Markov Chain Mixing Times And Applications

Ivona Bezáková

(Rochester Institute of Technology)

Nayantara Bhatnagar

(University of Delaware)

Simons Institute for the Theory of Computing
Counting Complexity and Phase Transitions Bootcamp

January 25th, 2016

Introduction to Markov Chains

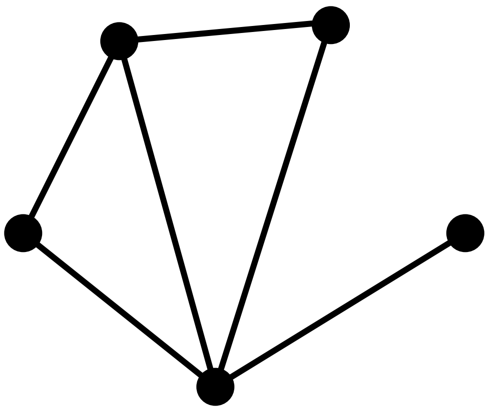
A (discrete) **Markov chain** is a random process that

- has a set of states Ω
- in one step moves from the current state to a random "neighboring" state
- the distribution for the move does not depend on previously visited states

Example:

A random walk on a graph

1. Start at a vertex
2. Randomly choose a neighbor and move there
3. Repeat step 2.



Introduction to Markov Chains

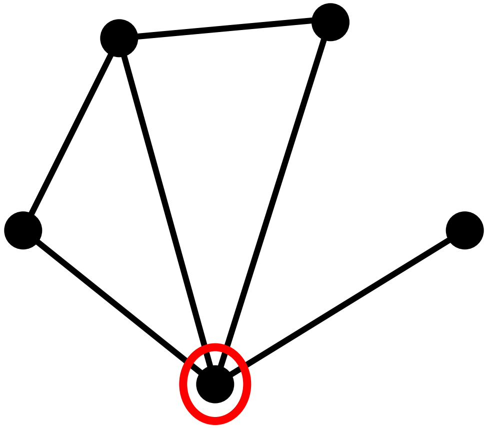
A (discrete) **Markov chain** is a random process that

- has a set of states Ω
- in one step moves from the current state to a random "neighboring" state
- the distribution for the move does not depend on previously visited states

Example:

A random walk on a graph

1. Start at a vertex
2. Randomly choose a neighbor and move there
3. Repeat step 2.



Introduction to Markov Chains

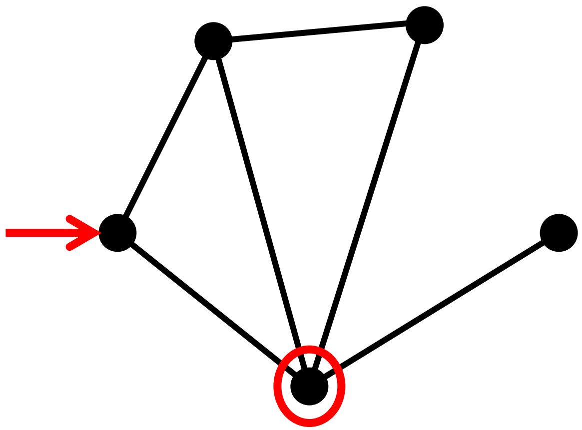
A (discrete) **Markov chain** is a random process that

- has a set of states Ω
- in one step moves from the current state to a random "neighboring" state
- the distribution for the move does not depend on previously visited states

Example:

A random walk on a graph

1. Start at a vertex
2. Randomly choose a neighbor and move there
3. Repeat step 2.



Introduction to Markov Chains

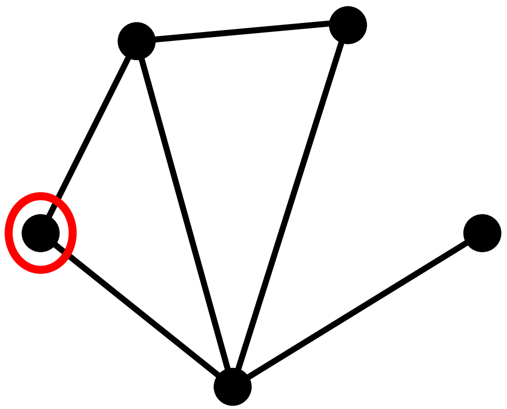
A (discrete) **Markov chain** is a random process that

- has a set of states Ω
- in one step moves from the current state to a random "neighboring" state
- the distribution for the move does not depend on previously visited states

Example:

A random walk on a graph

1. Start at a vertex
2. Randomly choose a neighbor and move there
3. Repeat step 2.



Introduction to Markov Chains

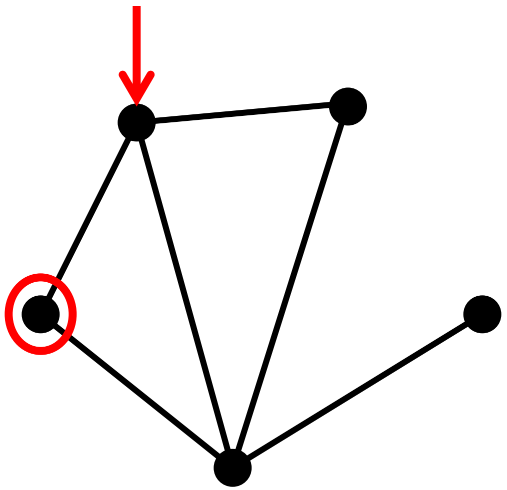
A (discrete) **Markov chain** is a random process that

- has a set of states Ω
- in one step moves from the current state to a random "neighboring" state
- the distribution for the move does not depend on previously visited states

Example:

A random walk on a graph

1. Start at a vertex
2. Randomly choose a neighbor and move there
3. Repeat step 2.



Introduction to Markov Chains

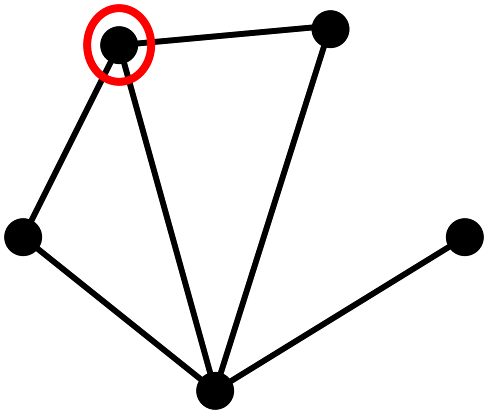
A (discrete) **Markov chain** is a random process that

- has a set of states Ω
- in one step moves from the current state to a random "neighboring" state
- the distribution for the move does not depend on previously visited states

Example:

A random walk on a graph

1. Start at a vertex
2. Randomly choose a neighbor and move there
3. Repeat step 2.



Introduction to Markov Chains

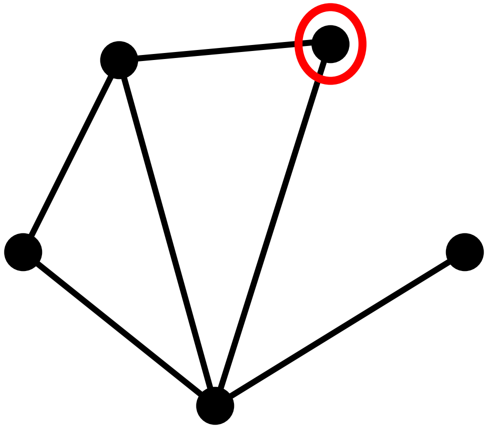
A (discrete) **Markov chain** is a random process that

- has a set of states Ω
- in one step moves from the current state to a random "neighboring" state
- the distribution for the move does not depend on previously visited states

Example:

A random walk on a graph

1. Start at a vertex
2. Randomly choose a neighbor and move there
3. Repeat step 2.



Introduction to Markov Chains

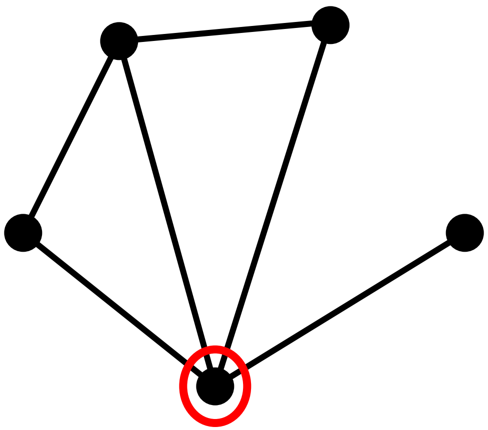
A (discrete) **Markov chain** is a random process that

- has a set of states Ω
- in one step moves from the current state to a random "neighboring" state
- the distribution for the move does not depend on previously visited states

Example:

A random walk on a graph

1. Start at a vertex
2. Randomly choose a neighbor and move there
3. Repeat step 2.



Introduction to Markov Chains

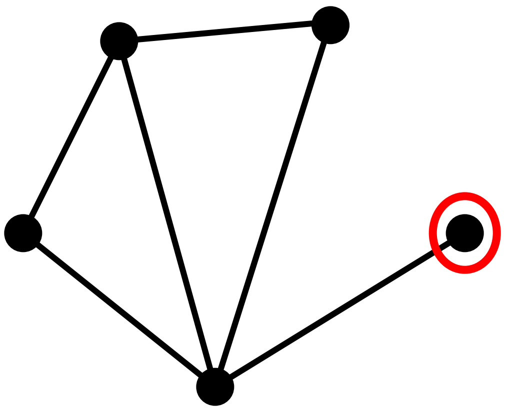
A (discrete) **Markov chain** is a random process that

- has a set of states Ω
- in one step moves from the current state to a random "neighboring" state
- the distribution for the move does not depend on previously visited states

Example:

A random walk on a graph

1. Start at a vertex
2. Randomly choose a neighbor and move there
3. Repeat step 2.



Introduction to Markov Chains

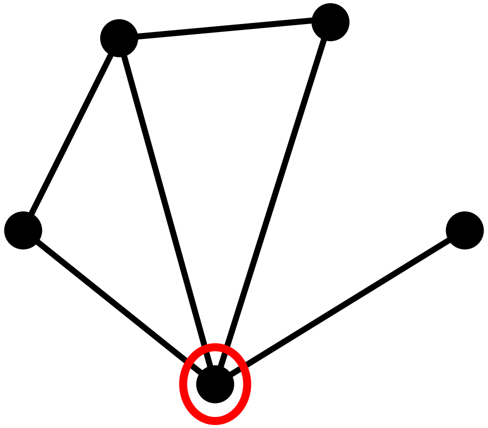
A (discrete) **Markov chain** is a random process that

- has a set of states Ω
- in one step moves from the current state to a random "neighboring" state
- the distribution for the move does not depend on previously visited states

Example:

A random walk on a graph

1. Start at a vertex
2. Randomly choose a neighbor and move there
3. Repeat step 2.



Introduction to Markov Chains

A (discrete) **<u>Markov chain</u>** is a random process that

• has a set of states Ω

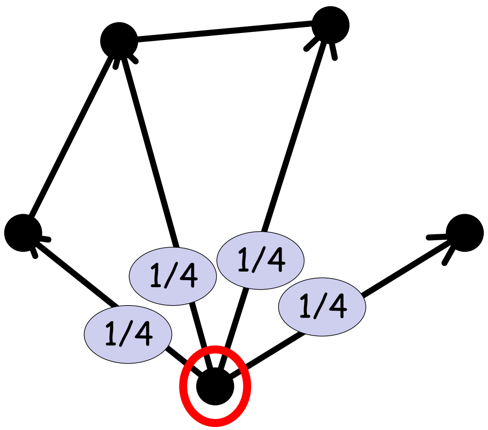
• in one step moves from the current state to a random "neighboring" state

• the distribution for the move does not depend on previously visited states

<u>Example:</u>

A random walk on a graph

1. Start at a vertex
2. Randomly choose a neighbor and move there
3. Repeat step 2.



Introduction to Markov Chains

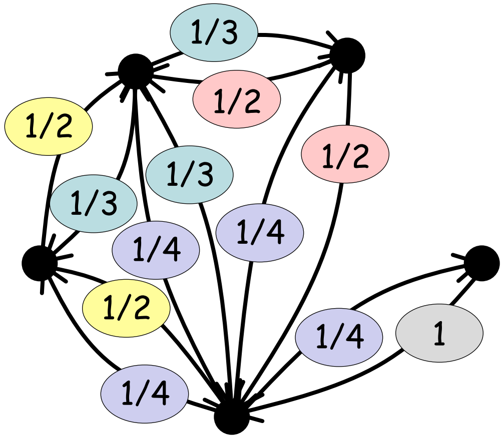
A (discrete) **Markov chain** is a random process that

- has a set of states Ω
- in one step moves from the current state to a random "neighboring" state
- the distribution for the move does not depend on previously visited states

Example:

A random walk on a graph

1. Start at a vertex
2. Randomly choose a neighbor and move there
3. Repeat step 2.



Introduction to Markov Chains

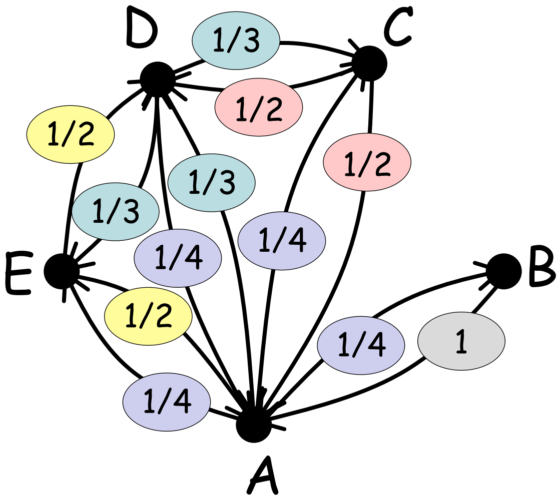
A (discrete) **Markov chain** is a random process that

- has a set of states Ω
- in one step moves from the current state to a random "neighboring" state
- the distribution for the move does not depend on previously visited states

Example:

Transition matrix for this MC:

	A	B	C	D	E
A	0	1/4	1/4	1/4	1/4
B	1	0	0	0	0
C	1/2	0	0	1/2	0
D	1/3	0	1/3	0	1/3
E	1/2	0	0	1/2	0



Introduction to Markov Chains

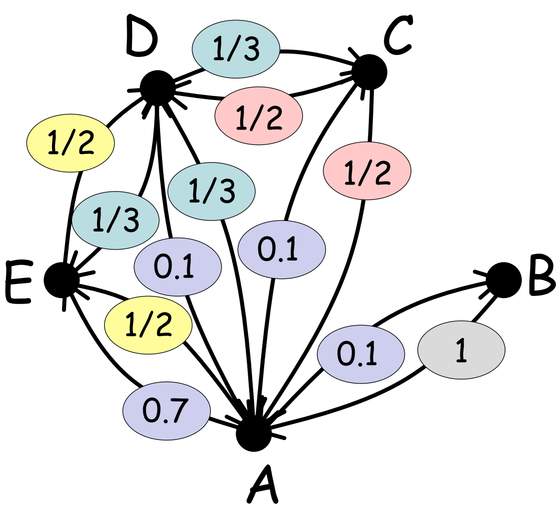
A (discrete) **Markov chain** is a random process that

- has a set of states Ω
- in one step moves from the current state to a random "neighboring" state
- the distribution for the move does not depend on previously visited states

Example:

Transition matrix for another MC:

	A	B	C	D	E
A	0	0.1	0.1	0.1	0.7
B	1	0	0	0	0
C	1/2	0	0	1/2	0
D	1/3	0	1/3	0	1/3
E	1/2	0	0	1/2	0



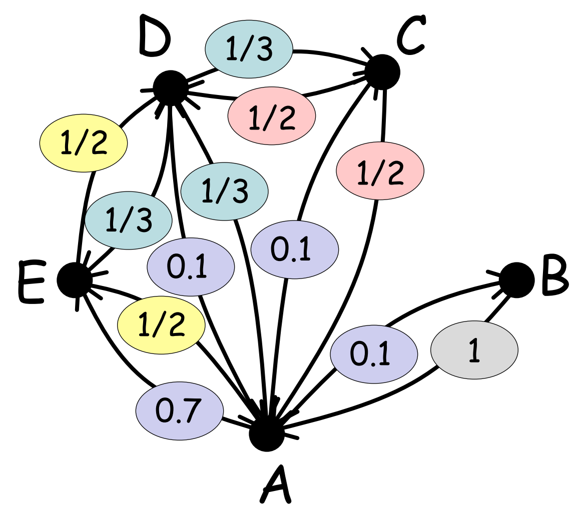
Introduction to Markov Chains

Def: A (discrete) **Markov chain** M is a pair (Ω, P) where P is an $|\Omega| x |\Omega|$ matrix where each of its rows is a distribution. Ω is the state space and P is the transition matrix.

Example:

Transition matrix for another MC:

	A	B	C	D	E
A	0	0.1	0.1	0.1	0.7
B	1	0	0	0	0
C	1/2	0	0	1/2	0
D	1/3	0	1/3	0	1/3
E	1/2	0	0	1/2	0



Introduction to Markov Chains

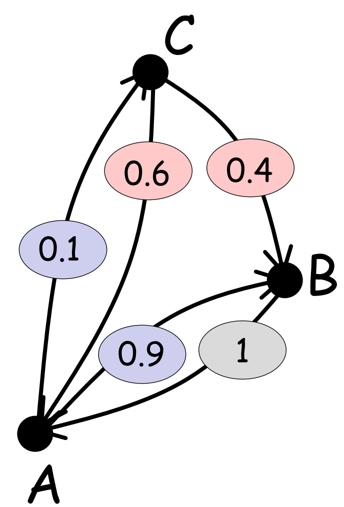
Def: A (discrete) **Markov chain** M is a pair (Ω, P) where P is an $|\Omega| \times |\Omega|$ matrix where each of its rows is a distribution. Ω is the state space and P is the transition matrix.

A distribution π on Ω is stationary if $\pi P = \pi$.

Example:

A simple MC:

	A	B	C
A	0	0.9	0.1
B	1	0	0
C	0.6	0.4	0



Introduction to Markov Chains

Def: A (discrete) **Markov chain** M is a pair (Ω, P) where P is an $|\Omega| \times |\Omega|$ matrix where each of its rows is a distribution. Ω is the state space and P is the transition matrix.

A distribution π on Ω is stationary if $\pi P = \pi$.

Example:

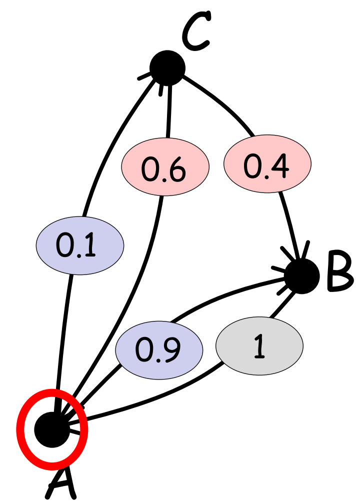
A simple MC:

P=

	A	B	C
A	0	0.9	0.1
B	1	0	0
C	0.6	0.4	0

Start in A:

- start distribution: $\sigma = (1,0,0)$



Introduction to Markov Chains

Def: A (discrete) **Markov chain** M is a pair (Ω, P) where P is an $|\Omega| \times |\Omega|$ matrix where each of its rows is a distribution. Ω is the state space and P is the transition matrix.

A distribution π on Ω is stationary if $\pi P = \pi$.

Example:

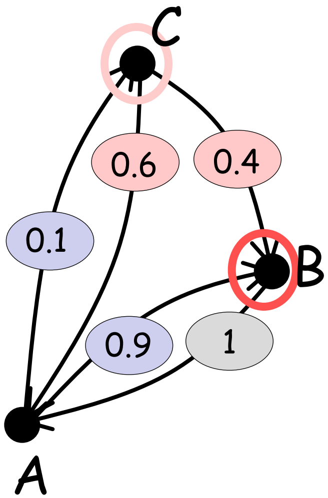
A simple MC:

P=

	A	B	C
A	0	0.9	0.1
B	1	0	0
C	0.6	0.4	0

Start in A:

- start distribution: $\sigma = (1,0,0)$
- after one step: $\sigma P = (0,0.9,0.1)$



Introduction to Markov Chains

Def: A (discrete) **Markov chain** M is a pair (Ω, P) where P is an $|\Omega| \times |\Omega|$ matrix where each of its rows is a distribution. Ω is the state space and P is the transition matrix.

A distribution π on Ω is stationary if $\pi P = \pi$.

Example:

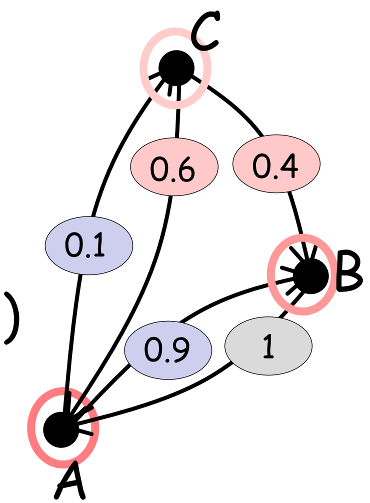
A simple MC:

P=

	A	B	C
A	0	0.9	0.1
B	1	0	0
C	0.6	0.4	0

Start in A:

- start distribution: $\sigma = (1,0,0)$
- after one step: $\sigma P = (0,0.9,0.1)$
- after t steps: σP^t



Introduction to Markov Chains

<u>Def:</u> A Markov chain M=(Ω,P) is

• <u>irreducible</u> if there is a path in the transition graph from every state to every other state

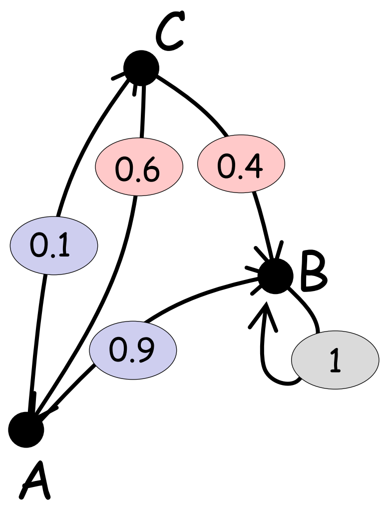
• <u>aperiodic</u> if for each state s, the gcd of all walk lengths from s to s is 1

<u>Example:</u>

Another simple MC (not irreducible):

P=

	A	B	C
A	0	0.9	0.1
B	0	1	0
C	0.6	0.4	0



Introduction to Markov Chains

Def: A Markov chain $M=(\Omega,P)$ is

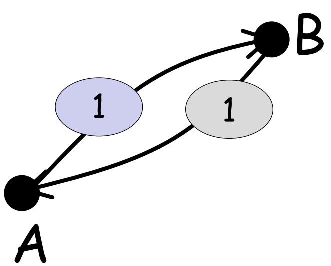
- irreducible if there is a path in the transition graph from every state to every other state
- aperiodic if for each state s, the gcd of all walk lengths from s to s is 1

Example:

Even simpler MC (not aperiodic):

P=

	A	B
A	0	1
B	1	0



Introduction to Markov Chains

Def: A Markov chain $M=(\Omega,P)$ is

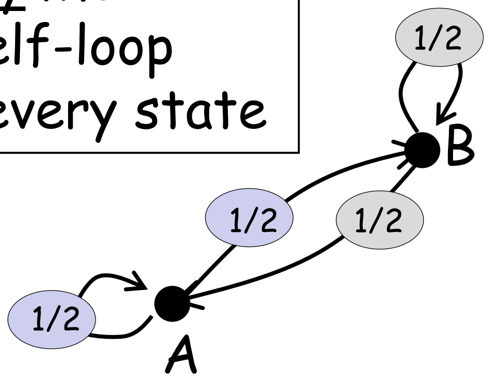
- irreducible if there is a path in the transition graph from every state to every other state
- aperiodic if for each state s, the gcd of all walk lengths from s to s is 1

Example:

Even simpler MC (now aperiodic):

Lazy MC:
a self-loop
at every state

$$P = \begin{array}{c|cc} & A & B \\ A & 1/2 & 1/2 \\ B & 1/2 & 1/2 \end{array}$$



Introduction to Markov Chains

Def: A Markov chain $M=(\Omega,P)$ is

- irreducible if there is a path in the transition graph from every state to every other state
- aperiodic if for each state s, the gcd of all walk lengths from s to s is 1
- ergodic if both irreducible and aperiodic

Thm: An ergodic MC has a unique stationary distribution.

Introduction to Markov Chains

Thm: An ergodic MC has a unique stationary distribution.

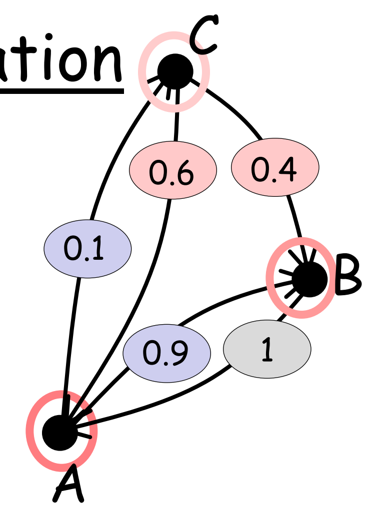
Def: For ϵ>0, mixing time is # steps needed to get ϵ-close to the stationary distribution π. Formally:

For a start state x, let σ_x be the corresponding starting distribution. Then

$t_{mix}(\epsilon)$ = minimum t such that for every x, $||\sigma_x P^t - \pi||_{tv} < \epsilon$

where, for two distributions μ,ν, their total variation distance is

$$||\mu - \nu||_{tv} = \frac{1}{2} \sum_{x \in \Omega} |\mu(x) - \nu(x)|$$



Example:

$||(1,0,0) - (0,0.9,0.1)||_{tv} = \frac{1}{2}(1+0.9+0.1) = 1$

$||(0.5,0.4,0.1) - (0.6,0.4,0)||_{tv} = \frac{1}{2}(0.1+0.1) = 0.1$

Introduction to Markov Chains

Thm: An ergodic MC has a unique stationary distribution.

$\pi P = \pi$

Observations:

- Eigenvalue: 1, with eigenvector π
- All eigenvalues, in absolute value, ≤ 1
- Mixing time depends on the spectral gap (difference between 1 and the 2nd largest eigenvalue in absolute value)

Thm: For an ergodic MC, let $1 = \lambda_1 > \lambda_2 \geq \ldots \geq \lambda_{min}$ be the eigenvalues and $\pi_{min} = \min_x \pi(x)$. Then

$$t_{mix}(\varepsilon) \leq \frac{1}{1 - \max\{|\lambda_2|, |\lambda_{\min}|\}} \log\left(\frac{1}{\varepsilon \pi_{\min}}\right)$$

$$t_{mix}(\varepsilon) \geq \frac{|\lambda_2|}{1 - \max\{|\lambda_2|, |\lambda_{\min}|\}} \log\left(\frac{1}{2\varepsilon}\right)$$

Exponential State Space

A MC on Colorings:

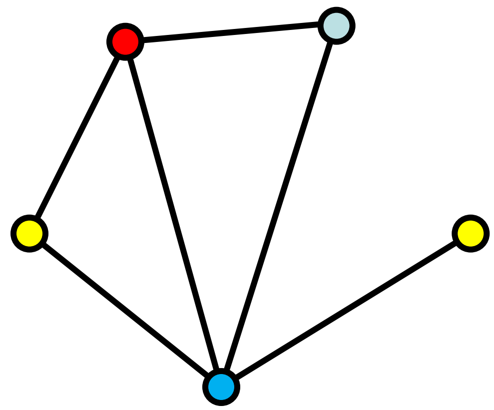
Given is a graph G and a set [q]. A coloring assigns to each vertex a color from [q], so that adjacent vertices have different colors. Let Ω be the set of all colorings.

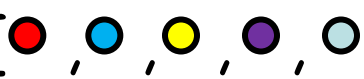
MC on Ω:

1. Choose a random vertex v and a random color c.
2. If no neighbor of v is colored by c, recolor v by c, otherwise, do nothing.

Example:

G:



[q] = {  }

Exponential State Space

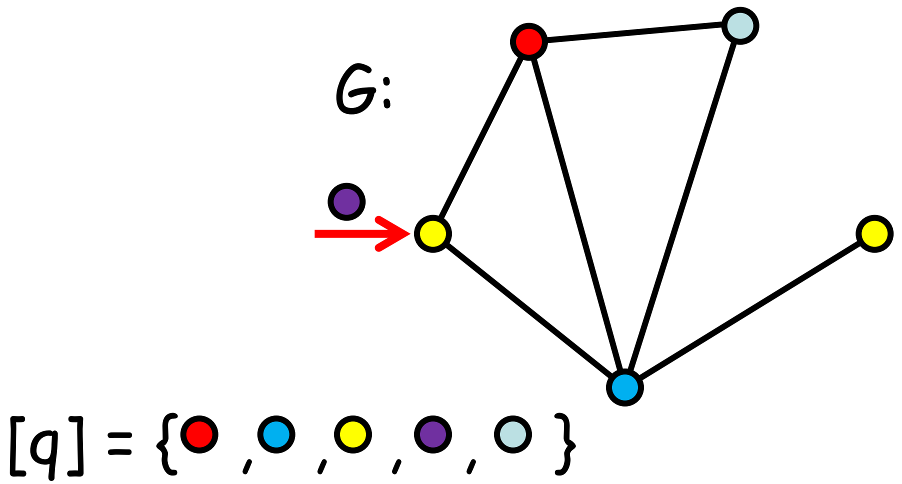
A MC on Colorings:

Given is a graph G and a set [q]. A coloring assigns to each vertex a color from [q], so that adjacent vertices have different colors. Let Ω be the set of all colorings.

MC on Ω:

1. Choose a random vertex v and a random color c.
2. If no neighbor of v is colored by c, recolor v by c, otherwise, do nothing.

Example:



Exponential State Space

A MC on Colorings:

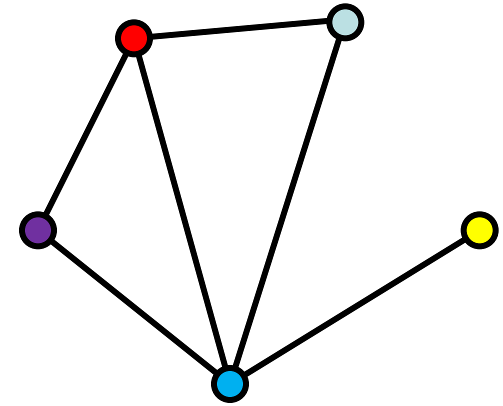
Given is a graph G and a set [q]. A coloring assigns to each vertex a color from [q], so that adjacent vertices have different colors. Let Ω be the set of all colorings.

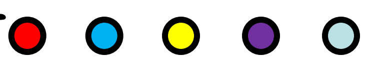
MC on Ω:

1. Choose a random vertex v and a random color c.
2. If no neighbor of v is colored by c, recolor v by c, otherwise, do nothing.

Example:

G:



[q] = {  }

Exponential State Space

A MC on Colorings:

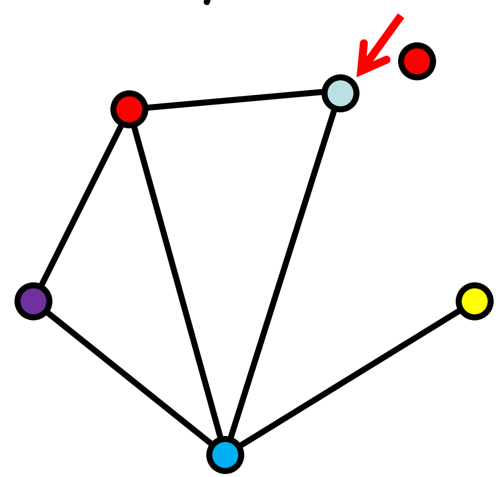
Given is a graph G and a set [q]. A <u>coloring</u> assigns to each vertex a color from [q], so that adjacent vertices have different colors. Let Ω be the set of all colorings.

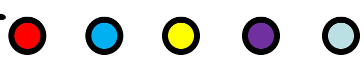
MC on Ω:

1. Choose a random vertex v and a random color c.
2. If no neighbor of v is colored by c, recolor v by c, otherwise, do nothing.

Example:

G:



$[q] = \{$  $\}$

Exponential State Space

A MC on Colorings:

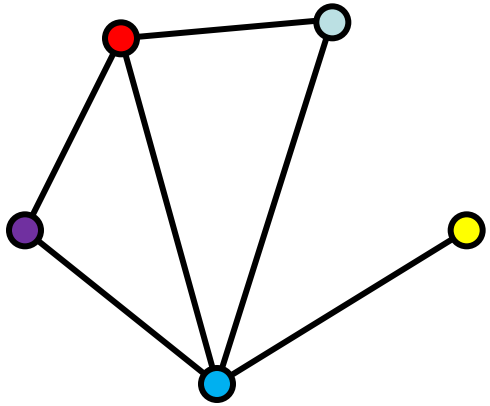
Given is a graph G and a set $[q]$. A coloring assigns to each vertex a color from $[q]$, so that adjacent vertices have different colors. Let Ω be the set of all colorings.

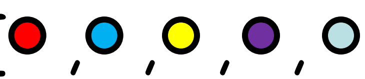
MC on Ω:

1. Choose a random vertex v and a random color c.
2. If no neighbor of v is colored by c, recolor v by c, otherwise, do nothing.

Example:

G:



$[q] = \{$  $\}$

Exponential State Space

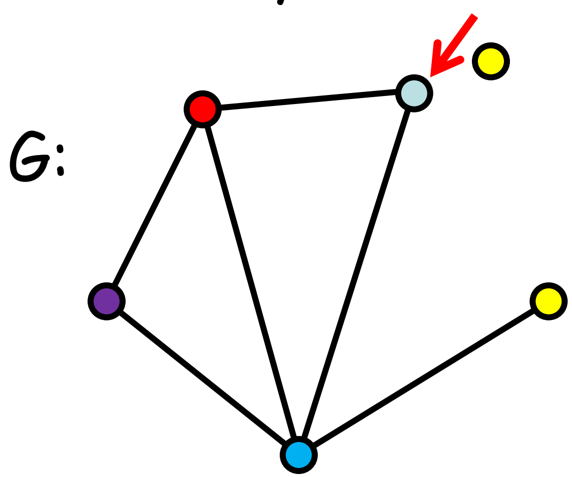
A MC on Colorings:

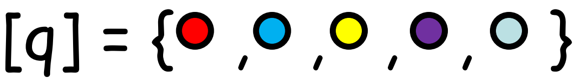
Given is a graph G and a set [q]. A <u>coloring</u> assigns to each vertex a color from [q], so that adjacent vertices have different colors. Let Ω be the set of all colorings.

MC on Ω:

1. Choose a random vertex v and a random color c.
2. If no neighbor of v is colored by c, recolor v by c, otherwise, do nothing.

Example:





Exponential State Space

A MC on Colorings:

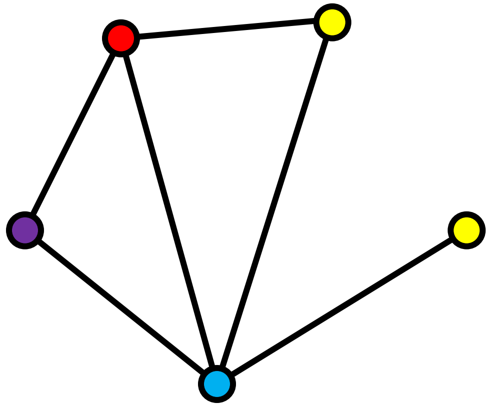
Given is a graph G and a set [q]. A coloring assigns to each vertex a color from [q], so that adjacent vertices have different colors. Let Ω be the set of all colorings.

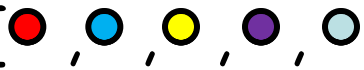
MC on Ω:

1. Choose a random vertex v and a random color c.
2. If no neighbor of v is colored by c, recolor v by c, otherwise, do nothing.

Example:

G:



[q] = {  }

Exponential State Space

A MC on Colorings:

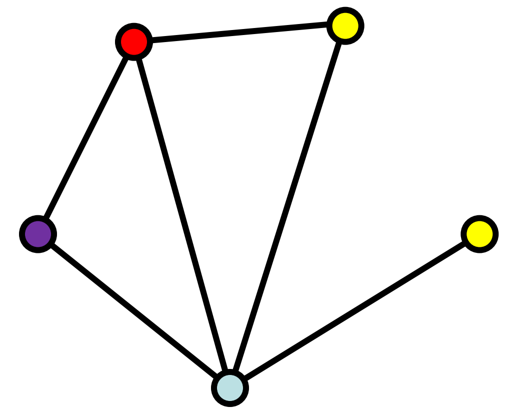
Given is a graph G and a set [q]. A coloring assigns to each vertex a color from [q], so that adjacent vertices have different colors. Let Ω be the set of all colorings.

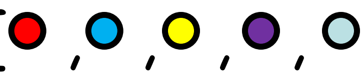
MC on Ω:

1. Choose a random vertex v and a random color c.
2. If no neighbor of v is colored by c, recolor v by c, otherwise, do nothing.

Example:

G:



[q] = {  }

Exponential State Space

A MC on Colorings:

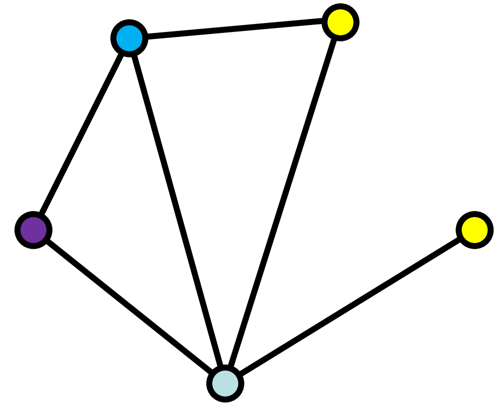
Given is a graph G and a set [q]. A coloring assigns to each vertex a color from [q], so that adjacent vertices have different colors. Let Ω be the set of all colorings.

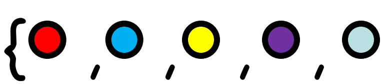
MC on Ω:

1. Choose a random vertex v and a random color c.
2. If no neighbor of v is colored by c, recolor v by c, otherwise, do nothing.

Example:

G:



[q] = {  }

Exponential State Space

A MC on Colorings:

Given is a graph G and a set [q]. A coloring assigns to each vertex a color from [q], so that adjacent vertices have different colors. Let Ω be the set of all colorings.

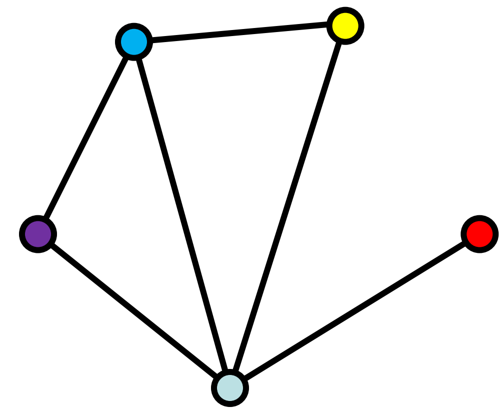
MC on Ω:

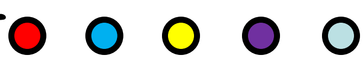
1. Choose a random vertex v and a random color c.
2. If no neighbor of v is colored by c, recolor v by c, otherwise, do nothing.

Example:

1. $|\Omega| \leq q^n$ where n = #vertices
2. aperiodic (self-loops)

G:



[q] = {  }

Exponential State Space

A MC on Colorings:

Given is a graph G and a set [q]. A coloring assigns to each vertex a color from [q], so that adjacent vertices have different colors. Let Ω be the set of all colorings.

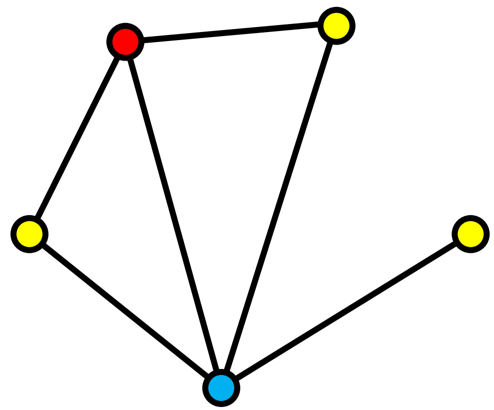
MC on Ω:

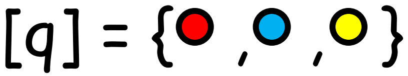
1. Choose a random vertex v and a random color c.
2. If no neighbor of v is colored by c, recolor v by c, otherwise, do nothing.

Example:

1. $|\Omega| \leq q^n$ where n = #vertices
2. aperiodic (self-loops)
3. not irreducible…

G:





Exponential State Space

A MC on Colorings:

Given is a graph G and a set [q]. A coloring assigns to each vertex a color from [q], so that adjacent vertices have different colors. Let Ω be the set of all colorings.

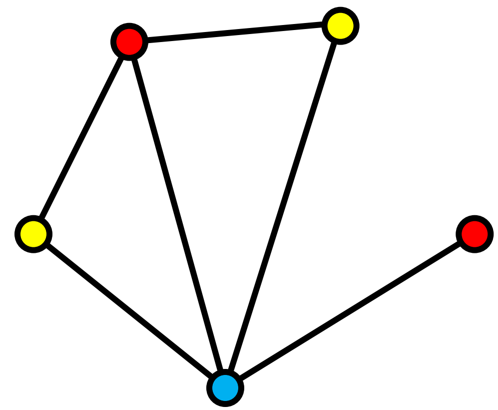
MC on Ω:

1. Choose a random vertex v and a random color c.
2. If no neighbor of v is colored by c, recolor v by c, otherwise, do nothing.

Example:

1. $|\Omega| \leq q^n$ where n = #vertices
2. aperiodic (self-loops)
3. not irreducible…

G:



[q] = {  }

Exponential State Space

A MC on Colorings:

Given is a graph G and a set [q]. A coloring assigns to each vertex a color from [q], so that adjacent vertices have different colors. Let Ω be the set of all colorings.

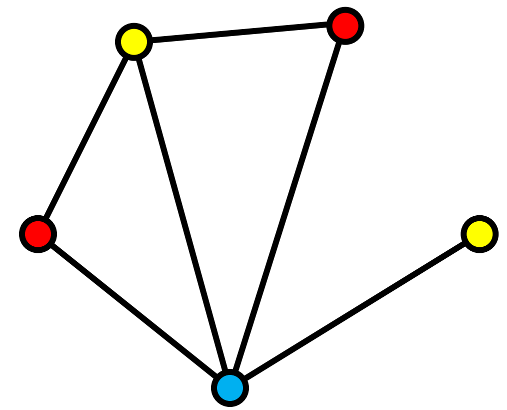
MC on Ω:

1. Choose a random vertex v and a random color c.
2. If no neighbor of v is colored by c, recolor v by c, otherwise, do nothing.

Example:

1. $|\Omega| \leq q^n$ where n = #vertices
2. aperiodic (self-loops)
3. if q > maxdeg+1 then irreducible

G:



[q] = {  }

Exponential State Space

A MC on Colorings:

Given is a graph G and a set [q]. A coloring assigns to each vertex a color from [q], so that adjacent vertices have different colors. Let Ω be the set of all colorings.

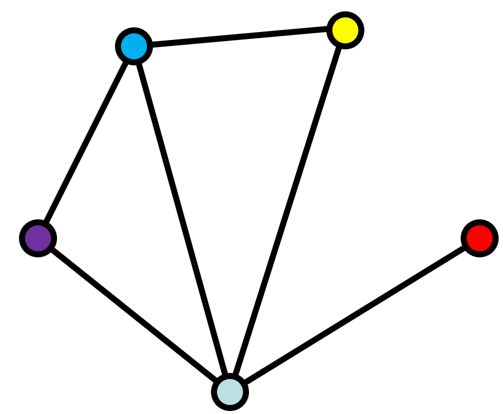
MC on Ω:

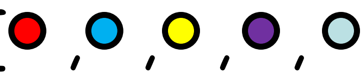
1. Choose a random vertex v and a random color c.
2. If no neighbor of v is colored by c, recolor v by c, otherwise, do nothing.

Example:

1. $|\Omega| \leq q^n$ where n = #vertices
2. aperiodic (self-loops)
3. if q > maxdeg+1 then irreducible

G:

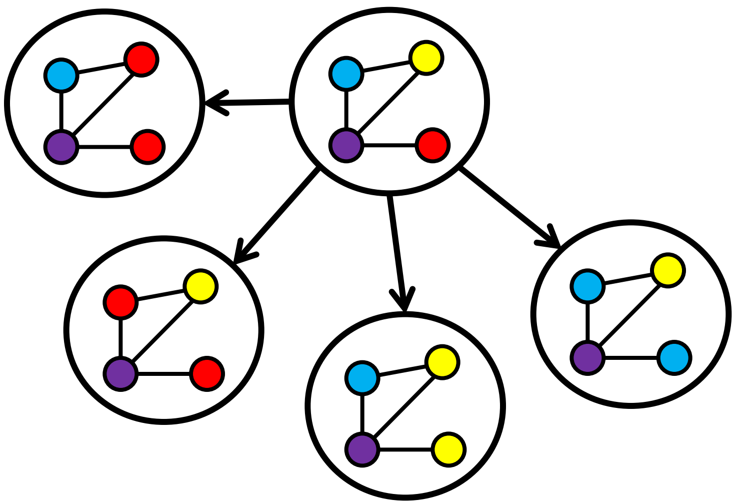


[q] = {  }

Exponential State Space

<u>A MC on Colorings:</u>

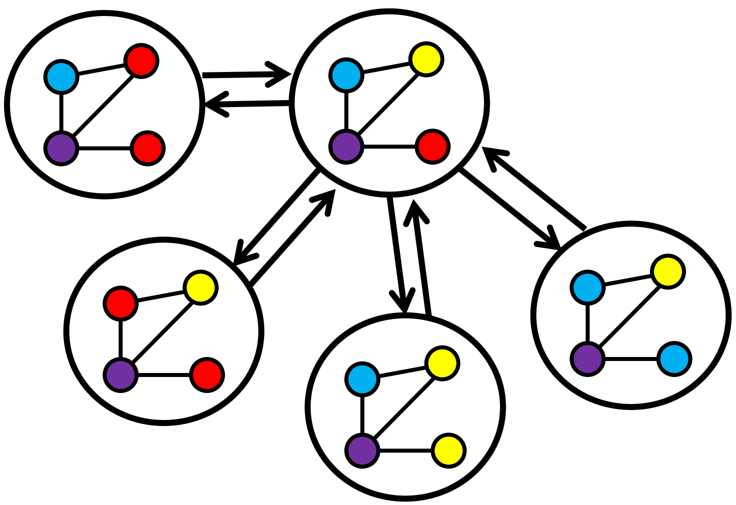
- ergodic: unique stationary distribution π
- transition probability for neighboring x,y: $P(x,y) = 1/(nq)$



Exponential State Space

A MC on Colorings:

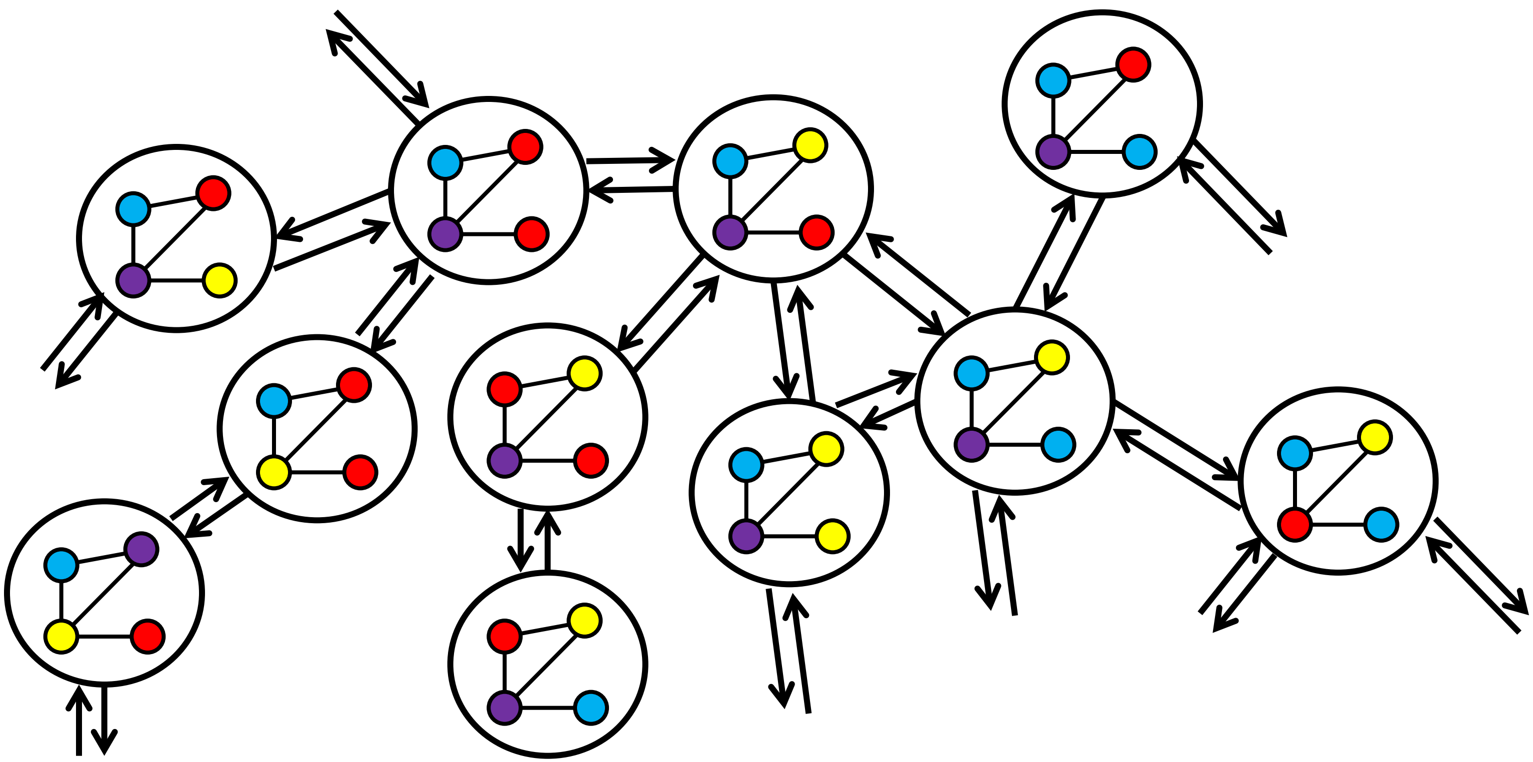
- ergodic: unique stationary distribution π
- transition probability for neighboring x,y: $P(x,y) = 1/(nq)$
- symmetric: $P(x,y)=P(y,x)$



Exponential State Space

A MC on Colorings:

- ergodic: unique stationary distribution π
- transition probability for neighboring x,y: $P(x,y) = 1/(nq)$
- symmetric: $P(x,y)=P(y,x)$

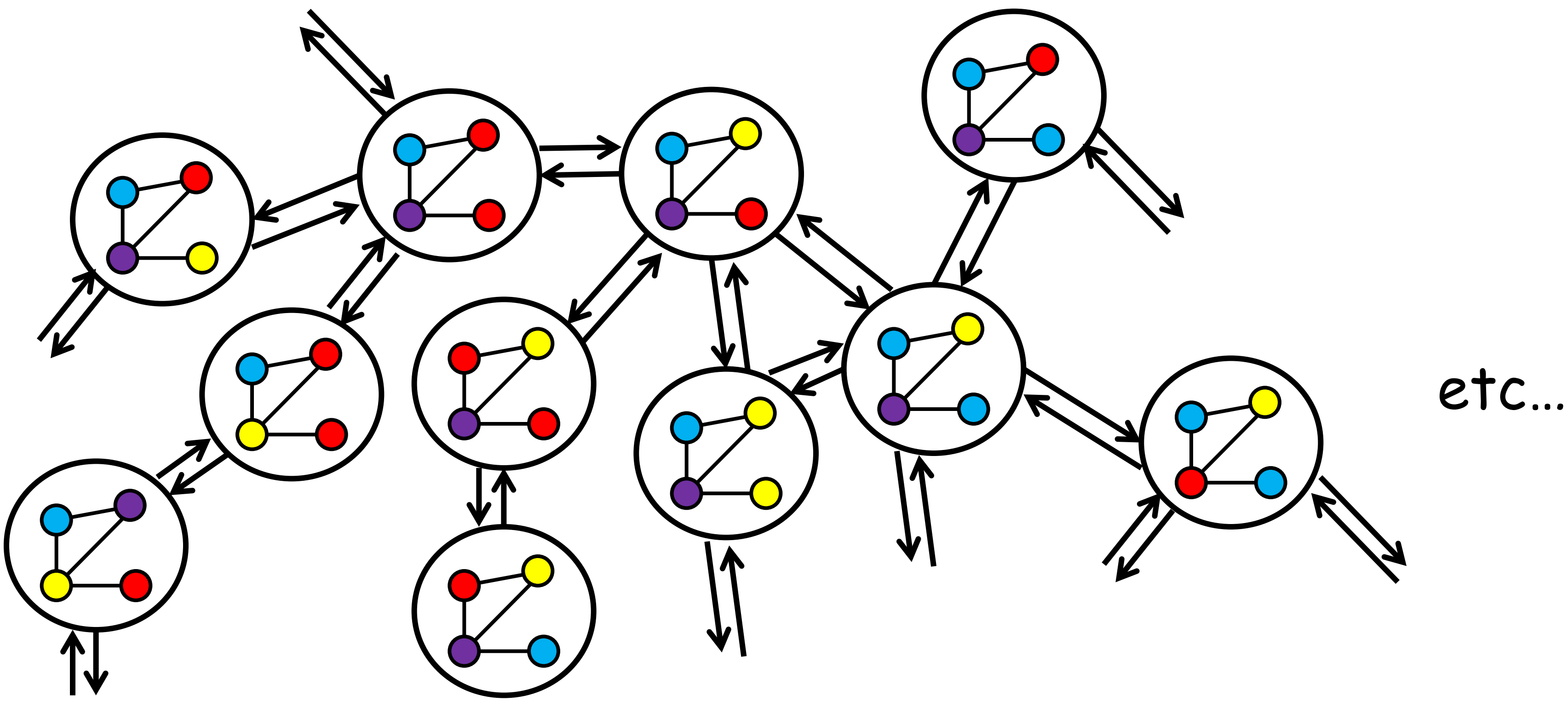


etc...

Exponential State Space

<u>A MC on Colorings:</u>

- ergodic: unique stationary distribution $\pi = 1/|\Omega|$ (uniform)
- transition probability for neighboring x,y: P(x,y) = 1/(nq)
- symmetric: P(x,y)=P(y,x)



Exponential State Space

A MC on Colorings:

- ergodic: unique stationary distribution $\pi = 1/|\Omega|$ (uniform)
- transition probability for neighboring x,y: $P(x,y) = 1/(nq)$
- symmetric: $P(x,y)=P(y,x)$

More generally:

A MC is reversible, if there is a distribution π such that:

$$\pi(x)\, P(x,y) = \pi(y)\, P(y,x)$$

[Also known as detailed balance condition.]

Then, π is stationary.

Designing a Markov Chain

Want: to sample from a target distribution π

How (Metropolis filter):

- start with a symmetric ergodic MC (Ω, P') -> uniform stationary distribution
- modify the probability of a move from x to y as follows:

$$P(x,y) = \min\{ \pi(y)/\pi(x) \, , 1 \} \, P'(x,y)$$

A MC is reversible, if there is a distribution π such that:

$$\pi(x) \, P(x,y) = \pi(y) \, P(y,x)$$

[Also known as detailed balance condition.]

Then, π is stationary.

Designing a Markov Chain

Want: to sample from a target distribution π

How (Metropolis filter):

- start with a symmetric ergodic MC (Ω, P') -> uniform stationary distribution
- modify the probability of a move from x to y as follows:

$$P(x,y) = \min \{ \pi(y)/\pi(x) \, , 1 \} \, P'(x,y)$$

Example: MC on colorings, with $\pi(x) \sim \alpha^{\#\text{ red vertices in } x}$ where $\alpha > 1$

1. Choose a random vertex v and a random color c.
2. If can recolor v by c and recoloring it
   a) increases ($\geq$) # red vertices: recolor v by c
   b) decreases # red vertices: recolor v by c with probability $1/\alpha$

Designing a Markov Chain

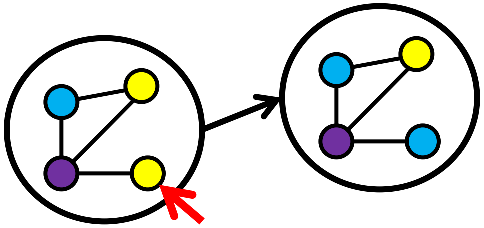
Want: to sample from a target distribution π

How (Metropolis filter):

- start with a symmetric ergodic MC (Ω, P') -> uniform stationary distribution
- modify the probability of a move from x to y as follows:

$$P(x,y) = \min\{\, \pi(y)/\pi(x)\ ,\ 1 \,\}\, P'(x,y)$$

Example: MC on colorings, with $\pi(x) \sim \alpha^{\#\text{ red vertices in } x}$ where $\alpha > 1$



1. Choose a random vertex v and a random color c.
2. If can recolor v by c and recoloring it
   a) increases ($\geq$) # red vertices: recolor v by c
   b) decreases # red vertices: recolor v by c with probability $1/\alpha$

Designing a Markov Chain

Want: to sample from a target distribution π

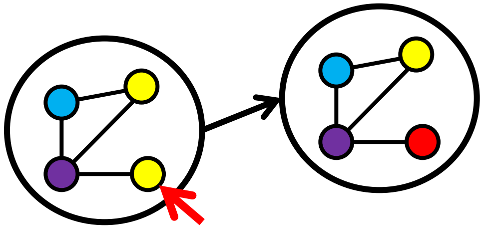
How (Metropolis filter):

• start with a symmetric ergodic MC (Ω, P') -> uniform stationary distribution

• modify the probability of a move from x to y as follows:

$$P(x,y) = \min\{\ \pi(y)/\pi(x)\ ,\ 1\ \}\ P'(x,y)$$

Example: MC on colorings, with $\pi(x) \sim \alpha^{\#\text{ red vertices in } x}$ where $\alpha > 1$



1. Choose a random vertex v and a random color c.
2. If can recolor v by c and recoloring it
   a) increases ($\geq$) # red vertices: recolor v by c
   b) decreases # red vertices: recolor v by c with probability $1/\alpha$

Designing a Markov Chain

Want: to sample from a target distribution π

How (Metropolis filter):

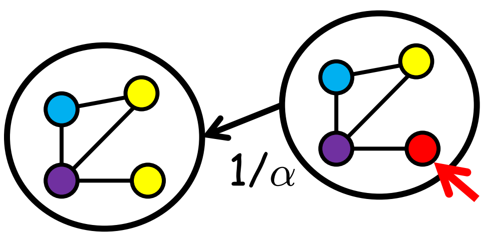
• start with a symmetric ergodic MC (Ω,P') -> uniform stationary distribution

• modify the probability of a move from x to y as follows:

$$P(x,y) = \min \{ \pi(y)/\pi(x) \, , 1 \} \, P'(x,y)$$

Example: MC on colorings, with $\pi(x) \sim \alpha^{\text{\# red vertices in x}}$ where $\alpha > 1$

1. Choose a random vertex v and a random color c.
2. If can recolor v by c and recoloring it
   a) increases ($\geq$) # red vertices: recolor v by c
   b) decreases # red vertices: recolor v by c with probability $1/\alpha$



Designing a Markov Chain

Want: to sample from a target distribution π

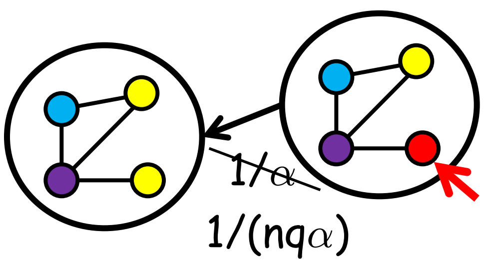
How (Metropolis filter):

- start with a symmetric ergodic MC (Ω,P') -> uniform stationary distribution
- modify the probability of a move from x to y as follows:

$$P(x,y) = \min\{\ \pi(y)/\pi(x)\ ,\ 1\ \}\ P'(x,y)$$

Example: MC on colorings, with $\pi(x) \sim \alpha^{\#\text{ red vertices in x}}$ where $\alpha > 1$

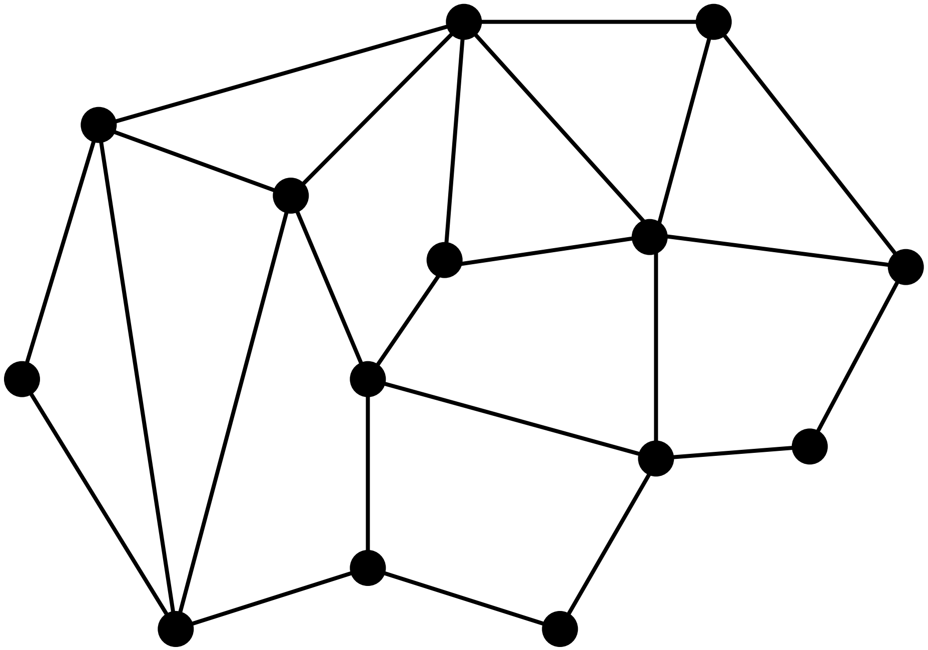
1. Choose a random vertex v and a random color c.
2. If can recolor v by c and recoloring it
   a) increases ($\geq$) # red vertices: recolor v by c
   b) decreases # red vertices: recolor v by c with probability $1/\alpha$



Relation to Counting

Given an undirected graph $G=(V,E)$, a **matching** $M \subseteq E$ is a set of vertex disjoint edges. A matching is **perfect** if $|M|=n/2$, where n = # vertices (and m = # edges).

Example:

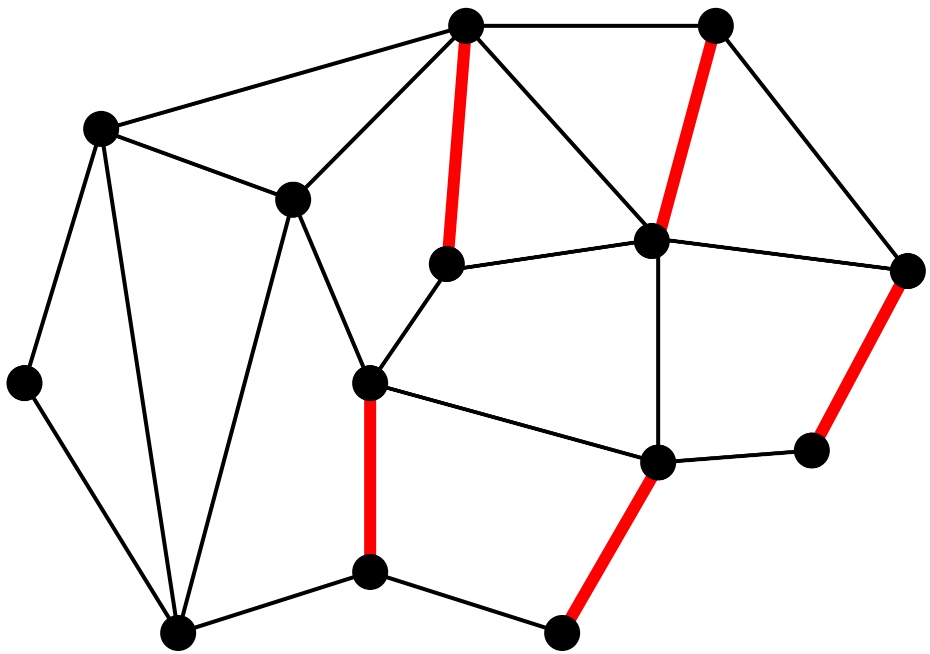


Relation to Counting

Given an undirected graph G=(V,E), a **matching** M⊆E is a set of vertex disjoint edges. A matching is **perfect** if |M|=n/2, where n = # vertices (and m = # edges).

Example:

A matching

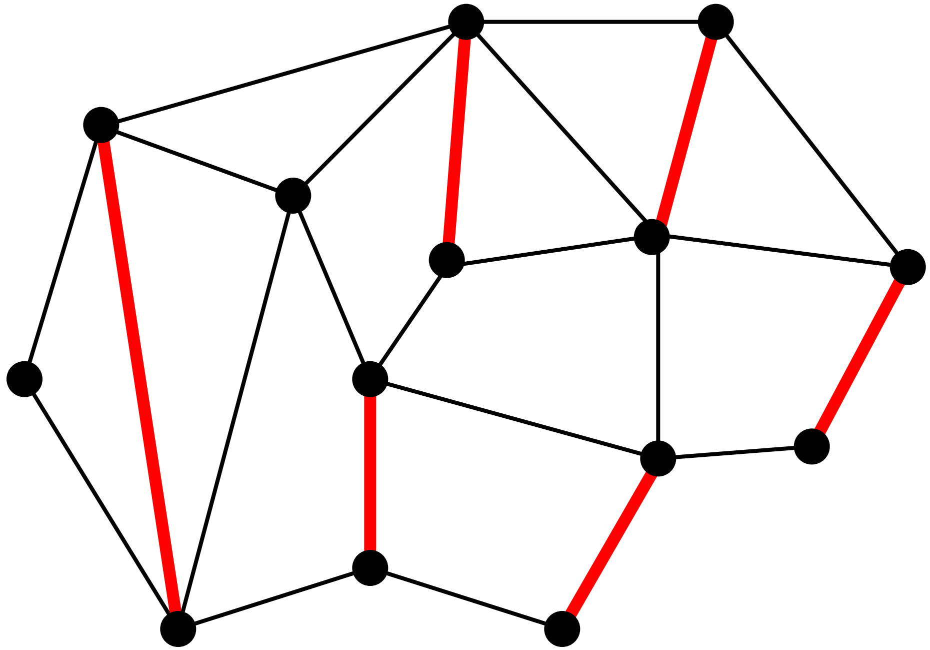


Relation to Counting

Given an undirected graph G=(V,E), a **matching** M⊆E is a set of vertex disjoint edges. A matching is **perfect** if |M|=n/2, where n = # vertices (and m = # edges).

Example:

Another perfect matching

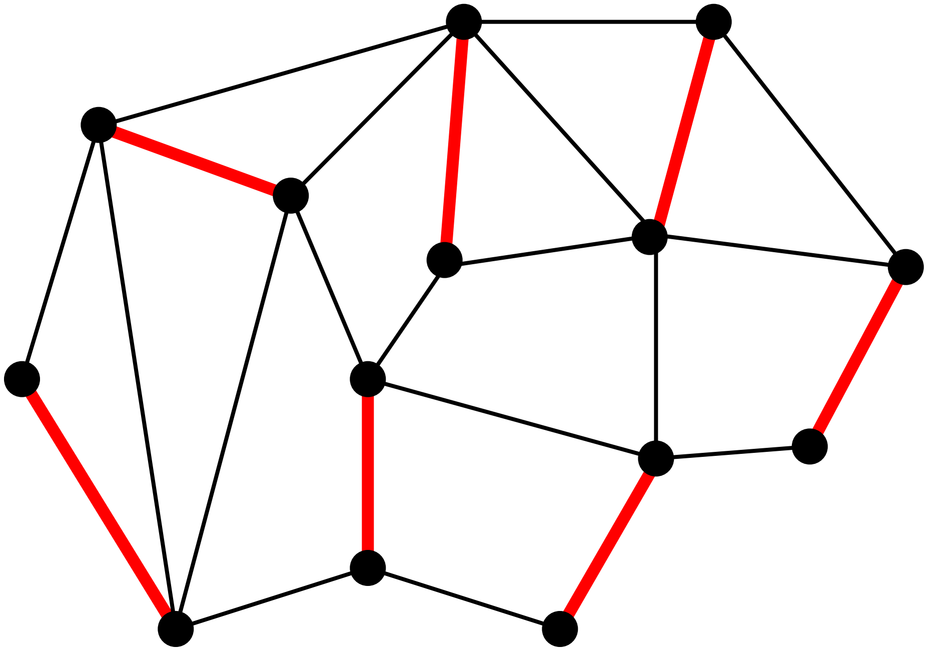


Relation to Counting

Given an undirected graph G=(V,E), a **matching** M⊆E is a set of vertex disjoint edges. A matching is **perfect** if |M|=n/2, where n = # vertices (and m = # edges).

Example:

A perfect matching



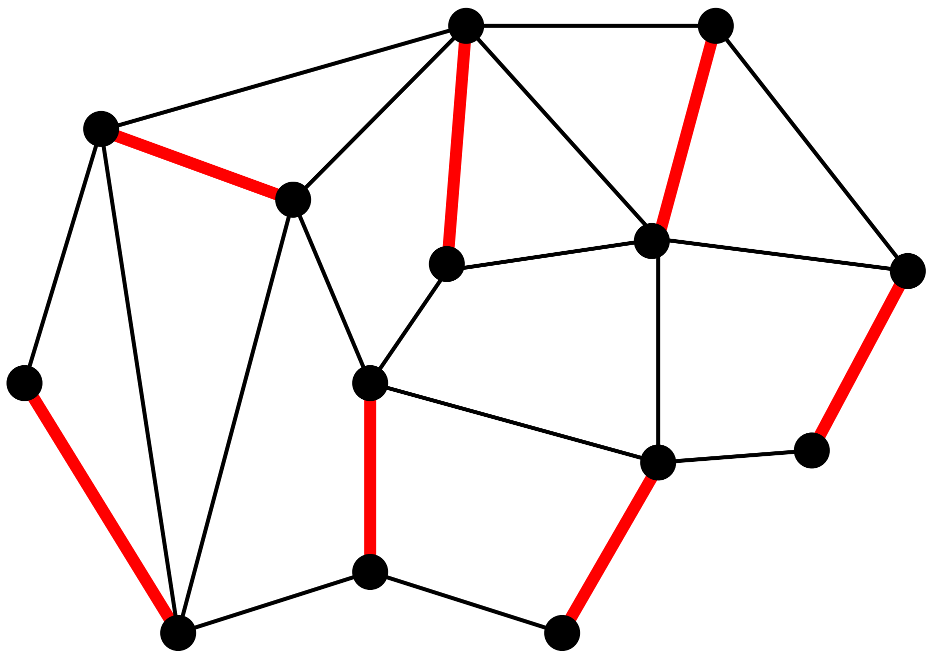
Relation to Counting

Given an undirected graph G=(V,E), a **matching** M⊆E is a set of vertex disjoint edges. A matching is **perfect** if |M|=n/2, where n = # vertices (and m = # edges).

Known:

• Finding a maximum matching can be done in polynomial time [O(m√n) Micali-Vazirani '80]

• Counting perfect matchings is #P-complete [Valiant '79]



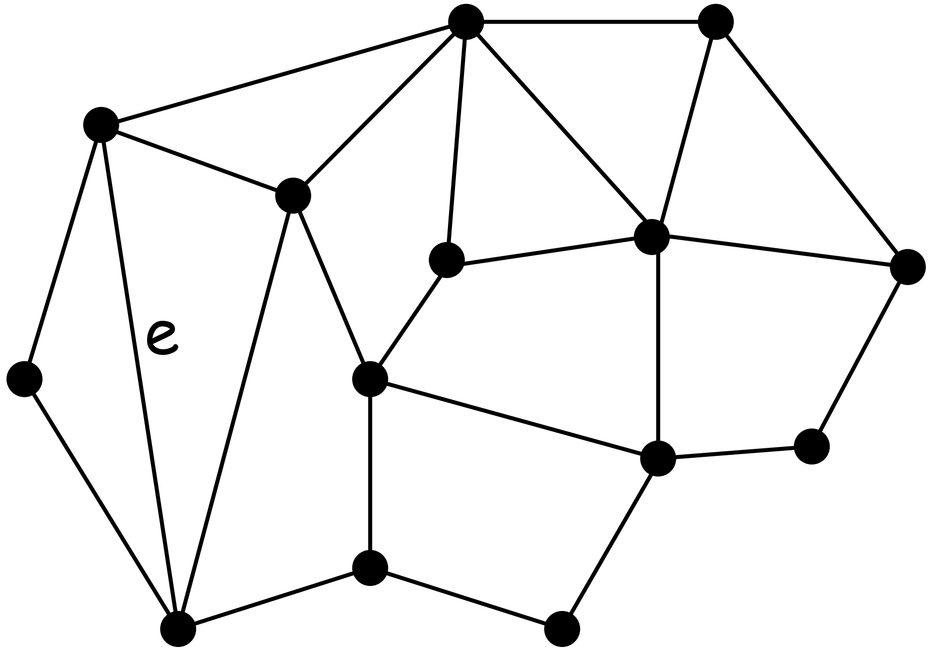
Goal: *Approximate* counting

From Sampling to Counting

Suppose we can sample a uniformly random matching.

Then:

- Let e be an arbitrary edge. Use sampling to determine the fraction of matchings that do not use e.

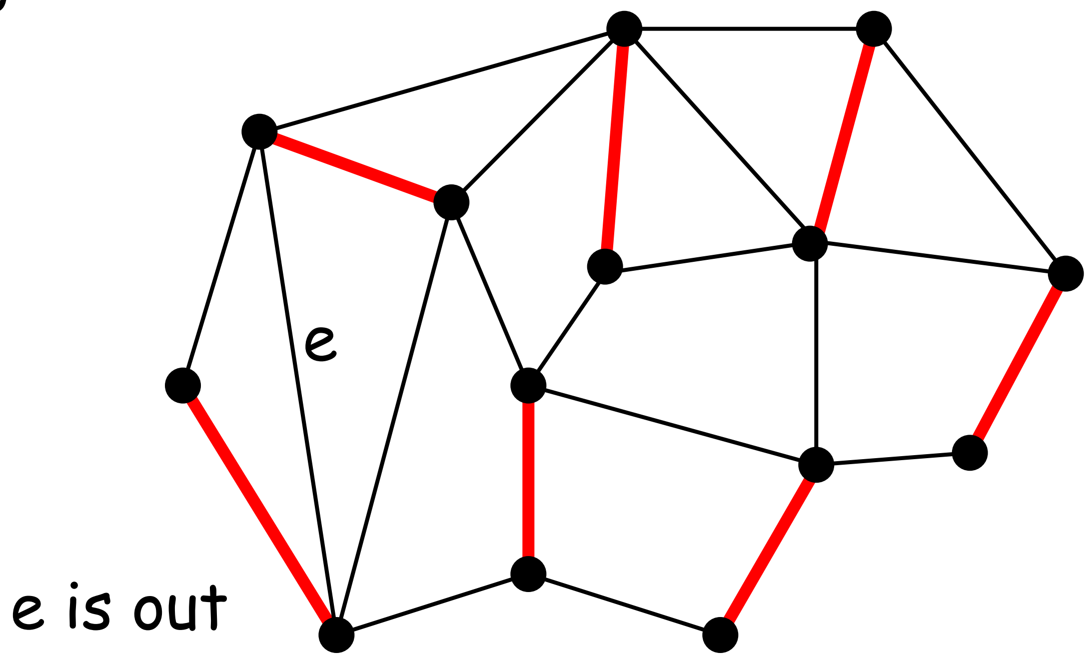


From Sampling to Counting

Suppose we can sample a uniformly random matching.

Then:

- Let e be an arbitrary edge. Use sampling to determine the fraction of matchings that do not use e.

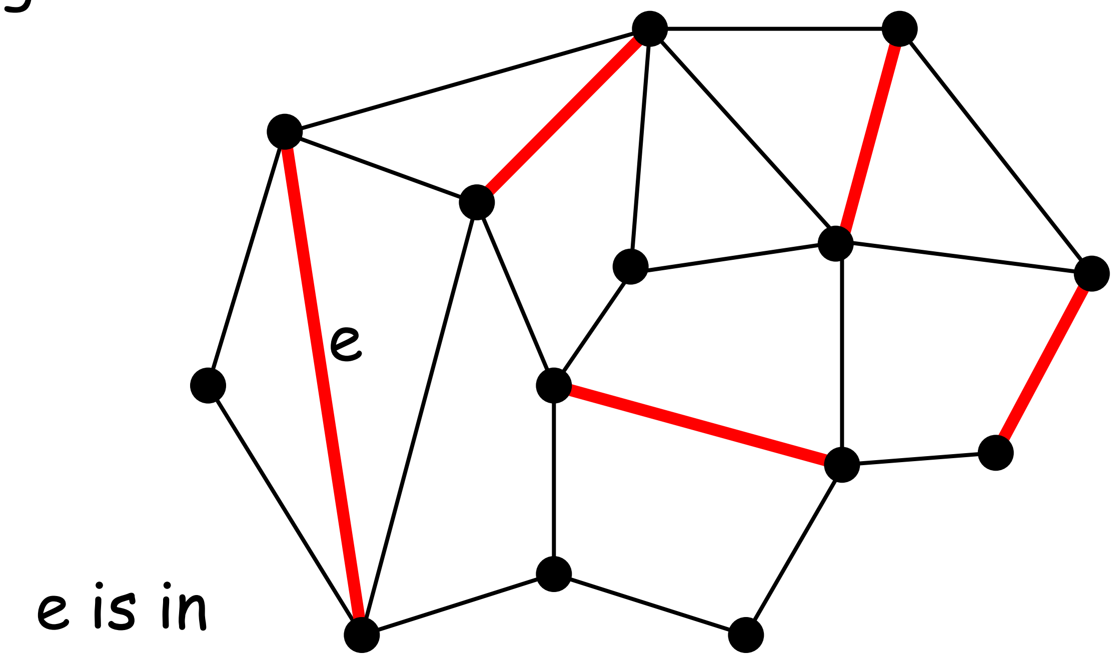


From Sampling to Counting

Suppose we can sample a uniformly random matching.

Then:

- Let e be an arbitrary edge. Use sampling to determine the fraction of matchings that do not use e.

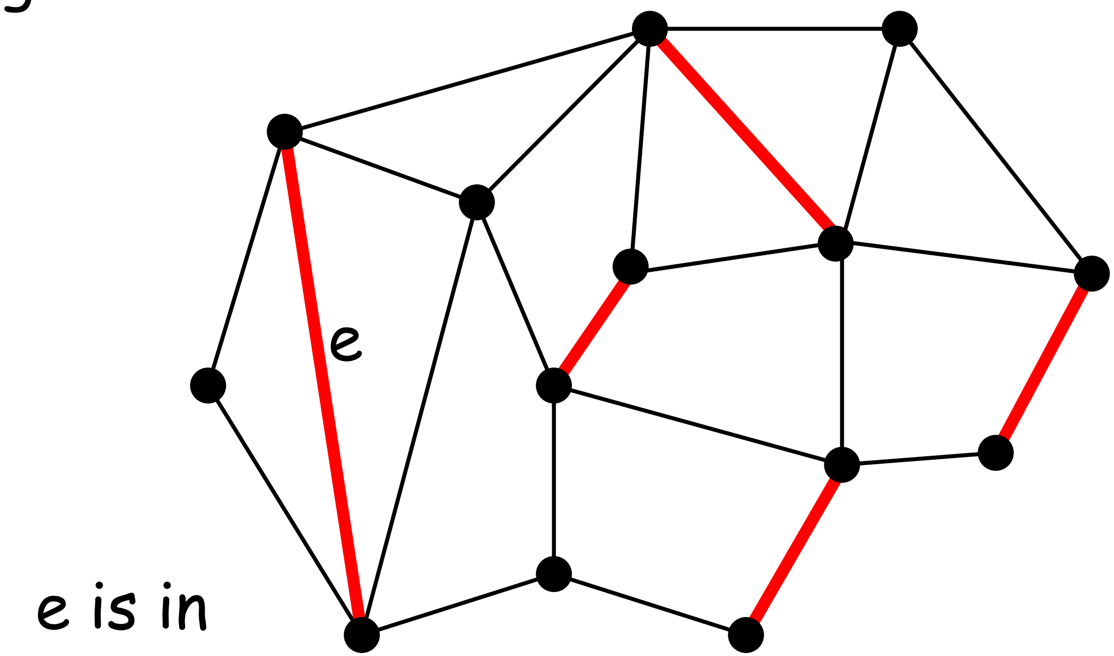


From Sampling to Counting

Suppose we can sample a uniformly random matching.

Then:

- Let e be an arbitrary edge. Use sampling to determine the fraction of matchings that do not use e.

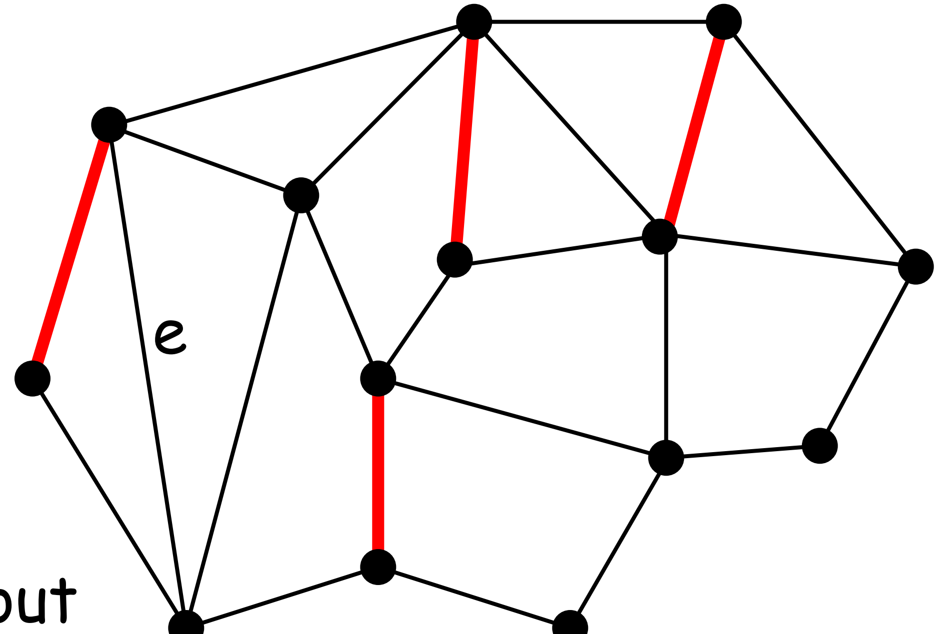


From Sampling to Counting

Suppose we can sample a uniformly random matching.

Then:

- Let e be an arbitrary edge. Use sampling to determine the fraction of matchings that do not use e.



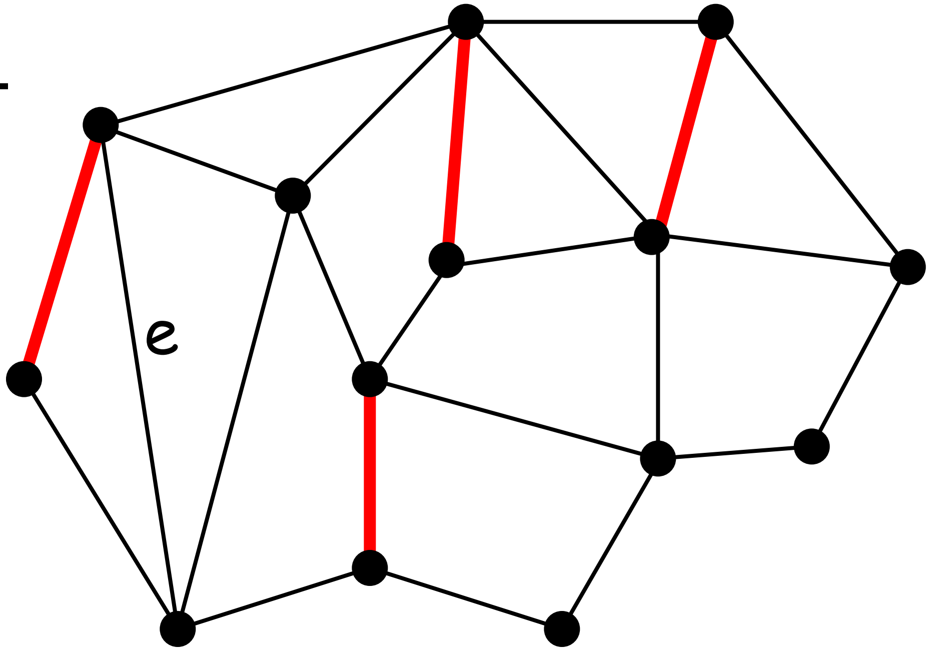
e is out

From Sampling to Counting

Estimator:

Indicator random variable X,
=1 if matching does not contain e and
=0 otherwise

$$E[X] = \frac{\#\text{ matchings without } e}{\#\text{ matchings}}$$



From Sampling to Counting

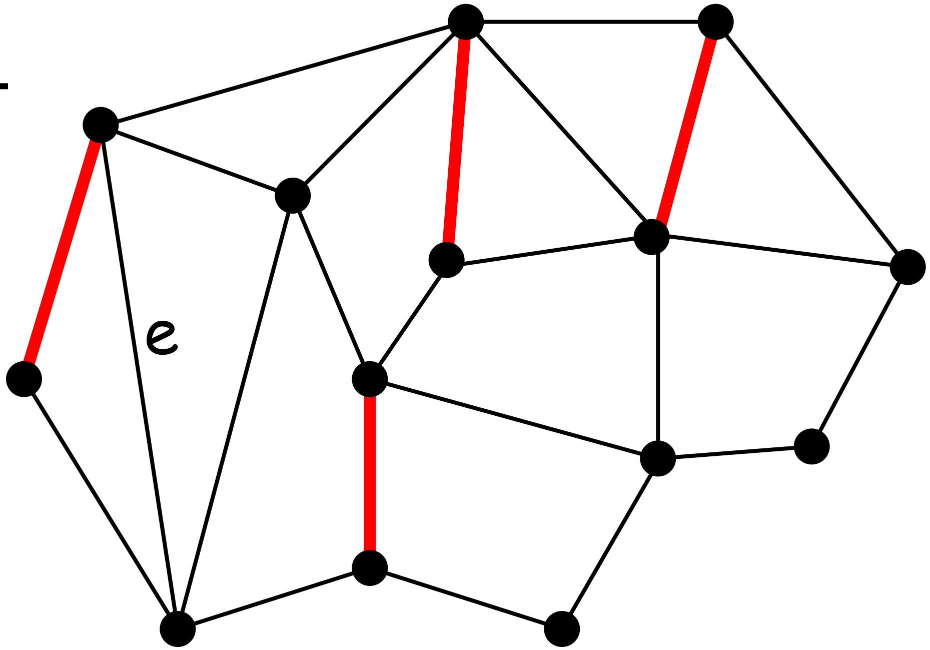
Estimator:

Indicator random variable X,
=1 if matching does not contain e and
=0 otherwise

$$E[X] = \frac{\#\text{ matchings without } e}{\#\text{ matchings}}$$

Observation:

$1/2 \le E[X] \le 1$

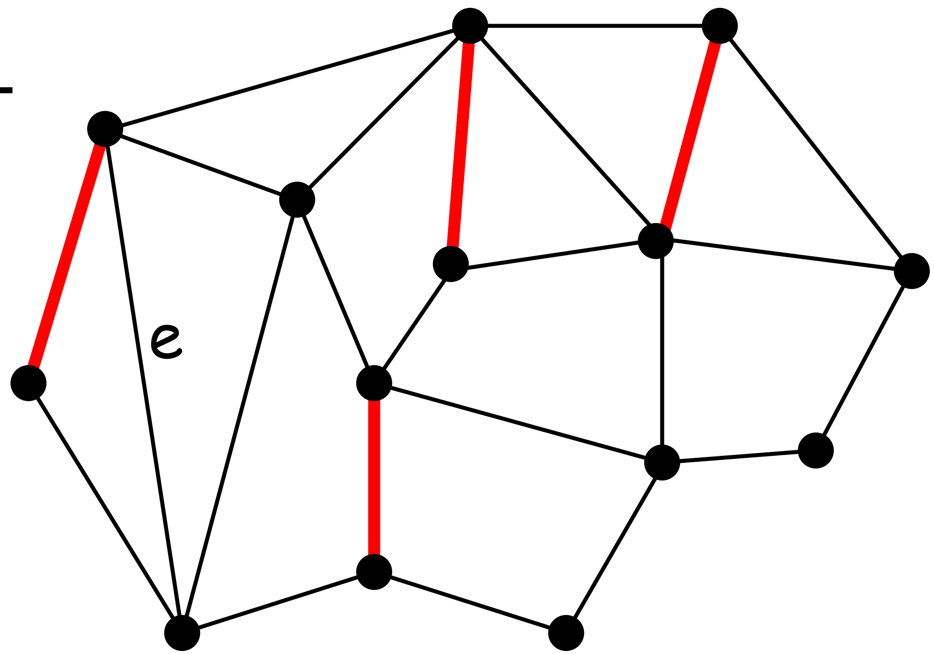


From Sampling to Counting

Estimator:

Indicator random variable X,
=1 if matching does not contain e and
=0 otherwise

$$E[X] = \frac{\#\text{ matchings without } e}{\#\text{ matchings}}$$



Observation:

$1/2 \leq E[X] \leq 1$

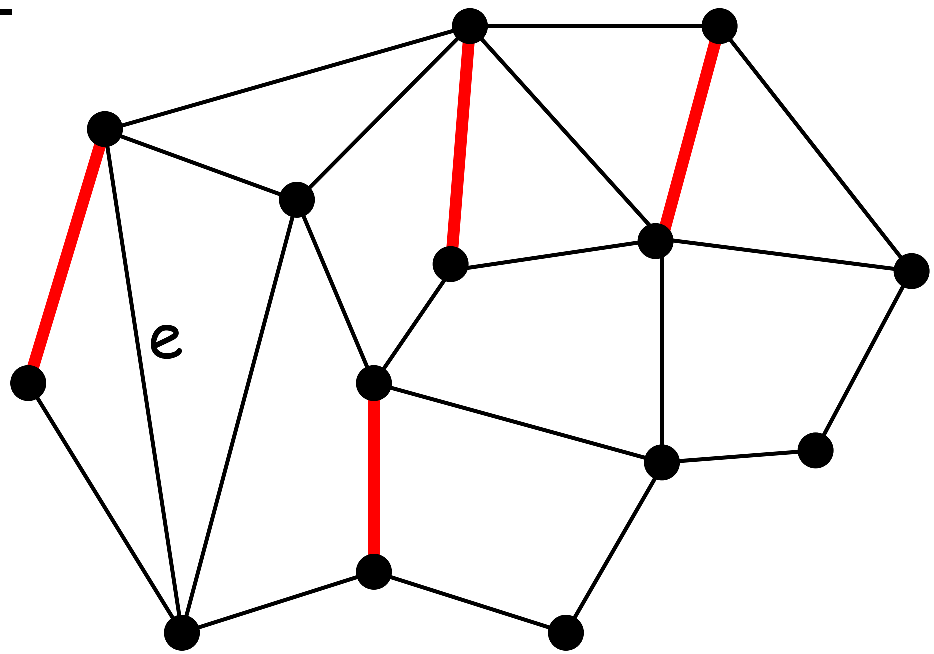
(For every matching with edge e, we can remove it to get a matching without e.)

From Sampling to Counting

Suppose we can sample a uniformly random matching.

• Let e be an arbitrary edge. Use sampling to determine the fraction of matchings that do not use e.

$$E[X] = \frac{\text{\# matchings without e}}{\text{\# matchings}}$$



• Let $G_0 = G$ and $G_1 = G\text{-}e$, then:

$$E[X] = \frac{\text{\# matchings in } G_1}{\text{\# matchings in } G_0}$$

From Sampling to Counting

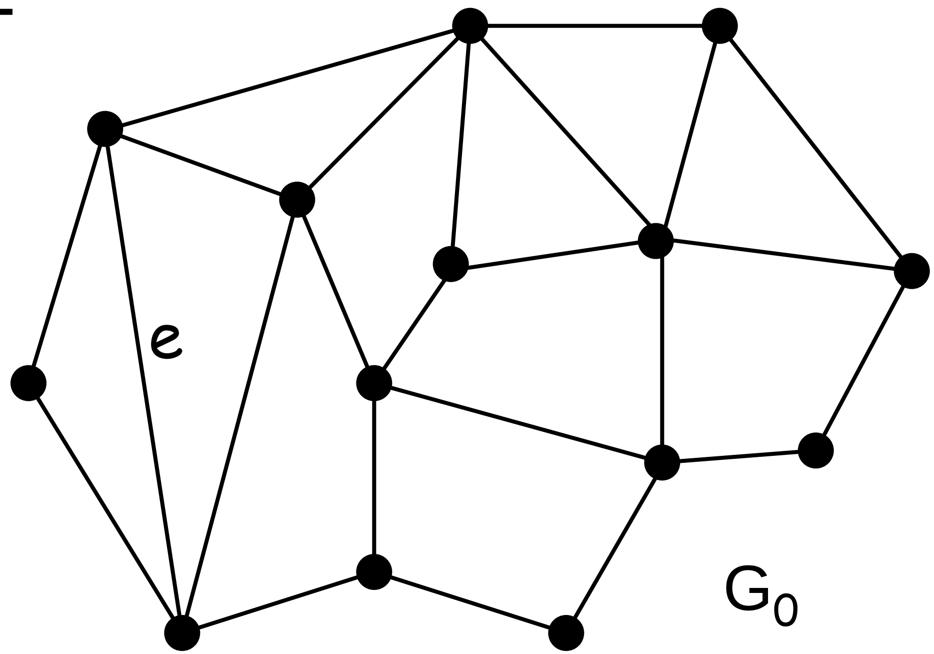
Suppose we can sample a uniformly random matching.

- Let e be an arbitrary edge. Use sampling to determine the fraction of matchings that do not use e.

$$E[X] = \frac{\#\text{ matchings without e}}{\#\text{ matchings}}$$

- Let $G_0 = G$ and $G_1 = G\text{-}e$, then:

$$E[X] = \frac{\#\text{ matchings in } G_1}{\#\text{ matchings in } G_0}$$



From Sampling to Counting

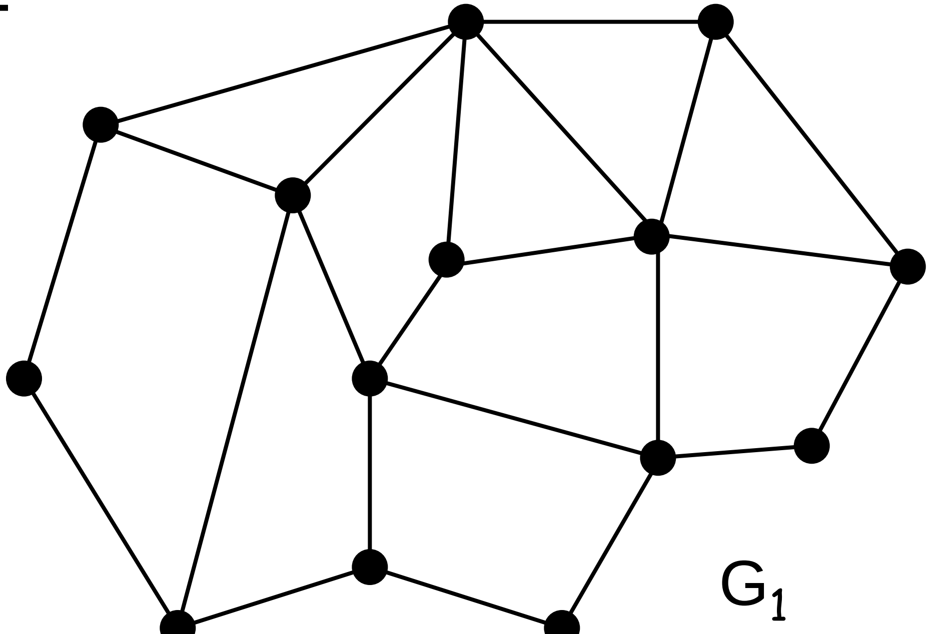
Suppose we can sample a uniformly random matching.

• Let e be an arbitrary edge. Use sampling to determine the fraction of matchings that do not use e.

$$E[X] = \frac{\text{\# matchings without e}}{\text{\# matchings}}$$

• Let $G_0 = G$ and $G_1 = G\text{-}e$, then:

$$E[X] = \frac{\text{\# matchings in } G_1}{\text{\# matchings in } G_0}$$



From Sampling to Counting

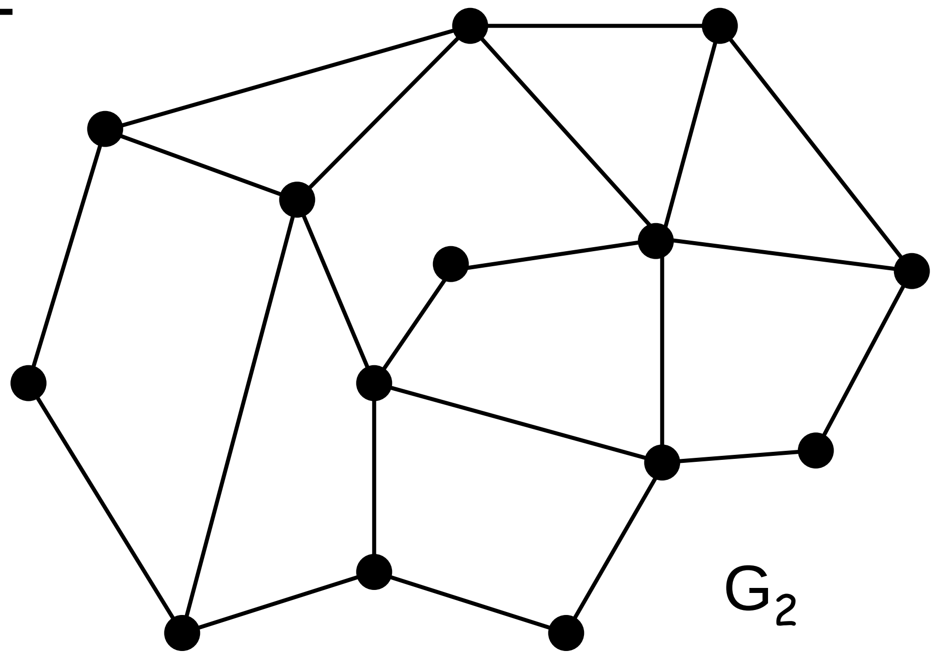
Suppose we can sample a uniformly random matching.

- Let e be an arbitrary edge. Use sampling to determine the fraction of matchings that do not use e.

$$E[X] = \frac{\text{\# matchings without e}}{\text{\# matchings}}$$

- Let $G_0 = G$ and $G_1 = G\text{-}e$, then:

$$E[X] = \frac{\text{\# matchings in } G_1}{\text{\# matchings in } G_0}$$



From Sampling to Counting

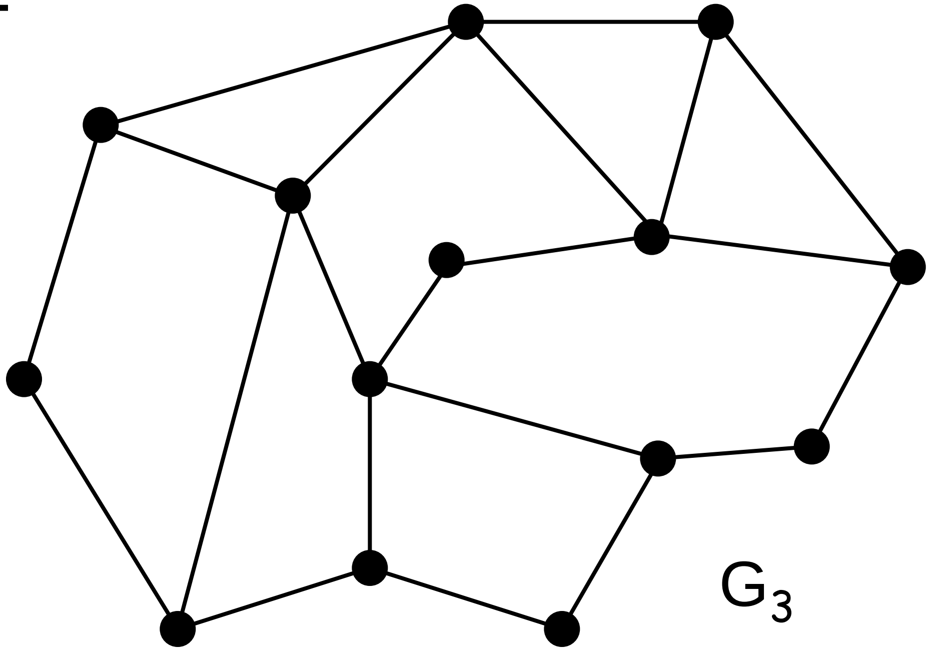
Suppose we can sample a uniformly random matching.

- Let e be an arbitrary edge. Use sampling to determine the fraction of matchings that do not use e.

$$E[X] = \frac{\text{\# matchings without e}}{\text{\# matchings}}$$

- Let $G_0 = G$ and $G_1 = G\text{-}e$, then:

$$E[X] = \frac{\text{\# matchings in } G_1}{\text{\# matchings in } G_0}$$



From Sampling to Counting

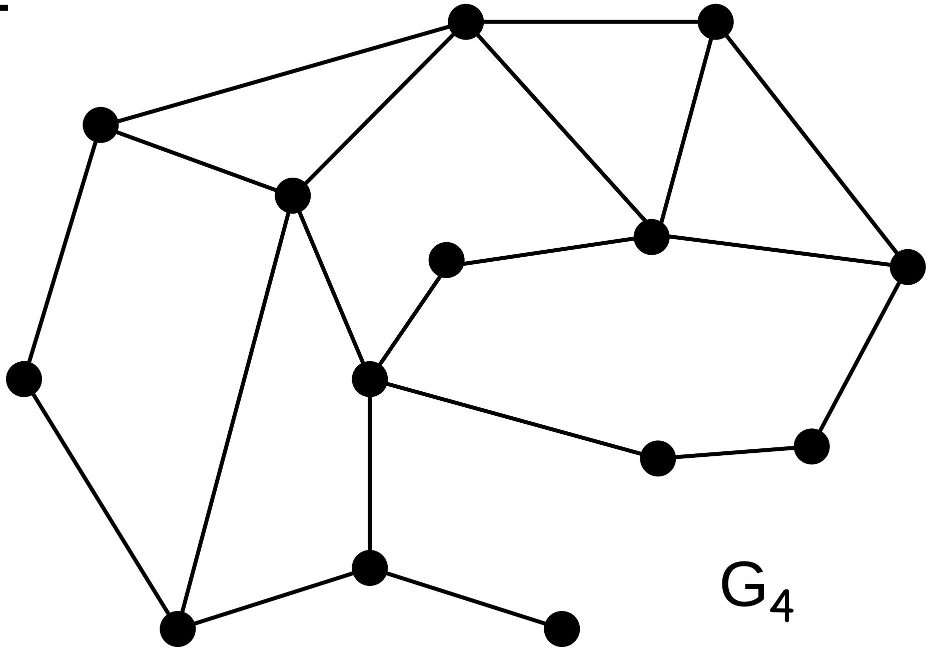
Suppose we can sample a uniformly random matching.

• Let e be an arbitrary edge. Use sampling to determine the fraction of matchings that do not use e.

$$E[X] = \frac{\#\text{ matchings without e}}{\#\text{ matchings}}$$

• Let $G_0 = G$ and $G_1 = G\text{-}e$, then:

$$E[X] = \frac{\#\text{ matchings in } G_1}{\#\text{ matchings in } G_0}$$



From Sampling to Counting

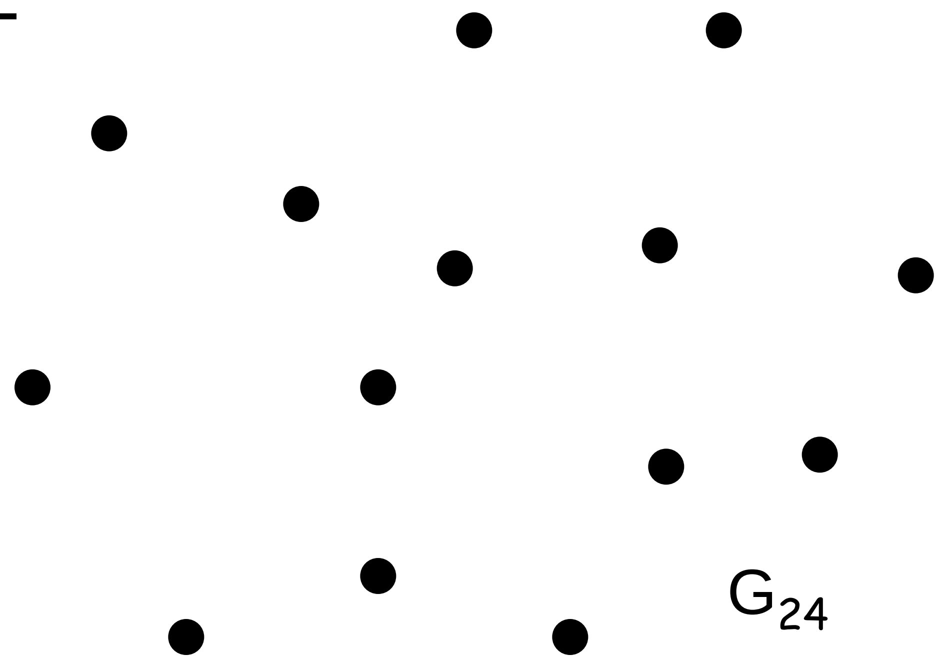
Suppose we can sample a uniformly random matching.

• Let e be an arbitrary edge. Use sampling to determine the fraction of matchings that do not use e.

$$E[X] = \frac{\text{\# matchings without } e}{\text{\# matchings}}$$

• Let $G_0 = G$ and $G_1 = G\text{-}e$, then:

$$E[X] = \frac{\text{\# matchings in } G_1}{\text{\# matchings in } G_0}$$



From Sampling to Counting

Suppose we can sample a uniformly random matching.

- Let e be an arbitrary edge. Use sampling to determine the fraction of matchings that do not use e.

$$E[X] = \frac{\#\text{ matchings without } e}{\#\text{ matchings}} = \frac{\#\text{ matchings in } G_1}{\#\text{ matchings in } G_0} = \frac{|M(G_1)|}{|M(G_0)|}$$

- Let $e_1,\ldots,e_m$ be the edges of G. Let $G_i = G-\{e_1,\ldots,e_i\}$. Let $M(G)$ be the set of all matchings in G. Estimate: $\frac{|M(G_{i+1})|}{|M(G_i)|}$
- Then:

$$\frac{|M(G_1)|}{|M(G_0)|}\;\frac{|M(G_2)|}{|M(G_1)|}\;\frac{|M(G_3)|}{|M(G_2)|}\cdots\frac{|M(G_m)|}{|M(G_{m-1})|} = \frac{|M(G_m)|}{|M(G_0)|} = \frac{1}{|M(G)|}$$

[**<u>Self-reducibility</u>**]

From Sampling to Counting: Technicalities

Terminology:

- **Almost uniform sampler**: for a tolerance parameter $\delta>0$, it produces a sample from a distribution within variation distance of δ from the uniform distribution

- **Fully polynomial almost uniform sampler (FPAUS)**: runs in time polynomial in input size and log $1/\delta$

- **Randomized approximation scheme**: for a counting problem and error tolerance ϵ, produce an answer within $(1+-\epsilon)$ factor of the count with probability $\geq 3/4$

- **Fully polynomial randomized approximation scheme (FPRAS)**: runs in time polynomial in input size and $1/\epsilon$

From Sampling to Counting: Technicalities

Thm [Jerrum-Valiant-Vazirani '86]:

FPAUS for sampling from all matchings => FPRAS for counting all matchings.

In particular, get an FPRAS with running time

$$O(\ T(n,m,\epsilon/(6m))\ m^2/\epsilon^2\ ),$$

where $T(n,m,\delta)$ is the running time of the FPAUS.

Goal: to design a Markov chain to sample matchings;
better mixing time => better running time of the FPRAS