Tony Johansson

Uppsala University

Uppsala, Sweden

The cover time of a biased random walk on a random regular graph of odd degree

RANDOM 2018

Princeton

August 20, 2018

Plan

- Definitions:
  Random walks and random regular graphs
- Results
- Idea behind proof
- A trick
- Some details

Cover time of a random walk

- Fix a graph $G = (V, E)$
- Pick some starting vertex v_0
- At time t move from v_{t-1} to v_t,
  chosen uniformly at random from neighbours of v_{t-1}
- Define random variable

$$C(G; v_0) = \min\{t : \{v_0, v_1, \ldots, v_t\} = V\}$$

- The vertex cover time of G is

$$C_V(G) = \max_{v_0 \in V} \mathbb{E}\left(C(G; v_0)\right)$$

- Edge cover time $C_E(G)$ is expected time to cover all edges

Cover time of a random walk

Feige (95): for any simple connected G on n vertices,

$$(1 - o(1))n \log n \leq C_V(G) \leq \frac{4n^3}{27}.$$

Applications:

- Graph exploration
- Message dissemination
- Web crawl
- Network search with only local information

Other random walks

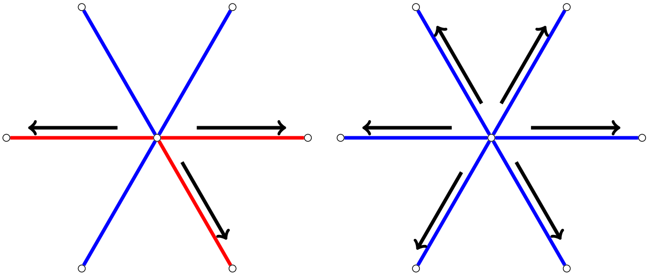
- Non-backtracking: avoid going from u to v to u
- Biased: choose vertices with probability based on degrees
- Biased/greedy: Avoid reusing edges

Slight name confusion!

- Today: biased random walk
- Introduced by Orenshtein and Shinkar (2014) as greedy random walk
- Introduced to speed up walk

Biased random walk

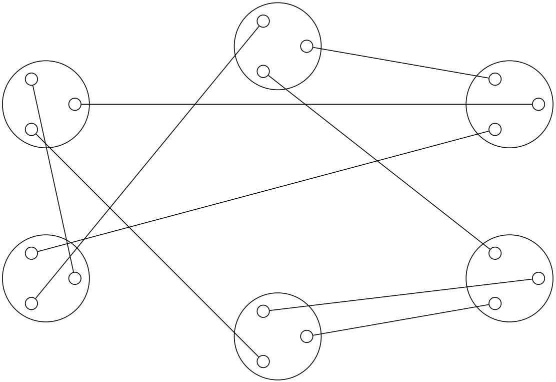
- Initially colour all edges red
- When edge is traversed, recolour it blue
- At time t, move from v_{t-1} along red edge chosen uniformly at random
  Only use blue edge if no red edges available



Random regular graph

Use configuration model

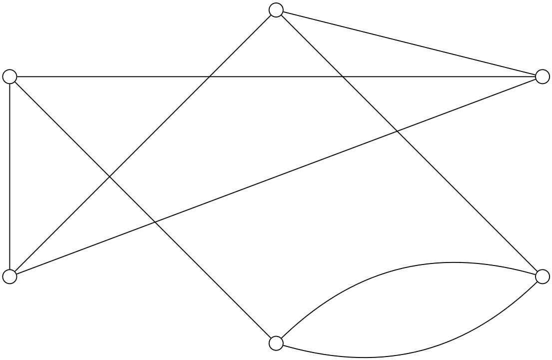
- Each vertex v associated with set $\mathcal{P}(v)$ of d *points*
- Pairing μ of $\bigcup_{v \in V} \mathcal{P}(v)$ chosen u.a.r.
- $u \in v$ if and only if $\mu(x) \in \mathcal{P}(v)$ for some $x \in \mathcal{P}(u)$



Random regular graph

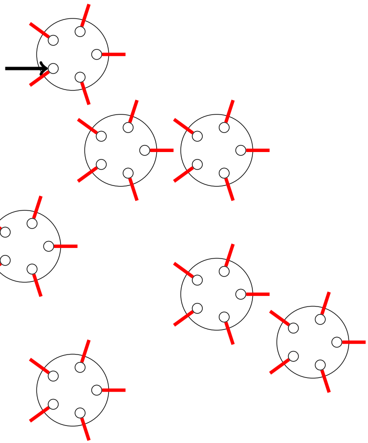
Use configuration model

- Each vertex v associated with set $\mathcal{P}(v)$ of d *points*
- Pairing μ of $\bigcup_{v \in V} \mathcal{P}(v)$ chosen u.a.r.
- $u \in v$ if and only if $\mu(x) \in \mathcal{P}(v)$ for some $x \in \mathcal{P}(u)$



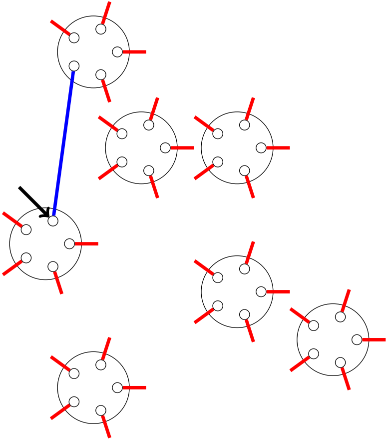
Walk on configuration model

Generate configuration as we go
Unused points chosen when available



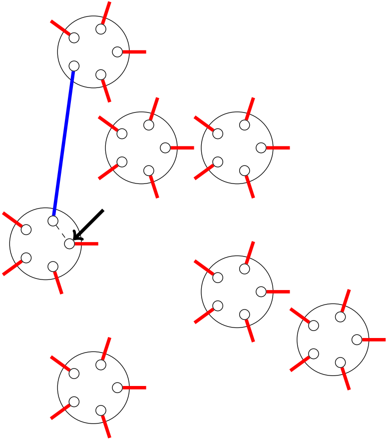
Walk on configuration model

Generate configuration as we go
Unused points chosen when available



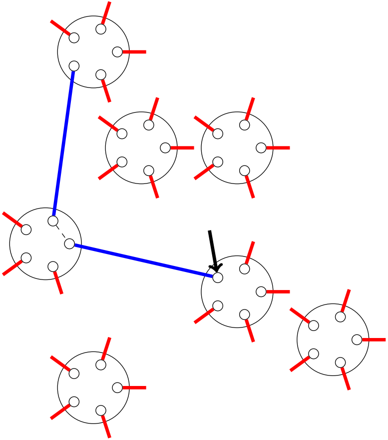
Walk on configuration model

Generate configuration as we go
Unused points chosen when available



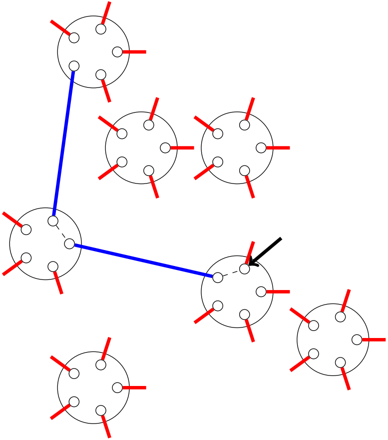
Walk on configuration model

Generate configuration as we go
Unused points chosen when available



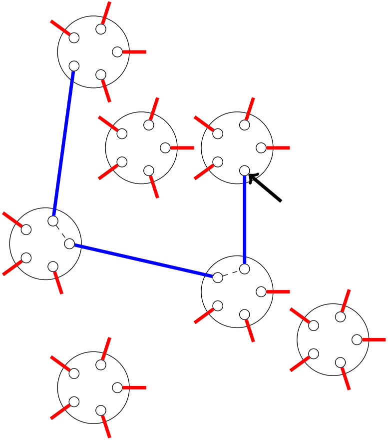
Walk on configuration model

Generate configuration as we go
Unused points chosen when available



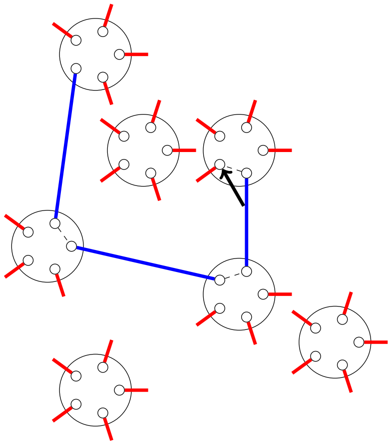
Walk on configuration model

Generate configuration as we go
Unused points chosen when available



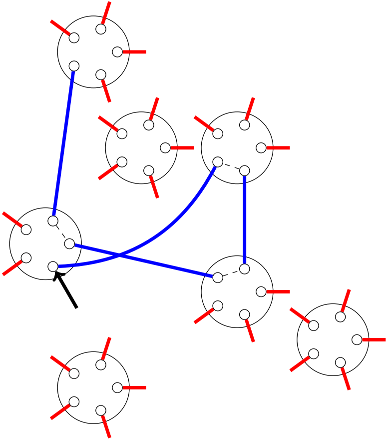
Walk on configuration model

Generate configuration as we go
Unused points chosen when available



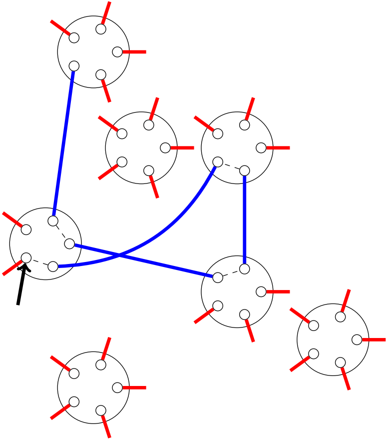
Walk on configuration model

Generate configuration as we go
Unused points chosen when available



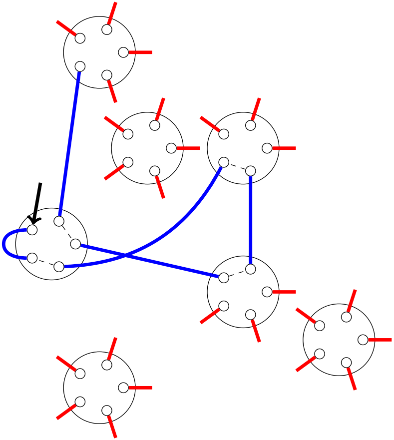
Walk on configuration model

Generate configuration as we go
Unused points chosen when available



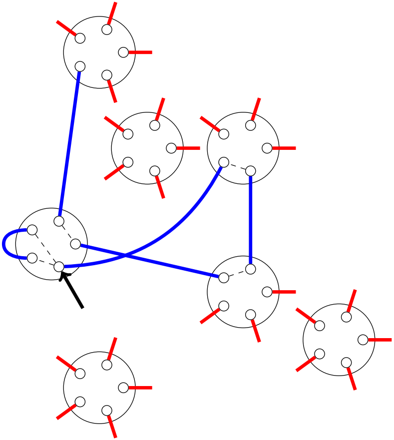
Walk on configuration model

Generate configuration as we go
Unused points chosen when available



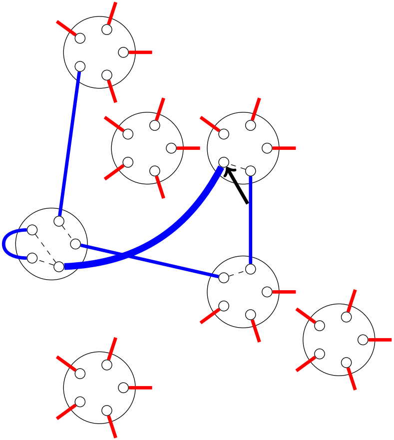
Walk on configuration model

Generate configuration as we go
Unused points chosen when available



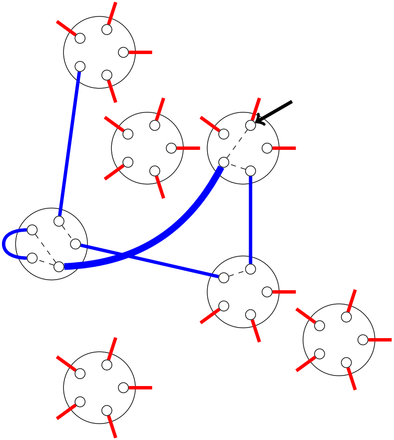
Walk on configuration model

Generate configuration as we go
Unused points chosen when available



Walk on configuration model

Generate configuration as we go
Unused points chosen when available



Plan

- Definitions:
  Random walks and random regular graphs
- Results
- Idea behind proof
- A trick
- Some details

Walks on random regular graphs

Let G_d denote the random d-regular graph, $d \geq 3$

Theorem (Cooper, Frieze 2005)

$$C_V^{simple}(G_d) \sim \frac{d-1}{d-2} n \log n.$$

Theorem (Cooper, Frieze 2016)

$$C_V^{non-backtracking}(G_d) \sim n \log n.$$

Theorem (Berenbrink, Cooper, Friedetzky 2015)

For $d \geq 4$ even,

$$C_V^{bias}(G_d) \sim \frac{dn}{2}.$$

Results

Theorem (Cooper, Frieze, J., AofA 2018)

With high probability, G_3 is such that

$$C_V^{bias}(G_3) \sim n \log n.$$

Results

Theorem (Cooper, Frieze, J., AofA 2018)

With high probability, G_3 is such that

$$C_V^{bias}(G_3) \sim n \log n.$$

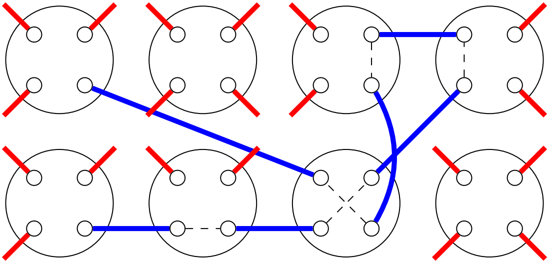
Theorem (J., RANDOM 2018)

For fixed odd $d \geq 5$, w.h.p. G_d is such that

$$C_V^{bias}(G_d) \sim \frac{1}{d-2} n \log n.$$

- Slower than even d by $\log n$, but quicker than other variations

Biased walk on 4-regular graph



The walk will almost always go from red edge to red edge,
as almost all vertices hold an even number of them

Note G_d contains $dn/2$ edges.

Theorem

Define $C(t)$ as the number of steps used to find t distinct edges. W.h.p., G_d is such that for all $dn/2(1-o(1)) \leq t \leq dn/2$,

$$C(t) \sim \frac{d}{2(d-2)} n \log\left(\frac{dn}{dn-2t+1}\right)$$

Corollary

W.h.p., G_d is such that

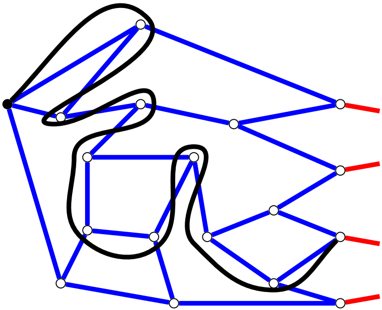
$$C_V^{bias}(G_d) \sim \frac{1}{d-2} n \log n,$$
$$C_E^{bias}(G_d) \sim \frac{d}{2(d-2)} n \log n.$$

Plan

- Definitions:
  Random walks and random regular graphs
- Results
- Idea behind proof
- A trick
- Some details

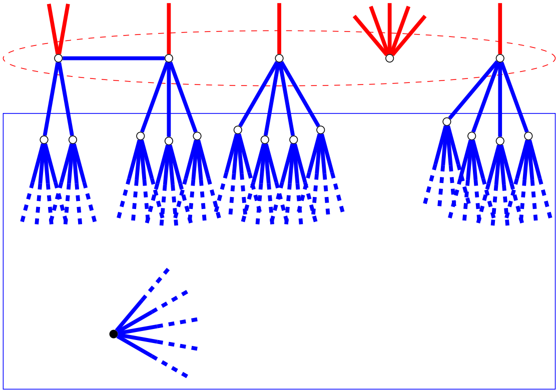
Proof idea

- Near end of process, almost all edges blue (think $\sqrt{n}$ red edges)
- If walker surrounded by blue edges, locally behaves like simple random walk
- Use hitting time theory for simple random walks to estimate time to find red edge



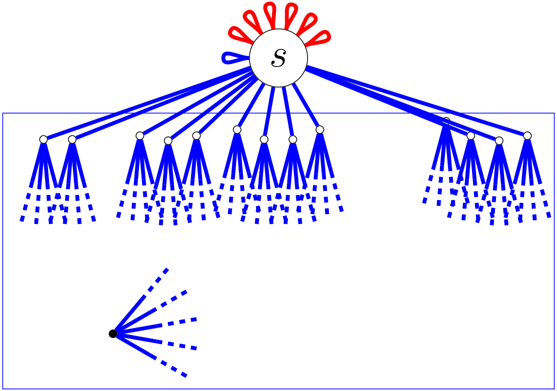
Proof overview

Define the boundary as vertices incident to at least one red edge



Proof overview

Contract boundary into one vertex s
– Does not change hitting time



Proof

Suppose

- G is d-regular with positive spectral gap
- S is contracted to a vertex s, so that
  - s has $|S|/2$ self-loops,
  - s is on no cycle of length $\leq \log \log n$

Then a simple random walk from a random starting point hits s in expected time $\approx \frac{d}{d-2} \frac{n}{|S|}$

Proof

Suppose

- G is d-regular with positive spectral gap
- S is contracted to a vertex s, so that
  - s has $|S|/2$ self-loops,
  - s is on no cycle of length $\leq \log\log n$

Then a simple random walk from a random starting point hits s in expected time $\approx \frac{d}{d-2}\frac{n}{|S|}$

Enough if almost all boundary vertices are:

- Incident to exactly one red edge,
- On no cycle of length $\leq \log\log n$, (immediate from G_d structure)
- At distance $\geq \log\log n$ from other boundary vertices

Plan

- Definitions:
  Random walks and random regular graphs
- Results
- Idea behind proof
- A trick
- Some details

Rerandomizing the walk

- Halt the walk after k steps, call this

$$W_k = (x_0, x_1, \ldots, x_k)$$

- Let $e_i = x_{i-1}x_i$ be the ith edge traversed
- Say that (e_i, e_{i+1}) is a link if
  e_i, e_{i+1} appear exactly once in W_k

Rerandomizing the walk

- Halt the walk after k steps, call this

$$W_k = (x_0, x_1, \ldots, x_k)$$

- Let $e_i = x_{i-1}x_i$ be the ith edge traversed
- Say that (e_i, e_{i+1}) is a link if
  e_i, e_{i+1} appear exactly once in W_k

Observation I: if $(e_i, e_{i+1}), (e_j, e_{j+1})$ are links then

$$W'_k = (x_0, \ldots, x_{i-1}, x_j, x_{i+1}, \ldots, x_{j-1}, x_i, x_{j+1}, \ldots, x_k)$$

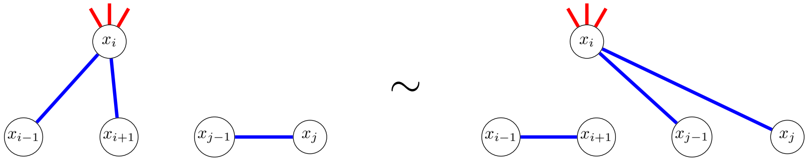
satisfies $\Pr\{W'_k\} = \Pr\{W_k\}$

Rerandomizing the walk

Observation II: if (e_i, e_{i+1}) is a link and e_j appears exactly once in W_k, then

$$W_k' = (x_0, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{j-1}, x_i, x_j, x_{j+1}, \ldots, x_k)$$

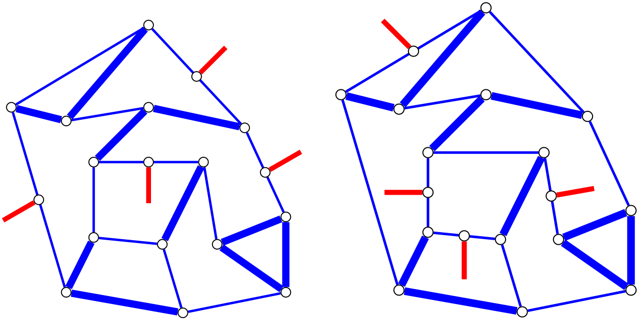
satisfies $\Pr\{W_k'\} = \Pr\{W_k\}$



Observations I and II easy to prove by calculating probability exactly

Rerandomizing the walk

The two pictures below are equally likely...



...if the blue edges touching red have been used exactly once

Rerandomizing the walk

Let Z be the boundary vertices v such that

- v is incident to exactly one red edge,
- The other $d - 1$ edges form $(d-1)/2$ links

We can decouple each $v \in Z$ and choose $(d-1)/2$ new links

Rerandomizing the walk

Let Z be the boundary vertices v such that

- v is incident to exactly one red edge,
- The other $d-1$ edges form $(d-1)/2$ links

We can decouple each $v \in Z$ and choose $(d-1)/2$ new links

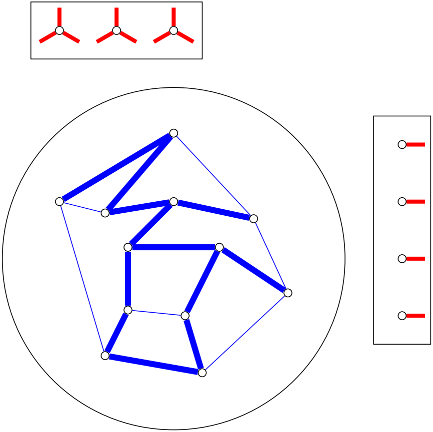
Vertices of Z will be far apart if

$$\#\text{once-visited edges} \gg |Z| \tag{1}$$

If we can show (1) and that almost all boundary vertices are in Z, we are done

Sprinkling

Sprinkle boundary vertices “into” once-visited edges

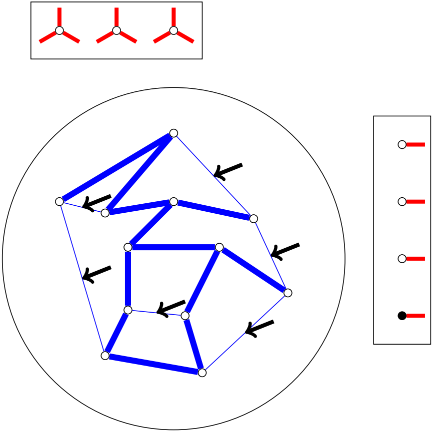


Thick visited twice

Thin once

Sprinkling

Sprinkle boundary vertices “into” once-visited edges

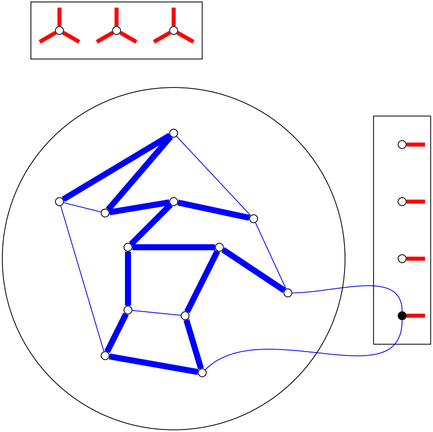


Thick visited twice

Thin once

Sprinkling

Sprinkle boundary vertices "into" once-visited edges

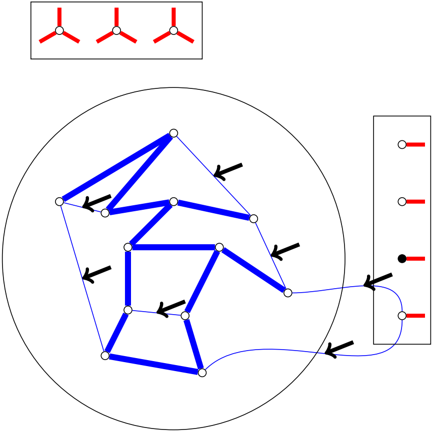


Thick visited twice

Thin once

Sprinkling

Sprinkle boundary vertices “into” once-visited edges

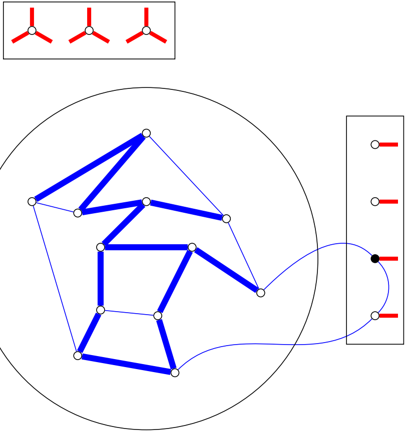


Thick visited twice

Thin once

Sprinkling

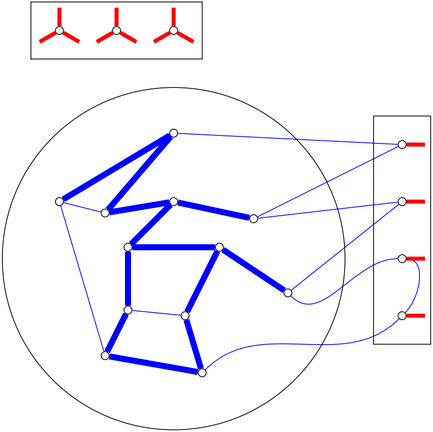
Sprinkle boundary vertices “into” once-visited edges



Thick visited twice
Thin once

Sprinkling

Sprinkle boundary vertices “into” once-visited edges



Thick visited twice

Thin once

Plan

- Definitions:
  Random walks and random regular graphs
- Results
- Idea behind proof
- A trick
- Some details

A size lemma

Lemma

If $t = (1-\delta)\frac{dn}{2}$ *where* $\delta = o(1)$ *then*

$$\textit{number of unvisited vertices} \sim n\delta^{d/2}$$

$$|Z| \sim \frac{d}{2}n\delta$$

$$\textit{number of once-visited edges} \gg n\delta$$

Conclusions:

- The $dn\delta/2$ red edges almost form a matching
- Z sprinkled into much larger
  set of once-visited edges, hence spread far apart

Set sizes: proof idea

- Hide boundary vertices
- Walk on blue edges
- Once-visited edge hides Z vertex w.p.

$$\frac{\#\text{boundary vertices}}{\#\text{once-visited edges}}$$

- Once vertex is found, walk to undiscovered vertex w.p.

$$\frac{d \times \#\text{undiscovered vertices}}{\#\text{red edges}}$$

  otherwise to another boundary vertex
- This sets up recursions for the quantities

Concluding remarks

- Unsurprisingly, the biased random walk is faster than the simple random walk in all known cases
- Open: Is there a G with

$$C_V^{bias}(G) > C_V^{simple}(G)?$$

- Open: Is the biased random walk recursive on $\mathbb{Z}^2$?

Thank you!