Uniformly random spanning trees in almost-linear time

Aaron Schild
aschild@berkeley.edu

University of California, Berkeley

June 25, 2018

The weighted uniformly random spanning tree problem

Given an undirected graph G with weights (conductances) $\{c_e\}_{e \in E(G)}$ on its edges, sample a spanning tree T of G with probability proportional to $\prod_{e \in E(T)} c_e$.

Motivation

Given an undirected graph G with weights (conductances) $\{c_e\}_{e\in E(G)}$ on its edges, sample a spanning tree T of G with probability proportional to $\prod_{e\in E(T)} c_e$.

- Direct applications

Motivation

Given an undirected graph G with weights (conductances) $\{c_e\}_{e \in E(G)}$ on its edges, sample a spanning tree T of G with probability proportional to $\prod_{e \in E(T)} c_e$.

- Direct applications
  - Symmetric [GSS11] and assymetric [AGM+10] traveling salesman
  - Dependent rounding

Motivation

Given an undirected graph G with weights (conductances) $\{c_e\}_{e \in E(G)}$ on its edges, sample a spanning tree T of G with probability proportional to $\prod_{e \in E(T)} c_e$.

- Direct applications
  - Symmetric [GSS11] and assymetric [AGM$^+$10] traveling salesman
  - Dependent rounding
  - Sparser cut sparsifiers [FH10, GRV09]

Motivation

Given an undirected graph G with weights (conductances) $\{c_e\}_{e \in E(G)}$ on its edges, sample a spanning tree T of G with probability proportional to $\prod_{e \in E(T)} c_e$.

- Direct applications
  - Symmetric [GSS11] and assymetric [AGM$^+$10] traveling salesman
  - Dependent rounding
  - Sparser cut sparsifiers [FH10, GRV09]
- Probability theory

Motivation

Given an undirected graph G with weights (conductances) $\{c_e\}_{e \in E(G)}$ on its edges, sample a spanning tree T of G with probability proportional to $\prod_{e \in E(T)} c_e$.

- Direct applications
  - Symmetric [GSS11] and assymetric [AGM+10] traveling salesman
  - Dependent rounding
  - Sparser cut sparsifiers [FH10, GRV09]
- Probability theory
- Sampling from determinantal point processes and correlated distributions

Motivation

Given an undirected graph G with weights (conductances) $\{c_e\}_{e \in E(G)}$ on its edges, sample a spanning tree T of G with probability proportional to $\prod_{e \in E(T)} c_e$.

- Direct applications
  - Symmetric [GSS11] and assymetric [AGM$^+$10] traveling salesman
  - Dependent rounding
  - Sparser cut sparsifiers [FH10, GRV09]
- Probability theory
- Sampling from determinantal point processes and correlated distributions
- Algorithmic applications of electrical flows

Matrix-based algorithms

m: number of edges
n: number of vertices
Idea: go through edges one by one and flip coins conditioned on prior choices

- Matrix-based algorithms (runtimes for weighted graphs)

Matrix-based algorithms

m: number of edges
n: number of vertices
Idea: go through edges one by one and flip coins conditioned on prior choices

- Matrix-based algorithms (runtimes for weighted graphs)
    - [Gue83] $O(mn^{\omega})$
    - [CMN96] $\tilde{O}(n^{\omega})$

Matrix-based algorithms

m: number of edges

n: number of vertices

Idea: go through edges one by one and flip coins conditioned on prior choices

- Matrix-based algorithms (runtimes for weighted graphs)
  - [Gue83] $O(mn^{\omega})$
  - [CMN96] $\tilde{O}(n^{\omega})$
  - [DKP$^+$17] $\tilde{O}(n^{4/3}m^{1/2} + n^2)$

Matrix-based algorithms

m: number of edges

n: number of vertices

Idea: go through edges one by one and flip coins conditioned on prior choices

- Matrix-based algorithms (runtimes for weighted graphs)
  - [Gue83] $O(mn^{\omega})$
  - [CMN96] $\tilde{O}(n^{\omega})$
  - [DKP$^+$17] $\tilde{O}(n^{4/3}m^{1/2} + n^2)$
  - [DPPR17] $\tilde{O}(n^2\delta^{-2})$

Matrix-based algorithms

m: number of edges

n: number of vertices

Idea: go through edges one by one and flip coins conditioned on prior choices

- Matrix-based algorithms (runtimes for weighted graphs)
  - [Gue83] $O(mn^{\omega})$
  - [CMN96] $\tilde{O}(n^{\omega})$
  - [DKP^{+}17] $\tilde{O}(n^{4/3}m^{1/2} + n^2)$
  - [DPPR17] $\tilde{O}(n^2\delta^{-2})$
  - No runtime dependence on weights (☺)

Matrix-based algorithms

m: number of edges

n: number of vertices

Idea: go through edges one by one and flip coins conditioned on prior choices

- Matrix-based algorithms (runtimes for weighted graphs)
  - [Gue83] $O(mn^{\omega})$
  - [CMN96] $\tilde{O}(n^{\omega})$
  - [DKP+17] $\tilde{O}(n^{4/3}m^{1/2} + n^2)$
  - [DPPR17] $\tilde{O}(n^2\delta^{-2})$
  - No runtime dependence on weights (☺)
  - Quadratic for sparse graphs (☹)

Aldous-Broder: a random walk-based algorithm

Algorithm:

- Pick an arbitrary vertex $u \in V(G)$.
- Do a random walk until all vertices have been visited.
- Return the edges used to visit each vertex for the first time.

Aldous-Broder: a random walk-based algorithm

Algorithm:

- Pick an arbitrary vertex $u \in V(G)$.
- Do a random walk until all vertices have been visited.
- Return the edges used to visit each vertex for the first time.

Generates a weighted uniformly random spanning tree of G!

Aldous-Broder: a random walk-based algorithm

Algorithm:

- Pick an arbitrary vertex $u \in V(G)$.
- Do a random walk until all vertices have been visited.
- Return the edges used to visit each vertex for the first time.

Generates a weighted uniformly random spanning tree of G!

- Runtime: cover time, which can be $\Omega(mn)$ ☹

Aldous-Broder: a random walk-based algorithm

Algorithm:

- Pick an arbitrary vertex $u \in V(G)$.
- Do a random walk until all vertices have been visited.
- Return the edges used to visit each vertex for the first time.

Generates a weighted uniformly random spanning tree of G!

- Runtime: cover time, which can be $\Omega(mn)$ ☹
- Only need first visits (at most n such visits) ☺

History

m: number of edges

n: number of vertices

- Matrix-based algorithms (runtimes for weighted graphs)
  - [Gue83]
  - [CMN96] $\tilde{O}(n^\omega)$
  - [DKP$^+$17] $\tilde{O}(n^{4/3}m^{1/2} + n^2)$
  - [DPPR17] $\tilde{O}(n^2\delta^{-2})$
  - No runtime dependence on weights (☺)
  - Quadratic for sparse graphs (☹)

History

m: number of edges

n: number of vertices

- Matrix-based algorithms (runtimes for weighted graphs)
  - [Gue83]
  - [CMN96] $\tilde{O}(n^\omega)$
  - [DKP$^+$17] $\tilde{O}(n^{4/3}m^{1/2} + n^2)$
  - [DPPR17] $\tilde{O}(n^2\delta^{-2})$
  - No runtime dependence on weights (☺)
  - Quadratic for sparse graphs (☹)
- Random-walk-based algorithms (runtimes for unweighted graphs)

History

m: number of edges

n: number of vertices

- Matrix-based algorithms (runtimes for weighted graphs)
  - [Gue83]
  - [CMN96] $\tilde{O}(n^\omega)$
  - [DKP+17] $\tilde{O}(n^{4/3}m^{1/2} + n^2)$
  - [DPPR17] $\tilde{O}(n^2\delta^{-2})$
  - No runtime dependence on weights (☺)
  - Quadratic for sparse graphs (☹)
- Random-walk-based algorithms (runtimes for unweighted graphs)
  - [Bro89, Ald90] $\tilde{O}(mn)$
  - [KM09] $\tilde{O}(m\sqrt{n})$
  - [MST15] $\tilde{O}(m^{4/3})$

History

m: number of edges

n: number of vertices

- Matrix-based algorithms (runtimes for weighted graphs)
  - [Gue83]
  - [CMN96] $\tilde{O}(n^{\omega})$
  - [DKP+17] $\tilde{O}(n^{4/3}m^{1/2} + n^2)$
  - [DPPR17] $\tilde{O}(n^2\delta^{-2})$
  - No runtime dependence on weights (☺)
  - Quadratic for sparse graphs (☹)
- Random-walk-based algorithms (runtimes for unweighted graphs)
  - [Bro89, Ald90] $\tilde{O}(mn)$
  - [KM09] $\tilde{O}(m\sqrt{n})$
  - [MST15] $\tilde{O}(m^{4/3})$
  - Polynomial dependence on weights (☹)

History

m: number of edges

n: number of vertices

- Matrix-based algorithms (runtimes for weighted graphs)
  - [Gue83]
  - [CMN96] $\tilde{O}(n^{\omega})$
  - [DKP+17] $\tilde{O}(n^{4/3}m^{1/2} + n^2)$
  - [DPPR17] $\tilde{O}(n^2\delta^{-2})$
  - No runtime dependence on weights (☺)
  - Quadratic for sparse graphs (☹)
- Random-walk-based algorithms (runtimes for unweighted graphs)
  - [Bro89, Ald90] $\tilde{O}(mn)$
  - [KM09] $\tilde{O}(m\sqrt{n})$
  - [MST15] $\tilde{O}(m^{4/3})$
  - Polynomial dependence on weights (☹)
  - Subquadratic running time (☺)

History

m: number of edges

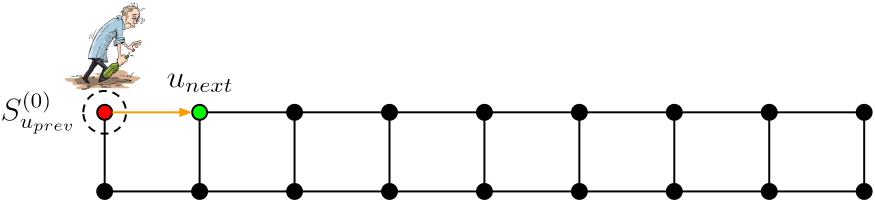
n: number of vertices

- Matrix-based algorithms (runtimes for weighted graphs)
  - [Gue83]
  - [CMN96] $\tilde{O}(n^\omega)$
  - [DKP$^+$17] $\tilde{O}(n^{4/3}m^{1/2} + n^2)$
  - [DPPR17] $\tilde{O}(n^2\delta^{-2})$
  - No runtime dependence on weights (☺)
  - Quadratic for sparse graphs (☹)
- Random-walk-based algorithms (runtimes for unweighted graphs)
  - [Bro89, Ald90] $\tilde{O}(mn)$
  - [KM09] $\tilde{O}(m\sqrt{n})$
  - [MST15] $\tilde{O}(m^{4/3})$
  - Polynomial dependence on weights (☹)
  - Subquadratic running time (☺)
  - This work $\tilde{O}(m^{1+o(1)})$

Aldous-Broder remix

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$

- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(0)}$.
  - Replace u with the non-$S_u^{(0)}$ endpoint of this edge.
- Return the edges used to visit each vertex for the first time.

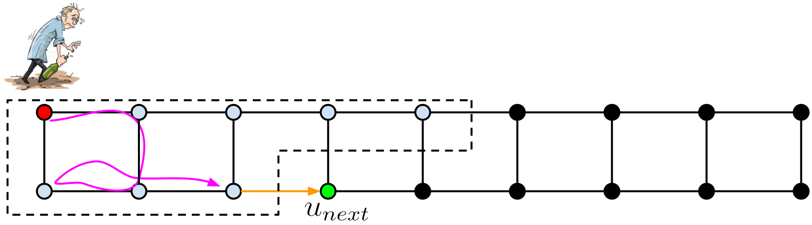


[0] laohblogger.com

Wishful thinking

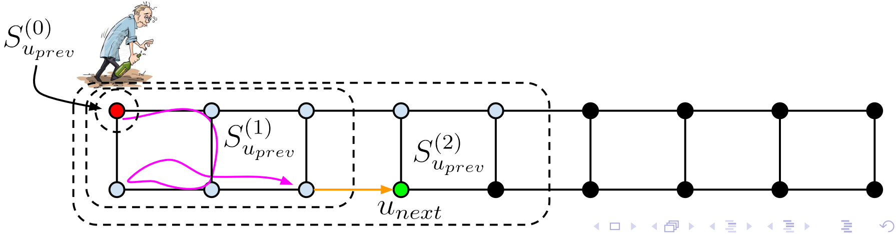
- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$

- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - Sample the edge that the random walk starting at u uses to exit the set of visited vertices.
  - Replace u with the unvisited endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



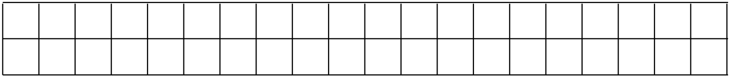
Shortcutting meta-algorithm

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$ and **pick** shortcutters $\{S_v^{(i)}\}_{i=1}^{\sigma_0}$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - Let $i^* \in \{0, 1, \ldots, \sigma_0\}$ be the maximum value of i for which all vertices in $S_u^{(i)}$ have been visited.
  - **Sample** the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the non-$S_u^{(i^*)}$ endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



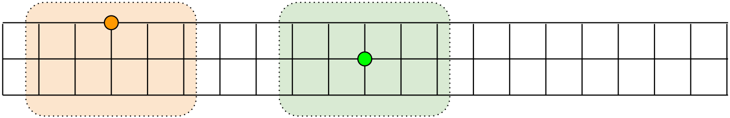
Example

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$ and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - Let $i^* \in \{0, 1, 2, 3\}$ be the maximum value of i for which all vertices in $S_u^{(i)}$ have been visited.
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



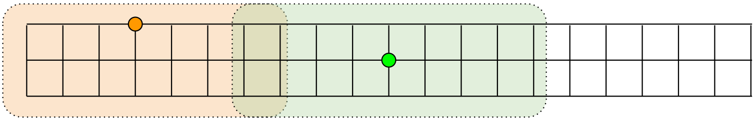
Example

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and **pick** shortcutters $S_v^{(1)}$ to be the 2-neighborhood of v for all $v \in V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - Let $i^* \in \{0, 1, 2, 3\}$ be the maximum value of i for which all vertices in $S_u^{(i)}$ have been visited.
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



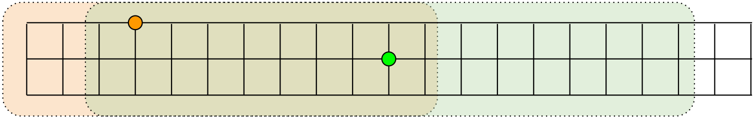
Example

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$ and **pick** shortcutters $S_v^{(2)}$ to be the 4-neighborhood of v for all $v \in V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - Let $i^* \in \{0, 1, 2, 3\}$ be the maximum value of i for which all vertices in $S_u^{(i)}$ have been visited.
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



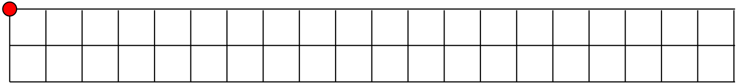
Example

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$ and **pick** shortcutters $S_v^{(3)}$ to be the 8-neighborhood of v for all $v \in V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - Let $i^* \in \{0, 1, 2, 3\}$ be the maximum value of i for which all vertices in $S_u^{(i)}$ have been visited.
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



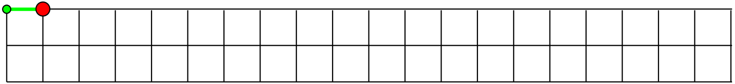
Example on walk step 1

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- **Pick** an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



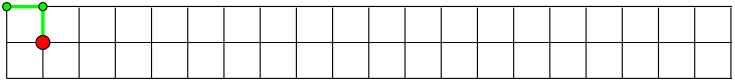
Example on walk step 2

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$ and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



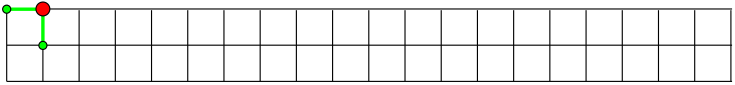
Example on walk step 3

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^{3}$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



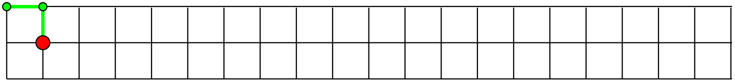
Example on walk step 4

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



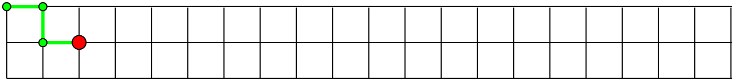
Example on walk step 5

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^{3}$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



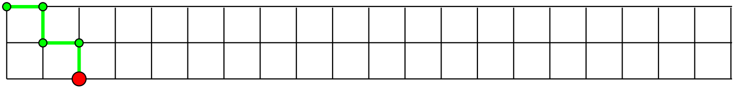
Example on walk step 6

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



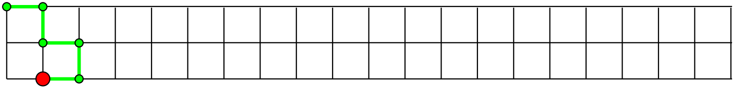
Example on walk step 7

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^{3}$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



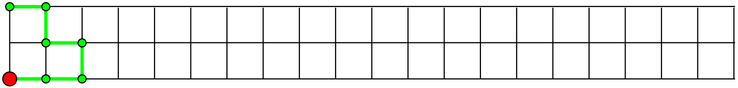
Example on walk step 8

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^{3}$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



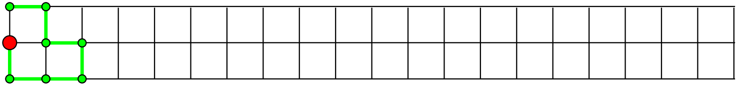
Example on walk step 9

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^{3}$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



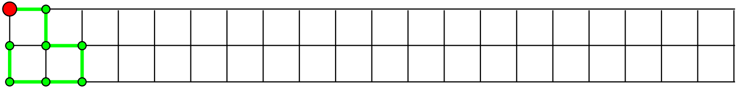
Example on walk step 10

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



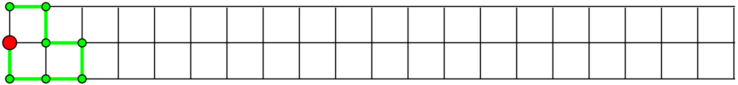
Example on walk step 11

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



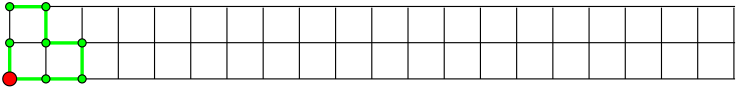
Example on walk step 12

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



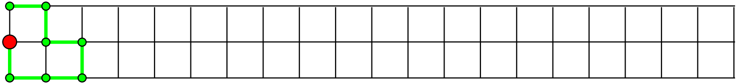
Example on walk step 13

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



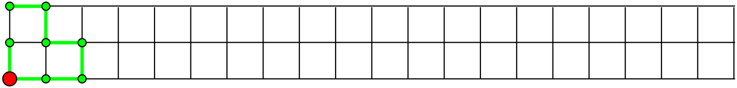
Example on walk step 14

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^{3}$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



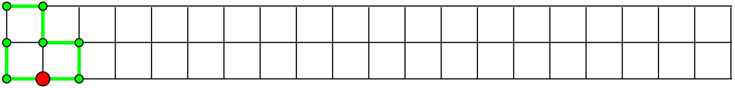
Example on walk step 15

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^{3}$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



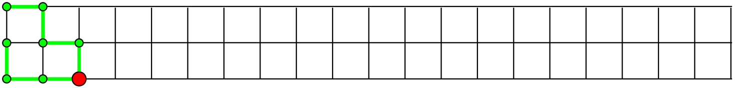
Example on walk step 16

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^{3}$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



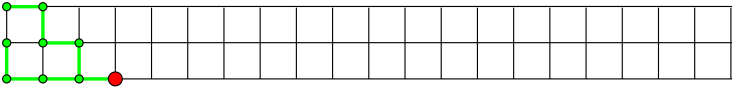
Example on walk step 17

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



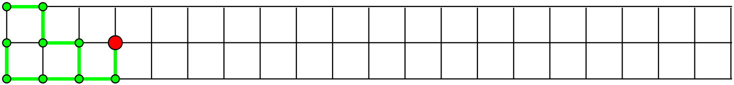
Example on walk step 18

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^{3}$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



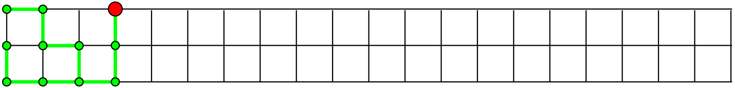
Example on walk step 19

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



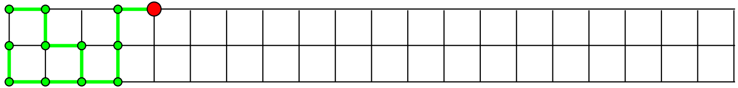
Example on walk step 20

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



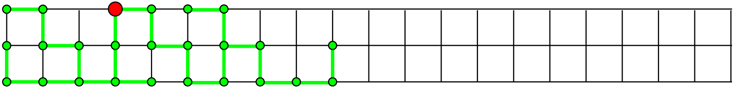
Example on walk step 21

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



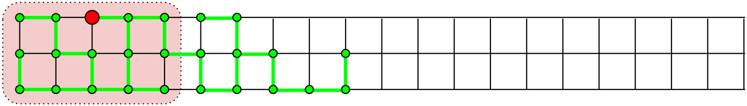
Example on walk step 64

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



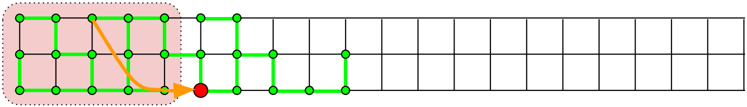
Example on walk step 65

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 1$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



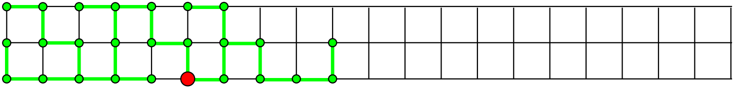
Example on walk step 65 $\rightarrow$ 120

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 1$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



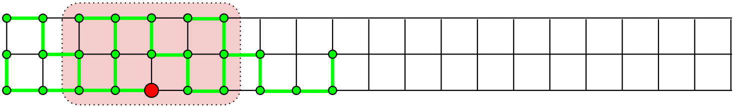
Example on walk step 120

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



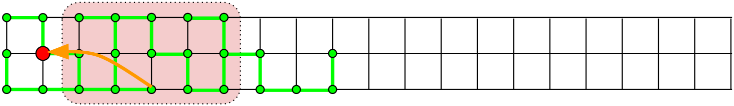
Example on walk step 121

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 1$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



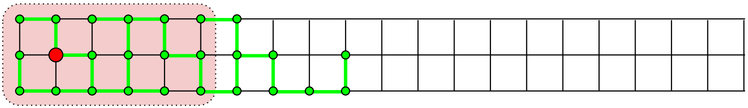
Example on walk step 121 $\to$ 127

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^{3}$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 1$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



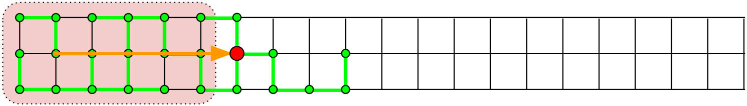
Example on walk step 127

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 2$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



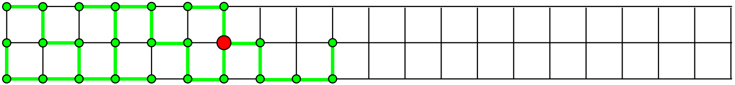
Example on walk step 127 $\rightarrow$ 204

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 2$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



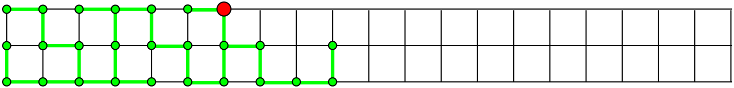
Example on walk step 204

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



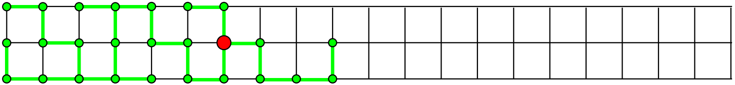
Example on walk step 205

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^{3}$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



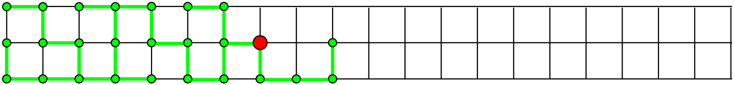
Example on walk step 206

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



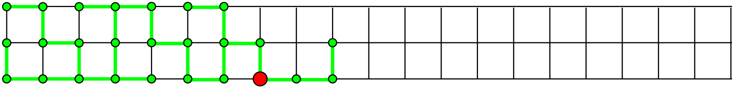
Example on walk step 207

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



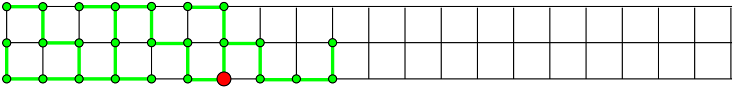
Example on walk step 208

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^{3}$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



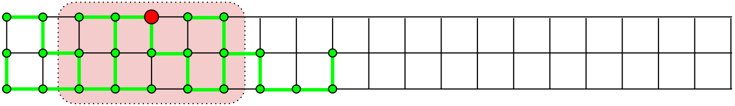
Example on walk step 209

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



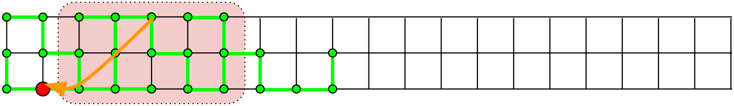
Example on walk step 213

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^{3}$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 1$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



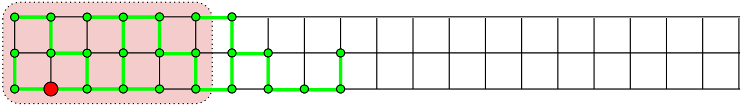
Example on walk step 213 $\to$ 246

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 1$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



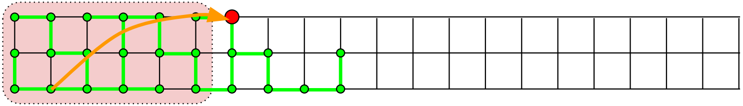
Example on walk step 246

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 2$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



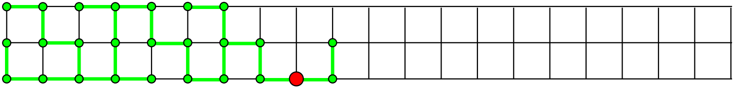
Example on walk step 246 $\rightarrow$ 307

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$ and pick shortcutters $\{S_v^{(i)}\}_{i=1}^{3}$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 2$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



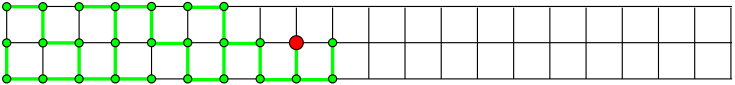
Example on walk step 313

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



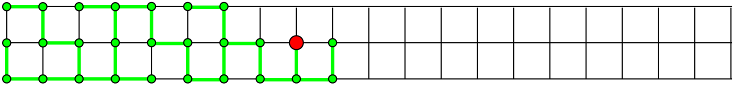
Example on walk step 314

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



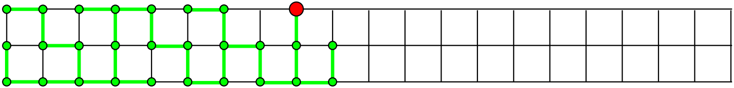
Example on walk step 316

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$ and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



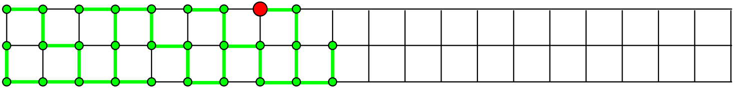
Example on walk step 317

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^{3}$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



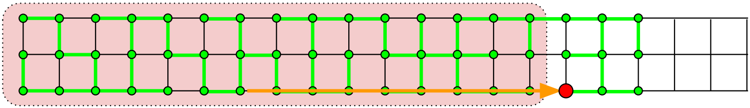
Example on walk step 318

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^{3}$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 0$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



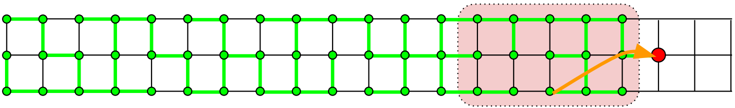
Example on walk step 485 $\rightarrow$ 821

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$ and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 3$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



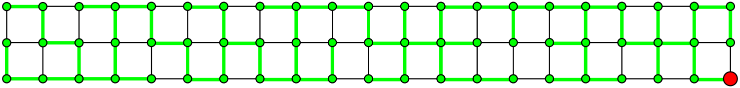
Example on walk step 821 $\rightarrow$ 826

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$ and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - $i^* \leftarrow 1$
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- Return the edges used to visit each vertex for the first time.



Example on walk step 875

- For each $v \in V(G)$, let $S_v^{(0)} = \{v\}$
  and pick shortcutters $\{S_v^{(i)}\}_{i=1}^3$ with $v \in S_v^{(i)} \subseteq V(G)$
- Pick an arbitrary vertex $u \in V(G)$.
- While there is an unvisited vertex
  - **DONE**
  - Sample the edge that the random walk starting at u uses to exit $S_u^{(i^*)}$.
  - Replace u with the endpoint of this edge.
- **Return** the edges used to visit each vertex for the first time.



Summary of existing shortcutting-based algorithms

σ_0: number of shortcutters

m: number of edges

n: number of vertices

Algorithm	σ_0	Shortcutting method	Runtime
[Bro89, Ald90]			$O(mn)$
[KM09]			$O(m\sqrt{n})$
[MST15]			$O(m^{4/3})$
This work			$O(m^{1+1/\sigma_0})$

Summary of existing shortcutting-based algorithms

σ_0: number of shortcutters
m: number of edges
n: number of vertices

Algorithm	σ_0	Shortcutting method	Runtime
[Bro89, Ald90]	0		$O(mn)$
[KM09]			$O(m\sqrt{n})$
[MST15]			$O(m^{4/3})$
This work			$O(m^{1+1/\sigma_0})$

Summary of existing shortcutting-based algorithms

σ_0: number of shortcutters
m: number of edges
n: number of vertices

Algorithm	σ_0	Shortcutting method	Runtime
[Bro89, Ald90]	0	N/A	$O(mn)$
[KM09]			$O(m\sqrt{n})$
[MST15]			$O(m^{4/3})$
This work			$O(m^{1+1/\sigma_0})$

Summary of existing shortcutting-based algorithms

σ_0: number of shortcutters
m: number of edges
n: number of vertices

Algorithm	σ_0	Shortcutting method	Runtime
[Bro89, Ald90]	0	N/A	$O(mn)$
[KM09]	1		$O(m\sqrt{n})$
[MST15]			$O(m^{4/3})$
This work			$O(m^{1+1/\sigma_0})$

Summary of existing shortcutting-based algorithms

σ_0: number of shortcutters
m: number of edges
n: number of vertices

Algorithm	σ_0	Shortcutting method	Runtime
[Bro89, Ald90]	0	N/A	$O(mn)$
[KM09]	1		$O(m\sqrt{n})$
[MST15]	2		$O(m^{4/3})$
This work			$O(m^{1+1/\sigma_0})$

Summary of existing shortcutting-based algorithms

σ_0: number of shortcutters
m: number of edges
n: number of vertices

Algorithm	σ_0	Shortcutting method	Runtime
[Bro89, Ald90]	0	N/A	$O(mn)$
[KM09]	1		$O(m\sqrt{n})$
[MST15]	2		$O(m^{4/3})$
This work	$\tilde{\Theta}(\log \log n)$		$O(m^{1+1/\sigma_0})$

Summary of existing shortcutting-based algorithms

σ_0: number of shortcutters
m: number of edges
n: number of vertices

Algorithm	σ_0	Shortcutting method	Runtime
[Bro89, Ald90]	0	N/A	$O(mn)$
[KM09]	1	Offline	$O(m\sqrt{n})$
[MST15]	2	Offline	$O(m^{4/3})$
This work	$\tilde{\Theta}(\log\log n)$		$O(m^{1+1/\sigma_0})$

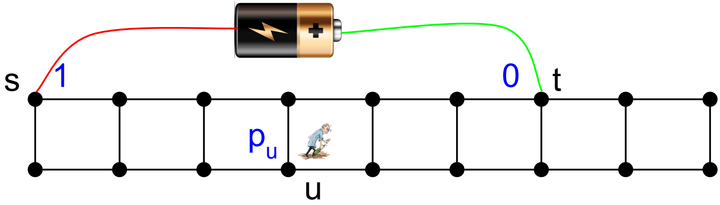
Summary of existing shortcutting-based algorithms

σ_0: number of shortcutters
m: number of edges
n: number of vertices

Algorithm	σ_0	Shortcutting method	Runtime
[Bro89, Ald90]	0	N/A	$O(mn)$
[KM09]	1	Offline	$O(m\sqrt{n})$
[MST15]	2	Offline	$O(m^{4/3})$
This work	$\tilde{\Theta}(\log \log n)$	Online	$O(m^{1+1/\sigma_0})$

Using Laplacian solvers to calculate hitting probabilities

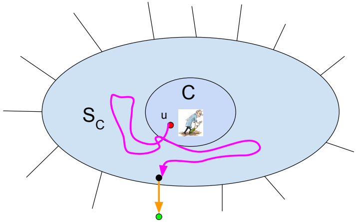
$$p_u = \Pr_u[\text{ random walk starting at } u \text{ hits } s \text{ before } t]$$



Can compute all p_us in $\tilde{O}(m)$ time! [ST14]

Shortcutting methods

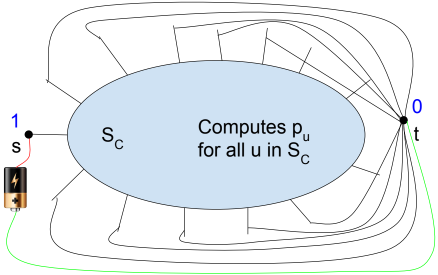
- **Given:** a shortcutter S_C with $C \subseteq S_C \subseteq V(G)$
- **Goal:** sample the edge that the random walk starting at u uses to exit S_C for some $u \in C$



Shortcutting method	Preprocessing	Query

Offline shortcutting

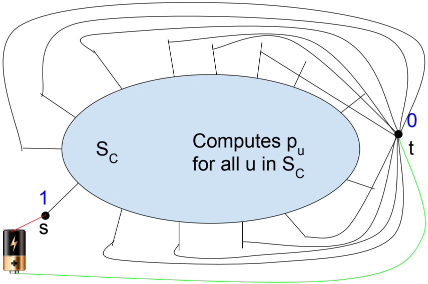
- **Given:** a shortcutter S_C with $C \subseteq S_C \subseteq V(G)$
- **Goal:** sample the edge that the random walk starting at u uses to exit S_C for some $u \in C$



Shortcutting method	Preprocessing	Query
Offline	$\tilde{O}(\lvert E(S_C)\rvert \lvert \partial S_C \rvert)$	$O(\log n)$

Offline shortcutting

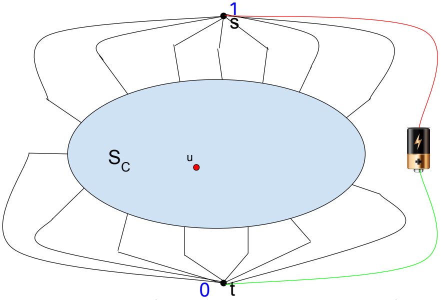
- **Given:** a shortcutter S_C with $C \subseteq S_C \subseteq V(G)$
- **Goal:** sample the edge that the random walk starting at u uses to exit S_C for some $u \in C$



Shortcutting method	Preprocessing	Query
Offline	$\tilde{O}(\lvert E(S_C)\rvert \lvert \partial S_C \rvert)$	$O(\log n)$

Online shortcutting

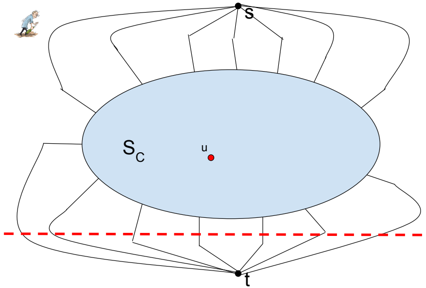
- **Given:** a shortcutter S_C with $C \subseteq S_C \subseteq V(G)$
- **Goal:** sample the edge that the random walk starting at u uses to exit S_C for some $u \in C$



Shortcutting method	Preprocessing	Query
Offline	$\tilde{O}(\lvert E(S_C)\rvert \lvert \partial S_C \rvert)$	$O(\log n)$
Online	None	$\tilde{O}(\lvert E(S_C)\rvert)$

Online shortcutting

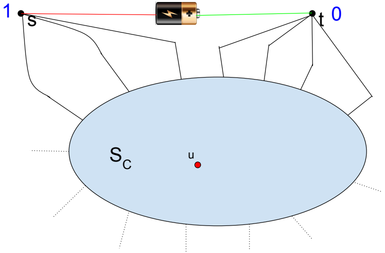
- **Given:** a shortcutter S_C with $C \subseteq S_C \subseteq V(G)$
- **Goal:** sample the edge that the random walk starting at u uses to exit S_C for some $u \in C$



Shortcutting method	Preprocessing	Query
Offline	$\tilde{O}(\lvert E(S_C)\rvert \lvert \partial S_C \rvert)$	$O(\log n)$
Online	None	$\tilde{O}(\lvert E(S_C)\rvert)$

Online shortcutting

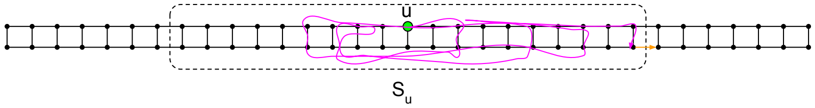
- **Given:** a shortcutter S_C with $C \subseteq S_C \subseteq V(G)$
- **Goal:** sample the edge that the random walk starting at u uses to exit S_C for some $u \in C$



Shortcutting method	Preprocessing	Query
Offline	$\tilde{O}(\lvert E(S_C)\rvert \lvert\partial S_C\rvert)$	$O(\log n)$
Online	None	$\tilde{O}(\lvert E(S_C)\rvert)$

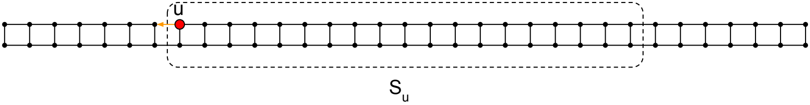
Using online shortcutting

Shortcutter work: $\tilde{O}(|E(S_u)|)$, random walk work $\Omega(|E(S_u)|^2)$ ☺



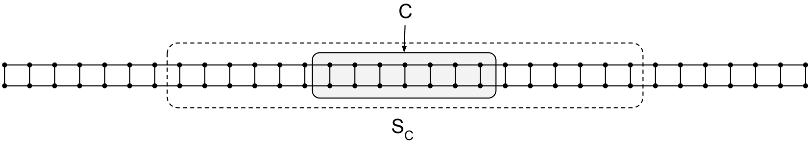
Using online shortcutting

Shortcutter work: $\tilde{\Omega}(|E(S_u)|)$, random walk work can be $O(1)$ ☹



Using online shortcutting

Core should be "well-separated" from the boundary of the shortcutter



Bounding work of online shortcutting

Bounding work of online shortcutting

Most random walk steps occur far away from an unvisited vertex

Bounding work of online shortcutting

Most random walk steps occur far away from an unvisited vertex

Lemma (Random walk bound)

Consider a random walk starting at an arbitrary vertex in a graph I and an edge $\{u, v\} = f \in E(I)$. The

- *expected number of times the random walk traverses f from $u \to v$*
- *before the distance R-neighborhood of u is covered*
- *is at most $\tilde{O}(c_f R)$, where c_f is the conductance of the edge f*

Schur complements

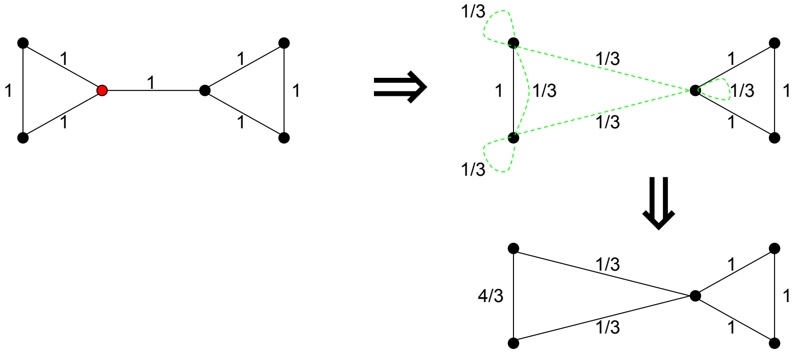
Let $\texttt{Schur}(G, S)$ be a graph with Laplacian matrix $A - BD^{-1}C$:

$$L = \begin{array}{cc} \left[\begin{array}{c|c} A := L[S,S] & B \\ \hline C := L[\bar{S},S] & D \end{array}\right] & \begin{array}{c} S \\ \bar{S} \end{array} \\ \begin{array}{cc} \quad S \quad & \quad \bar{S} \end{array} & \end{array}$$

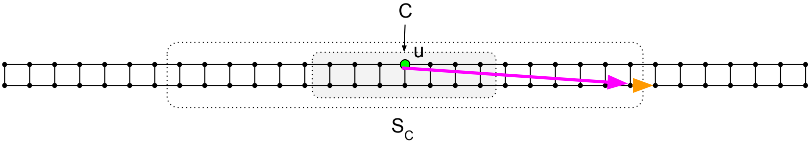
Graph-theoretic interpretation of Schur complements

$H := \texttt{Schur}(G, S)$: a graph with $V(H) = S$ and $E(H)$ and weights defined with the property that the following two distributions are equivalent:

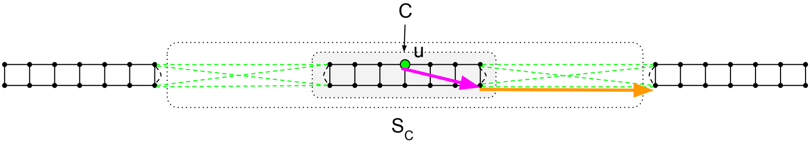
- the list of vertices visited by a random walk in H
- the list of vertices visited by a random walk in G with all vertices outside of S omitted



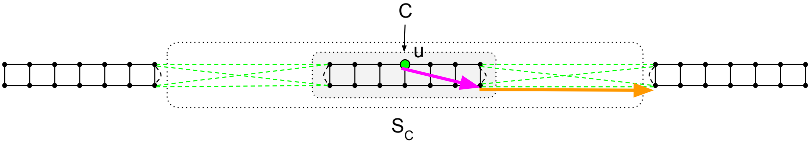
Charging shortcutter uses to Schur complement crossings



Charging shortcutter uses to Schur complement crossings

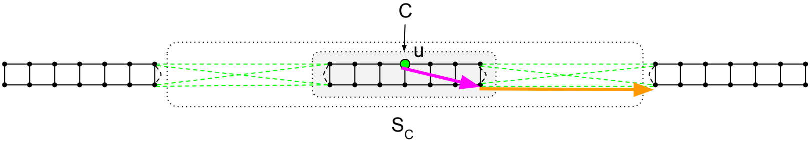


Charging shortcutter uses to Schur complement crossings



$R_i := m^{i/(\sigma_0+1)}$. To get almost-linear time,

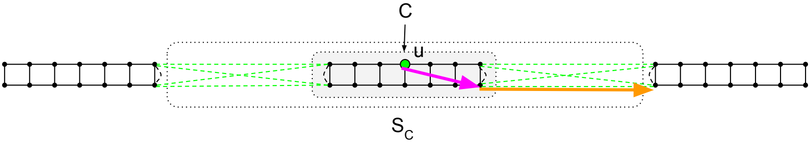
Charging shortcutter uses to Schur complement crossings



$R_i := m^{i/(\sigma_0+1)}$. To get almost-linear time,

- total green conductance $\leq \frac{1}{R_i}$ for $S_C^{(i)}$

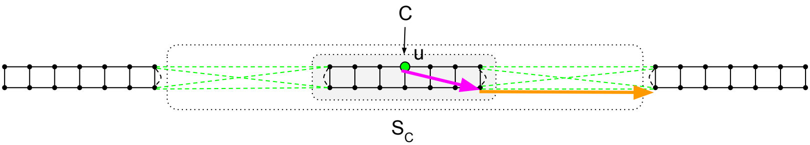
Charging shortcutter uses to Schur complement crossings



$R_i := m^{i/(\sigma_0+1)}$. To get almost-linear time,

- total green conductance $\leq \frac{1}{R_i}$ for $S_C^{(i)}$
- and distance to unvisited vertex $\leq R_{i+1}$

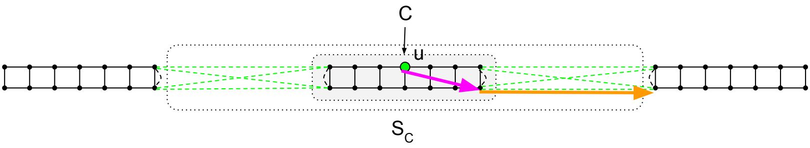
Charging shortcutter uses to Schur complement crossings



$R_i := m^{i/(\sigma_0+1)}$. To get almost-linear time,

- total green conductance $\leq \frac{1}{R_i}$ for $S_C^{(i)}$
- and distance to unvisited vertex $\leq R_{i+1}$
- implies number of uses is at most (distance to unvisited vertex)(weight of edges) $\leq \tilde{O}(R_{i+1}/R_i) = m^{o(1)}$ by edge crossing bound

Charging shortcutter uses to Schur complement crossings



$R_i := m^{i/(\sigma_0+1)}$. To get almost-linear time,

- total green conductance $\leq \frac{1}{R_i}$ for $S_C^{(i)}$
- and distance to unvisited vertex $\leq R_{i+1}$
- implies number of uses is at most (distance to unvisited vertex)(weight of edges) $\leq \tilde{O}(R_{i+1}/R_i) = m^{o(1)}$ by edge crossing bound
- work $\leq$ (total shortcutter size)(max number of uses) $\leq (m^{1+o(1)})m^{o(1)} = m^{1+o(1)}$

Conclusion

- An $m^{1+o(1)}\alpha^{o(1)}$-time algorithm for generating weighted uniformly random spanning trees
- Overcame barriers in graph-partitioning based approaches from before by using Schur complements
- Paradigm relevant for other problems?

Questions?

Bibliography

Arash Asadpour, Michel X. Goemans, Aleksander Madry, Shayan Oveis Gharan, and Amin Saberi.
An o(log n/ log log n)-approximation algorithm for the asymmetric traveling salesman problem.
In Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '10, pages 379–389, Philadelphia, PA, USA, 2010. Society for Industrial and Applied Mathematics.

David J. Aldous.
The random walk construction of uniform spanning trees and uniform labelled trees.
SIAM J. Discret. Math., 3(4):450–465, November 1990.

A. Broder.
Generating random spanning trees.
In Proceedings of the 30th Annual Symposium on Foundations of Computer Science, SFCS '89, pages 442–447, Washington, DC, USA, 1989. IEEE Computer Society.