Sensitivity Oracle for All-pairs Mincuts

Karthik Murali

School of Computer Science,
Carleton University, Ottawa, Canada

April 14, 2023

Introduction

Sensitivity Oracles

- Most data structures are designed for static graphs.
- But, real world graphs are dynamic.
- We need data structures that can handle failures and insertions of edges.

$\star$ **Sensitivity Oracles** $\star$

What shall we learn?

We will learn about a sensitivity oracle for all-pairs mincuts.

- **Given**
  - → Vertices s, t
  - → An edge e to be inserted/deleted
- **Output**
  - → Did the value of (s,t)-mincut change upon insertion/deletion?
  - → If yes, report one such mincut.

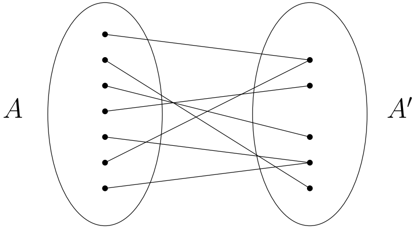
What is all this? I need definitions!

Preliminaries

Cuts in a graph

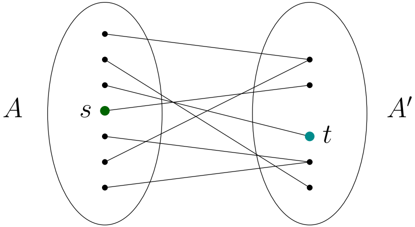
(All graphs G are undirected and unweighted.)

- **Cut:** A partition of $V(G)$ into non-empty sets (A, A').
- **Size of a cut:** No. of edges with one end in A and the other in A'.
- **Global mincut:** A cut of minimum size.



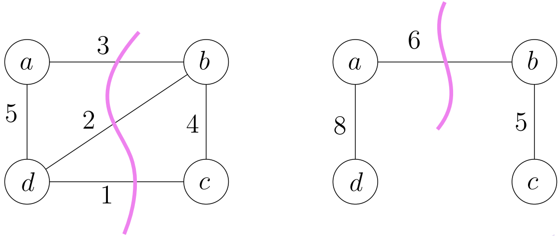
(s, t)-cuts

- (s, t)**-cut:** A cut with $s \in A$ and $t \in A'$.
- (s, t)**-mincut:** An (s, t)-cut of minimum size.



Gomory-Hu trees: a data structure for (s, t) mincuts

- **Gomory-Hu tree:** A weighted tree T of G such that:
  - → $V(T) = V(G)$
  - → $T - uv$ gives a (u, v)-mincut of G with size $w(uv)$
- A GHT can store (s, t)-mincuts for all pairs s, t of G. Moreover, (s, t)-mincut values can be obtained in $O(1)$ time.



What's our goal?

Goal: Design a data structure to answer the following queries:

- FT-MINCUT($\{s, t\}, e$): Does the deletion of e decrease the value of (s, t)-mincut?
- IN-MINCUT($\{s, t\}, e$): Does the insertion of e increase the value of (s, t)-mincut?

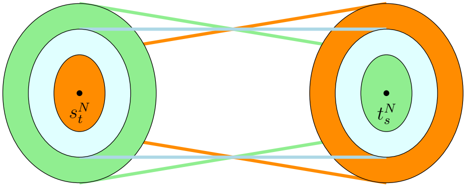
In both cases, report one such (s, t)-mincut.

Necessary and sufficient conditions for change in size of (s, t)-mincut

- **Edge** xy **is removed:** Size of an (s, t)-mincut decreases by 1 $\iff$ xy is present in some (s, t)-mincut
- **Edge** xy **is inserted:** Size of an (s, t)-mincut increases by 1 $\iff$ xy is present in every (s, t)-mincut

An alternate characterisation for the insertion case

- s_t^N - **Nearest mincut from s to t:** A set of vertices with the property that every (s,t)-mincut (A, A') with $s \in A$ has $s_t^N \subseteq A$.



- **Edge xy is inserted:** Size of an (s,t)-mincut increases by 1 $\iff$ xy is present in every (s,t)-mincut $\iff$ $x \in s_t^N$ and $y \in t_S^N$.

GHT as a sensitivity oracle for (s, t)-mincuts

- Store $O(n^2)$ copies of GHT, one for the insertion/deletion of each edge.
- Takes $O(n^3)$ space and $O(1)$ query-time.

Question: Can we achieve $O(1)$ query time using $O(n^2)$ space?

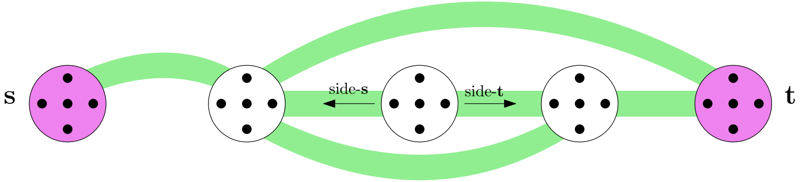
This is the goal of today's talk. So, the answer is YES!

Strip $\mathcal{D}_{s,t}$:

A sensitivity oracle for (s, t)-mincuts

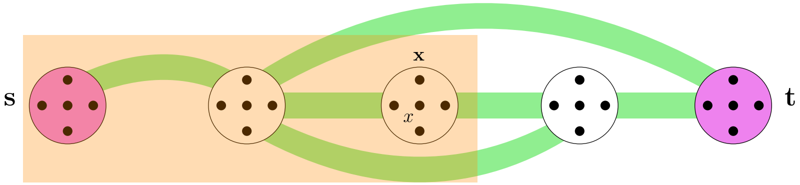
Structure of $\mathcal{D}_{s,t}$

- This is a quotient graph of G with a DAG-like structure.
- *Terminal nodes:* Nodes containing s and t; denoted by **s** and **t**.
- All edges incident to a non-terminal node belong to side-**s** or side-**t**.



Encoding of minimum (s, t)-cuts in $\mathcal{D}_{s,t}$

- $\mathcal{R}_s(x)$ - **Reachability cone of a vertex x:** The set of all vertices in the direction from **x** to **s**.



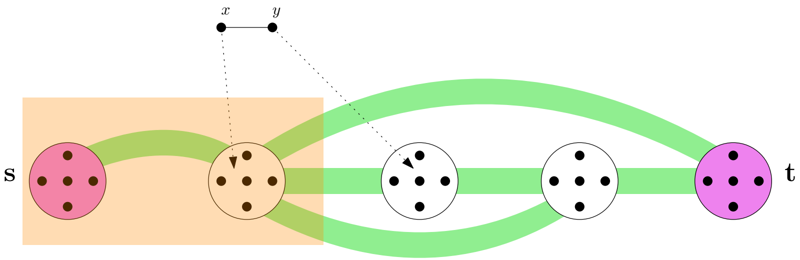
Theorem 1

A set of vertices A is an (s, t)-mincut $\iff$ the corresponding vertices in $\mathcal{D}_{s,t}$ defines a reachability cone.

Corollaries of Theorem 1

Corollary 2

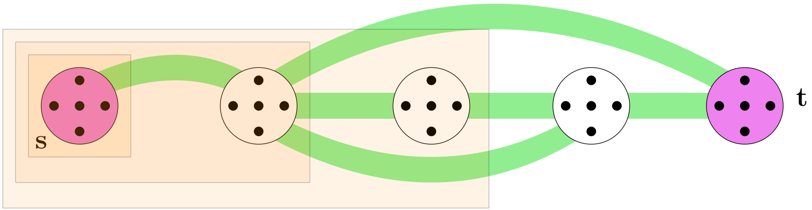
An edge xy belongs to an (s,t)-mincut if and only if it appears in $\mathcal{D}_{s,t}$.



Corollaries of Theorem 1

Corollary 3

The set of vertices mapped to $\mathbf{s}$ *defines* s_t^N*. Similarly, the set of vertices mapped to* $\mathbf{t}$ *defines* t_s^N*.*



Strip $\mathcal{D}_{s,t}$ as a sensitivity oracle

- **FT-MINCUT($\{s,t\}, xy$):** Check if x and y are mapped to different nodes $\mathbf{x}$ and $\mathbf{y}$. ($O(1)$ time)

  If yes, report vertices mapped to $\mathcal{R}_s(\mathbf{x})$. ($O(n)$ time)

- **IN-MINCUT($\{s,t\}, xy$):** Check if x is mapped to $\mathbf{s}$ and y is mapped to $\mathbf{t}$. ($O(1)$ time)

  If yes, report any mincut, say vertices mapped to $\mathbf{s}$. ($O(n)$ time)

$\mathcal{D}_{s,t}$ as a sensitivity oracle for all-pairs mincut

- Store $O(n^2)$ copies of $\mathcal{D}_{s,t}$ one for the insertion/deletion of each edge.

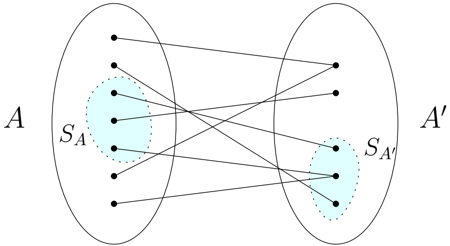
  The space occupied is $O(n^3)$ and query time is $O(1)$. This solution is no better than the trivial solution of using Gomory-Hu trees.

The connectivity carcass

A sensitivity oracle for Steiner mincuts

Steiner cuts

- Let $S \subseteq V(G)$ be a fixed set of vertices (Steiner set).
- ***S*-cut:** A cut (A, A') with $A \cap S \neq \emptyset$ and $A' \cap S \neq \emptyset$.
- ***S*-mincut:** An S-cut with minimum size.



What's our goal?

- **Notation:** c_S = size of S-mincut; $c_{s,t}$ = size of (s,t)-mincut

Observation 4

Let $s, t \in S$. Then $c_{s,t} \geq c_S$. Moreover, $c_{s,t} > c_S \iff s, t$ are not separated by any S-mincut.

- **Objective:** Suppose that $c_{s,t} = c_S$. Design a sensitivity oracle that can answer FT-MINCUT$(\{s,t\}, e)$ and IN-MINCUT$(\{s,t\}, e)$ in $O(1)$ time.

  Also, report the corresponding S-mincuts in $O(n)$ time.

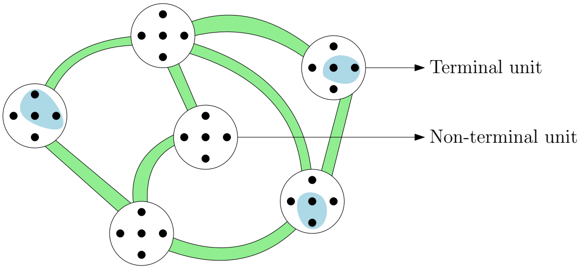
The Connectivity Carcass

Connectivity Carcass: A data structure that meets the objective. It has the following components (w.r.t a fixed Steiner set S):

1. Flesh graph $\mathcal{F}_S$
2. Skeleton $\mathcal{H}_S$
3. Projection map π_S

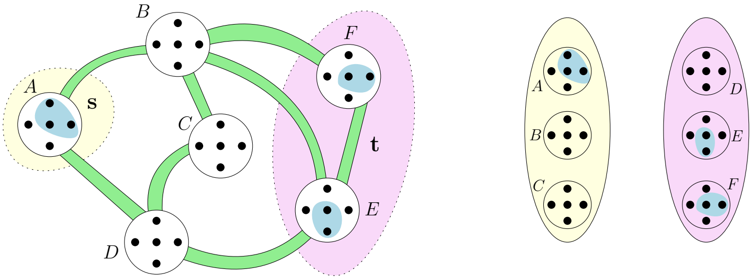
(1) The flesh graph $\mathcal{F}_S$

- A quotient graph of G; it generalises $\mathcal{D}_{s,t}$ to S-mincuts.
- **Units:** Vertices of $\mathcal{F}_S$. Vertices of G mapped to a unit are not separated by any S-mincut.
- **Terminal unit:** A unit that contains an S-vertex.



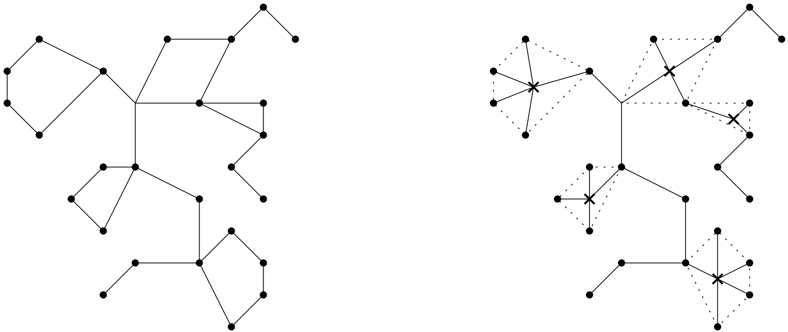
$\mathcal{F}_S \rightarrow$ Strip corresponding to an S-mincut

A strip for an S-mincut can be constructed from $\mathcal{F}_S$ by contracting the Steiner units on each side of the cut into source **s** and terminal **t**.



(2) The skeleton $\mathcal{H}_S$

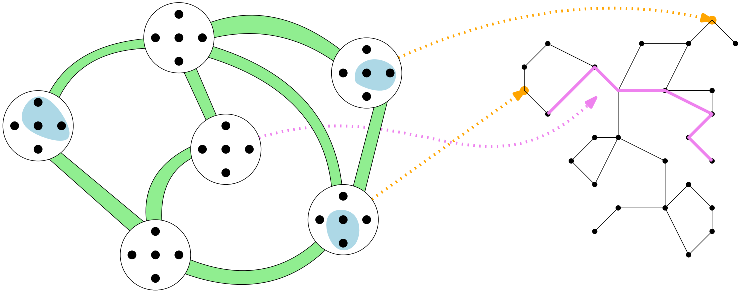
- $\mathcal{H}_S$ is a cactus graph (two cycles share ≤ 1 vertex).
- For fast computations, H_S is stored in the form of its "dual", the skeleton tree $T(\mathcal{H}_S)$.



(3) The projection map π_S

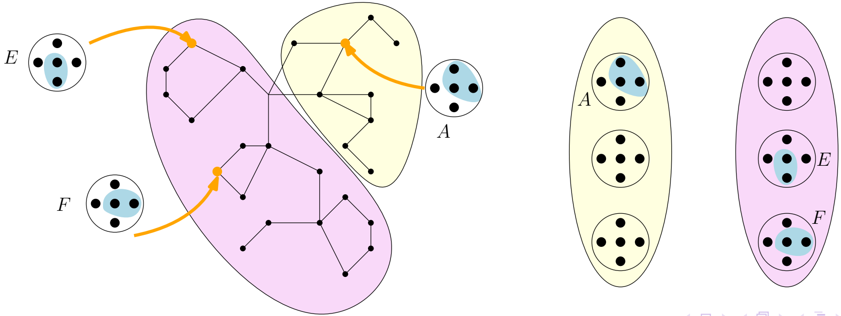
π_S is a mapping from units of $\mathcal{F}_S$ to paths of $\mathcal{H}_S$.

- Terminal units are mapped to nodes of $\mathcal{H}_S$.
- Non-terminal units are mapped to *proper paths*, i.e., paths which intersect ≤ 1 edge of each cycle



Mincuts of $\mathcal{H}_S$ store bunches of S

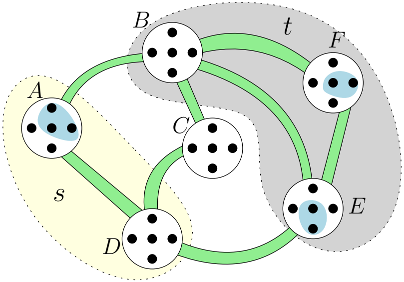
- **Bunch:** An equivalence of S-mincuts, where two S-mincuts are equivalent if they divide S in the same way.
- Every mincut of $\mathcal{H}_S$ (tree edge or 2 cycle edges) corresponds to a bunch given by the Steiner units on either side of the mincut.

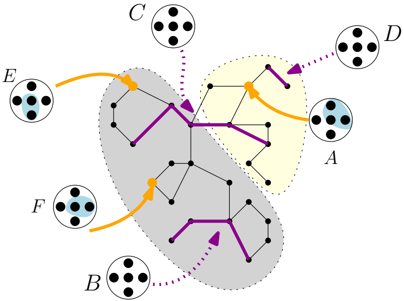


$\mathcal{F}_S \rightarrow$ Strip corresponding to a bunch

Consider any mincut of $\mathcal{H}_S$ and the corresponding bunch $\mathcal{B}$. To construct strip $\mathcal{D}_{\mathcal{B}}$:

1. All units which are mapped completely to one side of the cut get contracted into **s** or **t**.
2. Non-terminal units whose paths cross the cut are retained as such.





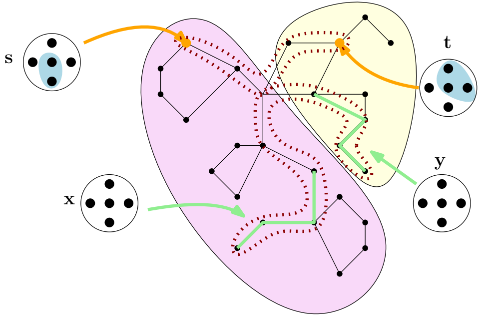
Necessary and sufficient condition for an edge to be in (s,t)-mincut

For any two vertices u, v, let $P(u,v) \rightarrow$ a path in $\mathcal{H}_S$ with prefix $\pi_S(\mathbf{u})$ and suffix $\pi_S(\mathbf{v})$.

Theorem 5

Let $s, t \in S$ such that $c_{s,t} = c_S$. Then xy belongs to an (s,t)-mincut $\iff$ there is a mincut of $\mathcal{H}_S$ whose edges intersect both $P(s,t)$ and $P(x,y)$.

The theorem is simplified for ease of presentation, but is not precisely correct



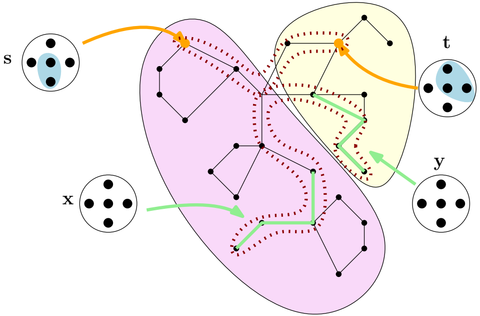
Proof of Theorem 5

Proof.

The following statements are equivalent:

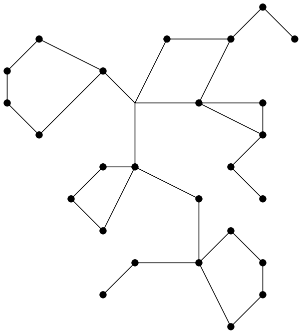
1. xy is an edge of some (s,t)-mincut.
2. xy is an edge of some bunch $\mathcal{B}$ that separates s and t.
3. There is an $x \rightsquigarrow y$ path in $\mathcal{D}_{\mathcal{B}}$ whose terminal nodes contain s and t.
4. There is a mincut of $\mathcal{H}_S$ corresponding to $\mathcal{B}$ whose edges intersect both $P(s,t)$ and $P(x,y)$

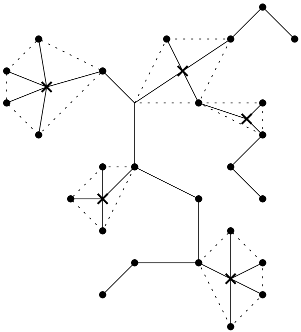
□



Answering the query FT-MINCUT

- Theorem 5 can be tested on the skeleton tree $T(\mathcal{H}_S)$.
- Testing whether two paths in a tree intersect can be done in $O(1)$ time using LCA queries.





Reporting (s, t)-mincut that contains xy

Theorem 5 gives the following method:

1. Find the mincut of $\mathcal{H}_S$ that intersects both $P(s, t)$ and $P(x, y)$.
2. Construct the strip $\mathcal{D}_{\mathcal{B}}$ associated with this mincut.
3. Compute the reachability cone $\mathcal{R}_{\mathbf{s}}(x)$ from the strip.

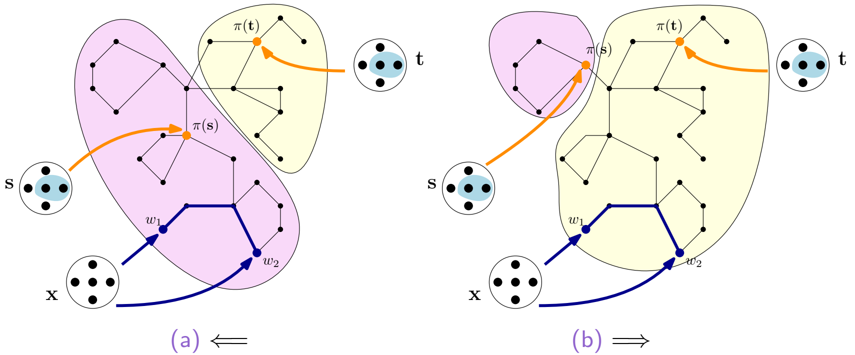
While this method requires $O(m)$ time, it is possible to reduce it to $O(n)$. (See [BP22] for more details.)

Necessary and sufficient conditions for a vertex to be in s_t^N

Let $x \in V(G)$ and $P(w_1, w_2) = \pi(\mathbf{x})$.

Theorem 6

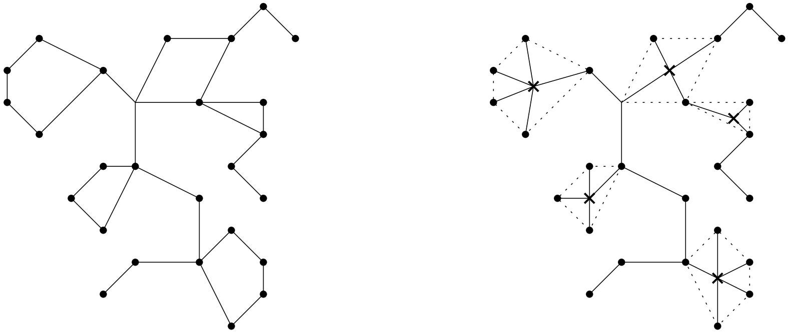
Let s, t be separated by some S-mincut. Then $x \in s_t^N$ if and only if $\pi(\mathbf{s})$ intersects all paths $P(w_1, \pi(\mathbf{t}))$ and $P(w_2, \pi(\mathbf{t}))$.



$x \in s_t^N$ iff $\mathbf{x}$ and $\mathbf{s}$ are mapped to the same side of every mincut of $\mathcal{H}_S$.

Answering the query IN-MINCUT

The conditions mentioned in Theorem 6 can be tested on $T(\mathcal{H}_S)$ in $O(1)$ time using LCA queries.



Summary

To answer the queries FT-MINCUT$(\{s,t\},e)$, FT-MINCUT$(\{s,t\},e)$ and to report the corresponding mincuts, we only need the following:

1. Projection map π_S
2. Skeleton tree $T(\mathcal{H}_S)$

Define $D(S) := (\pi_S, T(\mathcal{H}_S))$.

Hierarchical tree $\mathcal{T}_G$

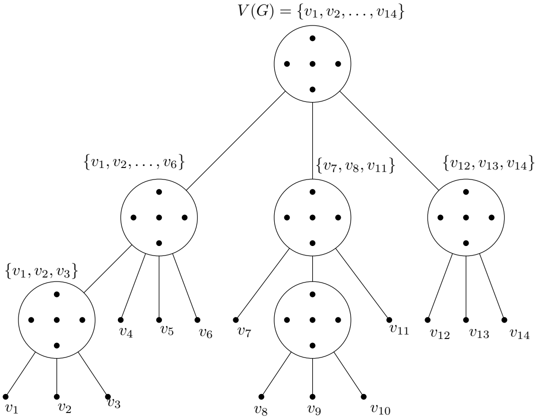
A sensitivity oracle for all-pairs mincuts

The hierarchical tree $\mathcal{T}_G$

- Root $= V(G)$ and leaves are individual vertices of G.
- Let S be the set of vertices at a node $w \in \mathcal{T}_G$. The children of w are the equivalence class of nodes satisfying the relation $c_{u,v} > c_S$.

Observation 7

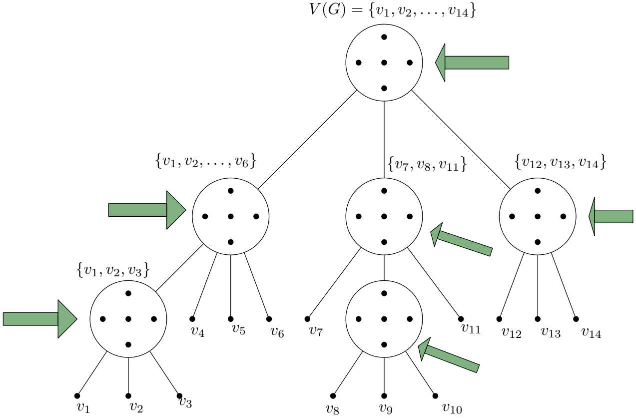
For any two vertices u, v, $c_{u,v} = c_S$ where S is the set of vertices at the LCA of u, v in $\mathcal{T}_G$.



Sensitivity oracle for all-pairs mincuts

To get a sensitivity oracle for all-pairs mincut:

1. Augment each internal node with $D(S)$.
2. To answer the queries $\textsc{ft-mincut}(\{s,t\}, e)$ and $\textsc{in-mincut}(\{s,t\}, e)$, use $D(S)$ at the LCA of s, t.
3. Space $= O(n^2)$; Query-time $= O(1)$



Summary

- We got a sensitivity oracle by appending nodes of $\mathcal{T}_G$ with $D(S)$.
- This occupies $O(n^2)$ space with $O(1)$ query-time. Moreover, corresponding mincuts can be returned in $O(n)$ time.
- The paper [BP22] discusses another data structure with $O(n)$ space and $O(\min(m, nc_{s,t}))$ query-time

Open questions

Open Problem 1

Design a sensitivity oracle for all-pairs mincuts that occupies $o(n^2)$ space and $o(m)$ query-time.

Open Problem 2

Design a sensitivity oracle for all pairs mincuts that can tolerate the removal or insertion of upto $k > 1$ edges.

References

[BP22] Surender Baswana and Abhyuday Pandey.
Sensitivity oracles for all-pairs mincuts.
In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 581–609. SIAM, 2022.