What can be *sampled* locally?

Yitong Yin
Nanjing University

Joint work with: Weiming Feng (Nanjing University)
Yuxin Sun (Nanjing University)

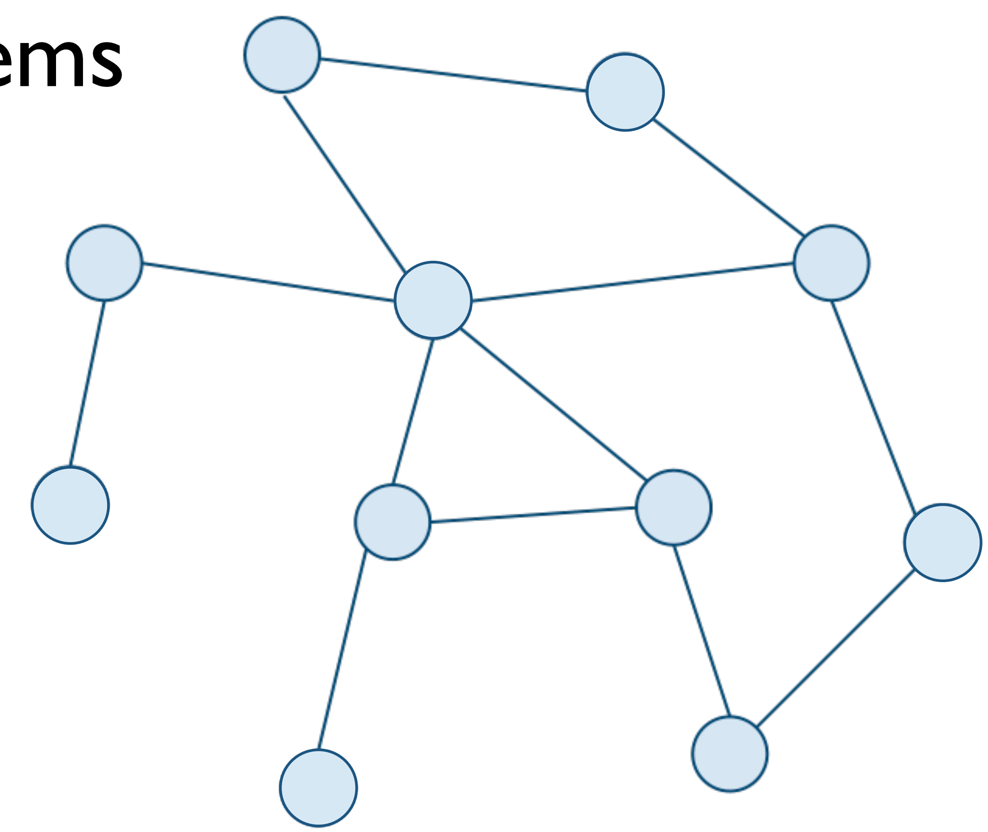
Local Computation

the *LOCAL* model [Linial '87]:

- In t rounds: each node can collect information up to distance t.

Locally Checkable Labeling (*LCL*) problems
[Noar, Stockmeyer '93] :

- CSPs with local constraints.
- Construct a feasible solution:
  vertex/edge coloring, Lovász local lemma
  - Find local optimum: MIS, MM
  - Approximate global optimum:
    maximum matching, minimum vertex cover, minimum dominating set



network $G(V,E)$

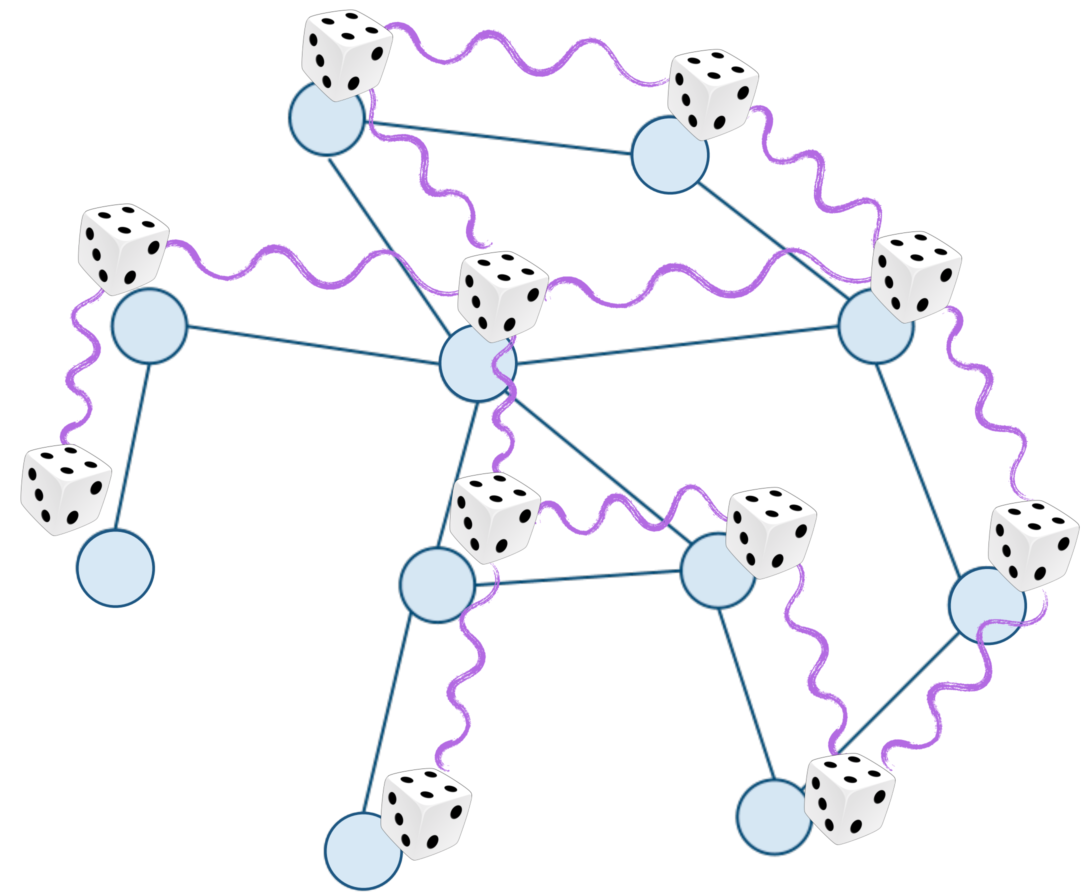
Q: "What locally definable problems are locally computable?"

by local constraints

in O(1) rounds
or in small number of rounds

“What can be *sampled* locally?”

- CSP with local constraints on the network:
  - proper q-coloring;
  - independent set;
- Sample a uniform random feasible solution:
  - distributed algorithms (in the $\mathcal{LOCAL}$ model)



network $G(V,E)$

Q: “What locally definable joint distributions are locally sample-able?”

Markov Random Fields
(MRF)

- Each vertex corresponds to a variable with finite domain $[q]$.
- Each edge $e=(u,v)\in E$ imposes a *weighted* binary constraint:

$$A_e : [q]^2 \to \mathbb{R}_{\geq 0}$$

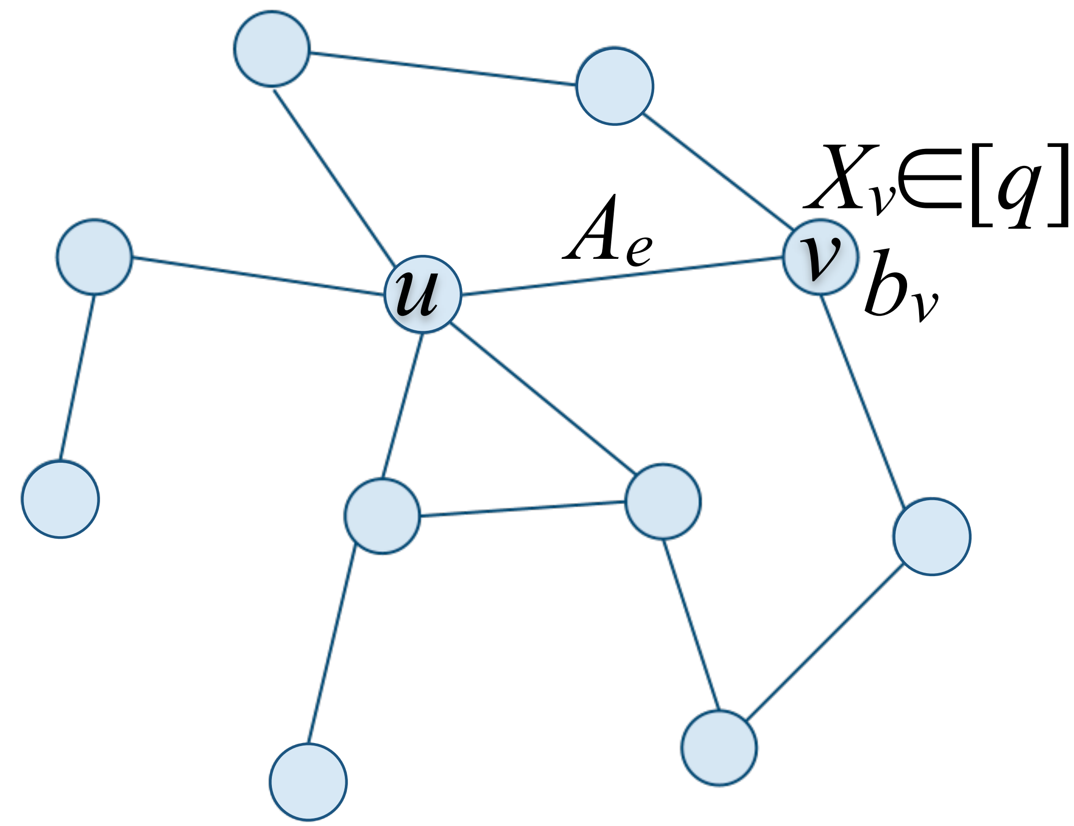
- Each vertex $v\in E$ imposes a *weighted* unary constraint:

$$b_v : [q] \to \mathbb{R}_{\geq 0}$$

- Gibbs distribution μ : $\forall \sigma\in[q]^V$

$$\mu(\sigma) \propto \prod_{e=(u,v)\in E} A_e(\sigma_u, \sigma_v) \prod_{v\in V} b_v(\sigma_v)$$

network $G(V,E)$:



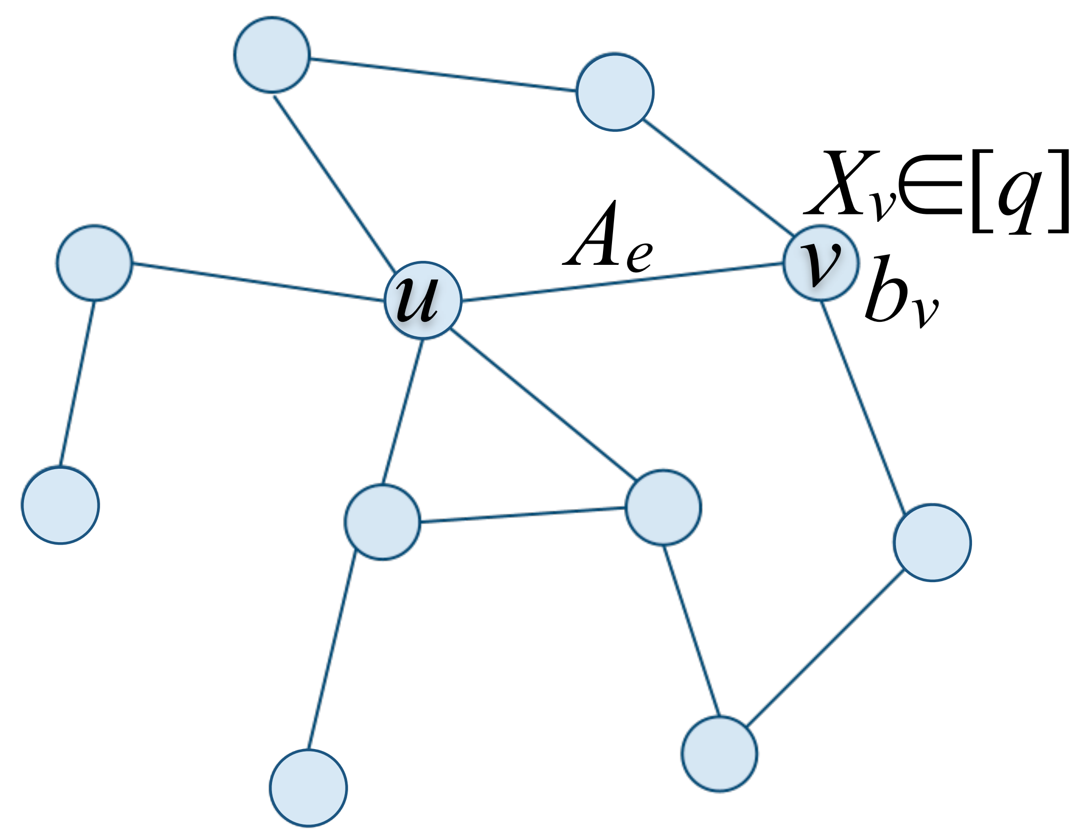
$\vec{X} \in [q]^V$ follows μ

Markov Random Fields
(MRF)

- Gibbs distribution μ : $\forall \sigma \in [q]^V$

$$\mu(\sigma) \propto \prod_{e=(u,v)\in E} A_e(\sigma_u, \sigma_v) \prod_{v \in V} b_v(\sigma_v)$$

network $G(V,E)$:



$\vec{X} \in [q]^V$ follows μ

- proper q-coloring:

$$A_e = \begin{bmatrix} 0 & & \\ & 0 & \mathbf{1} \\ \mathbf{1} & \ddots & \\ & & 0 \end{bmatrix} \qquad b_v = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

- independent set:

$$A_e = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \qquad b_v = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

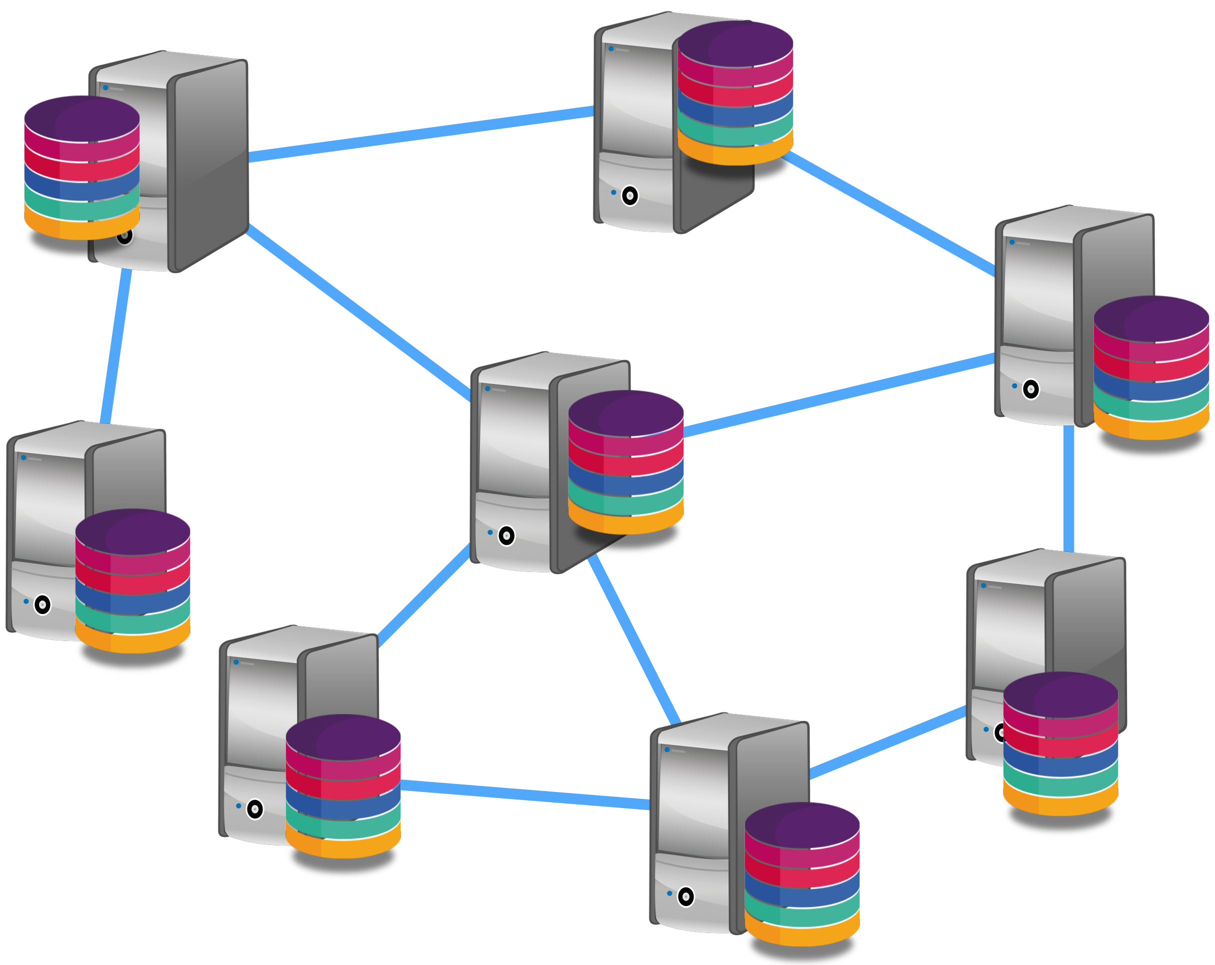
- local conflict colorings : $A_e \in \{0,1\}^{q \times q}, b_v \in \{0,1\}^q$

[Fraigniaud, Heinrich, Kosowski'16]

Hammersley–Clifford theorem (Fundamental Thm of random fields):
MRFs are universal for *conditional independent* positive distributions.

A Motivation:
Distributed Machine Learning

- Data are stored in a distributed system.
- Sampling from a *probabilistic graphical model* (e.g. the *Markov random field*) by distributed algorithms.

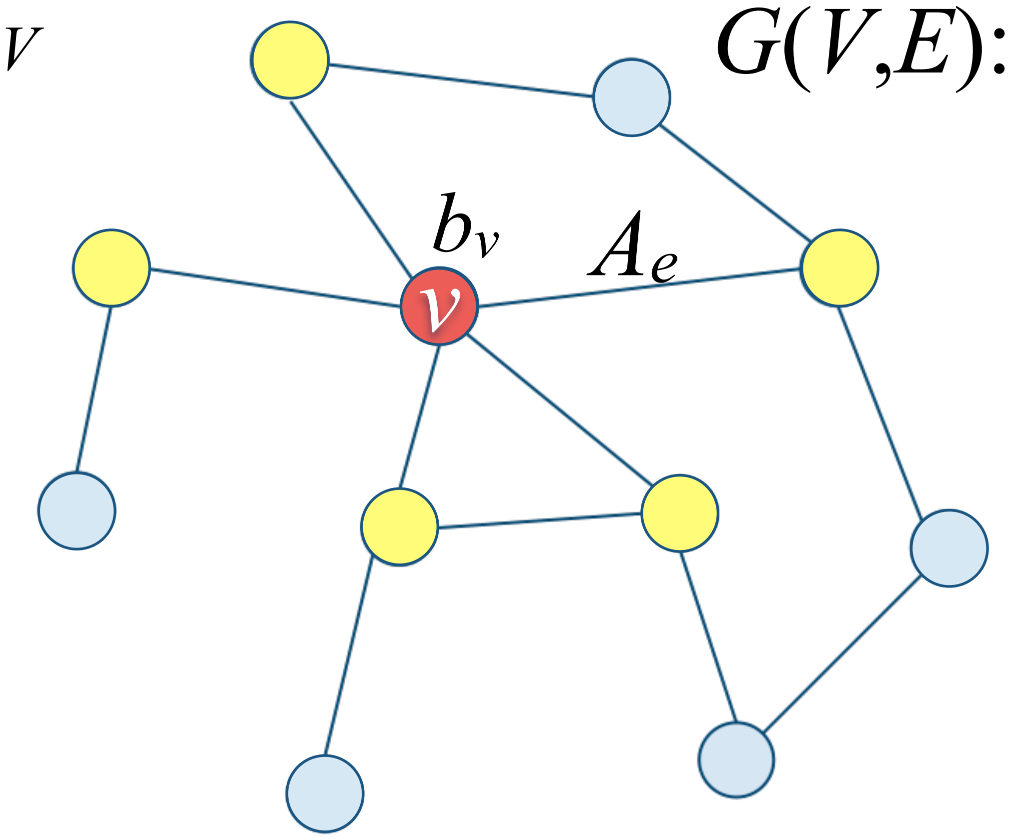


Glauber Dynamics

starting from an arbitrary $X_0 \in [q]^V$

transition for $X_t \to X_{t+1}$:

> pick a uniform random vertex v;
>
> resample $X(v)$ according to the marginal distribution induced by μ at vertex v conditioning on $X_t(N(v))$;



marginal distribution:

$$\Pr[X_v = x \mid X_{N(v)}] = \frac{b_v(x)\prod_{u\in N(v)} A_{(u,v)}(X_u, x)}{\sum_{y\in[q]} b_v(y)\prod_{u\in N(v)} A_{(u,v)}(X_u, y)}$$

MRF: $\forall \sigma \in [q]^V$,

$$\mu(\sigma) \propto \prod_{e=(u,v)\in E} A_e(\sigma_u, \sigma_v) \prod_{v\in V} b_v(\sigma_v)$$

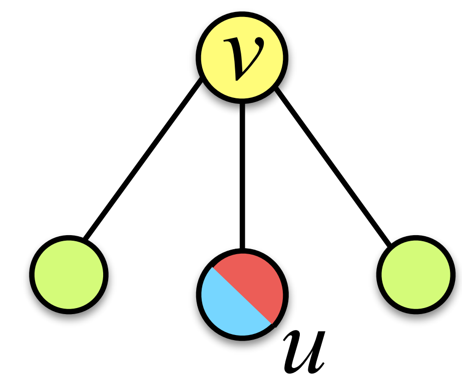
stationary distribution: μ

mixing time: $\tau_{\text{mix}} = \max_{X_0} \min\left\{t \mid d_{\text{TV}}(X_t, \mu) \le \frac{1}{2\mathrm{e}}\right\}$

Mixing of Glauber Dynamics

influence matrix $\{\rho_{v,u}\}_{v,u\in V}$:

$\rho_{v,u}$: max discrepancy (in total variation distance) of marginal distributions at v caused by any pair σ,τ of boundary conditions that differ only at u



Dobrushin's condition:

$$\|\boldsymbol{\rho}\|_\infty = \max_{v\in V}\sum_{u\in V}\rho_{v,u} \le 1-\epsilon$$

contraction of one-step optimal coupling in the worst case w.r.t. Hamming distance

Theorem (Dobrushin '70; Jerrum '95; Salas, Sokal '97):

Dobrushin's condition ⟹ $\tau_{\text{mix}} = O(n\log n)$ for Glauber dynamics

for q-coloring: Dobrushin's condition ⟺ $q\ge(2+\varepsilon)\Delta$

Δ = max-degree

Parallelization

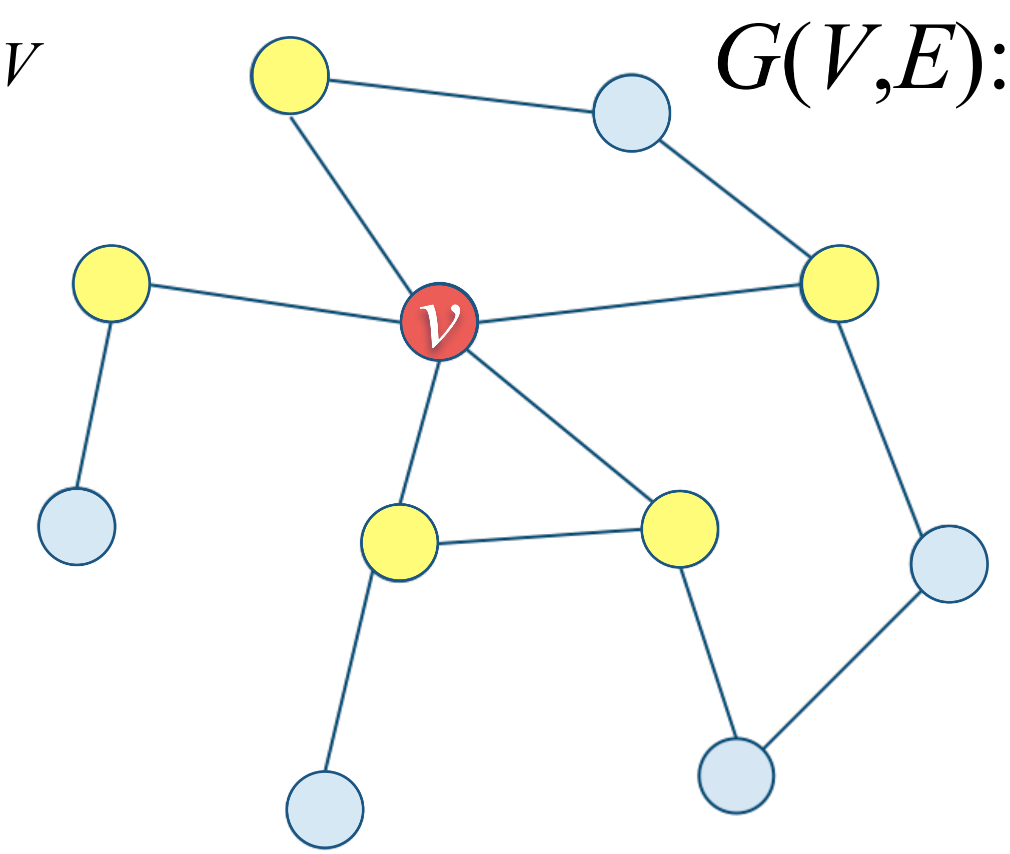
Glauber dynamics:

starting from an arbitrary $X_0 \in [q]^V$

transition for $X_t \to X_{t+1}$:

pick a uniform random vertex v;

resample $X(v)$ according to the marginal distribution induced by μ at vertex v conditioning on $X_t(N(v))$;



Parallelization:

- Chromatic scheduler [folklore] [Gonzalez *et al.*, AISTAT'11]: Vertices in the same color class are updated in parallel.
- "Hogwild!" [Niu, Recht, Ré, Wright, NIPS'11][De Sa, Olukotun, Ré, ICML'16]: All vertices are updated in parallel, ignoring concurrency issues.

Warm-up: *When Luby meets Glauber*

starting from an arbitrary $X_0 \in [q]^V$

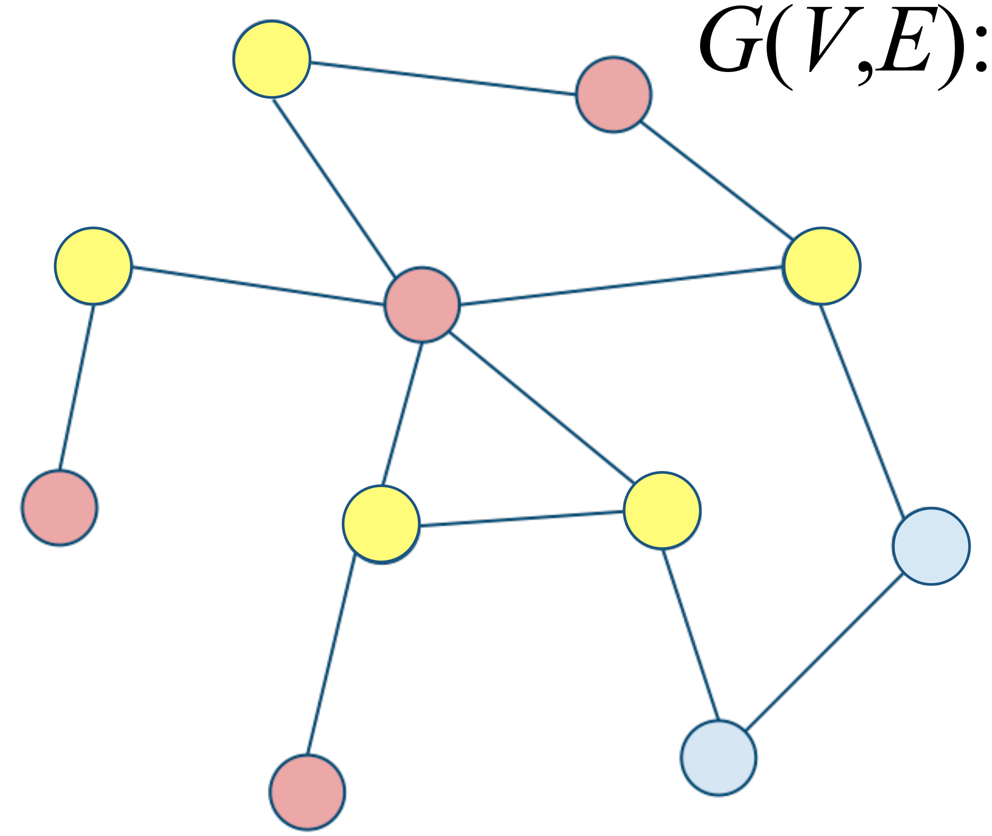
at each step, for each vertex $v \in V$:

Luby step

> independently sample a random number $\beta_v \in [0,1]$;
>
> if β_v is locally maximum among its neighborhood $N(v)$:

Glauber step

> resample $X(v)$ according to the marginal distribution induced by μ at vertex v conditioning on $X_t(N(v))$;



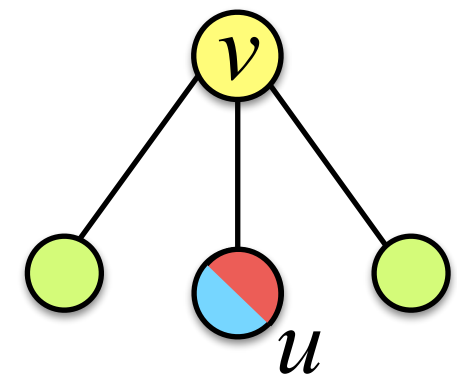
- Luby step: Independently sample a random independent set.
- Glauber step: For independent set vertices, update correctly according to the current marginal distributions.
- Stationary distribution: the Gibbs distribution μ.

Mixing of *LubyGlauber*

influence matrix $\{\rho_{v,u}\}_{v,u\in V}$

Dobrushin's condition:

$$\|\rho\|_\infty = \max_{v\in V}\sum_{u\in V}\rho_{v,u} \le 1-\epsilon$$



Theorem (Dobrushin '70; Jerrum '95; Salas, Sokal '97):

Dobrushin's condition ⟹ $\tau_{\text{mix}} = O(n\log n)$ for Glauber dynamics

Dobrushin's condition ⟹ $\tau_{\text{mix}} = O(\Delta\log n)$ for the *LubyGlauber* chain

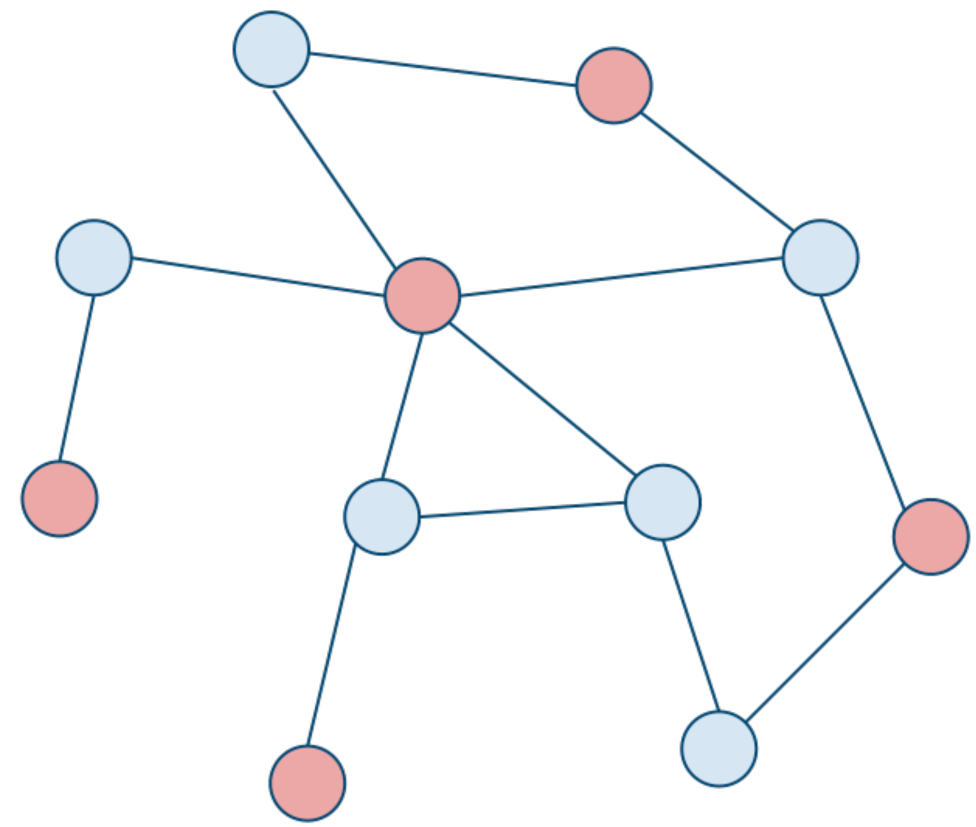
By a similar proof of [Hayes'04] [Dyer-Goldberg-Jerrum'06]

Crossing the Chromatic # Barrier

Glauber $\qquad\qquad$ *LubyGlauber*

$O(n \log n)$ → $O(\Delta \log n)$

parallel speedup
$= \theta(n/\Delta)$



Δ = max-degree

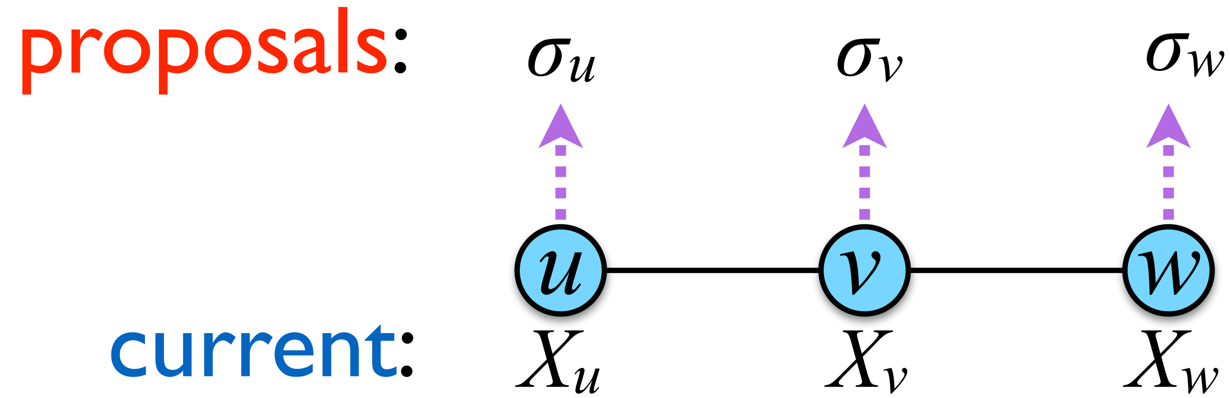
χ = chromatic no.

Do not update adjacent vertices simultaneously.

⇒ It takes $\geq \chi$ steps to update all vertices at least once.

Q: "How to update all variables simultaneously and still converge to the correct distribution?"

The *LocalMetropolis* Chain



starting from an arbitrary $X \in [q]^V$, at each step:

each vertex $v \in V$ independently proposes a random $\sigma_v \in [q]$ with probability $b_v(\sigma_v)/\sum_{i\in[q]} b_v(i)$;

Markov Random Fields

(MRF)

- Each vertex corresponds to a variable with finite domain $[q]$.
- Each edge $e=(u,v)\in E$ imposes a *weighted* binary constraint:

$$A_e : [q]^2 \to \mathbb{R}_{\geq 0}$$

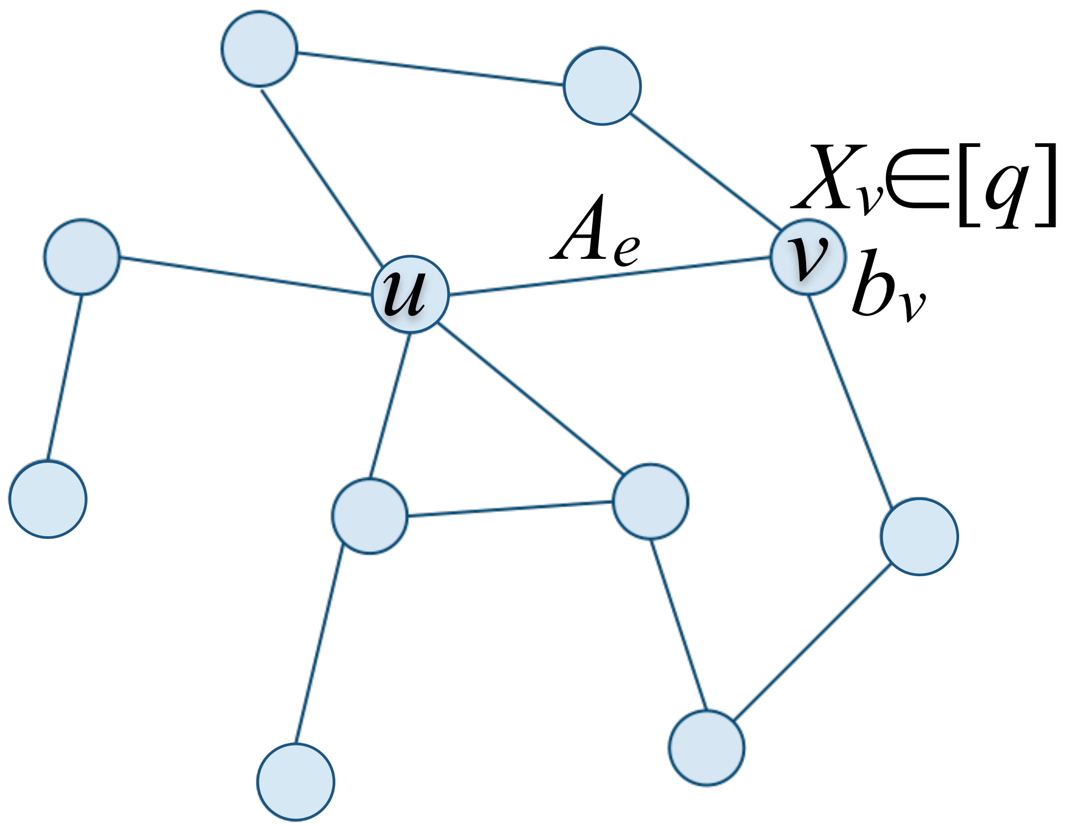
- Each vertex $v\in E$ imposes a *weighted* unary constraint:

$$b_v : [q] \to \mathbb{R}_{\geq 0}$$

- Gibbs distribution μ : $\forall \sigma\in[q]^V$

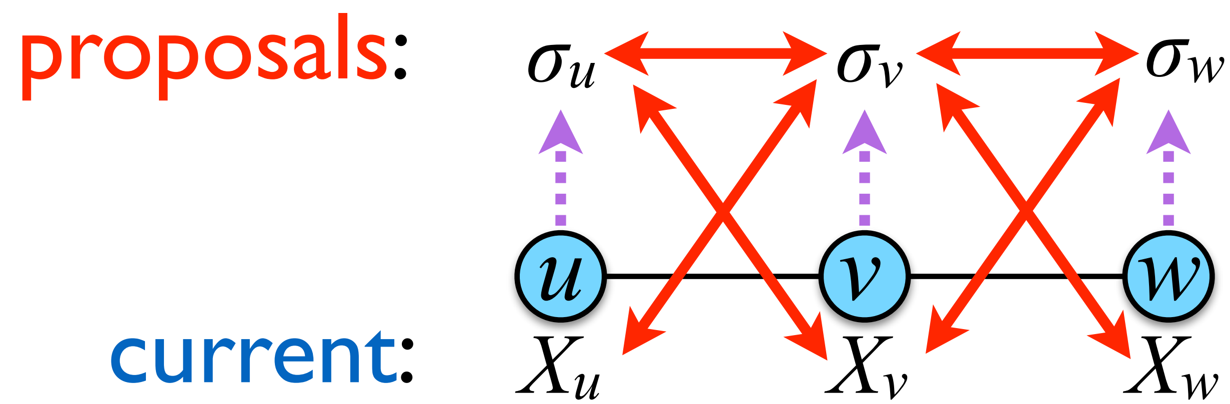
$$\mu(\sigma) \propto \prod_{e=(u,v)\in E} A_e(\sigma_u, \sigma_v) \prod_{v\in V} b_v(\sigma_v)$$

network $G(V,E)$:



$\vec{X} \in [q]^V$ follows μ

The *LocalMetropolis* Chain



starting from an arbitrary $X \in [q]^V$, at each step:

each vertex $v \in V$ independently proposes a random $\sigma_v \in [q]$ with probability $b_v(\sigma_v) / \sum_{i \in [q]} b_v(i)$;

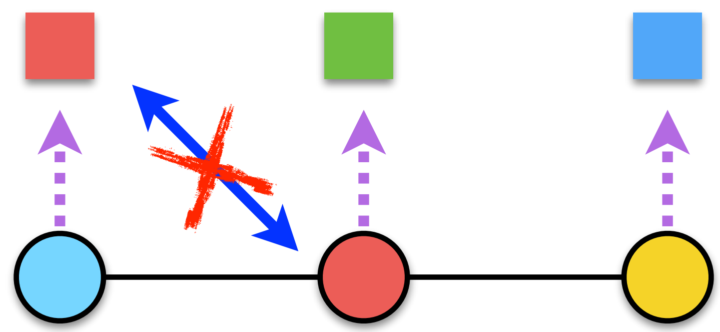
each edge $e=(u,v)$ passes its check independently with prob. $A_e(X_u, \sigma_v) A_e(\sigma_u, X_v) A_e(\sigma_u, \sigma_v) / \max_{i,j \in [q]} (A_e(i,j))^3$;

each vertex $v \in V$ accepts its proposal and update X_v to σ_v if *all incident edges pass their checks*;

a collective coin flipping made between u and v

- The *LocalMetropolis* chain is *time-reversible* and its stationary distribution is the MRF Gibbs distribution μ.

LocalMetropolis for q-Coloring



starting from an arbitrary $X \in [q]^V$, at each step, each vertex $v \in V$:

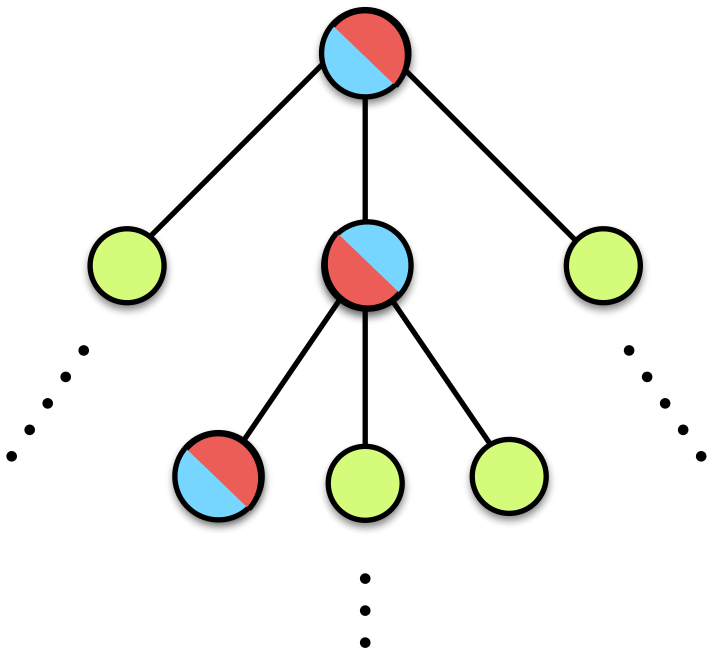
proposes a color $\sigma_v \in [q]$ uniformly and independently at random;

accepts the proposal and update X_v to σ_v if for all v's neighbors u:

$$X_u \neq \sigma_v \wedge \sigma_u \neq X_v \wedge \sigma_u \neq \sigma_v;$$

$$q \geq (2+\sqrt{2}+\epsilon)\Delta \Rightarrow \tau_{\text{mix}} = O(\log n)$$

for *LocalMetropolis* on q-coloring



each v:

proposes a uniform random color $\sigma_v \in [q]$;
update X_v to σ_v if for all v's neighbors u:
$X_u \neq \sigma_v \wedge \sigma_u \neq X_v \wedge \sigma_u \neq \sigma_v$;

X_{root} = red , Y_{root} = blue

$\forall$ non-root v, $X_v = Y_v \notin \{\text{red}, \text{blue}\}$

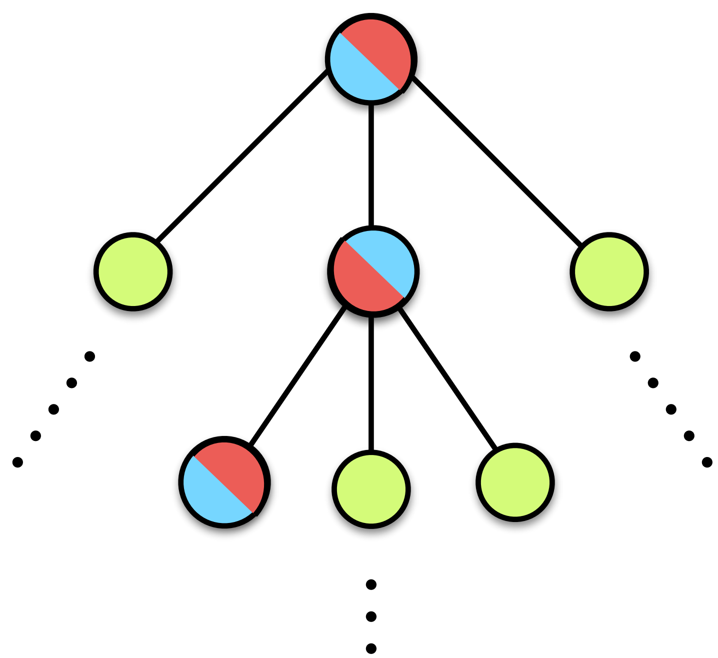
coupling: coupling the proposals (σ^X, σ^Y) so that $(X,Y) \xrightarrow{(\sigma^X,\sigma^Y)} (X',Y')$

vertex v proposes consistently: $\sigma_v^X = \sigma_v^Y$

vertex v proposes bijectively: $\sigma_v^X = \begin{cases} \texttt{red} & \text{if } \sigma_v^Y = \texttt{blue} \\ \texttt{blue} & \text{if } \sigma_v^Y = \texttt{red} \\ \sigma_v^Y & \text{otherwise} \end{cases}$

1. the root proposes consistently;
2. each child of the root proposes bijectively;
3. each vertex of depth ≥2 proposes bijectively if its parent proposed different colors in the two chains, and proposes consistently if otherwise;

Δ-regular tree



each v:

proposes a uniform random color $\sigma_v \in [q]$;
update X_v to σ_v if for all v's neighbors u:
$X_u \neq \sigma_v \wedge \sigma_u \neq X_v \wedge \sigma_u \neq \sigma_v$;

X_{root} = red , Y_{root} = blue

$\forall$ non-root v, $X_v = Y_v \notin \{\text{red}, \text{blue}\}$

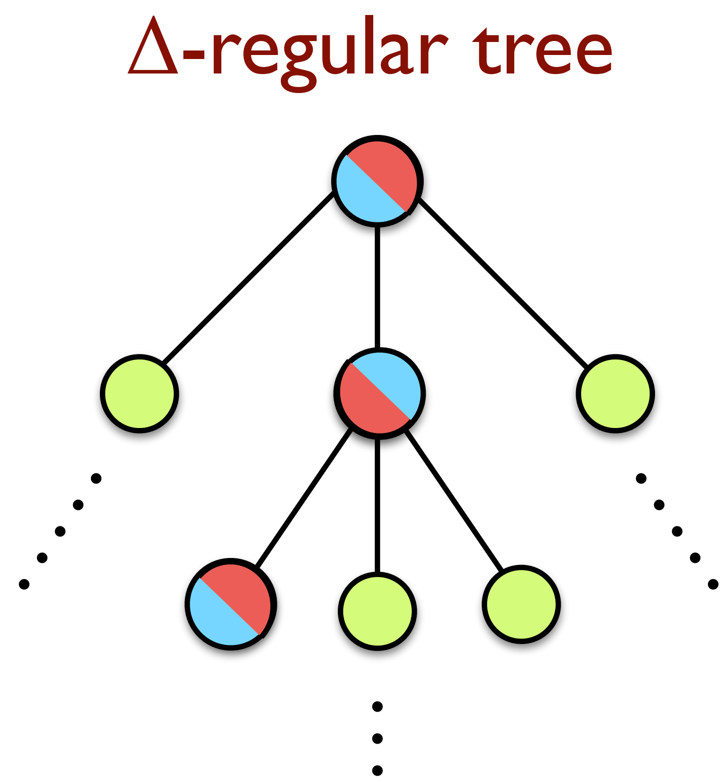
coupling: coupling the proposals (σ^X, σ^Y) so that $(X,Y) \xrightarrow{(\sigma^X,\sigma^Y)} (X',Y')$

root: $\Pr[X'_{\text{root}} \neq Y'_{\text{root}}] \leq 1 - \left(1-\frac{\Delta}{q}\right)\left(1-\frac{2}{q}\right)^{\Delta}$

non-root u at level l: $\Pr[X'_u \neq Y'_u] \leq \frac{1}{q}\left(1-\frac{2}{q}\right)^{\Delta-1}\left(\frac{2}{q}\right)^{\ell-1}$

$$\Pr[X'_{\text{root}} \neq Y'_{\text{root}}] + \sum_{\text{non-root } u} \Pr[X'_u \neq Y'_u] \leq 1 - \left(1-\frac{\Delta}{q}\right)\left(1-\frac{2}{q}\right)^{\Delta} + \frac{\Delta}{q-2\Delta}\left(1-\frac{2}{q}\right)^{\Delta-1}$$

(assume $q \geq \alpha\Delta$) $\qquad \leq 1 - \mathrm{e}^{-2/\alpha}\left(1-\frac{1}{\alpha}-\frac{1}{\alpha-2}\right)$



each v:

proposes a uniform random color $\sigma_v \in [q]$;

update X_v to σ_v if for all v's neighbors u:

$$X_u \neq \sigma_v \wedge \sigma_u \neq X_v \wedge \sigma_u \neq \sigma_v;$$

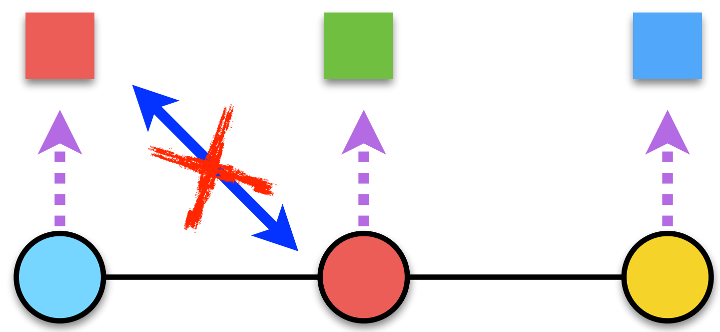
X_{root} = red , Y_{root} = blue

$\forall$ non-root v, $X_v = Y_v \notin \{\text{red}, \text{blue}\}$

for general graph:

1. deal with irregularity by the path coupling metric;
2. deal with cycles by the *self-avoiding walks*;
3. deal with red/blue non-root vertices by a monotone argument;

LocalMetropolis for q-Coloring



starting from an arbitrary $X \in [q]^V$, at each step, each vertex $v \in V$:

proposes a color $\sigma_v \in [q]$ uniformly and independently at random;

accepts the proposal and update X_v to σ_v if for all v's neighbors u:

$$X_u \neq \sigma_v \wedge \sigma_u \neq X_v \wedge \sigma_u \neq \sigma_v ;$$

$$q \geq (2 + \sqrt{2} + \epsilon)\Delta \quad \Rightarrow \quad \tau_{\mathrm{mix}} = O(\log n)$$

for *LocalMetropolis* on q-coloring

- The mixing time holds even for *unbounded* Δ and q.
- $q \geq (1+\varepsilon)\Delta$: each vertex is updated at $\Omega(1)$ rate in *LocalMetropolis*

Lower Bounds

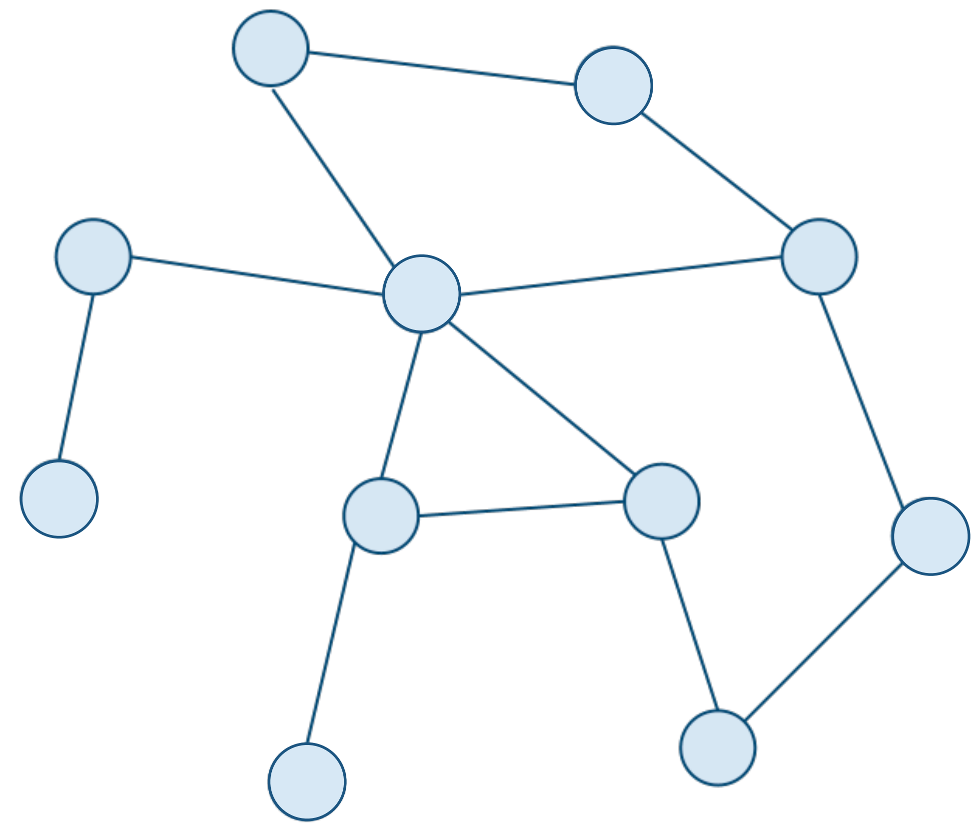
Q: “How local can a distributed sampling algorithm be?”

Q: “What *cannot* be sampled locally?”

The *LOCAL* Model

the *LOCAL* model:

- In t rounds: each node can collect information up to distance t.



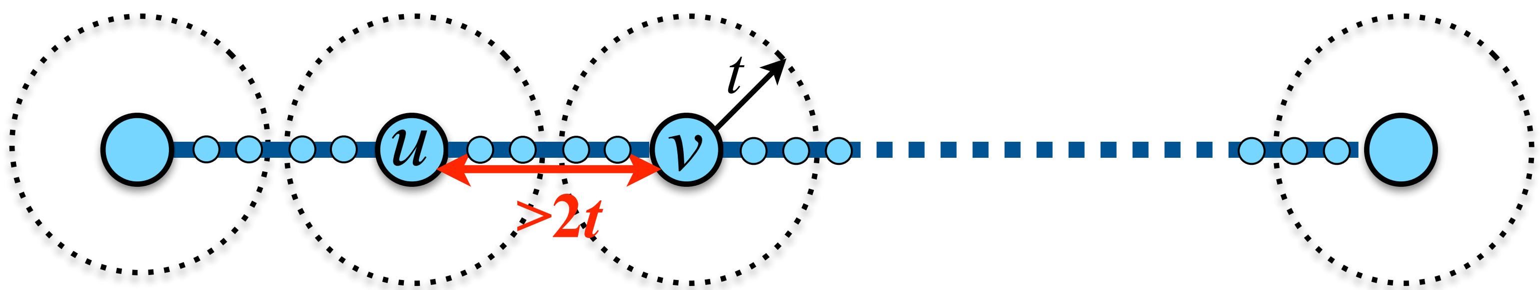
Outputs returned by vertices at distance >$2t$ from each other are mutually independent.

$\Omega(\log n)$ Lower Bound for Sampling

For any non-degenerate MRF, any distributed algorithm that samples from its distribution μ within bounded total variation distance requires $\Omega(\log n)$ rounds of communications.

outputs of t-round algorithm: mutually independent $\widetilde{X}_v$'s

path:



Gibbs distribution μ: exponential correlation between X_v's

$$\sigma_u \neq \tau_u : \quad \|\mu_v^{\sigma_u} - \mu_v^{\tau_u}\|_{\mathrm{TV}} \geq \exp(-O(t)) > n^{-1/4}$$

for a $t = O(\log n)$

$d_{\mathrm{TV}}(\boldsymbol{X}, \widetilde{\boldsymbol{X}}) > \frac{1}{2e}$ for *any* product distribution $\widetilde{\boldsymbol{X}}$

$\Omega(\log n)$ Lower Bound for Sampling

For any non-degenerate MRF, any distributed algorithm that samples from its distribution μ within bounded total variation distance requires $\Omega(\log n)$ rounds of communications.

- The $\Omega(\log n)$ lower bound holds for all MRFs with exponential correlation:
  - non-trivial MRFs with constant domain size.
- $O(\log n)$ is the new criteria of "*being local*" for distributed sampling algorithms.

An $\Omega(diam)$ Lower Bound

For any $\Delta \geq 6$, any distributed algorithm that samples uniform independent set within bounded total variation distance in graphs with max-degree Δ requires $\Omega(diam)$ rounds of communications.

Sampling almost uniform independent set in graphs with max-degree Δ by by poly-time Turing machines:

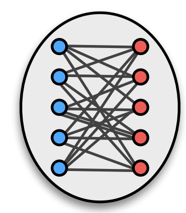
- [Weitz'06] If $\Delta \leq 5$, there are poly-time algorithms.
- [Sly'10] If $\Delta \geq 6$, there is no poly-time algorithm unless **NP**=**RP**.

The $\Omega(diam)$ lower bound holds for sampling from the hardcore model with *fugacity* $\lambda > \lambda_c(\Delta) = \frac{(\Delta-1)^{\Delta-1}}{(\Delta-2)^{\Delta}}$

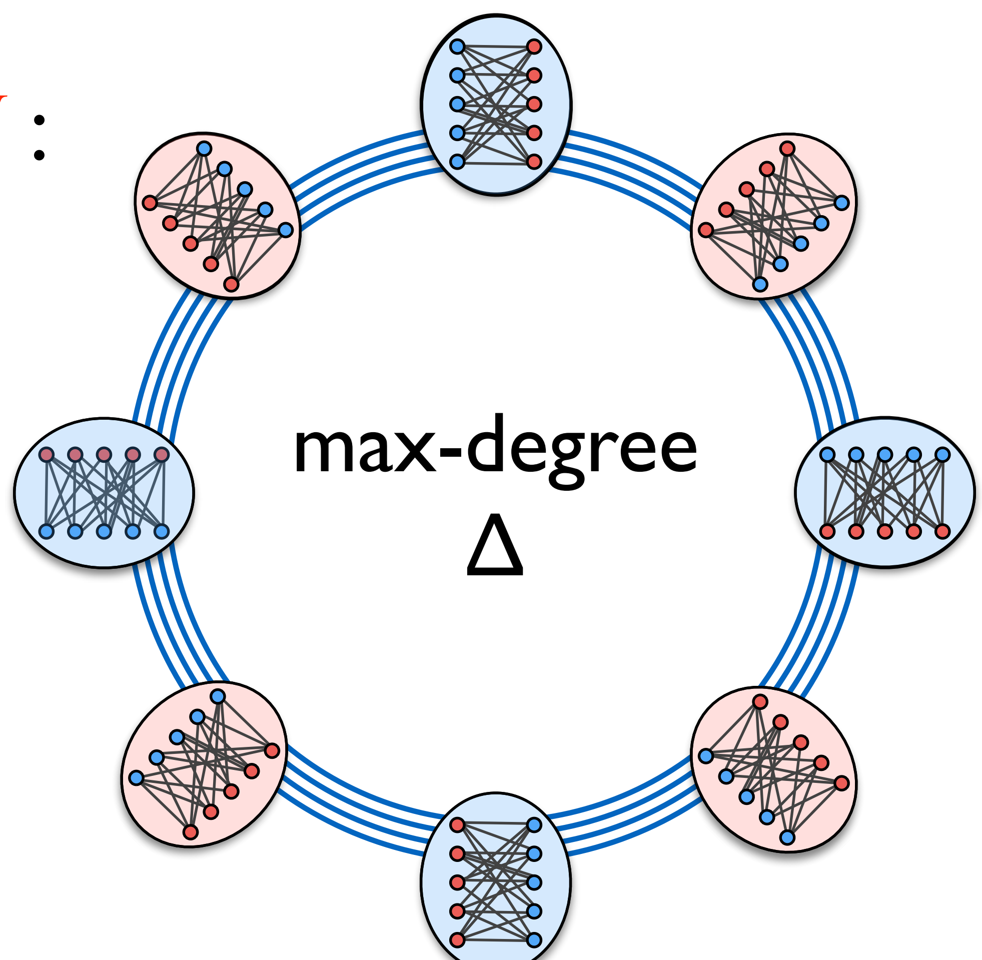
An $\Omega(diam)$ Lower Bound

For any $\Delta \geq 6$, any distributed algorithm that samples uniform independent set within bounded total variation distance in graphs with max-degree Δ requires $\Omega(diam)$ rounds of communications.

G: even cycle

H: random Δ-regular bipartite gadget  of [Sly'10]

G^H :



if $\Delta \geq 6$:

sample nearly uniform independent set in G^H

⇓

sample nearly uniform max-cut in even cycle G

(long-range correlation!)

An $\Omega(diam)$ Lower Bound

For any $\Delta \geq 6$, any distributed algorithm that samples uniform independent set within bounded total variation distance in graphs with max-degree Δ requires $\Omega(diam)$ rounds of communications.

A **strong separation** of *sampling* from other local computation tasks:

- Independent set is trivial to construct locally (because $\varnothing$ is an independent set).
- The $\Omega(diam)$ lower bound for sampling holds even when every vertex knows the entire graph:
  - The lower bound holds not because of the locality of input information, but because of the locality of randomness.

Summary

- Sampling from locally-defined joint distribution via distributed algorithms:
  - *LubyGlauber*: $O(\Delta \log n)$ rounds under Dobrushin condition;
  - *LocalMetropolis*: may achieve $O(\log n)$ rounds;
  - $\Omega(\log n)$ lower bound for sampling from almost all nontrivial joint distributions;
  - $\Omega(diam)$ lower bound for sampling from joint distributions exhibiting (non-uniqueness) phase transition property.
- **Open problems:**
  - better analysis of *LocalMetropolis*;
  - sampling: matchings, ferromagnetic Ising;
  - complexity hierarchy for distributed sampling?

Thank you!

Any questions?