Estimating the Permanent by Nesting Importance Sampling

Juha Harviainen [1] **Mikko Koivisto** [1]

Abstract

Sequential importance sampling (SIS) is one of the prominent methods for estimating high-dimensional integrals. For example, it is empirically the most efficient method known for estimating the permanent of nonnegative matrices, a notorious problem with numerous applications in computer science, statistics, and other fields. Unfortunately, SIS typically fails to provide accuracy guarantees due to difficulties in bounding the variance of the importance weights; for estimating the permanent with accuracy guarantees, the most efficient practical methods known are based on rejection sampling. Taking the best of both worlds, we give a variant of SIS, in which sampling is proportional to the upper bound used in rejection sampling. We show that this method is provably more efficient than its rejection sampling counterpart, particularly in high accuracy regimes. On estimating the permanent, we empirically obtain up to two orders-of-magnitude speedups over a state-of-the-art rejection sampling method.

1. Introduction

Importance sampling is a popular Monte Carlo method for approximating high-dimensional integrals (Robert & Casella, 2004, Ch. 3.3, 14). There, we draw samples from a *proposal distribution* and *reweight* them to have the expected value match the value of the integral. The variance of the estimator depends on the quality of the proposal distribution: the closer it is to being proportional to the weights (unnormalized probability masses or densities), the smaller the variance.

Many problems—like counting the number of linear extensions of a given partial order (Jensen & Beichl, 2020) or graphs with a given degree sequence (Bayati et al., 2010)—exhibit self-reducible nature. Essentially, the search space can be partitioned in a way that allows solving the original instance by solving smaller instances of the same problem. Correspondingly, *sequential importance sampling* (SIS) uses a proposal distribution where some randomized choices are made, after which the sample is completed by drawing the rest of the sample from the smaller instance.

In numerical integration, *accuracy guarantees* are desirable. Formally, we want an approximation scheme that computes an (ϵ, δ)*-approximation* (see, e.g., Ermon et al., 2013; Chakraborty et al., 2014) for any given $\epsilon, \delta > 0$: an estimate that has a relative error at most ϵ with probability at least $1 - \delta$. The scheme thus needs a *stopping rule* describing the requirements for the given ϵ and δ that must be satisfied until the algorithm can stop drawing new samples. A weak stopping rule leads to drawing unnecessary samples whereas finding an optimal one can be difficult. The main challenge with importance sampling is that no good stopping rules for it are known in general.

As a special case, asymptotically optimal stopping rules are known for random variables distributed in $[0, 1]$ (Dagum et al., 2000; Huber, 2006; Mnih et al., 2008). However, applying these rules for SIS requires bounding the importance weights of the samples. Again, this is often hard as one needs both a proposal distribution with small importance weights and a good upper bound for them that is either obtained analytically or efficiently computable.

On the other hand, *rejection sampling* is an exact sampling method that is usable for integration with accuracy guarantees. It is based on upper bounds for the weights of subsets of the sample space and has optimal stopping rules (Dagum et al., 2000; Huber, 2017). The method repeatedly draws samples and either *accepts* or *rejects* them to ensure that the accepted samples come proportionally to their weights. The algorithm stops after a certain number of accepted samples is drawn.

Of particular interest to us is the #P-hard problem of computing the permanent of a matrix (Valiant, 1979), which has numerous applications in, for example, probability and statistics (Bapat, 1990), multi-target tracking (Uhlmann, 2004; Kuck et al., 2019), physics (Beichl & Sullivan, 1999; Aaronson & Arkhipov, 2011), and constraint satisfaction problems (Bianco et al., 2019). Until now, the most ef-

[1]Department of Computer Science, University of Helsinki, Helsinki, Finland. Correspondence to: Juha Harviainen <juha.harviainen@helsinki.fi>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

ficient known method for approximating this sum over weighted permutations with accuracy guarantees has been self-reducible rejection sampling (Huber, 2006; Kuck et al., 2019; Harviainen & Koivisto, 2023). In practice, however, SIS schemes (Smith & Dawkins, 2001; Chen & Liu, 2007; Alimohammadi et al., 2023) converge much faster empirically. While some upper bounds are known for the number of samples needed by SIS for computing an (ϵ, δ)-approximation (Diaconis, 2019; Diaconis & Kolesnik, 2021; Alimohammadi et al., 2023), they are so large that applying them directly would lead to poor performance.

Our main question is whether one can incorporate the upper bounds from rejection sampling into SIS to bound the importance weights, enabling efficient stopping rules. Related ideas have shown promise in the dynamic importance sampling method for inference in graphical models (Lou et al., 2017), in which one iteratively expands a search tree and uses a proposal distribution based on upper bounds for unsolved subproblems.

The rejection sampling schemes for the permanent of a nonnegative matrix rely on upper bounds that are *nesting* (Huber, 2006; Kuck et al., 2019): a property that the bound is monotone in a sense over a partition tree of the search space. In this paper, we prove an upper bound for the importance weights when nesting bounds are used to guide SIS, giving a positive answer to the question. The running time of the presented *nesting importance sampling* (NIS) scheme depends on the tightness of the used bound, and thus combining it with methods that yield tighter nesting bounds, such as deep bounds (Harviainen et al., 2021), can be beneficial. Note that despite the similarity of the names, NIS is unrelated to the nested sampling method introduced by Skilling (2004).

To demonstrate the power of the presented method, we evaluate our novel NIS scheme against the state-of-the-art rejection sampler (Harviainen & Koivisto, 2023) on approximating the permanent. We compare the time usage for computing estimates for both synthetic and real-world instances with varying sizes and accuracies. We observe that NIS is consistently better and obtains up to two orders-of-magnitude speedups.

Even though we here focus on the permanent, for which good nesting upper bounds are already known, the presented idea is general: it is applicable to any self-reducible problem where we have nesting bounds. Thus, we believe NIS to become suitable for obtaining accuracy guarantees also in other integration problems once appropriate upper bounds are discovered. Notably, weakening a non-nesting bound slighly can be sufficient for transforming it into a nesting one (Huber, 2006), providing one way of obtaining such bounds.

2. Nesting Importance Sampling

We start by covering the used terminology and notation. Then, we introduce nesting importance sampling and bound the number of samples required by it.

2.1. Sequential Importance Sampling

Let Ω be a countable set of objects, and associate a nonnegative weight $a(x)$ for each object $x \in \Omega$. Then, a defines an unnormalized probability distribution whose normalizing constant is the weighted sum

$$a(\Omega) := \sum_{x \in \Omega} a(x).$$

In *importance sampling*, we draw samples $x_1, x_2, \ldots, x_N$ from a *proposal distribution* $q(x)$ for some N. Then, each ratio $a(x_i)/q(x_i)$ is an unbiased estimator of $a(\Omega)$ and so is their mean

$$\frac{1}{N} \sum_{i=1}^{N} \frac{a(x_i)}{q(x_i)}.$$

Explicitly maintaining all probabilities $q(x)$ becomes infeasible when $|\Omega|$ is large. Fortunately, sampling can often be performed in a *self-reducible* fashion: We split the sampling process into several parts by first drawing a part of the sample such that completing the missing part reduces into drawing a sample from a smaller instance. An alternative way of thinking this is repeatedly partitioning the set of objects and picking one of the sets until we are left with a singleton. Denote this sequence of subsets of Ω by

$$\Omega = \Omega_0 \supseteq \Omega_1 \supseteq \cdots \supseteq \Omega_\ell = \{x\}.$$

Then, *sequential importance sampling* (SIS) is defined by a proposal distribution that factorizes as

$$q(x) = q(\{x\}) = \prod_{j=1}^{\ell} q\left(\Omega_j \mid \Omega_{j-1}\right),$$

where $q\left(\Omega_j \mid \Omega_{j-1}\right)$ is the probability of choosing the set Ω_j from a partition of Ω_{j-1}. Again, this gives an unbiased estimator $a(x)/q(x)$. Algorithm 1 describes the method with the initial argument being $\Omega_0 = \Omega$.

The more samples we draw, the better estimates we get, but at which rate? Suppose we want the estimate to have a relative error at most ϵ. We say that the output of an estimator is an (ϵ, δ)-approximation of $a(\Omega)$ if its relative error exceeds ϵ with probability at most δ. Let Y be a random variable equal to $a(X)/q(X)$ for a random variable X drawn from q. Then, computing an (ϵ, δ)-approximation of $a(\Omega)$ requires

$$O\left(\frac{\mathbb{E}[Y^2]}{\mathbb{E}[Y]^2} \epsilon^{-2} \delta^{-1}\right)$$

Algorithm 1 Sequential importance sampling

```
Input: A set of objects $\Omega_{j-1}$.
Output: Estimate $a(x)/q(x)$ with $x \sim q$.
if $\Omega_{j-1} = \{x\}$ then
    return $a(x)$
end if
Partition $\Omega_{j-1}$ into $\Omega_j^1, \Omega_j^2, \ldots, \Omega_j^m$
Pick $k \in [m]$ with probabilities $q\left(\Omega_j^k \mid \Omega_{j-1}\right)$
Call Algorithm 1 on $\Omega_j^k$ and let $w$ be its output
return $w/q\left(\Omega_j^k \mid \Omega_{j-1}\right)$
```

samples by Chebyshev's inequality. Applying the median trick by computing an $(\epsilon, 1/4)$-approximation $O(\log \delta^{-1})$ times and taking their median lets us replace δ^{-1} by $\log \delta^{-1}$ in the total number of samples needed.

In an ideal case, we would have $q(x) = a(x)/a(\Omega)$ so that one sample would be sufficient for computing $a(\Omega)$ exactly. Unfortunately, sampling from such a distribution and computing $q(x)$ is hard without knowing the normalizing constants beforehand. Thus, we instead hope to discover a proposal distribution that approximates this ideal distribution sufficiently well. Problematically, the *critical ratio* $\mathbb{E}[Y^2]/\mathbb{E}[Y]^2$ of q is typically unknown and so we need a *stopping rule* that decides if we have achieved the desired accuracy and can stop drawing new samples based on the past draws.

2.2. Nesting

For the proposal distribution q to approximate each $a(\Omega_j)$ well, it should utilize the information available about them. In a sense, this is what *upper bounds* do: when given information about $a(\Omega_j)$, we compute an upper bound $U(\Omega_j) \geq a(\Omega_j)$ that provides a rough approximation of $a(\Omega_j)$. Could we then show a boundable critical ratio for a proposal distribution q where

$$q(\Omega_j^k \mid \Omega_{j-1}) = U(\Omega_j^k)\Big/\sum_{i=1}^{m} U(\Omega_j^i) \tag{1}$$

for the partition $\Omega_j^1, \Omega_j^2, \ldots, \Omega_j^m$ of Ω_{j-1}?

We will next answer the question in the affirmative for a family of upper bounds that are *nesting*, a type of monotonicity property. Before we define the nesting property, equip Ω with a fixed hierarchical collection of partitions, and say that Ω_j is *reachable* if we have a sequence of sets

$$\Omega = \Omega_0 \supseteq \Omega_1 \supseteq \cdots \supseteq \Omega_j$$

where Ω_i is one of the sets in the partition of Ω_{i-1} for all $i \in \{1, 2, \ldots, j\}$. We require that each set $\{x\}$ is reachable.

Definition 2.1. An upper bound U is nesting (with respect to a hierarchical collection of partitions) if for any reachable Ω_{j-1} and its partition $\Omega_j^1, \Omega_j^2, \ldots, \Omega_j^m$ we have

$$U(\Omega_{j-1}) \geq \sum_{i=1}^{m} U(\Omega_j^i)\,.$$

The use of nesting bounds stems from self-reducible rejection sampling literature for the permanent, where they ensure the correct distribution of the samples (Huber, 2006; Huber & Law, 2008; Harviainen & Koivisto, 2023). There, the term "nesting" was coined by Kuck et al. (2019), although they present it in a more general setting than just the permanent.

We call a SIS scheme that uses a nesting upper bound a *nesting importance sampling* (NIS) scheme. We proceed to analyzing the properties of the samples drawn by a NIS scheme, culminating in giving an upper bound for the number of samples needed for obtaining an (ϵ, δ)-approximation.

2.3. Guarantees

Consider the random variable

$$Y \coloneqq Y(x) = \frac{a(x)}{q(x)}$$

where q is defined by Equation (1) and a nesting upper bound U. We show an upper bound for the value of Y, leading to a bounded critical ratio:

Lemma 2.2. *Let $x \in \Omega$ such that $a(x) > 0$. Then, we have $0 < Y(x) \leq U(\Omega)$.*

Proof. Let $a(x) > 0$. Then, for the sequence of sets

$$\Omega = \Omega_0 \supseteq \Omega_1 \supseteq \cdots \supseteq \Omega_\ell = \{x\}.$$

for reaching $\{x\}$ we have that

$$U(\Omega_j) \geq a(\Omega_j) \geq a(x) > 0\,.$$

This guarantees that both the numerator and the denominator in Equation (1) are positive, and thus $q(\Omega_j \mid \Omega_{j-1}) > 0$.

For the upper bound, we have by definition that

$$\begin{aligned} Y(x) &= a(x)\Big/\prod_{j=1}^{\ell} q(\Omega_j \mid \Omega_{j-1}) \\ &= a(x)\prod_{j=1}^{\ell}\sum_{i=1}^{m} U(\Omega_j^i)/U(\Omega_j)\,, \end{aligned}$$

where $m = m(\Omega_{j-1})$ is a function of Ω_{j-1}. Apply the nesting property to obtain the upper bound

$$\begin{aligned} &a(x) \prod_{j=1}^{\ell} U(\Omega_{j-1})/U(\Omega_j) \\ &= a(x) \cdot \frac{U(\Omega_0)}{U(\Omega_\ell)} \\ &\leq U(\Omega) . \end{aligned}$$ □

Consequently,

$$\mathbb{E}[Y^2] \leq U(\Omega) \cdot \mathbb{E}[Y] = U(\Omega) \cdot a(\Omega),$$

resulting in the following corollary:

Corollary 2.3. *The critical ratio of Y is at most $U(\Omega)/a(\Omega)$.*

It should be noted that since the tightness of our upper bound for Y depends on how small the ratios

$$\sum_{i=1}^{m} U(\Omega_j^i)/U(\Omega_{j-1})$$

are, the critical ratio may be significantly smaller in practice.

The key fact we apply is that asymptotically optimal stopping rules are known for estimating the expected value of a $[0, 1]$-valued random variable, and $Y/U(\Omega)$ is such a variable.

Lemma 2.4 (Dagum et al., 2000)**.** *Let Y be a $[0, 1]$-valued random variable. Then, an (ϵ, δ)-approximation of $\mathbb{E}[Y]$ can be computed using*

$$O\left(\left(\frac{\mathbb{E}[Y^2]}{\mathbb{E}[Y]^2} + \frac{\epsilon}{\mathbb{E}[Y]}\right) \epsilon^{-2} \log \delta^{-1}\right)$$

samples of Y on average.

Crucially, the asymptotic bound holds even if the critical ratio is unknown. Huber & Jones (2019) have presented an improved approximation scheme that has the same asymptotic behavior as that of Dagum et al. (2000), but requires at least 60% fewer samples for small ϵ and even less in practice. The implementation of the scheme is rather involved, and so we refer the reader to their article for details. Bernstein's inequality gives an alternative near-optimal stopping rule (Mnih et al., 2008). Applying either of the stopping rules of Dagum et al. (2000) and Huber & Jones (2019) gives us a bound for the expected number of samples:

Theorem 2.5. *Let Y be a random variable obtained as an output of Algorithm 1 with a proposal distribution equipped with a nesting upper bound U. Then, an (ϵ, δ)-approximation of $\mathbb{E}[Y] = a(\Omega)$ can be computed using*

$$O\left(\left(\frac{\mathbb{E}[Y^2]}{\mathbb{E}[Y]^2} + \frac{U(\Omega) \cdot \epsilon}{a(\Omega)}\right) \epsilon^{-2} \log \delta^{-1}\right)$$

samples of Y on average.

Proof. Divide each Y by $U(\Omega)$ to obtain a $[0, 1]$-valued random variable. The number of samples follows from the fact that $\mathbb{E}[Y/U(\Omega)] = a(\Omega)/U(\Omega)$ whereas the critical ratio remains the same despite scaling Y. □

Since the critical ratio of Y is bounded by $U(\Omega)/a(\Omega)$, the expected number of samples needed is at worst

$$O\big(U(\Omega)/a(\Omega) \cdot \epsilon^{-2} \log \delta^{-1}\big) .$$

On the other hand, the term $U(\Omega) \cdot \epsilon/a(\Omega)$ may dominate the critical ratio in practice, leading to a complexity

$$O\big(U(\Omega)/a(\Omega) \cdot \epsilon^{-1} \log \delta^{-1}\big) .$$

In contrast, the expected number of samples required by rejection sampling is

$$O\big(U(\Omega)/a(\Omega) \cdot \epsilon^{-2} \log \delta^{-1}\big)$$

as we will discuss in Section 4.4 in the context of approximating the permanent (Huber, 2006).

3. Estimating the Permanent

We start by recalling some properties of the permanent of a matrix and how SIS is used in estimating it. Then, we present a NIS scheme for the problem.

3.1. The Permanent of a Matrix

One application of SIS is to approximate the permanent of an $n \times n$ matrix $A = (a_{ij})$ defined as the sum

$$\operatorname{per} A := \sum_{\sigma} a(\sigma), \quad \text{where} \quad a(\sigma) := \prod_{i=1}^{n} a_{\sigma(i),i} ,$$

over permutations of $[n] := \{1, 2, \ldots, n\}$.

The exact computation of the permanent is #P-hard (Valiant, 1979), with the fastest algorithms taking exponential time in the worst case (Ryser, 1963; Chakraborty et al., 2019). Even though permanents of nonnegative matrices are estimable in arbitrary precision in a fully polynomial time with rapidly mixing Markov chains (Jerrum et al., 2004; Bezáková et al., 2008), the polynomials have high degree and large constant factors (Newman & Vardi, 2020), making them infeasible. Thus, alternative methods are needed for approximation in practice. Here, we focus on developing an approach based on NIS—related techniques are discussed in Section 4.

The Laplace expansion enables computation of the permanent in a self-reducible form: letting A_{ij} be the submatrix of A with the ith row and the jth column removed, we have

$$\operatorname{per} A = \sum_{i=1}^{n} a_{ij} \operatorname{per} A_{ij}$$

for any column $j \in [n]$.

Many SIS-based approaches (Beichl & Sullivan, 1999; Chen & Liu, 2007; Alimohammadi et al., 2023) for approximating the permanent rely on this idea: the iterative construction of the permutation starts by fixing an entry from the first column of the matrix and then proceeds by sampling a permutation according to the remaining submatrix.

To apply the ideas from the previous section, we fix the entries using upper bounds for the permanents of submatrices. With some abuse of notation, let $U(A)$ be an upper bound for the total weight of the set of permutations whose weights are given by A. Then, we choose the ith entry from the first column with a probability proportional to $a_{i1} \cdot U(A_{i1})$.

3.2. Nesting Bounds

The known nesting upper bounds are *product bounds*, meaning they can be written in the form

$$U(A) = \prod_{i=1} u(a_{i,1:n})$$

for a function u, where $a_{i,1:n}$ is the vector $(a_{i1}, a_{i2}, \ldots, a_{in})$. We call u the *row bound* of U.

Note that for a product bound, choosing i with probability proportional to $a_{i1} \cdot U(A_{i1})$ is equivalent to choosing it with probability proportional to $a_{i1}/u(a_{i,2:n})$. Therefore, the values are easy to precompute and can be queried in constant time. Algorithm 2 describes an estimator for the permanent that works with any given product bound. Providing a nesting product bound yields a NIS scheme. Notice that Algorithm 2 draws a single sample, so it needs no stopping rule—one is required for stopping the sampling only when the algorithm is called repeatedly.

The simplest nesting product bound consists of taking the sum of entries on each row, that is, its row bound is $\sum_{j=1}^{n} a_{ij}$. To see that this naive bound U_{N} is an upper bound, take any permutation σ and notice that $a(\sigma)$ is one of the terms of $U_{\mathrm{N}}(A)$ when it is expanded to a sum of products.

To see that U_{N} is nesting, take any of the terms in $U_{\mathrm{N}}(A)$. The sum $\sum_{i=1}^{n} a_{i1} \cdot U_{\mathrm{N}}(A_{i1})$ includes the term if and only if the product has exactly one entry from the first column as a factor. Additionally, this is true for at most one value of i, proving the property.

With the naive bound, the row is chosen with probability proportional to $a_{ij}/\sum_{k=j+1}^{n} a_{ik}$. For binary matrices, this reduces into an older SIS scheme of Chen & Liu (2007), which is a special case of a sampler of Smith & Dawkins (2001) for nonnegative matrices. Curiously, the row is sampled proportionally to $a_{ij}^2/\sum_{k=j+1}^{n} a_{ik}$ in the latter work, which does *not* correspond to a nesting upper bound since it contains the square of a_{ij}.

Algorithm 2 SIS for permanent

```
Input: A nonnegative matrix $A$, a row bound $u$.
Output: Estimate $a(\sigma)/q(\sigma)$ with $\sigma \sim q$.
$out \leftarrow 1$
$used \leftarrow$ [False, False, ..., False]
for $j \leftarrow 1, 2, \ldots, n$ do
    $p \leftarrow [0, 0, \ldots, 0]$
    for $i \leftarrow 1, 2, \ldots, n$ do
        if $\neg used[i]$ then
            $p[i] \leftarrow a_{ij}/u(a_{i,(j+1):n})$
        end if
    end for
    if $p[i] = 0$ for all $i$ then
        return 0
    end if
    Normalize $p$
    Draw $i \sim p$
    $out \leftarrow out \cdot a_{ij}/p[i]$
    $used[i] \leftarrow$ True
end for
return $out$
```

Empirically, the NIS scheme that uses the naive bound appears to converge fast based on our preliminary experiments, suggesting it has low critical ratio. However, the ratio $U_{\mathrm{N}}(A)/\operatorname{per}(A)$ can be large, hindering the suitability of the bound for obtaining estimates with accuracy guarantees by using Theorem 2.5.

The tightest known nesting bound that works for all nonnegative matrices is the *extended Huber bound* U_{E} of Harviainen & Koivisto (2023), which is based on an earlier upper bound for binary matrices (Huber, 2006). It is defined by the row bound

$$\sum_{k=1}^{n} \frac{h(k) - h(k-1)}{\mathrm{e}} \cdot a_{ik}^{*}$$

where a_{ik}^{*} is the kth largest entry on the ith row and h is given by the recurrence $h(0) := 0, h(1) := \mathrm{e}$, and

$$h(k) = h(k-1) + 1 + \frac{1}{2h(k-1)} + \frac{3}{5h(k-1)^2}\,.$$

Deep bounds (Harviainen et al., 2021) are generalizations of upper bounds for the permanent. Denote the submatrix of A that has its columns J and rows I by A_{IJ}. Then, the depth-d variant of the bound U with $J := [d]$ is obtained as

$$\begin{aligned} U^{d}(A) &:= \sum_{\substack{I \subseteq [n] \\ |I| = d}} \operatorname{per} A_{IJ} \cdot U(A_{\bar{I}\bar{J}}) \\ &\geq \sum_{\substack{I \subseteq [n] \\ |I| = d}} \operatorname{per} A_{IJ} \cdot \operatorname{per} A_{\bar{I}\bar{J}} \\ &= \operatorname{per} A\,. \end{aligned}$$

For nesting bounds, the upper bound decreases as a function of d. After preprocessing with dynamic programming of complexity $O(2^d dn)$, the nesting importance sampling proceeds as follows: First, sample I with probability proportional to $\operatorname{per} A_{IJ} \cdot U(A_{\bar{I}\bar{J}})$ in time $O(nd)$ with stochastic backtracking. Then, sample the remaining permutation according to the submatrix $A_{\bar{I}\bar{J}}$ using regular nesting importance sampling.

4. Relation to Previous Work

We will next cover some of the relevant literature on approximating the permanent and discuss their applicability if accuracy guarantees are desired.

4.1. Sinkhorn Balancing

A common preprocessing method applied to the matrices is Sinkhorn balancing (Sinkhorn, 1964) that is performed by alternating between scaling the rows and the columns by the sums of their entries. The permanent changes by the same scaling factors. If the matrix contains no positive entries that are not part of any positive-valued permutation, this progress quickly converges into a doubly-stochastic matrix where both the row and column sums equal one (Soules, 1991). Removing redundant entries can be performed in polynomial time (Tassa, 2012). This typically tightens the upper bound as well.

The use of Sinkhorn balancing is partially motivated by van der Waerden's conjecture proven by Egorychev (1981) and Falikman (1981) that states that the permanent of a doubly stochastic matrix is at least $n!/n^n \approx \mathrm{e}^{-n}$. Both the naive and extended Huber bound are at most 1 for doubly stochastic matrices, and so

$$U(A)/\operatorname{per} A \leq \mathrm{e}^n.$$

In contrast, there are nonnegative matrices where this ratio is super-exponential (Soules, 2003).

4.2. SIS with Pruning

A common issue with both importance and rejection samplers is that they can draw samples of weight 0. This is because they do not check that the remaining submatrix has a permutation of non-zero weight. Diaconis (2019) suggests doing exactly that for binary matrices, and Alimohammadi et al. (2023) generalize it for nonnegative matrices, building upon earlier ideas of Beichl & Sullivan (1999) and Smith & Dawkins (2001). They maintain a perfect matching that allows computing which entries have become redundant. Then, an entry from the column is chosen with probability proportional to its value. Ignoring the redundant entries reduces the variance of the samples but increases sampling time. However, based on the experiments of Section 5, it appears unlikely that this idea would be helpful to our NIS scheme: the ratio $U(A) \cdot \epsilon / \operatorname{per}(A)$ seems to dominate the critical ratio in the the bound of Theorem 2.5, and thus reducing the latter would not lead to noticable speedups.

4.3. Godsil–Gutman Type Estimators

Godsil–Gutman type estimators (Godsil & Gutman, 1981) utilize the similarities between the determinant and the permanent. Let $B = (b_{ij})$ be a random $n \times n$ matrix obtained from A such that $b_{ij} = c_{ij}\sqrt{a_{ij}}$, where each c_{ij} is chosen uniformly at random from the set $\{-1, 1\}$. Then, $\mathbb{E}\left[(\det B)^2\right] = \operatorname{per} A$. A similar result applies for the estimator $|\det B|^2$ when the coefficients are fourth roots of unity (Karmarkar et al., 1993) or quaternions (Chien et al., 2003). Here, $|\cdot|^2$ is the sum of the squares of the coefficients of the determinant in the quaternion basis.

The critical ratio of the estimator based on complex numbers is almost linear in n for almost all binary matrices and polynomial for all dense matrices (Frieze & Jerrum, 1995). Unfortunately, there are instances that result in an exponential critical ratio (Karmarkar et al., 1993; Chien et al., 2003) with no known way of efficiently detecting such hard instances. Additionally, there are matrices with entries from $\{-1, 1\}$ for any n where the determinant of the matrix is roughly $n^{n/2}$ (Clements & Lindström, 1965). Thus, we would need to divide each $|\det B|^2$ by n^n to apply the stopping rule of Theorem 2.4, with the expected value being $\operatorname{per} A/n^n$. This ratio, however, can be extremely small, rendering the stopping rule impractical.

4.4. Self-reducible Rejection Sampling

In rejection sampling, we draw samples from a proposal distribution and either *accept* or *reject* them such that the distribution of accepted samples is proportional to their weights. Then, the empirical acceptance rate can be used to obtain an estimate of the partition function of the distribution. Here, we draw permutations and estimate the permanent.

Huber (2006) suggested the following algorithm for sampling permutations. Let U be a nesting upper bound. First, fix an entry a_{i1} from the first column with probability $a_{i1} \cdot U(A_{i1})/U(A)$, and reject the sample with probability

$$1 - \sum_{i=1}^{n} a_{i1} \cdot U(A_{i1})/U(A)\,.$$

If we did not reject the sample, draw the rest of the permutation according to the values of the submatrix A_{i1}.

Let σ be a permutation and denote the submatrix of A with its rows $[n] \setminus \{\sigma(1), \sigma(2), \ldots, \sigma(j)\}$ and columns $[n] \setminus \{1, 2, \ldots, j\}$ by $A_{\sigma(1)\sigma(2)\ldots\sigma(j)}$. Then, the probability

Table 1. The average running time for computing an (ϵ, δ)-approximation with $\delta = 0.05$ and varying values of ϵ.

		$\epsilon = 0.02$			$\epsilon = 0.01$			$\epsilon = 0.005$		
Instance	n	DEEPNIS	DEEPAR	Ratio	DEEPNIS	DEEPAR	Ratio	DEEPNIS	DEEPAR	Ratio
ENZYMES-g479	28	$\mathbf{3 \cdot 10^1}$	$1 \cdot 10^2$	4	$\mathbf{8 \cdot 10^1}$	$4 \cdot 10^2$	5	$\mathbf{3 \cdot 10^2}$	$2 \cdot 10^3$	6
ENZYMES-g192	31	$\mathbf{3 \cdot 10^1}$	$2 \cdot 10^2$	7	$\mathbf{9 \cdot 10^1}$	$9 \cdot 10^2$	10	$\mathbf{3 \cdot 10^2}$	$4 \cdot 10^3$	15
ENZYMES-g230	32	$\mathbf{2 \cdot 10^1}$	$3 \cdot 10^2$	11	$\mathbf{6 \cdot 10^1}$	$1 \cdot 10^3$	16	$\mathbf{2 \cdot 10^2}$	$4 \cdot 10^3$	21
ENZYMES-g575	51	$\mathbf{1 \cdot 10^4}$	–	–	$\mathbf{2 \cdot 10^4}$	–	–	–	–	–
ENZYMES-g283	52	$\mathbf{1 \cdot 10^2}$	$2 \cdot 10^3$	17	$\mathbf{3 \cdot 10^2}$	$8 \cdot 10^3$	28	$\mathbf{8 \cdot 10^2}$	$3 \cdot 10^4$	44
ENZYMES-g501	66	$\mathbf{3 \cdot 10^1}$	$3 \cdot 10^2$	10	$\mathbf{8 \cdot 10^1}$	$1 \cdot 10^3$	15	$\mathbf{3 \cdot 10^2}$	$5 \cdot 10^3$	21
Staircase-30	30	$\mathbf{7 \cdot 10^0}$	$6 \cdot 10^1$	9	$\mathbf{1 \cdot 10^1}$	$3 \cdot 10^2$	21	$\mathbf{2 \cdot 10^1}$	$1 \cdot 10^3$	43
cage5	37	$\mathbf{6 \cdot 10^0}$	$2 \cdot 10^2$	36	$\mathbf{1 \cdot 10^1}$	$8 \cdot 10^2$	72	$\mathbf{3 \cdot 10^1}$	$3 \cdot 10^3$	115
bcspwr01	39	$\mathbf{3 \cdot 10^1}$	$2 \cdot 10^2$	5	$\mathbf{8 \cdot 10^1}$	$6 \cdot 10^2$	8	$\mathbf{2 \cdot 10^2}$	$3 \cdot 10^3$	10
Staircase-45	45	$\mathbf{7 \cdot 10^2}$	$2 \cdot 10^4$	25	$\mathbf{1 \cdot 10^3}$	–	–	$\mathbf{2 \cdot 10^3}$	–	–
GD95-c	62	–	–	–	–	–	–	–	–	–
CAG-mat72	72	$\mathbf{9 \cdot 10^3}$	–	–	$\mathbf{2 \cdot 10^4}$	–	–	$\mathbf{3 \cdot 10^4}$	–	–

of sampling σ is

$$\prod_{j=1}^{n} \frac{a_{\sigma(j),j} \cdot U\big(A_{\sigma(1)\sigma(2)\ldots\sigma(j-1)}\big)}{U\big(A_{\sigma(1)\sigma(2)\ldots\sigma(j)}\big)} = \frac{a(\sigma)}{U(A)}$$

with an expected acceptance ratio $\operatorname{per} A/U(A)$. Thus, multiplying the empirical acceptance rate by $U(A)$ yields an estimate of $\operatorname{per} A$. It remains to choose a stopping rule ensuring the desired accuracy.

The Gamma Bernoulli approximation scheme (GBAS) of Huber (2017) described in Algorithm 3 gives an estimator of $p := \operatorname{per} A/U(A)$ that stops after obtaining k accepted draws and outputs $\hat{p}$. Here, we interpret the samples to be Bernoulli-distributed with parameter $\operatorname{per} A/U(A)$, where 1 corresponds to an accepted sample. For the estimator, it holds that $p/\hat{p} \sim \operatorname{Gamma}(k, k-1)$. To achieve the desired accuracy, we find the smallest k for which

$$\Pr\left(\left|\frac{\hat{p}}{p} - 1\right| > \epsilon\right) = \Pr\left(\frac{p}{\hat{p}} < \frac{1}{1+\epsilon} \text{ or } \frac{p}{\hat{p}} > \frac{1}{1-\epsilon}\right)$$

is at most δ by evaluating the cumulative distribution functions of $\operatorname{Gamma}(k, k-1)$. Asymptotically, k is of size $O\left(\epsilon^{-2} \log \delta^{-1}\right)$, and each accepted sample requires approximately $U(A)/\operatorname{per} A$ draws. This suggests NIS to have potential for speedups against self-reducible rejection sampling of up to a factor of ϵ^{-1}.

5. Empirical Evaluation

We evaluated the performance of NIS on real-world matrices (Rossi & Ahmed, 2015) licensed under CC-BY-SA and synthetic instances used by Harviainen et al. (2021) and Harviainen & Koivisto (2023). The used instantiation of our scheme, dubbed DEEPNIS, utilizes the depth-16 variant of the extended Huber bound. We also remove redundant entries from the matrix (Tassa, 2012) and apply 100 iterations of Sinkhorn balancing like Alimohammadi et al. (2023). DEEPNIS is compared against the state-of-the-art rejection sampler DEEPAR that uses the same bound but applies n^2 rounds of Sinkhorn balancing only if it tightens the ratio $U(A)/\operatorname{per} A$. Additionally, they try tightening the ratio by randomized scaling of the columns. In all cases, we computed (ϵ, δ)-approximations of the permanents of the instances with $\delta = 0.05$. The implementations are publicly available on GitHub[1].

Algorithm 3 GBAS

```
Input: Matrix $A$, integer $k$.
Output: Estimator $\hat{p}$.
$r \leftarrow 0,\ s \leftarrow 0$
while $s \neq k$ do
    Draw $a \sim \operatorname{Exp}(1)$, $x \sim \operatorname{Bernoulli}(\operatorname{per} A/U(A))$
    $r \leftarrow r + a,\ s \leftarrow s + x$
end while
return $\hat{p} := (k-1)/r$
```

For real-world matrices, we ran both schemes with varying values of $\epsilon \in \{0.005, 0.01, 0.02\}$ on each instance independently 20 times. Each run was allowed to use 12 hours (43 200 seconds) of time. The results are reported in Table 1 containing the average times used for obtaining the estimates. Additionally, we report the ratio of the average time used by DEEPAR to DEEPNIS. If any of the runs exceeded the time limit, we report "–" as otherwise the averages would not be comparable. The running times t are rounded in the table to a number of the form $b \cdot 10^k$ that minimizes the relative error from t.

The synthetic instances are from three classes of matrices. *Uniform* consists of $n \times n$ matrices with $\operatorname{Uniform}[0, 1]$-

[1] https://github.com/Sums-of-Products/nesting-importance-sampling/

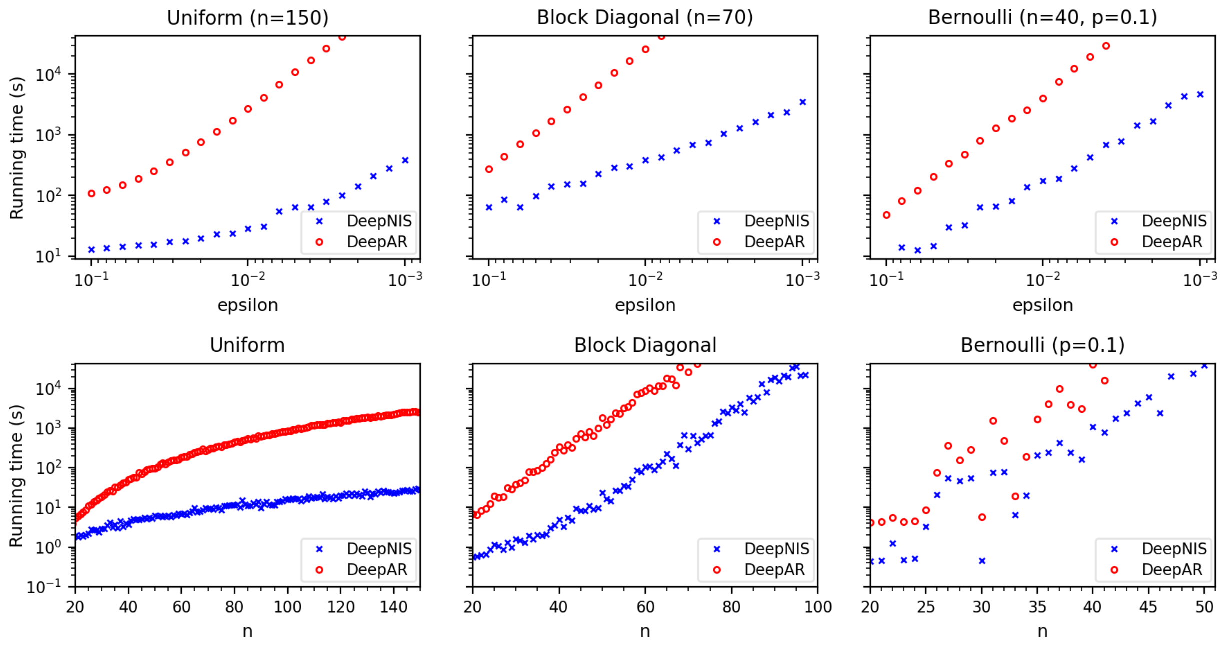

Figure 1. Comparison of running times over synthetic instances of varying values of $\epsilon \in [10^{-1}, 10^{-3}]$ on the first row and varying sizes with $\epsilon = 0.01$ on the second row.

distributed entries. *Block Diagonal* has blocks of 5×5 matrices with uniformly distributed entries on the main diagonal and zero elsewhere. *Bernoulli* has matrices where each entry is 1 with probability 0.1 and 0 otherwise. We evaluated the schemes with decreasing values of ϵ on a single instance from each class and with instances of increasing size n with $\epsilon = 0.01$. Only the results for instances with $n \geq 20$ are reported in Figure 1, since smaller instances can be solved exactly in less than a second (Ryser, 1963).

We see that DEEPNIS beats DEEPAR in all instances with the ratio of running times of DEEPAR to DEEPNIS being up to two orders of magnitude. Additionally, we see that the ratio increases as ϵ decreases, meaning that the time complexity of rejection sampling grows faster in ϵ^{-1} for the tested values of ϵ. This suggests that the critical ratio in the complexity bound of Theorem 2.5 is dominated by $U(\Omega) \cdot \epsilon / a(\Omega)$, although the critical ratio would start dominating eventually if we kept decreasing ϵ. In some instances, halving ϵ affected the ratio of running times by slightly over a factor of 2. This seems to be because of the variance in the random number of samples needed by the approximation scheme of Huber & Jones (2019). With DEEPAR, the running times over the 20 repetitions were fairly consistent, and the fastest and the slowest run on the same instance differed by a factor of 1.21. With DEEPNIS, they differed by a factor of 2.51.

As evidence for the correctness of the implementations, we note that the relative error from the sample average of all independent runs for real-world matrices exceeded ϵ in only $2.2\% < \delta$ of the runs. In other words, the estimates from DEEPNIS and DEEPAR agreed and were close to their mean. The structure of matrices in *Block Diagonal* enables exact computation of their permanents, and for these test instances the relative error from the exact value exceeded $\epsilon = 0.01$ in less than 1.2% of the runs.

6. Concluding Remarks

We presented a new method for approximate weighted counting or integration: nesting importance sampling. By sampling proportionally to a nesting upper bound, it yields bounded importance weights, thereby giving access to known efficient stopping rules to control the required number of samples. Nesting importance sampling can be viewed as a boosted variant of the sequential rejection sampling method of Huber (2006), reducing the number of samples but losing the ability to draw exact samples. The scheme is suitable for any setting where nesting upper bounds are available. Using the notoriously hard problem of approximating the permanent of a matrix as an example, we demonstrated the power of this approach by showing speedups of up to two orders of magnitude when compared against the state-of-the-art rejection sampling algorithm.

One curious nesting bound stemming from the previous work (Chen & Liu, 2007; Smith & Dawkins, 2001) is the naive bound: In our experiments (results not shown), the resulting estimator for the permanent appeared to converge rapidly, thus speaking for a low critical ratio of the estimator in practice. However, as the upper bound is large, it does not yield an efficient estimator with accuracy guarantees. It turns out that directly applying the nesting property with this bound leads to Lemma 2.2 greatly overestimating the importance weights. Developing a general method for tighter analysis of nesting but "naive" bounds could result in efficient approximation schemes for integration problems that are hard in the exact case, such as weighted counting of Bayesian network structures (Talvitie et al., 2019) or matchings of size k in undirected graphs (Curticapean, 2013).

Acknowledgements

Research partially supported by Research Council of Finland, Grant 351156.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Aaronson, S. and Arkhipov, A. The computational complexity of linear optics. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011*, pp. 333–342. ACM, 2011.

Alimohammadi, Y., Diaconis, P., Roghani, M., and Saberi, A. Sequential importance sampling for estimating expectations over the space of perfect matchings. *Ann. Appl. Probab.*, 33(2):999–1033, 2023.

Bapat, R. B. Permanents in probability and statistics. *Linear Algebra Appl.*, 127:3–25, 1990.

Bayati, M., Kim, J. H., and Saberi, A. A sequential algorithm for generating random graphs. *Algorithmica*, 58(4): 860–910, 2010.

Beichl, I. and Sullivan, F. Approximating the permanent via importance sampling with application to the dimer covering problem. *J. Comput. Phys.*, 149(1):128–147, 1999.

Bezáková, I., Štefankovič, D., Vazirani, V. V., and Vigoda, E. Accelerating simulated annealing for the permanent and combinatorial counting problems. *SIAM J. Comput.*, 37(5):1429–1454, 2008.

Bianco, G. L., Lorca, X., Truchet, C., and Pesant, G. Revisiting counting solutions for the global cardinality constraint. *J. Artif. Intell. Res.*, 66:411–441, 2019.

Chakraborty, S., Fremont, D. J., Meel, K. S., Seshia, S. A., and Vardi, M. Y. Distribution-aware sampling and weighted model counting for SAT. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2014*, pp. 1722–1730. AAAI Press, 2014.

Chakraborty, S., Shrotri, A. A., and Vardi, M. Y. On symbolic approaches for computing the matrix permanent. In *Proceedings of the Twenty-Fifth International Conference on Principles and Practice of Constraint Programming, CP 2019*, volume 11802 of *Lecture Notes in Computer Science*, pp. 71–90. Springer, 2019.

Chen, Y. and Liu, J. Sequential Monte Carlo methods for permutation tests on truncated data. *Stat. Sin*, 17(3):857–872, 2007.

Chien, S., Rasmussen, L. E., and Sinclair, A. Clifford algebras and approximating the permanent. *J. Comput. Syst. Sci.*, 67(2):263–290, 2003.

Clements, G. F. and Lindström, B. A sequence of (± 1)-determinants with large values. *Proc. Am. Math. Soc.*, 16 (3):548–550, 1965.

Curticapean, R. Counting matchings of size k is $\#$W[1]-hard. In *Proceedings of the 40th International Colloquium on Automata, Languages, and Programming, ICALP 2013*, volume 7965 of *Lecture Notes in Computer Science*, pp. 352–363. Springer, 2013.

Dagum, P., Karp, R. M., Luby, M., and Ross, S. M. An optimal algorithm for Monte Carlo estimation. *SIAM J. Comput.*, 29(5):1484–1496, 2000.

Diaconis, P. Sequential importance sampling for estimating the number of perfect matchings in bipartite graphs: An ongoing conversation with Laci. In *Building Bridges II: Mathematics of László Lovász*, pp. 223–233. Springer Berlin Heidelberg, 2019.

Diaconis, P. and Kolesnik, B. Randomized sequential importance sampling for estimating the number of perfect matchings in bipartite graphs. *Adv. Appl. Math.*, 131: 102247, 2021.

Egorychev, G. P. The solution of van der Waerden's problem for permanents. *Adv. Math.*, 42(3):299–305, 1981.

Ermon, S., Gomes, C. P., Sabharwal, A., and Selman, B. Taming the curse of dimensionality: Discrete integration by hashing and optimization. In *Proceedings of the Thirtieth International Conference on Machine Learning, ICML 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pp. 334–342. JMLR.org, 2013.

Falikman, D. I. Proof of the van der Waerden conjecture regarding the permanent of a doubly stochastic matrix. *Math. Notes*, 29(6):475–479, 1981.

Frieze, A. M. and Jerrum, M. An analysis of a Monte Carlo algorithm for estimating the permanent. *Comb.*, 15(1): 67–83, 1995.

Godsil, C. D. and Gutman, I. On the matching polynomial of a graph. *Algebraic methods in graph theory*, 25:241–249, 1981.

Harviainen, J. and Koivisto, M. A faster practical approximation scheme for the permanent. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023*, pp. 12216–12224. AAAI Press, 2023.

Harviainen, J., Röyskö, A., and Koivisto, M. Approximating the permanent with deep rejection sampling. In *Advances in Neural Information Processing Systems 34, NeurIPS 2021*, pp. 213–224, 2021.

Huber, M. Exact sampling from perfect matchings of dense regular bipartite graphs. *Algorithmica*, 44(3):183–193, 2006.

Huber, M. A Bernoulli mean estimate with known relative error distribution. *Random Struct. Algorithms*, 50(2): 173–182, 2017.

Huber, M. and Jones, B. Faster estimates of the mean of bounded random variables. *Math. Comput. Simul.*, 161: 93–101, 2019.

Huber, M. and Law, J. Fast approximation of the permanent for very dense problems. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2008*, pp. 681–689. SIAM, 2008.

Jensen, A. and Beichl, I. A sequential importance sampling algorithm for counting linear extensions. *ACM J. Exp. Algorithmics*, 25:1–14, 2020.

Jerrum, M., Sinclair, A., and Vigoda, E. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *J. ACM*, 51(4):671–697, 2004.

Karmarkar, N., Karp, R., Lipton, R., Lovász, L., and Luby, M. A Monte-Carlo algorithm for estimating the permanent. *SIAM J. Comput.*, 22(2):284–293, 1993.

Kuck, J., Dao, T., Rezatofighi, H., Sabharwal, A., and Ermon, S. Approximating the permanent by sampling from adaptive partitions. In *Advances in Neural Information Processing Systems 32, NeurIPS 2019*, pp. 8858–8869, 2019.

Lou, Q., Dechter, R., and Ihler, A. Dynamic importance sampling for anytime bounds of the partition function. In *Advances in Neural Information Processing Systems 30, NeurIPS 2017*, pp. 3196–3204, 2017.

Mnih, V., Szepesvári, C., and Audibert, J. Empirical Bernstein stopping. In *Proceedings of the Twenty-Fifth International Conference on Machine Learning, ICML 2008*, volume 307 of *ACM International Conference Proceeding Series*, pp. 672–679. ACM, 2008.

Newman, J. E. and Vardi, M. Y. FPRAS approximation of the matrix permanent in practice. *CoRR*, abs/2012.03367, 2020.

Robert, C. P. and Casella, G. *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer, 2004.

Rossi, R. A. and Ahmed, N. K. The network data repository with interactive graph analytics and visualization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI 2015*, pp. 4292–4293. AAAI Press, 2015. URL http://networkrepository.com.

Ryser, H. J. *Combinatorial mathematics*, volume 14. Mathematical Association of America, 1963.

Sinkhorn, R. A relationship between arbitrary positive matrices and doubly stochastic matrices. *Ann. Math. Stat.*, 35(2):876–879, 1964.

Skilling, J. Nested sampling. *AIP Conference Proceedings*, 735(1):395–405, 2004.

Smith, P. J. and Dawkins, B. Estimating the permanent by importance sampling from a finite population. *J. Stat. Comput. Simul.*, 70(3):197–214, 2001.

Soules, G. W. The rate of convergence of Sinkhorn balancing. *Linear Algebra Appl.*, 150:3–40, 1991.

Soules, G. W. New permanental upper bounds for nonnegative matrices. *Linear Multilinear Algebra*, 51(4):319–337, 2003.

Talvitie, T., Vuoksenmaa, A., and Koivisto, M. Exact sampling of directed acyclic graphs from modular distributions. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019*, volume 115 of *Proceedings of Machine Learning Research*, pp. 965–974. AUAI Press, 2019.

Tassa, T. Finding all maximally-matchable edges in a bipartite graph. *Theor. Comput. Sci.*, 423:50–58, 2012.

Uhlmann, J. K. Matrix permanent inequalities for approximating joint assignment matrices in tracking systems. *J. Frankl. Inst.*, 341(7):569–593, 2004.

Valiant, L. G. The complexity of computing the permanent. *Theor. Comput. Sci.*, 8(2):189–201, 1979.