Near-linear time samplers for matroid independent sets with applications

Xiaoyu Chen

based on joint work with

Heng Guo

Xinyuan Zhang

Zongrui Zou

Motivation: network reliability

(ALL-TERMINAL) RELIABILITY

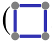
Given an graph (network) $G = (V, E)$, define a random subgraph $G(p)$ by removing each edge independently with probability p. How to estimate

$$Z_{\mathrm{rel}}(G, p) = \mathbf{Pr}\left[G(p) \text{ is connected}\right]?$$

In particular, $Z_{\mathrm{rel}}(G, p)$ can be given explicitly as

$$Z_{\mathrm{rel}}(G, p) = \sum_{R \subseteq E : (V,R) \text{ is connected}} p^{|E \setminus R|}(1-p)^{|R|}$$

For example:

$Z_{\mathrm{rel}}($  $, p) =$  $+$  $+$  $+$  $+$  $= (1-p)^4 + 4p(1-p)^3$

Motivation: network reliability

(ALL-TERMINAL) RELIABILITY

Given an graph (network) $G = (V, E)$, define a random subgraph $G(p)$ by removing each edge independently with probability p. How to estimate

$$Z_{\mathrm{rel}}(G, p) = \mathbf{Pr}\left[G(p) \text{ is connected}\right]?$$

Computing Z_{rel} exactly is #P-complete [Jerrum (1981), Provan and Ball (1983)]

According to standard reduction from approximate counting to sampling, it is sufficient to have a sampler for random connected spanning subgraph:

$$\forall S \subseteq E, \quad \mu(S) \propto \mathbf{1}[(V, E \setminus S) \text{ is connected}] \left(\frac{p}{1-p}\right)^{|S|}$$

- overhead is refined to $T_{\mathrm{counting}} = T_{\mathrm{sampling}} \times O(n)$ by Guo and He (2020)

- Output $(1 \pm \varepsilon) \cdot (1 - Z_{\mathrm{rel}})$ UNRELIABILITY

counting	
$\widetilde{O}(n^3)$	Karger (1995)
$\widetilde{O}(n^2)$	Karger (2020)
$m^{1+o(1)} + \widetilde{O}(n^{1.5})$	Cen, He, Li and Panigrahi (2023)

- Output $(1 \pm \varepsilon) \cdot Z_{\mathrm{rel}}$ RELIABILITY

counting	sampling	
$O(m^2 n^3)$	$O(m^2 n)$	Guo and Jerrum (2018)
$O(mn^2)$	$O(mn)$	Guo and He (2020)
$\widetilde{O}(mn^2)$	$\widetilde{O}(mn)$	Anari, Liu, Oveis Gharan, Vinzant and Vuong (2021)
$\widetilde{O}(mn)$	$\widetilde{O}(m)$	our result

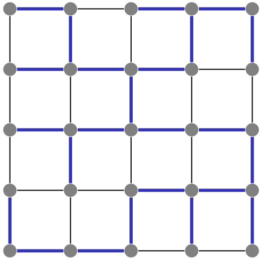
implicitly

- UNRELIABILITY $\neq$ RELIABILITY since Z_{rel} can be exponentially small

$\mathcal{M} = (\mathcal{U}, \mathcal{I})$ is a **matroid** if $\mathcal{I} \subseteq 2^{\mathcal{U}}$ satisfies

1. $\varnothing \in \mathcal{I}$;
2. if $S \in \mathcal{I}, T \subseteq S$, then $T \in \mathcal{I}$;
3. if $S, T \in \mathcal{I}$ and $|S| > |T|$, then $\exists\, e \in S \setminus T$ such that $T \cup \{e\} \in \mathcal{I}$

- a set $S \in \mathcal{I}$ is called an **independent set**
- maximal independent sets are called **bases**
  we use $\mathcal{B} \subseteq \mathcal{I}$ to denote the set of all bases

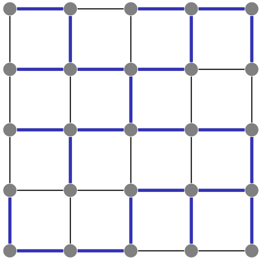


let $\mathcal{I} = \{S \subseteq E \mid \text{graph } (V, S) \text{ has no cycle}\}$
then $(E, \mathcal{I})$ is a (graphical) matroid

$\mathcal{M} = (\mathcal{U}, \mathcal{I})$ is a **matroid** if $\mathcal{I} \subseteq 2^{\mathcal{U}}$ satisfies

1. $\varnothing \in \mathcal{I}$;
2. if $S \in \mathcal{I}, T \subseteq S$, then $T \in \mathcal{I}$;
3. if $S, T \in \mathcal{I}$ and $|S| > |T|$, then $\exists\, e \in S \setminus T$ such that $T \cup \{e\} \in \mathcal{I}$

- a set $S \in \mathcal{I}$ is called an **independent set**
- maximal independent sets are called **bases**
  we use $\mathcal{B} \subseteq \mathcal{I}$ to denote the set of all bases



let $\mathcal{I} = \{S \subseteq E \mid \text{graph } (V, S) \text{ has no cycle}\}$
then $(E, \mathcal{I})$ is a (graphical) matroid

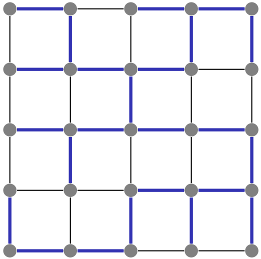
rank $\quad \mathrm{rk}(\cdot) \Longleftrightarrow \mathcal{I}$

For $S \subseteq \mathcal{U}$, $\mathrm{rk}(S) := \max_{A \subseteq S, A \in \mathcal{I}} |A|$

$\mathcal{M} = (\mathcal{U}, \mathcal{I})$ is a **matroid** if $\mathcal{I} \subseteq 2^{\mathcal{U}}$ satisfies

1. $\emptyset \in \mathcal{I}$;
2. if $S \in \mathcal{I}, T \subseteq S$, then $T \in \mathcal{I}$;
3. if $S, T \in \mathcal{I}$ and $|S| > |T|$, then $\exists\, e \in S \setminus T$ such that $T \cup \{e\} \in \mathcal{I}$

- a set $S \in \mathcal{I}$ is called an **independent set**
- maximal independent sets are called **bases**
  we use $\mathcal{B} \subseteq \mathcal{I}$ to denote the set of all bases



let $\mathcal{I} = \{S \subseteq E \mid \text{graph } (V, S) \text{ has no cycle}\}$
then $(E, \mathcal{I})$ is a (graphical) matroid

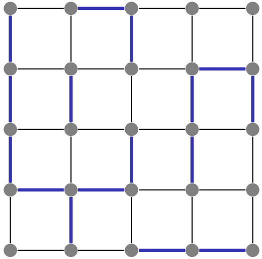
rank $\quad \mathrm{rk}(\cdot) \Longleftrightarrow \mathcal{I}$

For $S \subseteq \mathcal{U}$, $\mathrm{rk}(S) := \max_{A \subseteq S, A \in \mathcal{I}} |A|$

dual matroid $\quad \mathcal{M}^\star = (\mathcal{U}, \mathcal{I}^\star)$

$$\mathcal{I}^\star := \{S \subseteq \mathcal{U} \mid \exists B \subseteq \mathcal{U} \setminus S, B \in \mathcal{B}\}$$

B is a base of $\mathcal{M} \Longleftrightarrow \mathcal{U} \setminus B$ is a base of $\mathcal{M}^\star$



Random cluster model

Recall that the rank function is defined as $\mathrm{rk}(S) := \max_{A \subseteq S, A \in \mathcal{I}} |A|$. We focus on fast samplers for the following distribution where $q \geqslant 0$ and $\boldsymbol{\lambda} \in \mathbb{R}_{>0}^{U}$

$$\forall S \subseteq U, \quad \mu_{\mathcal{M},q,\boldsymbol{\lambda}}(S) \propto q^{|S|-\mathrm{rk}(S)} \cdot \prod_{e \in S} \lambda_e$$

Random cluster model

Recall that the rank function is defined as $\mathrm{rk}(S) := \max_{A \subseteq S, A \in \mathcal{I}} |A|$. We focus on fast samplers for the following distribution where $\mathbf{q} \geqslant 0$ and $\boldsymbol{\lambda} \in \mathbb{R}_{>0}^{U}$

$$\forall S \subseteq U, \quad \mu_{\mathcal{M},\mathbf{q},\boldsymbol{\lambda}}(S) \propto \mathbf{q}^{|S|-\mathrm{rk}(S)} \cdot \prod_{e \in S} \lambda_e$$

- when $\mathbf{q} = 0$, $\mu_{\mathcal{M},\mathbf{q},\boldsymbol{\lambda}}$ becomes a dist. on $\mathcal{I}$: $\mu_{\mathcal{M},0,\boldsymbol{\lambda}}(S) \propto \mathbf{1}[S \in \mathcal{I}] \cdot \prod_{e \in S} \lambda_e$

Random cluster model

Recall that the rank function is defined as $\mathrm{rk}(S) := \max_{A \subseteq S, A \in \mathcal{I}} |A|$. We focus on fast samplers for the following distribution where $q \geqslant 0$ and $\boldsymbol{\lambda} \in \mathbb{R}_{>0}^{U}$

$$\forall S \subseteq U, \quad \mu_{\mathcal{M},q,\boldsymbol{\lambda}}(S) \propto q^{|S|-\mathrm{rk}(S)} \cdot \prod_{e \in S} \lambda_e$$

- when $q = 0$, $\mu_{\mathcal{M},q,\boldsymbol{\lambda}}$ becomes a dist. on $\mathcal{I}$: $\mu_{\mathcal{M},0,\boldsymbol{\lambda}}(S) \propto \mathbf{1}[S \in \mathcal{I}] \cdot \prod_{e \in S} \lambda_e$
- when $q = 1$, $\mu_{\mathcal{M},q,\boldsymbol{\lambda}}$ becomes a product dist. : $\mu_{\mathcal{M},1,\boldsymbol{\lambda}}(S) \propto \prod_{e \in S} \lambda_e$

Random cluster model

Recall that the rank function is defined as $\mathrm{rk}(S) := \max_{A \subseteq S, A \in \mathcal{I}} |A|$. We focus on fast samplers for the following distribution where $q \geqslant 0$ and $\boldsymbol{\lambda} \in \mathbb{R}_{>0}^{U}$

$$\forall S \subseteq U, \quad \mu_{\mathcal{M},q,\boldsymbol{\lambda}}(S) \propto q^{|S|-\mathrm{rk}(S)} \cdot \prod_{e \in S} \lambda_e$$

- when $q = 0$, $\mu_{\mathcal{M},q,\boldsymbol{\lambda}}$ becomes a dist. on $\mathcal{I}$: $\mu_{\mathcal{M},0,\boldsymbol{\lambda}}(S) \propto \mathbf{1}[S \in \mathcal{I}] \cdot \prod_{e \in S} \lambda_e$
- when $q = 1$, $\mu_{\mathcal{M},q,\boldsymbol{\lambda}}$ becomes a product dist. : $\mu_{\mathcal{M},1,\boldsymbol{\lambda}}(S) \propto \prod_{e \in S} \lambda_e$
- when $q = 0$, $\mathcal{M}^\star = (E, \mathcal{I}^\star)$ be the **dual** graphical matroid, and $\lambda_e = \frac{p}{1-p}$, then $\mu_{\mathcal{M}^\star,q,\boldsymbol{\lambda}}$ becomes the dist. of random connected spanning subgraph:

$$\mu_{\mathcal{M}^\star,q,\boldsymbol{\lambda}}(S) \propto \underbrace{\mathbf{1}[S \in \mathcal{I}^\star]}_{\substack{E \setminus S \text{ contains} \\ \text{a spanning tree}}} \left(\frac{p}{1-p}\right)^{|S|} = \mathbf{1}[(V, E \setminus S) \text{ is connected}] \left(\frac{p}{1-p}\right)^{|S|}$$

Our result

For matroid $\mathcal{M} = (\mathcal{U}, \mathcal{I})$, it is meaningless to take $\mathcal{I}$ as input of the algorithm 🤔
We need some implicit representation/oracle of the matroid

Since $\mathrm{rk}(\cdot) \iff \mathcal{I}$, the rank function $\mathrm{rk}(\cdot)$ is a good choice. We relax it to

rank oracle/data structure $\mathcal{O}_r$

$\mathcal{O}_r$ is a data structure that mantains a set $S \subseteq \mathcal{U}$ that supports:

- to **insert** an element to S
- to **delete** an element from S
- and to **query** $\mathrm{rk}(S)$

Suppose the amortized costs of all these operations are bounded by $t_{\mathcal{O}_r}$

Our result

For matroid $\mathcal{M} = (\mathcal{U}, \mathcal{I})$, it is meaningless to take $\mathcal{I}$ as input of the algorithm 🤔
We need some implicit representation/oracle of the matroid

Since $\mathrm{rk}(\cdot) \iff \mathcal{I}$, the rank function $\mathrm{rk}(\cdot)$ is a good choice. We relax it to

rank oracle/data structure $\mathcal{O}_r$

$\mathcal{O}_r$ is a data structure that mantains a set $S \subseteq \mathcal{U}$ that supports:

- to **insert** an element to S
- to **delete** an element from S
- and to **query** $\mathrm{rk}(S)$

Suppose the amortized costs of all these operations are bounded by $t_{\mathcal{O}_r}$

Theorem

Given $\mathcal{O}_r$ for the matroid $\mathcal{M} = (\mathcal{U}, \mathcal{I})$, $|\mathcal{U}| = n$, parameters $0 \leqslant q \leqslant 1$ and $\boldsymbol{\lambda} \in \mathbb{R}^{\mathcal{U}}_{>0}$, there is an approximate sampler for $\mu_{\mathcal{M},q,\boldsymbol{\lambda}}$ in time

$$O((1 + \lambda_{\max}) n \log n (\log n + t_{\mathcal{O}_r}))$$ in expectation

Our result

For matroid $\mathcal{M} = (\mathcal{U}, \mathcal{I})$, it is meaningless to take $\mathcal{I}$ as input of the algorithm 🤔
We need some implicit representation/oracle of the matroid

Since $\mathrm{rk}(\cdot) \iff \mathcal{I}$, the rank function $\mathrm{rk}(\cdot)$ is a good choice. We relax it to

rank oracle/data structure $\mathcal{O}_r$

$\mathcal{O}_r$ is a data structure that mantains a set $S \subseteq \mathcal{U}$ that supports:

- to **insert** an element to S
- to **delete** an element from S
- and to **query** $\mathrm{rk}(S)$

Suppose the amortized costs of all these operations are bounded by $t_{\mathcal{O}_r}$

Let $\kappa(S)$ be the number of connected components in graph (V, S)

$$\left.\begin{array}{l} \text{for graphical matroid: } \mathrm{rk}(S) = |V| - \kappa(S) \\ \text{for dual graphical matroid: } \mathrm{rk}^\star(S) = |S| - \mathrm{rk}(E) + \mathrm{rk}(E \setminus S) \end{array}\right\} \text{only need } \kappa(\cdot)$$

The quantity $\kappa(S)$ can be maintained by dynamic graph connectivity data structure in $O(\log^2 n)$ amortized time [Wulff-Nilsen (2013)]

Polarized distribution

Given a matroid $\mathcal{M} = (X, \mathcal{I})$, we focus on the following dist.

$$\forall S \subseteq X, \quad \mu_{\mathcal{M},q,\boldsymbol{\lambda}}(S) \propto q^{|S|-\mathrm{rk}(S)} \prod_{e \in S} \lambda_e$$

Suppose $X = \{x_1, \cdots, x_n\}$, we take $Y = \{y_1, \cdots, y_n\}$ be auxiliary variables

Polarized distribution

Let $S \sim \mu_{\mathcal{M},q,\boldsymbol{\lambda}}$ and then $T \overset{\text{uni.}}{\sim} \binom{Y}{n-|S|}$; Let $\pi_{\mathcal{M},q,\boldsymbol{\lambda}}$ be the dist. of $S \cup T$

In particular, for $Z \subseteq X \cup Y$, we denote $Z_X = Z \cap X$, $Z_Y = Z \cap Y$ and

$$\forall Z \in \binom{X \cup Y}{n}, \quad \pi_{\mathcal{M},q,\boldsymbol{\lambda}}(Z) \propto q^{|Z_X|-\mathrm{rk}(Z_X)} \frac{\prod_{e \in Z_X} \lambda_e}{\binom{n}{n-|Z_X|}}$$

- If $Z \sim \pi_{\mathcal{M},q,\boldsymbol{\lambda}}$, then $Z_X \sim \mu_{\mathcal{M},q,\boldsymbol{\lambda}}$

Down-up walk

The dist. $\pi_{\mathcal{M},q,\lambda}$ is homogeneous and we can define down-up walk on it

1. select a subset $T \subseteq Z$ of size $n-1$ uniformly at random
2. update Z to Z' by selecting random $Z' \supseteq T$ according to the following law:

$$\mathbf{Pr}\left[Z'\right] \propto \pi_{\mathcal{M},q,\lambda}(Z')$$

By tools from log-concave polynomials and high-dimensional expander, one can show that the mixing time of the down-up walk is bounded by $O(n \log n)$

$\pi_{\mathcal{M},q,\lambda}$ **is log-concave** $(0 \leqslant q \leqslant 1)$
Brändén and Huh (2018,2020)
Anari, Oveis Gharan and Vinzant (2018)

$+$

down-up walk on LC dist. mixes fast
Cryan, Guo and Mousa (2021)

Down-up walk

The dist. $\pi_{\mathcal{M},\mathbf{q},\boldsymbol{\lambda}}$ is homogeneous and we can define down-up walk on it

1. select a subset $T \subseteq Z$ of size $n-1$ uniformly at random
2. update Z to Z' by selecting random $Z' \supseteq T$ according to the following law:

$$\mathbf{Pr}\left[Z'\right] \propto \pi_{\mathcal{M},\mathbf{q},\boldsymbol{\lambda}}(Z')$$

By tools from log-concave polynomials and high-dimensional expander, one can show that the mixing time of the down-up walk is bounded by $O(n \log n)$

What prevents us from getting a near-linear time sampler for $\mu_{\mathcal{M},\mathbf{q},\boldsymbol{\lambda}}$? 🤔

Down-up walk

The dist. $\pi_{\mathcal{M},\mathbf{q},\boldsymbol{\lambda}}$ is homogeneous and we can define down-up walk on it

1. select a subset $T \subseteq Z$ of size $n-1$ uniformly at random
2. update Z to Z' by selecting random $Z' \supseteq T$ according to the following law:

$$\mathbf{Pr}\left[Z'\right] \propto \pi_{\mathcal{M},\mathbf{q},\boldsymbol{\lambda}}(Z')$$

By tools from log-concave polynomials and high-dimensional expander, one can show that the mixing time of the down-up walk is bounded by $O(n \log n)$

What prevents us from getting a near-linear time sampler for $\mu_{\mathcal{M},\mathbf{q},\boldsymbol{\lambda}}$? 🤔

- a single step of the down-up walk is not easy to implement (naïve approach needs $O(n)$ time to enumerate all Z' and calculate weight)

Down-up walk

The dist. $\pi_{\mathcal{M},q,\lambda}$ is homogeneous and we can define down-up walk on it

1. select a subset $T \subseteq Z$ of size $n-1$ uniformly at random
2. update Z to Z' by selecting random $Z' \supseteq T$ according to the following law:

$$\mathbf{Pr}\left[Z'\right] \propto \pi_{\mathcal{M},q,\lambda}(Z')$$

By tools from log-concave polynomials and high-dimensional expander, one can show that the mixing time of the down-up walk is bounded by $O(n \log n)$

What prevents us from getting a near-linear time sampler for $\mu_{\mathcal{M},q,\lambda}$? 🤔

- a single step of the down-up walk is not easy to implement
  (naïve approach needs $O(n)$ time to enumerate all Z' and calculate weight)
- some workarounds can be found in previous works:
  1. strengthen the oracle [Anari, Liu, Oveis Gharan, Vinzant and Vuong (2021)]
     works for some special matroid, but does not work for network reliability
  2. compare the down-up walk to the Glauber dynamics [Mousa (2022)]
     the comparison involves an $O(n)$ overhead

Down-up walk

The dist. $\pi_{\mathcal{M},\mathbf{q},\boldsymbol{\lambda}}$ is homogeneous and we can define down-up walk on it

1. select a subset $T \subseteq Z$ of size $n-1$ uniformly at random
2. update Z to Z' by selecting random $Z' \supseteq T$ according to the following law:

$$\mathbf{Pr}\left[Z'\right] \propto \pi_{\mathcal{M},\mathbf{q},\boldsymbol{\lambda}}(Z')$$

By tools from log-concave polynomials and high-dimensional expander, one can show that the mixing time of the down-up walk is bounded by $O(n \log n)$

We give a direct implementation for one-step of the down-up walk in time

$$(1+\lambda_{\max})(\log n + t_{\mathcal{O}_r}) \text{ in expectation}$$

The down-walk (1.) can be implemented in time $O(\log n)$; what about (2.)?
Suppose $Z' = T \cup \{e\}$, then

$$\mathbf{Pr}\left[Z'\right] \propto \pi_{\mathcal{M},\mathbf{q},\boldsymbol{\lambda}}(Z') \propto \begin{cases} 1/\binom{n}{|T_X|}, & e \in Y \setminus T \\ q^{\mathbf{1}[\mathrm{rk}(T)=\mathrm{rk}(Z')]}\lambda_i/\binom{n}{|T_X|+1}, & e = x_i \in X \setminus T \end{cases}$$

Down-up walk

The dist. $\pi_{\mathcal{M},\mathbf{q},\boldsymbol{\lambda}}$ is homogeneous and we can define down-up walk on it

1. select a subset $T \subseteq Z$ of size $n-1$ uniformly at random
2. update Z to Z' by selecting random $Z' \supseteq T$ according to the following law:

$$\mathbf{Pr}\left[Z'\right] \propto \pi_{\mathcal{M},\mathbf{q},\boldsymbol{\lambda}}(Z')$$

By tools from log-concave polynomials and high-dimensional expander, one can show that the mixing time of the down-up walk is bounded by $O(n \log n)$

We give a direct implementation for one-step of the down-up walk in time

$$(1+\lambda_{\max})(\log n + t_{\mathcal{O}_r}) \text{ in expectation}$$

The down-walk (1.) can be implemented in time $O(\log n)$; what about (2.)?
Suppose $Z' = T \cup \{e\}$, then

$$\mathbf{Pr}\left[Z'\right] \propto \pi_{\mathcal{M},\mathbf{q},\boldsymbol{\lambda}}(Z') \propto \begin{cases} \frac{n-|T_X|}{1+|T_X|}, & e \in Y \setminus T \\ q^{\mathbf{1}[\mathrm{rk}(T)=\mathrm{rk}(Z')]}\lambda_i, & e = x_i \in X \setminus T \end{cases}$$

A rejection sampling approach

Suppose $Z' = T \cup \{e\}$, then

$$\mathbf{Pr}[Z'] \propto \begin{cases} \frac{n-|T_X|}{1+|T_X|}, & e \in Y \setminus T \\ q^{\mathbb{1}[\mathrm{rk}(T)=\mathrm{rk}(Z')]}\lambda_i, & e = x_i \in X \setminus T \end{cases}$$

$$\nu(e) \propto \begin{cases} \frac{n-|T_X|}{1+|T_X|}, & e \in Y \setminus T \\ \lambda_i, & e = x_i \in X \setminus T \end{cases}$$

In order to sample Z', we can first sample $e \sim \nu$ where then if $e \in X \setminus T$ and $\mathrm{rk}(T) = \mathrm{rk}(Z') = \mathrm{rk}(T \cup \{e\})$, we **reject** with prob. q; keep doing this until success

Let $\mathcal{E}$ be the event that the rejection happens, then

$$\mathbf{Pr}[\mathcal{E}] \leqslant \mathbf{Pr}_{e\sim\nu}[e \in X \setminus T] = \sum_{x_i \in X\setminus T} \frac{\lambda_i}{\sum_{x_i \in X\setminus T} \lambda_i + \sum_{y \in Y \setminus T} \frac{n-|T_X|}{1+|T_X|}}$$

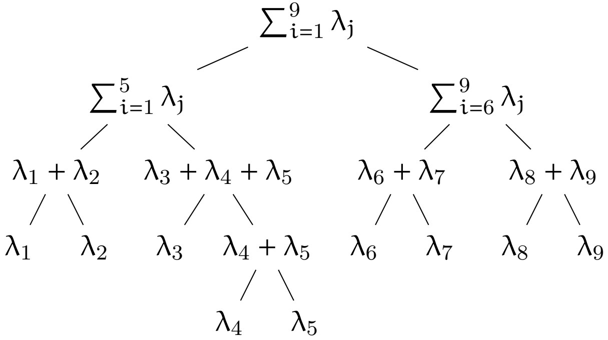
$$= \frac{\sum_{x_i \in X\setminus T} \lambda_i}{\sum_{x \in X \setminus T}(1+\lambda_i)} \leqslant \frac{\lambda_{\max}}{1+\lambda_{\max}}$$

- The rejection sampling will success in $(1+\lambda_{\max})$ rounds in expectation

Sample from $\nu(e)$ via self-balanced binary search trees (BST)

$$\nu(e) \propto \begin{cases} \frac{n-|T_X|}{1+|T_X|}, & e \in Y \setminus T \\ \lambda_i, & e = x_i \in X \setminus T \end{cases}$$

Suppose $X \setminus T = \{x_1, x_2, \cdots, x_9\}$, we can build a BST as follows



- The elements in $Y \setminus T$ can be handled similarly by another BST
- update and query on BST only cost $O(\log n)$ time

Thank you

arXiv:2308.09683