CS294 Partition Functions	**Fall 2023**

Lecture Note 3

Instructor: Alistair Sinclair

3.1 Exact Counting

As we have seen earlier, almost all interesting counting problems are #P-hard, so we cannot hope to find polynomial time algorithms that solve them exactly. In this lecture we give two famous counterexamples to this general rule. As we shall see, both examples make crucial use of *determinants*, which as is well known can be computed in polynomial time, and which turn out to be essentially the only tool available to us for problems of this kind.

3.1.1 Spanning Trees: The Matrix Tree Theorem

Consider the problem of counting spanning trees in a connected graph $G = (V, E)$. The following remarkable result, known as Kirchhoff's Matrix Tree Theorem[1], gives a simple exact algorithm for this problem. Throughout we will let n, m denote the numbers of vertices and edges respectively in G, and identify the vertex set with $\{1, 2, \ldots, n\}$.

Theorem 3.1. *The number of spanning trees of G is equal to the $(1,1)$ minor of the Laplacian of G.*

Recall that the *Laplacian* of G is the matrix $L = D - A$, where A is the adjacency matrix of G and D is the diagonal matrix whose (i,i) entry is the degree of vertex i. The (i,j) minor of a square matrix M is the determinant of the submatrix obtained by deleting the ith row and jth column of M. Since L has all row sums zero, it is a standard fact that all its (i,i) minors are equal (up to signs), so in fact the theorem applies to any such minor. (This is to be expected, since otherwise the theorem would appear to be asymmetric w.r.t. the choice of the index 1.)

Proof. Let B be the *incidence matrix* of G, i.e., the $|V| \times |E|$ matrix each of whose columns corresponds to a single edge $\{i, j\}$, where $i < j$, and contains an entry $+1$ in row i and -1 in row j. Observe **[Exercise!]** that $L = BB^T$. By the Cauchy-Binet formula, which generalizes the fact that $\det(AB) = \det(A)\det(B)$ for square matrices, we have

$$\det(L) = \sum_S \det(B_S)\det(B_S^T) = \sum_S \det(B_S)^2,$$

where the sum is over all subsets of columns S of cardinality n, and B_S is the corresponding $n \times n$ submatrix of B. Actually we want to work with the $(1,1)$ submatrix $L_{1,1}$ of L, so we instead write $L_{1,1} = B'B'^T$ where

[1] Although this theorem carries the name of Gustav Kirchhoff, in recognition of his seminal contributions to the understanding of electrical network theory [Kir47], it was actually first proved by Borchardt [Bor60]. Kirchhoff observed that spanning trees play a central role in electrical networks: for example, the effective resistance of an edge is equal to the probability that the edge appears in a random spanning tree.

B' is B with its first row removed, and obtain

$$\det(L_{1,1}) = \sum_S \det(B'_S)\det(B'^T_S) = \sum_S \det(B'_S)^2, \tag{3.1}$$

where now the sum is over subsets S of $n-1$ columns (edges).

The theorem will follow immediately from (3.1) and the following claims:

(i) *If the edges of S correspond to a spanning tree, then* $\det(B'_S) \in \{\pm 1\}$.

(ii) *If the edges of S do not correspond to a spanning tree, then* $\det(B'_S) = 0$.

To verify these claims, note that since each column of B' has exactly two non-zero entries (except for columns corresponding to edges incident to vertex 1, which have just one), any non-zero term in $\det(B'_S)$ corresponds to choosing one endpoint of each edge in S (where edges incident to 1 must choose the other endpoint). Moreover, these endpoints must consist precisely of the $n-1$ vertices of G (omitting vertex 1). Orienting each edge of S towards its chosen endpoint, we obtain a collection of oriented edges in which each vertex of G (except vertex 1) has exactly one incoming edge. It's not hard to check [**Exercise!**] that such a collection must consist of a (possibly trivial) directed tree rooted at vertex 1 plus a collection of zero or more disjoint cycles.

Now in case (i) the edges of S form a spanning tree, and we get exactly one non-zero term in $\det(B'_S)$, corresponding to orienting all the edges away from the root vertex 1.

In case (ii) we have at least one cycle, not including vertex 1. The columns of B'_S corresponding to the edges in this cycle each contain one $+1$ and one -1 entry, and the rows corresponding to the vertices of the cycle each contain two of these entries. But then it's not hard to check [**Exercise!**] that there is a $\{\pm 1\}$ linear combination of these columns that sums to zero, implying that $\det(B'_S) = 0$. □

Notes: Spanning trees are precisely the bases of a graphic matroid. (A graphic matroid has as its base set the edges of a graph G, and independent sets are forests in G. Spanning trees are maximal forests.) Later in the course we will see a randomized approximation algorithm for counting the bases of *any* matroid. This problem is in general #P-complete (the graphic case being an isolated "easy" instance).

The counting algorithm implicit in Theorem 3.1 immediately leads to an algorithm that samples spanning trees uniformly in time $O(mn^3)$, where n, m are the numbers of vertices and edges respectively in G [**Exercise:** How?]. This sampling problem has received much attention, partly due to its connections with electrical networks and random walks, and partly due to more recent algorithmic applications to graph sparsification. A more efficient use of the Matrix Tree Theorem reduces the running time to that of matrix multiplication [CMN96]. An alternative approach based on simulating random walks was developed in [Ald90, Bro89, Wil96], and achieves $O(mn)$ expected time. More recent developments based on linear algebra (see, e.g., [MST15]) led to a near-linear-time $O(m^{1+o(1)})$ algorithm [Sch18]. Very recently, as a byproduct of algorithms for sampling bases of matroids that we will see later in the course, Anari et al. were able to achieve running time $O(m \log n)$, which is optimal [ALOV20].

3.1.2 Perfect Matchings in Planar Graphs: Pfaffian Orientations

As we saw in the previous lecture, the problem of counting perfect matchings in a graph is #P-complete, even when the graph is bipartite. However, if the graph is *planar* then there is a remarkable polynomial time algorithm for this problem, known as the FKT algorithm after its inventors, Fisher, Kasteleyn and Temperley [Kas61, TF61]. Those gentlemen were mathematical physicists, and their interest in counting perfect matchings came from their correspondence with *dimer coverings*, the maximally dense case of the monomer-dimer model in which all sites are occupied by dimers.

Let $G = (V, E)$ be an undirected graph with $n = 2k$ vertices. We can identify each perfect matching in G with an (unordered) partition $\pi = \{\{i_1, j_1\}, \{i_2, j_2\}, \ldots, \{i_k, j_k\}\}$ of the vertex set $[2k]$ into pairs. The key ingredient in the algorithm is the *Pfaffian*[2].

Let $A = A_G$ be the skew-symmetric adjacency matrix of G, i.e., $A = \{a_{ij}\}$, where

$$a_{ij} := \begin{cases} +1 & \text{if } \{i,j\} \in E \text{ and } i < j; \\ -1 & \text{if } \{i,j\} \in E \text{ and } i > j; \\ 0 & \text{if } \{i,j\} \notin E. \end{cases}$$

The Pfaffian of A is defined as

$$\mathrm{Pf}(A) := \sum_{\pi = \{\{i_1,j_1\},\{i_2,j_2\},\ldots,\{i_k,j_k\}\}} \mathrm{sgn}(\pi) \prod_{\ell=1}^{k} a_{i_\ell, j_\ell}. \tag{3.2}$$

Here $\mathrm{sgn}(\pi)$ is the sign of the permutation $\begin{pmatrix} 1 & 2 & 3 & 4 & \cdots & n-1 & n \\ i_1 & j_1 & i_2 & j_2 & \cdots & i_k & j_k \end{pmatrix}$. Since there is a 1-1 correspondence between perfect matchings and non-zero terms in (3.2), we will write $\mathrm{Pf}(A) = \sum_M s(M)$, where the sum is over perfect matchings M and $s(M) := \mathrm{sgn}(\pi) \prod_{\ell=1}^{k} a_{i_\ell, j_\ell}$ is the contribution of M to the Pfaffian. Note that $s(M) \in \{\pm 1\}$ and, crucially, is independent of the order in which we list the pairs or the elements of the pairs! [**Exercise:** Why?]

The Pfaffian enumerates perfect matchings in G, but because they come with signs there could be cancellations. On the other hand, the signs make it easy to compute the Pfaffian, as the following standard fact due to Cayley [Cay49] shows:

Theorem 3.2. *For any $2k \times 2k$ skew-symmetric matrix A,* $\mathrm{Pf}(A)^2 = \det(A)$.

This means that we can compute $\mathrm{Pf}(A)$ just by computing a determinant and taking its square root. (Theorem 3.2 actually expresses the deeper fact that the determinant of a skew-symmetric matrix, viewed as a polynomial in its entries, is the square of another polynomial, given by the Pfaffian.)

Exercise: Prove Theorem 3.2. [Hint: View each term in the determinant (which corresponds to a permutation) as a *cycle cover* of G in the obvious way. By skew-symmetry, all covers containing odd cycles cancel. Even cycle covers correspond precisely to pairs of perfect matchings. Be sure to check the signs!]

Now we'd be done if we could somehow arrange that all terms $s(M)$ in $\mathrm{Pf}(A)$ have the *same sign*, so there are no cancellations between perfect matchings. We'll do this by *orienting* the edges of G (i.e., flipping the signs of entries a_{ij} and a_{ji} in A for some edges $\{i, j\}$), which will have the effect of changing the signs of some of the matchings. An orientation that ensures that $s(M)$ has the same sign for all M is called a *Pfaffian orientation*. It turns out that most graphs don't have a Pfaffian orientation, but planar graphs do (and we can find one easily).

Exercise: Let A_n be the standard skew-symmetric adjacency matrix of the complete graph K_n, with n even. Show that $\mathrm{Pf}(A_n) = 1$ (even though there are exponentially many perfect matchings!).

Exercise: Show that the complete bipartite graph $K_{3,3}$ has no Pfaffian orientation (and hence that the same holds for any graph that includes $K_{3,3}$ as a minor).

Here's a sufficient condition for an orientation to be Pfaffian:

[2] Named by Cayley after 19th century German mathematician Johann Friedrich Pfaff, in recognition of his related work on differential equations.

Lemma 3.3. *Suppose there is an orientation in which every even cycle C in G for which $G \setminus V(C)$ contains a perfect matching is oddly oriented, i.e., the number of edges with each of the two orientations around the cycle is odd. Then this orientation is Pfaffian.*

Proof. Consider any two distinct perfect matchings, M_1, M_2 in G. Our goal is to show that, under the hypothesis of the lemma, $s(M_1)s(M_2) = 1$, since this is the condition that the contributions of both matchings to $\mathrm{Pf}(A)$ have the same sign. Now $M_1 \sqcup M_2$ (disjoint union) consists of a collection of disjoint even alternating cycles. Consider any non-trivial cycle C. Clearly $G \setminus V(C)$ contains a perfect matching, so by the hypothesis C is oddly oriented. Note first that we can relabel the vertices of G so that C is the cycle $1 \to 2 \to \ldots \to \ell \to 1$ for some even ℓ; this relabelling is equivalent to applying a permutation to the vertex set, which has the same effect on both $s(M_1)$ and $s(M_2)$. Second, note that we can represent the portions of M_1, M_2 respectively that lie on C by the (partial) pairings $\pi_1 = \{\{1,2\},\{3,4\},\ldots,\{\ell-1,\ell\}\}$ and $\pi_2 = \{\{2,3\},\{4,5\},\ldots,\{\ell,1\}\}$. (As we saw earlier, this choice of pairing has no effect on $s(M_1)$ or $s(M_2)$.) With this choice, the contribution of this cycle to the product $s(M_1)s(M_2)$ is

$$\mathrm{sgn}(\pi_1)\mathrm{sgn}(\pi_2)\prod_{i=1}^{\ell} a_{i,i+1}, \tag{3.3}$$

where we interpret the index $\ell + 1$ as 1. Now note that $\mathrm{sgn}(\pi_1) = +1$ (the identity permutation) and $\mathrm{sgn}(\pi_2) = -1$ (a single cycle of even length). Also, since C is oddly oriented, among all the oriented edges $a_{i,i+1}$ around C, an odd number of them are negative. Putting this together ensures that the product in (3.3) is equal to $+1$. We can repeat the same argument to show that the contribution from every cycle is $+1$, and hence $s(M_1)s(M_2) = +1$, as required. ☐

In preparation for showing that every planar graph has a Pfaffian orientation, we need to simplify the condition in Lemma 3.3 in the planar graph setting.

Lemma 3.4. *Suppose a planar drawing of graph G has an orientation in which the number of clockwise edges around every (not necessarily even) face (except possibly the outer face) is odd. Then this orientation is Pfaffian.*

Proof. Consider any even cycle C in G such that $G \setminus V(C)$ contains a perfect matching. Note that this implies that the number of vertices strictly inside C is necessarily even. Let c be the number of edges on C, c^+ of which are clockwise: we need to show that c^+ is odd. Let G' be the subgraph of G where all vertices and edges outside C have been removed, and suppose G' has $v + c$ vertices, $e + c$ edges and $f + 1$ faces. (Thus G' has v vertices strictly inside C, e edges not including those of C, and f interior faces.) Applying Euler's formula to G' gives

$$(v + c) - (e + c) + (f + 1) = 2 \quad \Rightarrow \quad v - e + f = 1. \tag{3.4}$$

Label the interior faces of G' as $F_1, \ldots, F_f$, and let the number of boundary edges of F_i be c_i, c_i^+ of which are oriented clockwise. Then since each edge in G' except for anti-clockwise edges of C appears clockwise in exactly one interior face, $\sum_i c_i^+ = e + c^+$. But also, since by hypothesis each c_i^+ is odd, we have $f \equiv \sum_i c_i^+ \pmod 2$, so that $f \equiv e + c^+ \pmod 2$. Plugging this into (3.4), and recalling that v is even, gives

$$v + c^+ \equiv 1 \pmod 2 \quad \Rightarrow \quad c^+ \equiv 1 \pmod 2.$$

Hence by Lemma 3.3 the orientation is Pfaffian. ☐

The final step is to use the condition in Lemma 3.4 to construct a Pfaffian orientation in any planar graph.

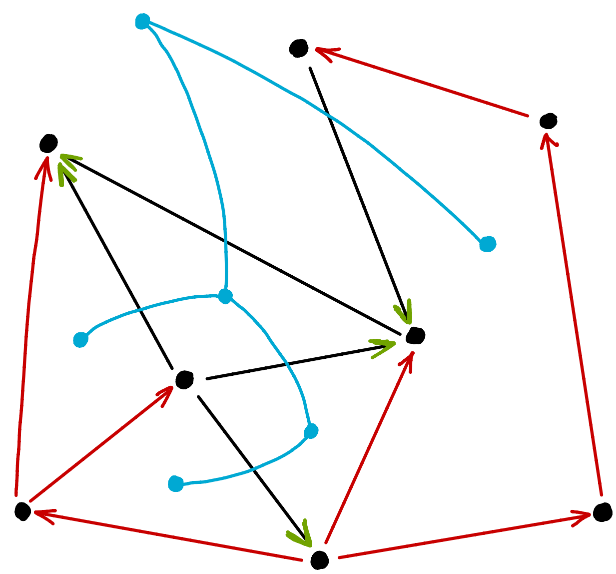

Figure 3.1: Example of construction in proof of Lemma 3.5 on an 8-vertex planar graph G (red and black edges). The spanning tree T and its orientation are shown in red, the dual spanning tree T' in blue, and the orientation of the remaining edges of G with green arrows.

Lemma 3.5. *Every connected planar graph G has a Pfaffian orientation, and one can be found in polynomial time.*

Proof. Take any spanning tree T of G and orient its edges arbitrarily. Now construct a subgraph T' of the planar dual of G by connecting any two adjacent faces that are separated by an edge of G that is not in T. Note that T' must also be a tree [**Exercise:** Why?]. Now, starting from a leaf of T', delete the leaf and its edge and orient the corresponding edge of G so that the face containing the leaf has an odd number of clockwise edges. (See Figure 3.1.) Continue until all edges have been oriented and T' is empty. The resulting orientation satisfies the condition in Lemma 3.4 and hence is Pfaffian. Clearly these operations can all be carried out in polynomial time. □

Combining the above lemmas with Theorem 3.2, we have proved:

Corollary 3.6. *There is a polynomial time algorithm for counting the number of perfect matchings exactly in any planar graph.*

Exercise: Write down the 8×8 skew-symmetric matrix corresponding to the Pfaffian orientation in Figure 3.1. Use (e.g.) Mathematica to compute the determinant of this matrix: you should get 16, implying that the number of perfect matchings in the original planar graph is 4. Check that the graph does indeed have exactly 4 perfect matchings.

Notes: As we saw in the previous lecture, counting *all* matchings in a graph is also #P-complete. Perhaps surprisingly, unlike perfect matchings, this problem remains #P-complete even when restricted to planar graphs [Jer87].

As mentioned earlier, the determinantal techniques used for the above two counting algorithms, and variants of them, are apparently the only somewhat widely applicable algorithmic tricks we have for exact counting. At root, these algorithms make use of "magical" cancellations of positive and negative terms in the determinant (a highly non-trivial sum of exponentially many terms that we can equally magically compute). The so-called "holographic algorithms" framework, introduced by Valiant [Val04] (see also [CC18] for a comprehensive account of more recent developments) aims to capture the full generality and universality of this technique. For a statistical physics view of exact algorithms, including some more examples, see the classic book by Baxter [Bax08].

References

[Ald90] D.J. Aldous. A random walk construction of uniform spanning trees and uniform labelled trees. *SIAM Journal on Discrete Mathematics*, 3:450–465, 1990.

[ALOV20] N. Anari, K. Liu, S. Oveis Gharan, and C. Vinzant. Log-concave polynomials IV: Approximate exchange, tight mixing times, and faster sampling of spanning trees. *arXiv, abs/2004.07220*, 2020.

[Bax08] R.J. Baxter. *Exactly solved models in statistical mechanics.* Dover, 2008.

[Bor60] C.W. Borchardt. Über eine Interpolationsformel für eine Art symmetrischer Functionen und über deren Anwendung. *Abh. der Preuß. Akad. der Wiss.*, pages 1–20, 1860.

[Bro89] A. Broder. Generating random spanning trees. *Proceedings of the 30th IEEE FOCS*, pages 442–447, 1989.

[Cay49] A. Cayley. Sur les déterminants gauches. *Journal für die reine und angewandte Mathematik*, 38:93–96, 1949.

[CC18] J.-Y. Cai and X. Chen. *Complexity Dichotomies for Counting Problems, Vol I.* Cambridge University Press, 2018.

[CMN96] C.J. Colbourn, W.J. Myrvold, and E. Neufeld. Two algorithms for unranking arborescences. *Journal of Algorithms*, 20:268–281, 1996.

[Jer87] M.R. Jerrum. Two-dimensional monomer-dimer systems are computationally intractable. *Journal of Statistical Physics*, 48:121–134, 1987. Erratum in *Journal of Statistical Physics*, 59:1087–1088, 1990.

[Kas61] P.W. Kasteleyn. The statistics of dimers on a lattice. I. The number of dimer arrangements on a quadratic lattice. *Physica*, 27:1209–1225, 1961.

[Kir47] G. Kirchoff. Über die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Verteilung galvanischer Ströme geführt wird. *Ann. Phys. Chem.*, 72:497–508, 1847.

[MST15] A. Mądry, D. Straszak, and J. Tarnawski. Fast generation of random spanning trees and the effective resistance metric. *Proceedings of ACM-SIAM SODA*, pages 2019–2036, 2015.

[Sch18] A. Schild. An almost-linear time algorithm for uniform random spanning tree generation. *Proceedings of the 50th ACM-SIAM SODA*, pages 214–227, 2018.

[TF61] H.N.V. Temperley and M.E. Fisher. Dimer problem in statistical mechanics—An exact result. *Philosophical Magazine*, 6:1061–1063, 1961.

[Val04] L.G. Valiant. Holographic algorithms. *Proceedings of IEEE FOCS*, pages 306–315, 2004.

[Wil96] D.B. Wilson. Generating random spanning trees more quickly than the cover time. *Proceedings of the 28th Annual ACM STOC*, pages 296–303, 1996.