Counting Structures in Grid Graphs, Cylinders and Tori Using Transfer Matrices: Survey and New Results

Mordecai J. Golin* Yiu Cho Leung* Yajun Wang* Xuerong Yong**

*Department of Computer Science

Hong Kong Univeristy of Science and Technology

**DIMACS

Outline

1. **Introduction:**
   Graphs & Problems

2. Survey of Results

3. The Transfer Matrix Technique

4. Related Work & Open Problems

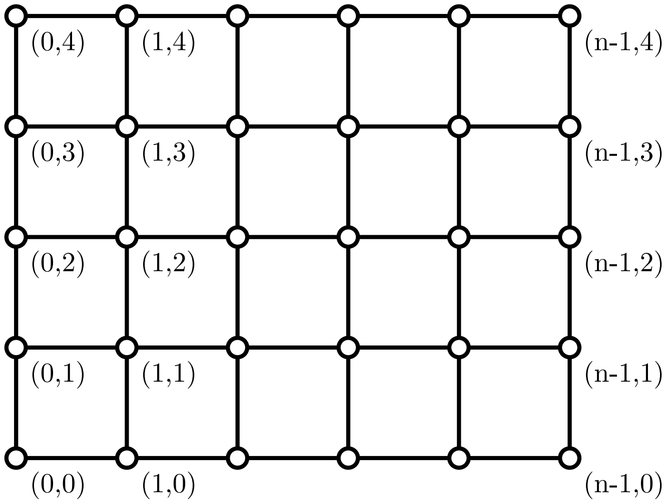
Four Graphs & a Problem

(n, m) denotes width n and height m

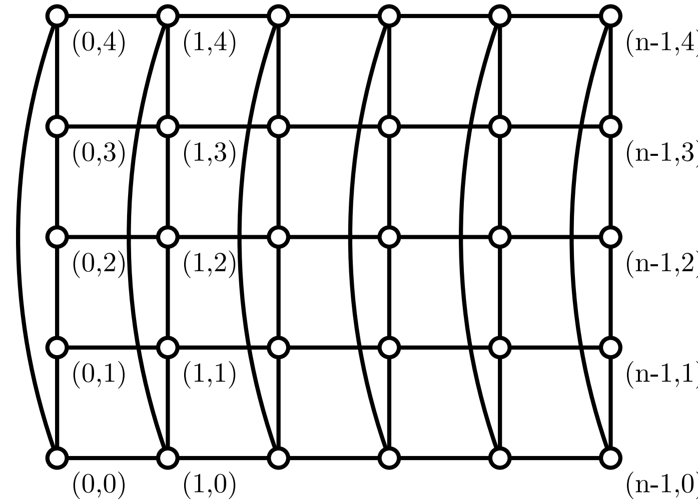
- $G(n, m)$: *Grid Graph*
- $FC(n, m)$: *Fat-Cylinder*
  Connect left side of Grid to right
- $TC(n, m)$: *Thin-Cylinder*
  Connect bottom of Grid to top
- $T(n, m)$: *Torus*
  Connect left side of Thin-Cylinder to right

The Problem: Count the number of structures, e.g., spanning trees, Hamiltonian paths, independent sets, etc., in the given graphs.

Four Graphs

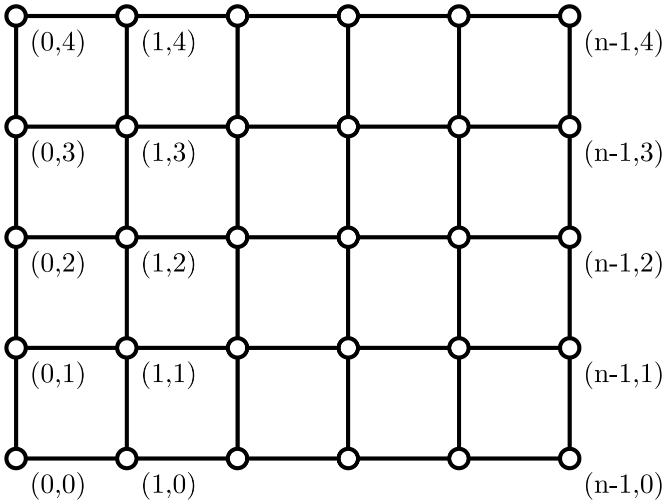


Grid Graph $G(6,5)$

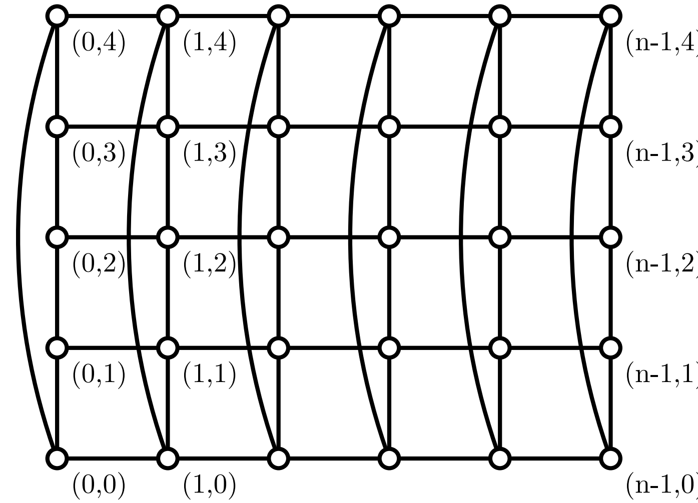


Thin Cylinder $TC(6,5)$

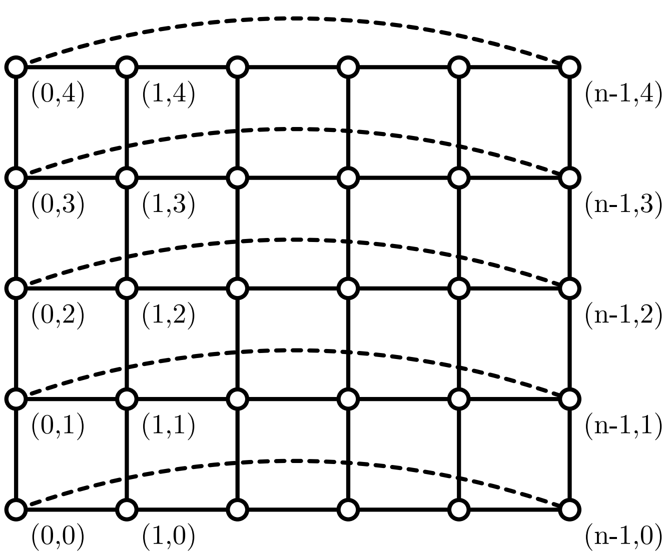
Four Graphs



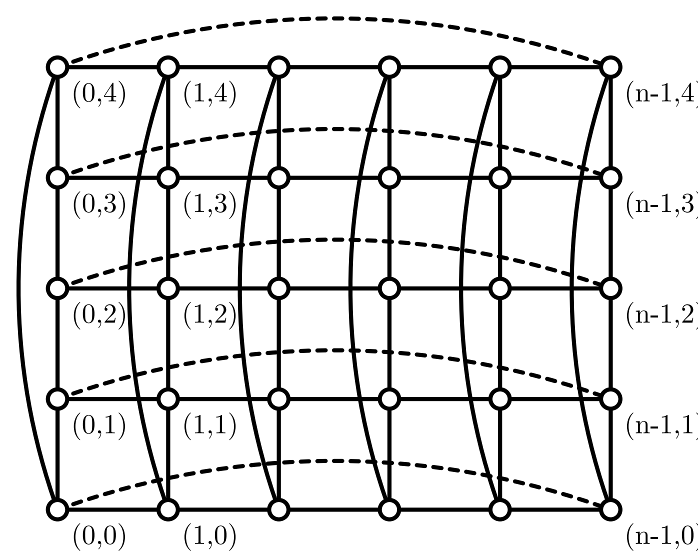
Grid Graph $G(6,5)$



Thin Cylinder $TC(6,5)$



Fat Cylinder $FC(6,5)$



Torus $T(6,5)$

The Problem

Fix a structure type $\mathcal{S}$, e.g, spanning trees (ST), Hamiltonian paths, independent sets, etc..

Let $\mathcal{G}$ be one of the four graphs introduced.

Let $\mathcal{S}_{\mathcal{G}}(n, m)$ be *set of $\mathcal{S}$ structures in graph $\mathcal{G}(n, m)$*.
e.g., $ST_T(m, n)$ is set of spanning trees in $n \times m$ torus.

The Problem

Fix a structure type $\mathcal{S}$, e.g, spanning trees (ST), Hamiltonian paths, independent sets, etc..

Let $\mathcal{G}$ be one of the four graphs introduced.

Let $\mathcal{S}_{\mathcal{G}}(n,m)$ be *set of $\mathcal{S}$ structures in graph* $\mathcal{G}(n,m)$.
e.g., $ST_T(m,n)$ is set of spanning trees in $n \times m$ torus.

Generic problem: For fixed $\mathcal{S}, \mathcal{G}$, calculate $|\mathcal{S}_{\mathcal{G}}(n,m)|$

"Usual" problem:
For fixed $\mathcal{S}, \mathcal{G}$ and fixed m (height):
(i) derive closed form for $f(n) = |\mathcal{S}_{\mathcal{G}}(n,m)|$
(ii) show that $f(n)$ satisfies constant-order linear recurrence relation (RR)

History

Came to this problem via circulant graphs (see later).

Reviewing literature noted large number of
such results on grids for many different structures
most using same technique (*transfer matrix*),
many without knowing about other papers,
or that technique is well-known
Almost no results on tori.

History

Came to this problem via circulant graphs (see later).

Reviewing literature noted large number of
such results on grids for many different structures
most using same technique (*transfer matrix*),
many without knowing about other papers,
or that technique is well-known
Almost no results on tori.

This paper:
(i) survey of results, put all into same general framework
(ii) one "new" result *(Eulerian Tours on tori)*
(iii) observation that, with small twist,
framework will work for tori (and cylinders) as well,
so all structures can be counted on tori (cylinders).
(iv) uses general framework to introduce open questions.

Outline

1. Introduction:

   Graphs & Problems

2. Survey of Results

3. The Transfer Matrix Technique

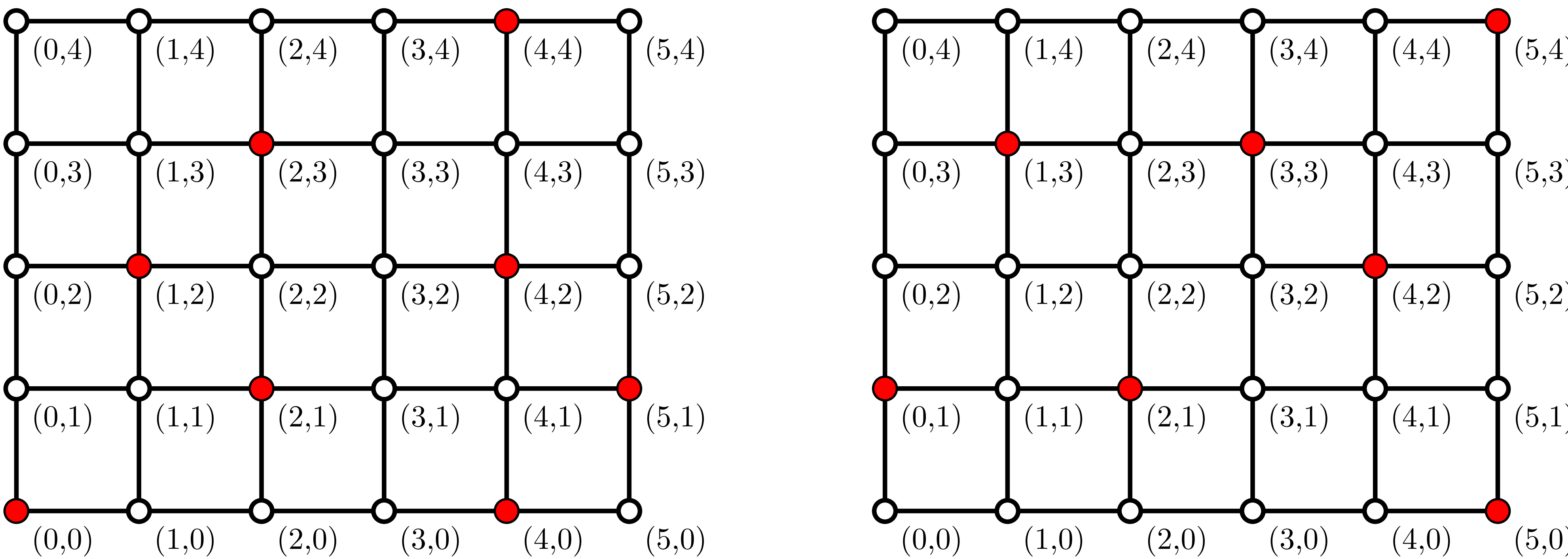
4. Related Work & Open Problems

Counting Structures in Grids

- Independent Sets/$2D$ $(1, \infty)$ RLL codes
- Dimer Matchings
- Hamilton Cycles
- Spanning Trees/Forests
- Eulerian Orientations/Ice Condition
- Eulerian Tour
- Cycle Covers/Directed Cycle Covers
- Acyclic Orientations
- Colorings

Independent Sets

Two independent sets of $G(6,5)$:



Independent Sets

An **Independent Set** is a set of vertices s.t.
$V' \subseteq V \; s.t. \forall u, v \in V', (u, v) \notin E$.

Independent sets in grid graphs are in 1-1 correspondence with 2-Dimensional $(1, \infty)$ run-length limited codes.

- K. Engel *On the Fibonacci number of an $M \times N$ lattice* 1990
- N. Calkin and H. Wilf *The number of independent sets in a grid graph.* 1998
- A. Kato and K. Zeger *On the capacity of two-dimensional run-length constrained channels.* 1999
- R. M. Roth et al. *Efficient coding schemes for the hard-square model.* 2001
- S. Halevy et al. *Improved bit-stuffing bounds on two-dimensional constrains.* 2004

Dimer Matchings

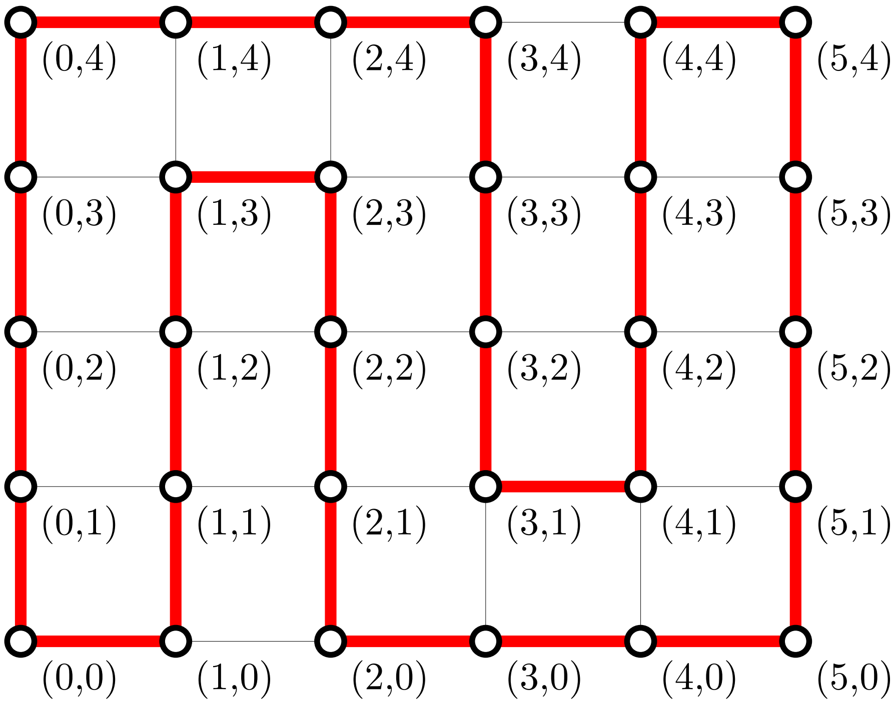
A **Dimer Matching** is a placement of 1×2 "dominos" that covers $\mathcal{G}$ such that a domino covers nodes u, v iff $(u, v) \in E$.

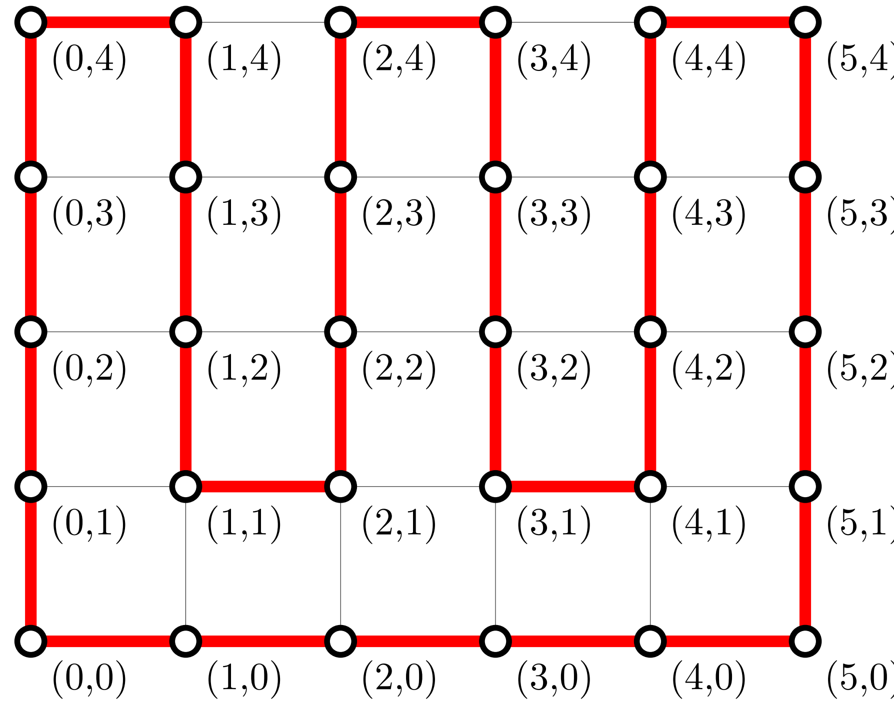
Only complicated structure in which a "closed formula" is known in m, n for counting problem in $G(m, n)$

- H. N. V. Temperley and M. E. Fisher. *Dimer problem in statistical mechanics: an exact result.* 1961
- P. W. Kasteleyn. *The statistics of dimers on a lattice, i. the number of dimer arrangements on a quadratic lattice.* 1961
- R. Stanley. *On dimer coverings of rectangles of fixed width.* 1985

Hamiltonian Cycles

Two Hamiltonian cycles of $G(6,5)$:





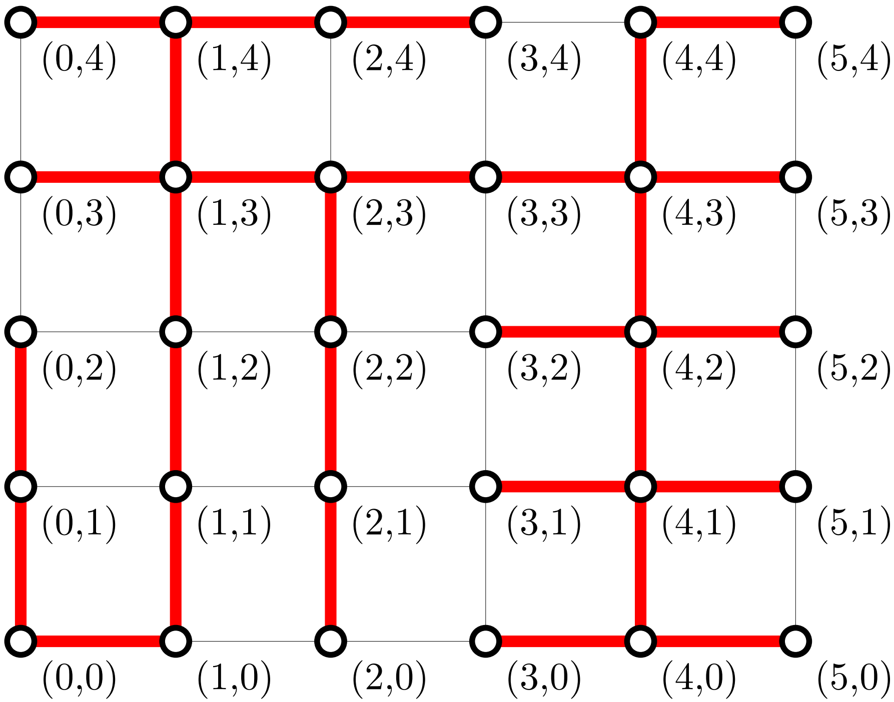
Hamiltonian Cycles

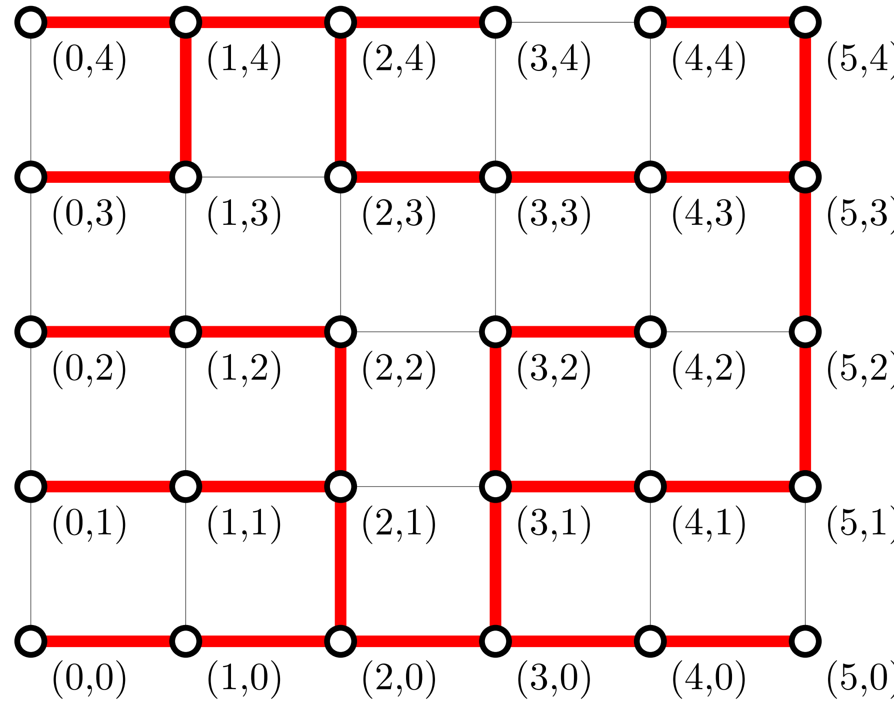
A simple cycle that contains all of the vertices.

- Y. H. Kwong and D. G. Rogers. *A matrix method for counting Hamiltonian cycles on a grid graphs.* 1994
- R. Stoyan and V. Strehl. *Enumerations of Hamiltonian circuits in rectangular grids.* 1996
  nice use of Motzkin words to enumerate "states"

Spanning Trees (Forests)

Two spanning trees of $G(6,5)$:





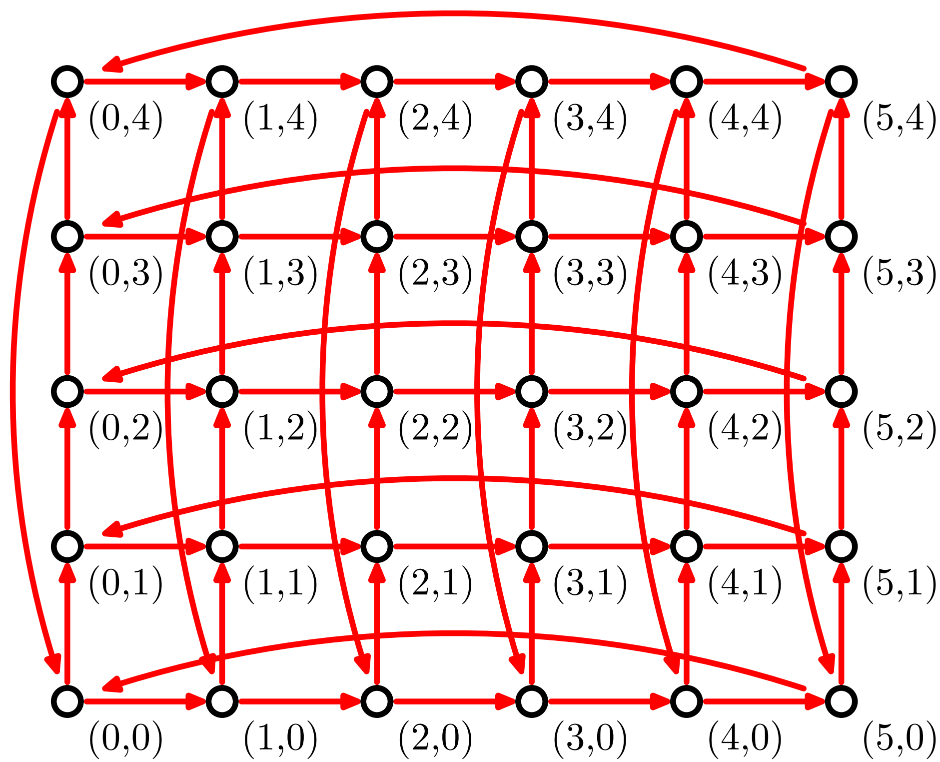
Spanning Trees (Forests)

A **Spanning Tree** is a connected acyclic subgraph containing all vertices.

- C. Merino and Welsh. *Forest, colourings and acyclic orientations of the square lattice.* 1999
- R. Shrock and F. Y. Wu. *Spanning trees on graphs and lattices in d dimensions.* 2000
- M. Desjarlais and R. Molina. *Counting spanning trees in grid graphs.* 2000

Eulerian Orientations/Ice Condition

A simple example for $T(6,5)$:



Eulerian Orientations/Ice Condition

An **Eulerian Orientation (EO)** is orientation of edges in which every vertex has indegree = outdegree = 2 (only defined for tori).

- E. Lieb. *The residual entropy of square ice.* 1967

Lieb calculates closed form for *entropy*,

$$\lim_{n,m\to\infty} |EO_T(m,n)|^{1/mn}$$

Corresponding limit is not known for *any* other problem, despite extensive study for Independent sets
(this is Fibonacci number of graph and
$\exp$ of *capacity* of $2D$ RLL $(1,\infty)$-codes)

Eulerian Tour

An **Eulerian Tour** is an orientation of the edges along with a cyclical ordering of the edges such that the source of each edge is equal of the sink of its predecessor (only defined for tori).

- *this paper.*

Cycle Covers/Directed Cycle Cover

A **Cycle Cover** is a collection of simple cycles that together contain each vertex exactly once.

A **Directed Cycle Cover** is a *Cycle Cover* along with an orientation (clockwise/counterclockwise) of each vertex.

- H. L. Abbott and J. W. Moon. *On the number of cycle covers of the rectangular grid graph.* 1999

Acyclic Orientations

An **Acyclic Orientation** is an orientation of the edges that contains no directed cycle.

- C. Merino and Welsh. *Forests, colourings and acyclic orientations of the square lattice.* 1999
- N. Calkin et al. *Improved bounds for the number of forests and acyclic orientations in the square lattice.* 2003

k Colorings

A k **Coloring** is a function $f : E \to \{1, \ldots, k\}$ such that if $(u, v) \in E$ then $f(u) \neq f(v)$.

- C. Merino and Welsh. *Forests, colourings and acyclic orientations of the square lattice.* 1999

Outline

1. Introduction: Graphs & Problems

2. Survey of Results

3. **The Transfer Matrix Technique**

4. Related Work & Open Problems

Structural Properties of the Graphs

- $G(n+1, m)$ **can be built recursively from size** n **grid (and** $TC(n+1, m)$ **from size** n **thin-cylinder) by adding constant # of edges (independent of value of** n**) to right side.**

  This is property used by most papers to derive recurrence relation (RR) on $f(n)$ for G and TC.

Structural Properties of the Graphs

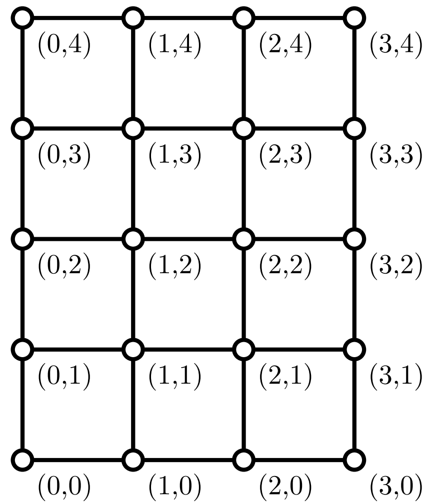
- **$G(n+1,m)$ can be built recursively from size n grid (and $TC(n+1,m)$ from size n thin-cylinder) by adding constant # of edges (independent of value of n) to right side.**

  This is property used by most papers to derive recurrence relation (RR) on $f(n)$ for G and TC.

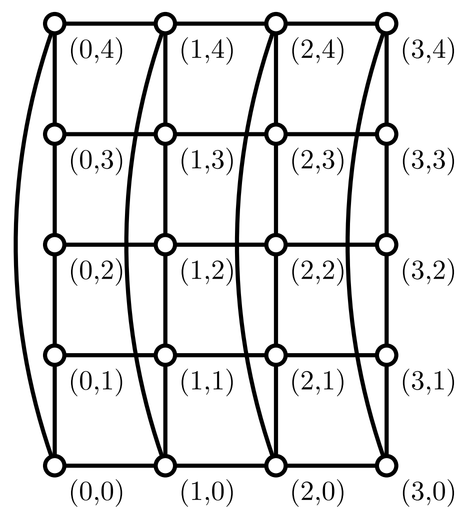
- **$FC(n,m)$ can be built from $G(n,m)$ (and $T(n,m)$ built from $TC(n,m)$) by adding constant # of edges, indep. of value of n. These are "hooking" edges, connecting left border to right**

  This is property we can use to derive RR for $f(n)$ when graph is FC and T.

Recursive Construction

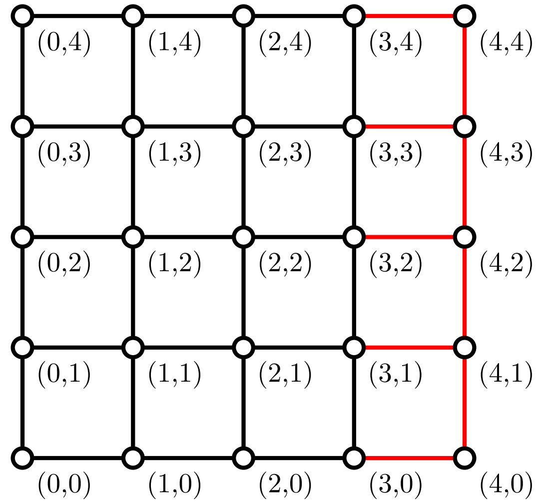


$$E_G(n+1,m) = E_G(n,m) \cup RtG(n,m)$$

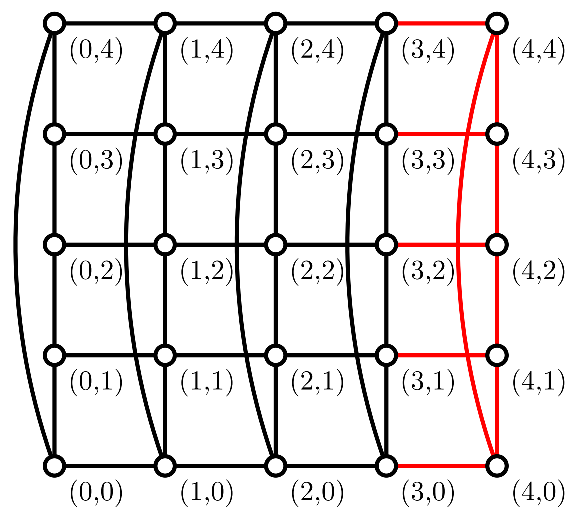


$$E_{TC}(n+1,m) = E_{TC}(n,m) \cup RtTC(n,m)$$

Recursive Construction

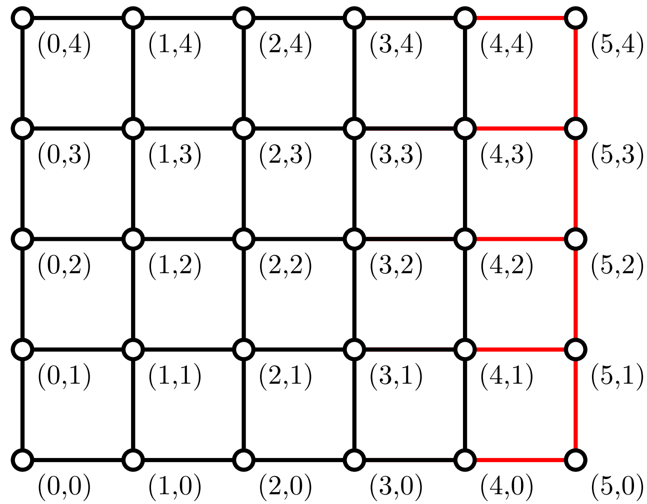


$$E_G(n+1,m) = E_G(n,m) \cup RtG(n,m)$$

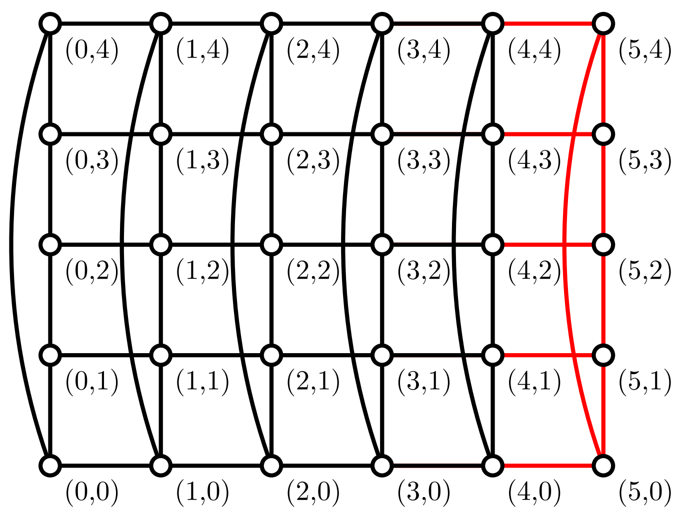


$$E_{TC}(n+1,m) = E_{TC}(n,m) \cup RtTC(n,m)$$

Recursive Construction

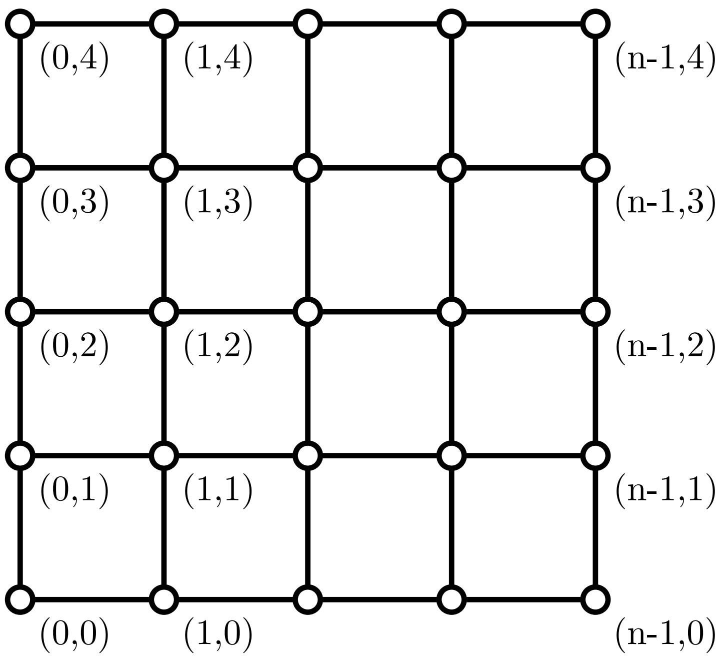


$$E_G(n+1,m) = E_G(n,m) \cup RtG(n,m)$$

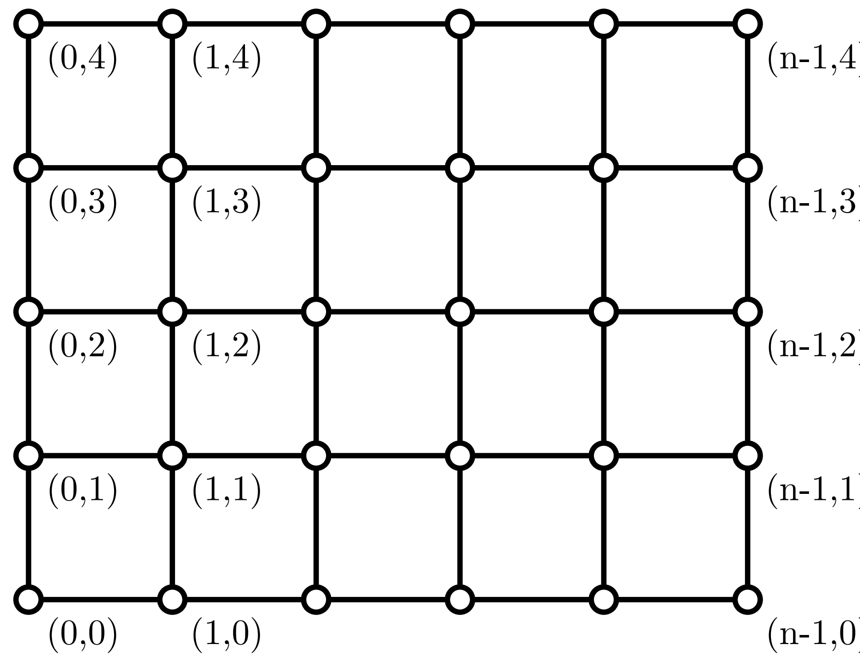


$$E_{TC}(n+1,m) = E_{TC}(n,m) \cup RtTC(n,m)$$

$E_{FC}(n,m) = E_G(n,m) \cup \mathtt{Side}(n,m)$

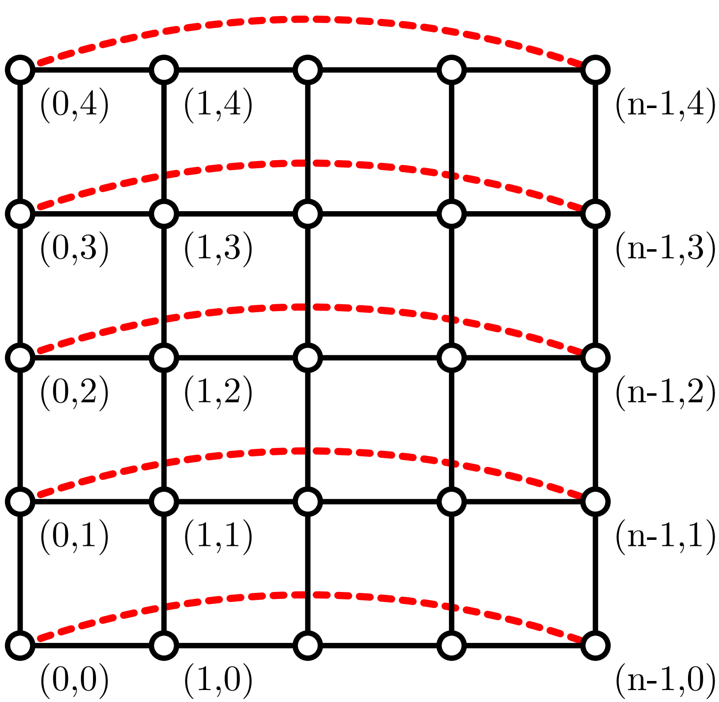


Grid Graph $G(5,5)$

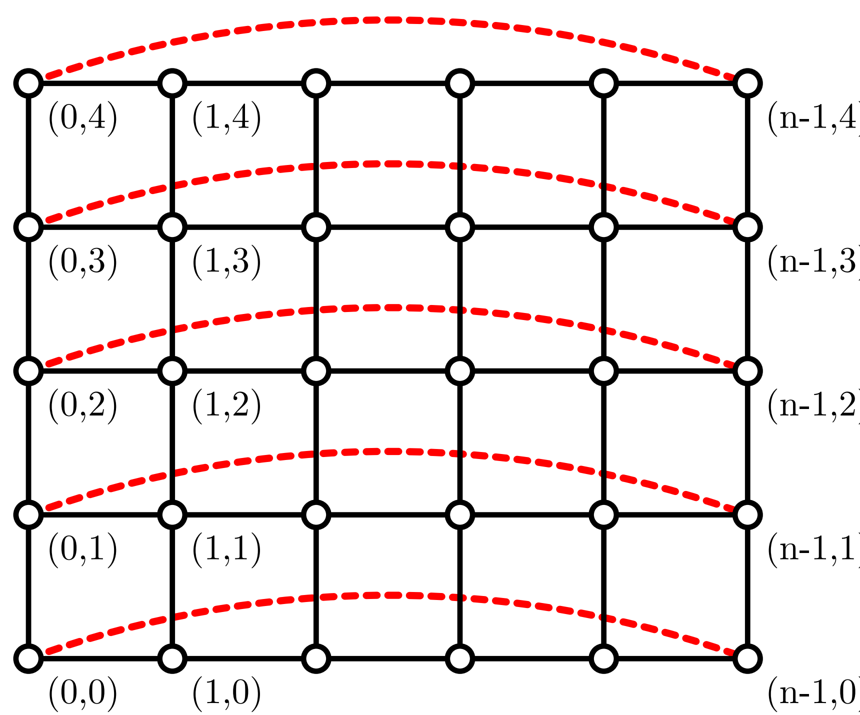


Grid Graph $G(6,5)$

$E_{FC}(n,m) = E_G(n,m) \cup \mathtt{Side}(n,m)$



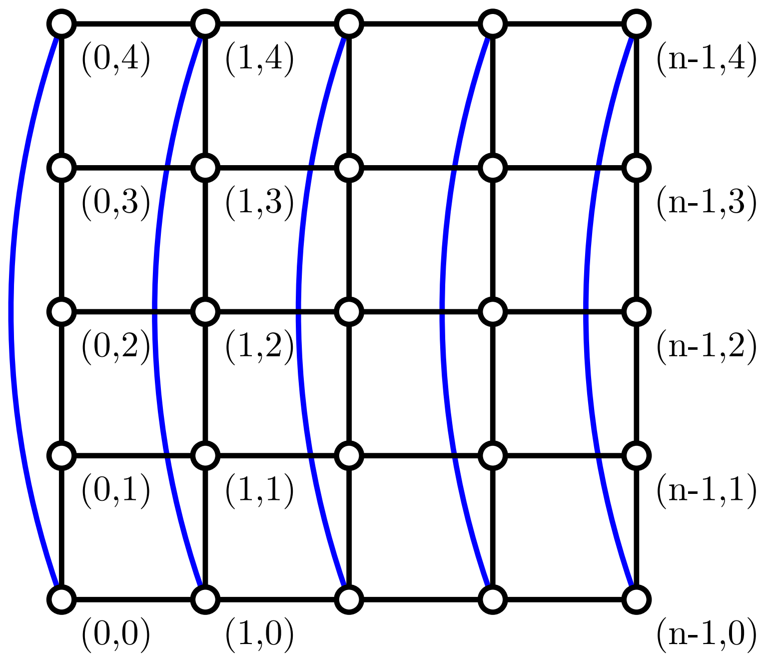
Fat Cylinder $FC(5,5)$



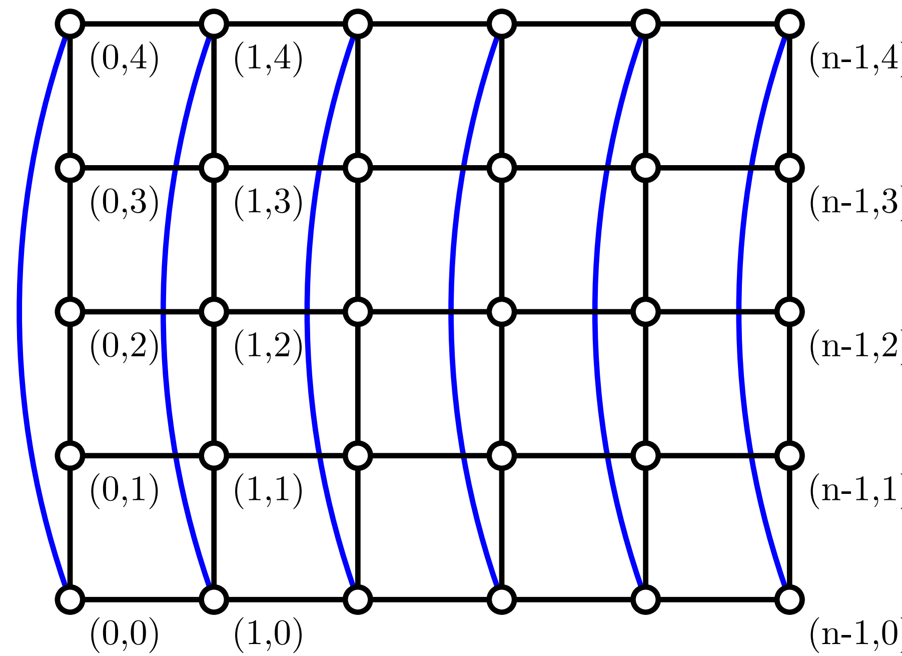
Fat Cylinder $FC(6,5)$

$Side(n,m)$ is independent of n

$E_T(n,m) = E_G(n,m) \cup \texttt{Top}(n,m) \cup \texttt{Side}(n,m)$

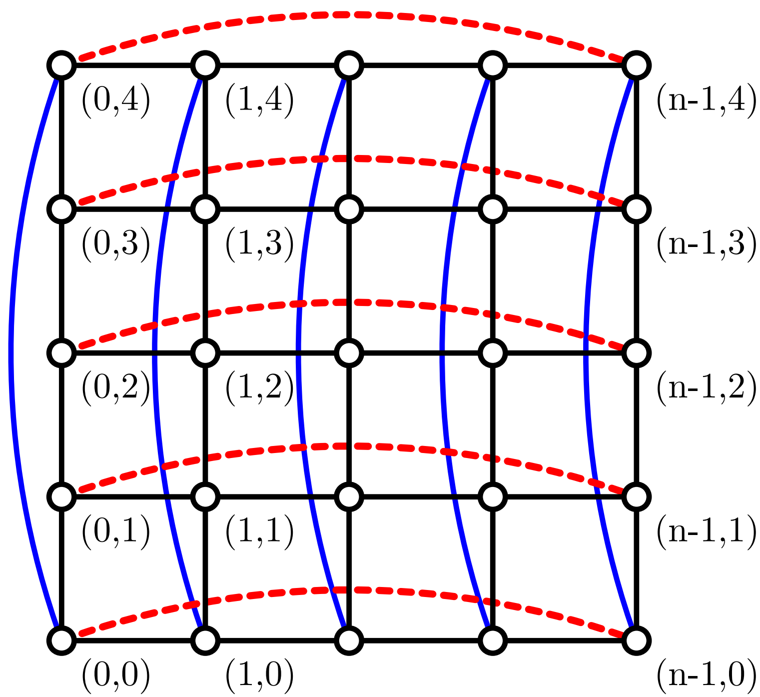


Thin Cylinder $FC(5,5)$

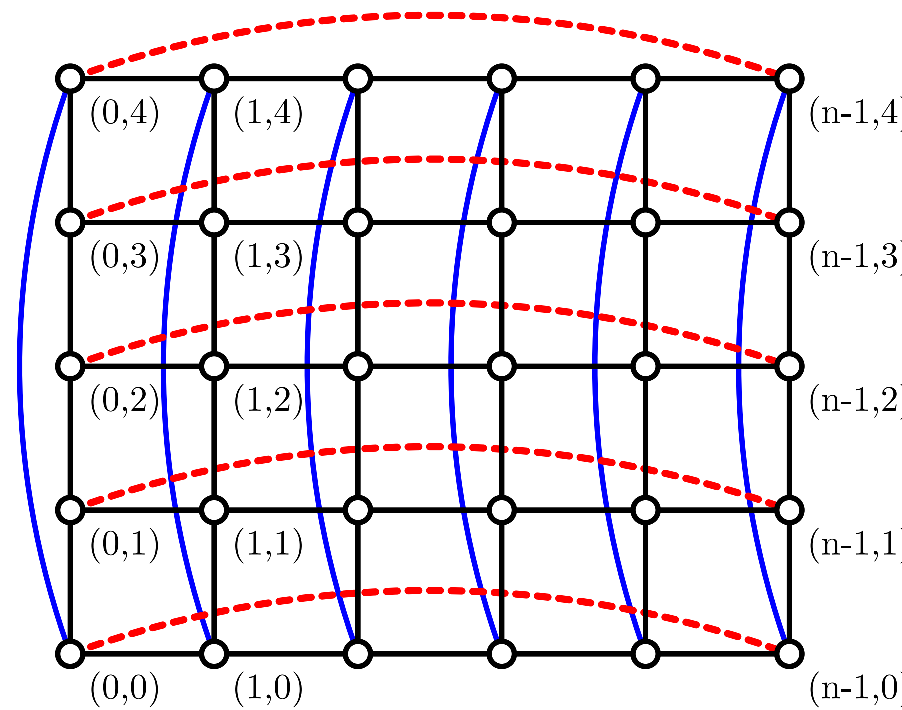


Thin Cylinder $FC(6,5)$

$$E_T(n,m) = E_G(n,m) \cup \texttt{Top}(n,m) \cup \texttt{Side}(n,m)$$



Torus $T(5,5)$



Torus $T(6,5)$

$Side(n,m)$ are independent of n

Recursively Constructable Graphs

A family G_n of graphs is *recursive* if its Tutte polynomial satisfies a linear recurrence relation with polynomial coefficients.

Biggs,Damerell, Sands (1972)

In this case the number of spanning trees and acyclic orientations will also satisfy a recurrence relation.

Recursively Constructable Graphs

We have seen that for $\mathcal{G} \in \{G, FC, TC, T\}$,
$\mathcal{G}(n+1, m)$ can be built from $\mathcal{G}(n, m)$ by
adding & subtracting a constant set of edges indep. of n

Such graphs were recently (2004) labelled
recursively constructable by Noy and Ribó who showed that
recursively constructable implies recursive
(and used grids, tori and cylinders as examples)

This immediately implies some of the results in the survey
but not others.

We now show that, as long as they obey a small # of
abstract properties, many other structures can be directly
counted by using the transfer matrix method.

Legal objects and a ST example

Fix m; Define $\mathcal{L}(n,m)$, the set of legal objects in $G(n,m)$;
superset of set of (good) $\mathcal{S}$-structures.

Legal objects **and a** ST **example**

Fix m; Define $\mathcal{L}(n,m)$, the set of legal objects in $G(n,m)$; superset of set of (good) $\mathcal{S}$-structures.

For ST in $G(n,m)$, legal objects are *forests* in which each component contains at least one "side" vertex.

Legal objects and a ST example

Fix m; Define $\mathcal{L}(n,m)$, the set of legal objects in $G(n,m)$; superset of set of (good) $\mathcal{S}$-structures.

For ST in $G(n,m)$, legal objects are *forests* in which each component contains at least one "side" vertex.

Define P, set of object *Classifications*.
For $L \in \mathcal{L}(n,m)$, $C(L) \in P$.
$\forall$ structures S, *and legal objects* L,
if* $C(L) = C(S) \Rightarrow L$ *is a structure.
***Define* $\mathcal{L}_X(n,m) = \{L \in \mathcal{L}(n,m),\ C(L) = X\}$.**

Important: Classification does *not* depend upon *value* of n.

Legal objects and a ST example

Fix m; Define $\mathcal{L}(n,m)$, the set of legal objects in $G(n,m)$;
superset of set of (good) $\mathcal{S}$-structures.

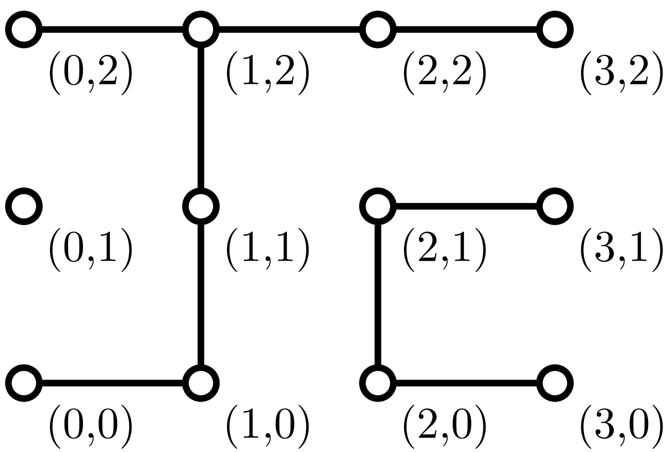
For ST in $G(n,m)$, legal objects are *forests* in which
each component contains at least one "side" vertex.

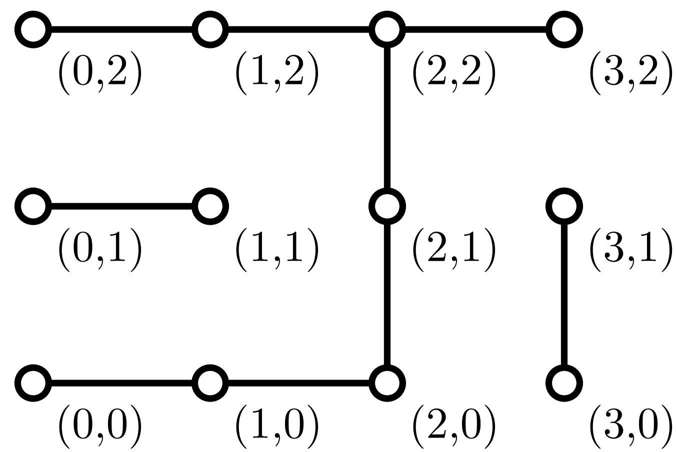
Define P, set of object *Classifications*.
For $L \in \mathcal{L}(n,m)$, $C(L) \in P$.
$\forall$ structures S, *and legal objects* L,
if* $C(L) = C(S) \Rightarrow L$ *is a structure.
***Define* $\mathcal{L}_X(n,m) = \{L \in \mathcal{L}(n,m),\ C(L) = X\}$.**

Important: Classification does *not* depend upon *value* of n.

In ST, classification of L is
partition of side nodes induced by components of L

Objects with same classification

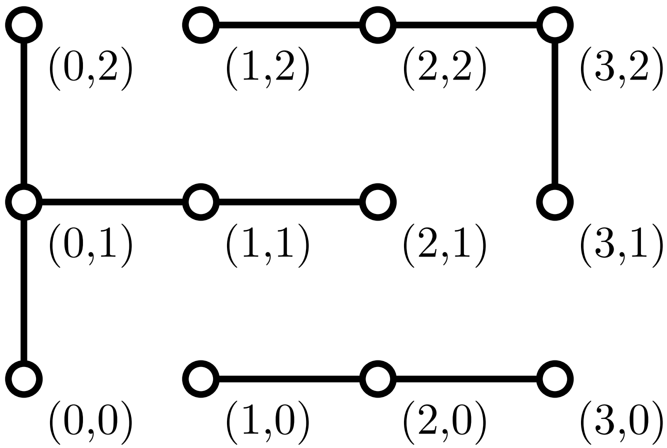


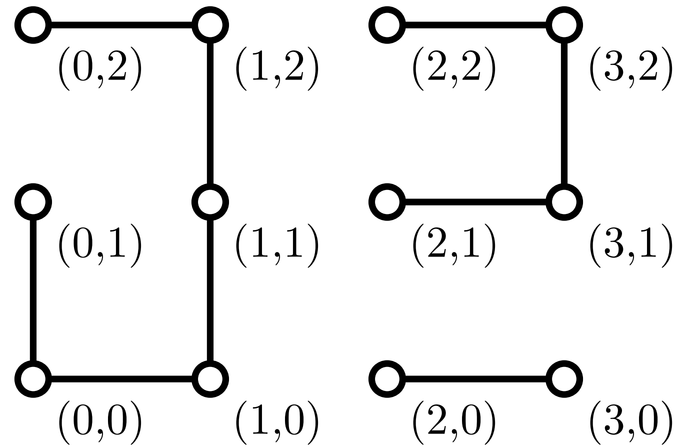


$$n = 4$$

$$X = \{\ \{(0,0),(0,2),(n-1,2)\}, \quad \{(0,1)\}, \quad \{(n-1,0),(n-1,1)\}$$

Objects with same classification





$$n = 4$$

$$X = \{\ \{(0,0),(0,1),(0,2)\}, \quad \{(n-1,0)\}, \quad \{(n-1,1),(n-1,2)\}$$

Another required property

Let $E \subseteq \mathtt{RtG}(n,m)$. **If** $L \in \mathcal{L}(n,m) \Rightarrow L \cup E \subseteq G(n+1,m)$

If $L_1, L_2 \in \mathcal{L}_X(n,m)$ **for some** $X \in P$ **then**

(i) either *both* $L_1 \cup E$ **and** $L_2 \cup E$ **are *not legal***

(ii) or *both are legal* and, for some $Y \in P$**,**

$$C(L_1 \cup E) = C(L_2 \cup E) = Y$$

We then write $X \cup E = Y$**.**

Another required property

Let $E \subseteq \mathtt{RtG}(n,m)$. **If** $L \in \mathcal{L}(n,m) \Rightarrow L \cup E \subseteq G(n+1,m)$

If $L_1, L_2 \in \mathcal{L}_X(n,m)$ **for some** $X \in P$ **then**

(i) either *both* $L_1 \cup E$ **and** $L_2 \cup E$ **are *not legal***

(ii) or *both are legal* and, for some $Y \in P$**,**

$$C(L_1 \cup E) = C(L_2 \cup E) = Y$$

We then write $X \cup E = Y$**.**

In spanning tree case if L_1, L_2 **are**

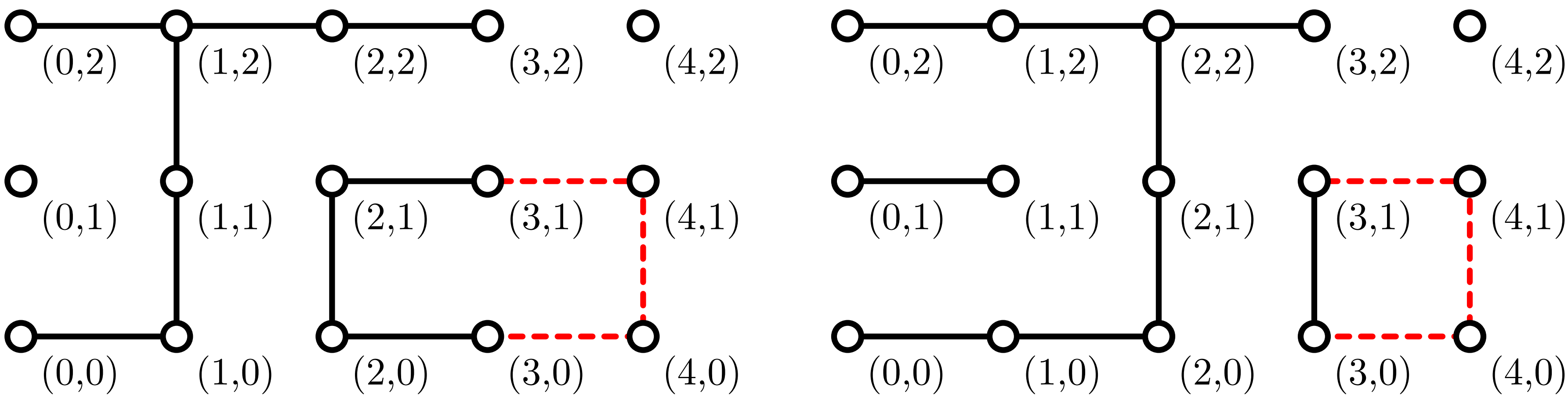
***legal forests* in** $G(n,m)$ **with same classification then**

(i) either *both* $L_1 \cup E$ **and** $L_2 \cup E$ **are *not forests* in** $G(n+1,m)$

(ii)or *both are legal forests* in $G(n+1,m)$ **and**

they have the same classification.

$X \cup E = \emptyset$

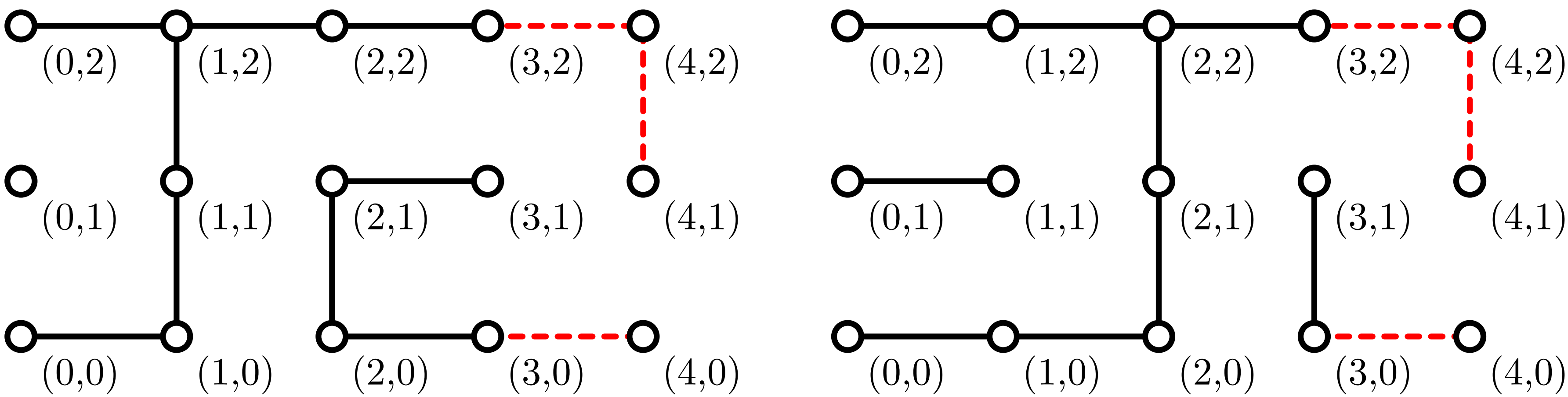


In $G(4,3)$,

$X = \{ \{(0,0),(0,2),(n-1,2)\}, \quad \{(0,1)\}, \quad \{(n-1,0),(n-1,1)\}$

In $G(5,3)$, $X \cup E$ is not legal.

$X \cup E \neq \emptyset$



In $G(4,3)$,

$X = \{\ \{(0,0),(0,2),(n-1,2)\}, \quad \{(0,1)\}, \quad \{(n-1,0),(n-1,1)\}$

In $G(5,3)$, $X \cup E =$

$X' = \{\ \{(0,0),(0,2),(n-1,1),(n-1,2)\}, \quad \{(0,1)\}, \quad \{(n-1,0)\}$

Final property

If L is legal in $\mathcal{L}(n+1,m)$ then
$L - \mathtt{RtG}(n,m)$ is legal in $\mathcal{L}(n,m)$

So,all legal objects in $G(n+1,m)$
can be built from legal objects in $G(n,m)$

Final property

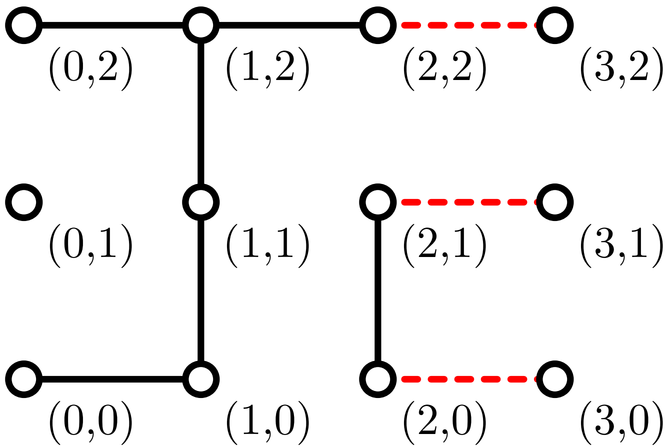
If L is legal in $\mathcal{L}(n+1,m)$ then
$L - \mathtt{RtG}(n,m)$ **is legal in** $\mathcal{L}(n,m)$

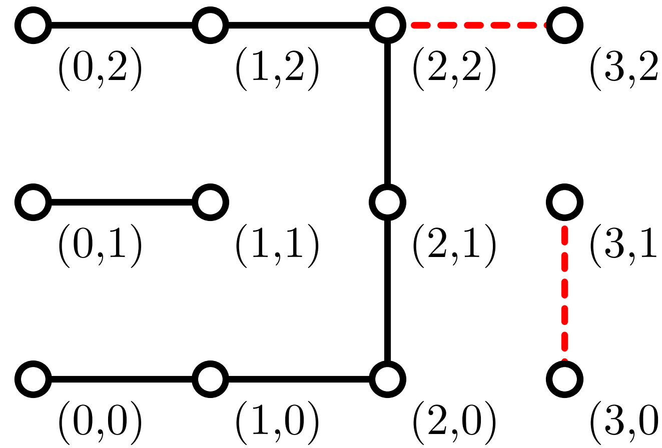
So,all legal objects in $G(n+1,m)$
can be built from legal objects in $G(n,m)$

For spanning trees this means that
if L is a forest in $G(n+1,m)$ in which
each component contains at least one side vertex
and the rightmost comb is removed

then what remains is a forest with the same property.

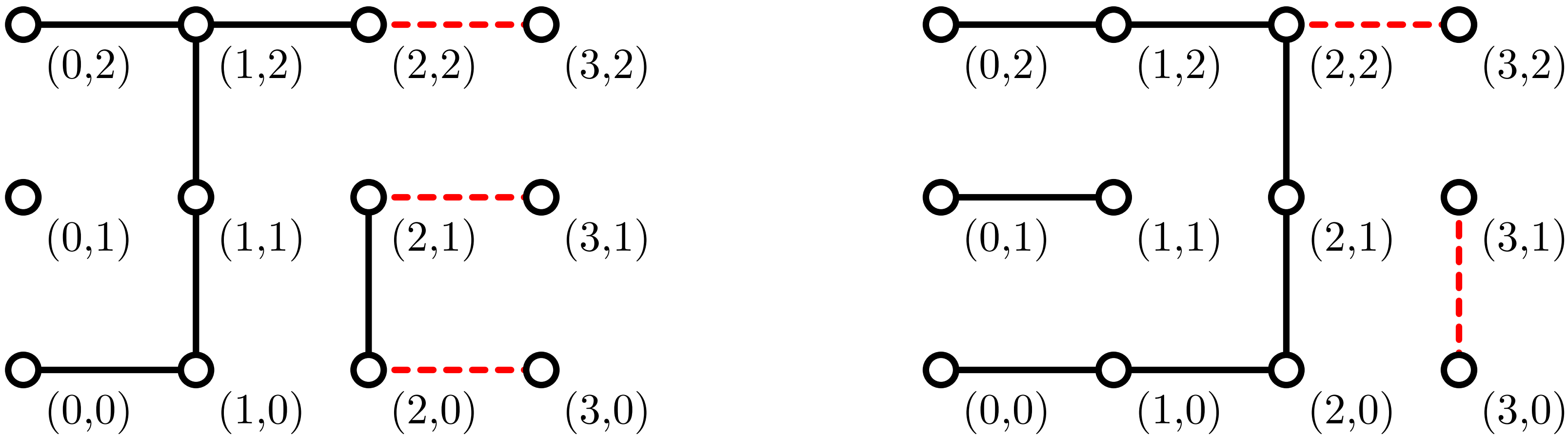
ST Example





In $G(4,3)$, $X =$
$\{\ \{(0,0),(0,2),(n-1,2)\},\ \{(0,1)\},\ \{(n-1,0),(n-1,1)\}\ \}$

ST Example



In $G(4,3)$, $X =$
$\{\ \{(0,0),(0,2),(n-1,2)\},\ \{(0,1)\},\ \{(n-1,0),(n-1,1)\}\ \}$

In $G(3,3)$ (left), $X =$
$\{\ \{(0,0),(0,2),(n-1,2)\},\ \{(0,1)\},\ \{(n-1,0),(n-1,1)\}\ \}$

In $G(3,3)$ (right),

$X = \{\ \{(0,0),(0,2),(n-1,0),(n-1,1),(n-1,2)\},\ \{(0,1)\}\}$

Calculating $f(n)$ in Grid Graphs

- $f_X(n) = |\mathcal{L}_X(m,n)|$. $\vec{f}(n) = (f_X(n))_{X \in P}$
- Let $a_{Y,X} = |\{E \subseteq \mathtt{RtG}(n,m) : X \cup E = Y\}|$.
  $A = (a_{Y,X})_{Y,X \in P}$ is the **Transfer Matrix**
- Then $\vec{f}(n+1) = A\vec{f}(n)$, so $\vec{f}(n) = A^{n-2}\vec{f}(2)$
- Recall L is a structure iff $C(L)$ is good.
  Then $f(n) = \sum_{X \text{good}} f_X(n)$.
- Set $b_X = 1$ if X good, $b_X = 0$ otherwise. $\vec{b} = (b_X)_{X \in P}$
- Then $f(n) = \vec{b}^t A^{n-2} f(2)$ and
  $f(n)$ satisfies RR; At worst, minimal polynomial of A.

Calculating $f(n)$ in FC

For FC, need 2 more properties (in addition to grid ones).

If L is a good structure in $FC(n,m)$
$L - \texttt{Side}(n,m)$ **is a legal object in** $G(n,m)$.

Calculating $f(n)$ in FC

For FC, need 2 more properties (in addition to grid ones).

If L is a good structure in $FC(n,m)$
$L - \texttt{Side}(n,m)$ is a legal object in $G(n,m)$.

In ST this just means that *deleting the edges hooking left & right from a spanning tree in $FC(n,m)$* leaves a legal object in $G(n,m)$.

Calculating $f(n)$ in FC

For FC, need 2 more properties (in addition to grid ones).

If L is a good structure in $FC(n,m)$
$L - \mathtt{Side}(n,m)$ **is a legal object in** $G(n,m)$.

In ST this just means that *deleting the edges hooking left & right from a spanning tree in* $FC(n,m)$ leaves a legal object in $G(n,m)$.

Let $E \subseteq \mathtt{Side}(n,m)$ and $X \in \mathcal{P}$. $\forall L \in \mathcal{L}_{\mathcal{X}}(n,m)$ **either**
(i) all $L \cup E$ are good structures in $FC(n,m)$, ($X \cup E$ good)
(ii) or *no* $L \cup E$ is a good structures in $FC(n,m)$

Calculating $f(n)$ in FC

For FC, need 2 more properties (in addition to grid ones).

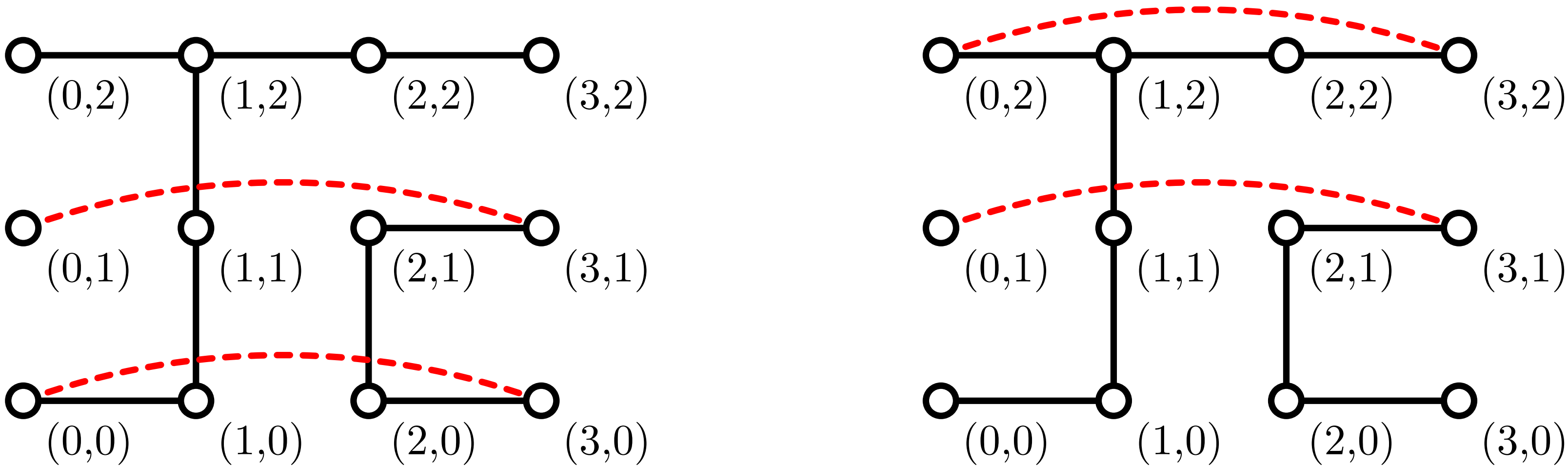
If L is a good structure in $FC(n,m)$
$L - \texttt{Side}(n,m)$ **is a legal object in** $G(n,m)$.

In ST this just means that *deleting the edges hooking left & right from a spanning tree in* $FC(n,m)$ leaves a legal object in $G(n,m)$.

Let $E \subseteq \texttt{Side}(n,m)$ and $X \in \mathcal{P}$. $\forall L \in \mathcal{L}_{\mathcal{X}}(n,m)$ **either**
(i) all $L \cup E$ are good structures in $FC(n,m)$, ($X \cup E$ good)
(ii) or *no* $L \cup E$ is a good structures in $FC(n,m)$

In Spanning tree case this just says that if L_1, L_2 legal forests in $G(n,m)$ with $C(L_1) = C(L_2)$, and $E \in \texttt{Side}(n,m)$ then either (i) both $L_1 \cup E$ and $L_2 \cup E$ are ST in $FC(n,m)$ or (ii) neither are

Example: adding different E



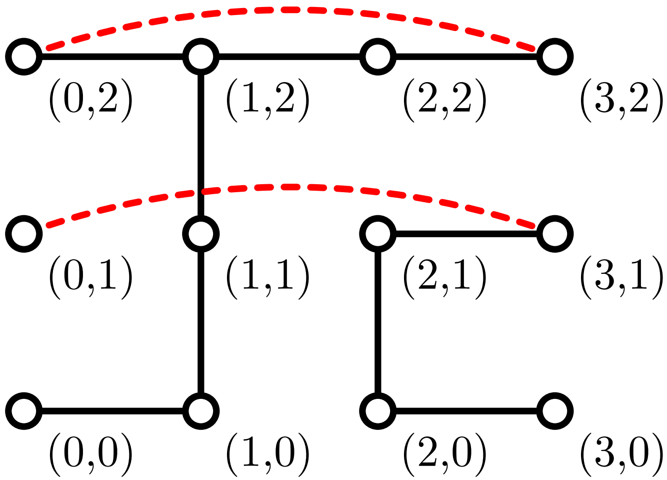
$X =$

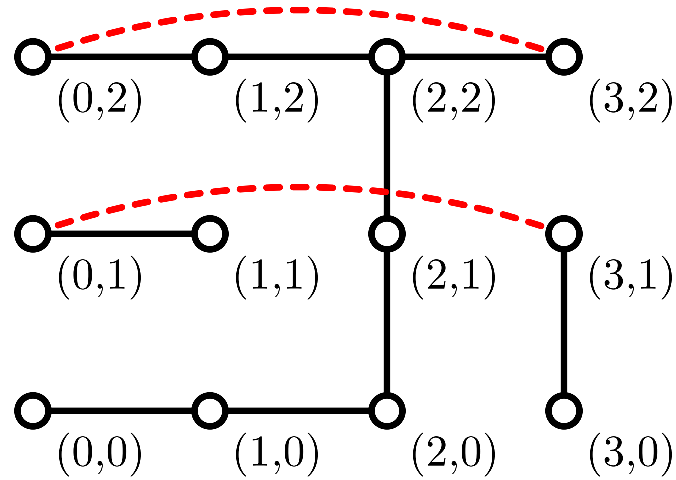
$\{ \ \{(0,0),(0,2),(n-1,2)\}, \ \{(0,1)\}, \ \{(n-1,0),(n-1,1)\} \ \}$

$X \cup \{ \ ((0,0),(3,0)), \ ((0,1),(3,1)) \ \}$ is a spanning tree.

$X \cup \{ \ ((0,1),(3,1)), \ ((0,2),(3,2)) \ \}$ is **NOT** a spanning tree.

Adding E to 2 objects with same X





$X =$
$\{\ \{(0,0),(0,2),(n-1,2)\},\ \{(0,1)\},\ \{(n-1,0),(n-1,1)\}\ \}$

$X \cup \{\ ((0,1),(3,1)),\ ((0,2),(3,2))\ \}$ is not a spanning tree.

Calculating $f(n)$ in FC (cont)

- Previously defined $f_X(n) = |\mathcal{L}_X(m,n)|$ and
  $\vec{f}(n) = (f_X(n))_{X \in P}$
- Already saw $\vec{f}(n) = A^{n-2}\vec{f}(2)$ and, when counting ST in *Grid Graphs*, $f(n) = \vec{b}^t A^{n-2}\vec{f}(2)$
- Define $c_X = |\{E \subseteq \texttt{Side}(n,m) : X \cup E$ good$\}|$
  $\vec{c} = (c_X)_{X \in P}$
- Then, when counting ST in Fat Cylinders,
  $f(n) = \sum_{X \in P} c_X f_X(n)$ or $f(n) = \vec{c}^t A^{n-2}\vec{f}(2)$

The Punchlines (new)

All structures in survey satisfy given abstract properties.
Therefore, general technique can be used for all of them.
In particular,

Grids and FC always share same transfer matrix.

Note: Simple observation but doesn't seem to have been explicitly noted before. Probably because, when concentrating specifically on grid graph,, there is a different natural transfer matrix

For all structures can also show that

1st properties hold for TC and 2nd properties for tori.

Thus, counting in TC and Tori can be done using the same transfer matrix (different from the G/FC one).

Outline

1. Introduction: Graphs & Problems
2. Survey of Results
3. The Transfer Matrix Technique
4. **Related Work & Open Problems**

Comparing $f(n)$ for different graphs

Have seen that $f(n) = \vec{a}^t A^n \vec{b}$; $\quad \Rightarrow f(n)$ satisfies RR
$\vec{a}$, A and b depend upon structure and graph
A, b are the same for Grids/FC and also for TC/Tori.

How do $f(n)$ of Grids & TC compare? Of TC & Tori?
How do RR of Grids & TC compare? Of TC & Tori?
Asymptotics of $f(n)$ (RR)?

Comparing $f(n)$ for different graphs

Have seen that $f(n) = \vec{a}^t A^n \vec{b}$; $\quad \Rightarrow f(n)$ satisfies RR
$\vec{a}$, A and b depend upon structure and graph
A, b are the same for Grids/FC and also for TC/Tori.

How do $f(n)$ of Grids & TC compare? Of TC & Tori?
How do RR of Grids & TC compare? Of TC & Tori?
Asymptotics of $f(n)$ (RR)?

We calculated $f(n)$ for ST for all four graphs for $n \times 3$.
Saw that RR for Grids "divided" RR for FC.
RR for TC "divided" RR for Tori.
Also, for appropriate ϕ_i, c_i, c_i', $i = 1, 2$.

$$\begin{aligned} \mathbf{ST}_G(n,3) &\sim c_1\phi_1^n, & \mathbf{ST}_{FC}(n,3) &\sim c_1' n\phi_1^n, \\ \mathbf{ST}_{TC}(n,3) &\sim c_2\phi_2^n, & \mathbf{ST}_T(n,3) &\sim c_2' n\phi_2^n \end{aligned}$$

Circulant Graphs and Tori

The **Circulant Graph**, $C_n^{s_1,s_2,\ldots,s_k}$ has

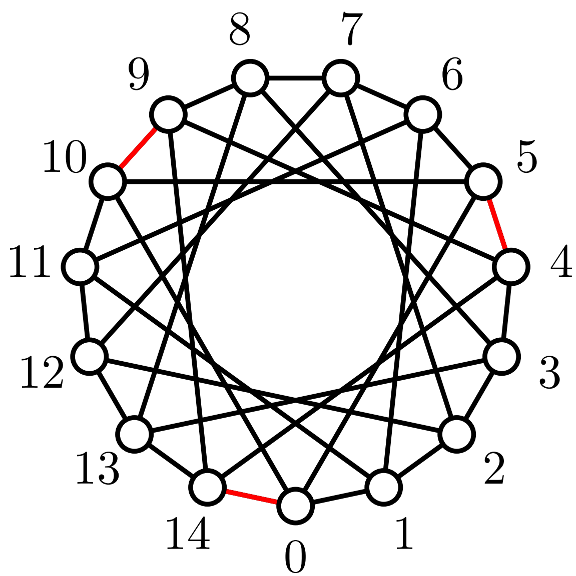
$$V = \{0,\ldots,n-1\}, \quad E = \{(i,j) : |i-j| \in \{s_1, s_2, \ldots, s_k\}\}$$

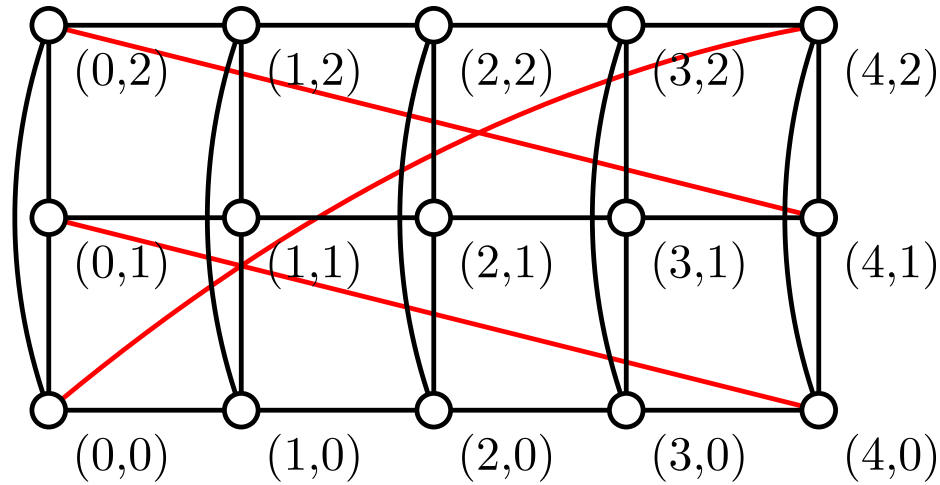
The s_i are **jumps** on the circle.

Circulant Graph $C_{mn}^{1,n}$ is *identical* to torus
except for side edges
all of whose left endpoints are shifted up by one.

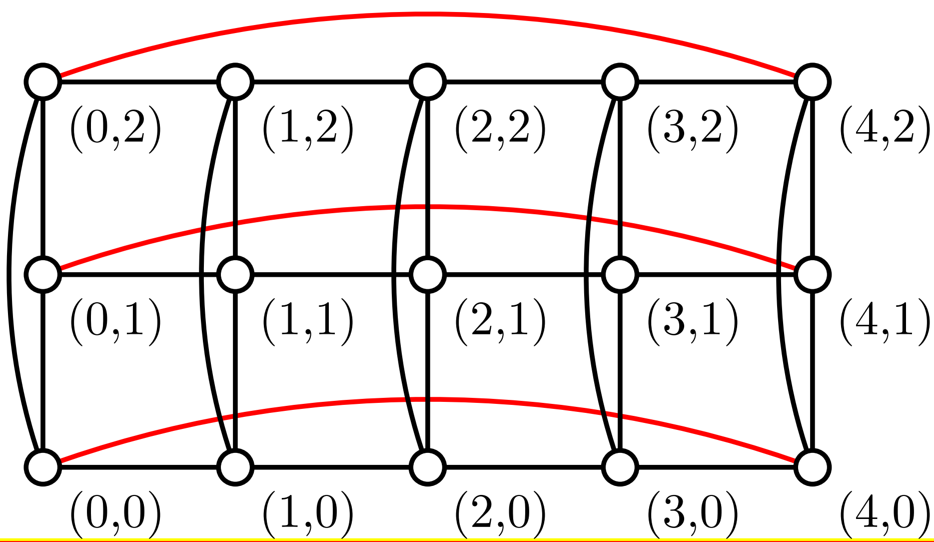
Is it possible to extend Lieb's result exactly calculating the "entropy" of the Eulerian Orientations of the torus to these circulant graphs?

Circulant Graphs and Tori





Circulant Graph $C_{3n}^{1,n}$ for $n = 5,$ both in circular form and lattice form.



Torus $T(5, 3)$

State Space/Eigenspace Reduction

The largest "practical" drawback to the technique is
large state space (*# of classifications*).
This tends to grow exponentially (or worse!) with m.

In general, very little can be done about this.
But, for many individual problems, reducing the state space can be a challenging (and fun) problem.

State Space/Eigenspace Reduction

The largest "practical" drawback to the technique is
large state space (*# of classifications*).
This tends to grow exponentially (or worse!) with m.

In general, very little can be done about this.
But, for many individual problems, reducing the state space can be a challenging (and fun) problem.

Example 1: For Hamiltonian Cycles on Grids, Stoyan & Strehl showed correspondence between reachable states in state space and Motzkin words (reducing state space)

State Space/Eigenspace Reduction

The largest “practical” drawback to the technique is
large state space (*# of classifications*).
This tends to grow exponentially (or worse!) with m.

In general, very little can be done about this.
But, for many individual problems, reducing the state space can be a challenging (and fun) problem.

Example 2: For many problems it is possible to show that transfer matrix has a special structure, reducing the size of the characteristic polynomial of the transfer matrix.

Strongly related to work done by Minc in the 80’s on calculating the permanent of circulant 0-1 matrices.

Outline

1. Introduction: Graphs & Problems
2. Survey of Results
3. The Transfer Matrix Technique
4. Related Work & Open Problems
5. **Questions?**