Constrained Codes as Networks of Relations

Moshe Schwartz
Electrical and Computer Engineering
Ben-Gurion University
Beer Sheva 84105, Israel
schwartz@ee.bgu.ac.il

Jehoshua Bruck
California Institute of Technology
1200 E California Blvd., Mail Code 136-93
Pasadena, CA 91125, U.S.A.
bruck@paradise.caltech.edu

***Abstract*— We revisit the well-known problem of determining the capacity of constrained systems. While the one-dimensional case is well understood, the capacity of two-dimensional systems is mostly unknown. When it is non-zero, except for the $(1,\infty)$-RLL system on the hexagonal lattice, there are no closed-form analytical solutions known. Furthermore, for the related problem of counting the exact number of constrained arrays of any given size, only exponential-time algorithms are known.**

We present a novel approach to finding the exact capacity of two-dimensional constrained systems, as well as efficiently counting the exact number of constrained arrays of any given size. To that end, we borrow graph-theoretic tools originally developed for the field of statistical mechanics, tools for efficiently simulating quantum circuits, as well as tools from the theory of the spectral distribution of Toeplitz matrices.

I. Introduction

While most storage devices record information on a two-dimensional surface, they emulate a one-dimensional environment by spacing tracks of recorded data. The distance between adjacent tracks in common devices is an order of magnitude larger than the distance between adjacent symbols along the track. The next big leap in storage density may be achieved by reducing the distance between tracks. This in turn, requires a two-dimensional constrained-coding scheme to be employed.

A two-dimensional constrained system $\mathcal{S}_{n,m}$ is simply a set of $n \times m$ arrays over some specified alphabet. The common example of such a system is the (d,k)-RLL constraint in which each row and each column of the array has runs of zeroes whose length is at least d and at most k. Other two-dimensional constraints forbid certain patterns in the arrays, such as the no-isolated-bit constraint in which every bit agrees with at least one of its four neighbors in the two-dimensional array.

An important measure associated with a constrained system is its *capacity*. Introduced by Shannon [14], the capacity of a constrained system $\mathcal{S}$ is defined as

$$\mathsf{cap}(S) \stackrel{\text{def}}{=} \lim_{n,m\to\infty} \frac{\log_2 |S_{n,m}|}{nm}.$$

While the one-dimensional case is well understood, there is little known about the capacity of two-dimensional systems.

In the case of two-dimensional (d,k)-RLL systems, Ito et al. [8] characterized the values of (d,k) for which the capacity is zero. General bounds on the capacity of (d,k)-RLL were given by Kato and Zeger [11], constructive lower bounds for (d,∞)-RLL by Halevy et al. [7], and non-constructive asymptotically-tight bounds for $(0,k)$-RLL by Schwartz and Vardy [13]. For the specific case of $(1,\infty)$-RLL, Calkin and Wilf [3] gave a numerical estimation method using transfer matrices. Only for the $(1,\infty)$-RLL constraint on the hexagonal lattice, Baxter [1] gave an exact *but not rigorous*[1] analytical solution using the corner transfer matrix method.

Other two-dimensional constraints do not fare any better. Halevy et al. [7] considered bit stuffing encoders for the two-dimensional no-isolated-bit constraint to constructively estimate its capacity. Non-constructively, Forchhammer and Laursen [6], estimated this capacity using random fields.

The method we present is general enough to encompass a wide variety of constraints (both local and global) though its expressive power is yet undetermined. We use only *mathematically-rigorous* tools to obtain exact capacity solutions and polynomial-time algorithms. The method is based on a series of reductions: **1)** A constrained system is first reduced to a network of relations in a way which enables us to connect the number of satisfying assignments to the network with the number of constrained arrays. **2)** This network of relations is transformed to a weighted graph using holographic reductions in such a way that the number of satisfying assignments to the network equals the weighted perfect matching of the graph. This is a many-to-many reduction in which the individual perfect matchings do not correspond in any one-to-one way to the original satisfying assignments. **3)** Finally, the weighted perfect matching of the graph is expressed as a Pfaffian of a certain skew-symmetric matrix by using the FKT (Fisher-Kasteleyn-Temperley) method. The capacity of the original constrained system is the limit of the Pfaffian, while the Pfaffian itself provides a polynomial-time algorithm for counting the number of constrained arrays.

The reductions and algorithms are given very briefly due to the severe page limit, while the polynomial-time algorithm is not described at all. For the complete results and proofs the reader is encouraged to read [12]. In Section II we introduce holographic reductions and the FKT method. We apply these tools in Section III to an example constrained system. We conclude in Section IV with a description of further results.

II. Background

A. Networks of Relations

We start by introducing networks of relations. For a discussion of the subject see [4] and references therein. Given some

This work was supported in part by the Caltech Lee Center for Advanced Networking.

[1] As Baxter notes in [2] page 409: "It is not mathematically rigorous, in that certain analyticity properties ... are assumed, and the results ... (which depend on assuming that various large-lattice limits can be interchanged) are used. However, I believe that these assumptions ... are in fact correct."

ground set Ω, a *relation on n variables* is a subset $R \subset \Omega^n$. A *network of relations* is a graph $G = (V, E)$ where we associate with each vertex $v \in V$ a relation R_v on $\deg(v)$ variables being the ordered set of incident edges on v. We can now assign every edge a value from Ω and check whether all the relations are satisfied. We say that an assignment is a *satisfying assignment* if for every $v \in V$, the relation R_v is satisfied.

Example 1. *Let us take as an example the network of relations shown in Figure* 1*. We use the ground set* $\Omega = \{0, 1\}$ *and define* $R_=$ *to be the* all-equal relation *on three variables,* $R_{\neq}$ *to be the* not-all-equal relation *on three variables, and* ϕ_+ *to be the* accept-all *relation on one variable:*

x_1	x_2	x_3	$R_=$	$R_{\neq}$
0	0	0	1	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

x_1	ϕ_+
0	1
1	1

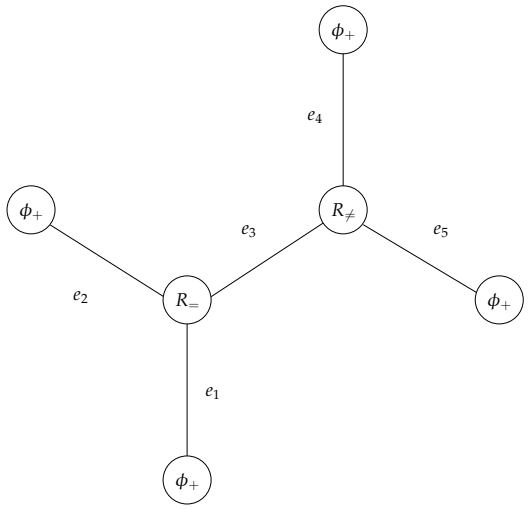

Figure 1. The network of relations of Example 1

We can easily see that there are exactly 6 *distinct satisfying assignments to this network which we list below:*

$$(e_1, \ldots, e_5) \in \{(0,0,0,0,1), (0,0,0,1,0), (0,0,0,1,1), (1,1,1,0,0), (1,1,1,0,1), (1,1,1,1,0)\}.$$

If we wanted to be completely accurate, we should have included a numbering of the incident edges to each vertex of Figure 1*. However, since all the relations in this example are symmetric, this is unnecessary.* □

B. Holographic Reductions

Holographic reductions were introduced by Valiant in [18] to show certain counting problems may be solved in polynomial time, and in [17] to simulate quantum circuits efficiently. Though the notion of networks of relations does not appear as such in his work, Valiant shows a many-to-many reduction from such networks to weighted graphs. This reduction preserves the total number of solutions, i.e., the number of satisfying assignments to the original network of relations equals the weighted perfect matching of the resulting graph. The reduction itself is realized by replacing each of the vertices of the original network with a small gadget.

Let $G = (V, E)$ be a graph. A *perfect matching* is a subset of edges $M \subseteq E$ such that every vertex $v \in V$ is incident to exactly one of the edges in M. The set of all perfect matchings will be denoted $\mathrm{PM}(G)$. We can now assign complex weights to the edges $w : E \to \mathbb{C}$, and define the *weighted perfect matching* of G to be

$$\mathrm{PerfMatch}(G) \stackrel{\mathrm{def}}{=} \sum_{M \in \mathrm{PM}(G)} \prod_{e \in M} w(e).$$

Our aim is to replace vertices in the network of relations, with gadgets which somehow capture the original relations. The gadgets are called *matchgates* and the resulting graph is called a *matchgrid*. At this point, just like in [18], we require the graph G to be planar as well as all the matchgates we use, resulting in a planar matchgrid graph. This is perhaps the most restrictive requirement we face during the process. We are, however, able to use non-planar graphs, though at a cost of increased computational complexity. We omit the details of the holographic reduction and the intuition behind it.

Example 2. *We complete the matchgrid for the network of relations of Example* 1*. The resulting matchgrid is shown in Figure* 2 *(where the arrows on the edges are to be disregarded at the moment). Each gray circle represents the place of the original vertex in the network of relations which is now occupied with a matchgate subgraph gadget. The skeptical reader is encouraged to verify that the weighted perfect matching of this graph is indeed* 6*, as is the total number of satisfying assignments.* □

C. The FKT Method

The FKT method gives a simple expression for the weighted perfect matching of certain graphs which is also computable efficiently. It was developed independently and concurrently by Fisher and Temperley [15], [5], and by Kasteleyn [9].

Let G be a graph with weights on the edges, and let $A = (a_{i,j})$ be its $n \times n$ adjacency matrix where $a_{i,j}$ is the weight of the edge between vertices i and j. Since we are interested in graphs with perfect matchings we assume n is even. An *orientation* of the an undirected graph G is simply an assignment of a direction to each of the edges of the graph. The solution given by Kasteleyn requires a special orientation called a *Pfaffian orientation* (for more details see [9]). Given a weighted graph G, and a Pfaffian orientation of its edges, we build a modified skew-symmetric adjacency matrix $A = (a_{i,j})$ as follows:

$$a_{i,j} = \begin{cases} 0 & \text{no edge between } i \text{ and } j \\ w(e_{i,j}) & \text{if } i \to j \\ -w(e_{i,j}) & \text{if } j \to i \end{cases}$$

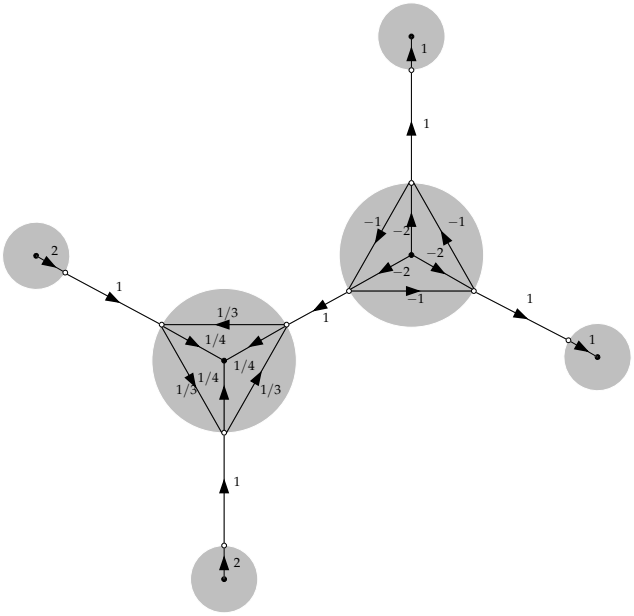

Figure 2. The complete matchgrid for the network of relations from Example 1 with a Pfaffian orientation of the edges

where $i \to j$ denotes the edge between vertices i and j is oriented from i to j. Note that A is not the adjacency matrix of the graph G in the usual sense. Using this construction Kasteleyn [9] showed that

$$\mathrm{PerfMatch}(G) = \pm \mathrm{Pf}(A) = \pm\sqrt{\det(A)}$$

where, $\mathrm{Pf}(A)$ is the Pfaffian of A and $\det(A)$ is the determinant of A. Since in most cases we know the sign of the outcome, the $\pm$ may be easily fixed.

It now remains a matter of finding out which graphs allow a Pfaffian orientation. Such graphs are called *Pfaffian orientable*. In his later work, Kasteleyn [10] showed that all planar graphs are Pfaffian orientable, which is the reason we required matchgrids to be planar in the previous section. For planar graphs, it was shown in [10], that if we orient the edges such that every clockwise walk on a face of the graph has an odd number of edges agreeing, then that orientation is a Pfaffian orientation. As a result, a simple polynomial time algorithm which finds such an orientation is also shown. A Pfaffian orientation for Example 1 is shown in Figure 2.

III. The capacity

Most constrained systems are easily defined by a finite set of forbidden patterns. For our example we choose a constraint we call the *Path-Cover Constraint (PC Constraint)*, and which we motivate by first examining its one-dimensional version. If we are given a graph $G = (V, E)$, a path-cover for the graph is a set of simple paths (open or closed) of positive length, which are vertex disjoint, and which cover all the vertices. An alternative way of stating this constraint is that given a graph, we assign either a **0** or a **1** to each of the edges, such that when removing the edges with a **0**, all the vertices remain with degree either 1 or 2.

In the one-dimensional case the graph G is simply the one-dimensional lattice with vertices $V = \{v_0, v_1, \ldots, v_n\}$ and edges $E = \{(v_i, v_{i+1}) \mid 0 \leqslant i \leqslant n-1\}$. It is easily seen that a valid PC assignment of values to edges is any assignment which does not contain two adjacent **0**'s. Thus, the one-dimensional PC constraint is the famous one-dimensional $(0,1)$-RLL constraint (and with bit-flipping, the $(1,\infty)$-RLL constraint). The capacity in this case is known to be $\log_2[(1+\sqrt{5})/2] = 0.69424\ldots$.

Turning to two-dimensions, we choose a the two-dimensional triangular grid as the graph G: we tile the plane with regular triangles, place a vertex at the center of each triangle, and draw an edge between vertices whose triangles share a face. Again, we assign either a **0** or a **1** to the edges of the graph such that after removing the edges assigned a **0**, all the remaining vertices are of degree either 1 or 2.

There is a multitude of possible reductions of the constrained $n \times n$ array on the plane, to a network of relations. We show a simple reduction for which the constrained arrays are in an "almost" one-to-one correspondence with the satisfying assignments to the network.

We think of the triangular grid as drawn on a plane. We replace each vertex of the grid with the relation $R_{\neq}$ on three variables. This relation makes sure that the three adjacent cells do not contain the same bit, i.e., the forbidden pattern of PC. It is easy to be convinced that, if we ignore the perimeter of the array, every constrained array induces exactly one satisfying assignment and vice versa. The resulting network of relations is shown in Figure 3.

We do however have to take care of the perimeter of the array as well. To do so, we connect dangling edges to extra vertices of the accept-all relation ϕ_+. Each such vertex has the potential of multiplying the number of satisfying assignments by a factor of 2. But since we have only $O(n)$ such vertices, this does not change the capacity as calculated by counting the total number of satisfying assignments. The extra accept-all vertices are also shown in Figure 3.

It is easily seen that the network of relations we built is bipartite by noting that upright triangles are connected only to inverted triangles, and vice versa. Conveniently for us, for the main bulk of the network we have just one type of relation, but we do have to specify some as recognizers and some as generators. We arbitrarily choose to build a generator $R_{\neq}$ in inverted triangles, and a recognizer $R_{\neq}$ in upright triangles. For the perimeter of the network we need to implement ϕ_+ both as a generator and as a recognizer.

Finding a Pfaffian orientation for the graph is an easy task. The orientation is not necessarily unique. The extremely regular nature of the graph suggests the existence of a simple orientation. If we closely examine the network of relations in Figure 3, we see that, apart from the perimeter of the array, it is made up of a single *basic block* and its translations. The simplest basic block is just a recognizer $R_{\neq}$ vertex and a generator $R_{\neq}$ vertex. This basic block may be oriented

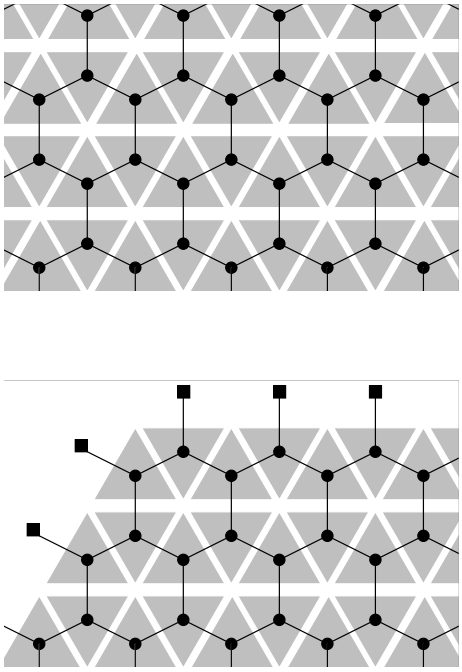

Figure 3. The top image shows part of a network of relations for the PC constraint. Each filled circle represents the $R_{\neq}$ relation. The gray triangles show the original cells of the triangular grid. The bottom image shows the top left corner of the array with the filled squares representing the ϕ_+ relation.

as shown in Figure 4. It is also easy to verify Kasteleyn's orientation rule for planar graphs: every clockwise walk on an inner face has an odd number of edges agreeing. This may be verified both for the inner faces of the block, and the inner faces created by the joining of a few blocks.

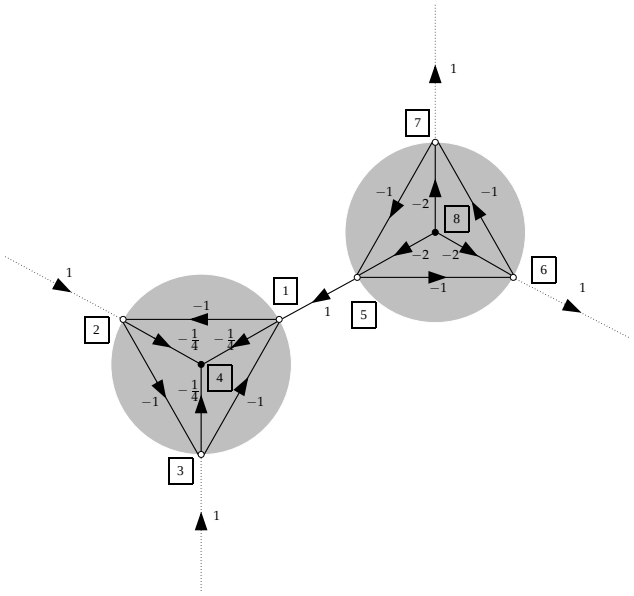

Figure 4. A Pfaffian orientation of the basic block. The dotted edges denote the edges between translations of the basic block. The numbers in squares index the vertices.

Finally, we also have to orient the edges which correspond to the ϕ_+ matchgates which lie on the perimeter of the array. Those matchgates do not contain any inner face themselves, and do not form an inner face with the rest of the graph. Thus, they may be oriented arbitrarily.

In light of the previous sections, the capacity of the constrained system S we are now examining is given by

$$\mathsf{cap}(S) = \lim_{n\to\infty} \frac{\log_2 |\mathrm{Pf}(A)|}{3n^2} = \lim_{n\to\infty} \frac{\log_2 \sqrt{\det(A)}}{3n^2}$$

where A is the skew-symmetric adjacency matrix of the matchgrid corresponding to the $n \times n$ constrained array. The 3 in the denominator comes from the fact that a basic block contains three bit storage positions (three edges from the original network to be assigned a value).

A derivation of an expression for the exact capacity largely depends on the ease of manipulating the matrix A. We first simplify it by noting that the matchgates for ϕ_+ contain just one edge which must be taken in any perfect matching, which also forces the edge connecting the matchgate to its single neighboring matchgate to be dropped. Since the weight of the edge is a constant, and since we have only $O(n)$ such matchgates along the perimeter, we may ignore them altogether without changing the resulting capacity calculation. So from now on, by abuse of notation, let A denote the skew-symmetric adjacency matrix with the ϕ_+ matchgates and their connecting edges removed.

The components for a compact representation of A are the skew-symmetric matrix for the basic block (where the vertices are indexed as in Figure 4),

$$B = \begin{pmatrix} 0 & -1 & 1 & -\frac{1}{4} & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & -\frac{1}{4} & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & -\frac{1}{4} & 0 & 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 2 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 & -2 & -2 & -2 & 0 \end{pmatrix},$$

and the matrix $\Delta_{i,j}$ (of the same dimensions as B) which is all zeroes except for position (i,j) which is 1. Furthermore, we need I_n the $n \times n$ identity matrix, and the $n \times n$ matrix U which is all zeroes except for positions $(i, i+1)$ which are 1.

We have an $n \times n$ array of basic blocks which we order in the natural way. This part of the graph is represented by the skew-symmetric adjacency matrix $I_n \otimes I_n \otimes B$. We still have to represent the edges between basic blocks in the same row, $I_n \otimes U \otimes \Delta_{6,2} - I_n \otimes U^T \otimes \Delta_{6,2}^T$, and the edges between basic blocks in different rows, $U \otimes I_n \otimes \Delta_{7,3} - U^T \otimes I_n \otimes \Delta_{7,3}^T$. Thus, we get an expression for the skew-symmetric adjacency matrix A,

$$A = I_n \otimes I_n \otimes B + I_n \otimes U \otimes \Delta_{6,2} - I_n \otimes U^T \otimes \Delta_{6,2}^T + U \otimes I_n \otimes \Delta_{7,3} - U^T \otimes I_n \otimes \Delta_{7,3}^T. \quad (1)$$

For the last step we rely on the theory of spectral distribution of Toeplitz matrices (see Tilli [16]). For natural numbers

$p, k \geqslant 1$, let an integrable p-variate function $f : [-\pi, \pi]^p \to \mathbb{C}^{k\times k}$ and a multi-index $n = (n_1, \ldots, n_p)$, $n_i \geqslant 1$ be given. The p-level Toeplitz matrix $T_n(f)$ is defined as

$$T_n(f) \stackrel{\text{def}}{=} \sum_{j_1=-n_1+1}^{n_1-1} \cdots \sum_{j_p=-n_p+1}^{n_p-1} J_{n_1}^{(j_1)} \otimes \cdots \otimes J_{n_p}^{(j_p)} \otimes a_{j_1,\ldots,j_p}(f)$$

where $J_m^{(l)}$ denotes the matrix of order m whose i, j entry equals 1 if $j - i = l$ and equals zero otherwise, and where

$$a_{j_1,\ldots,j_p}(f) \stackrel{\text{def}}{=} \frac{1}{(2\pi)^p} \int_{[-\pi,\pi]^p} f(x) e^{-\mathbf{i}(j_1 x_1 + \cdots + j_p x_p)} dx$$

is a matrix in $\mathbb{C}^{k\times k}$ and $\mathbf{i} = \sqrt{-1}$.

Theorem 3. *If $f : [-\pi, \pi]^p \to \mathbb{C}^{k\times k}$ is an integrable Hermitian matrix-valued function, then for any function F, uniformly continuous and bounded over $\mathbb{R}$ it holds*

$$\lim_{n\to\infty} \frac{1}{n_1 \ldots n_p} \sum_{j=1}^{kn_1\ldots n_p} F[\lambda_j(T_n(f))] = \\ = \frac{1}{(2\pi)^p} \int_{[-\pi,\pi]^p} \sum_{j=1}^{k} F[\lambda_j(f(x))] dx$$

where $\lambda_j(M)$ denotes the j-th eigenvalue of M.

To apply this theorem to our needs we notice the following: First, for an $n \times n$ matrix M we have $\log_2 |\det(M)| = \sum_{i=1}^{n} \log_2 |\lambda_i(M)|$. Second, we notice that A from (1) is a 2-level Toeplitz matrix, but it is skew-symmetric and not Hermitian as required. This is easily fixed by noting that $\mathbf{i}A$ is Hermitian, and since the order of A is a multiple of 4, then $\det(A) = \det(\mathbf{i}A)$. Thus, $A = T_n(f)$ where we define

$$f(\theta_1, \theta_2) = \mathbf{i}[B + e^{\mathbf{i}\theta_1}\Delta_{6,2} - e^{-\mathbf{i}\theta_1}\Delta_{6,2}^T \\ + e^{\mathbf{i}\theta_2}\Delta_{7,3} - e^{-\mathbf{i}\theta_2}\Delta_{7,3}^T].$$

We can prove that the eigenvalues of $\mathbf{i}A$ are bounded away from 0 (proof omitted). Then we can set F from Theorem 3 to behave like $\log_2(|x|)$ on the closed interval containing the eigenvalues and still be bounded and uniformly continuous. Now we are done, and the capacity of the constraint is exactly calculated as

$$\begin{aligned} \mathsf{cap}(S) &= \lim_{n\to\infty} \frac{\log_2 \sqrt{\det(A)}}{3n^2} \\ &= \lim_{n\to\infty} \frac{1}{6n^2} \log_2 |\det(T_n(f))| \\ &= \frac{1}{24\pi^2} \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} \log_2 |\det(f(\phi_1, \phi_2))| \, d\phi_1 d\phi_2 \\ &= \frac{1}{24\pi^2} \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} \log_2 |21 - 4\cos\phi_1 - 4\cos\phi_2 \\ &\qquad - 4\cos(\phi_1 - \phi_2)| d\phi_1 d\phi_2 \\ &= 0.72399217\ldots \end{aligned}$$

IV. Conclusion

We presented a general method for calculating the exact capacity of two-dimensional constrained systems. Through a series of reductions, such systems are transformed to networks of relations, then to a limit of a Pfaffian, and by the theory of Toeplitz determinants, to a double integral. This method may be also adapted to provide, to our knowledge, the first polynomial time algorithm for counting the number constrained arrays of a given dimension. For more results the reader is referred to [12].

Many open questions remain regarding the parameters of the method which were not introduced in detail. The main question is that of the expressive power of this method and the systems which may be amenable to this kind of treatment.

References

[1] R. J. Baxter, "Hard hexagons: exact solution," *J. Phys. A: Math. Gen.*, vol. 13, pp. L61–L70, 1980.

[2] ——, *Exactly Solved Models in Statistical Mechanics*. Academic Press, 1982.

[3] N. Calkin and H. Wilf, "The number of independent sets in the grid graph," *SIAM J. Discrete Math.*, vol. 11, pp. 54–60, 1998.

[4] M. Cook, "Networks of Relations," Ph.D. dissertation, California Institute of Technology, Pasadena CA, U.S.A., 2005. [Online]. Available: http://paradise.caltech.edu/papers/thesis011.pdf

[5] M. E. Fisher, "Statistical mechanics of dimers on plane lattice," *Phys. Rev.*, vol. 124, no. 6, pp. 1664–1672, Dec. 1961.

[6] S. Forchhammer and T. V. Laursen, "A model for the two-dimensional no isolated bit constraint," in *Proceedings of the 2006 IEEE International Symposium on Information Theory, ISIT2006, Seattle, WA, U.S.A.*, July 2006, pp. 1189–1193.

[7] S. Halevy, J. Chen, R. M. Roth, P. H. Siegel, and J. K. Wolf, "Improved bit-stuffing bounds on two-dimensional constraints," *IEEE Trans. on Inform. Theory*, vol. 50, no. 5, pp. 824–838, May 2004.

[8] H. Ito, A. Kato, Z. Nagy, and K. Zeger, "Zero capacity region of multidimensional run length constraints," *Elec. J. of Comb.*, vol. 6, 1999.

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

[10] ——, "Graph Theory and Crystal Physics," in *Graph Theory and Theoretical Physics*, F. Harary, Ed. Academic Press, 1967, pp. 43–110.

[11] A. Kato and K. Zeger, "On the capacity of two-dimensional run-length constrained channels," *IEEE Trans. on Inform. Theory*, vol. 45, pp. 1527–1540, July 1999.

[12] M. Schwartz and J. Bruck, "Constrained codes as networks of relations," in preparation.

[13] M. Schwartz and A. Vardy, "Tight asymptotic bounds on the capacity of multi-dimensional $(0, k)$-RLL," in *Proceedings of the 16th AAECC, Las Vegas, NV, U.S.A., Lecture Notes in Computer Science*, vol. 3857, Feb. 2006, pp. 225–234.

[14] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, July 1948.

[15] H. N. V. Temperley and M. E. Fisher, "Dimer problem in statistical mechanics — an exact result," *Phil. Mag.*, vol. 6, pp. 1061–1063, 1960.

[16] P. Tilli, "A note on the spectral distribution of Toeplitz matrices," *Linear and Multilinear Algebra*, vol. 45, pp. 147–159, 1998.

[17] L. G. Valiant, "Quantum circuits that can be simulated classically in polynomial time," *SIAM J. Comput.*, vol. 31, no. 4, pp. 1229–1254, 2002.

[18] ——, "Holographic algorithms," in *Proceedings of the 45th Annual IEEE Symposium on the Foundations of Computer Science (FOCS2004)*, Oct. 2004, pp. 306–315.