=b$ is solvable (i.e., consistent) iff $As=b$.

Let us first orient G arbitrarily to obtain graph $\vec{H}$. Now, for each edge e in G assign a GF[2] variable x_e. Any other orientation for G can be described in terms of $\vec{H}$ and an assignment for the variables x_e: $x_e=0$ if e has the same orientation as in $\vec{H}$, and $x_e=1$ otherwise.

PROBLEM 1 < PROBLEM 2. Let M be any perfect matching in G. The condition of oddly orienting an alternating cycle w.r.t. M can be written as a GF[2] equation over the variables occurring on the edges of this cycle. Thus each alternating cycle gives one equation, and G has a Pfaffian orientation iff these equations are consistent. Let us write this set of GF[2] equations as $Ax=b$ where x is the vector of the edge variables. The matrix A simply describes the set of alternating cycles in G. Notice that A may have exponentially many rows.

In [10], Lovász gives a polynomial time algorithm for computing the GF[2] rank of the set of perfect matchings of a graph, and also for finding a basis. Find such a basis for G, say B. Notice that the dimension of A is one less than the dimension of B; moreover, a basis for A can be obtained as follows: pick any vector in B and compute its GF[2] sum with each of the remaining vectors in B. These sums form a basis for A.

Since the basis of A is small (polynomially bounded), we can find a solution s, for x which satisfies all basis equations. This solution gives us an oriented graph $\vec{G}$. Using the principle stated above, $Ax=b$ is solvable iff $As=b$. Hence G has a Pfaffian orientation iff $\vec{G}$ is a Pfaffian orientation. This completes the reduction.

PROBLEM 2 < PROBLEM 1. We want to check if $\vec{G}$ is a Pfaffian orientation for G. Let s be the vector which describes the orientations of edges in $\vec{G}$ w.r.t. $\vec{H}$. So, we want to check if $As=b$. Obtain again a basis for A, and first check if s satisfies the basis equations. If so, s satisfies all equations iff the set of equations is consistent, i.e., iff G has a Pfaffian orientation. □

Remark. The principle stated above can be used to obtain a simpler proof of the theorem in [16]: That the problem of testing if a given graph is even is polynomial-time equivalent to EVEN CYCLE. Once again, assign a GF[2] variable to each edge, representing its weight. For each directed cycle, there is a GF[2] equation saying that its weight is odd. This gives a system of equations $Ax=b$, with b being the all 1 vector. Now, we can obtain a basis for A (i.e., the directed cycles of G) from the ear decomposition of G (assuming without loss of generality that G is strongly connected). The rest of the proof is similar to Theorem 3.1.

Corollary 3.2. *PROBLEM 1 is in Co-NP.*

Proof. From the proof of Theorem 3.1 it is easy to see that there is a short certificate that enables us to verify in polynomial time that G has no Pfaffian orientation: If G has no Pfaffian orientation, the equations $Ax=b$ are inconsistent. Then there must be a subset of rows of A, say A', such that $\text{rank}(A')<\text{rank}(A',b')$, where b' consists of the corresponding components of b. Notice that the number of rows in A' is at most $\text{rank}(A)+1$, i.e., polynomial in n. □

Corollary 3.3. *The following problem is in P: "Given a graph G, output a number k such that $k=\#M(G)$ if G has a Pfaffian orientation, and $k<\#M(G)$ otherwise", where $\#M(G)$ denotes the number of perfect matchings in G.*

Proof. As in Theorem 3.1, obtain an orientation for G, use it to substitute $+1$'s and -1's in the Tutte matrix of G, and output the square root of the determinant of the resulting matrix. If G has a Pfaffian orientation, each perfect matching will contribute $+1$ to this number, and it will be $\#M(G)$. Otherwise, some matchings will contribute $+1$ and others will contribute -1, making the resulting number strictly smaller than $\#M(G)$. □

Corollary 3.4. *The number of Pfaffian orientations in a graph is either zero, or a power of* 2.

Proof. Each solution to the equations $Ax=b$ corresponds to a Pfaffian orientation. Since the equations are over GF[2], the number of solutions is either zero or a power of 2. Notice that even though we do not know how to decide between these two cases, we can compute the exponent of 2 in polynomial time; it is simply $n-\text{rank}(A)$. □

Remark. Corollary 3.3 gives an alternative method for computing the number of perfect matchings in $K_{3,3}$-graphs. This method is quite different from the method used in [5,21]; however, at present we do not see clearly how these two methods relate to each other. By Corollary 3.4, we can enumerate all the orientations of $K_{3,3}$-free graphs; however, we do not see how to achieve this using the method in [5,21].

4. Bricks and braces

In this section, we address the complexity of the problems stated in Section 3. It is easy to see that the ideas in Theorem 2.1 yield the following:

Theorem 4.1. *The problem of testing if a given bipartite graph has a Pfaffian orientation is polynomial-time equivalent to* EVEN CYCLE.

As for general graphs, we resort to the decomposition of graphs given by Lovász [10] for finding the GF[2] rank of the set of perfect matchings. We may assume that $G(V,E)$ is *matching covered*, i.e., it is connected and every edge of G is in a perfect matching. A nonempty proper subset S of V determines a *cut*. Let $\nabla(S)$ denote the set of edges connecting S to $V-S$. S is a *tight cut* if every perfect matching of G contains exactly one edge of $\nabla(S)$. A tight cut is *nontrivial* if S and $V-S$ have at least two vertices each. Corresponding to a nontrivial tight cut S, two graphs can be obtained: One by contracting S to a single node, and the other by contracting $V-S$ to a single node. Lovász's procedure decomposes G by finding a nontrivial tight cut, creating the two contractions and continuing the process on the contractions until the pieces have no more nontrivial tight cuts. Lovász proves that the pieces are of two types, "bricks" and "braces", and the result of the decomposition is independent of the cuts chosen at each step. *Bricks* are bicritical, 3-connected graphs. A graph G is *bicritical* if $G-u-v$ has a perfect matching for any two vertices u and v. A bipartite graph $G(U,V,E)$ is called a *brace* if $|U|=|V|$, and each subset $X\subsetneq U$ with $0<|X|<|U|-1$ has at least $|X|+2$ neighbors in V.

Theorem 4.2. *The graph G has a Pfaffian orientation iff each of the bricks and* *braces in its decomposition have Pfaffian orientations.*

Proof. Suppose G is not a brick or a brace. Then G must have a nontrivial tight cut [10]. Let S determine such a cut in G, and let G_1 and G_2 be the corresponding contractions. Notice that by Lovász's procedure it is sufficient to prove that G has a Pfaffian orientation iff G_1 and G_2 have Pfaffian orientations.

Suppose G has a Pfaffian orientation $\vec{G}$. Let M be a perfect matching in G, and let (u, v) be the edge of M in $\nabla(S)$. Let $u \in S$ and $v \in V-S$. W.l.o.g. assume that in $\vec{G}$ this edge is directed from u to v. Let $S' \subseteq S$ be those vertices in S which have an edge incident from the set $\nabla(S)$. Pick any edge $(s,t) \neq (u,v)$ in $\nabla(S)$, $s \in S$ and $t \in V-S$. Since G is matching covered and S is a tight cut, there is a perfect matching N which contains (s,t) but not (u,v). The symmetric difference of M and N has an alternating cycle involving exactly the edges (s,t) and (u,v) from $\nabla(S)$. Therefore, for each vertex $s \in S'$, there is an even length alternating path from u to s.

Consider any such path from u to s, and compute the parity of the number of edges on this path directed along the direction of the path. Suppose there are two such paths with different parities. Consider the two alternating cycles formed using these paths and any one alternating path from t to v in $V-S$. One of these cycles must not be oddly oriented, giving a contradiction. Therefore the parity of every even length alternating path from u to s must be the same. This helps us partition S' into two sets S_e and S_o corresponding to even and odd parity respectively. Let G_1 be the graph obtained by contracting $V-S$ to v. To obtain a Pfaffian orientation for G_1, orient edges not incident on v as in $\vec{G}$, orient (u,v) from u to v, orient (w,v), $w \in S_e$ from w to v, and orient (w,v), $w \in S_o$, from v to w. A Pfaffian orientation for G_2 can similarly be obtained.

Next, given Pfaffian orientations $\vec{G}_1$ and $\vec{G}_2$ for G_1 and G_2, we want to obtain an orientation for G. First notice that we can switch the orientations of all edges incident at one vertex and still obtain a valid Pfaffian orientation. Therefore we may assume that the edge (u,v) is oriented from u to v in both $\vec{G}_1$ and $\vec{G}_2$. As before, consider even length alternating paths in S starting from u and obtain sets S_e and S_o. Similarly, consider even length alternating paths in $V-S$ starting from v, and obtain sets T_e and T_o. Direct the edges $E-\nabla(S)$ as in $\vec{G}_1$ or $\vec{G}_2$, and direct the edges in $\nabla(S)$ as follows:

(i) (u,v): from u to v,
(ii) (s,t), $s \in S_e$, $t \in T_e$: from s to t,
(iii) (s,t), $s \in S_e$, $t \in T_o$: from t to s,
(iv) (s,t), $s \in S_o$, $t \in T_e$: from t to s,
(v) (s,t), $s \in S_o$, $t \in T_o$: from s to t.

This will be a Pfaffian orientation for G. □

By Theorem 4.1, Pfaffian orienting braces reduces to EVEN CYCLE. On the other hand, characterizing the complexity of Pfaffian orienting bricks remains open. We finally use Theorem 4.1 to show that the following problem is also polynomial-time equivalent to EVEN CYCLE:

CYCLE SUM.

- *Instance:* A directed graph $G(V, E)$.
- *Question:* Is there a cycle in G which can be expressed as the GF[2] sum of an even number of cycles?
- *Comment:* Here we are expressing each cycle as an $|E|$-dimensional GF[2] vector.

Theorem 4.3. *CYCLE SUM is polynomial-time equivalent to EVEN CYCLE.*

Proof. Let $G(U, V, E)$ be a bipartite graph. We will reduce the problem of deciding whether G has a Pfaffian orientation to CYCLE SUM. Let M be a perfect matching in G. Orient the edges of G to obtain $\vec{G}$ as follows: Orient edges in M from U to V, and orient the remaining edges from V to U. Now assign a GF[2] variable to each edge and as in Theorem 3.1, write the system of equations, $Ax = b$, for oddly orienting the alternating cycles w.r.t. M. Notice that because of the special orientation given to G, b will be a vector of 1's. Therefore, $Ax = b$ will be inconsistent iff an odd number of rows of A add to 0. Next contract the edges of M in $\vec{G}$ to obtain a directed graph H. The alternating cycles w.r.t. M in G correspond to directed cycles in H. Furthermore, $Ax = b$ is inconsistent iff an odd number of cycles in H add to 0, or equivalently if H contains a cycle which can be written as the GF[2] sum of an even number of cycles.

The reduction from CYCLE SUM to the problem of determining whether a bipartite graph has a Pfaffian orientation is straightforward using the above-stated ideas. □

5. The complexity of testing if determinant equals permanent

Polya's idea of using determinants for computing permanents can be useful in a simpler setting: When the determinant equals the permanent. We study below the complexity of testing this. One can ask if the ideas of Polya, or Marcus and Minc can be used for computing permanents mod k, for a fixed integer k. A negative answer is given in [20], by showing that if k is not an exact power of 2, this problem is NP-hard under randomized reductions (in case k is a power of 2, a polynomial time algorithm is given in [19]). An extension of this problem is also studied.

Theorem 5.1. *Let A be an $n \times n$ integer matrix.*

(a) *If the entries of A are nonnegative integers, the problem* "$\det(A) = \mathrm{perm}(A)$?" *is polynomial-time equivalent to EVEN CYCLE. This holds even if A is a* 0–1 *matrix.*

(b) "$\det(A) = \mathrm{perm}(A)$?" *is NP-hard.*

(c) *If $k > 1$ is not an exact power of 2, the problem* "det(A) = perm(A)(mod k)?" *is NP-hard under randomized reductions, even if A is a* 0–1 *matrix.*

Proof. (a) Lemma 2.2 directly yields a reduction from EVEN CYCLE to the problem "det(A) = perm(A)?": Suppose G is the given digraph; add a self-loop on each vertex, and let A be the adjacency matrix of the resulting graph.

To show the reduction in the other direction, first notice that det(A) = perm(A) iff value(σ) = 0 for each odd permutation σ. Hence, it is sufficient to consider the problem for 0–1 matrices: Replace all nonzero entries of A by 1.

First check if value(σ) $\neq 0$ for some permutation σ, i.e., if $A(i, \sigma(i)) = 1$, $1 \leq i \leq n$, for some σ. This is simply the problem of determining if the bipartite graph, whose adjacency matrix is A, has a perfect matching. If not, det(A) = perm(A) = 0. Else let σ be this permutation. If σ is odd, det(A) $\neq$ perm(A). Else permute the columns of A by σ^{-1} to obtain matrix B whose diagonal entries are all 1. Since σ is even, det(B) = det(A). Also, perm(B) = perm(A). Let G be the digraph on n vertices whose adjacency matrix is B. By Lemma 2.2, det(B) = perm(B) iff G has no even cycle.

(b) We will reduce from the NP-hard problem "perm(A) = 0?" [19]. Suppose A is an $n \times n$ matrix. The reduced matrix B will be $(n+2) \times (n+2)$, with $b_{11} = b_{12} = b_{21} = b_{22} = 1$. The remaining entries in the first two rows and columns are 0. The rest of B, which is $n \times n$, is A. Clearly det(B) = 0 and perm(B) = 2 perm(A). Therefore det(B) = perm(B) iff perm(A) = 0.

(c) The proof is similar to [20, Corollary 3]. First reduce SAT to USAT. Let f be the formula obtained. Then using Valiant's [19] transformation, obtain a matrix A such that perm(A) equals $\#f \cdot 4^t$(mod k), where $\#f$ is the number of solutions f. Finally, obtain a matrix B by embedding A as in (b). Again det(B) $\equiv$ 0(mod k) and perm(B) $\equiv \#f \cdot 2 \cdot 4^t$(mod k). If $\#f = 1$, perm(B) $\not\equiv$ 0(mod k) since k is not an exact power of two. □

6. Discussion and open problems

Some of the ideas presented here, in particular Lemma 2.2, were carried further by Friedland [3] to partially settle a long open conjecture of Lovász [9]. Friedland shows that for $k \geq 7$, any k-regular digraph must contain an even cycle. Koh [7] had shown a 2-regular digraph which has no even cycle. The intermediate cases are still open. As such, this result does not yield an algorithmic schema for EVEN CYCLE since, for $k \geq 2$, the problem of checking whether a given digraph has a k-regular subdigraph is NP-complete.

Another open problem is to show that any 3-connected digraph must have an even cycle [9]. Currently only one 2-connected graph is known that does not have an even cycle; it is open whether this is the only such example (see [18]). Settling these open problems will yield a polynomial time algorithm for EVEN CYCLE. Towards this end, we leave the following open problem: $\exists c$ s.t. $\forall k \geq c$, every k-connected graph contains a 3-regular (respectively, 7-regular) subdigraph.

Acknowledgments

We wish to thank László Lovász and Peter Shor for valuable discussions, and Al Aho for his enthusiasm and support.

References

[1] C. Berge, Graphs and Hypergraphs (North-Holland, Amsterdam, 1973).
[2] R.A. Brualdi, Counting permutations with restricted positions: Permanents of (0, 1)-matrices, Linear Algebra Appl., to appear.
[3] S. Friedland, Every 7-regular digraph contains an even cycle, to appear.
[4] R.M. Karp, E. Upfal and A. Wigderson, Constructing a maximum matching is in random NC, Combinatorica 6(1) (1986) 35-48.
[5] P.W. Kasteleyn, Graph theory and crystal physics, in: F. Harary, ed., Graph Theory and Theoretical Physics (Academic Press, New York, 1967) 43-110.
[6] V. Klee, R. Ladner and R. Manber, Sign solvability revisited, Linear Algebra Appl. 59 (1984) 131-158.
[7] K.M. Koh, Even circuits in directed graphs and Lovász's conjecture, Bull. Malaysian Math. Soc. 7 (1976) 47-52.
[8] C.H.C. Little, An extension of Kasteleyn's method of enumerating the 1-factors of planar graphs, in: D. Holton, ed., Combinatorial Mathematics, Proceedings 2nd Australian Conference, Lecture Notes in Mathematics 403 (Springer, Berlin, 1974) 63-72.
[9] L. Lovász, Problem 2, in: M. Fiedler, ed., Recent Advances in Graph Theory, Proceedings Symposium Prague, 1974 (Academia Praha, Prague, 1975).
[10] L. Lovász, Matching structure and the matching lattice, J. Combin. Theory Ser. B 43 (1987) 187-222.
[11] L. Lovász and M. Plummer, Matching Theory (Academic Press, Budapest, 1986).
[12] R. Manber and J. Shao, On digraphs with the odd cycle property, J. Graph Theory 10 (1986) 155-165.
[13] M. Marcus and H. Minc, On the relation between the determinant and the permanent, Illinois J. Math. 5 (1961) 376-381.
[14] K. Mulmuley, U.V. Vazirani and V.V. Vazirani, Matching is as easy as matrix multiplication, Combinatorica 7(1) (1987) 105-113.
[15] G. Polya, Aufgabe 424, Arch. Math. Phys. (3) 20 (1913) 271.
[16] P. Seymour and C.T. Thomassen, Characterization of even directed graphs, J. Combin. Theory Ser. B 42 (1987) 36-45.
[17] C. Thomassen, Even cycles in directed graphs, European J. Combin. 6 (1985) 85-89.
[18] C. Thomassen, Sign-nonsingular matrices and even cycles in directed graphs, Linear Algebra Appl. 75 (1986) 27-41.
[19] L.G. Valiant, The complexity of computing the permanent, Theoret. Comput. Sci. 8 (1979) 189-201.
[20] L.G. Valiant and V.V. Vazirani, NP is as easy as detecting unique solutions, Theoret. Comput. Sci. 47 (1986) 85-93.
[21] V.V. Vazirani, NC algorithms for computing the number of perfect matchings in $K_{3,3}$-free graphs and related problems, Inform. and Computation 80(2) (1989) 152-164.