Chennai Mathematical Institute

Masters Thesis

Random Spanning Trees

Author:
Bhishmaraj S

Supervisor:
Dr. Samir Datta

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Computer Science at CMI

June 7, 2020

Declaration of Authorship

I, Bhishmaraj S, declare that this thesis titled, "Random Spanning Trees" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself .

Signed:

Date:

CHENNAI MATHEMATICAL INSTITUTE

Abstract

Faculty Name
Computer Science at CMI

Master of Science

Random Spanning Trees

by Bhishmaraj S

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too. . . . I see

Acknowledgements

The acknowledgments and the people to thank go here, don't forget to include your project advisor…

Contents

List of Figures

List of Tables

List of Abbreviations

LAH	**L**ist **A**bbreviations **H**ere
WSF	**W**hat (it) **S**tands **F**or

1 Introduction

Spanning trees have been a central object of study in graph theory for a long time. Kirchhoff, 1847 established a linear algebraic relationship between spanning trees of graphs and determinants while studying electric networks. In this thesis we discuss about some algorithms to sample **uniform spanning trees** (all spanning trees are equally likely to be sampled).

Sampling spanning trees happen to be a primitive used in various problems such as -

- Constructing expanders (Goyal, Rademacher, and Vempala, 2009, Frieze et al., 2014)
- Approximation algorithms for the travelling salesman problem(Gharan, Saberi, and Singh, 2011, Asadpour et al., 2017)
- Graph Sparcification (Fung and Harvey, 2010)
- Analysis of network reliability (Colbourn, 1987,Nel and Colbourn, 1990, Colbourn, Debroni, and Myrvold, 1988)
- Sequence shuffling problem in Bioinformatics (Kandel et al., 1996)

Bar-Ilan and Zernik, 1989 proposed a distributed algorithm for this problem.

1.0.1 Algorithms for sampling random spanning trees

There has been a lot of activity from the 80's in solving this problem as fast as possible.

Determinant Based

These algorithms are based on Kirchoff Matrix Tree theorem and involve computing determinants of the laplacian matrix of the graph to sample spanning trees. These notions would be made clear in later part of the thesis.

- Random Spanning Tree, Guénoche, 1983
- Unranking and ranking spanning trees of a graph, Colbourn, Day, and Nel, 1989
- Generating random combinatorial objects, Kulkarni, 1990
- Two Algorithms for Unranking Arborescences, Colbourn, Myrvold, and Neufeld, 1996
- Generating random spanning trees via fast matrix multiplication, Harvey and Xu, 2016

Random Walk Based

These algorithms simulate a random walk on the input graph and try to consutruct a spanning tree from this simulation.

- Generating random spanning trees, Broder, 1989
- The random walk construction of uniform spanning trees and uniform labelled trees, Aldous, 1990
- Generating random spanning trees more quickly than the cover time, Wilson, 1996
- How to Get a Perfectly Random Sample from a Generic Markov Chain and Generate a Random Spanning Tree of a Directed Graph, Propp and Wilson, 1998

Approximation Algorithms

These are more recent algorithms which sample a uniform spanning tree with a high probability. The main theme here is to employ tools from approximation algorithms.

- Faster generation of random spanning trees, Kelner and Madry, 2009
- Sampling Random Spanning Trees Faster than Matrix Multiplication, Durfee et al., 2017
- An Almost-Linear Time Algorithm for Uniform Random Spanning Tree Generation, Schild, 2018

2 Background

2.1 Markov Chains

2.1.1 Fundamental theorem of Markov chain

2.1.2 Markov chain tree theorem

2.2 Results from Spectral Graph Theory

2.2.1 Kirchoff Matrix Tree Theorem

2.2.2 Some properties of Laplacian

2.3 Electric Networks

3 Random Walk Approach

3.1 Aldous, Broder

3.2 Wilson

4 Matrix Approach

4.1 Harvey, Xu

Harvey and Xu, 2016 proposed a $\mathcal{O}(N^{\omega})$ (where ω is the exponent of fast matrix multiplication) algorithm for sampling a uniform spanning tree which is much simpler compared to the one proposed earlier by CMN which also has the same running time. The initial starting point for the algorithm is using the relationship between effective resistance and the probability that an edge belongs to a spanning tree. And also the fact that sampling an edge corresponds to contracting it and discarding the edge corresponds to deleting it. The following naive chain rule algorithm works on the same principle.

Naive chain rule algorithm

Input: $G = (V, E)$ and L_G^+
Output: Set of edges corresponding to a random spanning tree
for $e = (u, v) \in E$ **do**
    $R_e^{\text{eff}} = (\chi_u - \chi_v)^T \, L_G^+ \, (\chi_u - \chi_v)$;
    if $(X \sim \mathit{Bernoulli}(R_e^{\text{eff}})) = 1$ **then**
        Add edge e to the spanning tree;
        $G = G/e$;
    else
        $G = G \setminus e$;
    end
    Update L_G^+ ;
end

Algorithm 1: Sampling uniform spanning tree using chain rule

Computing L_G^+ takes $\mathcal{O}(N^3)$ hence the overall running time is $\mathcal{O}(MN^3)$

4.1.1 Summary of the paper

On a high level the main ideas used in the paper are

1. Use a divide and conquer algorithm to break the graph into smaller parts and sample on each parts seperately and update the pseudoinverse of the laplacian lazily only on the subgraph when needed

2. The important insight here is that the sampling probability of an edge depends only on 4 entries of the pseudoinverse of the laplacian. Hence we don't need to update all the entries of the matrix when the graph is modified

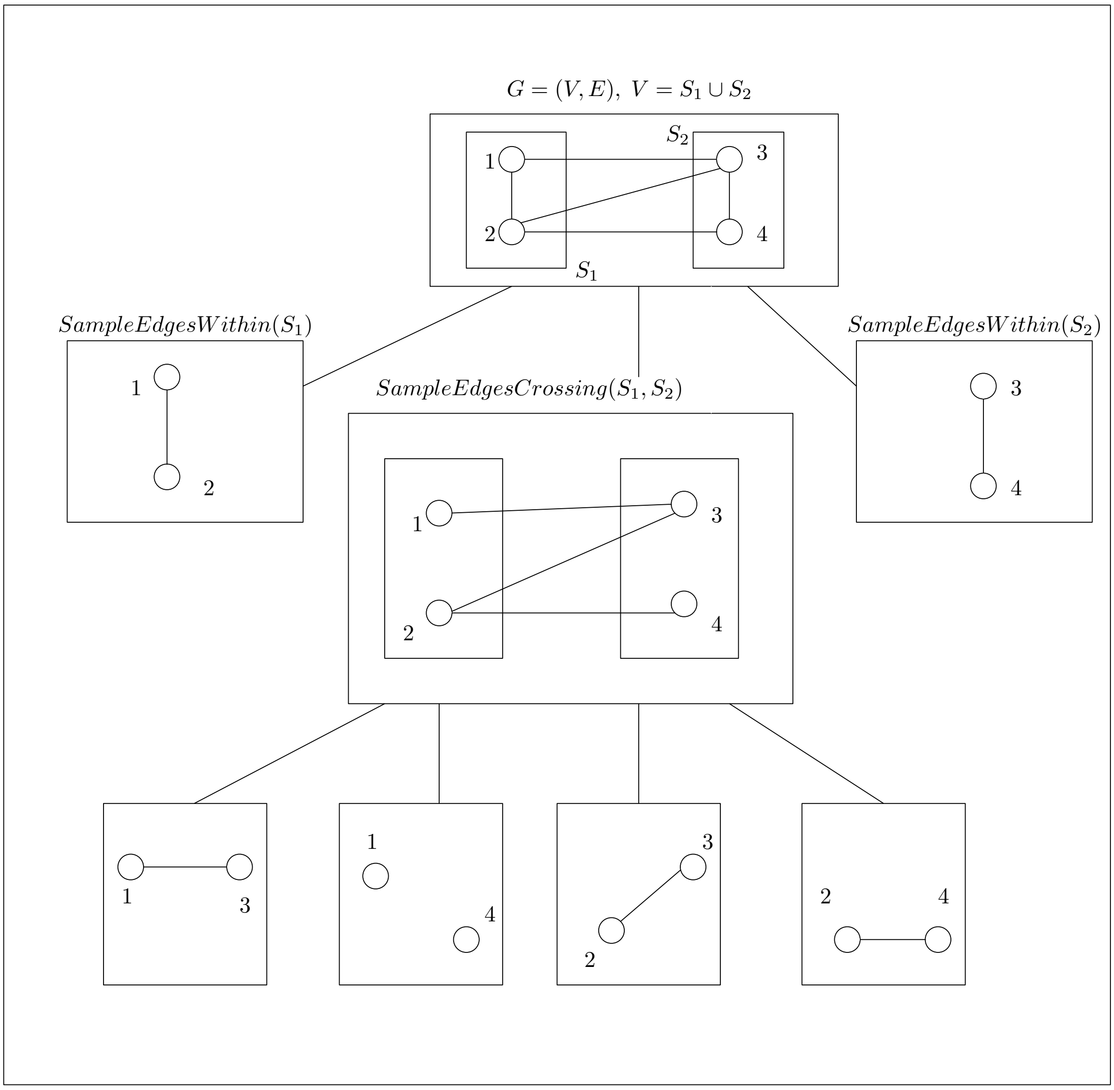
3. A well known method to compute inverse of a matrix with updates is to use the Sherman-Morrison-Woodbury formula. But in this case the formula has to be modified to work for the case where only a submatrix is modified.

4. Since while contracting an edge the number of vertices decreases it would get cumbersome to modify the dimension of the matrix evertime. So they overcome this issue by considering the formula on the limit case. When the graph is considered as a electric network then increasing the weight of an edge corresponds to shorting that link, hence in the limit case we get the same result as contracting the edge

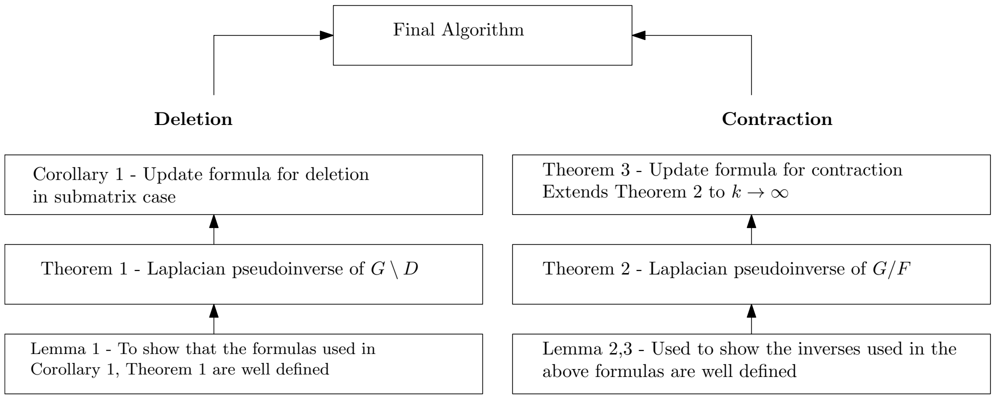
5. Also one of the main improvements over the previous algorithms (Colbourn, Myrvold, and Neufeld, 1996) is that the intricacies of LU decomposition is avoided since the current algorithm uses only matrix inversion as black box.

4.1.2 Harvey, Xu Algorithm

(TODO) Insert algorithm here and explain how it works



4.1.3 Structure of the paper



4.1.4 Facts used

Fact 1 (Woodbury matrix identity)**.** *Let $M \in \mathbb{M}_{n\times n}, U \in \mathbb{M}_{n\times k}, V \in \mathbb{M}_{n\times k}$. Suppose M is non-singular then $M+UV^T$ is non-singular $\iff$ $I+V^TM^{-1}U$ is non-singular. If $M+UV^T$ is non-singular, then*

$$(M+UV^T)^{-1} = M^{-1} - \left(M^{-1}\cdot U \cdot (I+V^TM^{-1}U)^{-1}\cdot V^T \cdot M^{-1}\right)$$

Fact 2. *For any $L \in \mathbb{M}_{n\times n}$ with $ker(L) = span(\mathbf{1})$, we have $LL^+ = I - \frac{\mathbf{1}\cdot\mathbf{1}^T}{n}$ and $P := I - \frac{\mathbf{1}\cdot\mathbf{1}^T}{n}$ is called the **projection matrix**.*

The folowing set of facts are about the properties of matrix operations (addition, multiplication, etc.) on sub-matrices. The first 4 are easy to see, so I haven't derived them. For the last one I have written a derivation using Shur's complement from Wikipedia

Fact 3 (Sub-matrices)**.** *For all the results below, S denotes a index set and S^c denotes it's complement.*

1. *For any $A, B \in \mathbb{M}_{n\times n}, (A+B)_{S,S} = A_{S,S} + B_{S,S}$*
2. *If $C = D\cdot E\cdot F$ then $C_{S,S} = D_{S,*}\cdot E \cdot F_{*,S}$*
3. *For $A \in \mathbb{M}_{m\times n}, B \in \mathbb{M}_{n\times l}$, If $A_{S^c,S^c} = 0$ or $B_{S^c,S^c} = 0$ then $(AB)_{S,S} = A_{S,S}\cdot B_{S,S}$*
4. *For any matrix C where $C = D\cdot E\cdot F$. If $D_{*,S^c} = 0$ and $F_{S^c,*} = 0$, then $C = D_{*,S}\cdot E_{S,S}\cdot F_{S,*}$*
5. *$D = \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix}$, and $E = \begin{bmatrix} A & B \\ X & Y \end{bmatrix}$ where $M, A \in \mathbb{M}_{n\times n}$ and If $(MA - I)$ is invertible Then,*

$$(DE-I)^{-1} = \begin{bmatrix} (MA-I)^{-1} & (MA-I)^{-1}\cdot M \cdot B \\ 0 & -I \end{bmatrix}$$

Proof. $(DE-I)^{-1}$ can be computed using Shur's Complement(Wikipedia contributors, 2020) .

Suppse $N = \begin{bmatrix} P & Q \\ R & S \end{bmatrix}$ and Shur's complement of block S and P is

$$N/S := P - QS^{-1}R \qquad N/P := S - RP^{-1}Q$$

Then

$$N^{-1} = \begin{bmatrix} P^{-1} + (P^{-1}Q(N/P)^{-1}RP^{-1}) & -(P^{-1}Q(N/P)^{-1}) \\ -((N/P)^{-1}RP^{-1}) & (N/P)^{-1} \end{bmatrix}$$

In our case $N = \begin{bmatrix} MA - I & MB \\ 0 & -I \end{bmatrix}$ and $(N/P) = -I$. From this it follows that

$$N^{-1} = (DE - I)^{-1} = \begin{bmatrix} (MA - I)^{-1} & (MA - I)^{-1} \cdot M \cdot B \\ 0 & -I \end{bmatrix}$$

■

Fact 4. *Let $A, B \in \mathbb{M}_{n\times n}$ with B being symmetric PSD. Suppose x is an eigenvector of AB corresponding to eigenvalue λ. Then $\sqrt{B}x$ is an eigenvector of $\sqrt{B}A\sqrt{B}$ corresponding to eigenvalue λ*

Fact 5 (Laplacian and graph connectivity (Fiedler value))**.** *Let G be a graph with n vertices. Suppose $(\lambda_1, \lambda_2 \cdots \lambda_n)$ be the eigenvalues corresponding to the eigenvectors $(v_1, v_2 \cdots v_n)$ of the Laplacian of G denoted as L_G. L_G is symmetric PSD with $\lambda_1 = 0$ and $v_1 = \mathbf{1}$. The following properties relate the eigenvalues of L_G with the connectivity of G :*

1. *$\lambda_2 > 0 \iff G$ is connected*
2. *G is disconnected $\iff \exists z$ with $z^T\mathbf{1} = 0$ and $z^T L_G z = 0$*

The above is true for L_G^+ also

Definition 1 (χ_u)**.** *χ_u is a vector of size $|V|$*

$$\chi_u(i) = \begin{cases} 1, & \text{if } i = u \\ 0, & \text{otherwise} \end{cases}$$

Definition 2 (Uniform random spanning tree)**.** *Let $\hat{T}$ be the random variable denoting a uniformly random spanning tree, then $\mathbb{P}(\hat{T} = T) = \frac{1}{|\mathcal{T}|}$, where $\mathcal{T}$ is the set of all spanning trees of G.*

Fact 6. *Given a graph $G = (V, E)$ with laplacian L_G, the effective resistance of an edge $e = \{u, v\} \in E$ is*

$$R_e^{\mathit{eff}} = (\chi_u - \chi_v)^T \; L_G^+ \; (\chi_u - \chi_v)$$

Then for any $e \in E$ we have

$$\mathbb{P}(e \in \hat{T}) = R_e^{\mathit{eff}}$$

4.1.5 Technical details of the results

Deletion

The first step in obtaining a updation formula for deletion is to make sure $(I - L_D L_G^+)$ is invertible. As the inverse of this term would be used in the expansion of $(L_G - L_D)^+$

For the first direction of Lemma 1, the main result used is **Fact 5** (G is disconnected $\iff \exists z$ with $z^T\mathbf{1} = 0$ and $z^T L_G^+ z = 0$). So if we can show this for a suitable z we are done. Now using the hypothesis that $(I - L_D L_G^+)$ is singular and **Fact 4** they derive the following $y^T \cdot \sqrt{L_G^+} \cdot L_{G\setminus D} \cdot \sqrt{L_G^+} \cdot y = 0$. As we an see the remaining part is to show $(z = \sqrt{L_G^+} y) \perp \mathbf{1}$

Lemma 1 (Formulas in **Theorem 1** are well defined)**.** *Let $G = (V, E)$ be a connected graph and $D \subseteq E$ then*
$\left(I - L_D L_G^+\right)$ *is invertible* $\iff$ $G \setminus D$ *is connected*

Proof. First let's show that If $(I - L_D L_G^+)$ is singular then $G \setminus D$ is disconnected

- Since $(I - L_D L_G^+)$ is singular $\exists x \neq 0$ s.t. $(I - L_D L_G^+)x = 0$

$$\implies L_D L_G^+ \, x = x \tag{4.1}$$
$$\implies 1 \in eigenvalues(L_D L_G^+) \tag{4.2}$$
$$\implies 1 \in eigenvalues((L_G - L_{G\setminus D})L_G^+) \tag{4.3}$$

- Let $x \perp \mathbf{1}$ be an eigenvector of $(L_G - L_{G\setminus D})L_G^+$ with eigenvalue 1.
- By **Fact 4**, $y = \frac{\sqrt{L_G^+}x}{\left\|\sqrt{L_G^+}x\right\|}$ is an eigenvector of $\sqrt{L_G^+}(L_G - L_{G\setminus D})\sqrt{L_G^+}$

$$= y^T \cdot \sqrt{L_G^+}(L_G - L_{G\setminus D})\sqrt{L_G^+} \cdot y = 1 \tag{4.4}$$
$$= y^T \sqrt{L_G^+} L_G \sqrt{L_G^+} y = (HOW) y^T L_G^+ L_G y = y^T P y \tag{4.5}$$
$$= y^T \left(I - \frac{\mathbf{1}^T\mathbf{1}}{n}\right) y = y^T y - \left(\frac{y^T \mathbf{1}^T \mathbf{1} y}{n}\right) = (HOW) y^T y = 1 \tag{4.6}$$

- $\therefore y^T \sqrt{L_G^+} L_{G\setminus D} \sqrt{L_G^+} y = 0$ now if we consider $z = \sqrt{L_G^+} y$ and show that $z^T\mathbf{1} = 0$ then we can use **Fact 5** to complete the proof

$$y^T \sqrt{L_G^+}\mathbf{1} = x^T \sqrt{L_G^+}\sqrt{L_G^+}\mathbf{1} = 0(\text{HOW is 1 in kernel of } L_G^+ \tag{4.7}$$
$$G \setminus D \text{ is disconnected} \tag{4.8}$$

Now to prove the converse, If $G \setminus D$ is disconnected then $I - L_D L_G^+$ is singular

- If $G \setminus D$ is disconnected then $\exists y \perp \mathbf{1}, ||y|| = 1$ we have
  1. $y^T \cdot \sqrt{L_G^+} \cdot L_{G\setminus D} \cdot \sqrt{L_G^+} \cdot y = 0(HOW)$
  2. $y^T \cdot \sqrt{L_G^+} \cdot L_G \cdot \sqrt{L_G^+} \cdot y = y^T y = 1$

- From (1) and (2) we get $y^T\sqrt{L_G^+}(L_G - L_{G\setminus D})\sqrt{L_G^+}y = 1$

$$
\begin{aligned}
&\implies y^T \cdot \sqrt{L_G^+} \cdot L_D \cdot \sqrt{L_G^+} \cdot y = 1 && (4.9)\\
&\implies 1 \in \text{eigenvalues}(L_D L_G^+)(HOW) && (4.10)\\
&\implies (I - L_D L_G^+) \text{ is singular} && (4.11)
\end{aligned}
$$

■

In **Theorem 1** they just show that the formula for the updated pseudoinverse is indeed true. This is shown using the following identity $LL^+ = P$

Theorem 1 (Update identity for Deletion)**.** *Let $G = (V, E)$ be a connected graph and $D \subseteq E$. If $G \setminus D$ is connected then*

$$(L_G - L_D)^+ = L_G^+ - \left(L_G^+ \cdot (L_D L_G^+ - I)^{-1} \cdot L_D \cdot L_G^+\right)$$

Proof. If R.H.S is indeed true then it should satisfy the property of $LL^+ = P$

$$(L_G - L_D) \cdot (L_G - L_D)^+$$

$$(L_G - L_D) \cdot \left(L_G^+ - \left(L_G^+ \cdot (L_D L_G^+ - I)^{-1} \cdot L_D \cdot L_G^+\right)\right)$$

$$\left[P - L_D L_G^+\right] - \left[(L_G L_G^+ - L_D L_G^+) \cdot (L_D L_G^+ - I)^{-1} \cdot L_D \cdot L_G^+\right]$$

$$\left[P - L_D L_G^+\right] + \left[\left((L_D L_G^+ - I) + \frac{\mathbf{1}\mathbf{1}^T}{n}\right) \cdot \left((L_D L_G^+ - I)^{-1} \cdot L_D \cdot L_G^+\right)\right]$$

$$\left[P - L_D L_G^+\right] + \left[(L_D L_G^+) + \left(\frac{\mathbf{1}\mathbf{1}^T}{n} \cdot (L_D L_G^+ - I)^{-1} \cdot L_D \cdot L_G^+\right)\right]$$

We can see that $-\mathbf{1}^T(L_D L_G^+ - I) = -\mathbf{1}^T L_D L_G^+ + (I\mathbf{1})^T = 0 + \mathbf{1}^T = \mathbf{1}^T$. Hence $\mathbf{1}^T(L_D L_G^+ - I)^{-1} = -\mathbf{1}^T$. And also $\mathbf{1}^T L_D = 0$. Hence,

$$P - L_D L_G^+ + L_D L_G^+ + \mathbf{1}^T L_D L_G^+ = P$$

■

Definition 3 (Submatrix)**.** *A submatrix of a martix A containing rows S and columns T is denoted as $A_{S,T}$*

Corollary 1 modifies the update formula in **Theorem 1** to work for submatrices and hence reduce the compliexity to $\mathcal{O}(|S|^\omega)$. They do this by first applying the facts related to submatrices **Fact 3.3, 3.5**.

Corollary 1 (Improved **Theorem 1** for submatrix)**.** *Let $G = (V, E)$ be a connected graph and $D \subseteq G$. For $S \subseteq V$ define $E[S]$ as $(S \times S) \cap E$. Suppose $E_D \subseteq E[S]$ and $G \setminus D$ is connected then*

$$(L_G - L_D)^+_{S,S} = (L_G^+)_{S,S} - \left((L_G^+)_{S,S} \cdot ((L_D)_{S,S}\ (L_G^+)_{S,S}\ -\ I)^{-1} \cdot (L_D)_{S,S} \cdot (L_G^+)_{S,S}\right)$$

Proof. From **Theorem 1** we know that $(L_G - L_D)^+ = L_G^+ - \left(L_G^+ \cdot (L_D L_G^+ - I)^{-1} \cdot L_D \cdot L_G^+\right)$. If we apply **Fact 3.1** to $(L_G - L_D)^+$ we get

$$(L_G^+)_{S,S} - \left(L_G^+ \cdot (L_D L_G^+ - I)^{-1} \cdot L_D \cdot L_G^+\right)_{S,S}$$

Applying **Fact 3.3** we get (HOW)

$$(L_G^+)_{S,S} - \left((L_G^+)_{S,S} \cdot (L_D L_G^+ - I)^{-1}_{S,S} \cdot (L_D)_{S,S} \cdot (L_G^+)_{S,S}\right)$$

Now applying **Fact 3.5** to $(L_D L_G^+ - I)^{-1}_{S,S}$

Fact 3.5 states that If $D = \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix}$, and $E = \begin{bmatrix} A & B \\ X & Y \end{bmatrix}$ where $M, A \in \mathbb{M}_{n\times n}$ and If $(MA - I)$ is invertible Then,

$$(DE - I)^{-1} = \begin{bmatrix} (MA - I)^{-1} & (MA - I)^{-1} \cdot M \cdot B \\ 0 & -I \end{bmatrix}$$

Here we have $L_D = \begin{bmatrix} (L_D)_{S,S} & 0 \\ 0 & 0 \end{bmatrix}$, (HOW can it always be this way) and $L_G = \begin{bmatrix} (L_G)_{S,S} & (L_G)_{S,S^c} \\ (L_G)_{S^c,S} & (L_G)_{S^c,S^c} \end{bmatrix}$

$$\therefore\ (L_D L_G - I)^{-1} = \begin{bmatrix} ((L_D)_{S,S}(L_G)_{S,S} - I)^{-1} & (L_D)_{S,S}(L_G)_{S,S^c} \\ 0 & -I \end{bmatrix}$$

Hence we get the required result

$$(L_G - L_D)^+_{S,S} = (L_G^+)_{S,S} - \left((L_G^+)_{S,S} \cdot ((L_D)_{S,S}\ (L_G^+)_{S,S}\ -\ I)^{-1} \cdot (L_D)_{S,S} \cdot (L_G^+)_{S,S}\right)$$

$\blacksquare$

Contraction

The main approach proposed to tackle contraction updates is to increase the weight of the edges that are to be contracted to a large value k.

Definition 4 (Incidence Matrix)**.** *Let $G = (V, E)$, given an edge $e = u, v \in E$ the incidence vector of e is defined as $v_e = (\chi_u - \chi_v)$. Given a set of edges $D = \{e_1, e_2 \cdots e_m\} \subseteq E$, the incidence matrix of D is defined as $B_D = [v_{e_1}|v_{e_2} \cdots |v_{e_m}]$*

Note - *I have used a different notation for the incidence matrix compared to the original paper I found it to be a bit confusing. And B is the common notation for incidence matrices in other resources.*

Definition 5 ($G+ke$)**.** *$G+ke$ is the weighted graph obtained by increasing e's weight by k*

Lemma 2 (Formulas in **Theorem 2** are well defined)**.** *Let $G = (V, E)$ be a connected graph. Given $F \subseteq E$ with $|F| = r$ and let B_F be the incidence matrix of F.*

$$B_F^T\ L_G^+\ B_F \text{ is invertible} \iff F \text{ is a forest}$$

Proof. First they show
F is a forest $\implies B_F^T\ L_G^+\ B_F$ is invertible

So the main idea of this proof is to show that $B_F^T\ L_G^+\ B_F$ is positive definite. This is enough because positive definite matrices are non singular (if not then then they will have 0 as an eigenvalue). Now using the following claim

Claim 1. *The incidence matrix of a acyclic graph has full column rank*

Hence for any $x \in \mathbb{R}^r, x \neq 0$, let $y = B_F x$ and $y \neq 0$. Also $y^T \mathbf{1} = x^T B_F^T \mathbf{1} = 0$. Hence $y \perp ker(L_G^+)$. Now since G is connected we have $\lambda_2(L_G) > 0$. Now since y corresponds to all vectors perpendicular to $ker(L_G^+)$ we can say that $y^T L_G^+ y > 0$ now expanding $y = B_F x$ we get $x^T(B_F^T\ L_G^+\ B_F)x > 0$. Hence $B_F^T\ L_G^+\ B_F$ is positive definite and hence invertible.

Now for the converse (TODO) ■

Lemma 3 (Formulas in **Theorem 2** are well defined)**.** *Let $G = (V, E)$ be a connected graph. Given $F \subseteq E$ and let B_F be the incidence matrix of F. For any $k > 0$,*

$$\textit{If } F \textit{ is a forest then } \left(\frac{I}{k} + B_F^T\ L_G^+\ B_F\right) \textit{ is invertible for any } k > 0$$

Proof. Suppose A, B are positive definite matrices then $A+B$ is also positive definite. Since A, B are positive definite we have $x^T Ax > 0, x^T Bx > 0$ for any x. Combining these two identity we get $x^T(A + B)x > 0$. Hence $A + B$ is also positive definite.

By **Lemma 2** $B_F^T\ L_G^+\ B_F$ is positive definite. And I/k is also positive definite because all the eigenvalues are $1/k$ and we have $k > 0$. Since positive definite matrices are non-singular, $\left(\frac{I}{k} + B_F^T\ L_G^+\ B_F\right)$ is invertible ■

Theorem 2 uses **Lemma 2** to show that the contraction update formula for finite k is well defined.

Theorem 2 (Contraction update formula for finite k)**.** *Let $G = (V, E)$ be a connected graph. Given $F \subseteq E$ and let B_F be the incidence matrix of F. For any $k > 0$,*

$$(L_G + k\ L_F)^+ = L_G^+ - \left(L_G^+ \cdot B_F \cdot (\frac{I}{k} + B_F^T\ L_G^+\ B_F)^{-1} \cdot B_F^T \cdot L_G^+\right)$$

Proof. They use the same strategy used in **Theorem 1**. Also note that $B_F B_F^T = L_F$

$$\left[L_G + kB_F B_F^T\right] \cdot \left[L_G^+ - \left(L_G^+\ B_F\ (\frac{I}{k} + B_F^T\ L_G^+\ B_F)^{-1}\ B_F^T\ L_G^+\right)\right]$$

$$= P + kB_F B_F^T L_G^+ - \left((L_G L_G^+ B_F + kB_F B_F^T L_G^+ B_F)\ (\frac{I}{k} + B_F^T L_G^+ B_F)^{-1}\ B_F^T\ L_G^+\right)$$

Here $L_G L_G^+ B_F = (I - \frac{\mathbf{1}\mathbf{1}^T}{n})B_F = B_F$. Because each column sum of B_F is 0.

$$= P + kB_F B_F^T L_G^+ - \left(kB_F\ (\frac{I}{k} + B_F^T L_G^+ B_F)\ (\frac{I}{k} + B_F^T L_G^+ B_F)^{-1}\ B_F^T\ L_G^+\right)$$

$$= P + kB_F B_F^T L_G^+ - kB_F B_F^T L_G^+ = P$$

■

Corollary 2 (Improves **Theorem 2** for sub-matrices)**.** *Let $G = (V, E)$ be a connected graph. Given $F \subseteq E$ and let B_F be the incidence matrix of F. Suppose $F \subseteq E[S]$, where $S \subseteq V$. For any $k > 0$,*

$$(L_G+k\ L_F)^+_{S,S} = (L_G^+)_{S,S} - \left((L_G^+)_{S,S}\ (B_F)_{S,*}\ (\frac{I}{k} + (B_F^T)_{S,*}\ (L_G^+)_{S,S}\ (B_F)_{S,*})^{-1}\ (B_F^T)_{S,*}\ (L_G^+)_{S,S}\right)$$

Proof. TODO ∎

In **Theorem 3** they finally show that the

Theorem 3 (Extends **Theorem 2** to $k \to \infty$ case)**.** *For a forest $F_1 \subseteq E$, let $G(k) = G + k\ F_1$ as defined in **Definition 3** . Let $F_2 \subseteq E$ be disjoint from F_1 such that $F_1 \cup F_2$ is a forest. Let B_{F_2} be the incidence matrix of F_2. For $k > 0$ define $N = \lim_{k\to\infty} L^+_{G(k)}$*

$$\lim_{k\to\infty} L^+_{G(k)+kF_2} = N - \left(N \cdot B_{F_2} \cdot (B^T_{F_2}\ N\ B_{F_2}) \cdot B^T_{F_2} \cdot N\right)$$

Also $ker\left(\lim_{k\to\infty} L^+_{G(k)+kF_2}\right) = span\ (B_{F_1\cup F_2} \cup \boldsymbol{1})$

Proof. TODO ∎

5 Conclusion

Bibliography

Aldous, David J (1990). "The random walk construction of uniform spanning trees and uniform labelled trees". In: *SIAM Journal on Discrete Mathematics* 3.4, pp. 450–465.

Asadpour, Arash et al. (2017). "An O(log n/log log n)-Approximation Algorithm for the Asymmetric Traveling Salesman Problem". In: *Operations Research* 65.4, pp. 1043–1061. DOI: 10.1287/opre.2017.1603. eprint: https://doi.org/10.1287/opre.2017.1603. URL: https://doi.org/10.1287/opre.2017.1603.

Bar-Ilan, Judit and Dror Zernik (1989). "Random leaders and random spanning trees". In: *Distributed Algorithms*. Ed. by Jean-Claude Bermond and Michel Raynal. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1–12. ISBN: 978-3-540-46750-2.

Broder, A. (1989). "Generating random spanning trees". In: *30th Annual Symposium on Foundations of Computer Science*, pp. 442–447.

Colbourn, Charles J. (1987). *The Combinatorics of Network Reliability*. USA: Oxford University Press, Inc. ISBN: 0195049209.

Colbourn, Charles J, Robert P.J Day, and Louis D Nel (1989). "Unranking and ranking spanning trees of a graph". In: *Journal of Algorithms* 10.2, pp. 271 –286. ISSN: 0196-6774. DOI: https://doi.org/10.1016/0196-6774(89)90016-3. URL: http://www.sciencedirect.com/science/article/pii/0196677489900163.

Colbourn, Charles J, Bradley M Debroni, and Wendy J Myrvold (1988). "Estimating the coefficients of the reliability polynomial". In: *Congressus Numerantium* 62, pp. 217–223.

Colbourn, Charles J., Wendy J. Myrvold, and Eugene Neufeld (1996). "Two Algorithms for Unranking Arborescences". In: *Journal of Algorithms* 20.2, pp. 268 – 281. ISSN: 0196-6774. DOI: https://doi.org/10.1006/jagm.1996.0014. URL: http://www.sciencedirect.com/science/article/pii/S0196677496900140.

Durfee, David et al. (2017). "Sampling Random Spanning Trees Faster than Matrix Multiplication". In: *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2017. Montreal, Canada: Association for Computing Machinery, 730–742. ISBN: 9781450345286. DOI: 10.1145/3055399.3055499. URL: https://doi.org/10.1145/3055399.3055499.

Frieze, Alan et al. (2014). "Expanders via Random Spanning Trees". In: *SIAM Journal on Computing* 43.2, pp. 497–513. DOI: 10.1137/120890971. eprint: https://doi.org/10.1137/120890971. URL: https://doi.org/10.1137/120890971.

Fung, Wai Shing and Nicholas J. A. Harvey (2010). "Graph Sparsification by Edge-Connectivity and Random Spanning Trees". In: *CoRR* abs/1005.0265. arXiv: 1005.0265. URL: http://arxiv.org/abs/1005.0265.

Gharan, S. O., A. Saberi, and M. Singh (2011). "A Randomized Rounding Approach to the Traveling Salesman Problem". In: *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pp. 550–559.

Goyal, Navin, Luis Rademacher, and Santosh Vempala (2009). "Expanders via Random Spanning Trees". In: *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '09. New York, New York: Society for Industrial and Applied Mathematics, 576–585.

Guénoche, A (1983). "Random spanning tree". In: *Journal of Algorithms* 4.3, pp. 214 –220. ISSN: 0196-6774. DOI: https://doi.org/10.1016/0196-6774(83)90022-6. URL: http://www.sciencedirect.com/science/article/pii/0196677483900226.

Harvey, Nicholas JA and Keyulu Xu (2016). "Generating random spanning trees via fast matrix multiplication". In: *LATIN 2016: Theoretical Informatics*. Springer, pp. 522–535.

Kandel, D. et al. (1996). "Shuffling biological sequences". In: *Discrete Applied Mathematics* 71.1, pp. 171 –185. ISSN: 0166-218X. DOI: https://doi.org/10.1016/S0166-218X(97)81456-4. URL: http://www.sciencedirect.com/science/article/pii/S0166218X97814564.

Kelner, Jonathan A. and Aleksander Madry (2009). "Faster Generation of Random Spanning Trees". In: *Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science*. FOCS '09. USA: IEEE Computer Society, 13–21. ISBN: 9780769538501.

Kirchhoff, G. (1847). "Ueber die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Vertheilung galvanischer Ströme geführt wird". In: *Annalen der Physik* 148.12, pp. 497–508. DOI: 10.1002/andp.18471481202. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/andp.18471481202. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/andp.18471481202.

Kulkarni, V.G (1990). "Generating random combinatorial objects". In: *Journal of Algorithms* 11.2, pp. 185 –207. ISSN: 0196-6774. DOI: https://doi.org/10.1016/0196-6774(90)90002-V. URL: http://www.sciencedirect.com/science/article/pii/019667749090002V.

Nel, Louis D. and Charles J. Colbourn (1990). "Combining monte carlo estimates and bounds for network reliability". In: *Networks* 20.3, pp. 277–298. DOI: 10.1002/net.3230200303. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/net.3230200303. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/net.3230200303.

Propp, James Gary and David Bruce Wilson (1998). "How to Get a Perfectly Random Sample from a Generic Markov Chain and Generate a Random Spanning Tree of a Directed Graph". In: *J. Algorithms* 27, pp. 170–217.

Schild, Aaron (2018). "An Almost-Linear Time Algorithm for Uniform Random Spanning Tree Generation". In: *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2018. Los Angeles, CA, USA: Association for Computing Machinery, 214–227. ISBN: 9781450355599. DOI: 10.1145/3188745.3188852. URL: https://doi.org/10.1145/3188745.3188852.

Wikipedia contributors (2020). *Schur complement — Wikipedia, The Free Encyclopedia*. [Online; accessed 28-May-2020]. URL: https://en.wikipedia.org/w/index.php?title=Schur_complement&oldid=947233275.

Wilson, David Bruce (1996). "Generating Random Spanning Trees More Quickly than the Cover Time". In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. STOC '96. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 296–303. ISBN: 0897917855. DOI: 10.1145/237814.237880. URL: https://doi.org/10.1145/237814.237880.