Faster Simulated Annealing Algorithms for Combinatorial Counting

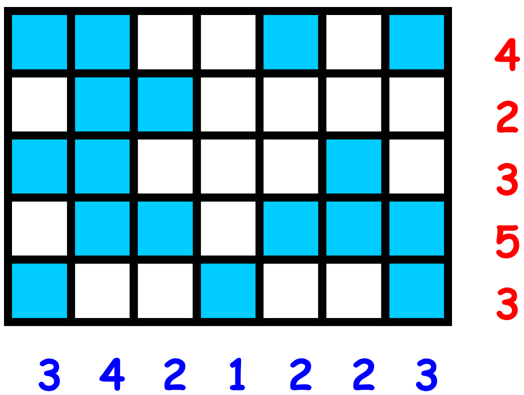
Ivona Bezáková

Rochester Institute of Technology

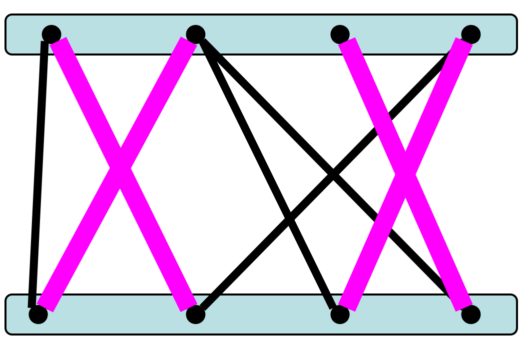
September 22, 2006

Problems

Binary contingency tables

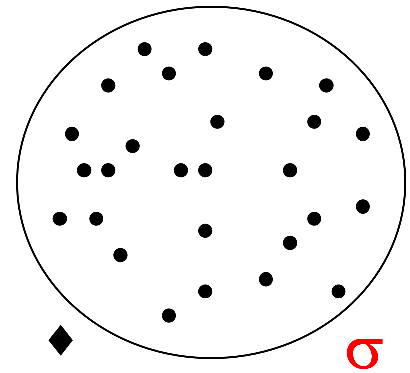


Permanent



Heuristics

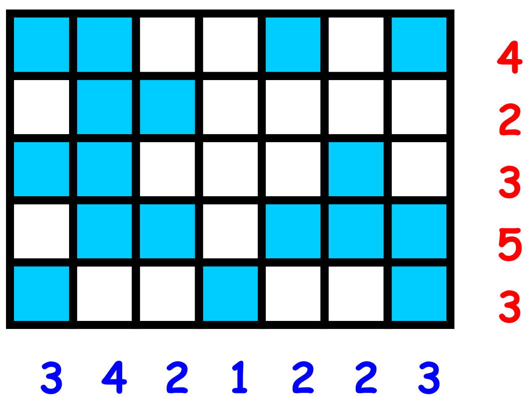
Importance sampling



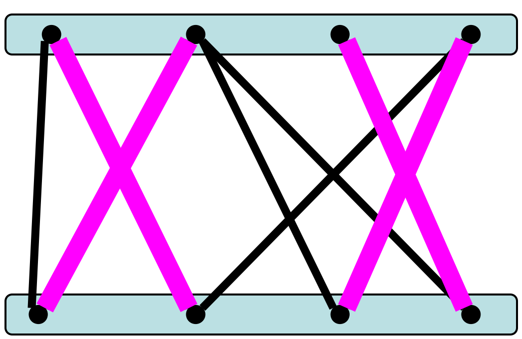
Simulated annealing

Problems

Binary contingency tables

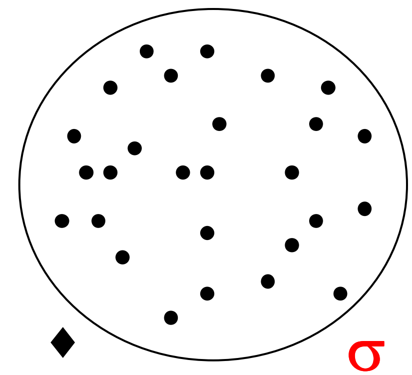


Permanent



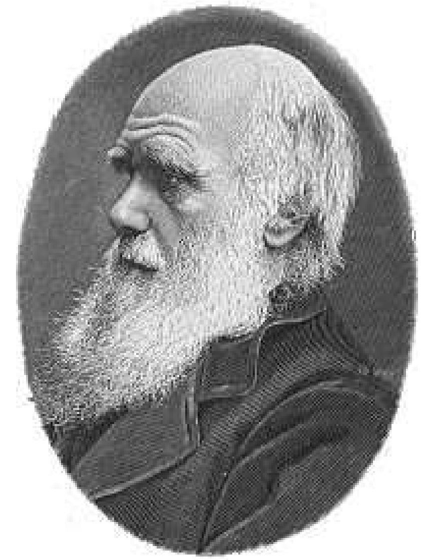
Heuristics

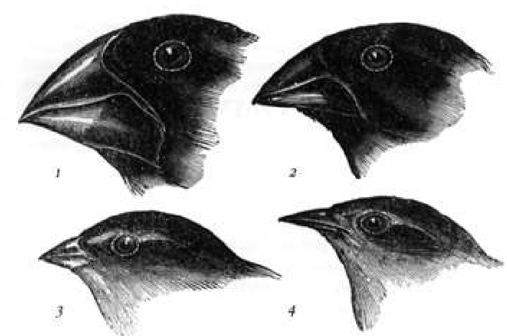
Importance sampling



Simulated annealing

Darwin's Finches





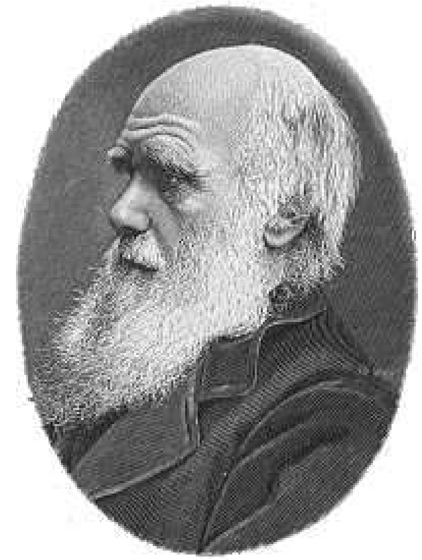
The Voyage of the Beagle

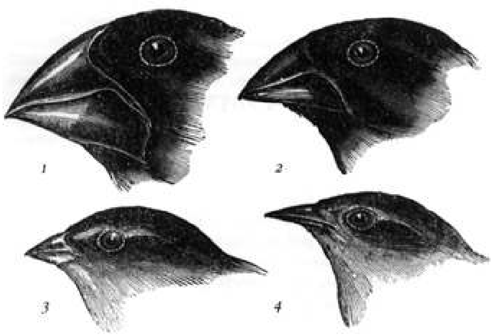
Galápagos archipelago (1835)

Darwin's Finches

Distribution of Darwin's Finches on Visitor Islands

	Santa Cruz	Plaza	Santa Fe	San Cristobal	Espanola	Floreana	Isabela	Fernandina	Santiago	Rabida
Small Ground Finch	●	●	●	●	●	●	●	●	●	●
Medium Ground Finch	●	●	●	●		●	●	●	●	●
Large Ground Finch	●						●	●	●	●
Cactus Ground Finch	●	●	●	●		●	●		●	●
Large Cactus Ground Finch					●					
Sharp-beaked Ground Finch								●	●	
Vegetarian Finch	●			●		●	●	●	●	●
Small Tree Finch	●		●	●		●	●	●	●	●
Medium Tree Finch						●				
Large Tree Finch	●					●	●	●	●	●
Woodpecker Finch	●			●			●		●	
Mangrove Finch							●	●		
Warbler Finch	●		●	●	●	●	●	●	●	●





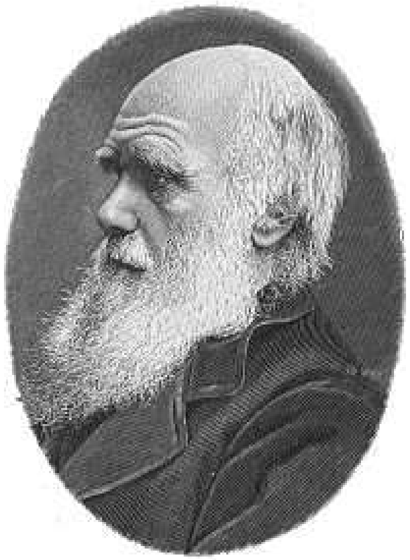
Darwin's Finches

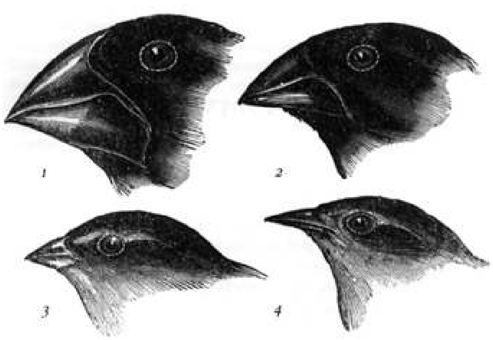
Distribution of Darwin's Finches on Visitor Islands

	Santa Cruz	Plaza	Santa Fe	San Cristobal	Espanola	Floreana	Isabela	Fernandina	Santiago	Rabida
Small Ground Finch	●	●	●	●	●	●	●	●	●	●
Medium Ground Finch	●	●	●	●		●	●	●	●	●
Large Ground Finch	●						●	●	●	●
Cactus Ground Finch	●	●	●	●		●	●		●	●
Large Cactus Ground Finch					●					
Sharp-beaked Ground Finch								●	●	
Vegetarian Finch	●			●		●	●	●	●	●
Small Tree Finch	●		●	●		●	●	●	●	●
Medium Tree Finch						●				
Large Tree Finch	●					●	●	●	●	●
Woodpecker Finch	●			●			●		●	
Mangrove Finch							●	●		
Warbler Finch	●		●	●	●	●	●	●	●	●

8

10

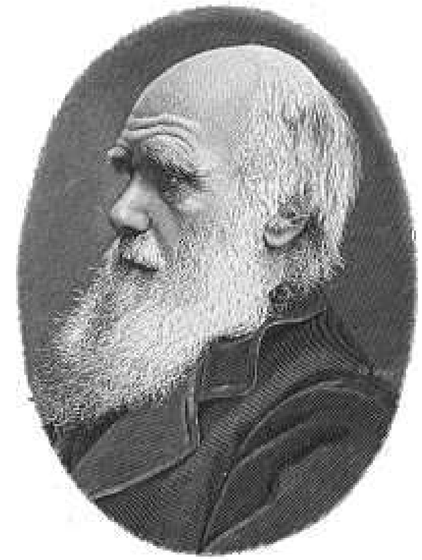




Darwin's Finches

Distribution of Darwin's Finches on Visitor Islands

	Santa Cruz	Plaza	Santa Fe	San Cristobal	Espanola	Floreana	Isabela	Fernandina	Santiago	Rabida	
Small Ground Finch	●	●	●	●	●	●	●	●	●	●	10
Medium Ground Finch	●	●	●	●		●	●	●	●	●	9
Large Ground Finch	●						●	●	●	●	6
Cactus Ground Finch	●	●	●	●		●	●		●	●	8
Large Cactus Ground Finch					●						2
Sharp-beaked Ground Finch								●	●		3
Vegetarian Finch	●			●		●	●	●	●	●	7
Small Tree Finch	●		●	●		●	●	●	●	●	8
Medium Tree Finch						●					1
Large Tree Finch	●					●	●	●	●	●	6
Woodpecker Finch	●			●			●		●		4
Mangrove Finch							●	●			2
Warbler Finch	●		●	●	●	●	●	●	●	●	10
	9	3	5	7	3	8	10	9	10	8	



chance

OR

competitive pressures

?

Binary Contingency Tables

Given: marginals (row sums, column sums)

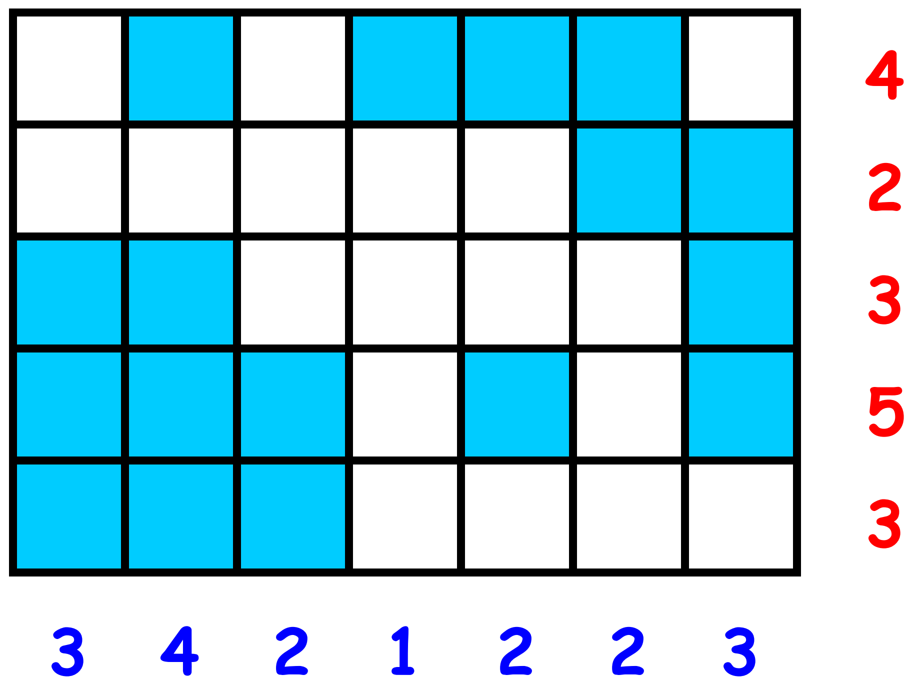
Goal:
- sample tables uniformly at random
- count tables

							4
							2
							3
							5
							3
3	4	2	1	2	2	3	

Binary Contingency Tables

Given: marginals (row sums, column sums)

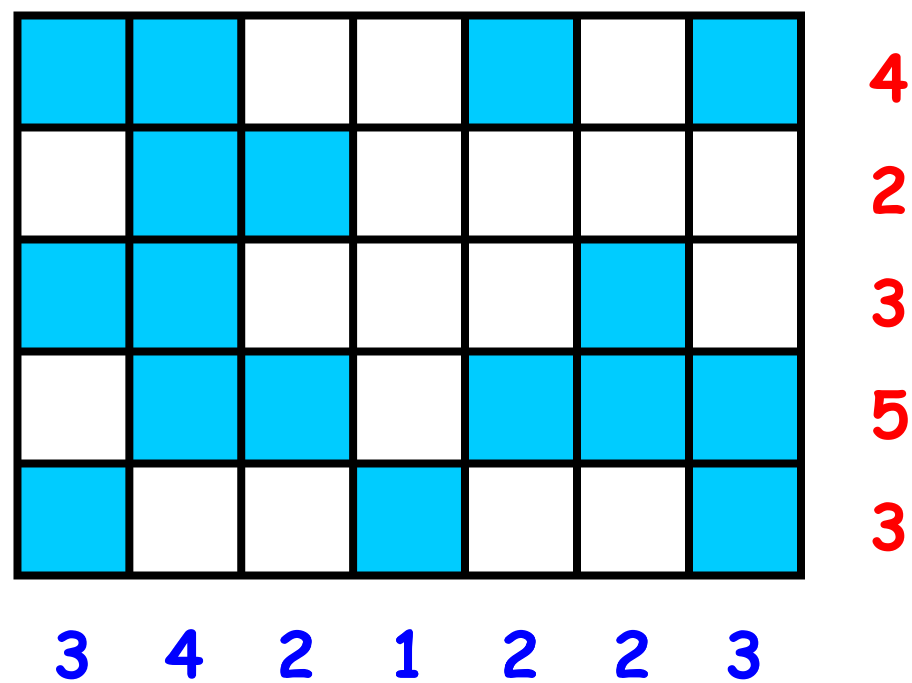
Goal:
- sample tables uniformly at random
- count tables



Binary Contingency Tables

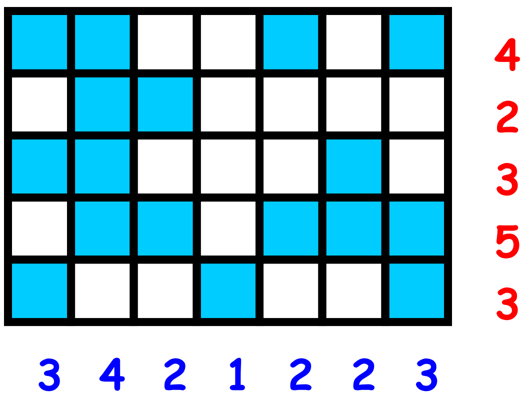
Given: marginals (row sums, column sums)

Goal:
- sample tables uniformly at random
- count tables

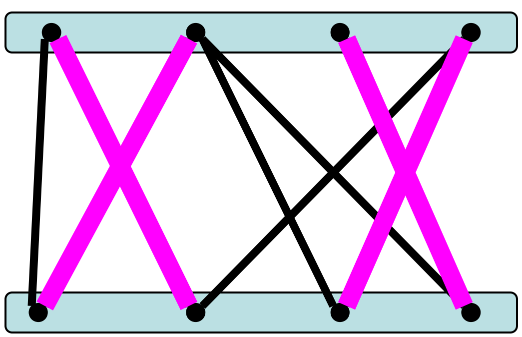


Problems

Binary
contingency tables

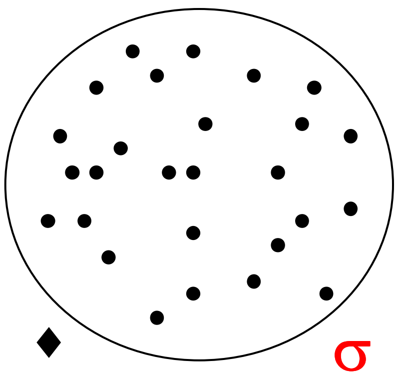


Permanent



Heuristics

Importance sampling



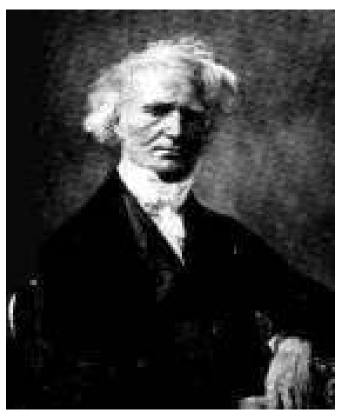
Simulated annealing

Permanent of an n x n matrix A

$$\mathrm{Per}(A) = \sum_{\pi \in S_n} \prod_{i=1}^{n} a_{i,\pi(i)}$$

in 1812:

Augustin-Louis Cauchy



Jacques Binet

A =

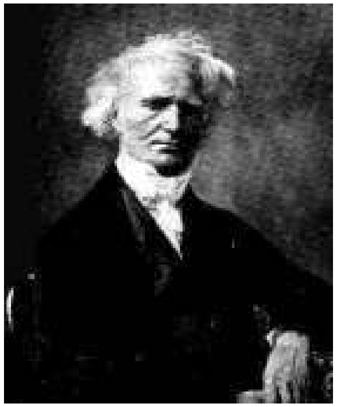
5	1	0	3	4
2	0	1	4	4
5	7	9	2	2
1	1	6	1	0
3	3	0	8	7

Permanent of an n x n matrix A

$$\mathrm{Per}(A) = \sum_{\pi \in S_n} \prod_{i=1}^{n} a_{i,\pi(i)}$$

in 1812:

Augustin-Louis
Cauchy



Jacques
Binet

$$A = \begin{bmatrix} 5 & 1 & 0 & 3 & 4 \\ 2 & 0 & 1 & 4 & 4 \\ 5 & 7 & 9 & 2 & 2 \\ 1 & 1 & 6 & 1 & 0 \\ 3 & 3 & 0 & 8 & 7 \end{bmatrix}$$

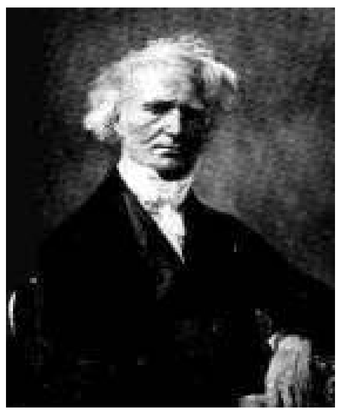
π

Permanent of an n x n matrix A

$$\mathrm{Per}(A) = \sum_{\pi \in S_n} \prod_{i=1}^{n} a_{i,\pi(i)}$$

in 1812:

Augustin-Louis Cauchy



Jacques Binet

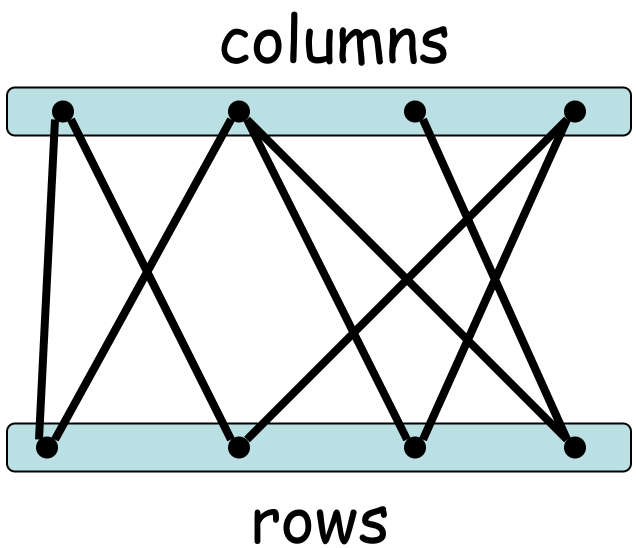
π'

A =

5	1	0	3	4
2	0	1	4	4
5	7	9	2	2
1	1	6	1	0
3	3	0	8	7

Permanent of an n x n matrix A

$$\mathrm{Per}(A) = \sum_{\pi \in S_n} \prod_{i=1}^{n} a_{i,\pi(i)}$$



columns

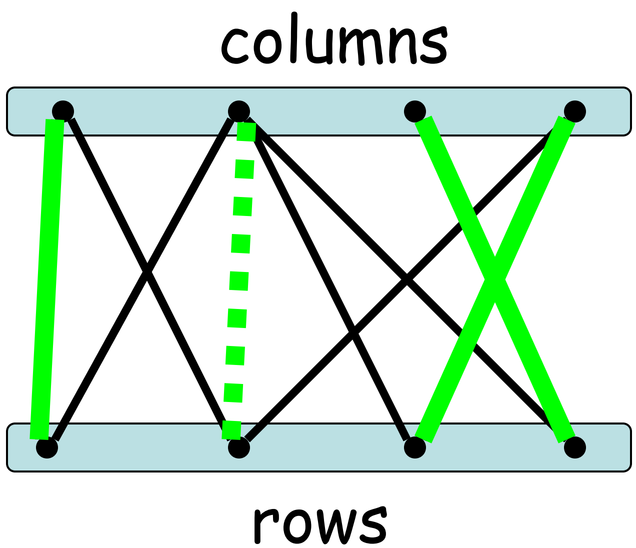
1	1	0	0
1	0	0	1
0	1	0	1
0	1	1	0

rows

Permanent of an $n \times n$ matrix A

0

$$\mathrm{Per}(A) = \sum_{\pi \in S_n} \prod_{i=1}^{n} a_{i,\pi(i)}$$



π

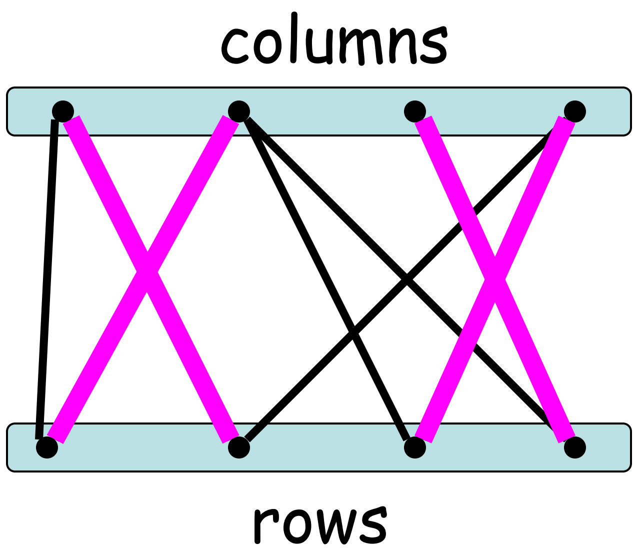
columns

1	1	0	0	rows
1	0	0	1	
0	1	0	1	
0	1	1	0	

Permanent of an n x n matrix A

1

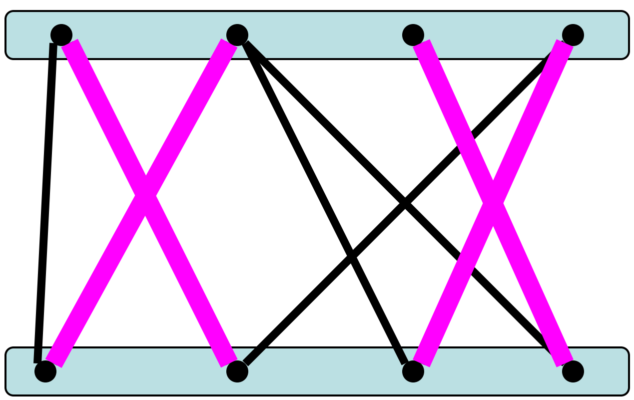
$$\mathrm{Per}(A) = \sum_{\pi \in S_n} \prod_{i=1}^{n} a_{i,\pi(i)}$$



$$\begin{array}{c}
\text{columns} \quad \pi \\
\begin{array}{|c|c|c|c|}
\hline
1 & 1 & 0 & 0 \\ \hline
1 & 0 & 0 & 1 \\ \hline
0 & 1 & 0 & 1 \\ \hline
0 & 1 & 1 & 0 \\ \hline
\end{array} \ \text{rows}
\end{array}$$

Permanent of an n x n matrix A

$$\mathrm{Per}(A) = \sum_{\pi \in S_n} \prod_{i=1}^{n} a_{i,\pi(i)}$$

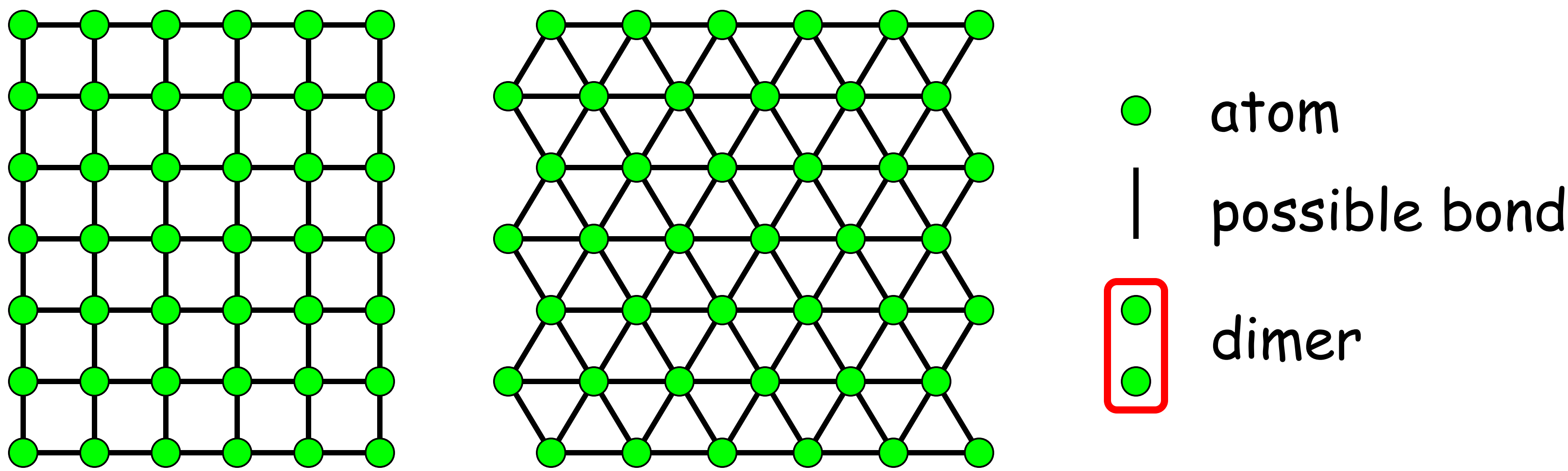


Count perfect matchings of a bipartite graph

Permanent: Counting Perfect Matchings

[Kasteleyn '63]   **planar graphs** in polytime

Dimer model



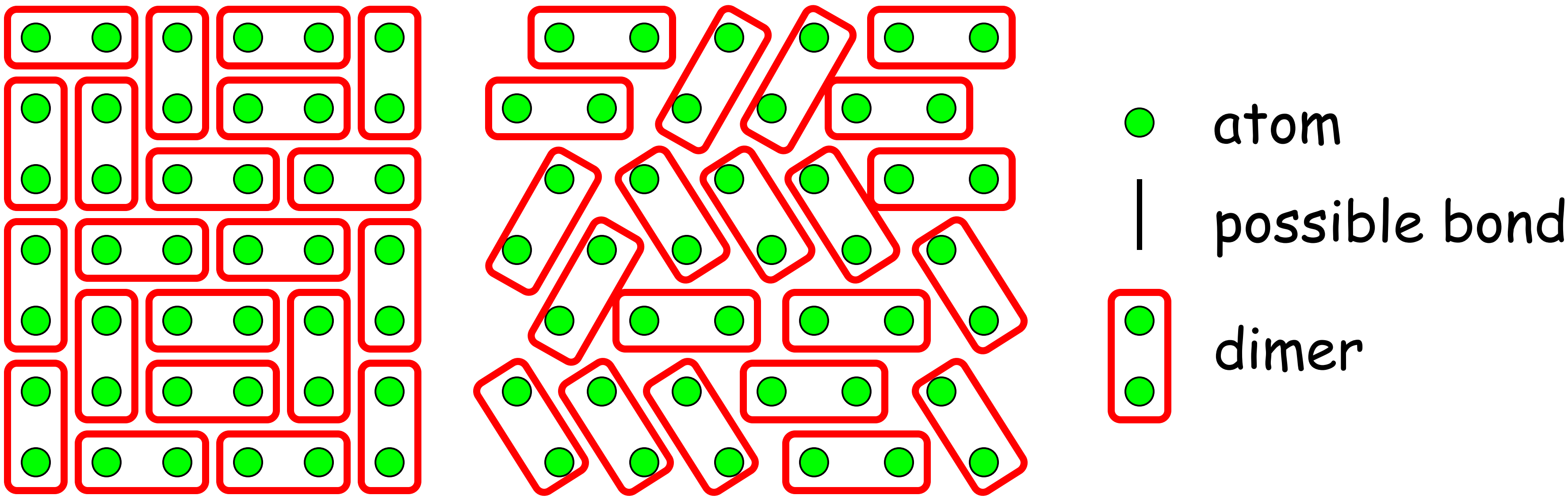
Partition function = permanent (in bipartite case)

- for computation of thermodynamic properties

Permanent: Counting Perfect Matchings

[Kasteleyn '63] **planar graphs** in polytime

Dimer model



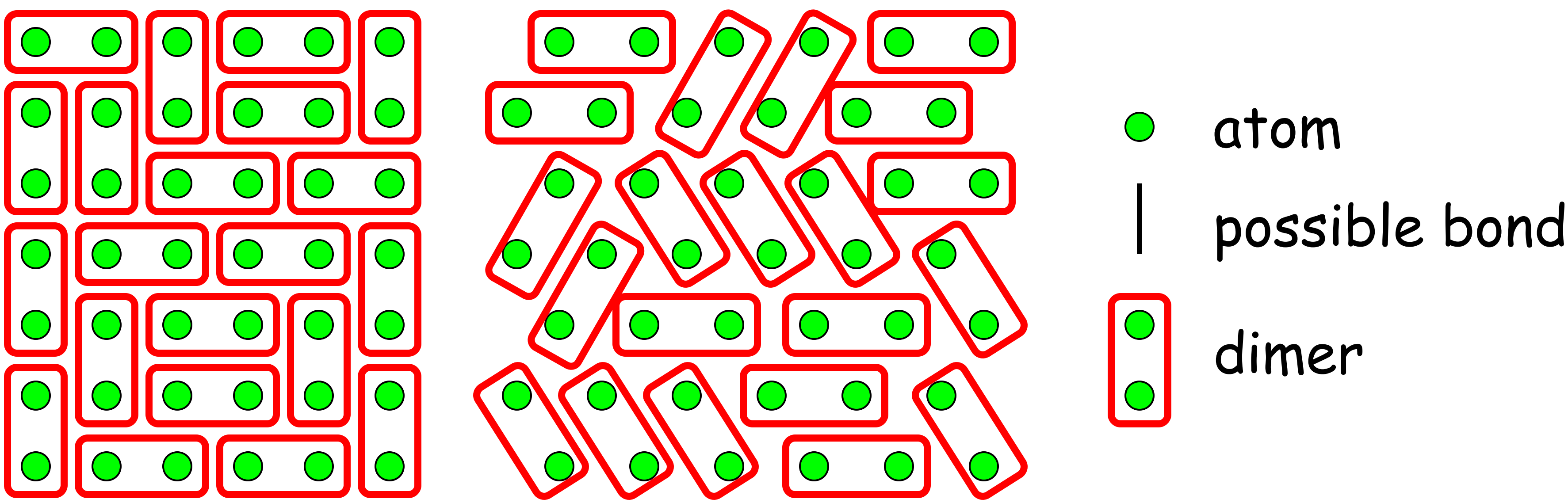
Partition function = permanent (in bipartite case)

- for computation of thermodynamic properties

Permanent: Counting Perfect Matchings

[Kasteleyn '63] **planar graphs** in polytime

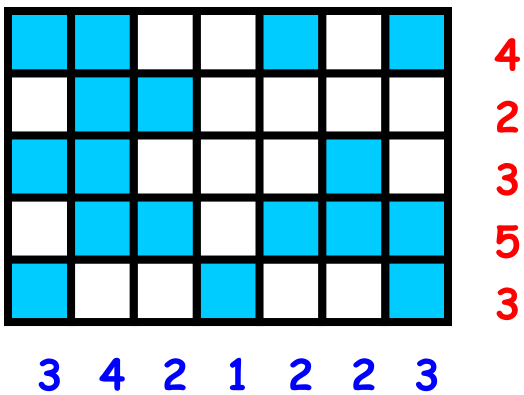
Dimer model



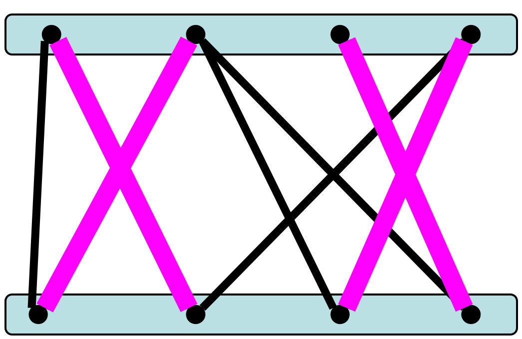
[Valiant '79] **#P-complete** for general (bipartite) graphs

Problems

Binary contingency tables

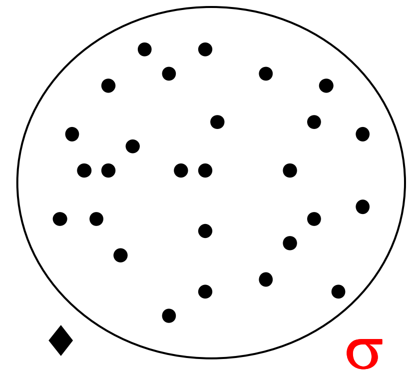


Permanent



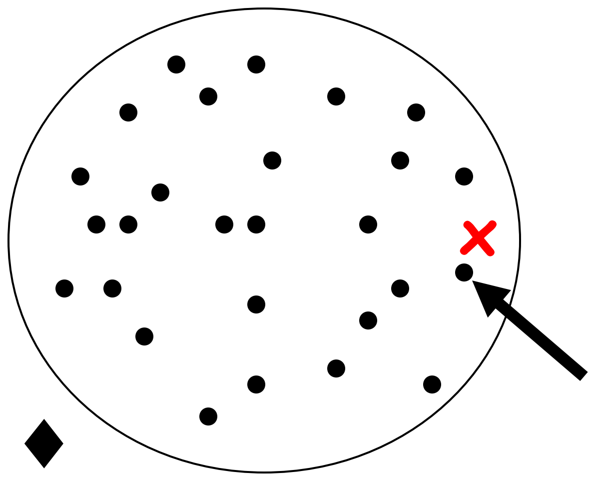
Heuristics

Importance sampling



Simulated annealing

Importance Sampling for counting problems



Probability distribution σ

on the points + ♦

with positive probability $\sigma(x)>0$

Random variable

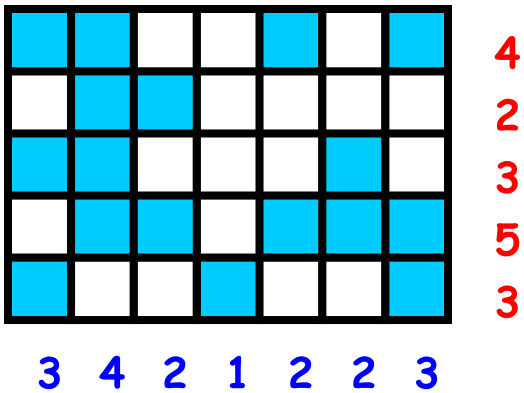
$$\eta(s) = \begin{cases} 1/\sigma(s) & \text{if s in the set} \\ 0 & \text{if s is ♦} \end{cases}$$

Unbiased estimator

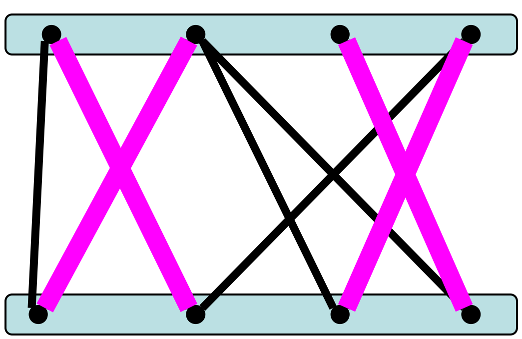
$$\boxed{E[\eta] = \sum \sigma(x).1/\sigma(x) = \text{size of the set}}$$

Problems

Binary contingency tables

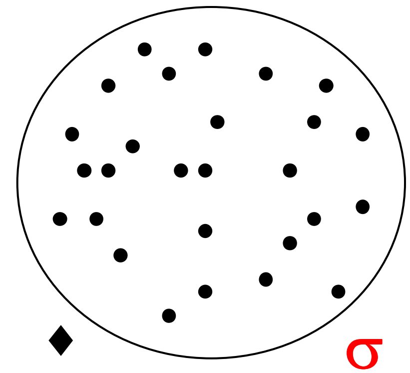


Permanent



Heuristics

Importance sampling



Simulated annealing

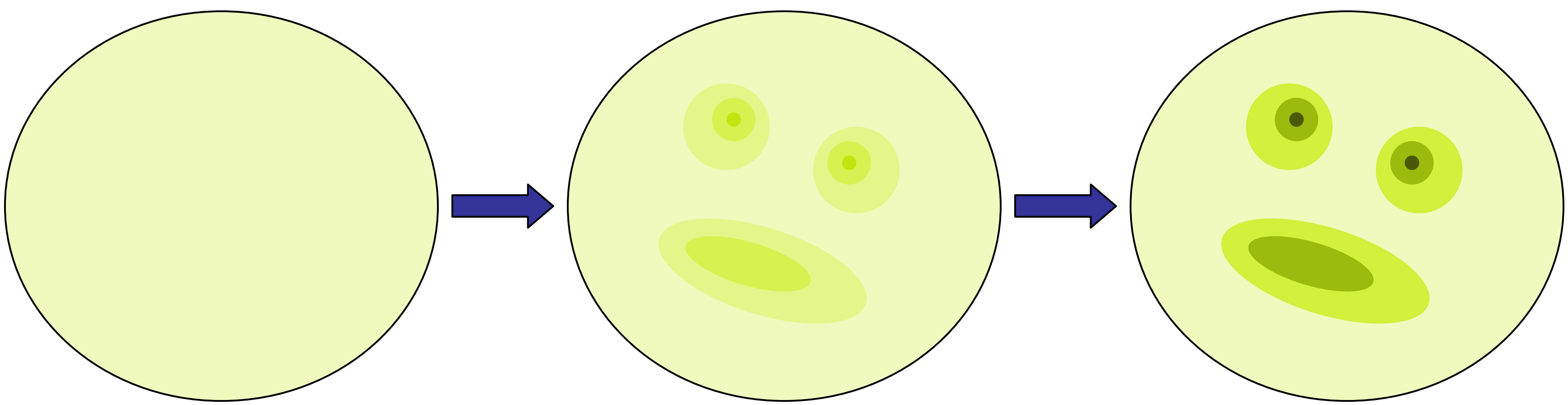
Simulated Annealing

[Kirkpatrick-Gelatt-Vecchi '83]
[Černý '85]

Unordered state → Ordered state

(EASY) (DIFFICULT)

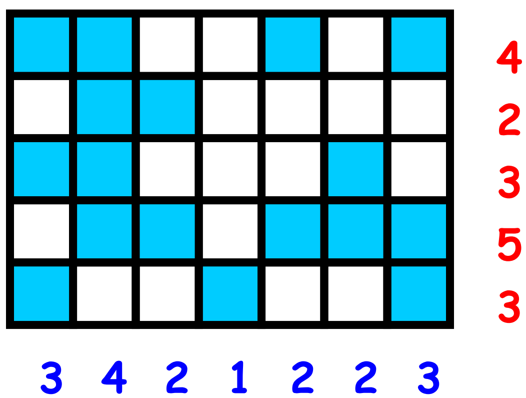
Cooling:



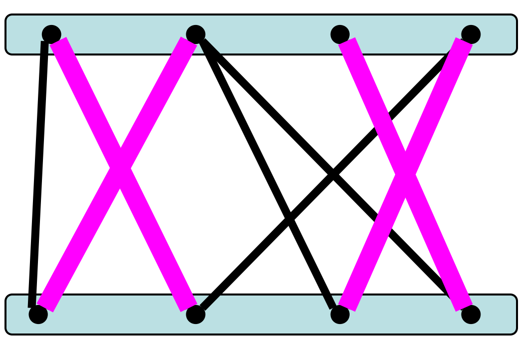
all equivalent differences more and more pronounced

Problems

Binary
contingency tables

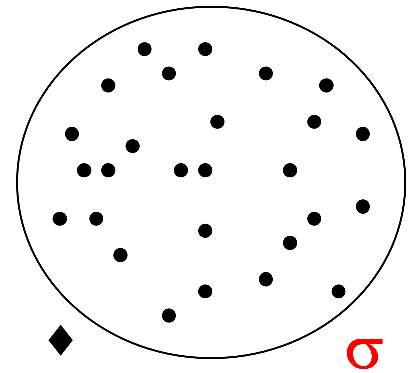


Permanent



Heuristics

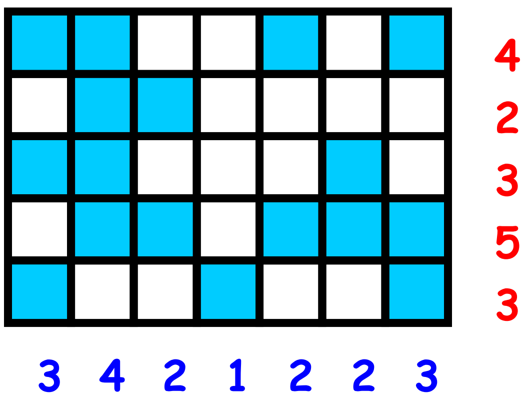
Importance sampling



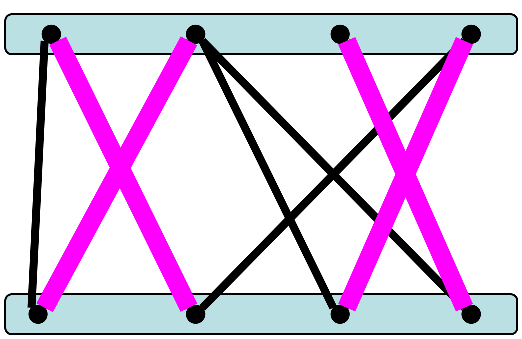
Simulated annealing

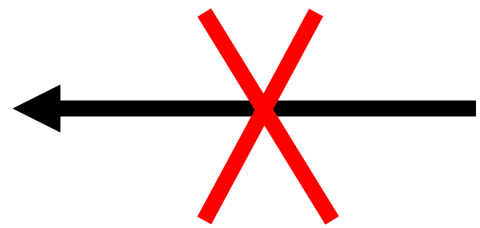
Problems

Binary contingency tables



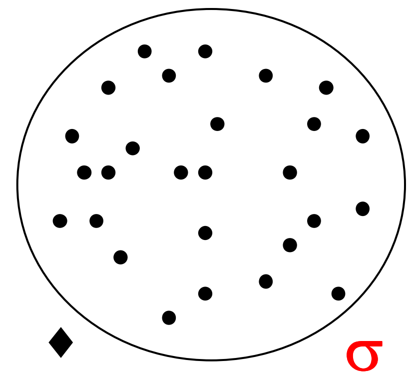
Permanent





Heuristics

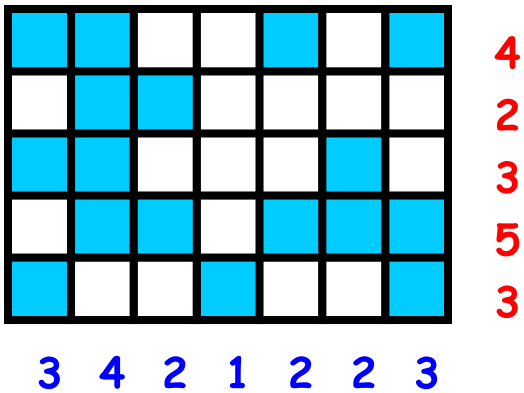
Importance sampling



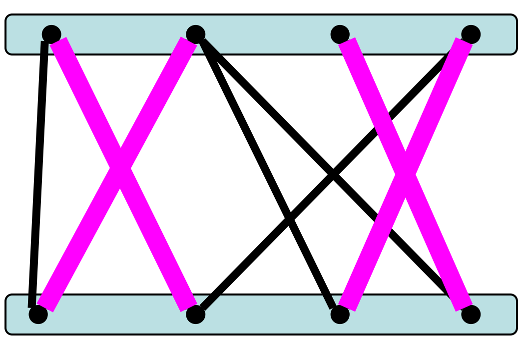
Simulated annealing

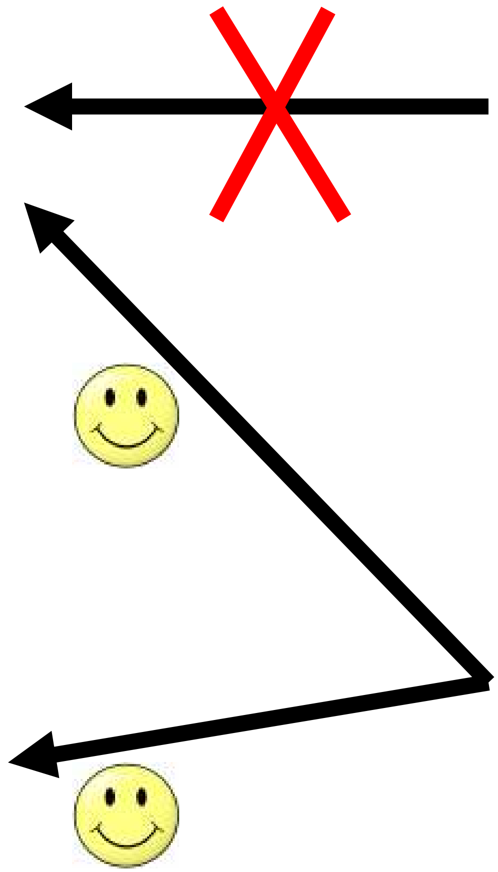
Problems

Binary contingency tables



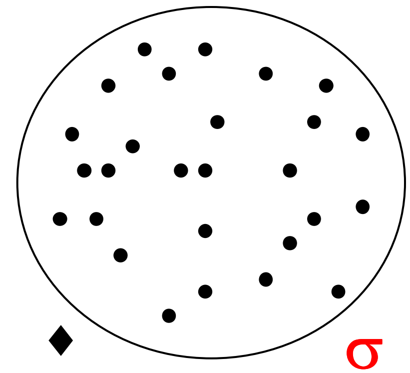
Permanent





Heuristics

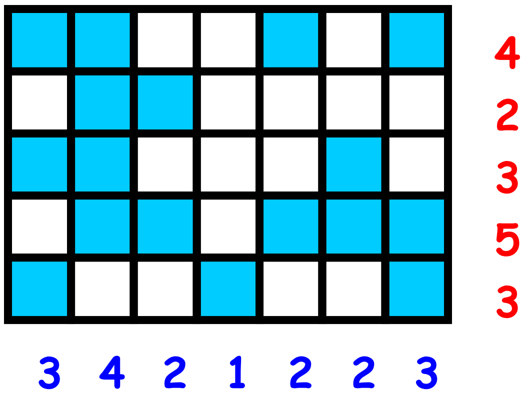
Importance sampling



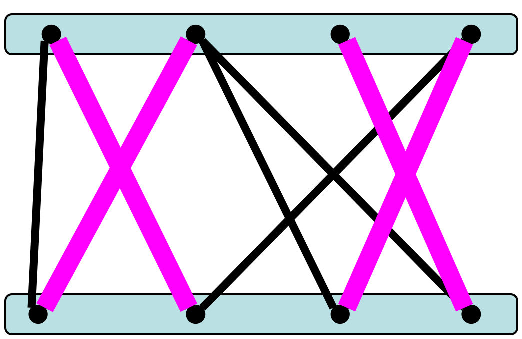
Simulated annealing

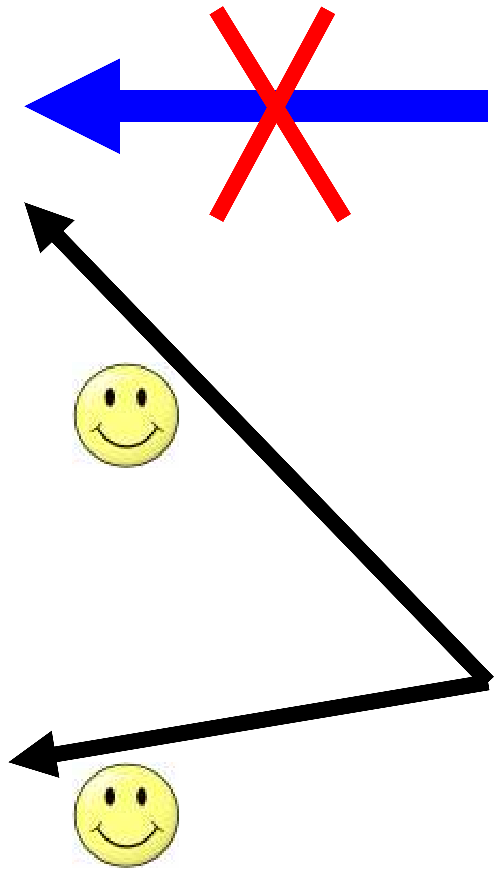
Problems

Binary
contingency tables



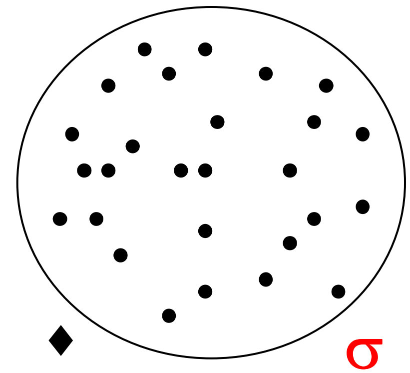
Permanent





Heuristics

Importance sampling



Simulated annealing

Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

a specific σ

- fill table column-by-column
- assign each column ignoring other column sums

							4
							2
							3
							5
							3
3	4	2	1	2	2	3	

Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

a specific σ

- fill table column-by-column
- assign each column ignoring other column sums

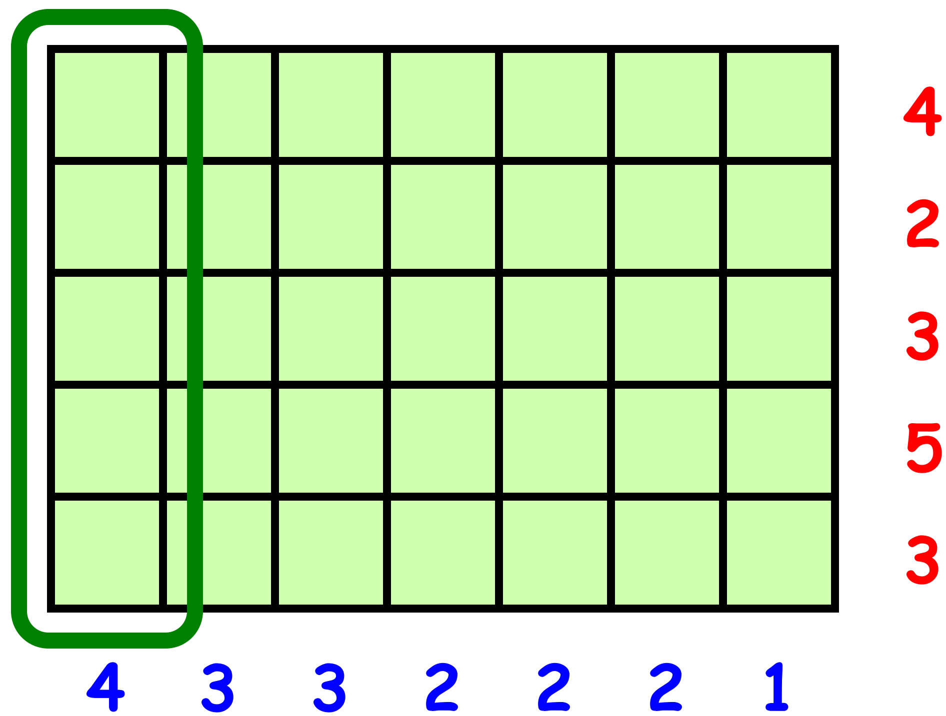
							4
							2
							3
							5
							3
4	3	3	2	2	2	1	

Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

a specific σ

- fill table column-by-column
- assign each column ignoring other column sums

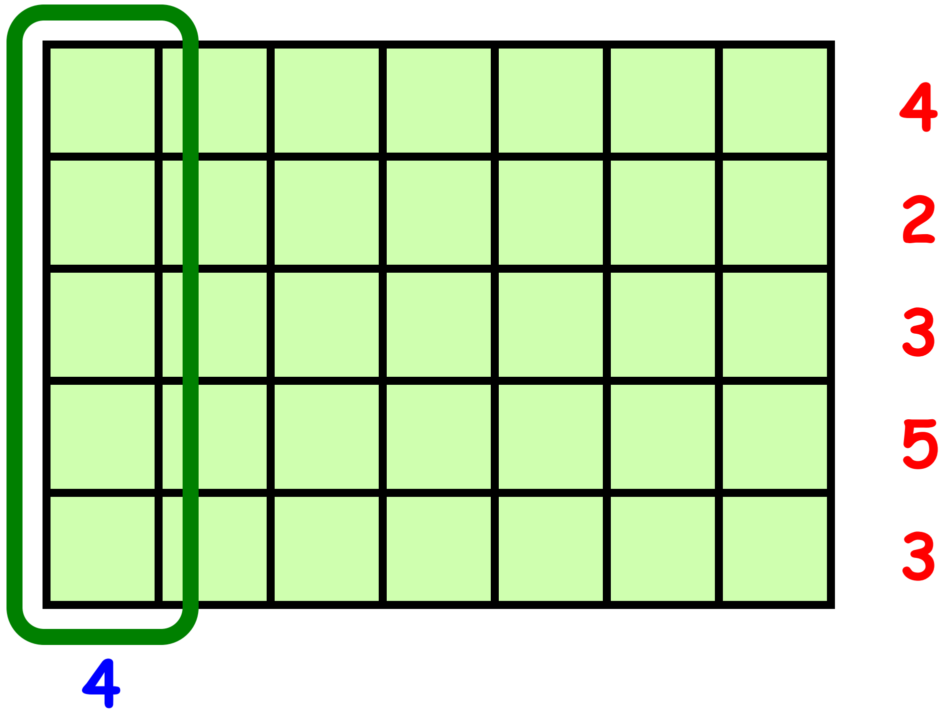


Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

a specific σ

- fill table column-by-column
- assign each column ignoring other column sums



Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

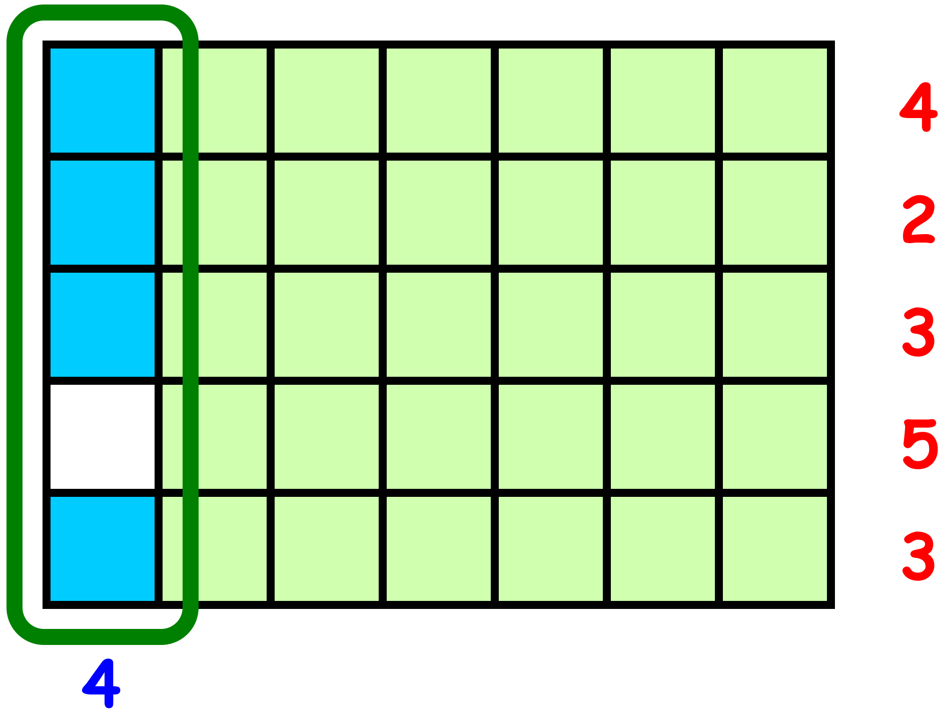
a specific σ

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

where product ranges over **i: rows with assignment 1**



Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

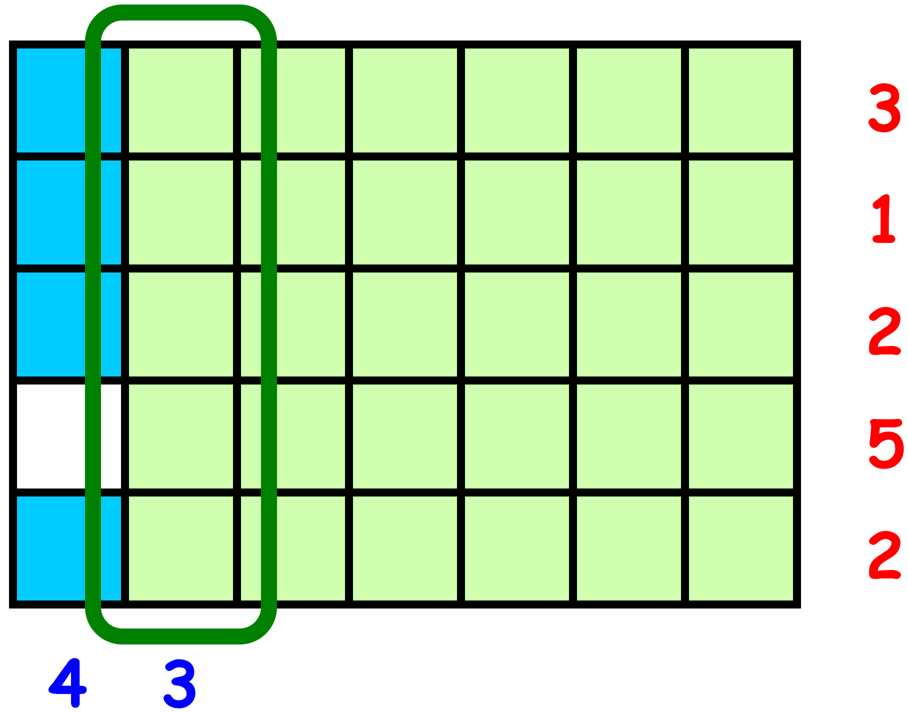
a specific σ

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

where product ranges over **i: rows with assignment 1**



Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

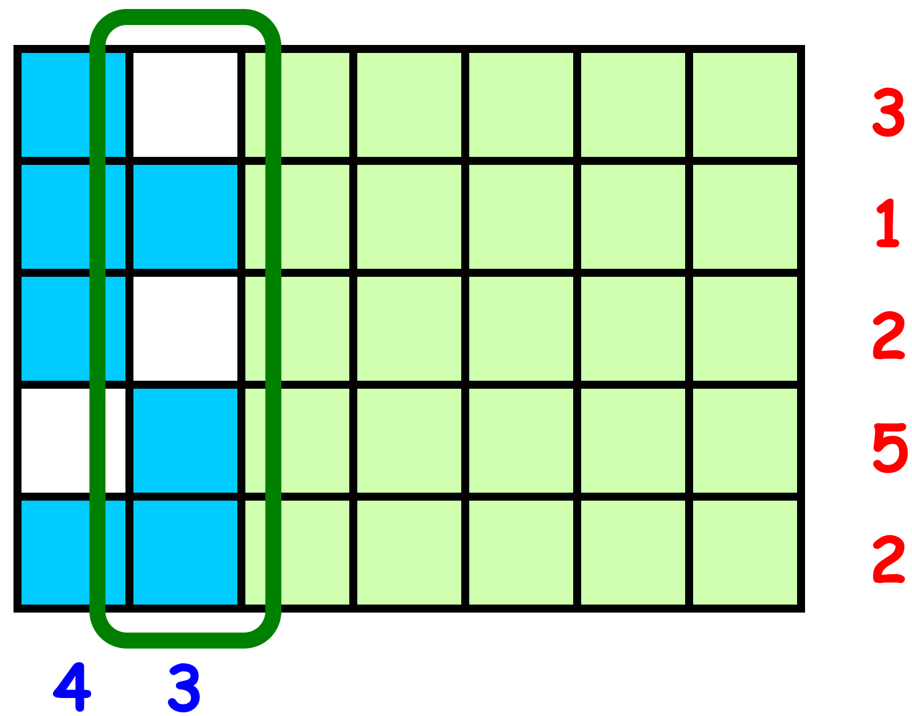
a specific σ

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

where product ranges over **i: rows with assignment 1**



Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

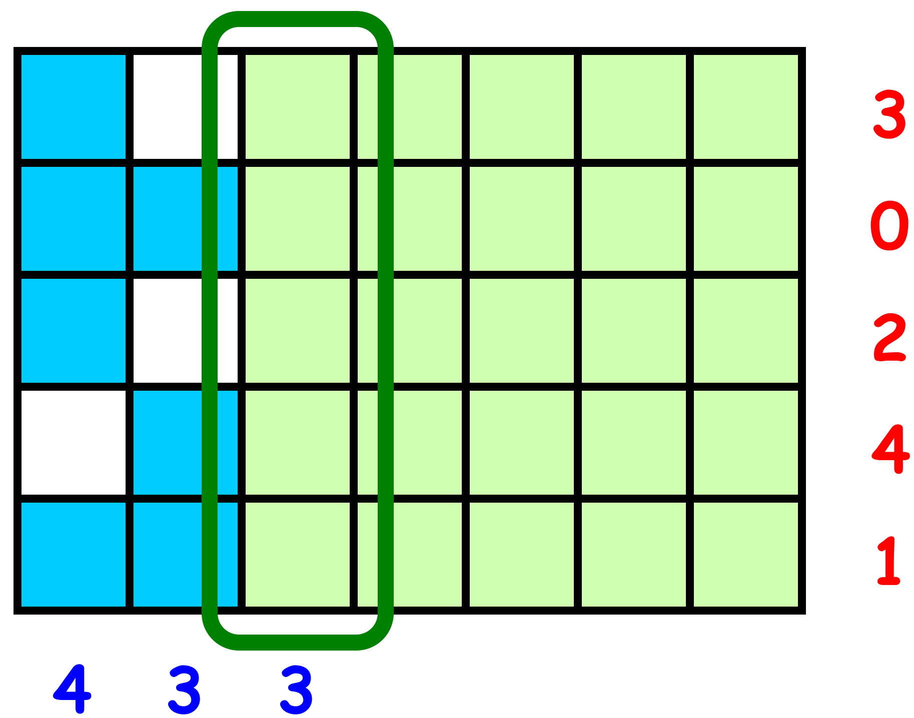
a specific σ

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

where product ranges over **i: rows with assignment 1**



Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

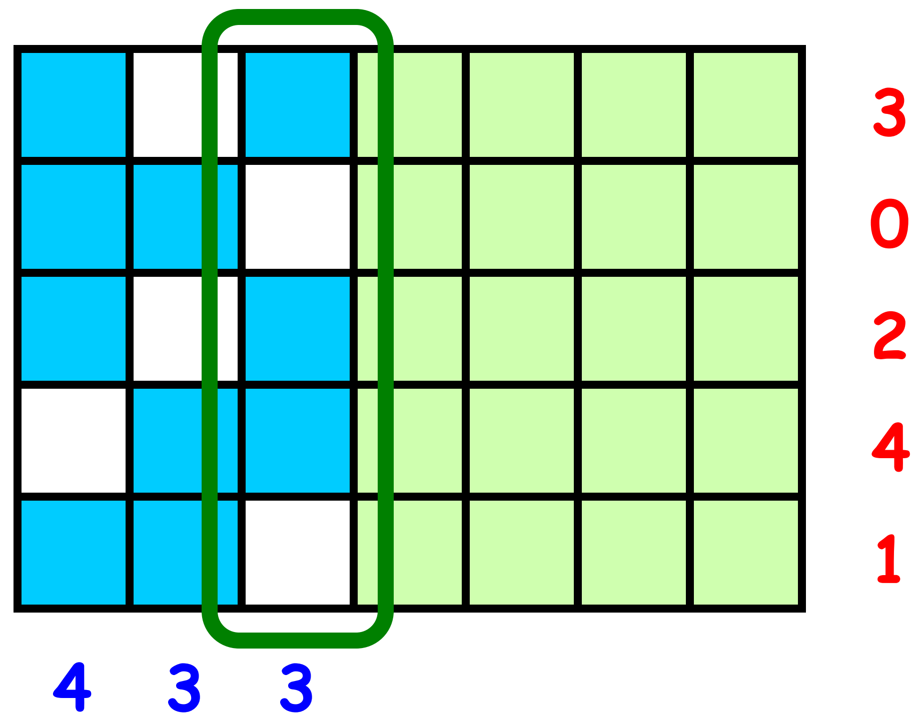
a specific σ

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

where product ranges over **i: rows with assignment 1**



Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

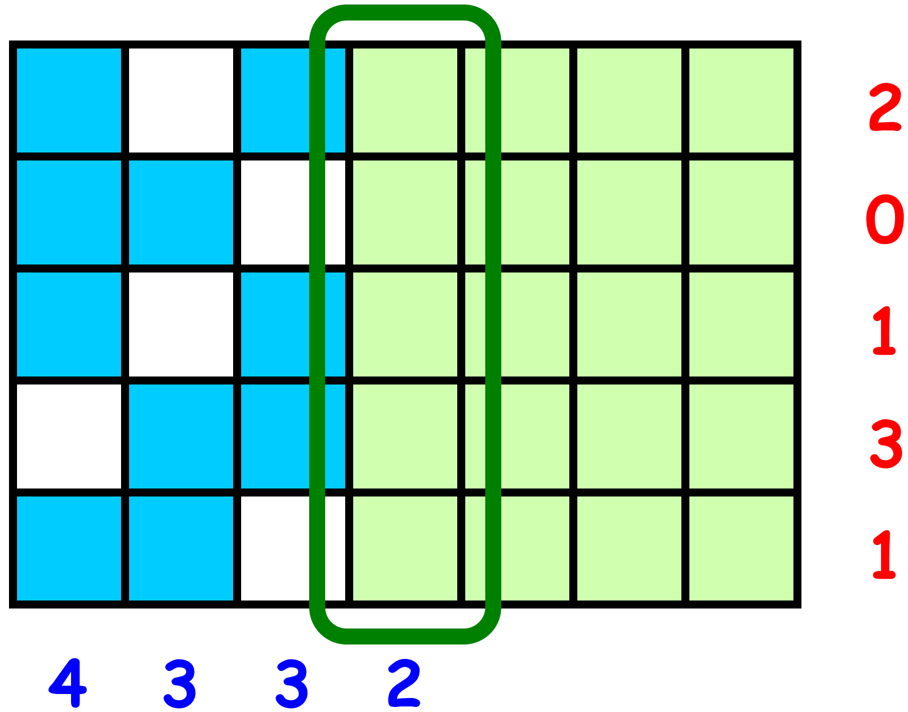
a specific σ

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

where product ranges over **i: rows with assignment 1**



Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

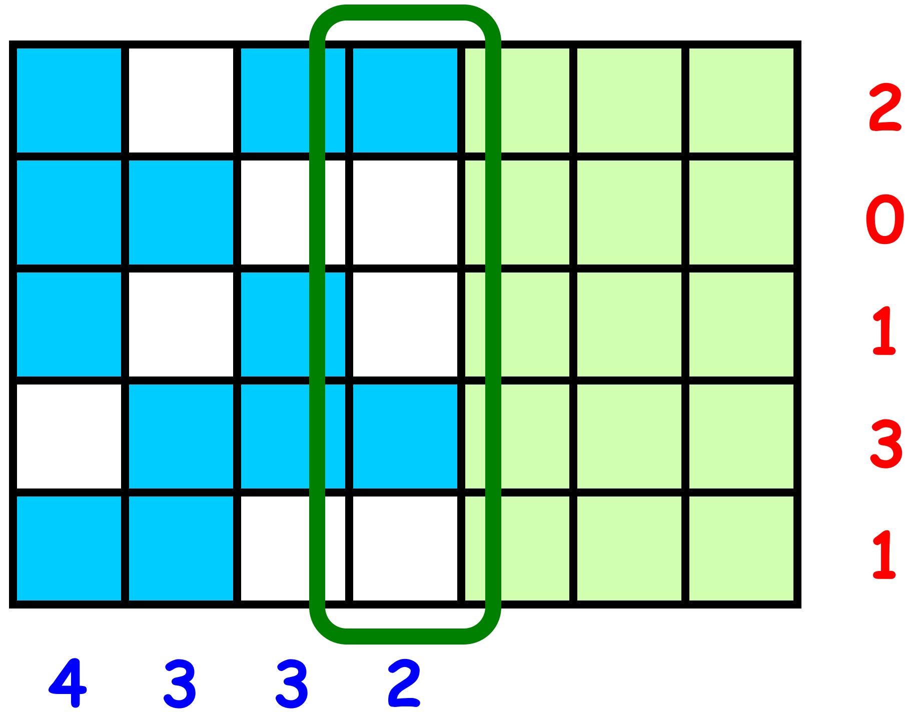
a specific σ

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

where product ranges over **i: rows with assignment 1**



Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

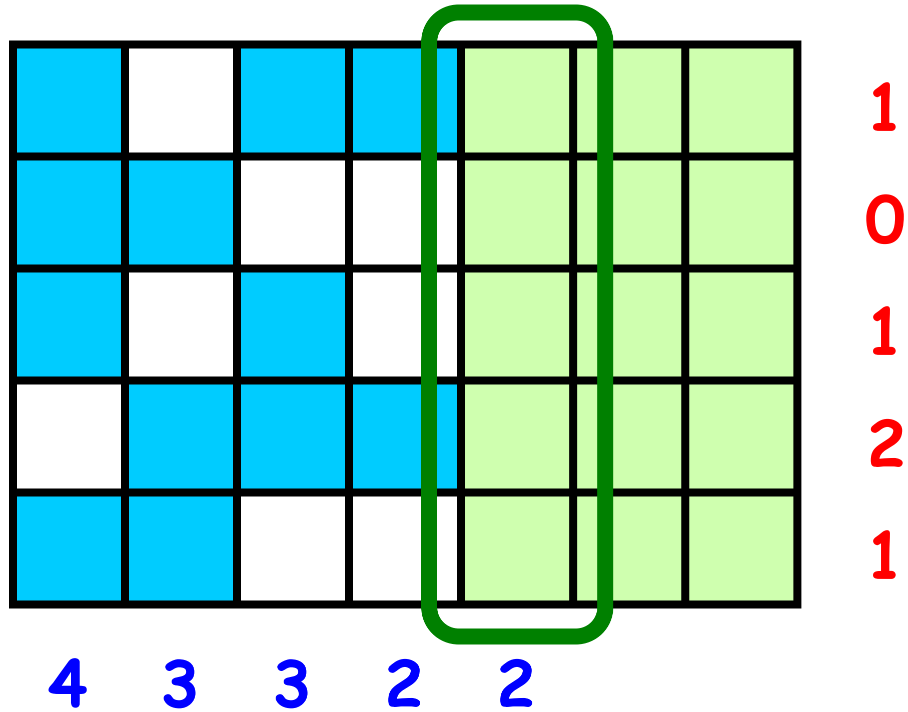
a specific σ

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

where product ranges over **i: rows with assignment 1**



Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

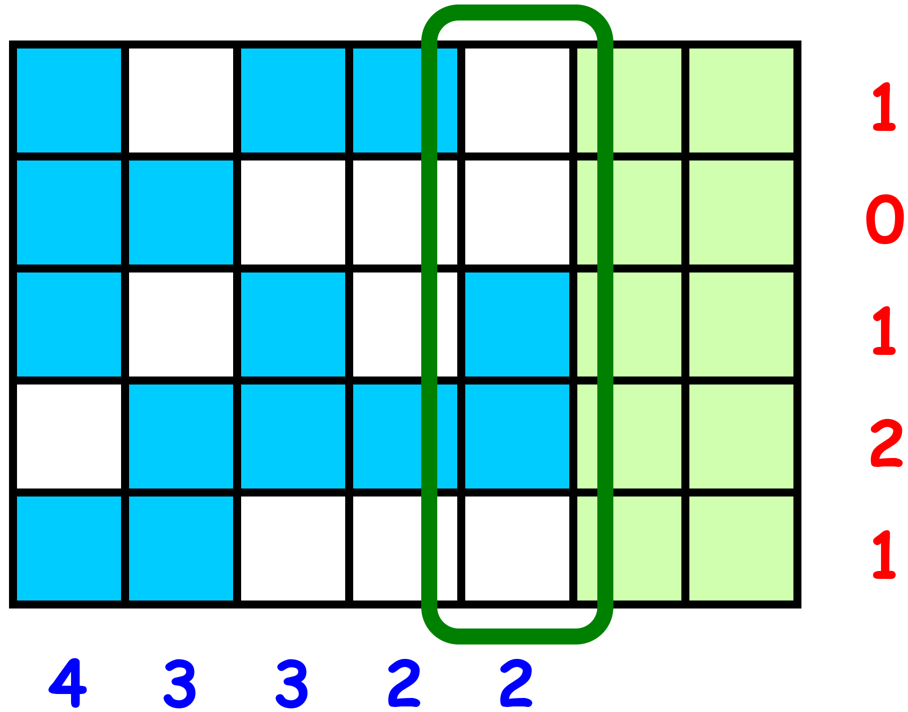
a specific σ

- fill table column-by-column
- assign each column ignoring other column sums

assign the column
with probability
proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

where product
ranges over **i: rows with assignment 1**



Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

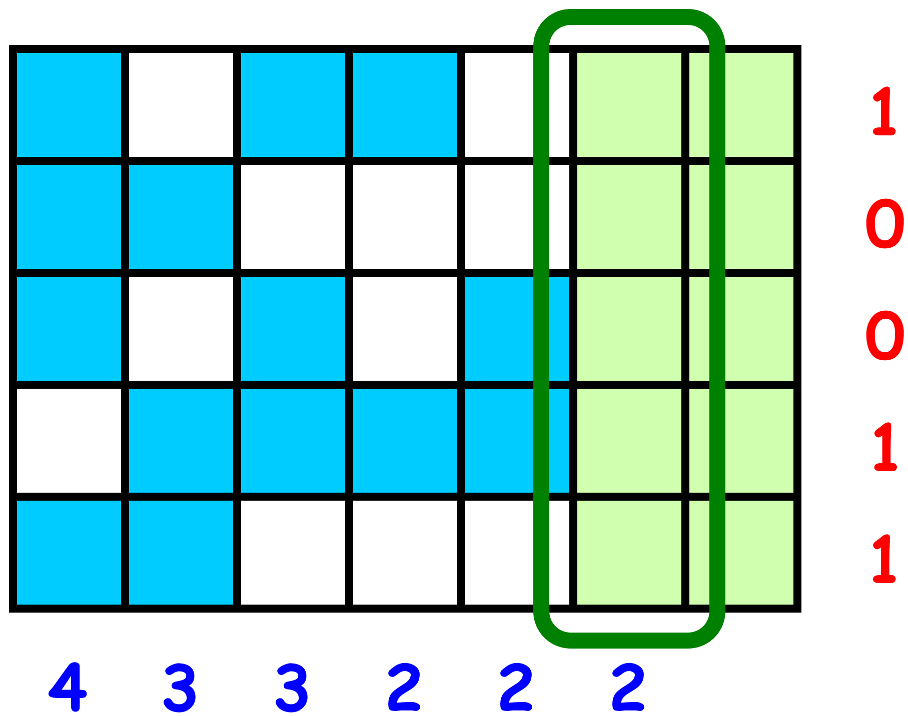
a specific σ

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

where product ranges over **i: rows with assignment 1**



Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

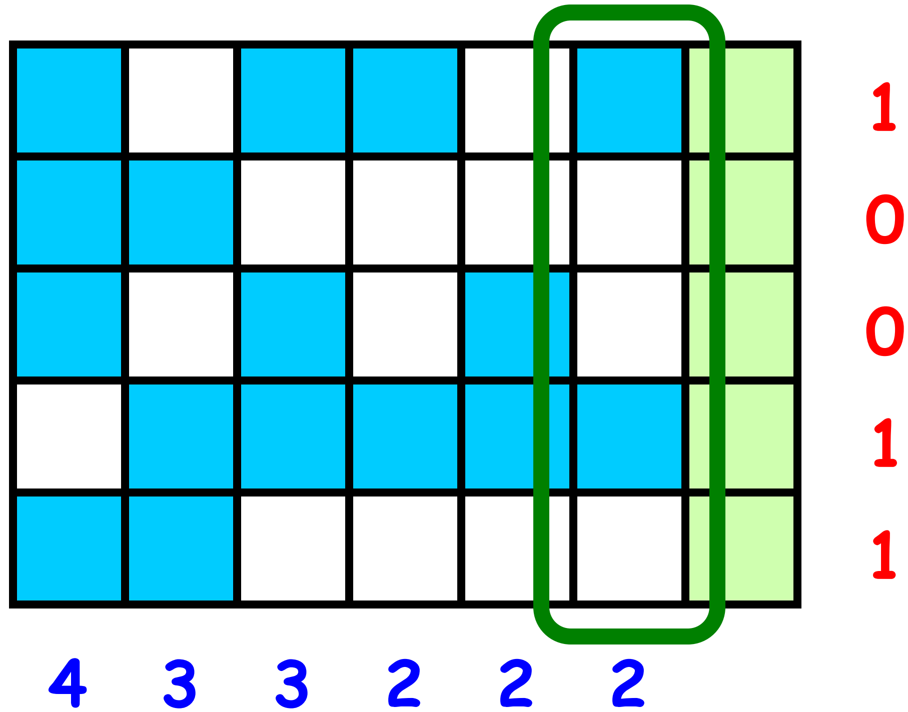
a specific σ

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

where product ranges over **i: rows with assignment 1**



Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

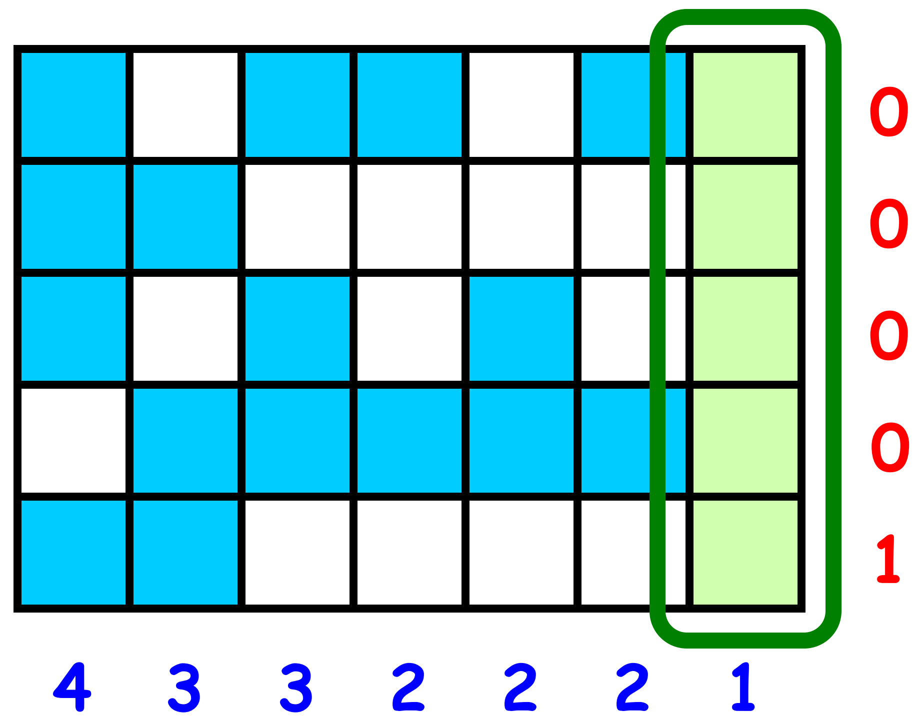
a specific σ

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\prod r_i/(n-r_i)$$

where product ranges over **i: rows with assignment 1**



Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

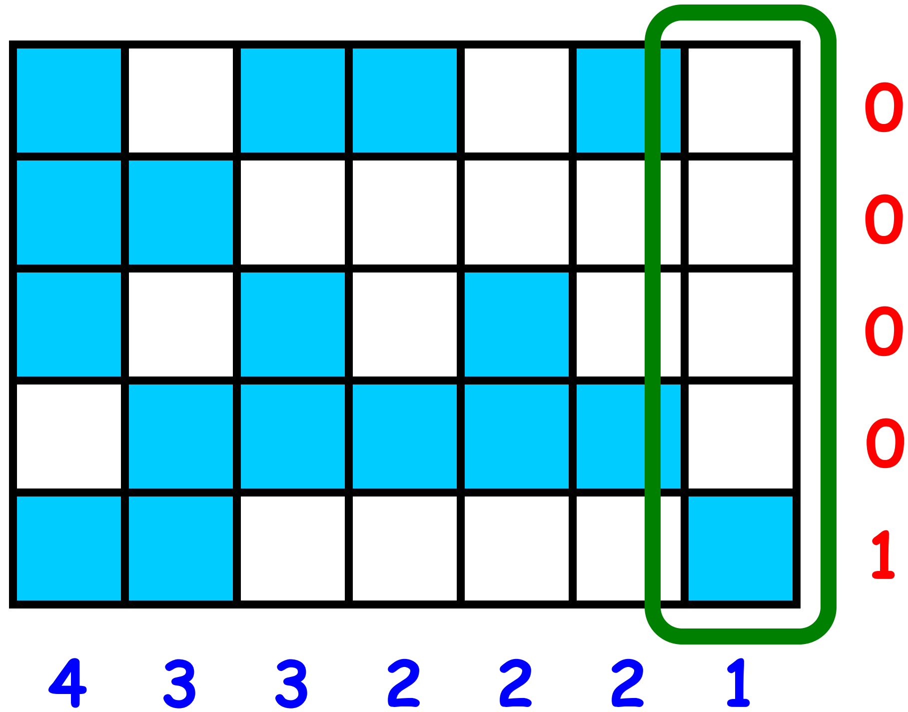
a specific σ

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

where product ranges over **i: rows with assignment 1**



Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

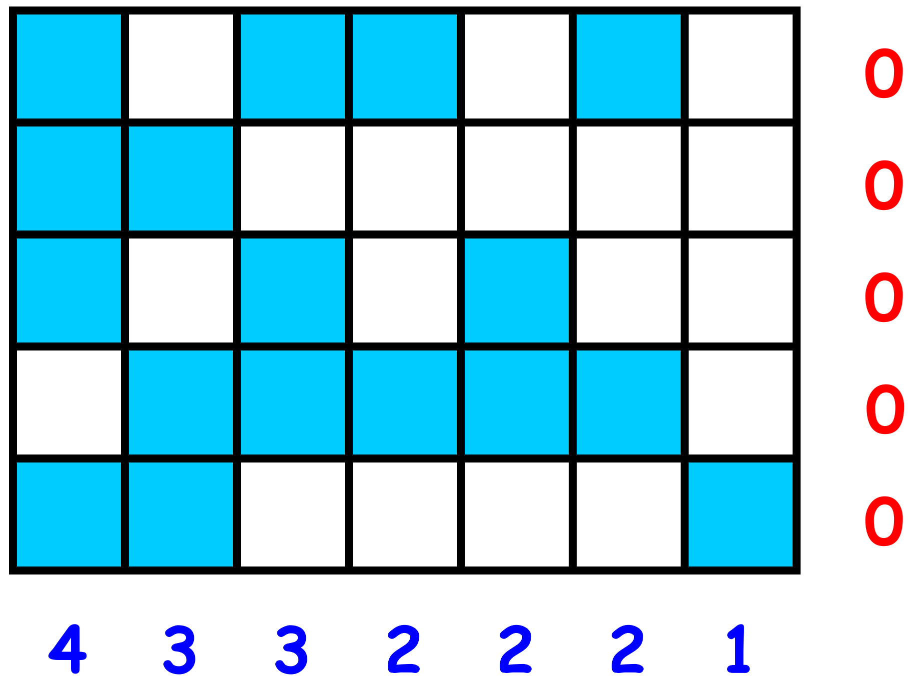
a specific σ

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

where product ranges over **i: rows with assignment 1**



Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

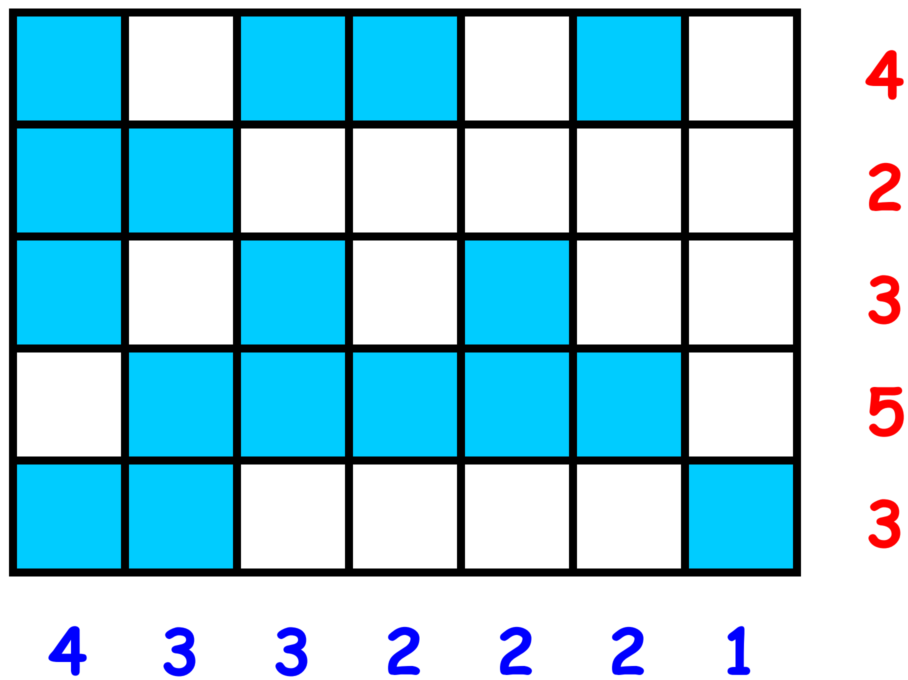
a specific σ

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

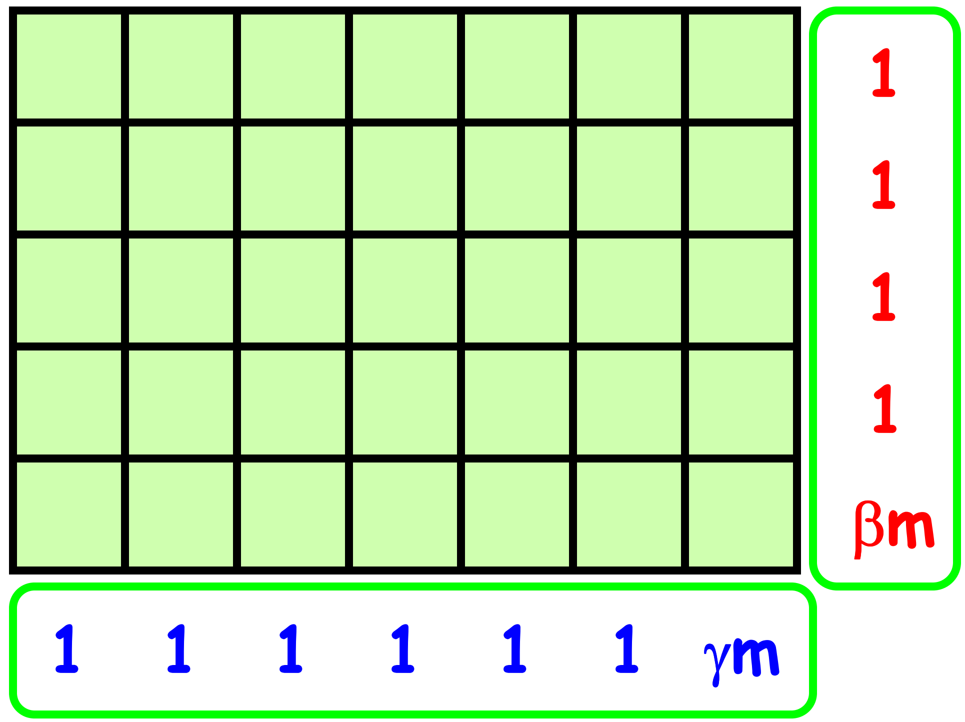
where product ranges over **i: rows with assignment 1**



A Counterexample for SIS

Thm [Bezáková-Sinclair-Štefankovič-Vigoda '06]:

For any $\beta \neq \gamma$, SIS output after any subexponential number of trials is **off by an exponential factor** (with high probability).

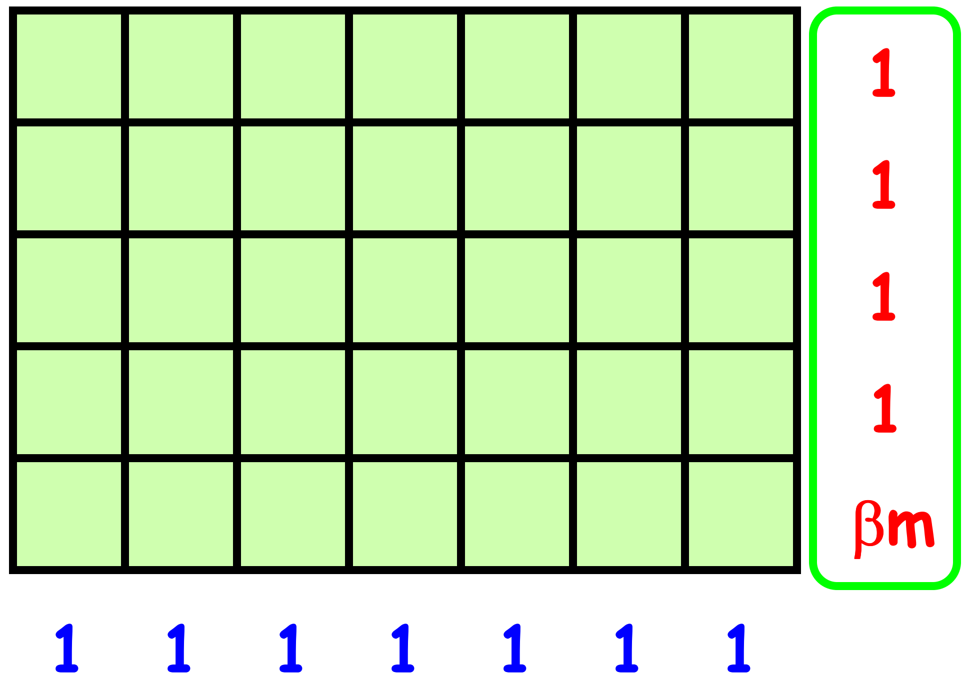


A Counterexample for SIS

Thm [Bezáková-Sinclair-Štefankovič-Vigoda '06]:

For any $\beta \neq \gamma$, SIS output after any subexponential number of trials is **off by an exponential factor** (with high probability).

Intuition



A Counterexample for SIS

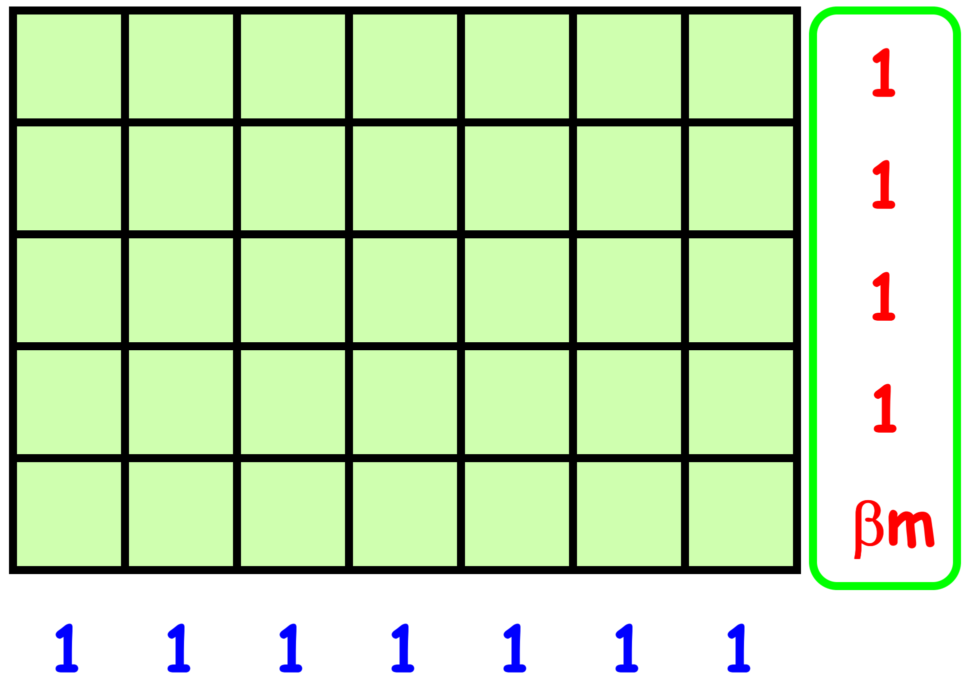
Thm [Bezáková-Sinclair-Štefankovič-Vigoda '06]:

For any $\beta \neq \gamma$, SIS output after any subexponential number of trials is **off by an exponential factor** (with high probability).

Intuition

Random table:

- randomly choose β**m** ones



A Counterexample for SIS

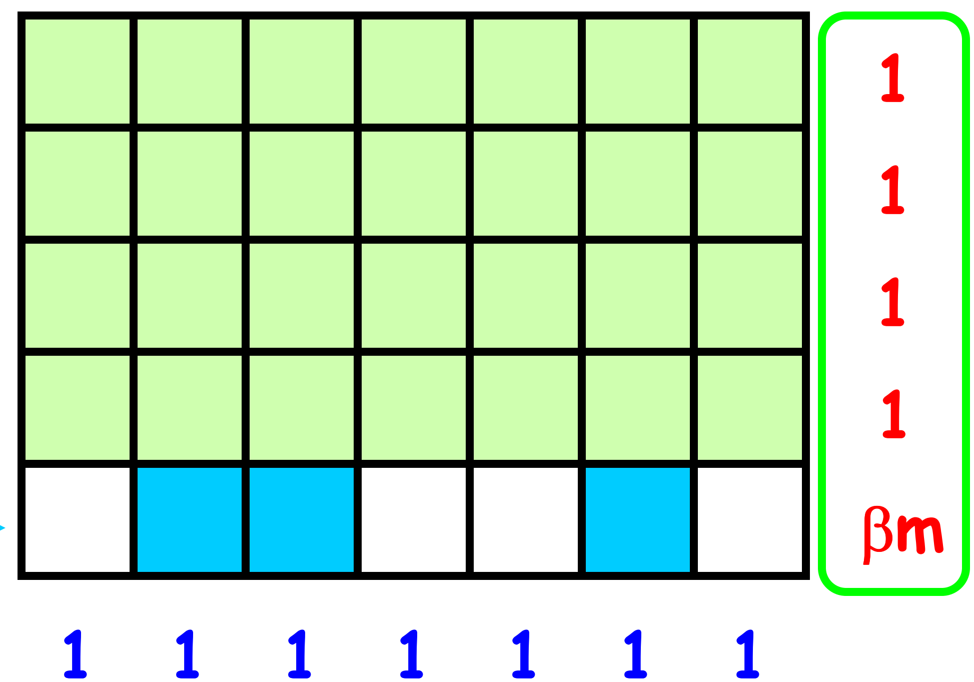
Thm [Bezáková-Sinclair-Štefankovič-Vigoda '06]:

For any $\beta \neq \gamma$, SIS output after any subexponential number of trials is **off by an exponential factor** (with high probability).

Intuition

Random table:

- randomly choose βm ones



A Counterexample for SIS

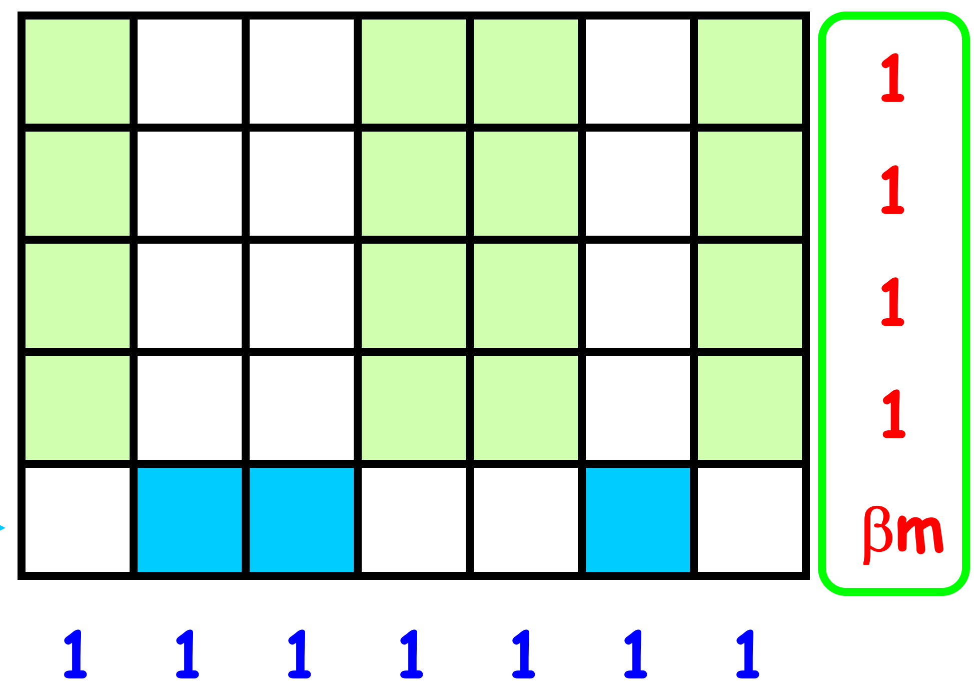
Thm [Bezáková-Sinclair-Štefankovič-Vigoda '06]:

For any $\beta \neq \gamma$, SIS output after any subexponential number of trials is **off by an exponential factor** (with high probability).

Intuition

Random table:

- randomly choose β**m** ones



A Counterexample for SIS

Thm [Bezáková-Sinclair-Štefankovič-Vigoda '06]:

For any $\beta \neq \gamma$, SIS output after any subexponential number of trials is **off by an exponential factor** (with high probability).

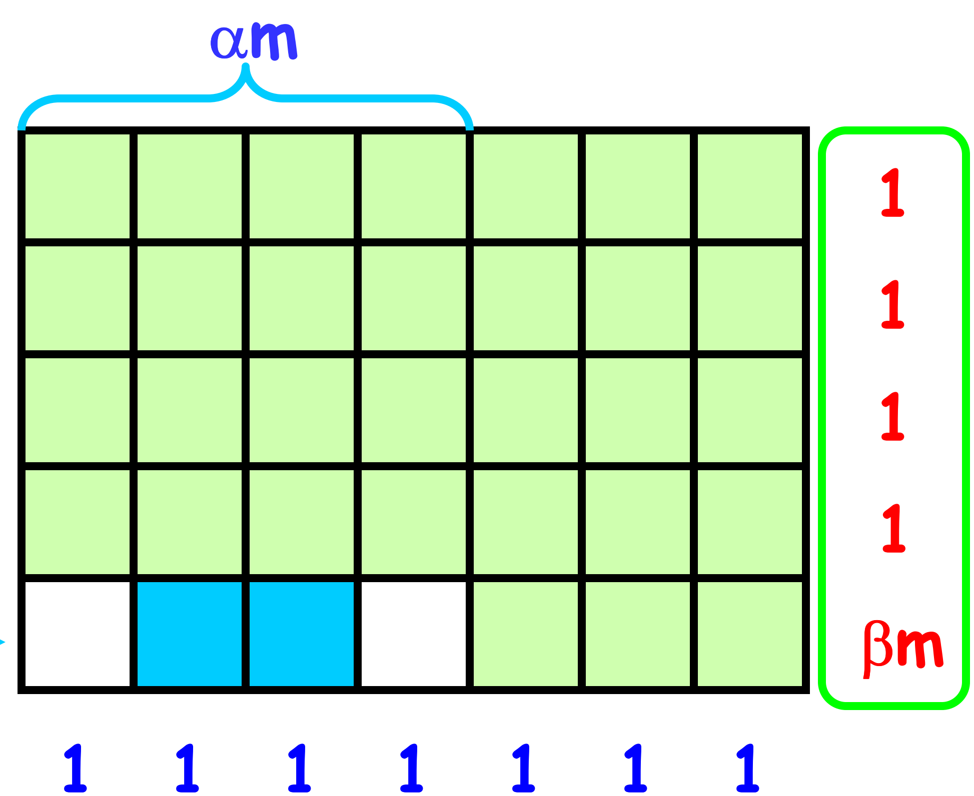
Intuition

Expect: $\alpha\beta m$ ones

SIS: asymptotically fewer

Random table:

- randomly choose βm ones



A Counterexample for SIS

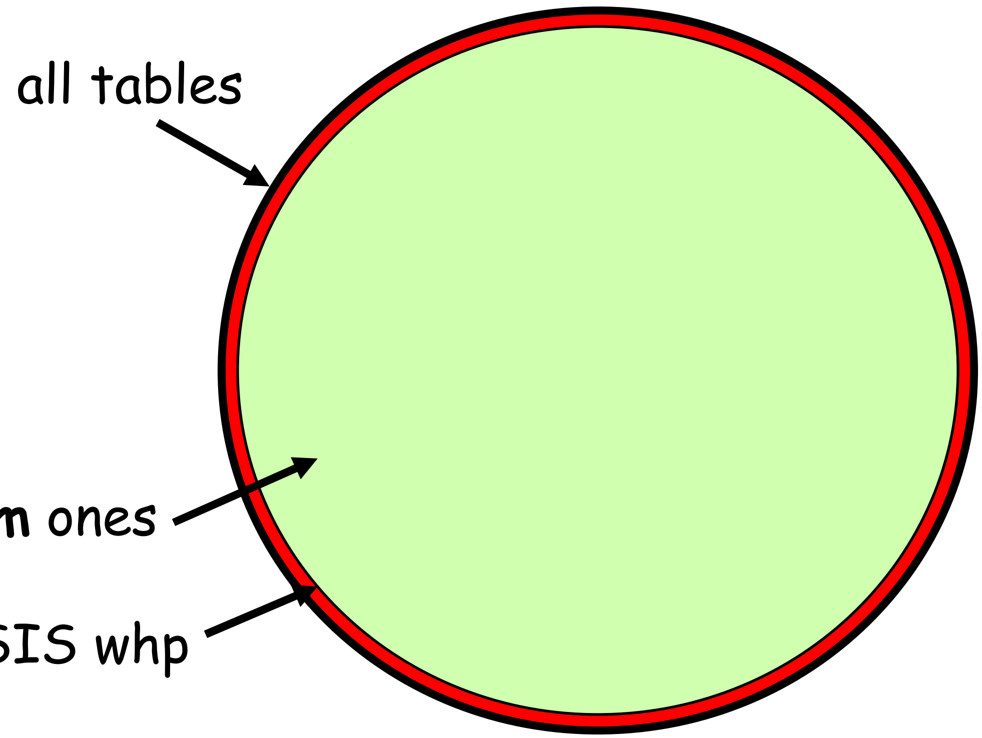
Thm [Bezáková-Sinclair-Štefankovič-Vigoda '06]:

For any $\beta \neq \gamma$, SIS output after any subexponential number of trials is **off by an exponential factor** (with high probability).

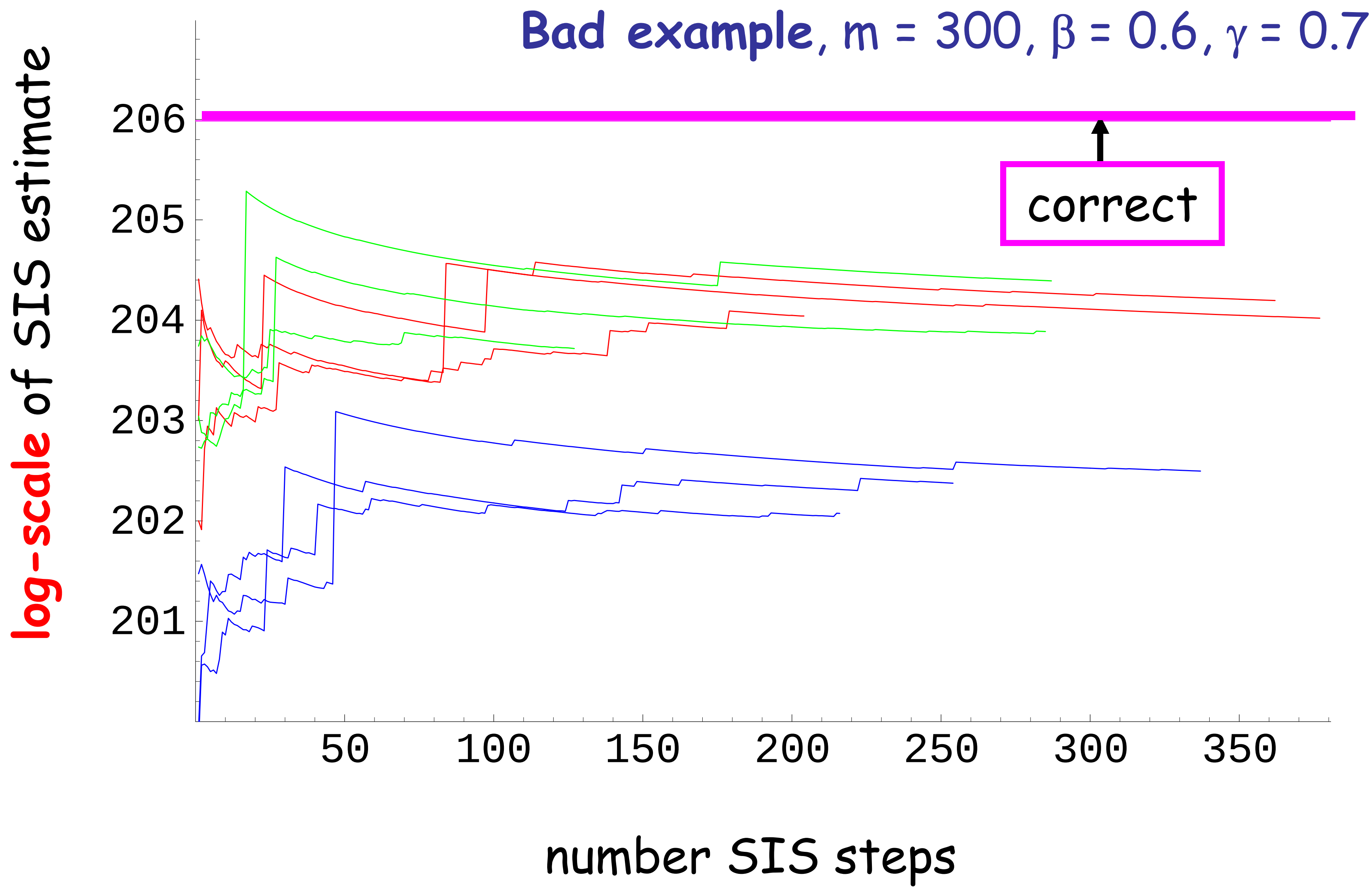
Intuition

Expect: $\alpha\beta\mathbf{m}$ ones

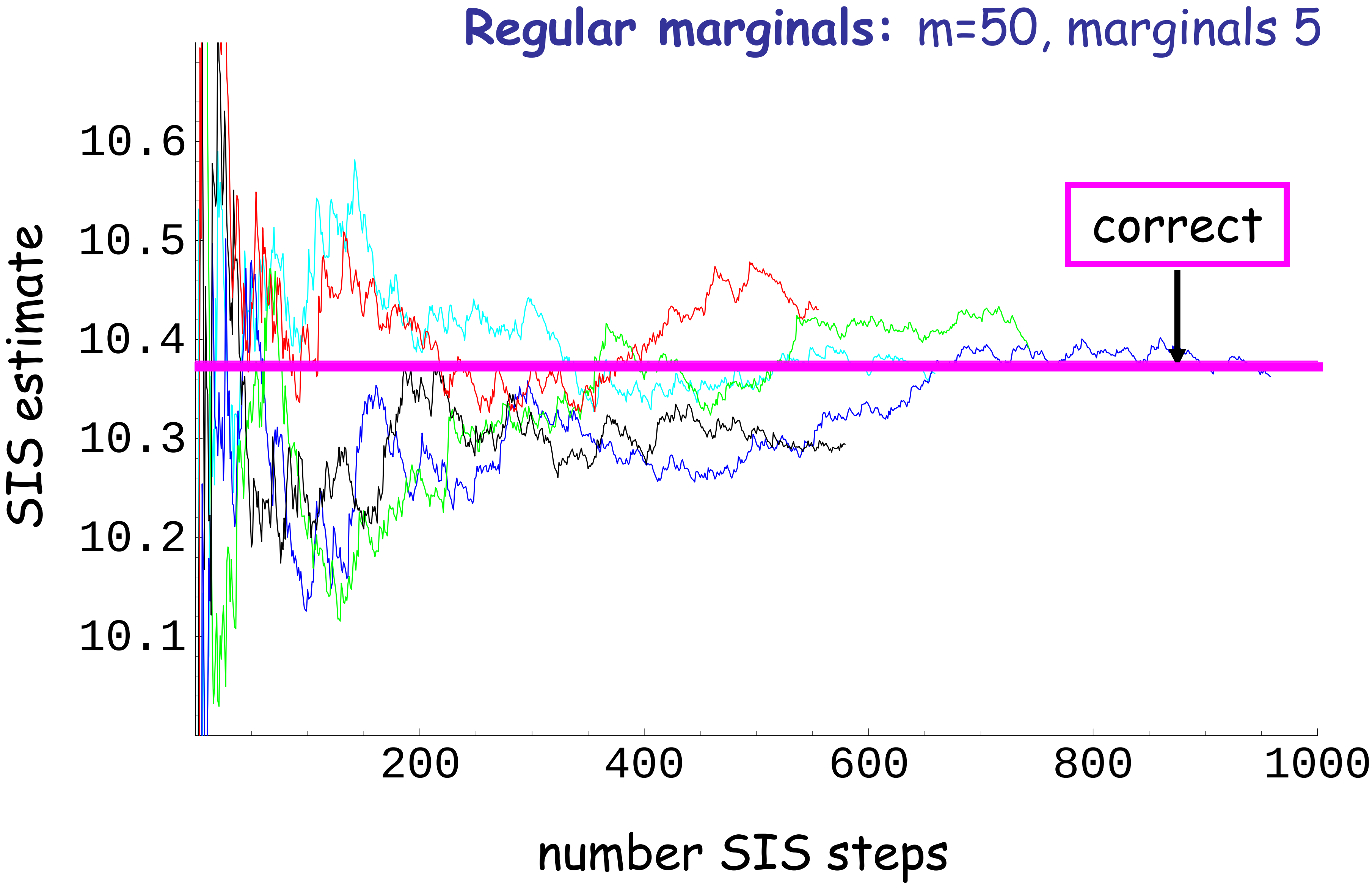
SIS: asymptotically fewer



SIS - Experimental Results



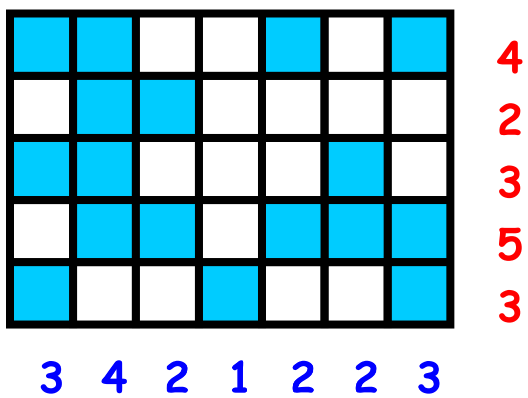
SIS - Experimental Results

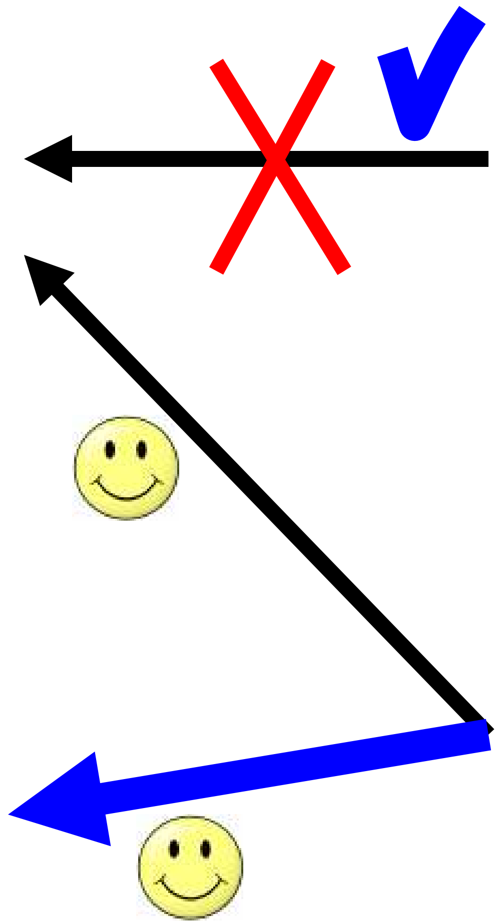


Problems

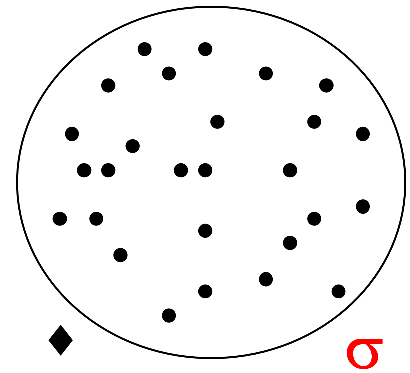
Heuristics

Binary contingency tables



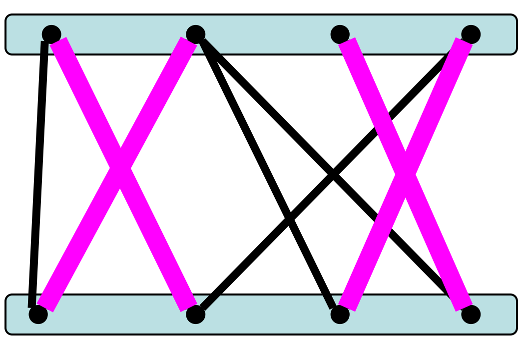


Importance sampling



Simulated annealing

Permanent

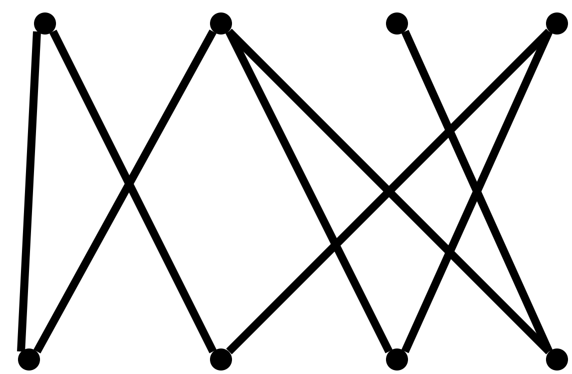


Broder Chain

[Broder '88]

What for: uniform sampling of perfect matchings

How: Markov chain on perfect + near-perfect matchings

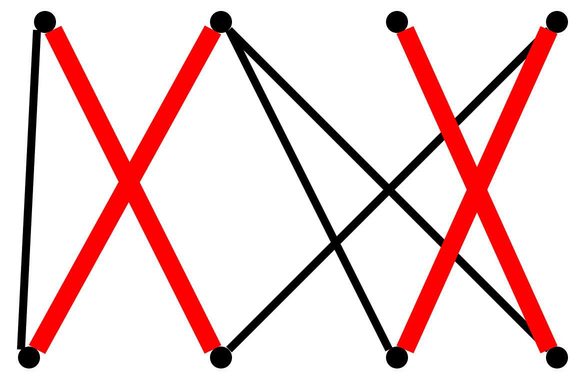


Broder Chain

[Broder '88]

What for: uniform sampling of perfect matchings

How: Markov chain on perfect + near-perfect matchings



Broder Chain

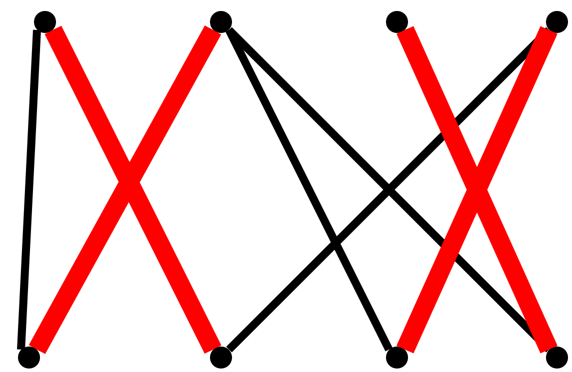
[Broder '88]

What for: uniform sampling of perfect matchings

How: Markov chain on perfect + near-perfect matchings

At a perfect matching:

- remove a random edge



Broder Chain

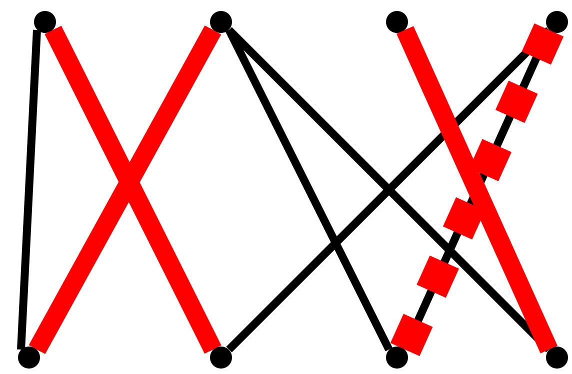
[Broder '88]

What for: uniform sampling of perfect matchings

How: Markov chain on perfect + near-perfect matchings

At a perfect matching:

- remove a random edge



Broder Chain

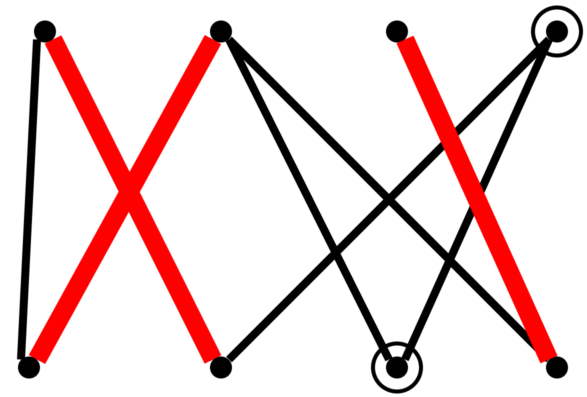
[Broder '88]

What for: uniform sampling of perfect matchings

How: Markov chain on perfect + near-perfect matchings

At a perfect matching:

- remove a random edge



Broder Chain

[Broder '88]

What for: uniform sampling of perfect matchings

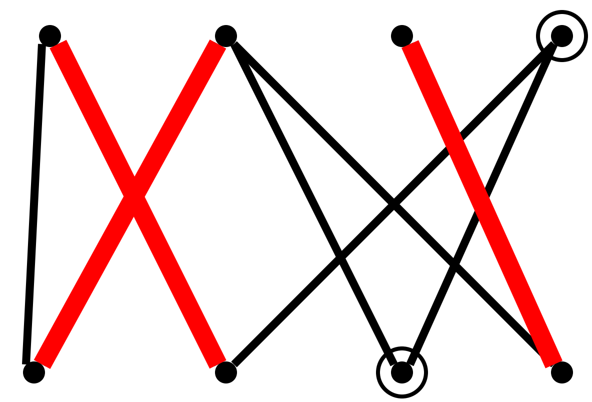
How: Markov chain on perfect + near-perfect matchings

At a perfect matching:

- remove a random edge

At a near matching:

- choose a random vertex



Broder Chain

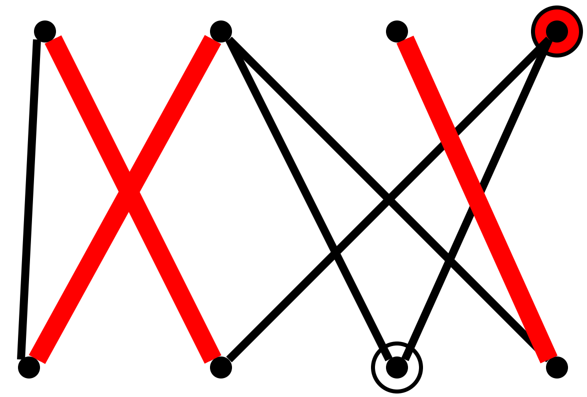
[Broder '88]

What for: uniform sampling of perfect matchings

How: Markov chain on perfect + near-perfect matchings

At a perfect matching:

- remove a random edge



At a near matching:

- choose a random vertex
  - if unmatched: match to the other hole

Broder Chain

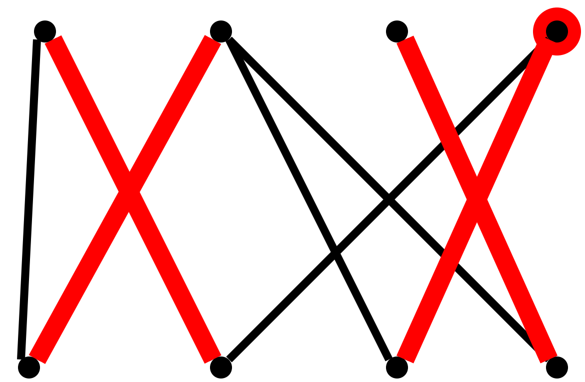
[Broder '88]

What for: uniform sampling of perfect matchings

How: Markov chain on perfect + near-perfect matchings

At a perfect matching:

- remove a random edge



At a near matching:

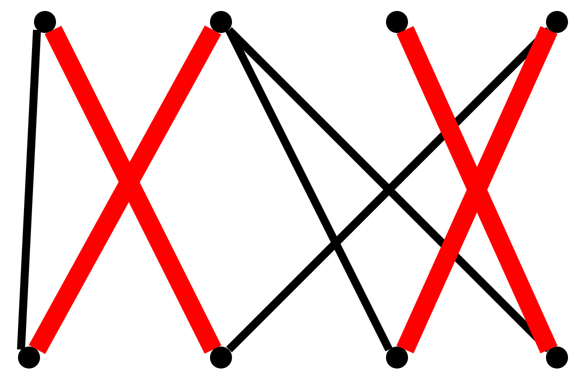
- choose a random vertex
    - if unmatched: match to the other hole

Broder Chain

[Broder '88]

What for: uniform sampling of perfect matchings

How: Markov chain on perfect + near-perfect matchings



At a perfect matching:

- remove a random edge

At a near matching:

- choose a random vertex
  - if unmatched: match to the other hole

Broder Chain

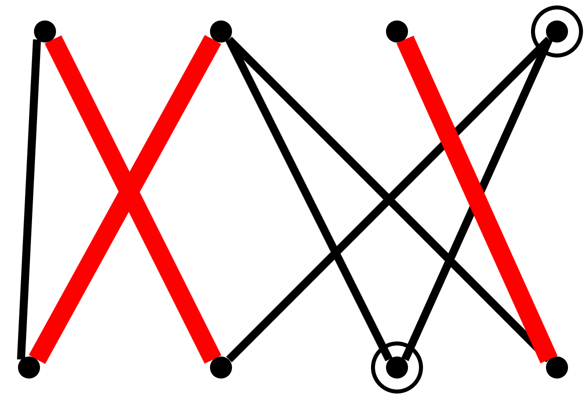
[Broder '88]

What for: uniform sampling of perfect matchings

How: Markov chain on perfect + near-perfect matchings

At a perfect matching:

- remove a random edge



At a near matching:

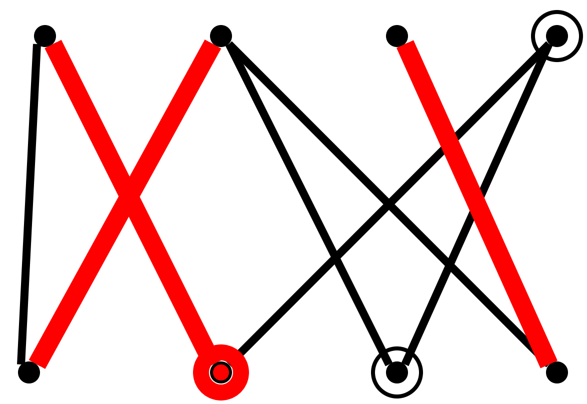
- choose a random vertex
    - if unmatched: match to the other hole

Broder Chain

[Broder '88]

What for: uniform sampling of perfect matchings

How: Markov chain on perfect + near-perfect matchings



At a perfect matching:

- remove a random edge

At a near matching:

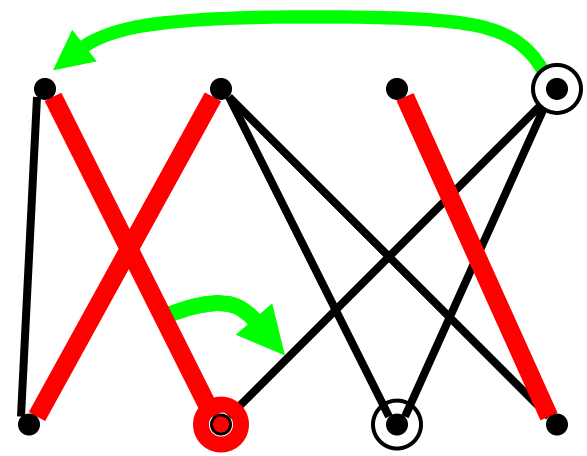
- choose a random vertex
  - if unmatched: match to the other hole
  - if matched: slide adjacent edge

Broder Chain

[Broder '88]

What for: uniform sampling of perfect matchings

How: Markov chain on perfect + near-perfect matchings



At a perfect matching:

- remove a random edge

At a near matching:

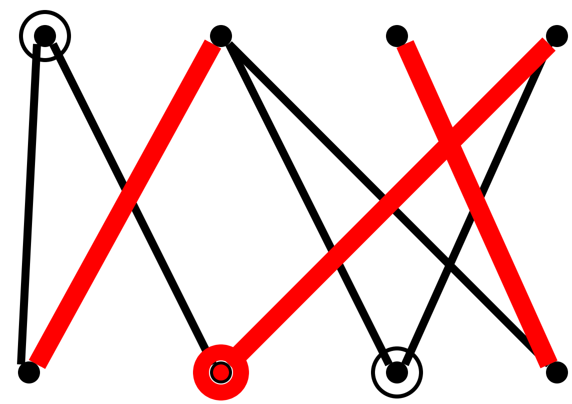
- choose a random vertex
  - if unmatched: match to the other hole
  - if matched: slide adjacent edge

Broder Chain

[Broder '88]

What for: uniform sampling of perfect matchings

How: Markov chain on perfect + near-perfect matchings



At a perfect matching:

- remove a random edge

At a near matching:

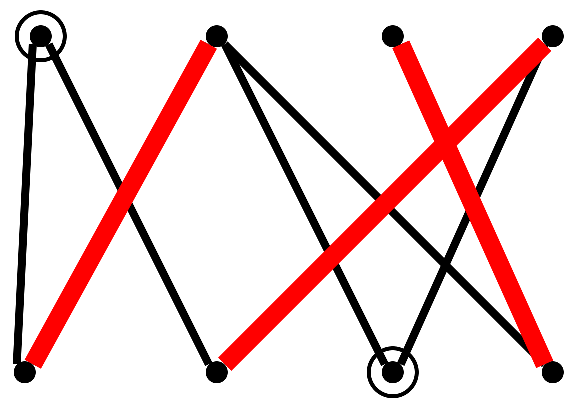
- choose a random vertex
  - if unmatched: match to the other hole
  - if matched: slide adjacent edge

Broder Chain

[Broder '88]

What for: uniform sampling of perfect matchings

How: Markov chain on perfect + near-perfect matchings



At a perfect matching:

- remove a random edge

At a near matching:

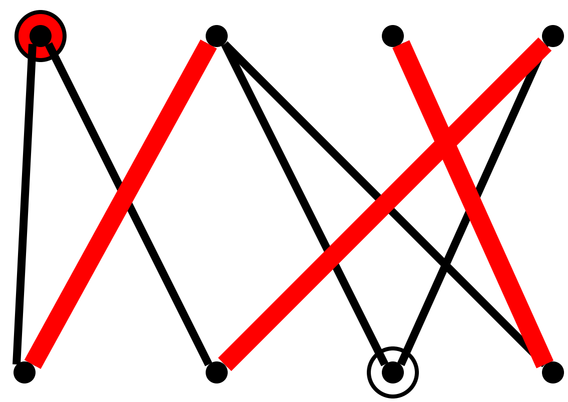
- choose a random vertex
  - if unmatched: match to the other hole
  - if matched: slide adjacent edge

Broder Chain

[Broder '88]

What for: uniform sampling of perfect matchings

How: Markov chain on perfect + near-perfect matchings



At a perfect matching:

- remove a random edge

At a near matching:

- choose a random vertex
  - if unmatched: match to the other hole
  - if matched: slide adjacent edge

Broder Chain

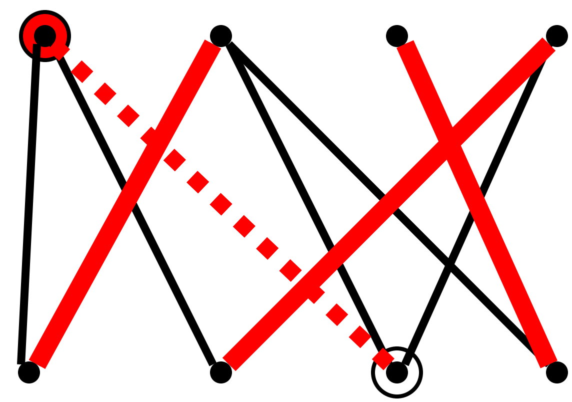
[Broder '88]

What for: uniform sampling of perfect matchings

How: Markov chain on perfect + near-perfect matchings

At a perfect matching:

- remove a random edge



At a near matching:

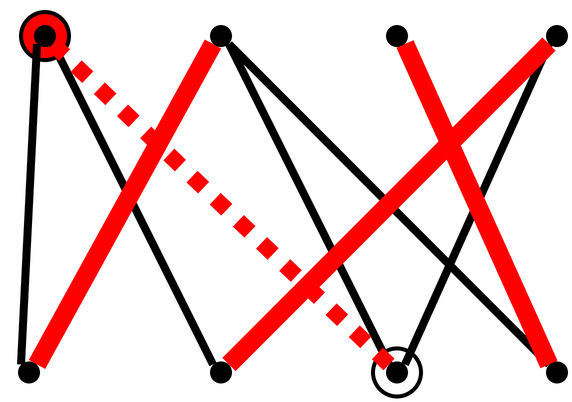
- choose a random vertex
  - if unmatched: match to the other hole
  - if matched: slide adjacent edge

Broder Chain

[Broder '88]

What for: uniform sampling of perfect matchings

How: Markov chain on perfect + near-perfect matchings



At a perfect matching:

- remove a random edge

At a near matching:

- choose a random vertex
  - if unmatched: match to the other hole
  - if matched: slide adjacent edge

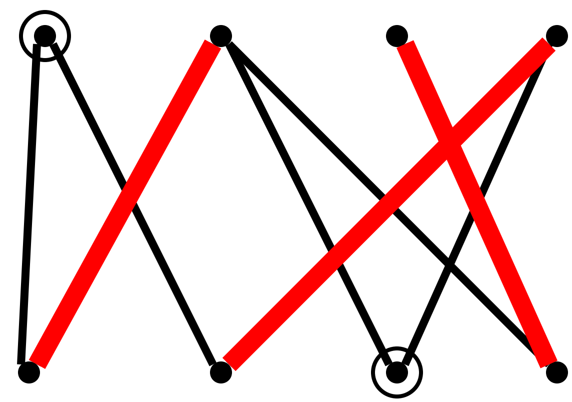
$\left.\begin{array}{l} \text{if unmatched: match to the other hole} \\ \text{if matched: slide adjacent edge} \end{array}\right\}$ (if can)

Broder Chain

[Broder '88]

What for: uniform sampling of perfect matchings

How: Markov chain on perfect + near-perfect matchings



At a perfect matching:

- remove a random edge

At a near matching:

- choose a random vertex
  - if unmatched: match to the other hole
  - if matched: slide adjacent edge

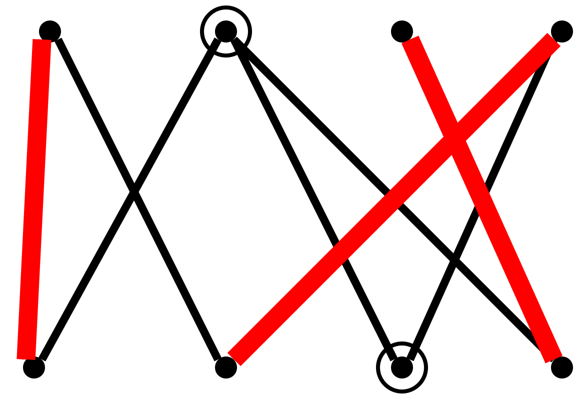
$\Big\}$ (if can)

Broder Chain

[Broder '88]

What for: uniform sampling of perfect matchings

How: Markov chain on perfect + near-perfect matchings



At a perfect matching:

- remove a random edge

At a near matching:

- choose a random vertex
  - if unmatched: match to the other hole
  - if matched: slide adjacent edge

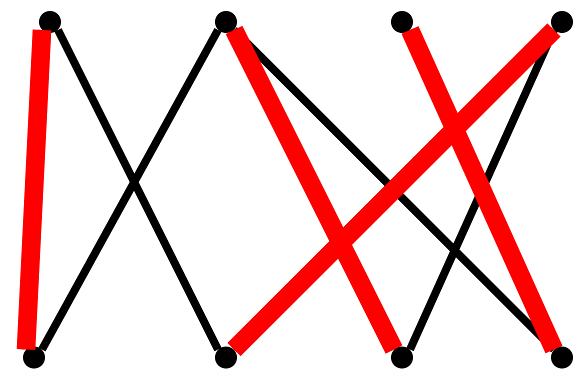
(if can)

Broder Chain

[Broder '88]

What for: uniform sampling of perfect matchings

How: Markov chain on perfect + near-perfect matchings



At a perfect matching:

- remove a random edge

At a near matching:

- choose a random vertex
  - if unmatched: match to the other hole
  - if matched: slide adjacent edge

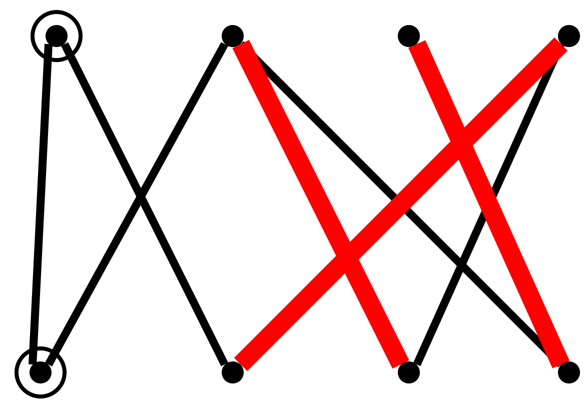
$\Big\}$ (if can)

Broder Chain

[Broder '88]

What for: uniform sampling of perfect matchings

How: Markov chain on perfect + near-perfect matchings



At a perfect matching:

- remove a random edge

At a near matching:

- choose a random vertex
  - if unmatched: match to the other hole
  - if matched: slide adjacent edge

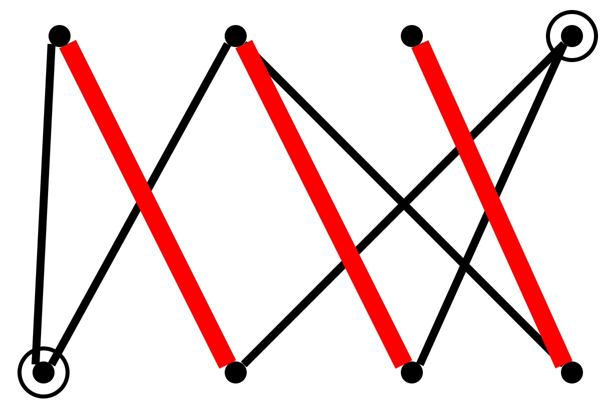
(if can)

Broder Chain

[Broder '88]

What for: uniform sampling of perfect matchings

How: Markov chain on perfect + near-perfect matchings



At a perfect matching:

- remove a random edge

At a near matching:

- choose a random vertex
  - if unmatched: match to the other hole
  - if matched: slide adjacent edge

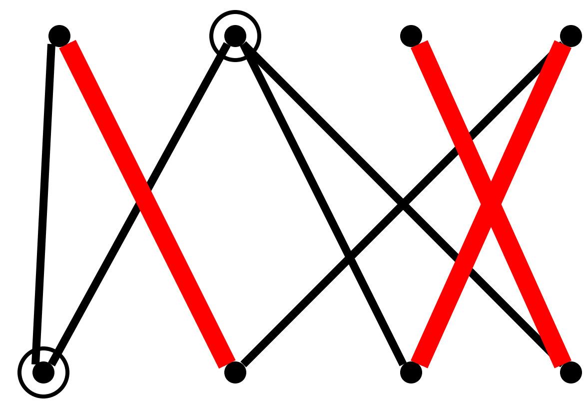
(if can)

Broder Chain

[Broder '88]

What for: uniform sampling of perfect matchings

How: Markov chain on perfect + near-perfect matchings



At a perfect matching:

- remove a random edge

At a near matching:

- choose a random vertex
  - if unmatched: match to the other hole
  - if matched: slide adjacent edge

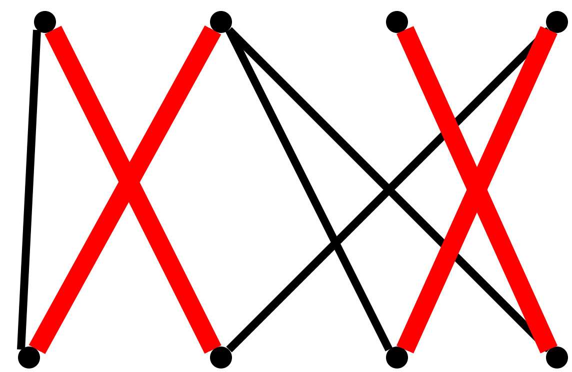
$\Big\}$ (if can)

Broder Chain

[Broder '88]

What for: uniform sampling of perfect matchings

How: Markov chain on perfect + near-perfect matchings



At a perfect matching:

- remove a random edge

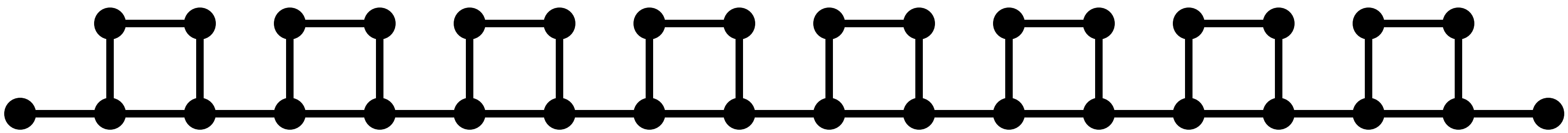
At a near matching:

- choose a random vertex
  - if unmatched: match to the other hole
  - if matched: slide adjacent edge

$\left.\begin{array}{l}\text{if unmatched: match to the other hole}\\ \text{if matched: slide adjacent edge}\end{array}\right\}$ (if can)

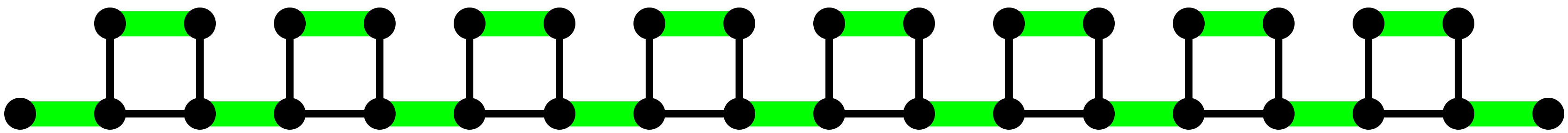
Broder Chain

Mixes in polynomial time? Even if it did…



Broder Chain

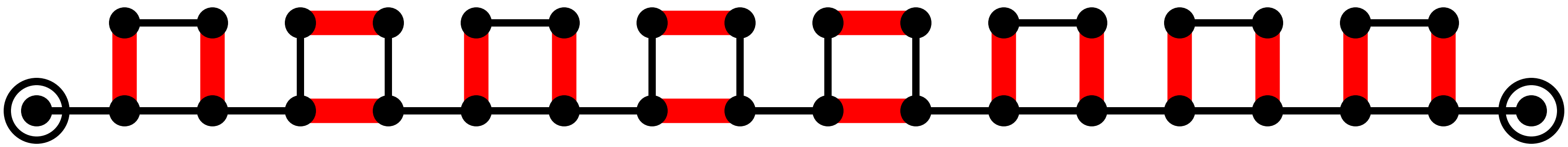
Mixes in polynomial time ? Even if it did…



1 perfect matching

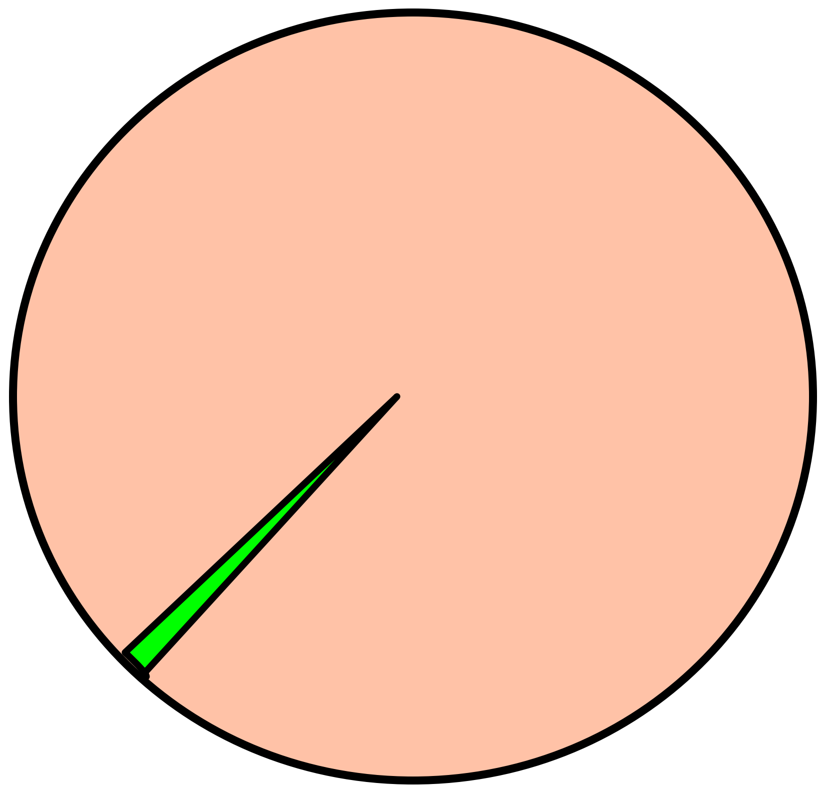
Broder Chain

Mixes in polynomial time ? Even if it did...



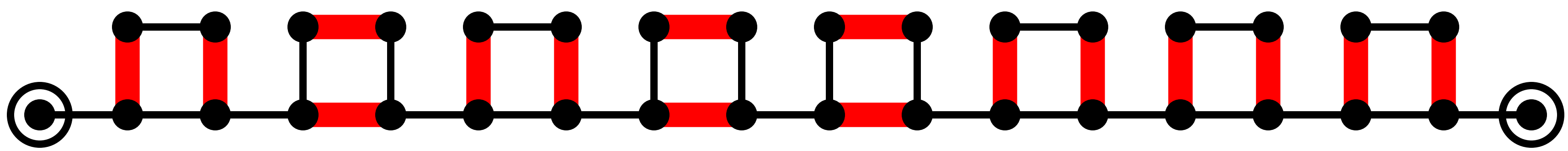
1 perfect matching

$\geq 2^{(n/4)}$ near matchings



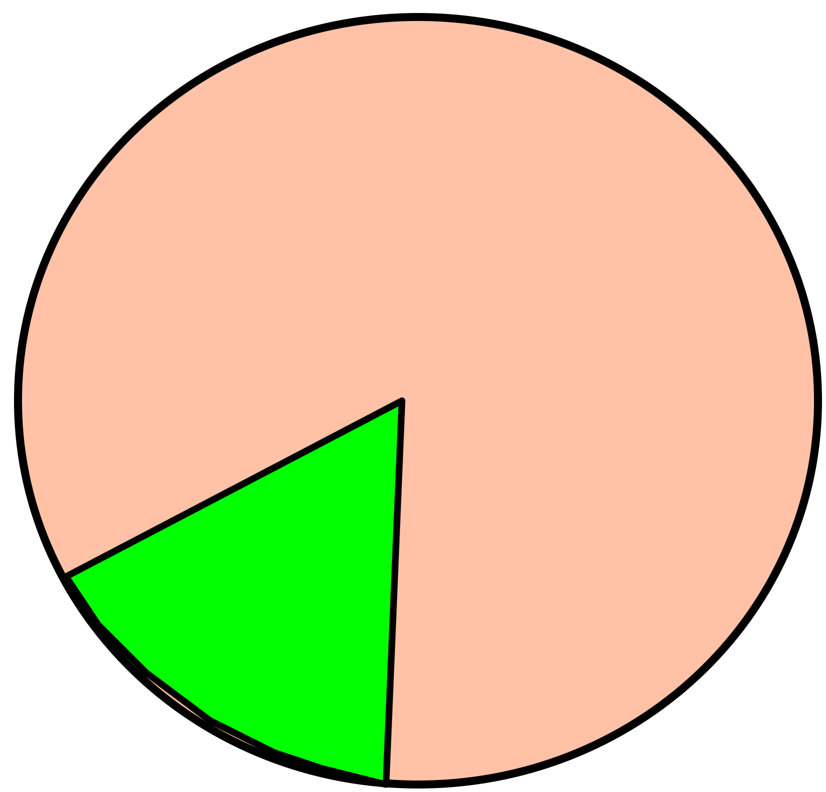
Broder Chain

Mixes in polynomial time ? Even if it did...



1 perfect matching

$\geq 2^{(n/4)}$ near matchings



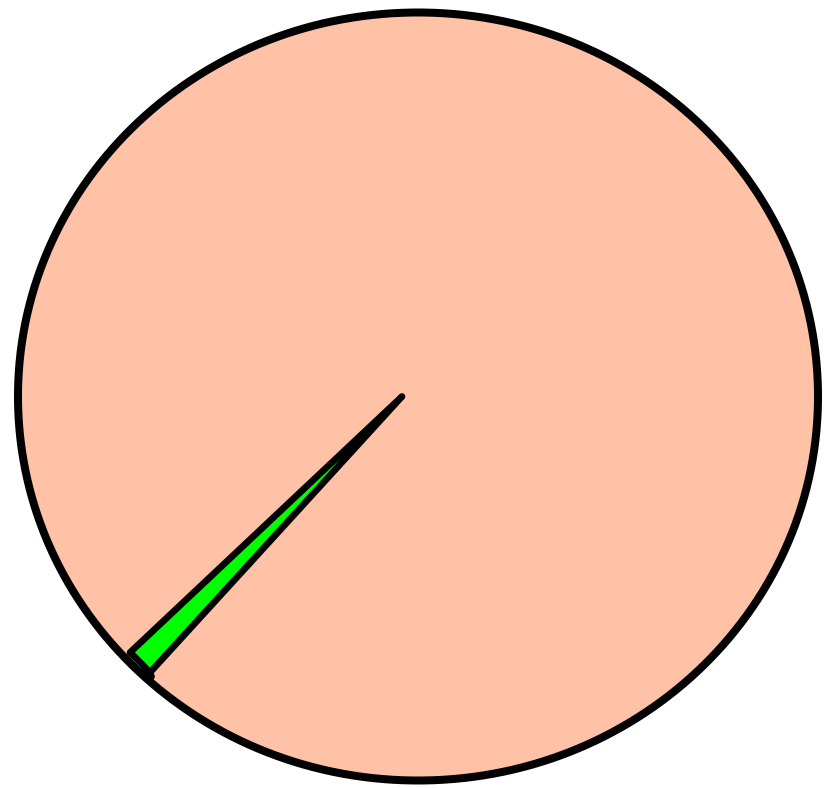
Thm [Jerrum-Sinclair '89]:

Rapid mixing if perfects **polynomially** related to nears.

Broder Chain

Idea [Jerrum-Sinclair-Vigoda '01]:

Change the **weight** so that **perfect matchings** take **polynomial** fraction.

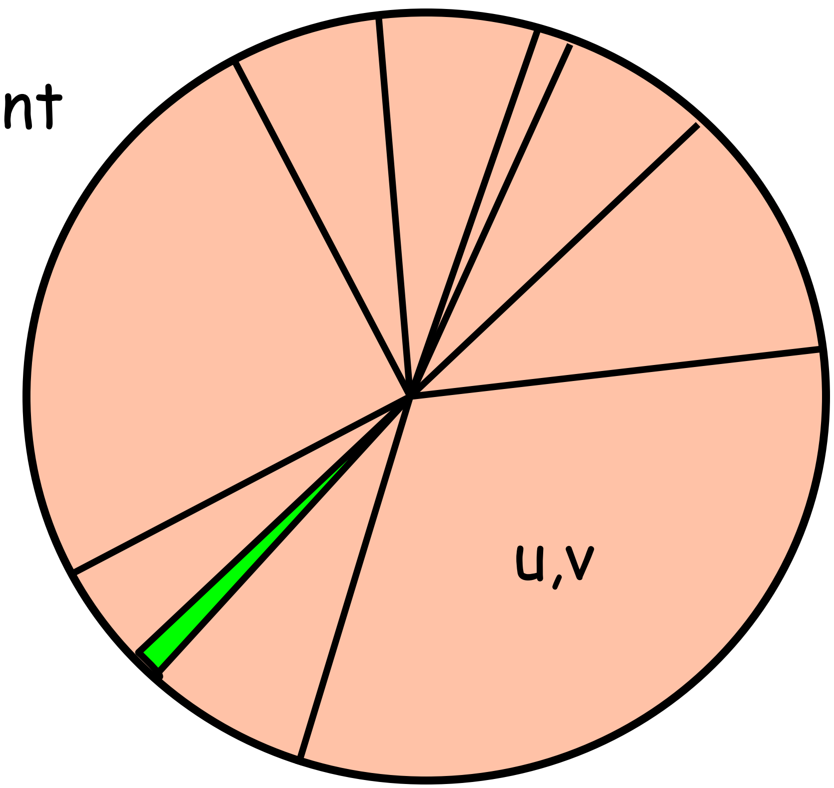


Broder Chain

Idea [Jerrum-Sinclair-Vigoda '01]:

Change the **weight** so that **perfect matchings** take **polynomial** fraction.

n^2+1 regions, very different size

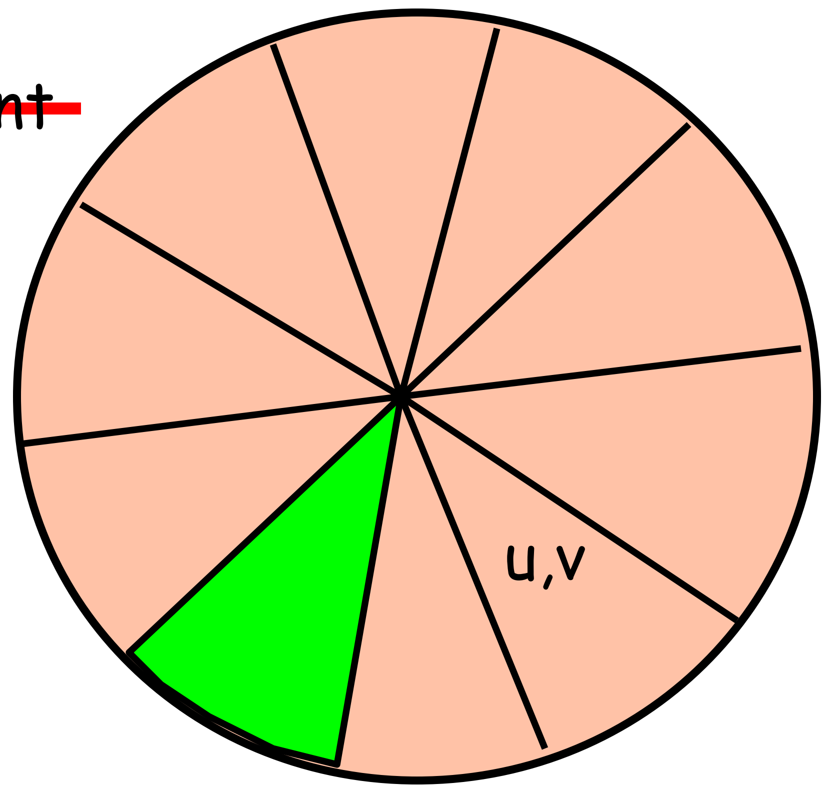


Broder Chain

Idea [Jerrum-Sinclair-Vigoda '01]:

Change the **weight** so that **perfect matchings** take **polynomial** fraction.

n^2+1 regions,
~~very different~~
the same size

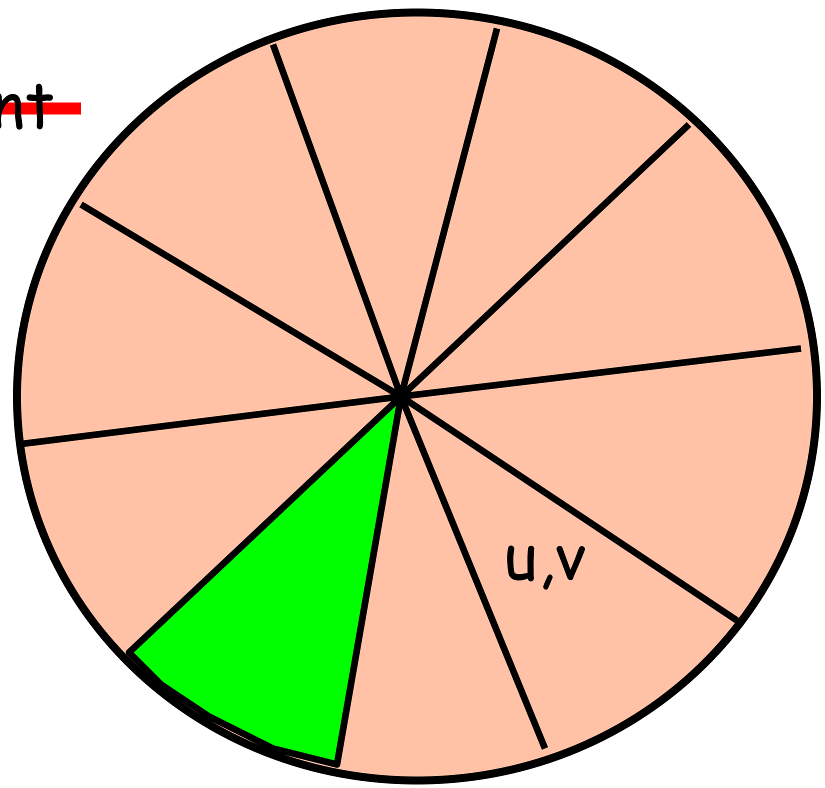


Broder Chain

Idea [Jerrum-Sinclair-Vigoda '01]:

Change the **weight** so that **perfect matchings** take **polynomial** fraction.

n^2+1 regions,
~~very different~~
the same size



Ideal weights
(for a matching with holes u,v):

$$\boxed{\dfrac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}}$$

Broder Chain

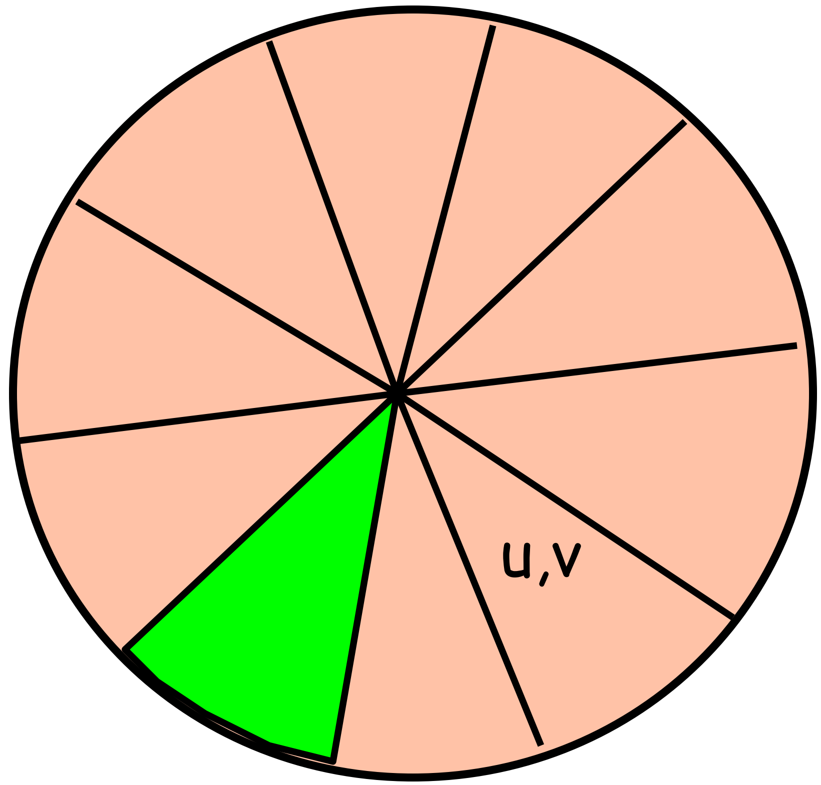
Good: A perfect matching sampled with prob. $1/(n^2+1)$

Bad: Computing ideal weights as hard as original problem **?**

Solution: **Approximate**

Ideal weights
(for a matching with holes u,v):

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$



Simulated Annealing for Permanent

Unordered state → Ordered state

(EASY) (DIFFICULT)

$\boxed{G}$

Solution: **Approximate**

Ideal weights
(for a matching with holes u,v):

$$\boxed{\frac{\textbf{(\# perfects)}}{\textbf{(\# nears with holes u,v)}}}$$

Simulated Annealing for Permanent

Unordered state → Ordered state

(EASY) (DIFFICULT)

$K_{n,n}$ → G

Solution: **Approximate**

Ideal weights
(for a matching with holes u,v):

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$

Simulated Annealing for Permanent

Unordered state → Ordered state

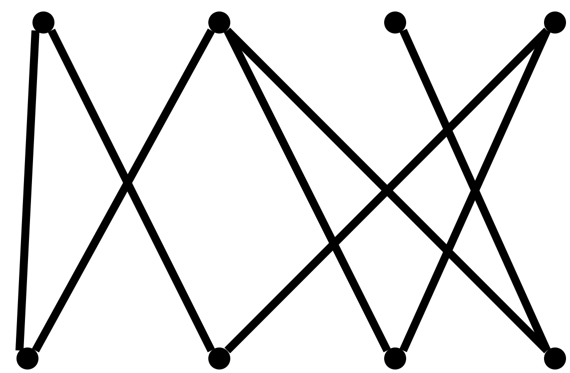
(EASY) (DIFFICULT)

$K_{n,n}$ → G

Solution: **Approximate**

Ideal weights
(for a matching with holes u,v):

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$



Simulated Annealing for Permanent

Unordered state → Ordered state

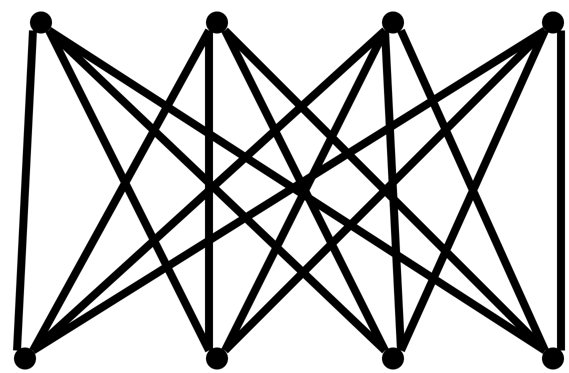
(EASY) (DIFFICULT)

$K_{n,n}$ → G

Solution: **Approximate**

Ideal weights
(for a matching with holes u,v):

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$



Simulated Annealing for Permanent

Unordered state → Ordered state

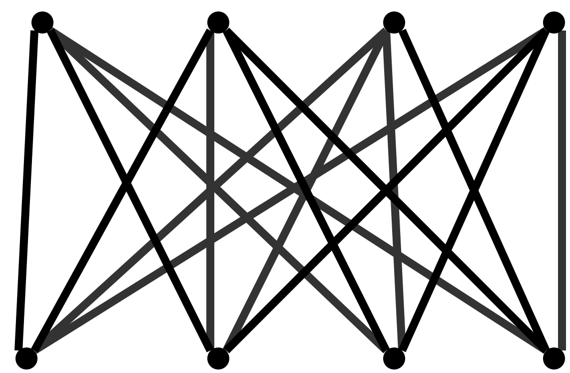
(EASY) (DIFFICULT)

$K_{n,n}$ → G

Solution: **Approximate**

Ideal weights
(for a matching with holes u,v):

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$



Simulated Annealing for Permanent

Unordered state → Ordered state

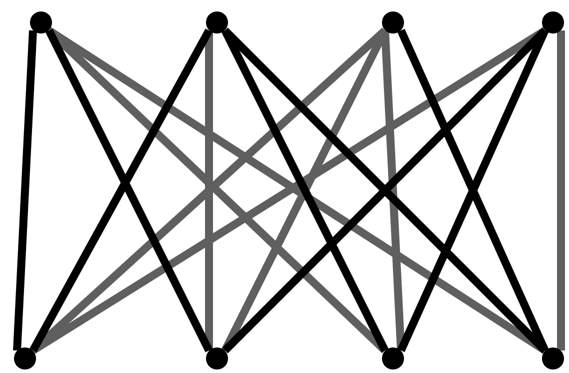
(EASY) (DIFFICULT)

$K_{n,n}$ → G

Solution: **Approximate**

Ideal weights
(for a matching with holes u,v):

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes } u,v)}$$



Simulated Annealing for Permanent

Unordered state → Ordered state

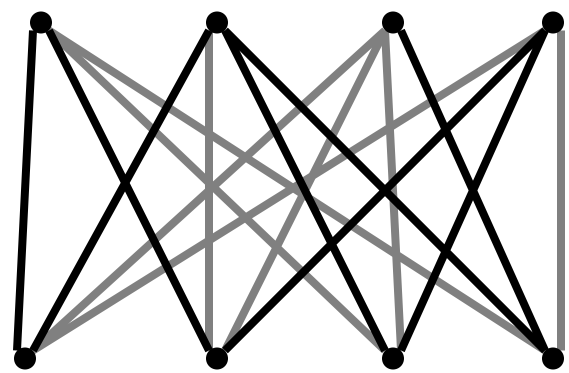
(EASY) (DIFFICULT)

$K_{n,n}$ → G

Solution: **Approximate**

Ideal weights
(for a matching with holes u,v):

$$\frac{(\#\ \text{perfects})}{(\#\ \text{nears with holes } u,v)}$$



Simulated Annealing for Permanent

Unordered state → Ordered state

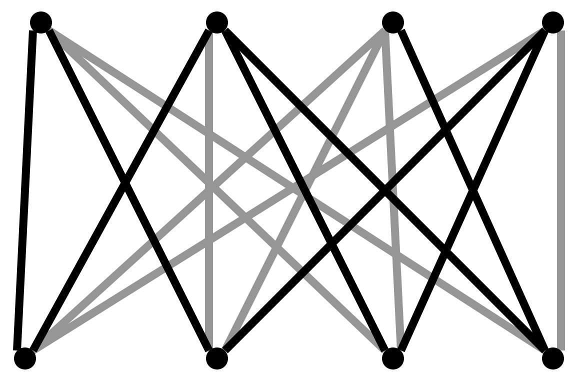
(EASY) (DIFFICULT)

$\boxed{K_{n,n}} \longrightarrow \boxed{G}$

Solution: **Approximate**

Ideal weights
(for a matching with holes u,v):

$$\boxed{\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}}$$



Simulated Annealing for Permanent

Unordered state → Ordered state

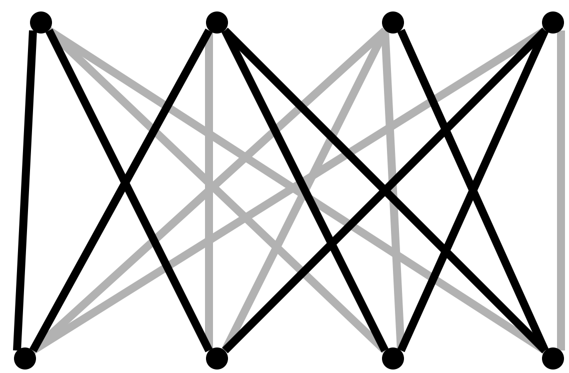
(EASY) (DIFFICULT)

$K_{n,n}$ → G

Solution: **Approximate**

Ideal weights
(for a matching with holes u,v):

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes } u,v\text{)}}$$



Simulated Annealing for Permanent

Unordered state → Ordered state

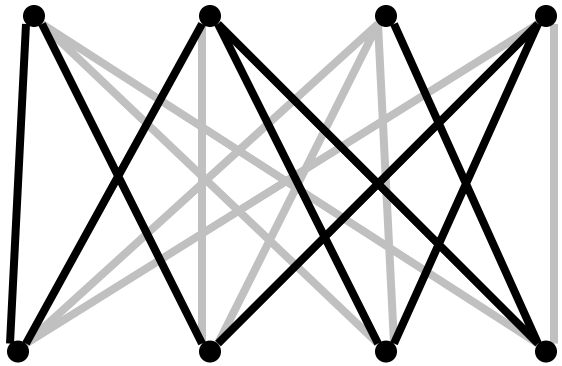
(EASY) (DIFFICULT)

$K_{n,n}$ → G

Solution: **Approximate**

Ideal weights
(for a matching with holes u,v):

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$



Simulated Annealing for Permanent

Unordered state → Ordered state

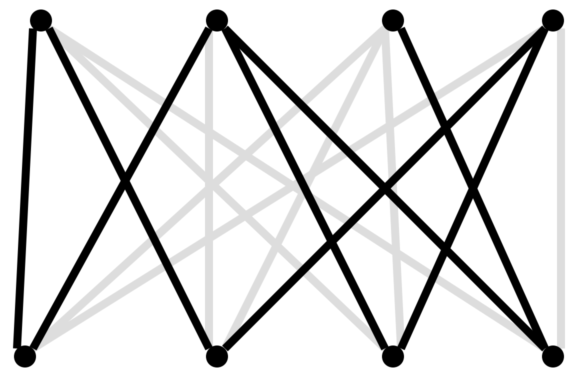
(EASY) (DIFFICULT)

$K_{n,n} \longrightarrow G$

Solution: **Approximate**

Ideal weights
(for a matching with holes u,v):

$$\frac{(\#\ \text{perfects})}{(\#\ \text{nears with holes } u,v)}$$



Simulated Annealing for Permanent

Unordered state → Ordered state

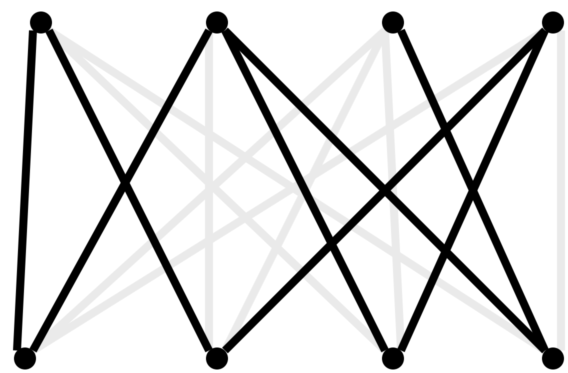
(EASY) (DIFFICULT)

$K_{n,n}$ → G

Solution: **Approximate**

Ideal weights
(for a matching with holes u,v):

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$



Simulated Annealing for Permanent

Unordered state → Ordered state

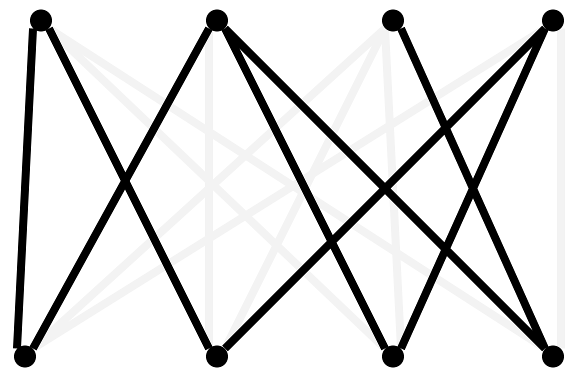
(EASY) (DIFFICULT)

$K_{n,n}$ → G

Solution: **Approximate**

Ideal weights
(for a matching with holes u,v):

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$



Simulated Annealing for Permanent

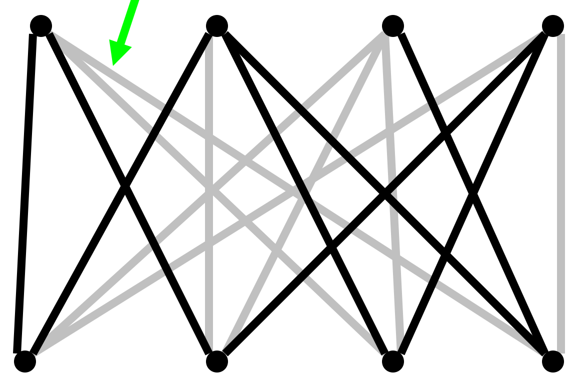
Unordered state → Ordered state

(EASY) (DIFFICULT)

$K_{n,n}$ → G

Solution: **Approximate**

$\boxed{\lambda}$ = 1 ... ~0



Ideal weights

(for a matching with holes u,v):

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$

New Cooling Schedule

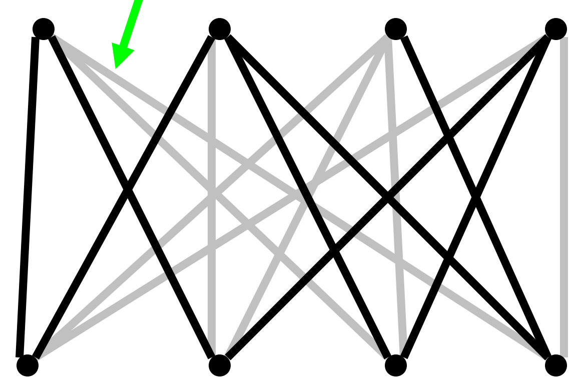
[Bezáková-Štefankovič-
Vazirani-Vigoda SODA '06]

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$

← ratio of polynomials in λ
(don't know coefficients !)

Want: decrease λ so that **polynomials drop by $\leq$ 2 factor**

$\boxed{\lambda}$ = 1 ... ~0



New Cooling Schedule

[Bezáková-Štefankovič-Vazirani-Vigoda SODA '06]

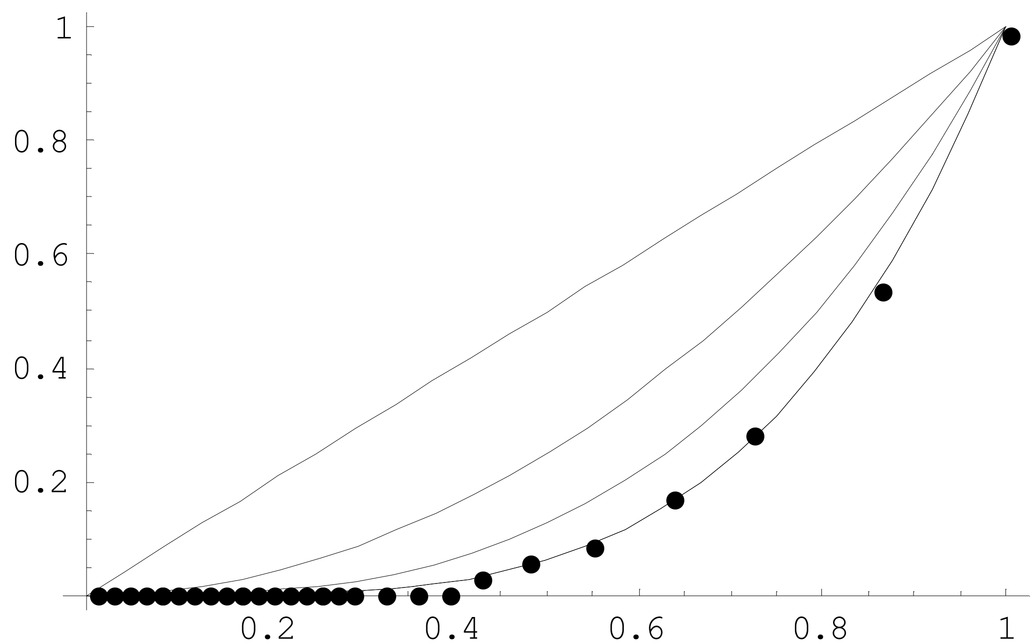
$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$

← ratio of polynomials in λ
(don't know coefficients !)

Want: decrease λ so that **polynomials drop by $\leq$ 2 factor**

Worst case:

$\boxed{\lambda^j, \quad j=1\ldots n}$



New Cooling Schedule

[Bezáková-Štefankovič-Vazirani-Vigoda SODA '06]

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$

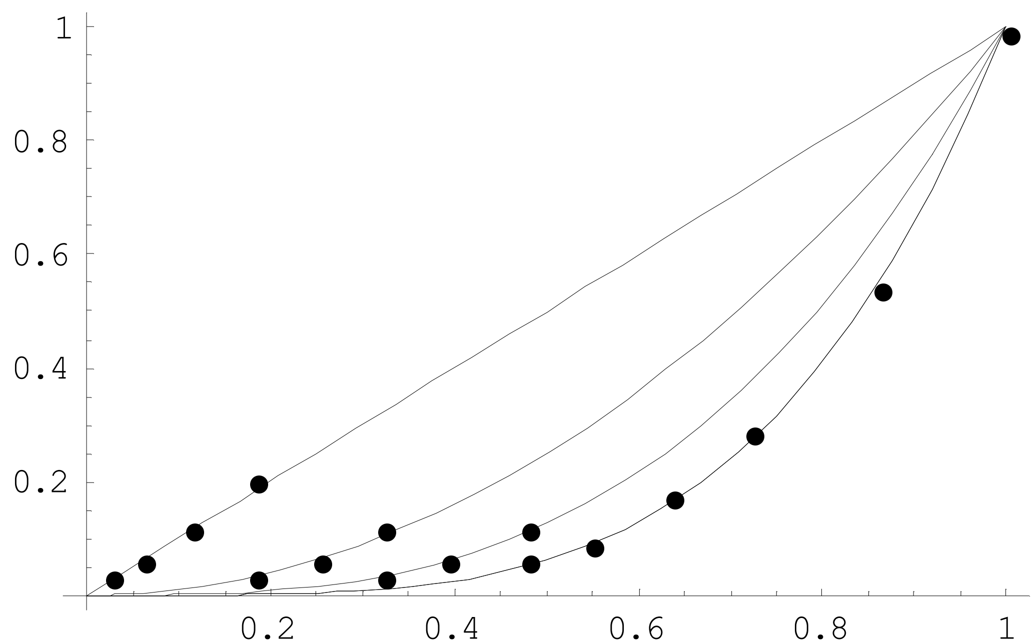
← ratio of polynomials in λ
(don't know coefficients !)

Want: decrease λ so that **polynomials drop by $\leq$ 2 factor**

Worst case:

$\boxed{\lambda^j, \quad j=1 \ldots n}$

Idea: make use of low degrees.



New Cooling Schedule

[Bezáková-Štefankovič-Vazirani-Vigoda SODA '06]

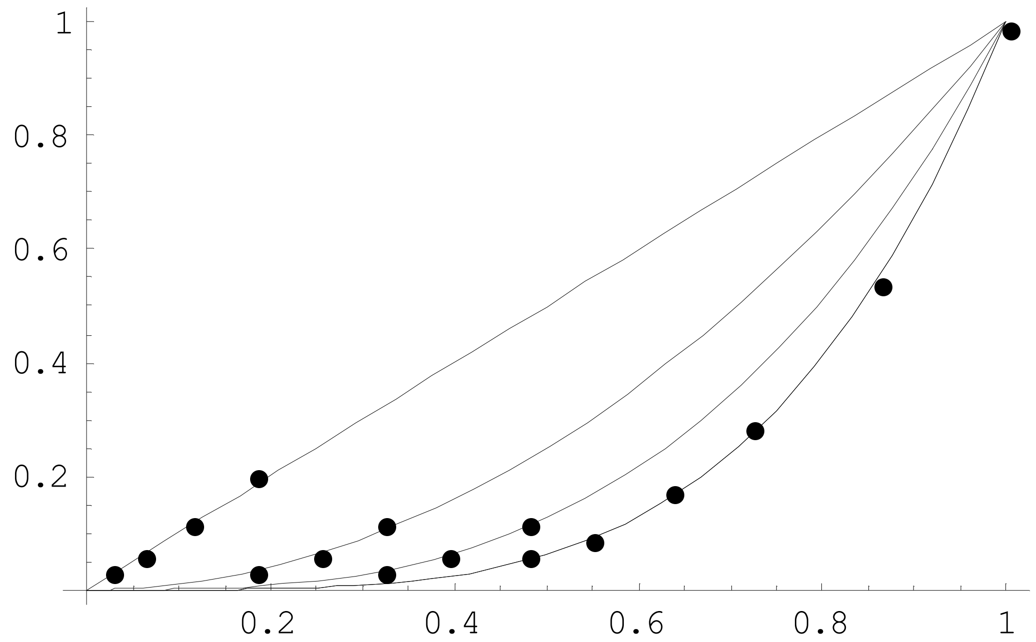
$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$

← ratio of polynomials in λ
(don't know coefficients !)

Want: decrease λ so that **polynomials drop by $\leq$ 2 factor**

Thm [BŠVV '06]:

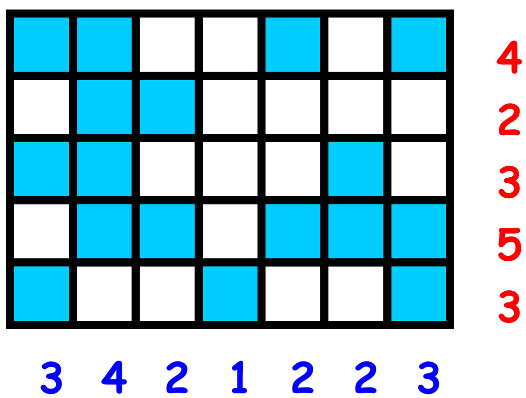
Permanent of nonnegative matrix in time **$O^*(n^7)$**.



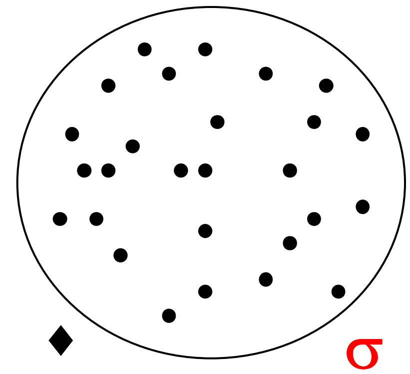
Problems

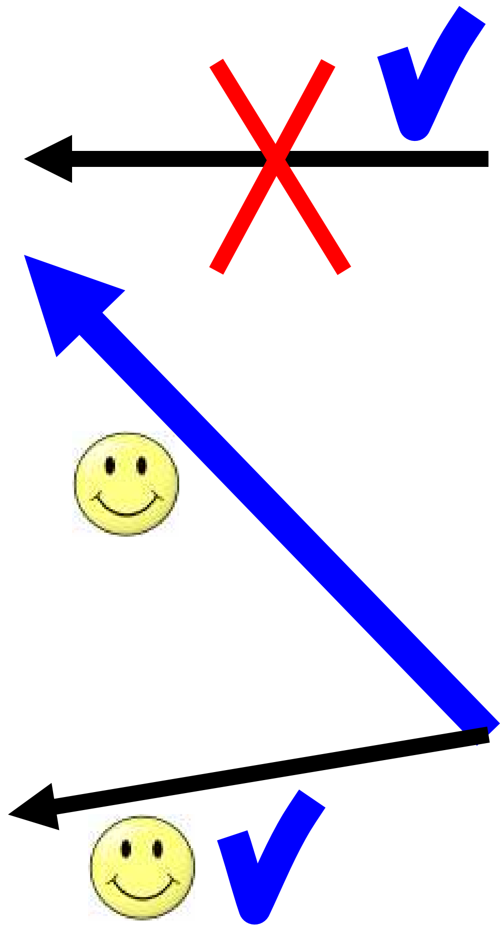
Heuristics

Binary contingency tables

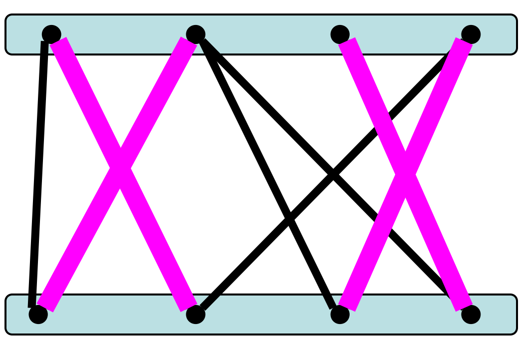


Importance sampling



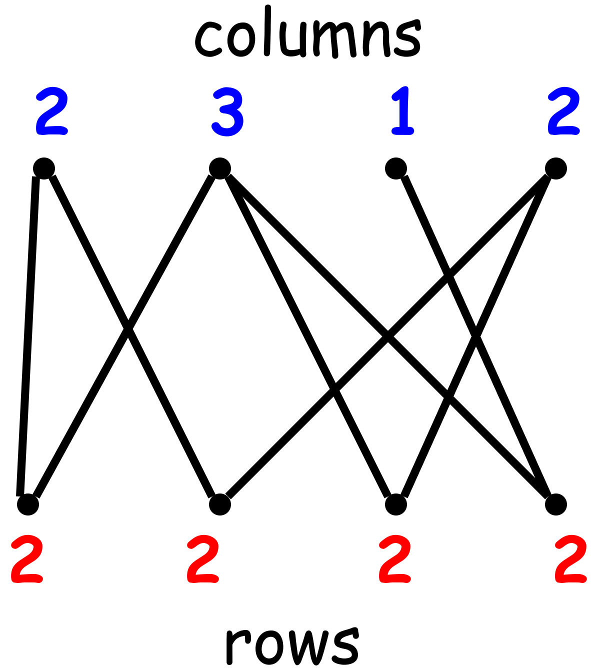


Permanent



Simulated annealing

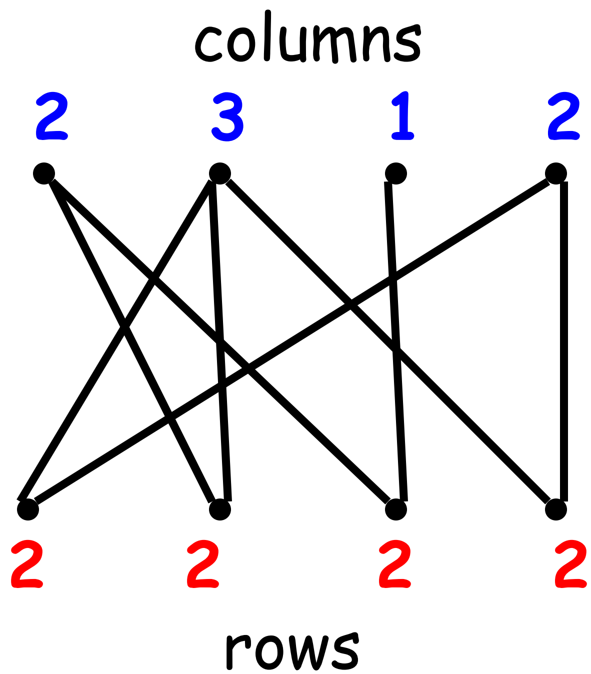
BCT: Bipartite Graphs with Given Degrees



columns

rows					
	1	1	0	0	2
	1	0	0	1	2
	0	1	0	1	2
	0	1	1	0	2
	2	3	1	2	

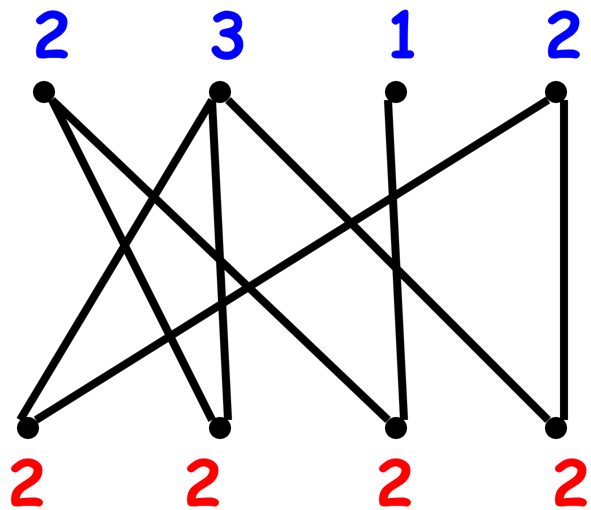
BCT: Bipartite Graphs with Given Degrees



columns

rows					
	0	1	0	1	2
	1	1	0	0	2
	1	0	1	0	2
	0	1	0	1	2
	2	3	1	2	

BCT: Bipartite Graphs with Given Degrees



columns

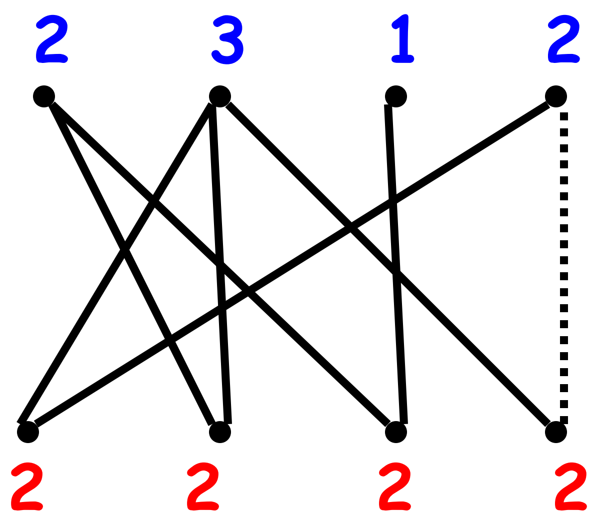
rows					
	0	1	0	1	2
	1	1	0	0	2
	1	0	1	0	2
	0	1	0	1	2
	2	3	1	2	

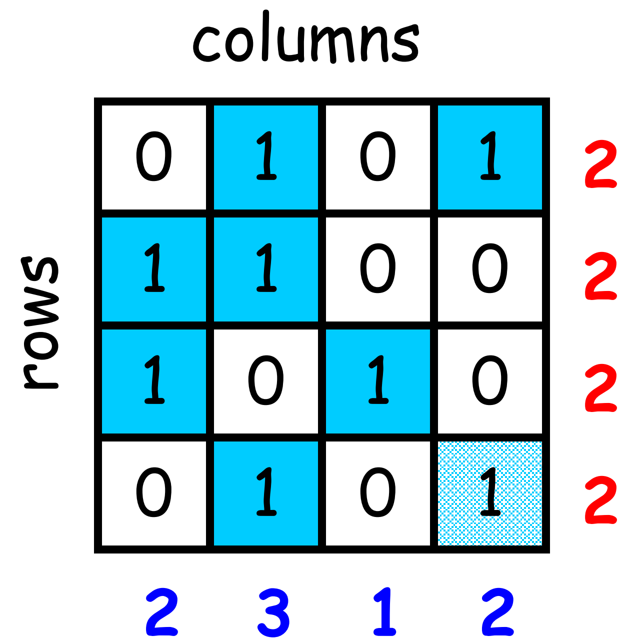
"Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

BCT: Bipartite Graphs with Given Degrees



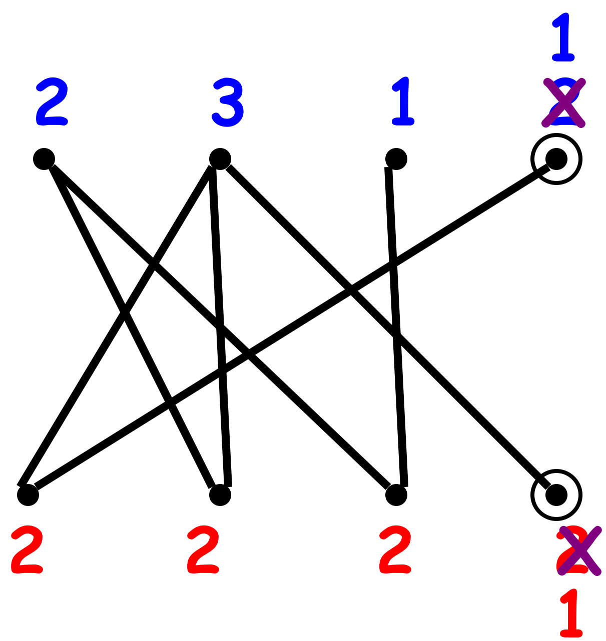


"Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

BCT: Bipartite Graphs with Given Degrees



columns

rows

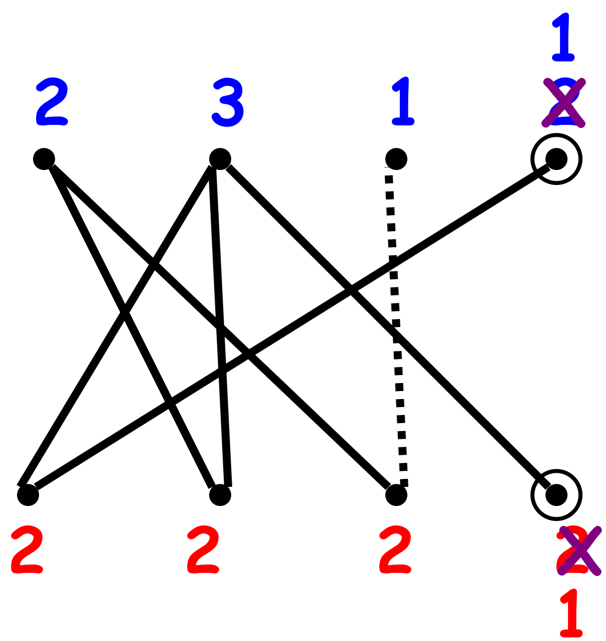
0	1	0	1	2
1	1	0	0	2
1	0	1	0	2
0	1	0	0	~~2~~ 1
2	3	1	~~2~~ 1	

"Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

BCT: Bipartite Graphs with Given Degrees



columns

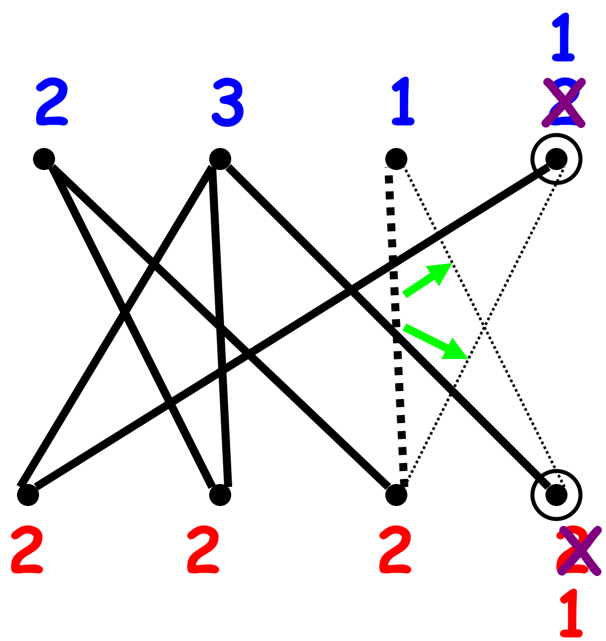
rows					
	0	1	0	1	2
	1	1	0	0	2
	1	0	1	0	2
	0	1	0	0	~~2~~ 1
	2	3	1	~~2~~ 1	

"Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

BCT: Bipartite Graphs with Given Degrees



columns

rows

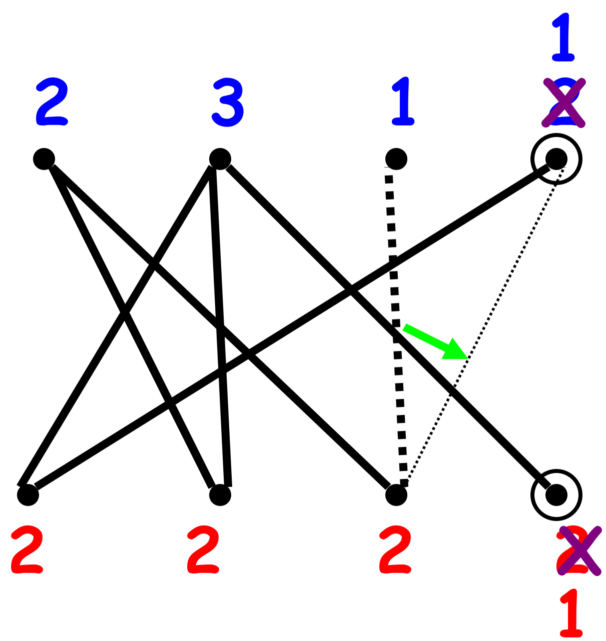
0	1	0	1	2
1	1	0	0	2
1	0	1	0	2
0	1	0	0	~~2~~ 1
2	3	1	~~2~~ 1	

"Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

BCT: Bipartite Graphs with Given Degrees



columns

rows

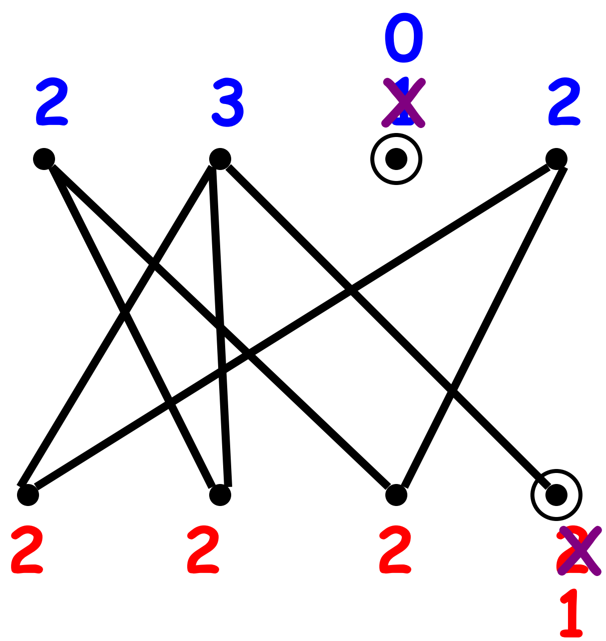
0	1	0	1	2
1	1	0	0	2
1	0	1	0	2
0	1	0	0	~~2~~ 1
2	3	1	~~2~~ 1	

"Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

BCT: Bipartite Graphs with Given Degrees



columns

rows

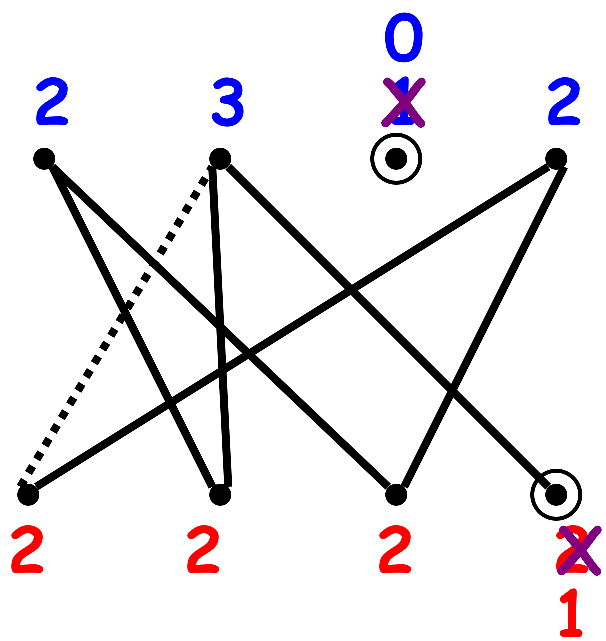
0	1	0	1	2
1	1	0	0	2
1	0	0	1	2
0	1	0	0	~~2~~ 1
2	3	~~1~~ 0	2	

"Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

BCT: Bipartite Graphs with Given Degrees



columns

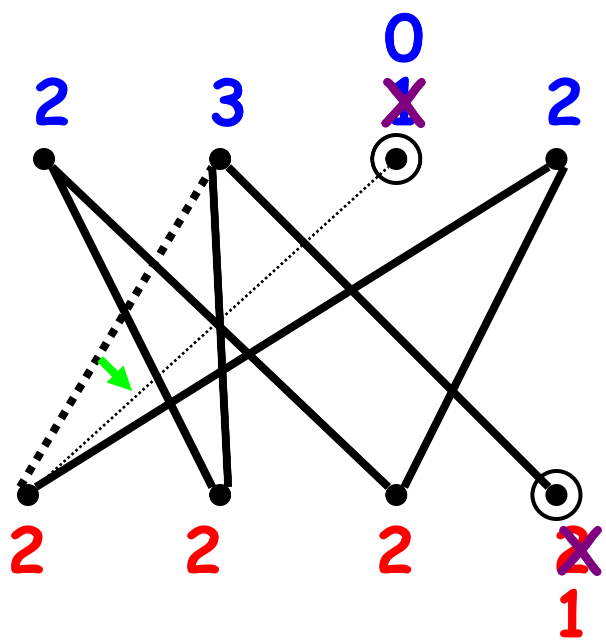
rows				
0	1	0	1	2
1	1	0	0	2
1	0	0	1	2
0	1	0	0	~~2~~ 1
2	3	~~1~~ 0	2	

"Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

BCT: Bipartite Graphs with Given Degrees



columns

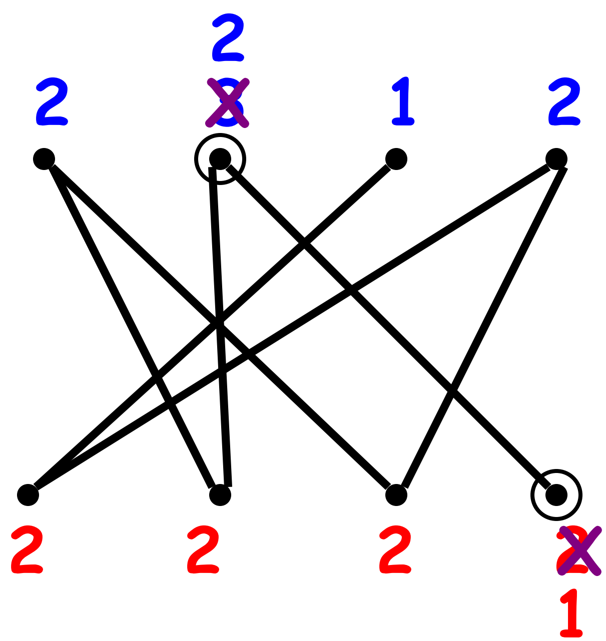
rows					
	0	1	0	1	2
	1	1	0	0	2
	1	0	0	1	2
	0	1	0	0	~~2~~ 1
	2	3	~~1~~ 0	2	

"Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

BCT: Bipartite Graphs with Given Degrees



columns

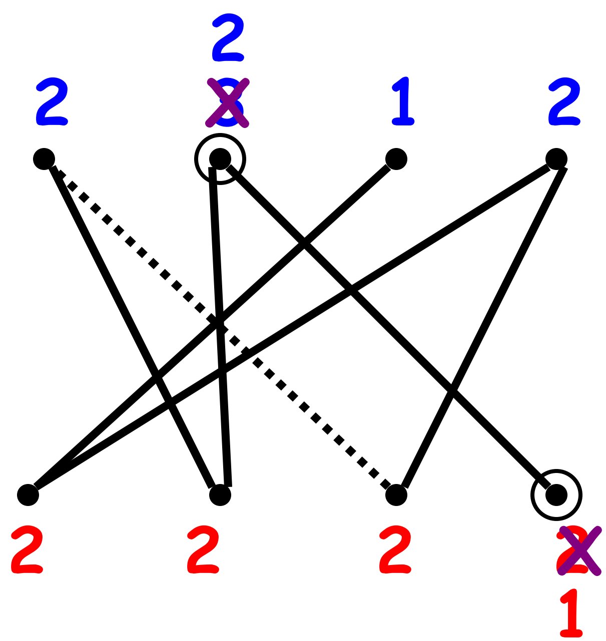
rows	0	1	0	1	2
	1	0	1	0	2
	1	0	0	1	2
	0	1	0	0	~~2~~ 1
	2	~~3~~ 2	1	2	

"Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

BCT: Bipartite Graphs with Given Degrees



columns

rows

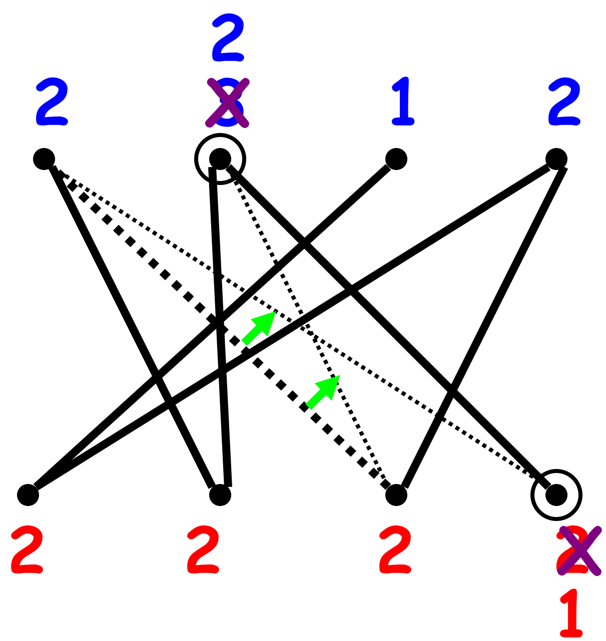
0	1	0	1	2
1	0	1	0	2
1	0	0	1	2
0	1	0	0	~~2~~ 1
2	~~3~~ 2	1	2	

"Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

BCT: Bipartite Graphs with Given Degrees



columns

rows

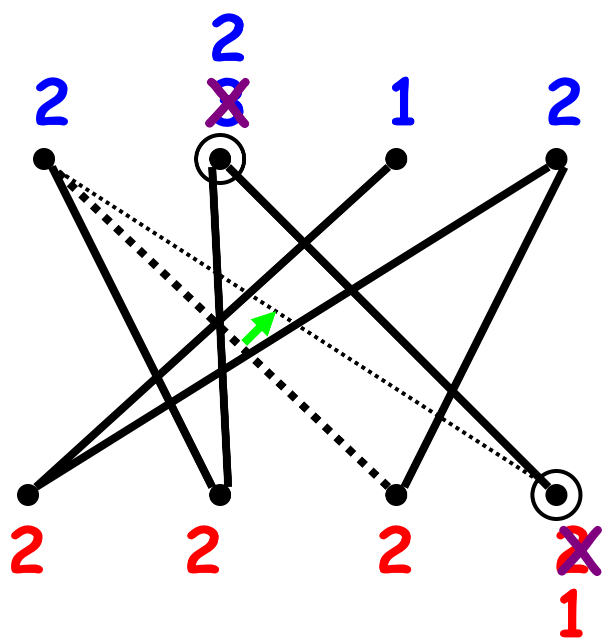
0	1	0	1	2
1	0	1	0	2
1	0	0	1	2
0	1	0	0	~~2~~ 1
2	~~3~~ 2	1	2	

"Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

BCT: Bipartite Graphs with Given Degrees



columns

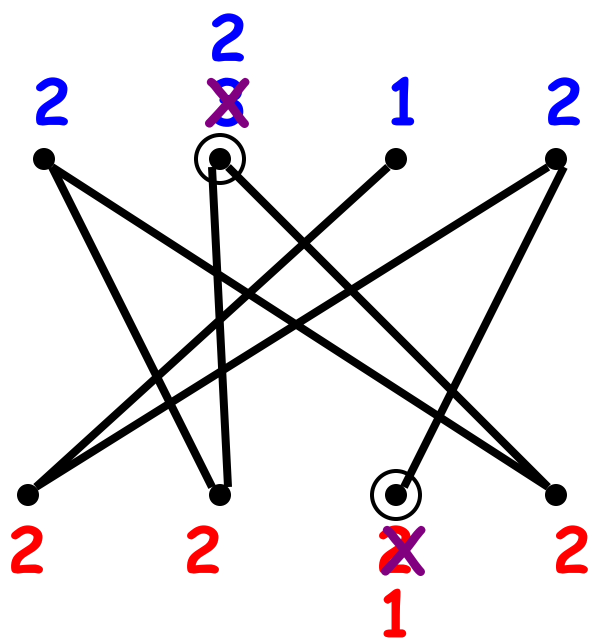
rows					
	0	1	0	1	2
	1	0	1	0	2
	1	0	0	1	2
	0	1	0	0	~~2~~ 1
	2	~~3~~ 2	1	2	

"Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

BCT: Bipartite Graphs with Given Degrees



columns

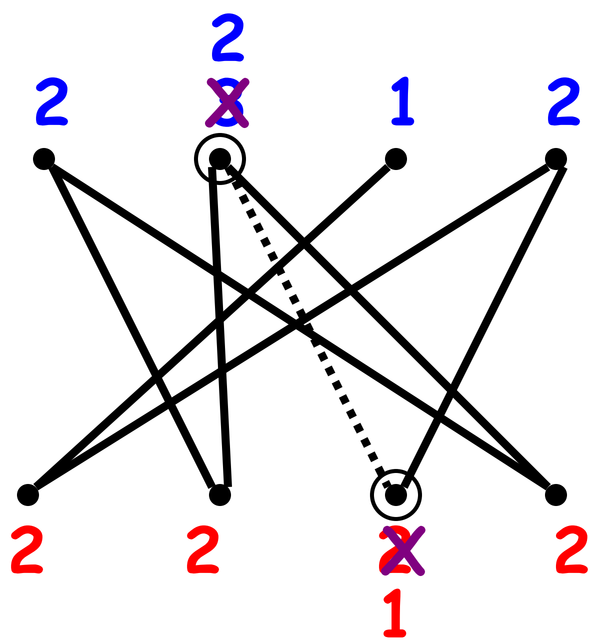
$$\text{rows}\quad\begin{array}{|c|c|c|c|c}
\hline
0 & 1 & 0 & 1 & 2 \\
\hline
1 & 0 & 1 & 0 & 2 \\
\hline
0 & 0 & 0 & 1 & \cancel{2}\ 1 \\
\hline
1 & 1 & 0 & 0 & 2 \\
\hline
\multicolumn{1}{c}{2} & \multicolumn{1}{c}{\cancel{3}\ 2} & \multicolumn{1}{c}{1} & \multicolumn{1}{c}{2} &
\end{array}$$

"Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

BCT: Bipartite Graphs with Given Degrees



columns

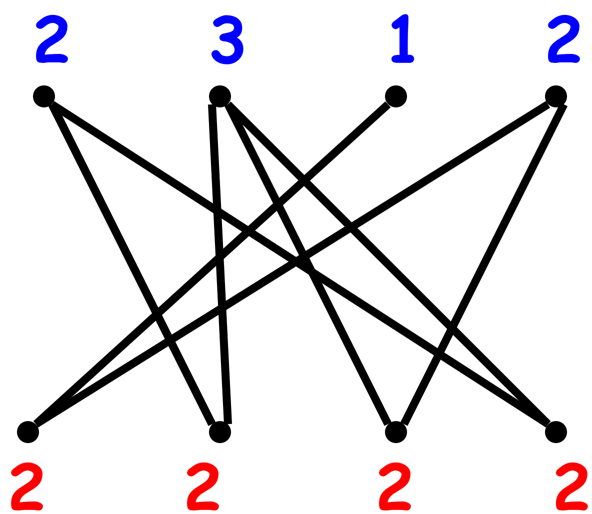
rows					
	0	1	0	1	2
	1	0	1	0	2
	0	0	0	1	~~2~~ 1
	1	1	0	0	2
	2	~~3~~ 2	1	2	

"Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

BCT: Bipartite Graphs with Given Degrees



columns

rows					
	0	1	0	1	2
	1	0	1	0	2
	0	1	0	1	2
	1	1	0	0	2
	2	3	1	2	

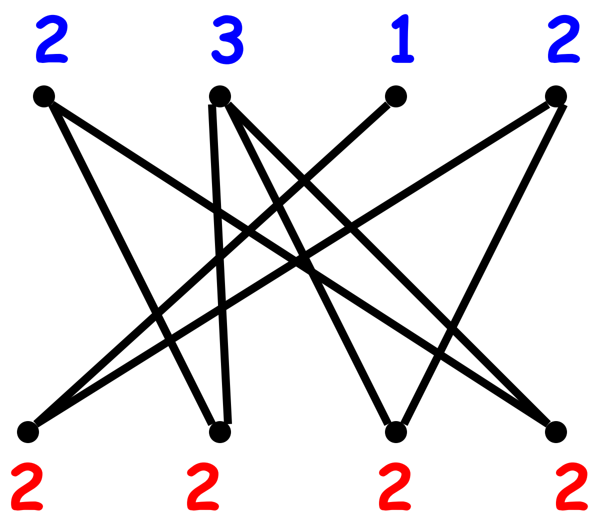
"Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

Simulated Annealing for BCT ?

[Bezáková-Bhatnagar-Vigoda SODA '06]



Ideal weights

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$

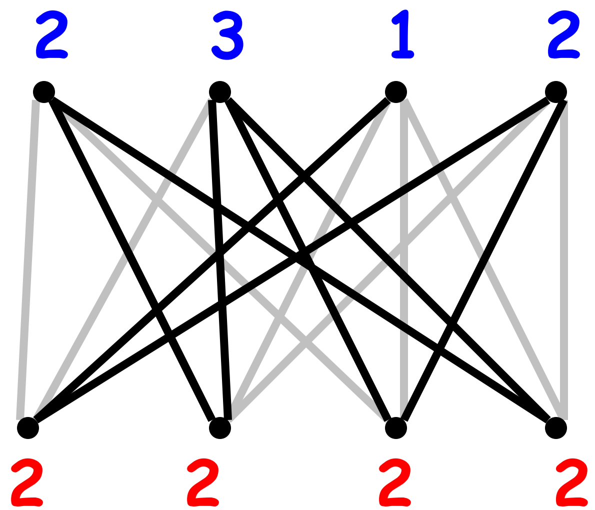
Unordered state → Ordered state

(EASY) (DIFFICULT)

? → 2 3 1 2 / 2 2 2 2

Simulated Annealing for BCT ?

[Bezáková-Bhatnagar-Vigoda SODA '06]



Ideal weights

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$

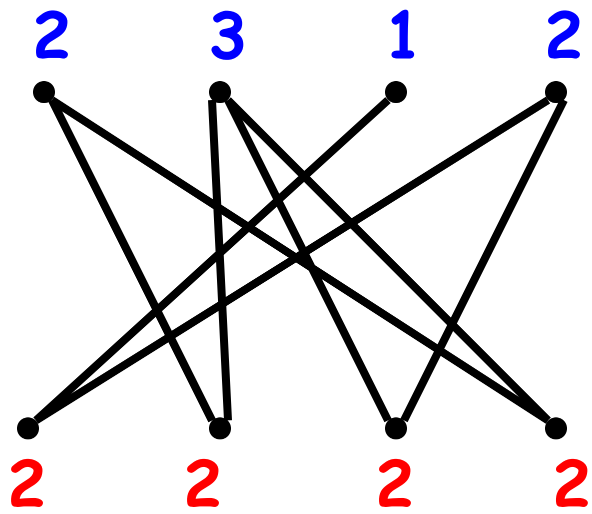
Unordered state → Ordered state

(EASY) (DIFFICULT)

? → 2 3 1 2 / 2 2 2 2 on $K_{n,n}$

Simulated Annealing for BCT ?

[Bezáková-Bhatnagar-Vigoda SODA '06]



Ideal weights

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes } u,v\text{)}}$$

Unordered state → Ordered state

(EASY)

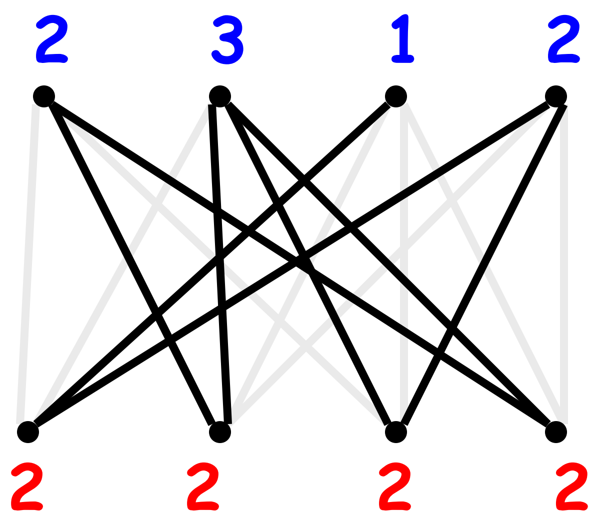
2 3 1 2
2 2 2 2
on G^*

(DIFFICULT)

2 3 1 2
2 2 2 2
on $K_{n,n}$

Simulated Annealing for BCT ?

[Bezáková-Bhatnagar-Vigoda SODA '06]



Thm [BBV '06]:

*G** s.t. between any two vertices exists an "alternating" path of length $\leq$ **5**.

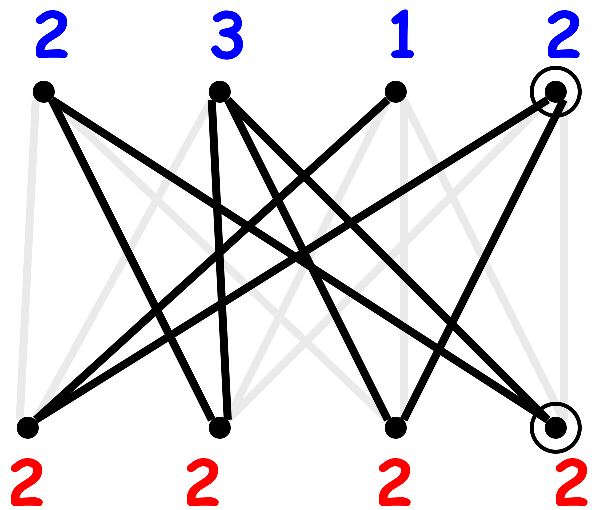
Unordered state → Ordered state

(EASY) (DIFFICULT)

2 3 1 2 2 2 2 2 on *G**	2 3 1 2 2 2 2 2 on $K_{n,n}$

Simulated Annealing for BCT ?

[Bezáková-Bhatnagar-Vigoda SODA '06]



Thm [BBV '06]:

G* s.t. between any two vertices exists an "alternating" path of length $\leq$ **5**.

Unordered state → Ordered state

(EASY)

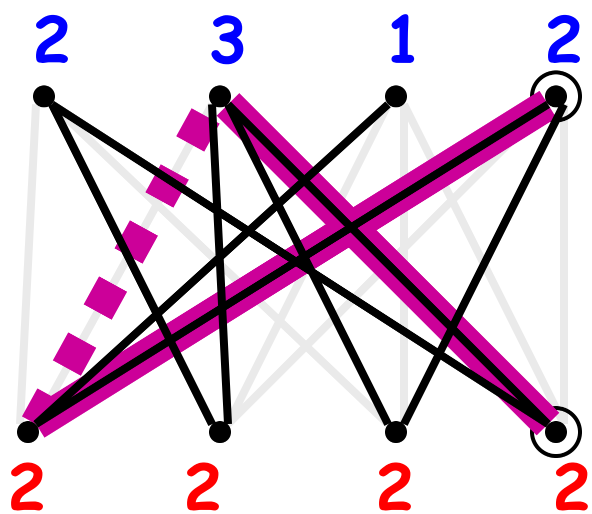
2 3 1 2
2 2 2 2
on ***G****

(DIFFICULT)

2 3 1 2
2 2 2 2
on $K_{n,n}$

Simulated Annealing for BCT ?

[Bezáková-Bhatnagar-Vigoda SODA '06]



Thm [BBV '06]:

G* s.t. between any two vertices exists an "alternating" path of length $\leq$ **5**.

Unordered state → Ordered state

(EASY)

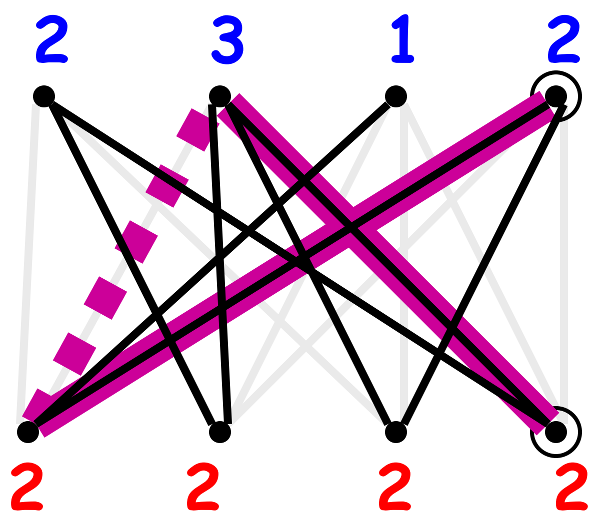
2 3 1 2
2 2 2 2
on ***G****

(DIFFICULT)

2 3 1 2
2 2 2 2
on $K_{n,n}$

Simulated Annealing for BCT ?

[Bezáková-Bhatnagar-Vigoda SODA '06]



Thm [BBV '06]:

G* s.t. between any two vertices exists an "alternating" path of length $\leq$ **5**.

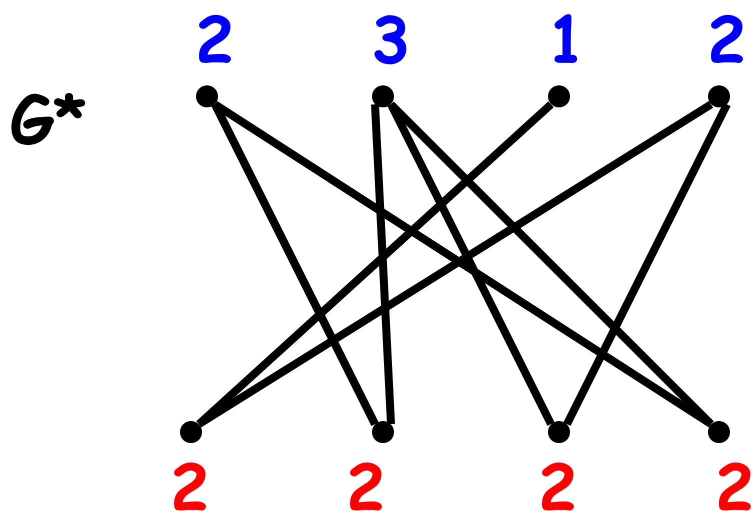
Note: some graphs force linear length paths.

Thm [BBV '06]:

Counting **subgraphs** (of a bipartite graph) **with given degrees** in poly-time.

Simulated Annealing for BCT ?

[Bezáková-Bhatnagar-Vigoda SODA '06]



Thm [BBV '06]:

G* s.t. between any two vertices exists an "alternating" path of length $\leq$ **5**.

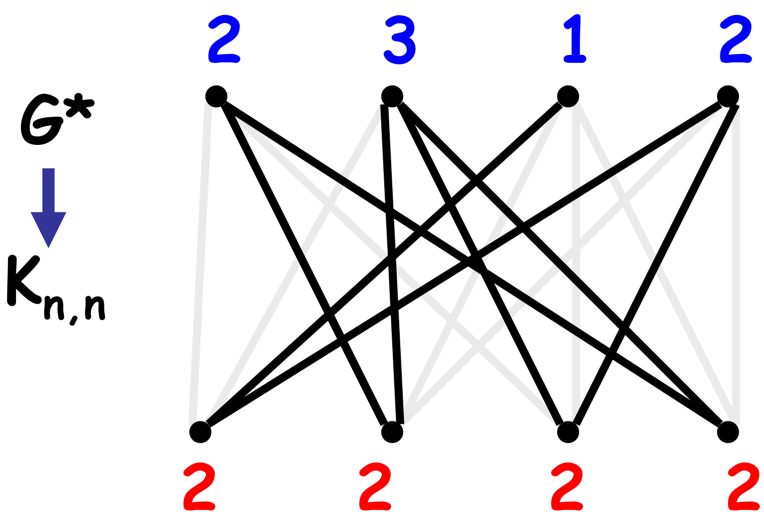
Note: some graphs force linear length paths.

Thm [BBV '06]:

Counting **subgraphs** (of a bipartite graph) **with given degrees** in poly-time.

Simulated Annealing for BCT ?

[Bezáková-Bhatnagar-Vigoda SODA '06]



Thm [BBV '06]:

G* s.t. between any two vertices exists an "alternating" path of length $\leq$ **5**.

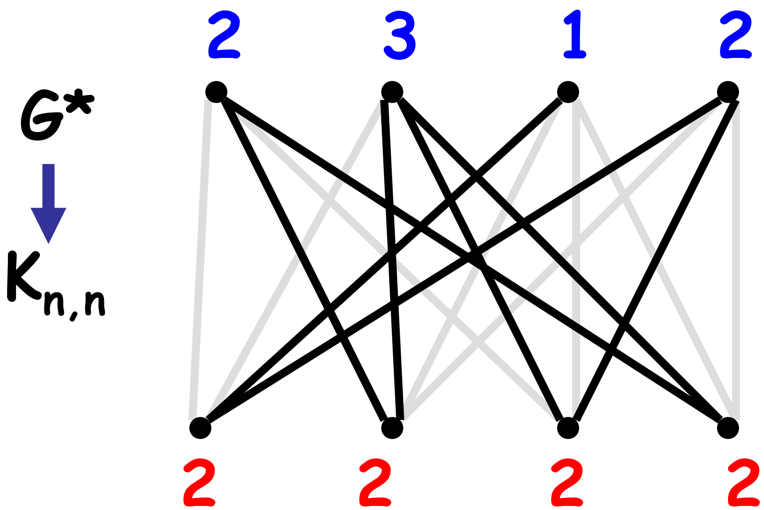
Note: some graphs force linear length paths.

Thm [BBV '06]:

Counting **subgraphs** (of a bipartite graph) **with given degrees** in poly-time.

Simulated Annealing for BCT ?

[Bezáková-Bhatnagar-Vigoda SODA '06]



Thm [BBV '06]:

G* s.t. between any two vertices exists an "alternating" path of length $\leq$ **5**.

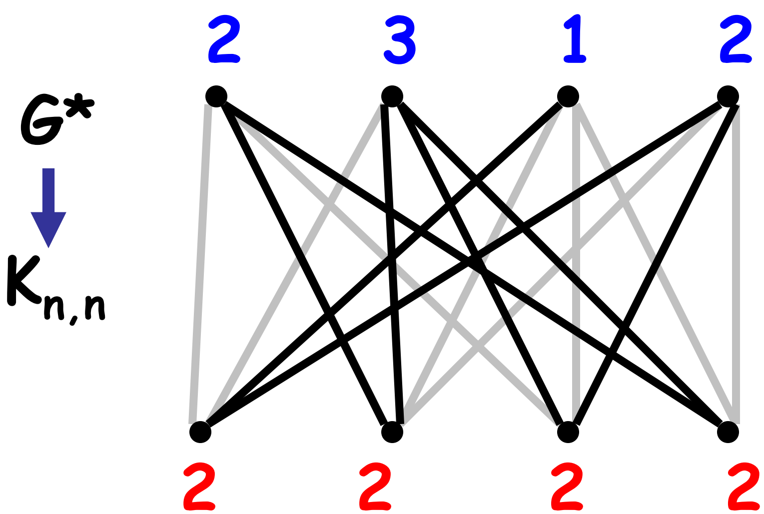
Note: some graphs force linear length paths.

Thm [BBV '06]:

Counting **subgraphs** (of a bipartite graph) **with given degrees** in poly-time.

Simulated Annealing for BCT ?

[Bezáková-Bhatnagar-Vigoda SODA '06]



Thm [BBV '06]:

G* s.t. between any two vertices exists an "alternating" path of length $\leq$ **5**.

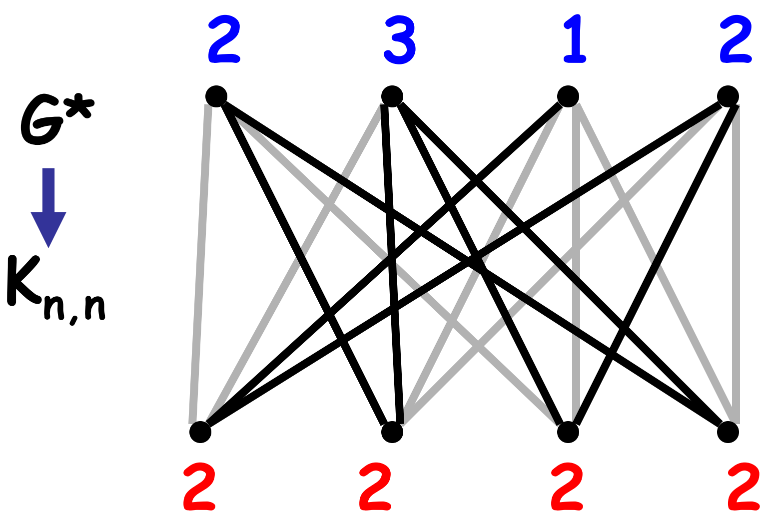
Note: some graphs force linear length paths.

Thm [BBV '06]:

Counting **subgraphs** (of a bipartite graph) **with given degrees** in poly-time.

Simulated Annealing for BCT ?

[Bezáková-Bhatnagar-Vigoda SODA '06]



Thm [BBV '06]:

G* s.t. between any two vertices exists an "alternating" path of length $\leq$ **5**.

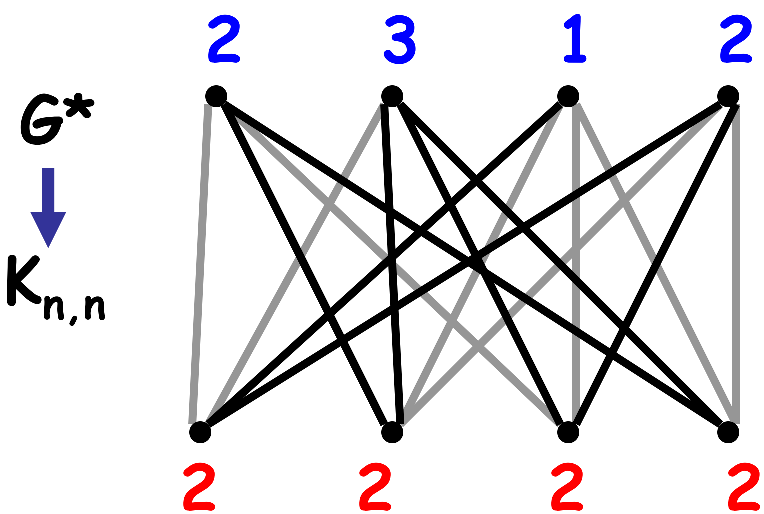
Note: some graphs force linear length paths.

Thm [BBV '06]:

Counting **subgraphs** (of a bipartite graph) **with given degrees** in poly-time.

Simulated Annealing for BCT ?

[Bezáková-Bhatnagar-Vigoda SODA '06]



Thm [BBV '06]:

G* s.t. between any two vertices exists an "alternating" path of length $\leq$ **5**.

Note: some graphs force linear length paths.

Thm [BBV '06]:

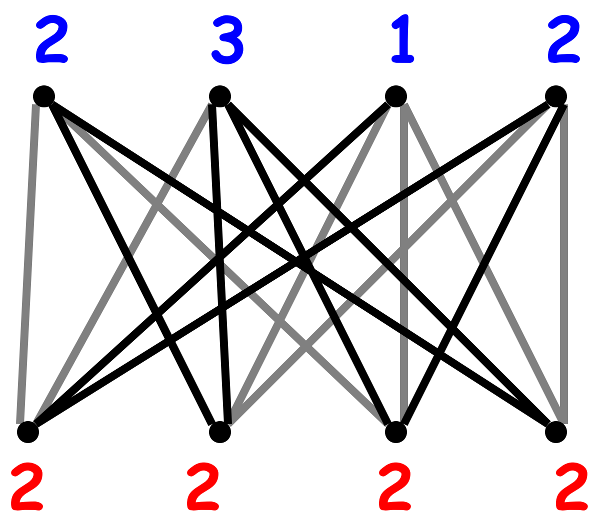
Counting **subgraphs** (of a bipartite graph) **with given degrees** in poly-time.

Simulated Annealing for BCT ?

[Bezáková-
Bhatnagar-
Vigoda
SODA '06]

G^*

$K_{n,n}$



Thm [BBV '06]:

G^* s.t. between any two vertices exists an "alternating" path of length $\leq$ **5**.

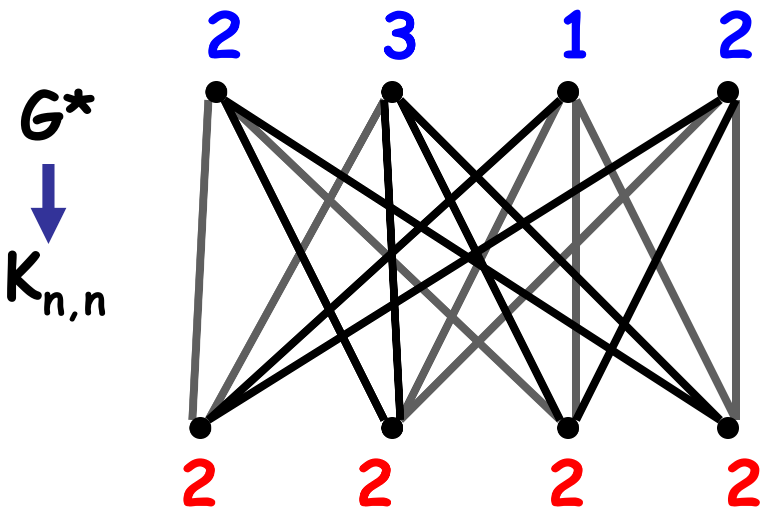
Note: some graphs force linear length paths.

Thm [BBV '06]:

Counting **subgraphs** (of a bipartite graph) **with given degrees** in poly-time.

Simulated Annealing for BCT ?

[Bezáková-Bhatnagar-Vigoda SODA '06]



Thm [BBV '06]:

G^* s.t. between any two vertices exists an "alternating" path of length $\leq$ **5**.

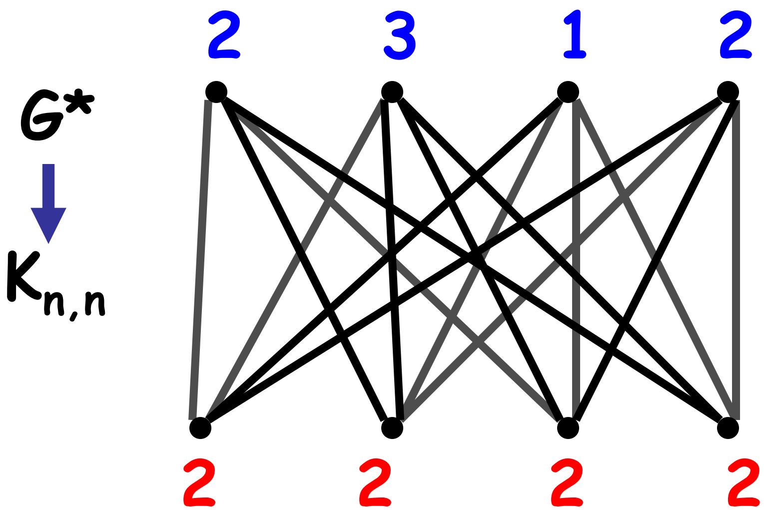
Note: some graphs force linear length paths.

Thm [BBV '06]:

Counting **subgraphs** (of a bipartite graph) **with given degrees** in poly-time.

Simulated Annealing for BCT ?

[Bezáková-Bhatnagar-Vigoda SODA '06]



Thm [BBV '06]:

G* s.t. between any two vertices exists an "alternating" path of length $\leq$ **5**.

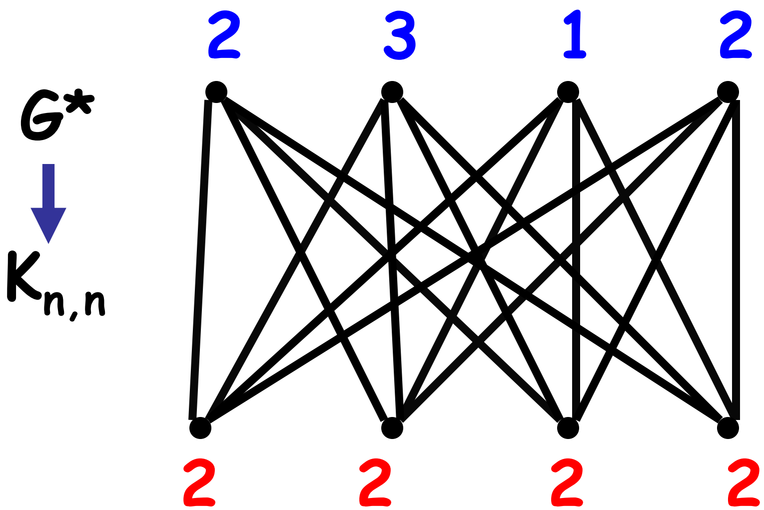
Note: some graphs force linear length paths.

Thm [BBV '06]:

Counting **subgraphs** (of a bipartite graph) **with given degrees** in poly-time.

Simulated Annealing for BCT ?

[Bezáková-Bhatnagar-Vigoda SODA '06]



Thm [BBV '06]:

G* s.t. between any two vertices exists an "alternating" path of length $\leq$ **5**.

Note: some graphs force linear length paths.

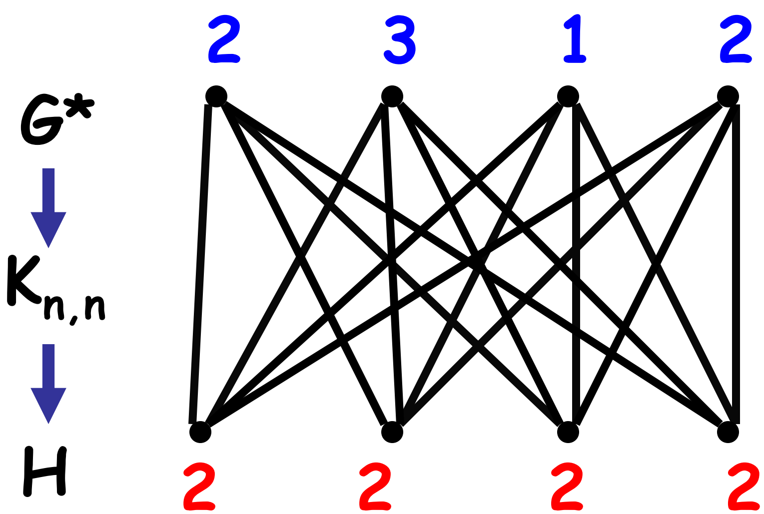
Thm [BBV '06]:

H

Counting **subgraphs** (of a bipartite graph) **with given degrees** in poly-time.

Simulated Annealing for BCT ?

[Bezáková-Bhatnagar-Vigoda SODA '06]



Thm [BBV '06]:

G* s.t. between any two vertices exists an "alternating" path of length $\leq$ **5**.

Note: some graphs force linear length paths.

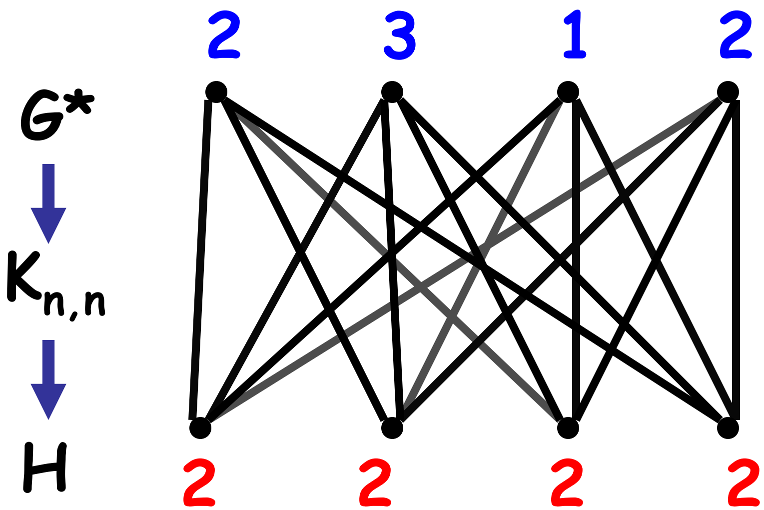
Thm [BBV '06]:

$\boxed{H}$

Counting **subgraphs** (of a bipartite graph) **with given degrees** in poly-time.

Simulated Annealing for BCT ?

[Bezáková-Bhatnagar-Vigoda SODA '06]



Thm [BBV '06]:

G^* s.t. between any two vertices exists an "alternating" path of length $\leq$ **5**.

Note: some graphs force linear length paths.

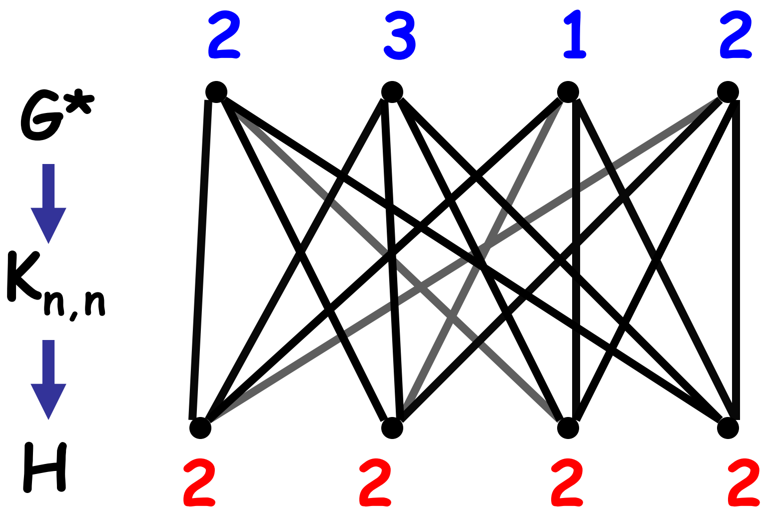
$\boxed{H}$

Thm [BBV '06]:

Counting **subgraphs** (of a bipartite graph) **with given degrees** in poly-time.

Simulated Annealing for BCT ?

[Bezáková-Bhatnagar-Vigoda SODA '06]



Thm [BBV '06]:

G* s.t. between any two vertices exists an "alternating" path of length $\leq$ **5**.

Note: some graphs force linear length paths.

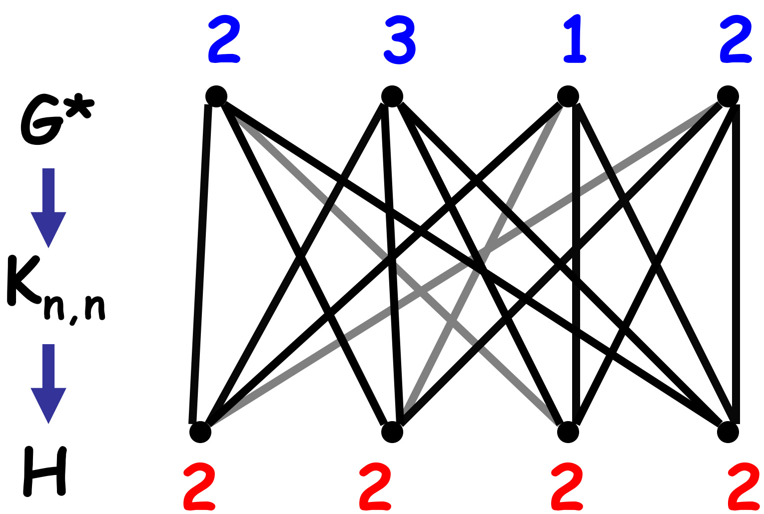
Thm [BBV '06]:

$\boxed{H}$

Counting **subgraphs** (of a bipartite graph) **with given degrees** in poly-time.

Simulated Annealing for BCT ?

[Bezáková-
Bhatnagar-
Vigoda
SODA '06]



Thm [BBV '06]:

G* s.t. between any two vertices exists an "alternating" path of length $\leq$ **5**.

Note: some graphs force linear length paths.

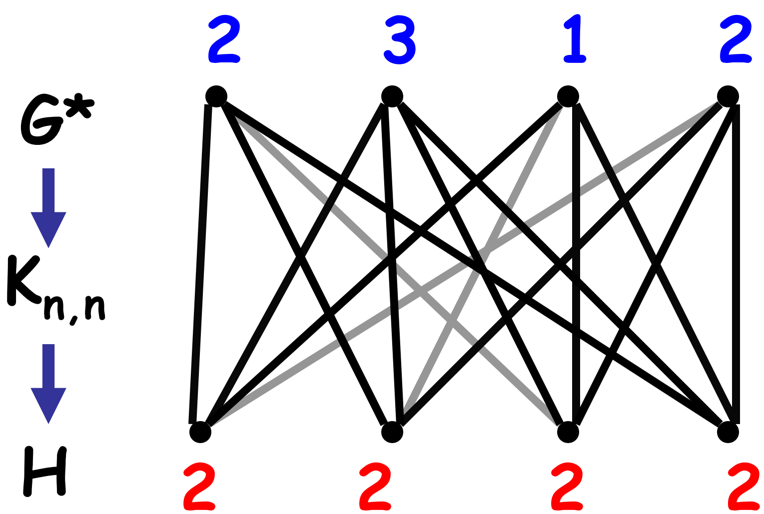
Thm [BBV '06]:

$\boxed{H}$

Counting **subgraphs** (of a bipartite graph) **with given degrees** in poly-time.

Simulated Annealing for BCT ?

[Bezáková-Bhatnagar-Vigoda SODA '06]



Thm [BBV '06]:

G* s.t. between any two vertices exists an "alternating" path of length $\leq$ **5**.

Note: some graphs force linear length paths.

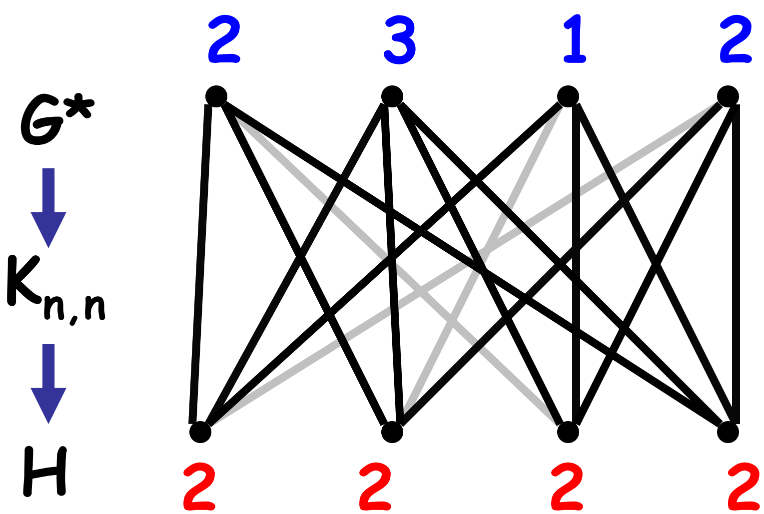
Thm [BBV '06]:

$\boxed{H}$

Counting **subgraphs** (of a bipartite graph) **with given degrees** in poly-time.

Simulated Annealing for BCT ?

[Bezáková-Bhatnagar-Vigoda SODA '06]



Thm [BBV '06]:

G* s.t. between any two vertices exists an "alternating" path of length $\leq$ **5**.

Note: some graphs force linear length paths.

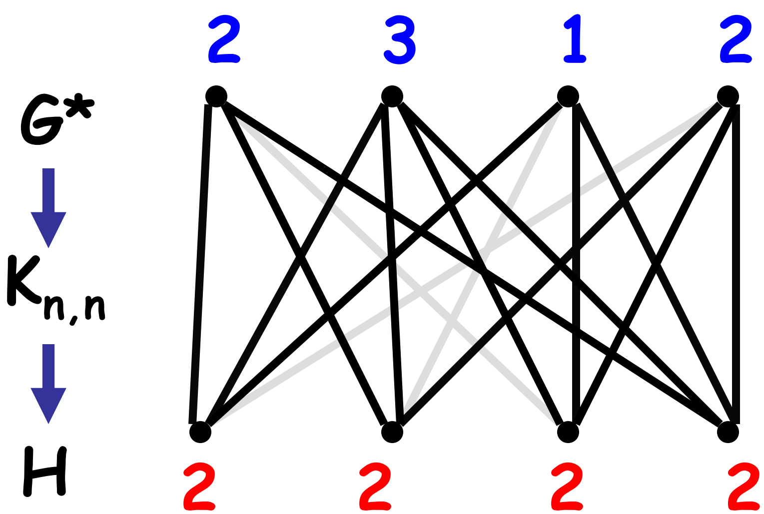
Thm [BBV '06]:

$\boxed{H}$

Counting **subgraphs** (of a bipartite graph) **with given degrees** in poly-time.

Simulated Annealing for BCT ?

[Bezáková-Bhatnagar-Vigoda SODA '06]



Thm [BBV '06]:

G* s.t. between any two vertices exists an "alternating" path of length $\leq$ **5**.

Note: some graphs force linear length paths.

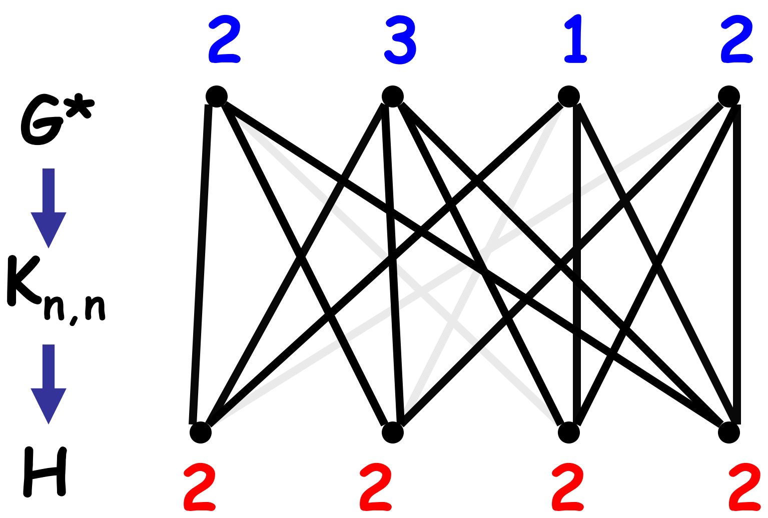
Thm [BBV '06]:

$\boxed{H}$

Counting **subgraphs** (of a bipartite graph) **with given degrees** in poly-time.

Simulated Annealing for BCT ?

[Bezáková-
Bhatnagar-
Vigoda
SODA '06]



Thm [BBV '06]:

G* s.t. between any two vertices exists an "alternating" path of length $\leq$ **5**.

Note: some graphs force linear length paths.

Thm [BBV '06]:

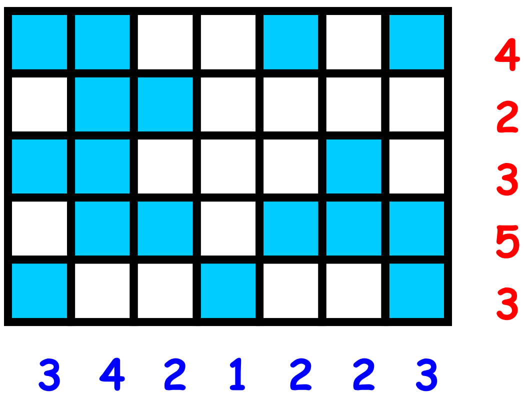
$\boxed{H}$

Counting **subgraphs** (of a bipartite graph) **with given degrees** in poly-time.

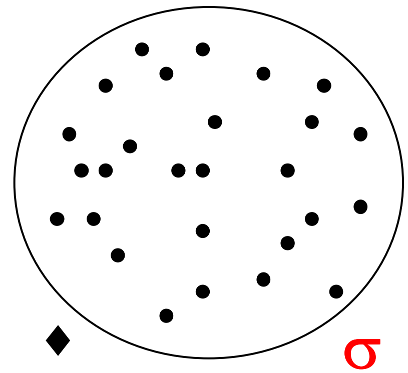
Problems

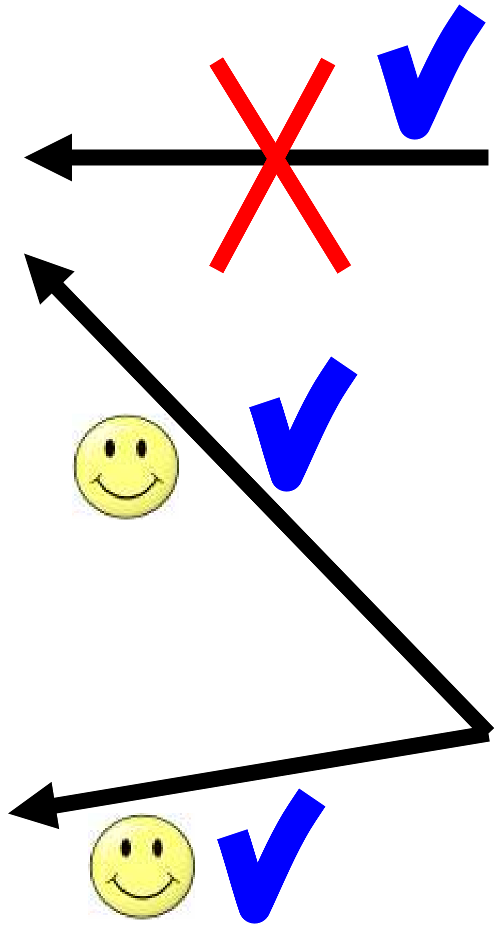
Heuristics

Binary contingency tables



Importance sampling





Simulated annealing

Permanent