Efficient approximation of the permanent

Bogdan Grechuk • 3 Apr 2026

This post continues my series on favorite theorems of the 21st century, in each year and each of the seven major subject categories. For an overview of the categories and my previous selections, see this earlier post. My choice for 2001 in *Algorithms and Complexity* is the algorithm of Jerrum, Sinclair and Vigoda (Jerrum et al. 2004) that can efficiently approximate the permanent of a matrix with non-negative entries to any given accuracy.

In this previous blog post about the *Algorithms and Complexity* category, we considered only decision problems, that is, problems with a "Yes" or "No" answer. For example, we discussed the k-*SAT* problem, where the task is to decide whether there exists a truth assignment to the variables that satisfies all clauses. More generally, we may ask to count *how many* such assignments exist. As another example, there is a decision problem of determining whether there exists a perfect matching in a given graph G, and the corresponding counting problem asks how many such matchings there are.

Note that the k-SAT problem is in the class NP because any satisfying assignment serves as a "certificate" that the problem has a "Yes" answer, and the correctness of this certificate can be verified in polynomial time. The counting version asks us to count such certificates. Similarly, any perfect matching in a graph G is a certificate that the graph has a perfect matching. More generally, for any problem in class NP, we may formulate a counting problem asking for the number of corresponding certificates. The class of all such counting problems is denoted $\#P$. The class of counting problems solvable in polynomial time is called FP, and, similarly to P$\neq$NP, there is a widely believed conjecture that $\#P \not\subseteq$FP. We say that a counting problem is $\#P$-*hard* if any problem in $\#P$ reduces to it in polynomial time, and, similarly to NP-hardness for decision problems, $\#P$-hardness is convincing evidence that a counting problem is intractable.

Amazingly, some $\#P$-hard problems can be efficiently solved approximately to *any* given accuracy. A famous example is the problem of computing the permanent. Let A be an $n \times n$ matrix with integer entries $a(i,j)$, $i = 1, \dots, n$, $j = 1, \dots, n$. The *permanent* of A is

$$\mathrm{per}(A) = \sum_{\sigma} \prod_{i=1}^{n} a(i, \sigma(i)),$$

where the sum is over all permutations σ of $\{1, 2, \ldots, n\}$. The evaluation of the permanent of a matrix is an old and well-studied problem going back to at least the 1812 memoirs of Binet and Cauchy. In 1979, Valiant (Valiant 1979) proved that exact computation of the permanent is $\#P$-hard, even if all $a(i, j)$ are either 0 or 1. However, Jerrum, Sinclair and Vigoda (Jerrum et al. 2004) presented in 2001 an algorithm which, for any matrix A with non-negative entries, computes (with high probability) an arbitrarily close approximation to its permanent in time that depends polynomially on the input size and the desired error. Given Valiant's result, this is essentially the best we could hope for.

In general, a *polynomial-time approximation scheme* (PTAS) is an algorithm which, given a parameter $\epsilon > 0$ and an instance of an optimization problem of length n, produces in time $P_\epsilon(n)$ a solution which differs from the optimal one by a factor of $1 + \epsilon$. Here, P_ϵ is a polynomial which depends on ϵ. If the running time is bounded by a polynomial in both n and ϵ^{-1}, we say that the problem has a *fully polynomial-time approximation scheme* (FPTAS). If the corresponding algorithm is randomized, we call it a *fully polynomial randomized approximation scheme* (FPRAS). More concretely, a FPRAS for the permanent computation is a randomized algorithm which, given an $n \times n$ matrix A and $\epsilon \in (0, 1]$ as an input, runs in time polynomial in n and ϵ^{-1}, and outputs a (random) number Z such that

$$P[\exp(-\epsilon)Z \leq \mathrm{per}(A) \leq \exp(\epsilon)Z] \geq \frac{3}{4}.$$

The probability $\frac{3}{4}$ here can be made arbitrarily close to 1 by running the algorithm several times and averaging the outputs.

Theorem 1 *There exists a fully polynomial randomized approximation scheme (FPRAS) for the permanent computation of an arbitrary $n \times n$ matrix A with non-negative entries.*

The proof of Theorem 1 first treats the important special case when all entries $a(i, j)$ are either 0 or 1, and then extends to the general case of arbitrary non-negative entries. The starting point is the 1986 observation of Jerrum, Valiant and Vazirani (Jerrum et al. 1986) that the approximate permanent computation for 0-1 matrices is equivalent to sampling perfect matchings from a bipartite graph G almost uniformly at random. The latter problem is solved using the "Markov chain Monte Carlo" method, first proposed by Broder (Broder 1986) in 1986, which considers a random walk on the space of all perfect and "near-perfect" matchings (that is, matchings with 2 unmatched vertices) on G, such that, no matter where we start the walk, its position after polynomially many

steps is nearly uniformly random. The method works unless the number of perfect matchings is negligible compared to the number of near-perfect ones. Jerrum, Sinclair and Vigoda (Jerrum et al. 2004) developed a modified version of this algorithm, in which each near-perfect matching gets a weight in such a way that the number of weighted near-perfect matchings becomes not much higher than the number of perfect matchings.

References

Broder, Andrei Z. 1986. “How Hard Is It to Marry at Random? (On the Approximation of the Permanent).” *Proceedings of the eighteenth annual ACM symposium on the Theory of Computing*, 50–58.

Jerrum, Mark R., Leslie G. Valiant, and Vijay V. Vazirani. 1986. “Random Generation of Combinatorial Structures from a Uniform Distribution.” *Theoret. Comput. Sci.* 43: 169–88.

Jerrum, Mark, Alistair Sinclair, and Eric Vigoda. 2004. “A Polynomial-Time Approximation Algorithm for the Permanent of a Matrix with Nonnegative Entries.” *J. ACM* 51 (4): 671–97.

Valiant, Leslie G. 1979. “The Complexity of Computing the Permanent.” *Theoretical Computer Science* 8 (2): 189–201.