Quantum Speedup for Sampling Random Spanning Trees

Chenghua Liu[1] Minbo Gao[1] Zhengfeng Ji[2] Simon Apers[3]

[1] Institute of Software, Chinese Academy of Sciences, Beijing, China

[2] Department of Computer Science and Technology, Tsinghua University, Beijing, China

[3] Université Paris Cité, CNRS, IRIF, Paris, France

Introduction

Random Spanning Trees-1

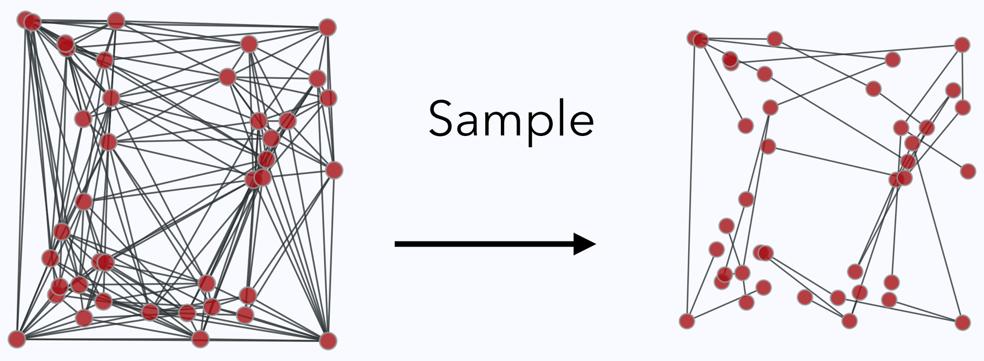
$G = (V, E, w),\ |V| = n, |E| = m, w \in \mathbb{R}_+^E$

Let $\mathcal{T}_G$ denote the set of all spanning trees of G

Define the distribution W_G on $\mathcal{T}_G$ by

$$P_{X \sim W_G}[X = T] \propto \prod_{e \in T} w_e$$

Goal: sample a spanning tree $T \sim W_G$



Many Applications: algorithm design, machine learning, statistics …

Random Spanning Trees-2

Sampling RSTs has been widely studied classically, with three main algorithmic approaches:

1. Determinant-Based Methods:
Guenoche 83 & Kulkarni 90: $O(mn^3)$;
Colbourn, Myrvold, and Neufeld 96: $O(n^\omega)$.

2. Effective Resistance-Based Methods:
Harvey and Xu 16: $O(n^\omega)$;
Durfee, Kyng, Peebles, Rao and Sachdeva 17: $\widetilde{O}(n^{4/3}m^{1/2} + n^2)$;
Durfee, Peebles, Peng and Rao 17: $\widetilde{O}(n^2/\varepsilon^2)$.

3. Random Walk-Based Methods：
Broder 89 & Aldous 90: $O(mn)$ for unweighted;
Wilson 96; Kelner, Madry 09; Madry, Straszak,Tarnawski 14;
Schild 18: $m^{1+o(1)}$;
The current state-of-the-art: based on down-up random walks, $O(m \log^2 m)$.

Our Results

Theorem.
There exists a quantum algorithm that, given query access to the adjacency list of a connected graph G and accuracy parameter ε, with high probability, outputs a spanning tree of G drawn from a distribution which is ε-close to W_G in total variation distance. The algorithm runs in $\widetilde{O}(\sqrt{mn}\log(1/\varepsilon))$ time.

Lower bound:
Let $\varepsilon < 1/2$ be a constant. For any graph G, consider the problem of sampling a random spanning tree from a distribution ε-close to W_G, given adjacency-list access to G. The quantum query complexity of this problem is $\Omega(\sqrt{mn})$.

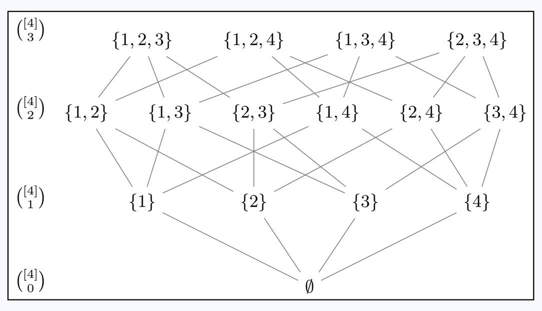
Preliminaries

Background: down-up walk

Consider a distribution μ over size-k subsets of $[m]$

We define the down operator $D_{k\to\ell}$ and the up operator $U_{\ell\to k}$, which map between sets of different sizes: from size-k to size-ℓl subsets, and vice versa.

$$U_{\ell\to k}(T,S) = \begin{cases} 0 & \text{if } T \nsubseteq S, \\ \dfrac{\mu(S)}{\sum_{S':T\subseteq S'} \mu(S')} & \text{otherwise} \end{cases} \qquad D_{k\to\ell}(S,T) = \begin{cases} 0 & \text{if } T \nsubseteq S, \\ \dfrac{1}{\binom{k}{\ell}} & \text{otherwise} \end{cases}$$



The **down operator** randomly selects a smaller subset (moving downward). And the **up operator** moves upward by choosing a size-k superset with probability proportional to $\mu(S)$.

Down-up walk for sampling RSTs

Starting from $S_0 \in$ **supp**(μ), one step of the down-up walk M_μ^t, $t \geq k+1$:

1. Sample $T \in \binom{[m]\backslash S_0}{t-k}$ uniformly at random

2. Let $S_1 \sim \mu_{S_0 \cup T}$, and update $S_0 \leftarrow S_1$ $\qquad \mu_{S_0\cup T}$ is μ restricted to $S_0 \cup T$

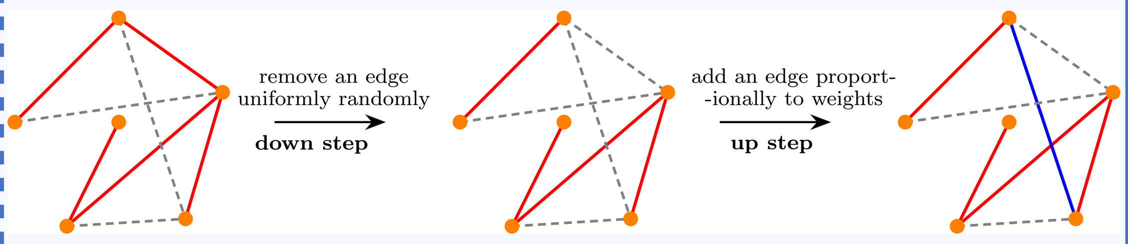
Lemma. Proposition 25 in [ADVY22]
The complement of S_1 is distributed according to $\bar{\mu}_0 D_{(m-k)\to(m-t)} U_{(m-t)\to(m-k)}$ if we start with $S_0 \sim \mu_0$, where $\bar{\mu}(S) := \mu([m] \setminus S)$. Moreover, for any distribution μ that is strongly Rayleigh, the chain is irreducible, aperiodic and has stationary distribution μ.

W_G is strongly Rayleigh, and 1-step down-up walk becomes:
Remove an edge uniformly randomly
Add a new edge sampled proportionally to the edge weights between components

Down-up walk for sampling RSTs

[Nima, Kuikui, Shayan, Cynthia, Thuy-Duong STOC21]

1-step down-up walk for sampling RSTs:
1. remove an edge to split the tree
2. add a new edge sampled proportionally to the edge weights between components
The chain has the mixing time $\widetilde{O}(n)$.



Their final algorithm uses an "up-down" walk: first add an edge, then remove one. Although the mixing time is $\widetilde{O}(m)$, each sample step runs in amortized $\widetilde{O}(1)$ time via link-cut trees.

Quantum Algorithm-1

Barrier and Idea for Quantum Speedups

Revisit the Down-Up Walk:
We can sample an edge between the two resulting components in $\widetilde{O}(\sqrt{m})$ time using Grover Search. But overall complexity remains $\widetilde{O}(\sqrt{mn}) \in \widetilde{\Omega}(m)$, offering no speedup.

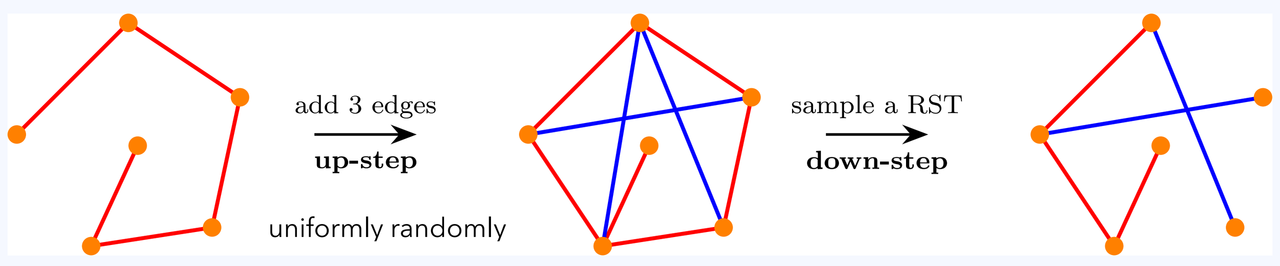
Inspired by domain sparsification techniques [AD20, ADVY22, ALV22]

Key Ideas：

- *Large-Step Walks*: Modify $\Theta(n)$ edges in each step (vs. 1 in classical), reducing mixing time to $\widetilde{O}(1)$.
- *Isotropy transformation*: Reduce sampling domain size from m to $O(n)$ using isotropic transformation (enables large-step walks to work well).

Quantum Sampling RSTs

Framework.
Large-Step Walks: Modify $\Theta(n)$ edges in each step, reducing mixing time to $\widetilde{O}(1)$.



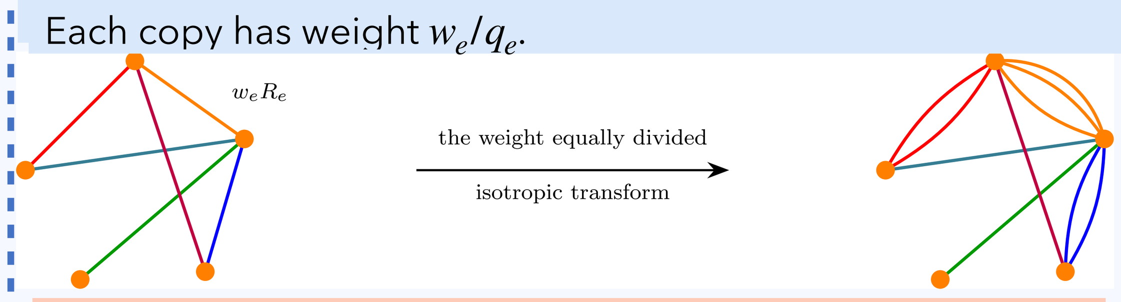
$n-1$ edges $\qquad$ $\Theta(n)$ edges

Requirement.
Perform an isotropic transformation–that is, adjust the graph so that the marginal probabilities of all edges are approximately equal.
The marginal probability of an edge e is given by:
$\Pr[e \in T, T \sim W_G] = w_e \cdot R(e)$

$R(e) := (\delta_i - \delta_j)^\top L_G^+(\delta_i - \delta_j),\ e = \{i, j\}$

Isotropic Transformation

Defintion.
Given a graph $G = (V, E, w)$ and a vector $\widetilde{R} \in \mathbb{R}^E$ approximating effective resistances R, the isotropic-transformed multigraph $G' = (V, E', w')$ is constructed as follows:
Each edge $e \in E$ is replaced by $q_e = \lceil m \cdot w_e \widetilde{R}_e/(2n) \rceil$ parallel edges.
Each copy has weight w_e/q_e.



Proposition.
The transformed graph satisfies $|E'| \leq 2m$, and the marginals are nearly uniform:
$\Pr[e' \in S, S \sim W_{G'}] \leq 2n/m = o(1)$.
Then the mixing time is $\widetilde{O}(1)$, by the analysis in [ALV22].

Quantum Algorithm-2

Implicit Isotropic Transformation

Quantum Graph Sparsification [AdW22] Time $\widetilde{O}(\sqrt{mn})$
Rather than explicitly computing this isotropic transformation, we utilize a quantum data structure $\mathscr{R}$ which provides quantum query access to effective resistances, to ``implicitly'' construct and maintain the isotropic-transformed multigraph.

Quantum Isotropic Sampling with $\mathscr{R}$ (up step) Time $\widetilde{O}(\sqrt{mn})$
Sample $\Theta(n)$ edges from the isotropic-transformed multigraph uniformly at random (a sampling-without-replacement variant of multiple-state preparation [Ham22]).

Quantum Minimum Spanning Tree [DHHM06] Time $\widetilde{O}(\sqrt{mn})$
Find a spanning tree with maximum product of edge weights as a "good" starting point for the down-up walk.

Quantum Lower Bound

Lower bound:
Let $\varepsilon < 1/2$ be a constant. For any graph G, consider the problem of sampling a random spanning tree from a distribution ε-close to W_G. The quantum query complexity of this problem is $\Omega(\sqrt{mn})$.

Follows via reduction from finding n marked elements among m, which has quantum query complexity $\Theta(\sqrt{mn})$. The reduction encodes the search into edge weights so that a uniform spanning tree reveals the marked elements. Similar to the $\Omega(\sqrt{mn})$ lower bound for MST in [DHHM06].

Open Questions & References

Open questions.
1. Faster algorithm for unweighted graphs?
2. The down-up walk is a powerful tool in classical algorithms (e.g., colorings, matchings). Can our quantum approach yield speedups for them?
3. Determinantal Point Processes (DPPs)?

References:

[AD20] Anari and Dereziński. *Isotropy and log-concave polynomials*. FOCS, 2020.
[ADVY22] Anari, Dereziński, Vuong, and Yang. *Domain sparsification via entropic independence*. ITCS, 2022.
[ALV22] Anari, Liu, and Vuong. *Optimal sublinear sampling of spanning trees and DPPs*. FOCS, 2022.
[AdW22] Apers and de Wolf. *Quantum speedup for graph sparsification and Laplacian solving*. SIAM J. Comput., 2022.
[Ham22] Hamoudi. *Preparing many copies of a quantum state in the black-box model*. Phys. Rev. A, 2022.
[DHHM06] Dürr, Heiligman, Høyer, and Mhalla. *Quantum query complexity of some graph problems*. SIAM J. Comput., 2006.

Poster designed using Keynote