Linear Algebra and its Applications 430 (2009) 1364–1374

Contents lists available at ScienceDirect

Linear Algebra and its Applications

journal homepage: www.elsevier.com/locate/laa

Efficiently computing the permanent and Hafnian of some banded Toeplitz matrices[☆]

Moshe Schwartz

Department of Electrical and Computer Engineering, Ben-Gurion University, Beer Sheva 84105, Israel

ARTICLE INFO

Article history:
Received 18 September 2008
Accepted 27 October 2008
Available online 11 December 2008

Submitted by R.A. Brualdi

AMS classification:
15A15
15A36

Keywords:
Toeplitz matrix
Banded Toeplitz matrix
Permanent
Hafnian

ABSTRACT

We present a new efficient method for computing the permanent and Hafnian of certain banded Toeplitz matrices. The method covers non-trivial cases for which previous known methods do not apply. The main idea is to use the elements of the first row and column, which determine the entire Toeplitz matrix, to construct a digraph in which certain paths correspond to permutations that the permanent and Hafnian count. Since counting paths can be done efficiently, the permanent and Hafnian for those matrices is easily obtainable.

1. Introduction

Efficiently computing the permanent and its counterpart, the Hafnian, of matrices is a notoriously difficult problem. Even if we restrict ourselves to $(0, 1)$ matrices, it was shown by Valiant in [22,21] that computing the permanent is a $\#P$-complete problem. The class $\#P$ consists of those problems which compute a function f, where f is the number of accepting paths of a non-deterministic polynomial-time (NP) Turing machine. Thus, efficiently computing the permanent of just $(0, 1)$ matrices would imply not only polynomial-time machines to NP problems, but also counting the number of their solutions.

[☆] This research was supported by a grant from the G.I.F., the German-Israeli Foundation for Scientific Research and Development (Grant no. 2179-1785.10/2007).

E-mail address: schwartz@ee.bgu.ac.il
URL: http://www.ee.bgu.ac.il/~schwartz

0024-3795/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.laa.2008.10.029

The most efficient algorithm to date for computing the permanent of a general $n \times n$ matrix is due to Ryser [18], with a complexity of $O(n2^n)$.

The permanent and Hafnian are connected to the well-known combinatorial problems of cycle covers and perfect matchings in graphs. They also have applications in numerous other problems (see [14,2] and references therein).

Since the problem of computing the permanent and Hafnian of general matrices is unlikely to be solved efficiently, attention has been given to more limited classes of matrices with a special structure. One such notable case is that of matrices whose permanent and Hafnian may be computed by the determinant and the Pfaffian, respectively, of a matrix with the same size. This was shown in the early work of Temperley and Fisher [20], and (independently and concurrently) in the work of Kasteleyn [7,8] – works which were all motivated by a problem from statistical mechanics. For the current state of these methods the reader is referred to [3,10] and references therein. These techniques were lately adopted in other areas, such as holographic reductions by Valiant [23], and constrained coding by Schwartz and Bruck [19].

Another family of matrices with special structure is that of circulants and Toeplitz matrices. Some explicit solutions or recurrence formulas for some cases of $(0,1)$ circulants may be found in [9,13,15,16]. Other $(0,1)$ circulants and very sparse Toeplitz matrices are further discussed in [4].

In this paper, we present novel algorithms for computing the permanent and Hafnian of banded Toeplitz matrices for cases which were not covered by previously-known methods. The narrower the band, the more efficient our algorithms are. For an $n \times n$ Toeplitz matrix A, and a $2n \times 2n$ Toeplitz matrix A', both of bandwidth m, the algorithms to compute $\operatorname{per}(A)$ and $\operatorname{Hf}(A')$ run in time $O\left(\binom{2m}{m}^3 \log n\right)$ and $O(2^{3m} \log n)$, respectively. At the extreme, for a constant bandwidth, the algorithms run in time logarithmic in the size of the matrix. Furthermore, in that case, the technique we present also allows us to compute both $\lim_{n\to\infty} \operatorname{per}(A)^{1/n}$ as well as $\lim_{n\to\infty} \operatorname{Hf}(A')^{1/n}$ efficiently. These are important in a variety of applications, see for example [6,1,19], as well as the running example we show which is a packing problem described in [12].

Loosely speaking, the main idea behind our algorithms is to construct a digraph which depends only on m and the specific non-zero diagonals in the matrices. Certain paths in that digraph correspond to permutations used by the permanent and Hafnian. Since counting paths can be done efficiently, we are able to compute the permanent and Hafnian using simple matrix multiplication of the weighted adjacency matrix of the constructed digraph.

The paper is organized as follows. In Section 2, we present the definitions and notation used throughout the paper. A simple motivating problem is also presented, which will be later used as a running example. We continue in Section 3 to describe in full detail the algorithm for computing the permanent of banded Toeplitz matrices, and analyze it. We conclude in Section 4 by showing the analogous algorithm for computing the related Hafnian.

2. Preliminaries and motivation

For two integers, $m, n \in \mathbb{Z}$, $m \leqslant n$, let $[m,n]$ denote the set $\{m, m+1, \ldots, n\}$, and let $[n]$ be defined as $[1,n]$. Denote by S_n the set of all permutations over $[n]$. Any permutation $\sigma \in S_n$ may be written as $\sigma = [\sigma(1), \ldots, \sigma(n)]$.

Let $A = (a_{i,j})$ be an $n \times n$ matrix over some ring R. The *permanent* of A is defined as

$$\operatorname{per}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i,\sigma(i)}.$$

The usual combinatorial interpretation of the permanent views A as the weighted adjacency matrix of a digraph G_A, i.e., the weight of edge $i \to j$ is $a_{i,j}$. In that case, $\operatorname{per}(A)$ is simply the weighted cycle cover[1] of G_A. Another common interpretation defines G'_A to be a bi-partite graph over a set of n left vertices,

[1] A cycle cover is a subset of the edges covering all the vertices by disjoint simple cycles. Each cycle cover is scored by the product of the weights of its edges. The weighted cycle cover is the sum of the scores over all cycle covers.

V_L, and a set of n right vertices, V_R, with an edge of weight $a_{i,j}$ connecting $v_i \in V_L$ with $v'_j \in V_R$. In that case, $\text{per}(A)$ is the weighted perfect matching[2] of the graph G'_A.

Consider an infinite sequence $\{t_k\}_{k=-\infty}^{\infty}$ over a ring R. An $n \times n$ *Toeplitz matrix* $A = (a_{i,j})$ induced by $\{t_k\}$ is a matrix for which $a_{i,j} = t_{j-i}$. We call the set $T = \{k \in \mathbb{Z} | t_k \neq 0\}$, the *support set* of $\{t_k\}$. A permutation $\sigma \in S_n$ is said to be *of type* T, for some $T \subseteq \mathbb{Z}$, if $\sigma(i) - i \in T$ for all $i \in [n]$. If there exists a constant $m \in \mathbb{N}_0$ such that $t_k = 0$ for all $|k| > m$, i.e., the support T of $\{t_k\}$ satisfies $T \subseteq [-m, m]$, then we say the matrix is *banded* and that its *bandwidth* is m. We can now write

$$\text{per}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i,\sigma(i)} = \sum_{\sigma \in S_n} \prod_{i=1}^{n} t_{\sigma(i)-i}.$$

If we define a *score function for permutations*, $\omega : S_n \to R$, as $\omega(\sigma) = \prod_{i=1}^{n} t_{\sigma(i)-i}$, then $\text{per}(A)$ is simply the sum of the scores of all permutations. We note that only permutations of type T contribute to this sum. Furthermore, since for any $i,j \in [n]$ we have $-n < j - i < n$, if we define $T' = T \cap [-n+1, n-1]$, then only permutation of type T' contribute to the above-mentioned sum. Thus

$$\text{per}(A) = \sum_{\substack{\sigma \in S_n \\ \sigma \text{ of type } T'}} \omega(\sigma). \tag{1}$$

Finally, we present one motivating problem for this work which will be used as a running example – the open problem of calculating the size of balls in S_n under the l_∞-norm (see [5]). Given two permutations $\sigma, \sigma' \in S_n$, their l_∞-distance is defined as

$$d_\infty(\sigma, \sigma') = \max_{i \in [n]} |\sigma(i) - \sigma'(i)|.$$

A *ball* of radius r centered about $\sigma \in S_n$ is defined as

$$\mathscr{B}_r(\sigma) = \{\sigma' \in S_n | d_\infty(\sigma, \sigma') \leqslant r\}.$$

It is well known (see [5]) that the size of a ball, $|\mathscr{B}_r(\sigma)|$, does not depend on the choice of the center, σ. Therefore, if $\varepsilon \in S_n$ denotes the identity permutation, we can examine $|\mathscr{B}_r(\varepsilon)|$. By our previous definitions it is now evident that $|\mathscr{B}_r(\varepsilon)| = \text{per}(A)$ where A is a $(0,1)$ Toeplitz matrix of size $n \times n$ induced by $\{t_k\}$

$$t_k = \begin{cases} 1, & -r \leqslant k \leqslant r, \\ 0, & \text{otherwise}, \end{cases}$$

which is of type $T = \{-r, \ldots, -1, 0, 1, \ldots, r\} = [-r, r]$. Expressions for $|\mathscr{B}_r(\varepsilon)|$ are known only for $r = 1$ and for $r = 2$ (see [11]). Knowing the ball size, as well as $\lim_{n\to\infty} \sqrt[n]{|\mathscr{B}_r(\varepsilon)|}$, is essential for bounding the efficiency of ball packing in S_n under the l_∞-norm (see [12] for an application).

3. Method description

Fix some $n \times n$ Toeplitz matrix A induced by $\{t_k\}$, and let $T = \{p_1, \ldots, p_\ell\}$, $p_1 < p_2 < \cdots < p_\ell$, be the support set of $\{t_k\}$. It is obvious that $\text{per}(A) = 0$ if $p_1 > 0$ or $p_\ell < 0$. Therefore, we will assume $p_1 \leqslant 0 \leqslant p_\ell$. Loosely speaking, the method we propose constructs a digraph in which certain paths correspond to permutations of type T. We will start by intuitively describing the construction of the digraph, and then follow with rigorous definitions and proofs.

We want to construct all the permutations $\sigma = [\sigma(1), \ldots, \sigma(n)] \in S_n$ of type T by setting $\sigma(1)$ in the first step, then $\sigma(2)$ in the second step, all the way to $\sigma(n)$ in the nth step. Let us closely examine the ith step: we would like to choose a valid value for $\sigma(i)$ such that σ is of type T. Hence, only values in the set $\{i + p_1, i + p_2, \ldots, i + p_\ell\}$ are possible. From these, however, we have to remove those values

[2] A perfect matching is a subset of the edges whose induced subgraph contains all the vertices, each with degree 1. Each perfect matching is scored by the product of the weights of its edges. The weighted perfect matching is the sum of the scores over all perfect matchings.

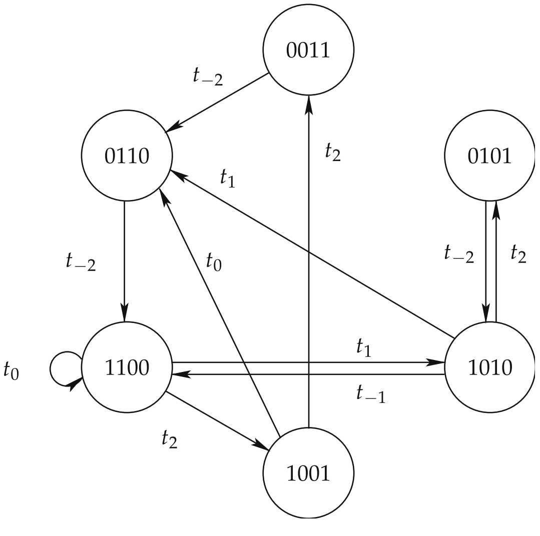

Fig. 1. $\mathscr{G}_T$ for $T = \{-2, -1, 0, 1, 2\}$.

which have been used in previous steps. If we assume that all the previous steps have chosen valid values, then it is guaranteed that $i + p_\ell$ has not been used in previous steps.

To that end, the algorithm we propose stores a "state" in the form of a binary string of length $p_\ell - p_1$ which is indexed by $[p_1, p_\ell - 1]$. The string is interpreted in the ith step as the availability map of the interval $i + [p_1, p_\ell - 1]$: a "0" in position j indicates that symbol $i + j$ has not been used so far, while a "1" indicates that it has. We now append a "0" at the end of string, since as we noted before, we are guaranteed that $i + p_\ell$ has not been chosen yet. After our choice of $\sigma(i)$ we update the string by changing the relevant "0" to a "1", and by left shifting it to represent the availability map for step $i + 1$.

We need a few more definitions before describing the construction. Let $b = b_1 b_2 \cdots b_m$ be a length-m binary string, $b_i \in \{0, 1\}$ for all $i \in [m]$. By b^q, $q \in \mathbb{N}_0$, we denote a concatenation of b to itself q times. Thus, for example, $1^3(01)^2 0$ denotes the string 11101010. The *weight* of b, denoted $w(b)$, is the number of non-zero entries in it, i.e., $w(b) = |\{i | b_i \neq 0\}|$. By $b0$ we denote the binary string of length $m + 1$ constructed by appending to b a "0" as the $(m + 1)$st bit. We also define the *left-shift operator* denoted L, operating on a length-m binary string as follows: for $b = b_1 b_2 \cdots b_m$ we define $L(b) = b_2 b_3 \cdots b_m$, i.e., the string of length $m - 1$ which is created from b by removing the leftmost bit. Finally, given two length-m strings, b and b', we denote by $b + b'$ the symbol-wise addition over the ring R.

We are now ready to describe the construction of the digraph.

Construction 1. Let $T = \{p_1, \ldots, p_\ell\} \subseteq \mathbb{Z}$, where $p_1 < p_2 < \cdots < p_\ell$, and where $p_1 \leqslant 0 \leqslant p_\ell$. We construct a digraph $\mathscr{G}_T = (V, E)$ in the following way: we first define the vertex set V as

$$V = \{b = b_{p_1} b_{p_1+1} \cdots b_{p_\ell - 1} \in \{0, 1\}^{p_\ell - p_1} | w(b) = -p_1\},$$

where for convenience, we will index the bits in each string by $[p_1, p_\ell - 1]$.

Let e_k, for $k \in [p_1, p_\ell]$, denote a length-$(p_\ell - p_1 + 1)$ binary string of all zeros, except for a "1" in the position with index k. We define the edge set E in the following way: for every $b \in V$ we construct all the edges of the form $b \to L(b0 + e_k)$, where $k \in T$ and $(b0)_k = 0$, i.e., we change exactly one "0" in $b0$ to a "1", and in a position allowed by T. Therefore, we may write the edge set as

$$E = \{b \to L(b0 + e_k) | b, L(b0 + e_k) \in V \wedge k \in T \wedge (b0)_k = 0\}.$$

We note that if $b_{p_1} = 0$, i.e., the leftmost bit is "0", for $L(b0 + e_k)$ to be in V we can only choose $k = p_1$. In that case $L(b0 + e_{p_1}) = L(b0)$, and b has a single outgoing edge.

Finally, given a sequence $\{t_k\}$ over a ring R, with support T, we assign weights to each of the edges. We define the edge-weight function, $W : E \to R$, as

$$W(b \to L(b0 + e_k)) = t_k.$$

Example 1. Motivated by the problem of computing the size of balls in S_n under the l_∞-norm, we start our running example by considering balls of radius $r = 2$. Fig. 1 shows $\mathcal{G}_T$ for $T = \{-2, -1, 0, 1, 2\} = [-r, r]$. The edges are marked with the appropriate weights from the sequence $\{t_k\}$.

The next theorem shows a correspondence between permutations of type T and certain paths in the graph $\mathcal{G}_T$.

Theorem 1. *Let $T = \{p_1, \ldots, p_\ell\} \subseteq \mathbb{Z}$, where $p_1 < p_2 < \cdots < p_\ell$, and where $p_1 \leqslant 0 \leqslant p_\ell$. Then there exists a bijection between the set of permutations from S_n of type T, and the paths of length n in $\mathcal{G}_T$ starting and ending in the vertex $1^{-p_1}0^{p_\ell}$.*

Proof. In the first direction we show that appropriate paths in $\mathcal{G}_T$ correspond to permutations of type T. Let us examine a path of length n in $\mathcal{G}_T$, starting and ending in $1^{-p_1}0^{p_\ell}$:

$$1^{-p_1}0^{p_\ell} = v_{i_0} \xrightarrow{k_1} v_{i_1} \xrightarrow{k_2} v_{i_2} \xrightarrow{k_3} \cdots \xrightarrow{k_n} v_{i_n} = 1^{-p_1}0^{p_\ell},$$

where the integer $k_r \in [p_1, p_\ell]$ above each edge is uniquely determined from the construction by the equation $v_{i_r} = L(v_{i_{r-1}}0 + e_{k_r})$, for all $r \in [n]$. We construct $\sigma = [\sigma(1), \ldots, \sigma(n)]$ by setting $\sigma(r) = r + k_r$ for every $r \in [n]$, and we contend that σ is a permutation in S_n of type T.

If σ were a permutation in S_n, then by our construction, not only is $k_r \in [p_1, p_\ell]$, but also $k_r \in T$. Thus, $\sigma(r) - r \in T$ for all $r \in [n]$ and the permutation is of type T. It follows that we only have to show that indeed σ is a permutation in S_n.

Using the observation that $\sigma(r) \in r + [p_1, p_\ell]$ and the fact that we start with the state $1^{-p_1}0^{p_\ell}$, it is easily seen that in the first $|p_1|$ steps we do not assign a non-positive value to any $\sigma(r)$. Thus, it is assured that $\sigma(r) \geqslant 1$ for all $r \in [n]$. In a similar fashion, since the path ends in state $1^{-p_1}0^{p_\ell}$, we can be certain that $\sigma(r) \leqslant n$ for all $r \in [n]$. Hence, it only remains to show that $\sigma(r) \neq \sigma(r')$ for every $r < r'$ and $r, r' \in [n]$. Let us assume to the contrary that $\sigma(r) = \sigma(r')$ for some $r < r'$ and $r, r' \in [n]$. Necessarily $(r + [p_1, p_\ell]) \cap (r' + [p_1, p_\ell]) \neq \emptyset$, which means that $r' - r \leqslant p_\ell - p_1$. Following our construction, at step r we chose k_r from T, but more importantly, we took $v_{i_{r-1}}0$ and changed the k_rth bit from a "0" to a "1". Thus, after $r' - r \leqslant p_\ell - p_1$ more steps, the $(k_r + r - r')$th bit of $v_{i_{r'-1}}$ remained set as "1", and so we could not have chosen $k_{r'} = k_r + r - r'$. It follows that $k_{r'} \neq k_r + r - r'$ and also that $\sigma(r') = r' + k_{r'} \neq r' + k_r + r - r' = \sigma(r)$, a contradiction. Therefore, σ is indeed a permutation in S_n. To finish this direction of the proof, it is obvious that different appropriate paths correspond to different permutations.

In the other direction we show that any permutation from S_n of type T corresponds to some appropriate path in $\mathcal{G}_T$. Let $\sigma \in S_n$ be a permutation of type T. We contend that the following path exists in $\mathcal{G}_T$:

$$1^{-p_1}0^{p_\ell} = v_{i_0} \to v_{i_1} \to v_{i_2} \to \cdots \to v_{i_n} = 1^{-p_1}0^{p_\ell},$$

where $v_{i_r} = L(v_{i_{r-1}}0 + e_{\sigma(r)-r})$ for all $r \in [n]$. Assume to the contrary such a path does not exist, and let r be the smallest integer in $[n]$ for which $v_{i_{r-1}} \to v_{i_r}$ does not exist as above. There are two possible reasons for $v_{i_{r-1}} \to v_{i_r}$ to not exist, and we will rule them both out to reach a contradiction.

The first reason we check is that $\sigma(r) - r \neq p_1$ and at the same time the leftmost bit of $v_{i_{r-1}}$ is a "0". We remember that in such a case, the only edge leaving $v_{i_{r-1}}$ is the edge $v_{i_{r-1}} \to L(v_{i_{r-1}}0 + e_{p_1})$. We first note that this cannot happen in the first $|p_1|$ steps of the path since we start with the vertex $v_{i_0} = 1^{-p_1}0^{p_\ell}$. We may therefore assume that $r > |p_1|$. Furthermore, this means that the symbol $r + p_1$ has not been assigned to any of the positions $\sigma(1), \ldots, \sigma(r-1)$. Since $\sigma(r) - r \neq p_1$, it follows that $r + p_1$ is assigned to some $\sigma(r')$, $r' > r$. But then we get $\sigma(r') - r' = r + p_1 - r' < p_1$, a contradiction to the fact that σ is of type T.

The second reason we check is that $(v_{i_{r-1}}0)_{\sigma(r)-r} = 1$. We first note that this cannot happen due to the $|p_1|$ "1"s in the initial state $v_{i_0} = 1^{-p_1}0^{p_\ell}$ since these correspond to the symbols of $[p_1 + 1, 0]$ which we do not use in any permutation in S_n. Thus, the only way $(v_{i_{r-1}}0)_{\sigma(r)-r} = 1$ may happen, is that in some previous step $r' < r$, where $r' \in [n]$, that "1" was set, and in the following steps leading to step r, shifted to its current position. Hence, $\sigma(r') - r' - (r - r') = \sigma(r) - r$, which reduces to $\sigma(r') = \sigma(r)$, a contradiction.

Lastly, having made sure the path exists, we still need to show that the last vertex reached, v_{i_n}, is indeed $1^{-p_1}0^{p_\ell}$. This is easily proved by noting that any "1" in the rightmost p_ℓ bits of v_n represents a symbol from $[n+1, n+p_\ell]$ having been assigned in one of the last p_ℓ steps. To complete the proof, it is also evident that different permutations result in different paths. □

Let us now define the *score function for paths*, $\omega : E^* \to R$, as

$$\omega(v_{i_0} \to v_{i_1} \to \cdots \to v_{i_n}) = \prod_{r=1}^{n} W(v_{i_{r-1}} \to v_{i_r}).$$

Lemma 2. *Let $\{t_k\}$ be a sequence of support $T \subseteq \mathbb{Z}$ over a ring R. For every permutation $\sigma \in S_n$ of type T and its corresponding path $v_{i_0} \to v_{i_1} \to \cdots \to v_{i_n}$ in $\mathscr{G}_T$ (defined as in the proof of Theorem 1) we have $\omega(\sigma) = \omega(v_{i_0} \to v_{i_1} \to \cdots \to v_{i_n})$, i.e., the score of the permutation equals the score of its corresponding path.*

Proof. By the correspondence defined in the proof of Theorem 1 it is obvious that for every $r \in [n]$

$$W(v_{i_{r-1}} \to v_{i_r}) = W(v_{i_{r-1}} \to L(v_{i_{r-1}}0 + e_{\sigma(r)-r})) = t_{\sigma(r)-r},$$

which immediately implies the lemma. □

Before introducing the main theorem, we define the *weighted adjacency matrix of $\mathscr{G}_T$*, denoted $\mathscr{A}(\mathscr{G}_T) = (w_{i,j})$, as the $|V| \times |V|$ matrix

$$w_{i,j} = \begin{cases} W(v_i \to v_j) & \text{if the edge } v_i \to v_j \text{ exists,} \\ 0 & \text{otherwise.} \end{cases}$$

It is well known that the sum of the scores of all the paths of length n in $\mathscr{G}_T$ from v_i to v_j is given by $(\mathscr{A}(\mathscr{G}_T)^n)_{i,j}$, i.e.,

$$\sum_{p \in P_{i \to j}(n)} \omega(p) = (\mathscr{A}(\mathscr{G}_T)^n)_{i,j}, \tag{2}$$

where $P_{i \to j}(n)$ denotes the set of all paths of length n from v_i to v_j. This leads us to the main theorem.

Theorem 3. *Let $\{t_k\}$ be a sequence over a ring R, let $T \subseteq \mathbb{Z}$ be its support, and let A be its induced $n \times n$ Toeplitz matrix. Let us denote $T' = T \cap [-n+1, n-1] = \{p_1, \ldots, p_\ell\}$, with $p_1 < \cdots < p_\ell$, and $-n < p_1 \leqslant 0 \leqslant p_\ell < n$. Furthermore, let us index the vertices of $\mathscr{G}_{T'}$ such that $v_1 = 1^{-p_1}0^{p_\ell}$. Then*

$$\mathrm{per}(A) = (\mathscr{A}(\mathscr{G}_{T'})^n)_{1,1}.$$

Proof. Let $P_{1 \to 1}(n)$ denote the set of all paths of length n in $\mathscr{G}_{T'}$ starting and ending in $v_1 = 1^{-p_1}0^{p_\ell}$. We now have

$$\mathrm{per}(A) = \sum_{\substack{\sigma \in S_n \\ \sigma \text{ of type } T'}} \omega(\sigma) = \sum_{p \in P_{1 \to 1}(n)} \omega(p) = (\mathscr{A}(\mathscr{G}_{T'})^n)_{1,1},$$

where the first equality is (1), the second is by Lemma 2, and the last is by (2). □

We can now analyze the resulting complexity of the suggested algorithm. We are only interested in the number of additions and multiplications over the ring R, since we see those as atomic operations. A more detailed analysis is required if more basic operations are assumed, for example, if we were to look at the representation of ring elements and the cost of manipulating it to create the operations of addition and multiplication.

Theorem 4. *Let A be an $n \times n$ banded Toeplitz matrix with bandwidth m. The complexity of computing $\mathrm{per}(A)$ using Construction 1 and Theorem 3 is $O\left(\binom{2m}{m}^3 \log n\right)$.*

Proof. The input to the algorithm may be given as a support set $T' \subseteq [-m, m]$, and the list of ring elements corresponding to this interval, $\{t_k\}_{k=-m}^{m}$, as well as an unary representation of n, i.e., the string 1^n. The digraph $\mathscr{G}_{T'}$ has $O\left(\binom{2m}{m}\right)$ vertices so building its weighted adjacency matrix takes $O\left(\binom{2m}{m}^2\right)$ time. We then need to raise the matrix to its nth power. Using a trivial multiplication algorithm of complexity $O\left(\binom{2m}{m}^3\right)$ and needing $O(\log n)$ such multiplications, we reach the result. □

Corollary 5. *For $m = O(1)$ the banded Toeplitz permanent algorithm we presented runs in time $O(\log n)$, and for $m = O(\log \log n)$ it runs in time $O((\log n)^4)$ – both polynomial in the length of a binary representation of n. For $m = O(\log n)$ the algorithm runs in time $O(n^3 \log n)$, which is polynomial in the value of n.*

An added benefit of this algorithm is the fact that it allows us, in some cases, to compute $\lim_{n\to\infty}(\text{per}(A))^{1/n}$ easily. This is shown in the next example.

Example 2. Continuing our running example, we remember the size of balls in S_n under the l_∞-norm is given by $\text{per}(A)$, where A is the $n \times n$ banded Toeplitz matrix induced by the binary sequence $\{t_k\}$ with support $T = [-r, r]$

$$t_k = \begin{cases} 1, & -r \leqslant k \leqslant r, \\ 0, & \text{otherwise.} \end{cases}$$

For this example, let us fix $r = 2$. The digraph $\mathscr{G}_T$ defined by Construction 1 was shown in Fig. 1. We note that $\mathscr{G}_T$ does not depend on the choice of n. Furthermore, its weighted adjacency matrix, $\mathscr{A}(\mathscr{G}_T)$ has only non-negative entries, and is primitive.[3] By Perron–Frobenius theory (see, for example, Chapter 1 in [17]), the spectral radius of $\mathscr{A}(\mathscr{G}_T)$, denoted $\lambda(\mathscr{A}(\mathscr{G}_T))$, is achieved by a real positive eigenvalue of $\mathscr{A}(\mathscr{G}_T)$. There also exist a left (row) eigenvector x, and a right (column) eigenvector y, both associated with $\lambda(\mathscr{A}(\mathscr{G}_T))$, and both with strictly positive entries, such that $x \cdot y = 1$. For these two vectors

$$\lim_{n\to\infty} \frac{1}{(\lambda(\mathscr{A}(\mathscr{G}_T)))^n} \mathscr{A}(\mathscr{G}_T)^n = yx.$$

It follows that

$$\lim_{n\to\infty} \sqrt[n]{\text{per}(A)} = \lim_{n\to\infty} \sqrt[n]{(\mathscr{A}(\mathscr{G}_T)^n)_{1,1}} = \lambda(\mathscr{A}(\mathscr{G}_T)).$$

For this example, with $r = 2$, the resulting $\lambda(\mathscr{A}(\mathscr{G}_T))$ is the largest positive root of the polynomial

$$\det(\lambda I - \mathscr{A}(\mathscr{G}_T)) = \lambda^6 - \lambda^5 - 2\lambda^4 - \lambda^3 - \lambda^2 + \lambda + 1.$$

4. Computing the Hafnian

In this section, we show an efficient method for computing the Hafnian. The method is a simple adaptation of the one shown in the previous section for the permanent. Since the proofs in this section are very similar to those of the previous section, we will only sketch them briefly.

A *canonical permutation* $\sigma \in S_{2n}$ is one for which

$$\sigma(1) < \sigma(2) \quad \sigma(3) < \sigma(4) \quad \cdots \quad \sigma(2n-1) < \sigma(2n),$$

as well as

$$\sigma(1) < \sigma(3) < \sigma(5) < \cdots < \sigma(2n-1).$$

The set of all canonical permutations is denoted by C_{2n}.

[3] A non-negative square matrix B is said to be primitive if there exists an integer $m \in \mathbb{N}$ such that all the entries in B^m are strictly positive.

Let $A=(a_{i,j})$ be a $2n\times 2n$ matrix over some ring R. The *Hafnian* of A is defined as

$$\mathrm{Hf}(A)=\sum_{\sigma\in C_{2n}}\prod_{i=1}^{n}a_{\sigma(2i-1),\sigma(2i)}.$$

We note that the Hafnian depends only on the elements of A positioned strictly above the main diagonal. Just like the permanent, the Hafnian also has a simple combinatorial interpretation. Assume the matrix A is symmetric,[4] and let G_A be the graph whose weighted adjacency matrix is A. It can be easily seen that $\mathrm{Hf}(A)$ is the weighted perfect matching of G_A. Since the permanent is also related to the weighted perfect matching of graphs, we have the following well known (see, for example, Chapter 8 in [14]) connection between the two:

$$\mathrm{per}(A)=\mathrm{Hf}\begin{pmatrix}0 & A\\ A & 0\end{pmatrix}.$$

This does not, however, make the previous section on permanents unnecessary, since the matrix on the right is not necessarily Toeplitz, and its bandwidth might be significantly higher.

Let A be a $2n\times 2n$ Toeplitz matrix induced by the infinite sequence $\{t_k\}$. Since the Hafnian uses only elements positioned strictly above the main diagonal, we are interested in sequences whose support, T, is strictly positive. We say a canonical permutation $\sigma\in C_{2n}$ is *of H-type T*, if $\sigma(2i)-\sigma(2i-1)\in T$ for all $i\in[n]$. Let us further define the *H-score function for canonical permutations*, $\bar{\omega}:C_{2n}\to R$, as $\bar{\omega}(\sigma)=\prod_{i=1}^{n}t_{\sigma(2i)-\sigma(2i-1)}$. It is clearly seen that

$$\mathrm{Hf}(A)=\sum_{\sigma\in C_{2n}}\prod_{i=1}^{n}a_{\sigma(2i-1),\sigma(2i)}=\sum_{\sigma\in C_{2n}}\prod_{i=1}^{n}t_{\sigma(2i)-\sigma(2i-1)}=\sum_{\substack{\sigma\in C_{2n}\\ \sigma\text{of H-type }T'}}\bar{\omega}(\sigma),$$

where $T'=T\cap[2n-1]$, since only canonical permutations of H-type T' contribute to the sum.

Fix some $2n\times 2n$ Toeplitz matrix A induced by $\{t_k\}$, and let $T=\{p_1,\dots,p_\ell\}$, $1\leqslant p_1<p_2<\cdots<p_\ell$, be the support of $\{t_k\}$. Let us define the operator $\bar{L}$, operating on a length-m binary string, as follows: for $b=b_1b_2\cdots b_m$ we define $\bar{L}(b)=b_2\cdots b_m0$, i.e., $\bar{L}(b)=L(b)0$. We will now show the analog of Construction 1.

Construction 2. Let $T=\{p_1,\dots,p_\ell\}\subseteq\mathbb{Z}$, where $1\leqslant p_1<p_2<\cdots<p_\ell$. We construct a digraph $\mathscr{H}_T=(V,E)$ in the following way: we first define the vertex set V as

$$V=\{b=b_1b_2\cdots b_{p_\ell}\in\{0,1\}^{p_\ell}|b_{p_\ell}=0\},$$

where for convenience, we will index the bits of each vertex by $[p_\ell]$.

Let e_k, $k\in[p_\ell]$, denote a length-p_ℓ binary string of all zeros, except for a "1" in the position with index k. We define the edge set E in the following way: for every $b\in V$ we construct all the edges of the form $b\to\bar{L}^{j+1}(b+e_k)$, where $k\in T$, $b_k=0$, and j is the largest integer such that $b+e_k=1^j\{0,1\}^{p_\ell-j}$. In other words, we change exactly one "0" in b to a "1" in a position allowed by T. We then remove the longest prefix of "1"s in the state string, and append the same number of "0"s to the right of the state string. Finally, we remove the leftmost bit (which must be a "0"), and append a "0" to the right of the string. We may write the edge set as

$$E=\Big\{b\to\bar{L}^{j+1}(b+e_k)\ |b\in V\wedge k\in T\wedge b_k=0\wedge$$
$$j\text{ is the largest integer such that }b+e_k=1^j\{0,1\}^{p_\ell-j}\Big\}.$$

We note that if $b+e_k=1^{p_\ell}$, then state b has a single outgoing edge $1^{p_\ell}\to 0^{p_\ell}$.

Finally, given a sequence $\{t_k\}$ of support T over a ring R, we assign weights to each of the edges. We define the edge-weight function, $W:E\to R$, as

[4] Since the Hafnian only depends on the elements strictly above the main diagonal, given those elements we can always complete the matrix to be symmetric.

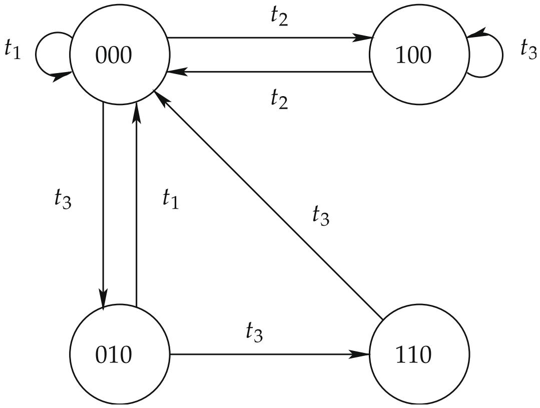

Fig. 2. $\mathscr{H}_T$ for $T = \{1, 2, 3\}$.

$$W\left(b \to \overline{L}^{j+1}(b + e_k)\right) = t_k,$$

where j and k are determined as defined above.

Example 3. Fig. 2 shows $\mathscr{H}_T$ for $T = \{1, 2, 3\}$. The edges are marked with the appropriate weights from the sequence $\{t_k\}$.

Theorem 6. *Let $T = \{p_1, \ldots, p_\ell\} \subseteq \mathbb{Z}$, where $1 \leqslant p_1 < p_2 < \cdots < p_\ell$. Then there exists a bijection between the set of canonical permutations from C_{2n} of type T, and the paths of length n in $\mathscr{H}_T$ starting and ending in the vertex 0^{p_ℓ}.*

Proof (*Sketch*). The proof is very similar to the proof of Theorem 1. The basic bijection is as follows: consider a path of the form

$$0^{p_\ell} = v_{i_0} \overset{j_1,k_1}{\to} v_{i_1} \overset{j_2,k_2}{\to} v_{i_2} \overset{j_3,k_3}{\to} \cdots \overset{j_n,k_n}{\to} v_{i_n} = 0^{p_\ell},$$

where the integers $k_r \in [p_\ell]$ and $j_r \in \mathbb{Z}$ above each edge are uniquely determined from the construction by the equation $v_{i_r} = \overline{L}^{j_r+1}(v_{i_{r-1}} + e_{k_r})$, for all $r \in [n]$. To this path we map the canonical permutation $\sigma = [\sigma(1), \ldots, \sigma(2n)]$ where

$$\sigma(2r-1) = r + \sum_{i=1}^{r-1} j_i \quad \text{and} \quad \sigma(2r) = \sigma(2r-1) + k_r$$

for all $r \in [n]$. Showing that this is indeed a bijection as required, is done in a similar fashion to the proof of Theorem 1. □

Lemma 7. *Let $\{t_k\}$ be a sequence of support $T \subseteq \mathbb{Z}$ over a ring R. For every canonical permutation $\sigma \in C_{2n}$ of H-type T and its corresponding path $v_{i_0} \to v_{i_1} \to \cdots \to v_{i_n}$ in $\mathscr{H}_T$ (defined as in the proof of Theorem 6) we have $\bar{\omega}(\sigma) = \omega(v_{i_0} \to v_{i_1} \to \cdots \to v_{i_n})$, i.e., the H-score of the permutation equals the score of its corresponding path.*

Proof. By the correspondence defined in the proof of Theorem 6 it is obvious that for every $r \in [n]$

$$W(v_{i_{r-1}} \to v_{i_r}) = W\left(v_{i_{r-1}} \to \overline{L}^{j_r+1}(v_{i_{r-1}} + e_{\sigma(2r)-\sigma(2r-1)})\right) = t_{\sigma(2r)-\sigma(2r-1)},$$

which immediately implies the lemma. □

The following theorem is the analog of Theorem 3 for the Hafnian.

Theorem 8. *Let $\{t_k\}$ be a sequence over a ring R, let $T \subseteq \mathbb{Z}$ be its support, and let A be its induced $2n \times 2n$ Toeplitz matrix. Denote $T' = T \cap [2n-1] = \{p_1, \ldots, p_\ell\}$, with $1 \leqslant p_1 < \cdots < p_\ell < 2n$. Furthermore, let us index the vertices of $\mathscr{H}_{T'}$ such that $v_1 = 0^{p_\ell}$. Then*

$$\mathrm{Hf}(A) = (\mathscr{A}(\mathscr{H}_{T'})^n)_{1,1}.$$

Proof. Let $P_{1\to 1}(n)$ denote the set of all paths of length n in $\mathscr{H}_{T'}$ starting and ending in $v_1 = 0^{p_\ell}$. We now have

$$\mathrm{Hf}(A) = \sum_{\substack{\sigma\in C_n \\ \sigma \text{ of H-type } T'}} \bar{\omega}(\sigma) = \sum_{p\in P_{1\to 1}(n)} \omega(p) = (\mathscr{A}(\mathscr{G}_{T'})^n)_{1,1}. \qquad \square$$

Theorem 9. *Let A be a $2n \times 2n$ banded Toeplitz matrix with bandwidth m. The complexity of computing $\mathrm{Hf}(A)$ using Construction 2 and Theorem 8 is $O(2^{3m} \log n)$.*

Proof. The proof is essentially the same as that of Theorem 4. We feed the algorithm a support set $T' \subseteq [m]$, the ring elements $\{t_k\}_{k=1}^m$, and an unary representation of n. This time, the digraph $\mathscr{H}_{T'}$ has $O(2^m)$ vertices and so its weighted adjacency matrix has $O(2^{2m})$ elements. Raising this matrix to its nth power may be done trivially in $O(2^{3m} \log n)$ time. $\square$

Corollary 10. *For $m = O(1)$ the banded Toeplitz Hafnian algorithm we presented runs in time $O(\log n)$, and for $m = O(\log\log n)$ it runs in time $O((\log n)^4)$ – both polynomial in the length of a binary representation of n. For $m = O(\log n)$ the algorithm runs in time $O(n^3 \log n)$, which is polynomial in the value of n.*

Again, the algorithm allows us, in some cases, to compute $\lim_{n\to\infty}(\mathrm{Hf}(A))^{1/n}$ efficiently, as seen in the following example.

Example 4. We extend Example 3, in which $T = \{1, 2, 3\}$, by further defining the binary sequence $\{t_k\}$ as

$$t_k = \begin{cases} 1, & k \in T, \\ 0, & \text{otherwise.} \end{cases}$$

The weighted adjacency matrix $\mathscr{A}(\mathscr{H}_T)$ of the digraph $\mathscr{H}_T$ (shown in Fig. 2) turns out to be primitive. If A is the $2n \times 2n$ banded Toeplitz matrix induced by $\{t_k\}$, then

$$\lim_{n\to\infty} \sqrt[n]{\mathrm{Hf}(A)} = \lim_{n\to\infty} \sqrt[n]{(\mathscr{A}(\mathscr{H}_T)^n)_{1,1}} = \lambda(\mathscr{A}(\mathscr{H}_T)),$$

where $\lambda(\mathscr{A}(\mathscr{H}_T))$ is the largest positive root of the polynomial

$$\det(\lambda I - \mathscr{A}(\mathscr{H}_T)) = \lambda^4 - 2\lambda^3 - \lambda^2 + 1.$$

References

[1] A. Böttcher, S.M. Grudsky, Toeplitz Matrices, Asymptotic Linear Algebra, and Functional Analysis, Birkhäuser, 2000.

[2] R.A. Brualdi, H.J. Ryser, Combinatorial Matrix Theory, Cambridge University Press, 1991.

[3] R.A. Brualdi, B.L. Shader, Matrices of Sign-Solvable Linear Systems, Cambridge University Press, 1995.

[4] B. Codenotti, V. Crespi, G. Resta, On the permanent of certain (0, 1) Toeplitz matrices, Linear Algebra Appl. 267 (1997) 65–100.

[5] M. Deza, H. Huang, Metrics on permutations, a survey, J. Combin. Inform. System Sci. 23 (1998) 173–185.

[6] R.M. Gray, On the asymptotic eigenvalue distribution of Toeplitz matrices, IEEE Trans. Inform. Theory 18 (1972) 725–730.

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

[8] P.W. Kasteleyn, Graph theory and crystal physics, in: F. Harary (Ed.), Graph Theory and Theoretical Physics, Academic Press, 1967, pp. 43–110.

[9] B.W. King, F.D. Parker, A Fibonacci matrix and the permanent function, Fibonacci Quart. 7 (1969) 539–544.
[10] G. Kuperberg, An exploration of the permanent-determinant method, Electron. J. Combin. 5 (1998).
[11] R. Lagrange, Quelques résultats dans la métrique des permutations, Ann. Sci. Ecole Norm. Sup. 79 (1962).
[12] T.-T. Lin, S.-C. Tsai, W.-G. Tzeng, Efficient encoding and decoding with permutation arrays, in: Proceedings of the 2008 IEEE International Symposium on Information Theory (ISIT2008), Toronto, Canada, 2008, pp. 211–214.
[13] N. Metropolis, M.L. Stein, P.R. Stein, Permanents of cyclic $(0, 1)$ matrices, J. Combin. Theory Ser. B 7 (1969) 291–321.
[14] H. Minc, Permanents, in: Encyclopedia of Mathematics and its Applications, vol. 6, Cambridge University Press, 1978.
[15] H. Minc, Recurrence formulas for permanents of $(0, 1)$ circulants, Linear Algebra Appl. 71 (1985) 241–265.
[16] H. Minc, Permanental compounds and permanents of $(0, 1)$ circulants, Linear Algebra Appl. 86 (1987) 11–42.
[17] H. Minc, Nonnegative Matrices, Wiley, New York, 1988.
[18] H.J. Ryser, Combinatorial mathematics, Carus Mathematical Monograph, vol. 14, Mathematical Association of America, 1963.
[19] M. Schwartz, J. Bruck, Constrained codes as networks of relations, IEEE Trans. Inform. Theory 54 (5) (2008) 2179–2195.
[20] H.N.V. Temperley, M.E. Fisher, Dimer problem in statistical mechanics – an exact result, Philos. Mag. 6 (1960) 1061–1063.
[21] L.G. Valiant, Completeness classes in algebra, in: ACM Symposium on the Theory of Comput., 1979, pp. 249–261.
[22] L.G. Valiant, The complexity of computing the permanent, Theoret. Comput. Sci. 8 (1979) 189–201.
[23] L.G. Valiant, Holographic algorithms, in: Proceedings of the 45th Annual IEEE Symposium on the Foundations of Computer Science (FOCS2004), 2004, pp. 306–315.