# The Channel Capacity of One and Two-Dimensional Constrained Codes

Xuerong Yong

Department of Computer Science

Hong Kong UST

# OUTLINE

1. Introduction

2. Capacity and Its Properties

3. Examples of Previous Work

4. Our Results

5. Possible Extensions

# 1. INTRODUCTION

- Capacity is an upper bound on the rate at which data can be transmitted over a channel or stored in a data medium.
- Shannon (1948) states that reliable transmision is only possible when we transmit information at a rate below its capacity.
- Already much work on 1-D capacity.
- 2-D problem in its infancy and currently very active (Only techniques for bounding).
- Our Work: (1) *theoretical results* for 2-D problems; (2) *techniques* for better bounding capacities. (The *ideas and techniques* are not confined to specific 2-D codes.)

## 1.1. DATA STORAGE DEVICES —- Applications

Different data storage devices have different constraints on the ways that data can be coded.

Examples of such devices:

- Conventional diskette and hard diskette drivers;
- Optical read-only drivers such as CD, CD-ROM drivers;
- Magnetic tape drivers, digital audio tape systems;
- Digital compact cassette audio tape systems.

⋮

## 1.2. CODES FOR RECORDING

A *(d,k)-Runlength Limited (RLL)* code is a set of codewords over $\{0,1\}$ all satisfying:

*$d$ ($k$) is the minimum (maximum) number of $0$'s separating consecutive $1$'s.*

*Example:* ((1,7)-RLL constrints)

$$...1000100000101000000001...; \quad (YES)$$

A 2-D RLL code is a set of binary arrays, each satisfying both horizontal and vertical (possibly different) RLL constraints.

*Example:* The 2-D $(1,\infty)$-*RLL* code

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} \mathbf{1} & 0 & \mathbf{1} & \mathbf{1} & 0 \\ \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

The first matrix is a good codeword, while the second one isn't.

## 1.3. SOME RECORDING STANDARDS

- Many commercial systems use the code with constraint $(d, k) = (1, 7)$.
- Magnetic disc drivers use the $(1, 3)$-RLL constraint;
- Compact audio discs use the $(2, 10)$-RLL constraint;
- 2-D codes are used in page oriented storage system, e.g. holographic recording, BUT mostly studied though for theoretical interests.

## 1.4. CONSTRAINED MATRICES

A *constrained matrix* is a binary array that does not contain specified forbidden submatrices. 2-D codes are sets of constrained matrices.

An example:
A 2-D $(1, \infty)$-*RLL* codeword is a constrained matrix that does not contain:

$$(1, 1) \quad \text{or} \quad \left(\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right)$$

Studied constraints include

- *Read/Write isolated* memory
- *general RLL restricted* matrices
- *Checkerboard-Code* matrices
- other pure combinatorial constraints, e.g., *kings placement* on chessboard

The 2-D *RLL* Constrained code $S^{(2)}_{1,\infty}$ also arises in statistical physics and graph theory. Previously studied separately on those areas.

- Burton and Steif (1994) called *hard-square*, or *hard-core lattice gas* system
- Engel (1982) called it *Fibonacci number of a lattice*,
- Calkin and Wilf (1998) called it *independent sets in grid graph.* Denoted the constraint by $S_{hs}$.

# 2.1. Definition of Capacity (1-D)

*Combinatorial description*: The capacity $cap(S)$ measures the growth rate of the number of sequences of size $n$, $N(n;S)$ in $S$.

$$cap(S) = \lim_{n\to\infty} \frac{\log_2 N(n;S)}{n}.$$

**NOTE**: Equivalent *Algebraic* and *Probabilistic* definitions exist, but we don't need them here.

Example: Consider the $(0,1)$-RLL code. It is easy to see that we have

$$N(n;S) = N(n-1;S) + N(n-2;S) \approx c((1+\sqrt{5})/2)^n,$$

the *Fibonacci* number. Therefore,

$$cap(S) = \log_2 (1+\sqrt{5})/2 = .694....$$

Intuitively, this means that, when storing data, using this code each physical bit can, in the asymptotic sense, store at most .694... 'bits of information'.

## 2.2. DEFINITION OF CAPACITY (2-D)

Now let $N(m,n;S)$ be number of $m \times n$ arrays that satisfy constraint $S$.

*The Capacity*, $cap(S)$, of $S$ measures the growth rate of $N(m,n;S)$ :

$$cap(S) = \lim_{n,m \to \infty} \frac{\log_2 N(m,n;S)}{nm}.$$

Intuition: $cap(S)$ is the asymptotically maximum amount of information that can be transmitted or stored per bit of the matrix.

Note: 2-D Capacity "usually" exists independent of double-limits order.
Kato and Zeger (1999).

Following is a list of recent work on bounding the 2-D *capacity* and other properties of such matrices. (Previous work of thesis work is described later.)

- K. Engel (1990)
- N. Calkin and H. Wilf (1998)
- W. Weeks and R. Blahut (1998)
- Z. Nagy and K. Zeger (1998)
- A. Kato and K. Zeger (1999)
- S. Forchhammer and J. Justesen (1999)
- R. Roth, P. H. Siegel, and J.K. Wolf (1999)
- R. Talyansky, T. Etzion, and R. Roth, (1999)
- M. Golin, et al. (1999)
- ...

## 2.3. Definition of Transfer Matrix

Let $C_m$ ($m \geq 0$) be the set of all binary ($m+1$) "row", or "column" vectors $v_i$.

*Transfer matrix* (horizontal, or vertical): Let $T_{|C_m|} = (t_{ij})$ be the $|C_m| \times |C_m|$ matrix indexed by $C_m$ such that:

$$t_{ij} = \begin{cases} 1 & \text{if} \begin{pmatrix} v_i \\ v_j \end{pmatrix}, \text{or } v_i v_j \text{ satisfies } S; \\ 0 & \text{otherwise.} \end{cases}$$

**Example:** In 2-D $(1, \infty)$ RLL constraint, all the legal row vectors with length 2 are

$$\begin{aligned} v_1 &= 00 \\ v_2 &= 01 \\ v_3 &= 10 \end{aligned}$$

then the horizontal transfer matrix (the vertical one can be obtained similarly) is

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}.$$

## 2.4. ABOUT TRANSFER MATRICES

- A constrained code can have *different* transfer matrices for horzontal and vertical directions.
- Transfer matrices usually satisfy recurrence relations.
- The size of transfer matrix is exponentially large compared with the length of the vetors.
- The transfer matrix can be (1) symmetric; or (2) non-symmetric but primitive; or (3) reducible (its graph is not strongly connected).

## 2.5. Why TRANSFER MATRICES?

### Two Known Theorems:

(1) Let $f(m,n)$ be the number of $m \times n$ matrices that satisfy $S_{hs}$. Then

$$f(m,n) = \mathit{1}^t \cdot T^n_{F_{m+3}} \cdot \mathit{1},$$

where $\mathit{1}$ is the vector of all $1's$.

(2) Let $\lambda_m$ be largest eigenvalue of $T_{F_{m+3}}$. Then

$$\begin{aligned}
cap(S_{hs}) &= \lim_{n,m\to\infty} \frac{\log_2 f(m,n)}{nm} \\
&= \lim_{n,m\to\infty} \frac{\log_2 \mathit{1}^t \cdot T^n_{F_{m+3}} \cdot \mathit{1}}{nm} \\
&= \lim_{m\to\infty} \frac{1}{m} \lim_{n\to\infty} \frac{\log_2 \mathit{1}^t \cdot T^n_{F_{m+3}} \cdot \mathit{1}}{n} \\
&= \lim_{m\to\infty} \frac{\log_2 \lambda_m}{m}
\end{aligned}$$

## 2.6. BOUNDING WITH TRANSFER MATRICES

Recall that

$$cap(S_{hs}) = \lim_{m\to\infty} \frac{\log_2 \lambda_m}{m}$$

where $\lambda_m$ is largest eigenvalue of $T_{F_{m+3}}$.

Without knowing all transfer matrices this does not help directly. But, can prove, $\forall m$,

$$\log_2 \frac{\lambda_{2m+1}}{\lambda_{2m}} \quad \leq \quad cap(S_{hs}) \quad \leq \quad \frac{\log_2 \lambda_m}{m}$$

(there are similar types of bounds for associated cylindrical T.Ms).

Current technology for proving bounds on $cap(S_{hs})$ is to calculate as many $\lambda_m$ as possible and plug them into these equations and use best results.

This is very frustrating!
Would like to be more intelligent.

## 3. EXAMPLES OF PREVIOUS WORK

### (I) Capacity of Hard Square System

It seems "not difficult" to consider the capacities of 1-D constraints. BUT no one knows how to calculate closed formulas for capacities of 2-D constraints.

What is known is how to calculate better and better upper and lower bounds on capacities using *transfer matrix techniques*.

$S_{hs}$ often serves as a testbed for developing "better" bounding techniques.

- Weber (1988): $.53602 \leq cap(S_{hs}) \leq .63598$
- Engel (1990): $.58789 \leq cap(S_{hs}) \leq .59756$.
- Calkin and Wilf (1998): $.587891 \leq cap(S_{hs}) \leq .588339$.
- Nagy and Zeger (2000): $.587891161775 \leq cap(S_{hs}) \leq .587891161868$.

**(II) Read/Write Isolated Memory**

- A binary $(0,1)$ memory is *read isolated* if it contains no consecutive 1's (This is the 1-D $(1,\infty)$ RLL); it is *write isolated* if rewritable and no two consecutive positions can be changed during rewriting.

- A *read/write isolated memory* is a binary rewritable storage medium obeying the two *different* restrictions.

  Example: $\begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$, $\begin{pmatrix} \mathit{1} & \mathit{0} & 0 & 1 \\ \mathit{1} & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$. (If we read vertically, write horizontally, then only the first is fine.)

- The read/write isolated medium has two *completely different* transfer matrices (the vertical, and the horizontal).

**Cohn's Results**

Cohn (1995) considered the *Read/Write Isolated Memory.* (He did not consider it as a 2-D code but rather as a time-constrained code. His results are translated into our motivation)

He proved that that the 'horizontal' transfer matrix $A_k$ satisfies a recurrence relation, and that $A_k$ has size $f_{k+2}$, the $(k+2)th$ Fibonacci number.

He obtained by using the matrices that

$$.509... < cap(S_{rw}) < .560...$$

Figure 1: An example on $6 \times 10$ board

**(III) Nonattacking Kings Problem**

Kings placed in an $2m \times 2n$ chessboard and not attack each other, but one king in each $2 \times 2$ cell. (Hard Square $S_{hs}$ with extra constraints.)

Let $N(m,n)$ be the number of ways that $mn$ kings can be placed on a $2m \times 2n$ chessboard. Originally Knuth was interesed in the asymptotics of $N(m,n)$ for $m = n$.

Using transfer matrix techniques, Wilf (1995) obtained that

$$N(m,n) = (c_m n + d_m)(m+1)^n + O(\theta_m^n), \; n \to +\infty.$$

where $c_m > 0$, $d_m$, $0 < \theta_m < m+1$ are **unknown.**

# 4. OUR RESULTS

## 4.1. Overview of Our Work

- establish *theoretical results* (used Hard Square code as a testbed, derived properties of transfer matrices ).
- develop *techniques* for calculating capacity.

The *main* work:

- Decomposition of transfer matrix
- Distribution of positive and negative eigenvalues of $T_{F_{m+3}}$
- Recurrence relation for inverse of $T_{F_{m+3}}$
- Derivation of order of recurrence relation (in $n$) for $f(m, n)$. (Lower than you would think)
- Another expression for $f(m, n)$ in terms of $T_{F_{m+3}}$
- Compressing transfer matrix—-new technique for upper bounding capacity
- Peeling off unimportant eigenvalues—-new technique for two-sided bounding capacity
- Transfer matrix of kings problem
- The aymptotics of the number of placements

  . . .

## 4.2. TRANSFER MATRIX of $S_{hs}$

(**Lemma 2.1.**) Index $F_{m+3}$ columns in lexicographical order. The transfer matrix satisfies

$$T_{F_{m+3}} = \begin{pmatrix} T_{F_{m+2}} & T_{F_{m+2}}^{(F_{m+1})} \\ * & 0_{F_{m+1}} \end{pmatrix}.$$

$T_{F_{m+2}}^{(F_{m+1})}$ is the first $F_{m+1}$ columns of $T_{F_{m+2}}$;

$0_{F_{m+1}}$ a null matrix;

"$*$" stands for the transpose;

The first three matrices are:

$$T_{F_3} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, T_{F_4} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}, T_{F_5} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

## 4.3. THE $L^tDL$ DECOMPOSITION

(**Lemma 2.2.**) $T_{F_{m+3}}$ has the following $L^tDL$ decomposition:

$$T_{F_{m+3}} = L^t_{F_{m+3}} D_{F_{m+3}} L_{F_{m+3}},$$

where

$$L_{F_{m+3}} = \begin{pmatrix} L_{F_{m+2}} & \begin{pmatrix} L_{F_{m+1}} \\ 0 \end{pmatrix} \\ 0 & L_{F_{m+1}} \end{pmatrix}, \quad D_{F_{m+3}} = \begin{pmatrix} D_{F_{m+2}} & 0 \\ 0 & -D_{F_{m+1}} \end{pmatrix}.$$

Initial conditions are

$$L_{F_3} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad L_{F_4} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

and

$$D_{F_3} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad D_{F_4} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}.$$

An Example:

$$T_{F_5} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} L^t_{F_4} & 0 \\ (L^t_{F_3}, 0) & L^t_{F_3} \end{pmatrix} \begin{pmatrix} D_{F_4} & 0 \\ 0 & -D_{F_3} \end{pmatrix} \begin{pmatrix} L_{F_4} & \begin{pmatrix} L_{F_3} \\ 0 \end{pmatrix} \\ 0 & L_{F_3} \end{pmatrix}$$

$$= L^t_{F_5} D_{F_5} L_{F_5}.$$

## 4.4. EIGENVALUE DISTRIBUTION AND INVERSE

(**Theorem 2.1.**) Let $P_m$ and $N_m$ be the number of positive and negative eigenvalues of $T_{F_{m+3}}$. Then

$$N_m = N_{m-1} - N_{m-2} + F_{m-2}$$

with $N_0 = 1$ and $N_1 = 2$ and

$$P_m - N_m = -\frac{2}{\sqrt{3}} \sin \frac{m\pi}{3}.$$

$(|P_m - N_m| \leq 1)$

**(Corollary 2.1.)** The inverse of $T_{F_{m+3}}$ is an $(-1,\ 0,\ 1)-$ matrix satisfying

$$T_{F_{m+3}}^{-1} = \begin{pmatrix} \begin{pmatrix} -T_{F_m}^{-1} & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} T_{F_m}^{-1} \\ 0 \end{pmatrix} & T_{F_{m+1}}^{-1} \\ (T_{F_m}^{-1}, 0) & -T_{F_m}^{-1} & 0 \\ T_{F_{m+1}}^{-1} & 0 & -T_{F_{m+1}}^{-1} \end{pmatrix}.$$

## 4.5. THE ORDER OF $f(m,n)$'s R.R.

**(Theorem 2.2.)** Would assume that, for fixed $m$, $f(m,n)$ satisfies a recurrence relation of order $F_{m+3}$. $f(m,n)$ actually satisfies a r.r. with order at most

$$\begin{cases} (F_{m+3}+F_{\frac{m+3}{2}})/2, & \text{if } m \text{ is odd;} \\ (F_{m+3}+F_{\frac{m+6}{2}})/2, & \text{if } m \text{ is even.} \end{cases}$$

**Example:** The recurrence relations for $m = 1, 2, 3$ have, respectively, orders $2, 4, 5$ (instead of $3, 5, 8$)

$$f(n,1) = 2f(n-1,1) + f(n-2,1);$$

$$f(n,2) = 2f(n-1,2) + 6f(n-2,2) - f(n-4,2);$$

$$\begin{aligned} f(n,3) = {} & 4f(n-1,3) + 9f(n-2,3) - 5f(n-3,3) \\ & -4f(n-4,3) + f(n-5,3). \end{aligned}$$

(Initial conditions are given by $f(m,n) = f(n,m) = \mathit{1}^t T^n_{F_{m+3}} \mathit{1}$).

## 4.6. ANOTHER EXPRESSION FOR $f(m, n)$

**(Theorem 2.4.)** Let $\varphi(\lambda) = det(\lambda I - T_{F_{m+3}})$ be the characteristic polynomial of $T_{F_{m+3}}$. Then

$$f(m,n) = \frac{|\lambda_m I - T_{11}|}{\varphi'(\lambda_m)}\lambda_m^{n+2} + \frac{|\lambda_{j_2} I - T_{11}|}{\varphi'(\lambda_{j_2})}\lambda_{j_2}^{n+2} + \cdots + \frac{|\lambda_{j_{r_m}} I - T_{11}|}{\varphi'(\lambda_{j_{r_m}})}\lambda_{j_{r_m}}^{n+2},$$

where

$$r_m \leq \begin{cases} (F_{m+3} + F_{\frac{m+3}{2}})/2, & \text{if } m \text{ is odd;} \\ (F_{m+3} + F_{\frac{m+6}{2}})/2, & \text{if } m \text{ is even.} \end{cases}$$

$\lambda_m$ is the largest eigenvalues of $T_{F_{m+3}}$.

$\lambda_{j_2}, \lambda_{j_3}, \ldots, \lambda_{j_{r_m}}$ is a specified subset of the eigenvalues of $T_{F_{m+3}}$.

$T_{11}$ is the bottom-right $(F_{m+3} - 1) \times (F_{m+3} - 1)$ submatrix of $T_{F_{m+3}}$ i.e.,

$$T_{F_{m+3}} = \begin{pmatrix} 1 & \mathit{1}^t \\ \mathit{1} & T_{11} \end{pmatrix},$$

and $\frac{|\lambda_j I - T_{11}|}{\varphi'(\lambda_j)} > 0$.

## 4.7. New bounds on capacity of read/write isolated memory

- Cohn (1995) proved

$$0.509... \leq C \leq 0.560297....$$

- Golin et al. (1999) improved to

$$0.53500... \leq C \leq 0.55209....$$

- We proved

$$0.5350150... \leq C \leq 0.5396225...$$

  by introducing a new *compressed matrix technique.*

## 4.8. Upper Bounding Capacity

Let $Q_0(x) = 1, \quad Q_1(x) = \begin{pmatrix} 1 & x \\ x & x \end{pmatrix}$, and

$$Q_m(x) = \begin{pmatrix} Q_{m-1}(x) & \hat{Q}_{m-2}(x) \\ \hat{Q}_{m-2}(x)^t & Q_{m-2}(x) \end{pmatrix}, \quad \hat{Q}_{m-2}(x) = \begin{pmatrix} Q_{m-2}(x) \\ 0 \end{pmatrix},$$

where $Q_m$ is of size $f_{m+2} \times f_{m+2}$. Then

(**Lemma 3.3.**) Let $\rho(Q_m(x))$ be the largest eigenvalue of $Q_m(x)$. Then,

$$\forall m \geq 1, \quad C \leq \frac{\log_2 \rho\left(Q_m\left(2^{-C}\right)\right)}{m},$$

where $C$ is its capacity.

(**Theorem 3.1.**) For $\forall x \in [0.69015866, 1)$ and $\forall m \geq 1$, the largest eigenvalue of compressed matrix $Q_m(x)$ satisfies

$$2^C \leq (\rho(Q_m(x)))^{\frac{1}{m}} < (\rho(A_m))^{\frac{1}{m}} = \lambda_m^{1/m},$$

where $Q_m(1) = A_m$.

| $m$ | $F_{m+2}$ | $\rho(Q_m(x_0))$ | $\frac{\log_2(\rho(Q_m(x_0)))}{m}$ |
|---|---|---|---|
| 1 | 2 | 1.5524196988 | 0.6345186448 |
| 2 | 3 | 2.1732673267 | 0.5599328234 |
| 3 | 5 | 3.2543210622 | 0.5674521968 |
| 4 | 8 | 4.6152889202 | 0.5516052411 |
| 5 | 13 | 6.8116188717 | 0.5535995429 |
| 6 | 21 | 9.7408000844 | 0.5473400461 |
| 7 | 34 | 14.2668740681 | 0.5477996234 |
| 8 | 55 | 20.5065208518 | 0.5447513549 |
| 9 | 89 | 29.9058714421 | 0.5447063169 |
| 10 | 144 | 43.0634915800 | 0.5428393393 |
| 11 | 233 | 62.7213694907 | 0.5428077414 |
| 12 | 377 | 90.6138568737 | 0.5418049818 |
| 13 | 610 | 131.2975410982 | 0.5412843145 |
| 14 | 987 | 190.3410264771 | 0.5408887676 |
| 15 | 1597 | 276.1769813000 | 0.5406299513 |
| 16 | 2584 | 399.7392197852 | 0.5401822073 |
| 17 | 3181 | 579.6907368666 | 0.5399493895 |
| 18 | 5765 | 839.3943441870 | 0.5396224962 |

Table 1: $F_m$ is the $m$th Fibonacci number. $\rho(Q_m(x_0))$ is the largest eigenvalue of the $F_{m+2} \times F_{m+2}$ compressed matrix $Q_m(x_0)$.

## 4.9. LOWERBOUNDING CAPACITY $C$

**Lemma 3.4.** Let $p(x) = det(xI - \bar{A}_r)$ be the characteristic polynomial of the $2^r \times 2^r$ horizontal transfer matrix $\bar{A}_r$. Then

$$p(x) = det(xI - \bar{A}_r) = x^{2^{r-1}} det(xI - M_r),$$

where $M_r$ is an $2^{r-1} \times 2^{r-1}$ nonnegative matrix.

(Previous work required calculating eigenvalues of $2^r \times 2^r$.)

Table 2 gives the numerical results of our computations on the largest eigenvalues of $\bar{A}_r$ and the bounds of the capacity. We computed the largest eigenvalues of $DM_rD^{-1}$ using *Power Method* and Matlab.

| $r$ | $2^r$ | $\mu_r$ | $\log_2 \frac{\mu_r}{\mu_{r-1}}$ |
|---|---|---|---|
| 1 | 2 | 1.618 | |
| 2 | 4 | 2.302775637 | 0.5091622465 |
| 3 | 8 | 3.346462191 | 0.5392628601 |
| 4 | 16 | 4.845619214 | 0.5340443229 |
| 5 | 32 | 7.021562462 | 0.5351110622 |
| 6 | 64 | 10.17359346 | 0.5349653450 |
| 7 | 128 | 14.74105370 | 0.5350103028 |
| 8 | 256 | 21.35908135 | 0.5350099454 |
| 9 | 512 | 30.948359597568 | 0.535010307223 |
| 10 | 1024 | 44.842824649376 | 0.535014214230 |
| 11 | 2048 | 64.975322373528 | 0.535014730251 |
| 12 | 4096 | 94.146459043118 | 0.535014947823 |
| 13 | 8192 | 136.414197132806 | 0.535015053352 |
| 14 | 16384 | 197.658348650048 | 0.535015093868 |

Table 2: $\mu_r$ is the largest eigenvalue of the $2^r \times 2^r$ transfer matrix $\bar{A}_r$. For $r > 8$ the $\mu_r$ were calculated by calculating the largest eigenvalues of $DM_rD^{-1}$.

Plugging the values in Table 2 into the inequalities on page 5 derives our new lower bound $0.5350150 \leq C$.

## 4.10. NONATTACKING KINGS PROBLEM

Its transfer matrix is reducible (neither symmetric, nor primitive, nor irreducible).

**The current techniques can not be applied to this problem.**

By proving that its matrix $\Lambda_m$ is *persymmetric*, *i.e*,

$$(\Lambda_m)_{i,j} = (\Lambda_m)_{n-j+1,n-i+1},$$

we obtained the constants $c_m > 0$, $d_m$, $\theta_m$ in Wilf's formula

$$N(m,n) = (c_m n + d_m)(m+1)^n + O(\theta_m^n),$$

This permits, for example, showing that capacity

$$cap(S) = \lim_{n,m\to\infty} \frac{\log_2 N(m,n;S)}{nm} = 0.$$

## 4.11. Transfer matrix of kings problem

The first two matrices are given by.

$$
\begin{pmatrix}
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1
\end{pmatrix},
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\
\\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1
\end{pmatrix}.
$$

Generally, it has size $(m+1)2^m$.

## REWIEW —- OVERVIEW OF THE WORK

- establish *theoretical results* (use Hard Square code as a testbed), then apply them to the analyses and computations.
- develop *techniques* for calculating capacity that reduces the complexity of computations so as to tighten the bounds.

The *main* work:

- Decomposition of transfer matrix
- Distribution of positive and negative eigenvalues of $T_{F_{m+3}}$
- Recurrence relation for inverse of $T_{F_{m+3}}$
- Derivation of order of recurrence relation (in $n$) for $f(m, n)$. Lower than you would think
- Another expression for $f(m, n)$ in terms of $T_{F_{m+3}}$
- Compressing transfer matrix—-new technique for upper bounding capacity
- Peeling off unimportant eigenvalues—-new technique for two-sided bounding capacity
- Transfer matrix of kings problem

- The asymptotics of the number of placements
- $\cdots$

## 5. POSSIBLE EXTENSIONS

- Apply the techniques to other problems, e.g., Eulerian orientations on a grid torus

- A Conjecture posed by Engel (1990) for hard square system:

$$\log_2\left(\frac{\lambda_{2m}}{\lambda_{2m-1}}\right) \geq cap(S). \qquad (1)$$

This conjecture seems also true for other constraints from numerical results, (e.g., for Read/Write constraints.)

If it is true, it would dramatically improve all known results.