CS 263: Counting and Sampling

Nima Anari

Stanford University

slides for

Bipartite Perfect Matchings

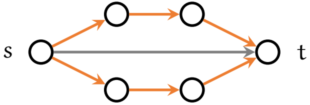
Review

- P, P' reversible with same stationary distribution

Review

- P, P' reversible with same stationary distribution
- Comparison: route Q' through Q with low congestion and length.

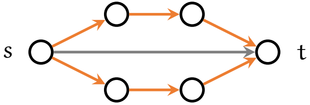
$$\pi(\text{path} \mid X_0 = s, X_\ell = t)$$



Review

- P, P' reversible with same stationary distribution
- Comparison: route Q' through Q with low congestion and length.

$$\pi(\text{path} \mid X_0 = s, X_\ell = t)$$



Congestion

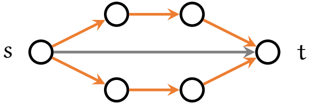
Suppose π is dist over paths and Q is ergodic flow. Congestion is

$$\max\left\{ \frac{\mathbb{P}_{\text{path}\sim\pi}[(x\to y)\in\text{path}]}{Q(x,y)} \;\middle|\; x \neq y \right\}$$

Review

- P, P' reversible with same stationary distribution
- Comparison: route Q' through Q with low congestion and length.

$$\pi(\text{path} \mid X_0 = s, X_\ell = t)$$



Congestion

Suppose π is dist over paths and Q is ergodic flow. Congestion is

$$\max\left\{ \frac{\mathbb{P}_{\text{path}\sim\pi}[(x\to y)\in\text{path}]}{Q(x,y)} \,\middle|\, x \neq y \right\}$$

Lemma: comparison

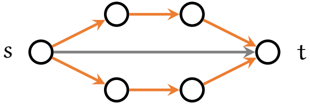
Suppose ρ, ρ' are χ^2 contraction rates:

$$\rho \geqslant \frac{\rho'}{(\text{congestion})\cdot(\text{max length})}$$

Review

- P, P' reversible with same stationary distribution
- Comparison: route Q' through Q with low congestion and length.

$$\pi(\text{path} \mid X_0 = s, X_\ell = t)$$



Congestion

Suppose π is dist over paths and Q is ergodic flow. Congestion is

$$\max\left\{ \frac{\mathbb{P}_{\text{path}\sim\pi}[(x\to y)\in\text{path}]}{Q(x,y)} \;\middle|\; x \neq y \right\}$$

Lemma: comparison

Suppose ρ, ρ' are χ^2 contraction rates:

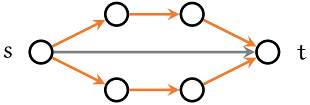
$$\rho \geqslant \frac{\rho'}{(\text{congestion})\cdot(\text{max length})}$$

- If len $\leqslant 1$, can use any $\mathcal{D}_\phi$.

Review

- P, P' reversible with same stationary distribution
- Comparison: route Q' through Q with low congestion and length.

$$\pi(\text{path} \mid X_0 = s, X_\ell = t)$$



Congestion

Suppose π is dist over paths and Q is ergodic flow. Congestion is

$$\max\left\{ \frac{\mathbb{P}_{\text{path}\sim\pi}[(x\to y)\in\text{path}]}{Q(x,y)} \;\middle|\; x \neq y \right\}$$

Lemma: comparison

Suppose ρ, ρ' are χ^2 contraction rates:

$$\rho \geqslant \frac{\rho'}{(\text{congestion})\cdot(\text{max length})}$$

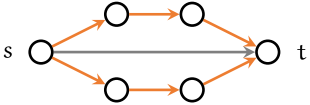
- If len $\leqslant 1$, can use any $\mathcal{D}_\phi$.
- Canonical paths: a few-to-one mapping enc from (s,t)-pairs whose path passes $x \to y$ to Ω:

$$\mu(s)\mu(t) \leqslant C \cdot \mu(\mathsf{enc}(s,t))Q(x,y)$$

Review

- P, P' reversible with same stationary distribution
- Comparison: route Q' through Q with low congestion and length.

$$\pi(\text{path} \mid X_0 = s, X_\ell = t)$$



Congestion

Suppose π is dist over paths and Q is ergodic flow. Congestion is

$$\max\left\{ \frac{\mathbb{P}_{\text{path}\sim\pi}[(x\to y)\in\text{path}]}{Q(x,y)} \;\middle|\; x \neq y \right\}$$

Lemma: comparison

Suppose ρ, ρ' are χ^2 contraction rates:

$$\rho \geqslant \frac{\rho'}{(\text{congestion})\cdot(\text{max length})}$$

- If len $\leqslant 1$, can use any $\mathcal{D}_\phi$.
- Canonical paths: a few-to-one mapping enc from (s,t)-pairs whose path passes $x \to y$ to Ω:

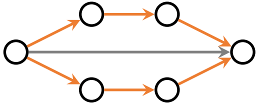
$$\mu(s)\mu(t) \leqslant C \cdot \mu(\mathsf{enc}(s,t))Q(x,y)$$

- If M-to-one, then cong $\leqslant CM$.

Review

- P, P' reversible with same stationary distribution
- Comparison: route Q' through Q with low congestion and length.

$$\pi(\text{path} \mid X_0 = s, X_\ell = t)$$



Congestion

Suppose π is dist over paths and Q is ergodic flow. Congestion is

$$\max\left\{ \frac{\mathbb{P}_{\text{path}\sim\pi}[(x\to y)\in\text{path}]}{Q(x,y)} \;\middle|\; x \neq y \right\}$$

Lemma: comparison

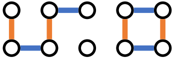
Suppose ρ, ρ' are χ^2 contraction rates:

$$\rho \geqslant \frac{\rho'}{(\text{congestion})\cdot(\text{max length})}$$

- If len $\leqslant 1$, can use any $\mathcal{D}_\phi$.
- Canonical paths: a few-to-one mapping enc from (s,t)-pairs whose path passes $x \to y$ to Ω:

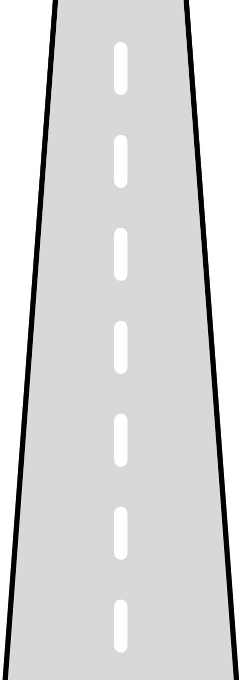
$$\mu(s)\mu(t) \leqslant C \cdot \mu(\mathsf{enc}(s,t))Q(x,y)$$

- If M-to-one, then cong $\leqslant CM$.
- Matching walks mix in $\mathsf{poly}(n)$.



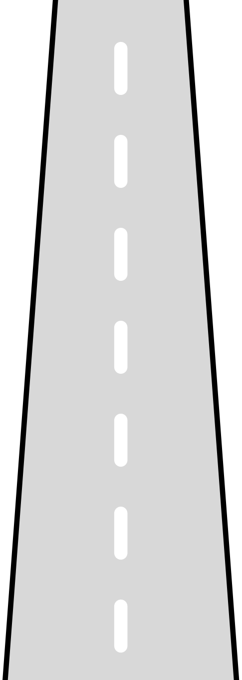
Perfect Matchings

- Monomer-dimer systems
- Log-concave sequences
- Bipartite graphs



Perfect Matchings

- Monomer-dimer systems
- Log-concave sequences
- Bipartite graphs

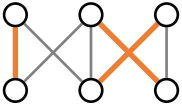


Monomer-dimer systems

- Markov chain on matchings mixes in $\mathsf{poly}(n)$ time [Jerrum-Sinclair'89].

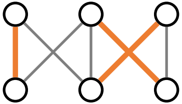
Monomer-dimer systems

- Markov chain on matchings mixes in $\mathrm{poly}(n)$ time [Jerrum-Sinclair'89].
- What about perfect matchings?



Monomer-dimer systems

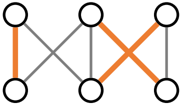
- Markov chain on matchings mixes in $\mathbf{poly}(n)$ time [Jerrum-Sinclair'89].
- What about perfect matchings?



- This is open. No strong indication/evidence either way! ☹

Monomer-dimer systems

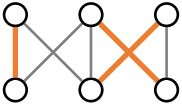
- Markov chain on matchings mixes in **poly**(n) time [Jerrum-Sinclair'89].
- What about perfect matchings?



- This is open. No strong indication/evidence either way! ☹
- However, for bipartite graphs, [Jerrum-Sinclair-Vigoda'04] showed we can approx sample/count in **poly**(n) time. ☺

Monomer-dimer systems

- Markov chain on matchings mixes in **poly**(n) time [Jerrum-Sinclair'89].
- What about perfect matchings?

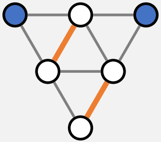


- This is open. No strong indication/evidence either way!
- However, for bipartite graphs, [Jerrum-Sinclair-Vigoda'04] showed we can approx sample/count in **poly**(n) time.

Monomer-dimer system

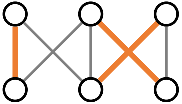
Prob of matching $\propto$

$$\prod_{e \in M} \underbrace{\lambda_e}_{\text{dimer}} \cdot \prod_{v \not\in M} \underbrace{z_v}_{\text{monomer}}$$



Monomer-dimer systems

- Markov chain on matchings mixes in **poly**(n) time [Jerrum-Sinclair'89].
- What about perfect matchings?

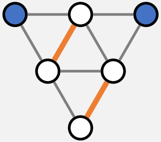


- This is open. No strong indication/evidence either way! 🙁
- However, for bipartite graphs, [Jerrum-Sinclair-Vigoda'04] showed we can approx sample/count in **poly**(n) time. 🙂

Monomer-dimer system

Prob of matching $\propto$

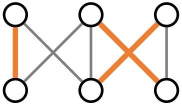
$$\prod_{e\in M} \underbrace{\lambda_e}_{\text{dimer}} \cdot \prod_{v \not\sim M} \underbrace{z_v}_{\text{monomer}}$$



- Monomer weights z_v can be absorbed into λ_e. So assume wlog that $z_v = 1$.

Monomer-dimer systems

- Markov chain on matchings mixes in **poly**(n) time [Jerrum-Sinclair'89].
- What about perfect matchings?

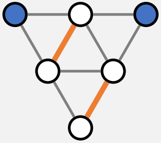


- This is open. No strong indication/evidence either way! 🙁
- However, for bipartite graphs, [Jerrum-Sinclair-Vigoda'04] showed we can approx sample/count in **poly**(n) time. 🙂

Monomer-dimer system

Prob of matching $\propto$

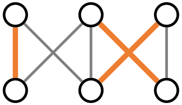
$$\prod_{e \in M} \underbrace{\lambda_e}_{\text{dimer}} \cdot \prod_{v \not\in M} \underbrace{z_v}_{\text{monomer}}$$



- Monomer weights z_v can be absorbed into λ_e. So assume wlog that $z_v = 1$.
- Mixing time is **poly**$(n, \lambda_{\max})$ [Jerrum-Sinclair] 🙂

Monomer-dimer systems

- Markov chain on matchings mixes in $\mathbf{poly}(n)$ time [Jerrum-Sinclair'89].
- What about perfect matchings?

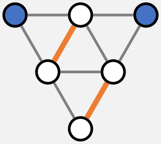


- This is open. No strong indication/evidence either way! ☹
- However, for bipartite graphs, [Jerrum-Sinclair-Vigoda'04] showed we can approx sample/count in $\mathbf{poly}(n)$ time. ☺

Monomer-dimer system

Prob of matching $\propto$

$$\underbrace{\prod_{e\in M}\lambda_e}_{\text{dimer}}\cdot\underbrace{\prod_{v\not\in M} z_v}_{\text{monomer}}$$



- Monomer weights z_v can be absorbed into λ_e. So assume wlog that $z_v = 1$.
- Mixing time is $\mathbf{poly}(n, \lambda_{\max})$ [Jerrum-Sinclair] ☺
- Sampling/counting possible in $\mathbf{poly}(n, \log \lambda_{\max})$ time on bipartite graphs [Jerrum-Sinclair-Vigoda]. ☺

Theorem [Jerrum-Sinclair]

Mixing time is $\mathsf{poly}(n, \lambda_{\mathsf{max}})$.

Theorem [Jerrum-Sinclair]

Mixing time is $\mathsf{poly}(n, \lambda_{\mathsf{max}})$.

Proof: for the $x \to y$ transition:

Theorem [Jerrum-Sinclair]

Mixing time is $\mathsf{poly}(n, \lambda_{\mathsf{max}})$.

Proof: for the $x \to y$ transition:

- Same encoding as before:

  $\mathsf{enc}(s, t) = s \oplus t \oplus x -$ couple edges

Theorem [Jerrum-Sinclair]

Mixing time is $\mathsf{poly}(n, \lambda_{\mathsf{max}})$.

Proof: for the $x \to y$ transition:

- Same encoding as before:

  $\mathsf{enc}(s, t) = s \oplus t \oplus x$ – couple edges
- Using notation $\lambda^S = \prod_{e \in S} \lambda_e$:

  $$\lambda^s \lambda^t \leqslant \underbrace{\mathsf{poly}(\lambda_{\mathsf{max}})}_{\text{couple edges}} \cdot \lambda^{\mathsf{enc}(s,t)} \lambda^x$$

Theorem [Jerrum-Sinclair]

Mixing time is $\mathrm{poly}(n, \lambda_{\mathrm{max}})$.

Proof: for the $x \to y$ transition:

- Same encoding as before:

  $\mathrm{enc}(s, t) = s \oplus t \oplus x$ – couple edges

- Using notation $\lambda^S = \prod_{e \in S} \lambda_e$:

  $$\lambda^s \lambda^t \leqslant \mathrm{poly}(\underset{\uparrow}{\lambda}_{\mathrm{max}}) \cdot \lambda^{\mathrm{enc}(s,t)} \lambda^x$$

- Similarly: couple edges

  $$\lambda^s \lambda^t \leqslant \mathrm{poly}(\lambda_{\mathrm{max}}) \cdot \lambda^{\mathrm{enc}(s,t)} \lambda^y$$

Theorem [Jerrum-Sinclair]

Mixing time is $\mathsf{poly}(n, \lambda_{\mathsf{max}})$.

Proof: for the $x \to y$ transition:

- Same encoding as before:

  $\mathsf{enc}(s, t) = s \oplus t \oplus x$ – couple edges

- Using notation $\lambda^S = \prod_{e \in S} \lambda_e$:

  $$\lambda^s \lambda^t \leqslant \underbrace{\mathsf{poly}(\lambda_{\mathsf{max}})}_{\text{couple edges}} \cdot \lambda^{\mathsf{enc}(s,t)} \lambda^x$$

- Similarly:

  $$\lambda^s \lambda^t \leqslant \mathsf{poly}(\lambda_{\mathsf{max}}) \cdot \lambda^{\mathsf{enc}(s,t)} \lambda^y$$

- Using Metropolis filter we get

  $$Q(x, y) \geqslant \frac{\min\{\mu(x), \mu(y)\}}{\mathsf{poly}(n)}$$

Theorem [Jerrum-Sinclair]

Mixing time is $\mathsf{poly}(n, \lambda_{\mathsf{max}})$.

Proof: for the $x \to y$ transition:

- Same encoding as before:

  $\mathsf{enc}(s, t) = s \oplus t \oplus x$ – couple edges

- Using notation $\lambda^S = \prod_{e \in S} \lambda_e$:

  $$\lambda^s \lambda^t \leqslant \underbrace{\mathsf{poly}(\lambda_{\mathsf{max}})}_{\text{couple edges}} \cdot \lambda^{\mathsf{enc}(s,t)} \lambda^x$$

- Similarly:

  $$\lambda^s \lambda^t \leqslant \mathsf{poly}(\lambda_{\mathsf{max}}) \cdot \lambda^{\mathsf{enc}(s,t)} \lambda^y$$

- Using Metropolis filter we get

  $$Q(x, y) \geqslant \frac{\mathsf{min}\{\mu(x), \mu(y)\}}{\mathsf{poly}(n)}$$

- So we have $\mu(s)\mu(t) \leqslant$

  $\mathsf{poly}(n, \lambda_{\mathsf{max}}) \cdot \mu(\mathsf{enc}(s, t)) Q(x, y)$

Theorem [Jerrum-Sinclair]

Mixing time is $\mathsf{poly}(n, \lambda_{\mathsf{max}})$.

Proof: for the $x \to y$ transition:

- Same encoding as before:

  $\mathsf{enc}(s, t) = s \oplus t \oplus x$ – couple edges

- Using notation $\lambda^S = \prod_{e \in S} \lambda_e$:

  $$\lambda^s \lambda^t \leqslant \mathsf{poly}(\lambda_{\mathsf{max}}) \cdot \lambda^{\mathsf{enc}(s,t)} \lambda^x$$

  couple edges

- Similarly:

  $$\lambda^s \lambda^t \leqslant \mathsf{poly}(\lambda_{\mathsf{max}}) \cdot \lambda^{\mathsf{enc}(s,t)} \lambda^y$$

- Using Metropolis filter we get

  $$Q(x, y) \geqslant \frac{\min\{\mu(x), \mu(y)\}}{\mathsf{poly}(n)}$$

- So we have $\mu(s)\mu(t) \leqslant$

  $$\mathsf{poly}(n, \lambda_{\mathsf{max}}) \cdot \mu(\mathsf{enc}(s, t)) Q(x, y)$$

- What if we want perfect matchings?

Theorem [Jerrum-Sinclair]

Mixing time is $\mathsf{poly}(n, \lambda_{\mathsf{max}})$.

Proof: for the $x \to y$ transition:

- Same encoding as before:

  $\mathsf{enc}(s, t) = s \oplus t \oplus x$ – couple edges

- Using notation $\lambda^S = \prod_{e \in S} \lambda_e$:

  $$\lambda^s \lambda^t \leqslant \underbrace{\mathsf{poly}(\lambda_{\mathsf{max}})}_{\text{couple edges}} \cdot \lambda^{\mathsf{enc}(s,t)} \lambda^x$$

- Similarly:

  $$\lambda^s \lambda^t \leqslant \mathsf{poly}(\lambda_{\mathsf{max}}) \cdot \lambda^{\mathsf{enc}(s,t)} \lambda^y$$

- Using Metropolis filter we get

  $$Q(x, y) \geqslant \frac{\mathsf{min}\{\mu(x), \mu(y)\}}{\mathsf{poly}(n)}$$

- So we have $\mu(s)\mu(t) \leqslant$

  $\mathsf{poly}(n, \lambda_{\mathsf{max}}) \cdot \mu(\mathsf{enc}(s, t)) Q(x, y)$

- What if we want perfect matchings?
- Idea 1: restrict chain to perfect and near-perfect matchings.

  one fewer edge

Theorem [Jerrum-Sinclair]

Mixing time is $\mathrm{poly}(n, \lambda_{\mathrm{max}})$.

Proof: for the $x \to y$ transition:

- Same encoding as before:

  $\mathrm{enc}(s, t) = s \oplus t \oplus x -$ couple edges
- Using notation $\lambda^S = \prod_{e \in S} \lambda_e$:

  $$\lambda^s \lambda^t \leqslant \underbrace{\mathrm{poly}(\lambda_{\mathrm{max}})}_{\text{couple edges}} \cdot \lambda^{\mathrm{enc}(s,t)} \lambda^x$$
- Similarly:

  $$\lambda^s \lambda^t \leqslant \mathrm{poly}(\lambda_{\mathrm{max}}) \cdot \lambda^{\mathrm{enc}(s,t)} \lambda^y$$
- Using Metropolis filter we get

  $$Q(x, y) \geqslant \frac{\min\{\mu(x), \mu(y)\}}{\mathrm{poly}(n)}$$
- So we have $\mu(s)\mu(t) \leqslant$

  $\mathrm{poly}(n, \lambda_{\mathrm{max}}) \cdot \mu(\mathrm{enc}(s, t)) Q(x, y)$

- What if we want perfect matchings?
- Idea 1: restrict chain to perfect and near-perfect matchings.

  one fewer edge
- Idea 2: set $\lambda_e = \lambda$ very large.

Theorem [Jerrum-Sinclair]

Mixing time is $\mathsf{poly}(n, \lambda_{\mathsf{max}})$.

Proof: for the $x \to y$ transition:

- Same encoding as before:

  $\mathsf{enc}(s, t) = s \oplus t \oplus x$ – couple edges

- Using notation $\lambda^S = \prod_{e \in S} \lambda_e$:

  $$\lambda^s \lambda^t \leqslant \underbrace{\mathsf{poly}(\lambda_{\mathsf{max}})}_{\text{couple edges}} \cdot \lambda^{\mathsf{enc}(s,t)} \lambda^x$$

- Similarly:

  $$\lambda^s \lambda^t \leqslant \mathsf{poly}(\lambda_{\mathsf{max}}) \cdot \lambda^{\mathsf{enc}(s,t)} \lambda^y$$

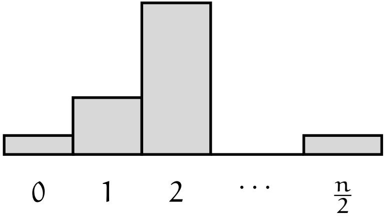
- Using Metropolis filter we get

  $$Q(x, y) \geqslant \frac{\min\{\mu(x), \mu(y)\}}{\mathsf{poly}(n)}$$

- So we have $\mu(s)\mu(t) \leqslant$

  $\mathsf{poly}(n, \lambda_{\mathsf{max}}) \cdot \mu(\mathsf{enc}(s, t)) Q(x, y)$

- What if we want perfect matchings?
- Idea 1: restrict chain to perfect and $\underbrace{\text{near-perfect}}_{\text{one fewer edge}}$ matchings.
- Idea 2: set $\lambda_e = \lambda$ very large.
- Dist of matching size:



Theorem [Jerrum-Sinclair]

Mixing time is $\text{poly}(n, \lambda_{\max})$.

Proof: for the $x \to y$ transition:

- Same encoding as before:

  $\text{enc}(s,t) = s \oplus t \oplus x -$ couple edges

- Using notation $\lambda^S = \prod_{e \in S} \lambda_e$:

  $$\lambda^s \lambda^t \leqslant \underbrace{\text{poly}(\lambda_{\max})}_{\text{couple edges}} \cdot \lambda^{\text{enc}(s,t)} \lambda^x$$

- Similarly:

  $$\lambda^s \lambda^t \leqslant \text{poly}(\lambda_{\max}) \cdot \lambda^{\text{enc}(s,t)} \lambda^y$$

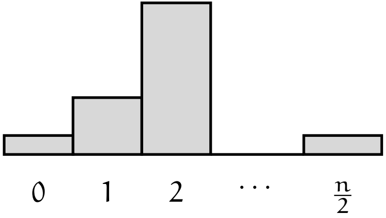
- Using Metropolis filter we get

  $$Q(x,y) \geqslant \frac{\min\{\mu(x), \mu(y)\}}{\text{poly}(n)}$$

- So we have $\mu(s)\mu(t) \leqslant$

  $\text{poly}(n, \lambda_{\max}) \cdot \mu(\text{enc}(s,t)) Q(x,y)$

- What if we want perfect matchings?
- Idea 1: restrict chain to perfect and $\underbrace{\text{near-perfect}}_{\text{one fewer edge}}$ matchings.
- Idea 2: set $\lambda_e = \lambda$ very large.
- Dist of matching size:



- If $\lambda^k \cdot \#(k\text{-matchings})$ maximized for $k = \frac{n}{2}$, use rejection sampling.

Fact: log-concavity of matchings

If m_k is #(k-matchings), then

$$\frac{m_0}{m_1} \leqslant \frac{m_1}{m_2} \leqslant \cdots \leqslant \frac{m_{n/2-1}}{m_{n/2}}$$

Fact: log-concavity of matchings

If m_k is $\#(k\text{-matchings})$, then

$$\frac{m_0}{m_1} \leqslant \frac{m_1}{m_2} \leqslant \cdots \leqslant \frac{m_{n/2-1}}{m_{n/2}}$$

- So just need to set

$$\lambda \geqslant m_{n/2-1}/m_{n/2}$$

Fact: log-concavity of matchings

If m_k is $\#(k\text{-matchings})$, then

$$\frac{m_0}{m_1} \leqslant \frac{m_1}{m_2} \leqslant \cdots \leqslant \frac{m_{n/2-1}}{m_{n/2}}$$

- So just need to set

$$\lambda \geqslant m_{n/2-1}/m_{n/2}$$

- Corollary: if

$$m_{n/2-1} \leqslant \text{poly}(n) \cdot m_{n/2}$$

  can sample perfect matchings. ☺

Fact: log-concavity of matchings

If m_k is $\#(k\text{-matchings})$, then

$$\frac{m_0}{m_1} \leqslant \frac{m_1}{m_2} \leqslant \cdots \leqslant \frac{m_{n/2-1}}{m_{n/2}}$$

- So just need to set
$$\lambda \geqslant m_{n/2-1}/m_{n/2}$$
- Corollary: if
$$m_{n/2-1} \leqslant \mathsf{poly}(n) \cdot m_{n/2}$$
can sample perfect matchings. ☺
- Note: same cond for idea 1.

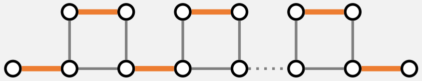
Fact: log-concavity of matchings

If m_k is $\#$(k-matchings), then

$$\frac{m_0}{m_1} \leqslant \frac{m_1}{m_2} \leqslant \cdots \leqslant \frac{m_{n/2-1}}{m_{n/2}}$$

- So just need to set
$$\lambda \geqslant m_{n/2-1}/m_{n/2}$$
- Corollary: if
$$m_{n/2-1} \leqslant \mathrm{poly}(n) \cdot m_{n/2}$$
can sample perfect matchings. ☺
- Note: same cond for idea 1.

Bad example: chain of boxes



Fact: log-concavity of matchings

If m_k is #(k-matchings), then

$$\frac{m_0}{m_1} \leqslant \frac{m_1}{m_2} \leqslant \cdots \leqslant \frac{m_{n/2-1}}{m_{n/2}}$$

- So just need to set

$$\lambda \geqslant m_{n/2-1}/m_{n/2}$$

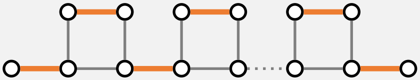
- Corollary: if

$$m_{n/2-1} \leqslant \mathrm{poly}(n) \cdot m_{n/2}$$

can sample perfect matchings. 🙂

- Note: same cond for idea 1.

Bad example: chain of boxes



- There are bad examples. 🙁

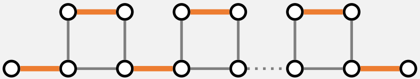
Fact: log-concavity of matchings

If m_k is #(k-matchings), then

$$\frac{m_0}{m_1} \leqslant \frac{m_1}{m_2} \leqslant \cdots \leqslant \frac{m_{n/2-1}}{m_{n/2}}$$

- So just need to set
$$\lambda \geqslant m_{n/2-1}/m_{n/2}$$
- Corollary: if
$$m_{n/2-1} \leqslant \mathrm{poly}(n) \cdot m_{n/2}$$
can sample perfect matchings. 🙂
- Note: same cond for idea 1.

Bad example: chain of boxes



- There are bad examples. 🙁
- In chain of boxes, we have 1 perfect and $2^{\Omega(n)}$ near-perfect matchings.

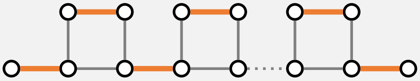
Fact: log-concavity of matchings

If m_k is #(k-matchings), then

$$\frac{m_0}{m_1} \leqslant \frac{m_1}{m_2} \leqslant \cdots \leqslant \frac{m_{n/2-1}}{m_{n/2}}$$

- So just need to set
$$\lambda \geqslant m_{n/2-1}/m_{n/2}$$
- Corollary: if
$$m_{n/2-1} \leqslant \mathrm{poly}(n) \cdot m_{n/2}$$
can sample perfect matchings.
- Note: same cond for idea 1.

Bad example: chain of boxes



- There are bad examples.
- In chain of boxes, we have 1 perfect and $2^{\Omega(n)}$ near-perfect matchings.
- Exercise: modify chain of boxes to get slow mixing for idea 1.

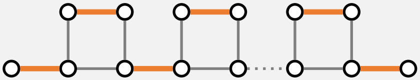
Fact: log-concavity of matchings

If m_k is $\#(k\text{-matchings})$, then

$$\frac{m_0}{m_1} \leqslant \frac{m_1}{m_2} \leqslant \cdots \leqslant \frac{m_{n/2-1}}{m_{n/2}}$$

- So just need to set
$$\lambda \geqslant m_{n/2-1}/m_{n/2}$$
- Corollary: if
$$m_{n/2-1} \leqslant \text{poly}(n) \cdot m_{n/2}$$
can sample perfect matchings. 🙂
- Note: same cond for idea 1.

Bad example: chain of boxes



- There are bad examples. ☹️
- In chain of boxes, we have 1 perfect and $2^{\Omega(n)}$ near-perfect matchings.
- Exercise: modify chain of boxes to get slow mixing for idea 1.
- Idea: since there can be many more near-perfect matchings, why not reweigh matchings based on size?

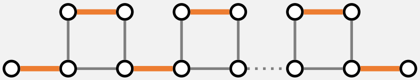
Fact: log-concavity of matchings

If m_k is $\#$(k-matchings), then

$$\frac{m_0}{m_1} \leqslant \frac{m_1}{m_2} \leqslant \cdots \leqslant \frac{m_{n/2-1}}{m_{n/2}}$$

- So just need to set
$$\lambda \geqslant m_{n/2-1}/m_{n/2}$$
- Corollary: if
$$m_{n/2-1} \leqslant \mathrm{poly}(n) \cdot m_{n/2}$$
can sample perfect matchings. 🙂
- Note: same cond for idea 1.

Bad example: chain of boxes



- There are bad examples. ☹️
- In chain of boxes, we have 1 perfect and $2^{\Omega(n)}$ near-perfect matchings.
- Exercise: modify chain of boxes to get slow mixing for idea 1.
- Idea: since there can be many more near-perfect matchings, why not reweigh matchings based on size?
- [Jerrum-Sinclair-Vigoda'04] showed this works on bipartite graphs.

- Let Ω_S be the class of matchings whose monomers are S. Example: $\Omega_\emptyset$ is perfect matchings, and $\Omega_{\{u,v\}}$ matchings that only miss u, v.

- Let Ω_S be the class of matchings whose monomers are S. Example: $\Omega_\emptyset$ is perfect matchings, and $\Omega_{\{u,v\}}$ matchings that only miss u, v.
- Let λ^M denote monomer-dimer weight of M:

$$\lambda^M = \prod_{e \in M} \lambda_e$$

- Let Ω_S be the class of matchings whose monomers are S. Example: $\Omega_\emptyset$ is perfect matchings, and $\Omega_{\{u,v\}}$ matchings that only miss u, v.
- Let λ^M denote monomer-dimer weight of M:
$$\lambda^M = \prod_{e \in M} \lambda_e$$
- We get weights for each class:
$$\lambda(\Omega_S) = \sum_{M \in \Omega_S} \lambda^M$$

- Let Ω_S be the class of matchings whose monomers are S. Example: $\Omega_\emptyset$ is perfect matchings, and $\Omega_{\{u,v\}}$ matchings that only miss u, v.
- Let λ^M denote monomer-dimer weight of M:

$$\lambda^M = \prod_{e \in M} \lambda_e$$

- We get weights for each class:

$$\lambda(\Omega_S) = \sum_{M \in \Omega_S} \lambda^M$$

- Define modified distribution on matchings:

$$\mu(M) \propto \frac{\lambda^M}{\lambda(\Omega_{\text{monomers}(M)})}$$

- Let Ω_S be the class of matchings whose monomers are S. Example: $\Omega_\emptyset$ is perfect matchings, and $\Omega_{\{u,v\}}$ matchings that only miss u, v.
- Let λ^M denote monomer-dimer weight of M:
$$\lambda^M = \prod_{e \in M} \lambda_e$$
- We get weights for each class:
$$\lambda(\Omega_S) = \sum_{M \in \Omega_S} \lambda^M$$
- Define modified distribution on matchings:
$$\mu(M) \propto \frac{\lambda^M}{\lambda(\Omega_{\text{monomers}(M)})}$$
- Note: $\mu(\Omega_S)$ is the same for all S.

- Let Ω_S be the class of matchings whose monomers are S. Example: $\Omega_\emptyset$ is perfect matchings, and $\Omega_{\{u,v\}}$ matchings that only miss u, v.
- Let λ^M denote monomer-dimer weight of M:
$$\lambda^M = \prod_{e \in M} \lambda_e$$
- We get weights for each class:
$$\lambda(\Omega_S) = \sum_{M \in \Omega_S} \lambda^M$$
- Define modified distribution on matchings:
$$\mu(M) \propto \frac{\lambda^M}{\lambda(\Omega_{\text{monomers}(M)})}$$
- Note: $\mu(\Omega_S)$ is the same for all S.

Theorem [Jerrum-Sinclair-Vigoda'04]

If P is Metropolis walk restricted to perfect and near-perfect matchings weighted $\propto \mu$, and graph is bipartite

$$t_{\text{mix}}(P, \mathbb{1}_{\text{max-weight } M}) = \text{poly}(n)$$

- Let Ω_S be the class of matchings whose monomers are S. Example: $\Omega_\emptyset$ is perfect matchings, and $\Omega_{\{u,v\}}$ matchings that only miss u, v.
- Let λ^M denote monomer-dimer weight of M:
$$\lambda^M = \prod_{e \in M} \lambda_e$$
- We get weights for each class:
$$\lambda(\Omega_S) = \sum_{M \in \Omega_S} \lambda^M$$
- Define modified distribution on matchings:
$$\mu(M) \propto \frac{\lambda^M}{\lambda(\Omega_{\text{monomers}(M)})}$$
- Note: $\mu(\Omega_S)$ is the same for all S.

Theorem [Jerrum-Sinclair-Vigoda'04]

If P is Metropolis walk restricted to perfect and near-perfect matchings weighted $\propto \mu$, and graph is bipartite

$$t_{\text{mix}}(P, \mathbb{1}_{\text{max-weight } M}) = \text{poly}(n)$$

- We need max-weight M as start to ensure $\log \chi^2(\nu_0 \parallel \mu) = \text{poly}(n)$.

- Let Ω_S be the class of matchings whose monomers are S. Example: $\Omega_\emptyset$ is perfect matchings, and $\Omega_{\{u,v\}}$ matchings that only miss u, v.
- Let λ^M denote monomer-dimer weight of M:

$$\lambda^M = \prod_{e \in M} \lambda_e$$

- We get weights for each class:

$$\lambda(\Omega_S) = \sum_{M \in \Omega_S} \lambda^M$$

- Define modified distribution on matchings:

$$\mu(M) \propto \frac{\lambda^M}{\lambda(\Omega_{\text{monomers}(M)})}$$

- Note: $\mu(\Omega_S)$ is the same for all S.

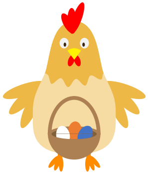
Theorem [Jerrum-Sinclair-Vigoda'04]

If P is Metropolis walk restricted to perfect and near-perfect matchings weighted $\propto \mu$, and graph is bipartite

$$t_{\text{mix}}(P, \mathbb{1}_{\text{max-weight } M}) = \text{poly}(n)$$

- We need max-weight M as start to ensure $\log \chi^2(\nu_0 \parallel \mu) = \text{poly}(n)$.

The Chicken-and-Egg Problem



- Let Ω_S be the class of matchings whose monomers are S. Example: $\Omega_\emptyset$ is perfect matchings, and $\Omega_{\{u,v\}}$ matchings that only miss u, v.
- Let λ^M denote monomer-dimer weight of M:
$$\lambda^M = \prod_{e\in M} \lambda_e$$
- We get weights for each class:
$$\lambda(\Omega_S) = \sum_{M\in\Omega_S} \lambda^M$$
- Define modified distribution on matchings:
$$\mu(M) \propto \frac{\lambda^M}{\lambda(\Omega_{\text{monomers}(M)})}$$
- Note: $\mu(\Omega_S)$ is the same for all S.

Theorem [Jerrum-Sinclair-Vigoda'04]

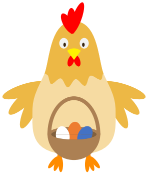
If P is Metropolis walk restricted to perfect and near-perfect matchings weighted $\propto \mu$, and graph is bipartite

$$t_{\text{mix}}(P, \mathbb{1}_{\text{max-weight } M}) = \text{poly}(n)$$

- We need max-weight M as start to ensure $\log \chi^2(\nu_0 \parallel \mu) = \text{poly}(n)$.

The Chicken-and-Egg Problem

- How can we compute $\lambda(\Omega_S)$?



- Let Ω_S be the class of matchings whose monomers are S. Example: $\Omega_\emptyset$ is perfect matchings, and $\Omega_{\{u,v\}}$ matchings that only miss u, v.
- Let λ^M denote monomer-dimer weight of M:

$$\lambda^M = \prod_{e \in M} \lambda_e$$

- We get weights for each class:

$$\lambda(\Omega_S) = \sum_{M \in \Omega_S} \lambda^M$$

- Define modified distribution on matchings:

$$\mu(M) \propto \frac{\lambda^M}{\lambda(\Omega_{\text{monomers}(M)})}$$

- Note: $\mu(\Omega_S)$ is the same for all S.

Theorem [Jerrum-Sinclair-Vigoda'04]

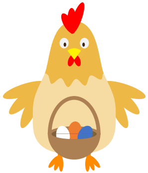
If P is Metropolis walk restricted to perfect and near-perfect matchings weighted $\propto \mu$, and graph is bipartite

$$t_{\text{mix}}(P, \mathbb{1}_{\text{max-weight } M}) = \text{poly}(n)$$

- We need max-weight M as start to ensure $\log \chi^2(\nu_0 \parallel \mu) = \text{poly}(n)$.

The Chicken-and-Egg Problem

- How can we compute $\lambda(\Omega_S)$?
- By sampling.



- Let Ω_S be the class of matchings whose monomers are S. Example: $\Omega_\emptyset$ is perfect matchings, and $\Omega_{\{u,v\}}$ matchings that only miss u, v.
- Let λ^M denote monomer-dimer weight of M:
$$\lambda^M = \prod_{e \in M} \lambda_e$$
- We get weights for each class:
$$\lambda(\Omega_S) = \sum_{M \in \Omega_S} \lambda^M$$
- Define modified distribution on matchings:
$$\mu(M) \propto \frac{\lambda^M}{\lambda(\Omega_{\text{monomers}(M)})}$$
- Note: $\mu(\Omega_S)$ is the same for all S.

Theorem [Jerrum-Sinclair-Vigoda'04]

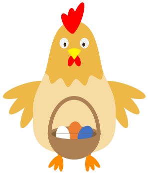
If P is Metropolis walk restricted to perfect and near-perfect matchings weighted $\propto \mu$, and graph is bipartite

$$t_{\text{mix}}(P, \mathbb{1}_{\text{max-weight } M}) = \text{poly}(n)$$

- We need max-weight M as start to ensure $\log \chi^2(\nu_0 \parallel \mu) = \text{poly}(n)$.

The Chicken-and-Egg Problem

- How can we compute $\lambda(\Omega_S)$?
- By sampling.
- How to sample?



- Let Ω_S be the class of matchings whose monomers are S. Example: $\Omega_\emptyset$ is perfect matchings, and $\Omega_{\{u,v\}}$ matchings that only miss u, v.
- Let λ^M denote monomer-dimer weight of M:
$$\lambda^M = \prod_{e\in M} \lambda_e$$
- We get weights for each class:
$$\lambda(\Omega_S) = \sum_{M\in\Omega_S} \lambda^M$$
- Define modified distribution on matchings:
$$\mu(M) \propto \frac{\lambda^M}{\lambda(\Omega_{\text{monomers}(M)})}$$
- Note: $\mu(\Omega_S)$ is the same for all S.

Theorem [Jerrum-Sinclair-Vigoda'04]

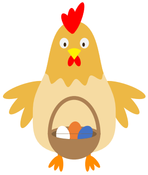
If P is Metropolis walk restricted to perfect and near-perfect matchings weighted $\propto \mu$, and graph is bipartite

$$t_{\text{mix}}(P, \mathbb{1}_{\text{max-weight } M}) = \text{poly}(n)$$

- We need max-weight M as start to ensure $\log \chi^2(\nu_0 \parallel \mu) = \text{poly}(n)$.

The Chicken-and-Egg Problem

- How can we compute $\lambda(\Omega_S)$?
- By sampling.
- How to sample?
- Use counting. ☹



- Resolving the chicken-and-egg problem: gradual change.

- Resolving the chicken-and-egg problem: gradual change.

Theorem

Chain mixes fast even if $\lambda(\Omega_S)$ are replaced by approximations in μ.

say factor 10 approx

- Resolving the chicken-and-egg problem: gradual change.

Theorem

Chain mixes fast even if $\lambda(\Omega_S)$ are replaced by approximations in μ.

say factor 10 approx

- Start with easy case. Take $G = K_{n/2,n/2}$, and $\lambda_e = 1$.

- Resolving the chicken-and-egg problem: gradual change.

Theorem

Chain mixes fast even if $\lambda(\Omega_S)$ are replaced by approximations in μ.

say factor 10 approx

- Start with easy case. Take $G = K_{n/2,n/2}$, and $\lambda_e = 1$.
- Slowly change λ_es:

$$\lambda^{(0)} \to \lambda^{(1)} \to \cdots \to \lambda^{(t)}$$

by $1 \pm 1/n$ each time

- Resolving the chicken-and-egg problem: gradual change.

Theorem

Chain mixes fast even if $\lambda(\Omega_S)$ are replaced by approximations in μ.

say factor 10 approx

- Start with easy case. Take $G = K_{n/2,n/2}$, and $\lambda_e = 1$.
- Slowly change λ_es:

$$\lambda^{(0)} \to \lambda^{(1)} \to \cdots \to \lambda^{(t)}$$

by $1 \pm 1/n$ each time

- Use Markov chain for each $\lambda^{(i)}$ to estimate $\lambda^{(i)}(\Omega_S)$ for $|S| \leqslant 2$.

- Resolving the chicken-and-egg problem: gradual change.

Theorem

Chain mixes fast even if $\lambda(\Omega_S)$ are replaced by approximations in μ.

say factor 10 approx

- Start with easy case. Take $G = K_{n/2,n/2}$, and $\lambda_e = 1$.
- Slowly change λ_es:

$$\lambda^{(0)} \to \lambda^{(1)} \to \cdots \to \lambda^{(t)}$$

by $1 \pm 1/n$ each time

- Use Markov chain for each $\lambda^{(i)}$ to estimate $\lambda^{(i)}(\Omega_S)$ for $|S| \leqslant 2$.
- Use estimates to define next μ.

- Resolving the chicken-and-egg problem: gradual change.

Theorem

Chain mixes fast even if $\lambda(\Omega_S)$ are replaced by approximations in μ.

say factor 10 approx

- Start with easy case. Take $G = K_{n/2,n/2}$, and $\lambda_e = 1$.
- Slowly change λ_es:

$$\lambda^{(0)} \to \lambda^{(1)} \to \cdots \to \lambda^{(t)}$$

by $1 \pm 1/n$ each time

- Use Markov chain for each $\lambda^{(i)}$ to estimate $\lambda^{(i)}(\Omega_S)$ for $|S| \leqslant 2$.
- Use estimates to define next μ.

- Note: $(\lambda_e = 0) \approx (\lambda_e = \exp(-n^2))$

- Resolving the chicken-and-egg problem: gradual change.

Theorem

Chain mixes fast even if $\lambda(\Omega_S)$ are replaced by approximations in μ.

say factor 10 approx

- Start with easy case. Take $G = K_{n/2,n/2}$, and $\lambda_e = 1$.
- Slowly change λ_es:

$$\lambda^{(0)} \to \lambda^{(1)} \to \cdots \to \lambda^{(t)}$$

by $1 \pm 1/n$ each time

- Use Markov chain for each $\lambda^{(i)}$ to estimate $\lambda^{(i)}(\Omega_S)$ for $|S| \leqslant 2$.
- Use estimates to define next μ.
- Note: $(\lambda_e = 0) \approx (\lambda_e = \exp(-n^2))$
- It just remains to prove fast mixing.

- Resolving the chicken-and-egg problem: gradual change.

Theorem

Chain mixes fast even if $\lambda(\Omega_S)$ are replaced by approximations in μ.

say factor 10 approx

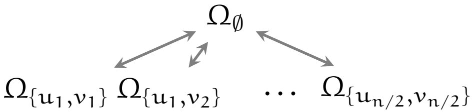
- Start with easy case. Take $G = K_{n/2,n/2}$, and $\lambda_e = 1$.
- Slowly change λ_es:

$$\lambda^{(0)} \to \lambda^{(1)} \to \cdots \to \lambda^{(t)}$$

by $1 \pm 1/n$ each time

- Use Markov chain for each $\lambda^{(i)}$ to estimate $\lambda^{(i)}(\Omega_S)$ for $|S| \leqslant 2$.
- Use estimates to define next μ.

- Note: $(\lambda_e = 0) \approx (\lambda_e = \exp(-n^2))$
- It just remains to prove fast mixing.
- We use canonical paths. Enough to consider $s \in \Omega_{\{u,v\}}$ and $t \in \Omega_\emptyset$.



- Resolving the chicken-and-egg problem: gradual change.

Theorem

Chain mixes fast even if $\lambda(\Omega_S)$ are replaced by approximations in μ.

say factor 10 approx

- Start with easy case. Take $G = K_{n/2,n/2}$, and $\lambda_e = 1$.
- Slowly change λ_es:

$$\lambda^{(0)} \to \lambda^{(1)} \to \cdots \to \lambda^{(t)}$$

by $1 \pm 1/n$ each time

- Use Markov chain for each $\lambda^{(i)}$ to estimate $\lambda^{(i)}(\Omega_S)$ for $|S| \leqslant 2$.
- Use estimates to define next μ.

- Note: $(\lambda_e = 0) \approx (\lambda_e = \exp(-n^2))$
- It just remains to prove fast mixing.
- We use canonical paths. Enough to consider $s \in \Omega_{\{u,v\}}$ and $t \in \Omega_\emptyset$.

$$\Omega_\emptyset$$

$$\Omega_{\{u_1,v_1\}} \quad \Omega_{\{u_1,v_2\}} \quad \cdots \quad \Omega_{\{u_{n/2},v_{n/2}\}}$$

- Assume $\lambda(\Omega_S)$ is accurate, because the inequality

$$\mu(s)\mu(t) \leqslant C\mu(\mathrm{enc}(s,t))Q(x,y)$$

is robust to approximation.

- Resolving the chicken-and-egg problem: gradual change.

Theorem

Chain mixes fast even if $\lambda(\Omega_S)$ are replaced by approximations in μ.

say factor 10 approx

- Start with easy case. Take $G = K_{n/2,n/2}$, and $\lambda_e = 1$.
- Slowly change λ_es:

$$\lambda^{(0)} \to \lambda^{(1)} \to \cdots \to \lambda^{(t)}$$

by $1 \pm 1/n$ each time

- Use Markov chain for each $\lambda^{(i)}$ to estimate $\lambda^{(i)}(\Omega_S)$ for $|S| \leqslant 2$.
- Use estimates to define next μ.

- Note: $(\lambda_e = 0) \approx (\lambda_e = \exp(-n^2))$
- It just remains to prove fast mixing.
- We use canonical paths. Enough to consider $s \in \Omega_{\{u,v\}}$ and $t \in \Omega_{\emptyset}$.

$$\begin{tikzcd}
 & \Omega_{\emptyset} \arrow[dl, leftrightarrow] \arrow[d, leftrightarrow] \arrow[drr, leftrightarrow] & & \\
\Omega_{\{u_1,v_1\}} & \Omega_{\{u_1,v_2\}} & \cdots & \Omega_{\{u_{n/2},v_{n/2}\}}
\end{tikzcd}$$

- Assume $\lambda(\Omega_S)$ is accurate, because the inequality

$$\mu(s)\mu(t) \leqslant C\mu(\mathrm{enc}(s,t))Q(x,y)$$

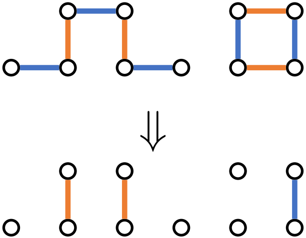
is robust to approximation.

- Use the same encoding as before:

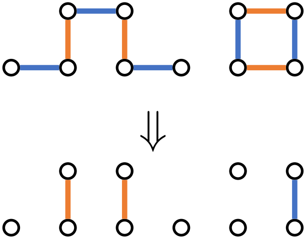
$$\mathrm{enc}(s,t) = s \oplus t \oplus x - \text{couple edges}$$

- Traverse alternating path first.
  Ensures all x on the st-path are perfect/near-perfect.

- Traverse alternating path first. Ensures all x on the st-path are perfect/near-perfect.
- Issue: encoding might not be perfect/near-perfect:



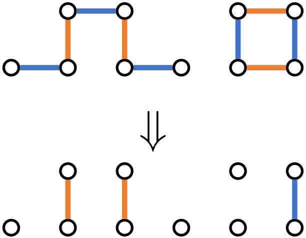
- Traverse alternating path first. Ensures all x on the st-path are perfect/near-perfect.
- Issue: encoding might not be perfect/near-perfect:



- This is fine! We still get cong $\leqslant$ poly(n) because

$$\sum_{|S|\leqslant 4} \mu(\Omega_S) \leqslant \text{poly}(n) \cdot \sum_{|S|\leqslant 2} \mu(\Omega_S)$$

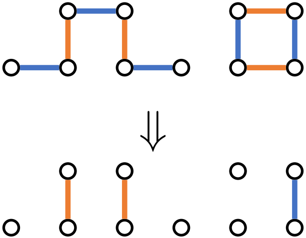
- Traverse alternating path first. Ensures all x on the st-path are perfect/near-perfect.
- Issue: encoding might not be perfect/near-perfect:



- This is fine! We still get cong $\leqslant \mathsf{poly}(n)$ because
$$\sum_{|S|\leqslant 4} \mu(\Omega_S) \leqslant \mathsf{poly}(n) \cdot \sum_{|S|\leqslant 2} \mu(\Omega_S)$$

- We just need to show $\mu(s)\mu(t) \leqslant$
$$\mathsf{poly}(n) \cdot \min\{\mu(x), \mu(y)\} \cdot \mu(\mathsf{enc})$$

- Traverse alternating path first. Ensures all x on the st-path are perfect/near-perfect.
- Issue: encoding might not be perfect/near-perfect:



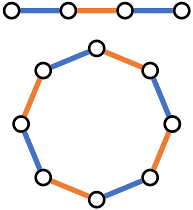
- This is fine! We still get cong $\leqslant$ poly(n) because

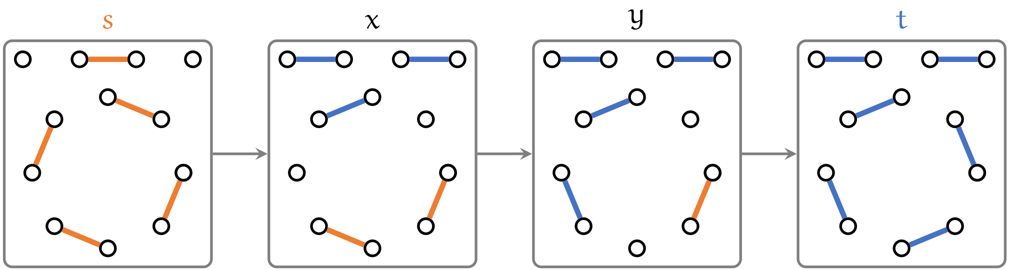
$$\sum_{|S|\leqslant 4} \mu(\Omega_S) \leqslant \text{poly}(n) \cdot \sum_{|S|\leqslant 2} \mu(\Omega_S)$$

- We just need to show $\mu(s)\mu(t) \leqslant$

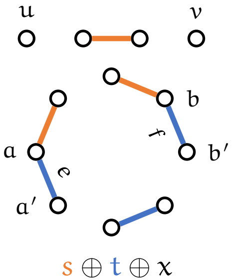
$$\text{poly}(n) \cdot \min\{\mu(x), \mu(y)\} \cdot \mu(\text{enc})$$

- It's a bit of case analysis, but hardest case is in the middle of unraveling a cycle:

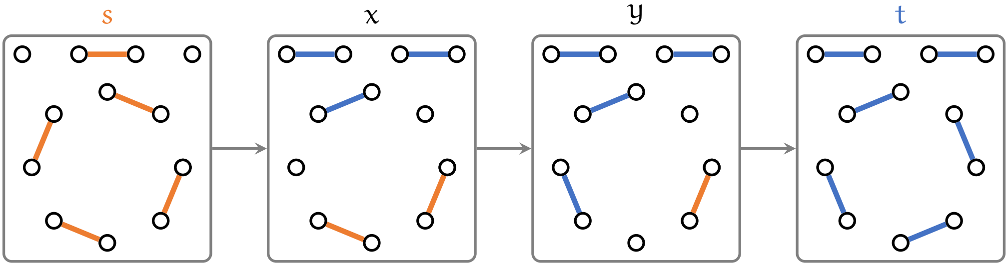




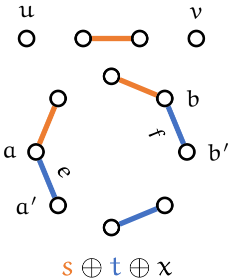
s
x
y
t

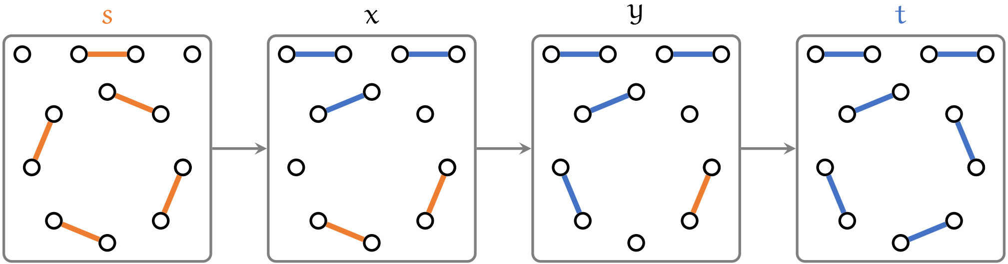


u
v
b
f
a
e
b′
a′
s ⊕ t ⊕ x

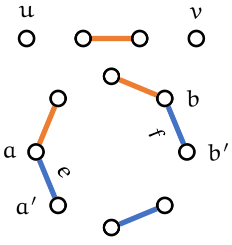


$\mathsf{enc} = s \oplus t \oplus x - e - f$





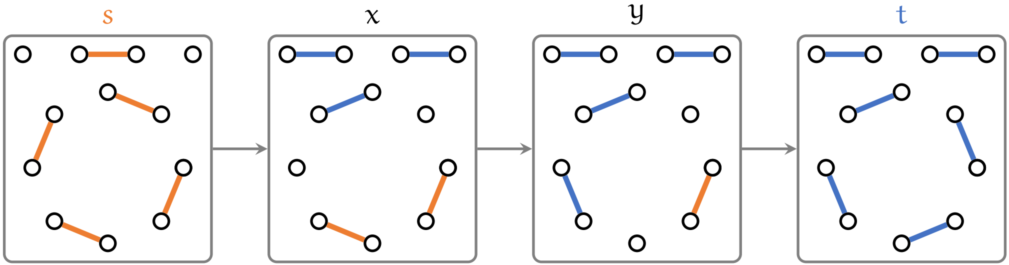
$$\mathsf{enc} = s \oplus t \oplus x - e - f$$



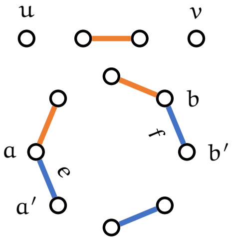
$s \oplus t \oplus x$

- Note that $\lambda^s \lambda^t = \lambda_e \lambda_f \lambda^x \lambda^{\mathsf{enc}}$. Let e's endpoints be a, a' and f's endpoints be b, b'. Prove: ⟵ via injective map

$$\lambda(\underset{\substack{\uparrow\\ t}}{\Omega_{\emptyset}})\lambda(\underset{\substack{\uparrow\\ s}}{\Omega_{\{u,v\}}}) \geqslant \frac{1}{\mathsf{poly}(n)} \cdot \lambda_e \lambda_f \lambda(\underset{\substack{\uparrow\\ x}}{\Omega_{\{a,b\}}})\lambda(\underset{\substack{\uparrow\\ \mathsf{enc}}}{\Omega_{\{u,v,a',b'\}}})$$



$$\mathsf{enc} = s \oplus t \oplus x - e - f$$



$s \oplus t \oplus x$

- Note that $\lambda^s \lambda^t = \lambda_e \lambda_f \lambda^x \lambda^{\mathsf{enc}}$. Let e's endpoints be a, a' and f's endpoints be b, b'. Prove: $\longleftarrow$ via injective map

$$\lambda(\underset{\overset{\uparrow}{t}}{\Omega_\emptyset})\lambda(\underset{\overset{\uparrow}{s}}{\Omega_{\{u,v\}}}) \geqslant \frac{1}{\mathsf{poly}(n)} \cdot \lambda_e \lambda_f \lambda(\underset{\overset{\uparrow}{x}}{\Omega_{\{a,b\}}})\lambda(\underset{\overset{\uparrow}{\mathsf{enc}}}{\Omega_{\{u,v,a',b'\}}})$$

- Thus $\mu(s)\mu(t) \leqslant \mathsf{poly}(n) \cdot \mu(x)\mu(\mathsf{enc})$. Similar ineqs yield $\mu(s)\mu(t) \leqslant \mathsf{poly}(n) \cdot \mu(y)\mu(\mathsf{enc})$. So cong $\leqslant \mathsf{poly}(n)$.