Almost-Linear Time Algorithms for Decremental Graphs: Min-Cost Flow and More via Duality

Li Chen

Stanford 05/17/24

Joint work with





Jan van den Brand
Georgia Tech

Rasmus Kyng
ETH

Yang P. Liu
IAS

Simon Meierhans
ETH

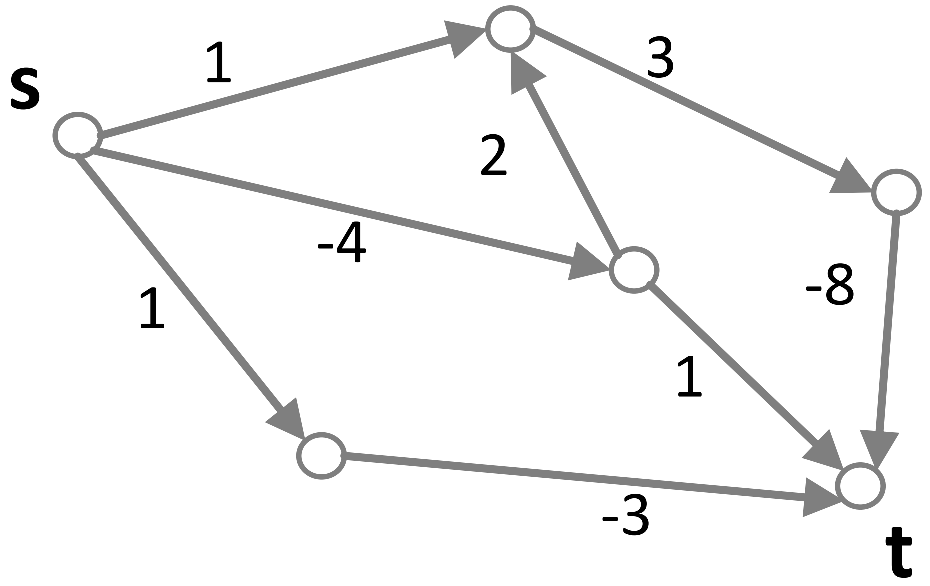
Maximilian Probst Gutenberg
ETH

Sushant Sachdeva
U. Toronto

Uncapacitated Min-Cost Flow

Directed graph G = (V, E). m edges, n vertices, source s, sink t

edge *costs* c_e, integer in $[-C, C]$, where $C = m^{O(1)}$

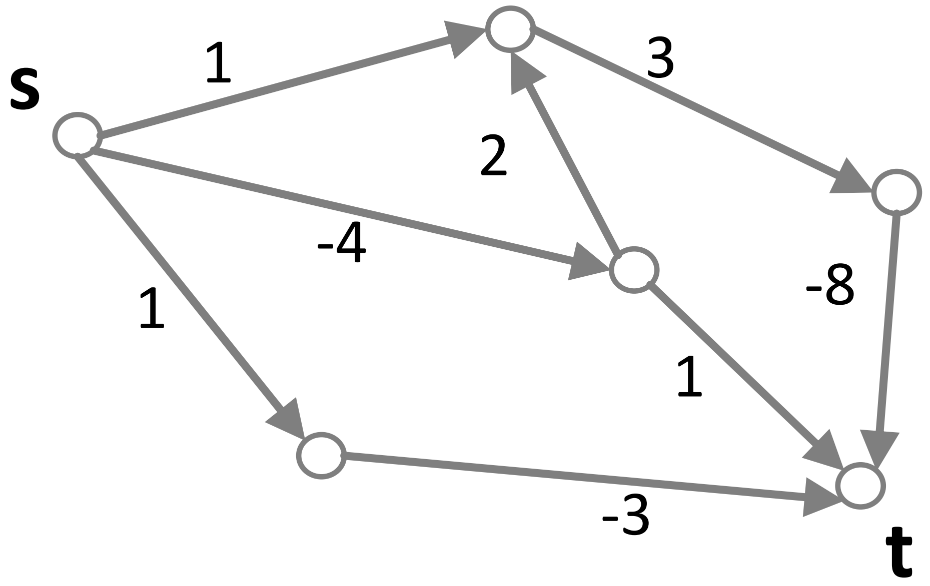


Uncapacitated Min-Cost Flow

Directed graph G = (V, E). m edges, n vertices, source s, sink t

edge *costs* c_e, integer in $[-C, C]$, where $C = m^{O(1)}$

Goal: Route 1 unit of flow from $s \to t$, minimize cost

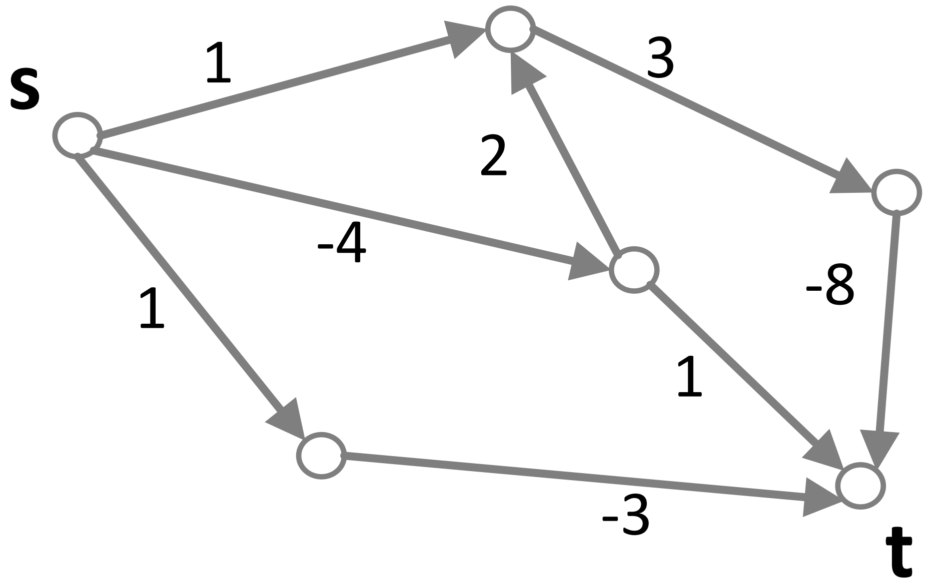


Uncapacitated Min-Cost Flow

Directed graph G = (V, E). m edges, n vertices, source s, sink t

edge *costs* c_e, integer in $[-C, C]$, where $C = m^{O(1)}$

Goal: Route 1 unit of flow from $s \to t$, minimize cost



Real-weighted shortest st-path

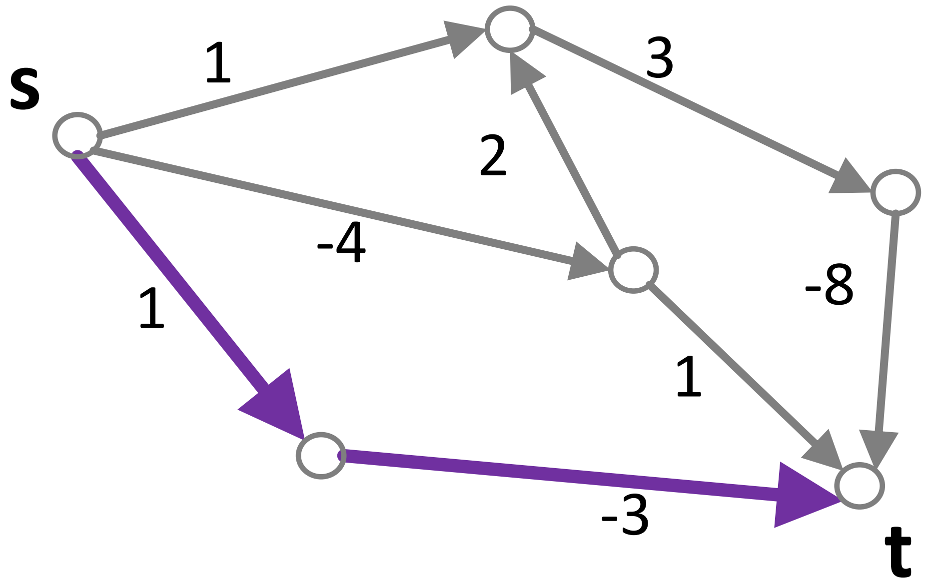
Uncapacitated Min-Cost Flow

Directed graph G = (V, E). m edges, n vertices, source s, sink t

edge *costs* c_e, integer in $[-C, C]$, where $C = m^{O(1)}$

Goal: Route 1 unit of flow from $s \to t$, minimize cost

Real-weighted shortest st-path



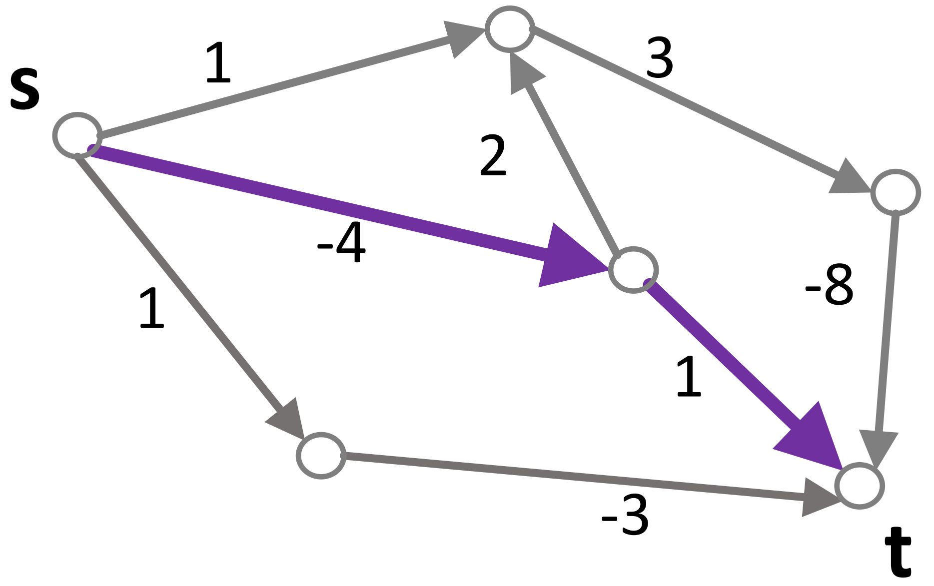
Cost = -2

Uncapacitated Min-Cost Flow

Directed graph G = (V, E). m edges, n vertices, source s, sink t

edge *costs* c_e, integer in $[-C, C]$, where $C = m^{O(1)}$

Goal: Route 1 unit of flow from $s \to t$, minimize cost



Cost = -3

Real-weighted shortest st-path

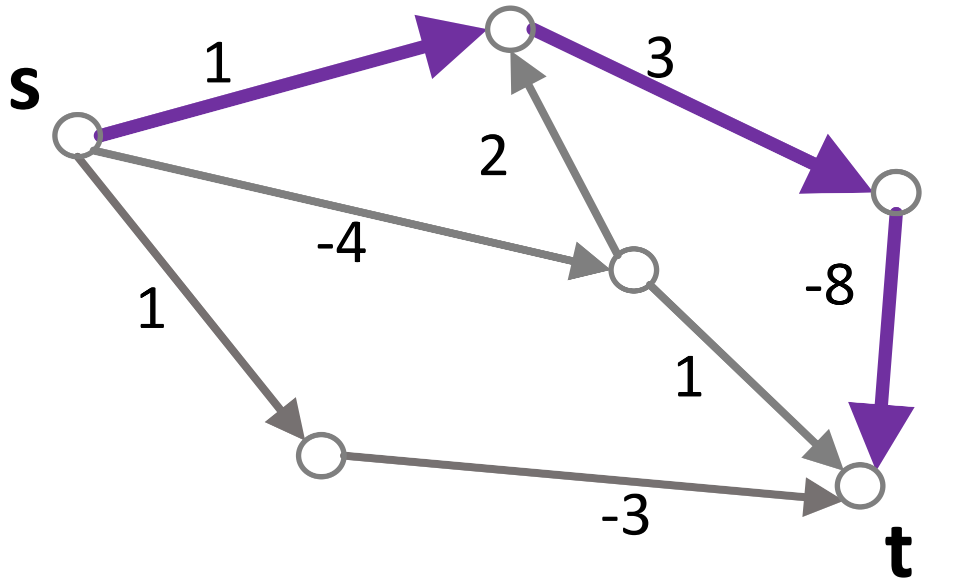
Uncapacitated Min-Cost Flow

Directed graph G = (V, E). m edges, n vertices, source s, sink t

edge *costs* c_e, integer in $[-C, C]$, where $C = m^{O(1)}$

Goal: Route 1 unit of flow from $s \to t$, minimize cost

Real-weighted shortest st-path

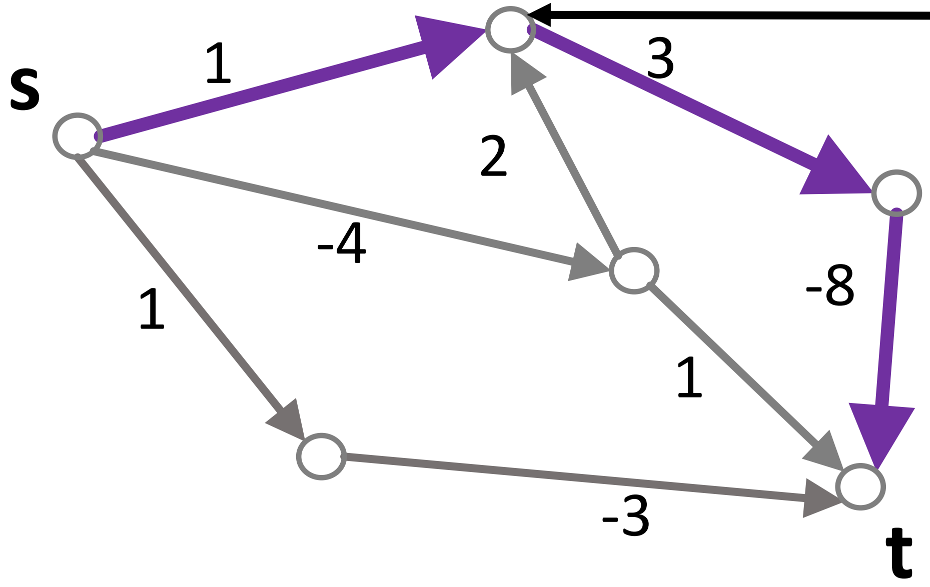


Cost = -4

Linear Algebraic View

- $f \in \mathbb{R}^E$, i.e. a real vector on the edges

Goal: minimize
$c^T f$



Flow conservation:
all other vertices have incoming flow=outgoing flow

Flow routes the demand:
Net flow on s is +1
Net flow on t is -1

Linear Program

$$\min_f c^T f = \sum_e c_e f_e$$

Min Cost

For all edges e

$$f_e \geq 0$$

Direction constraints

For all vertices x

$$B^T f = \chi_{s,t}$$

Unit flow from s to t

Transshipment Primal LP

$$\min_f c^T f = \sum_e c_e f_e$$

Min Cost

For all edges e

$$f_e \geq 0$$

Direction constraints

For all vertices x

$$B^T f = d$$

Route the demand

Transshipment Primal LP

$$\min_f c^T f = \sum_e c_e f_e$$ Min Cost

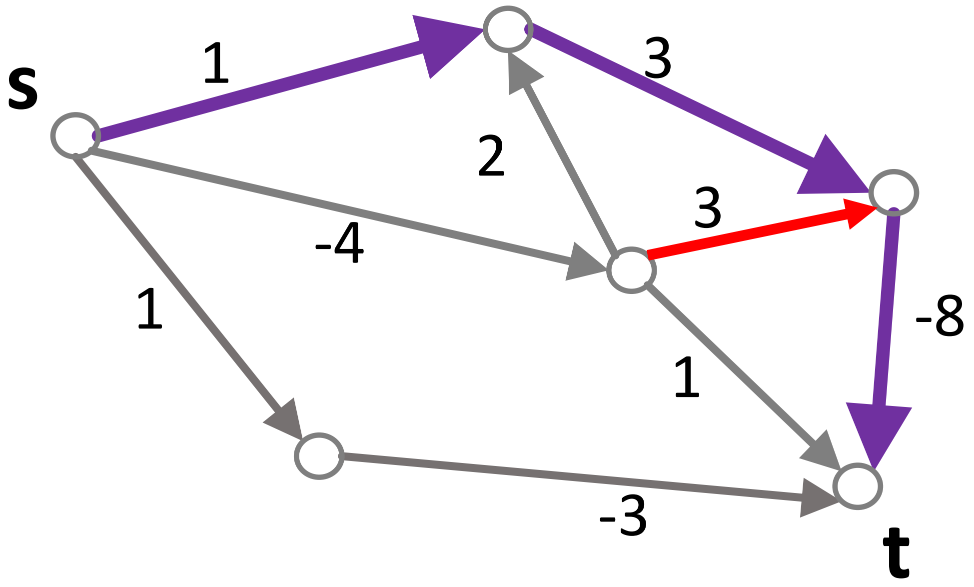
For all edges e $$f_e \geq 0$$ Direction constraints

For all vertices x $$B^T f = d$$ Route the demand

Generalize weighted bipartite matching, capacitated min-cost flow, shortest path, max flow, …

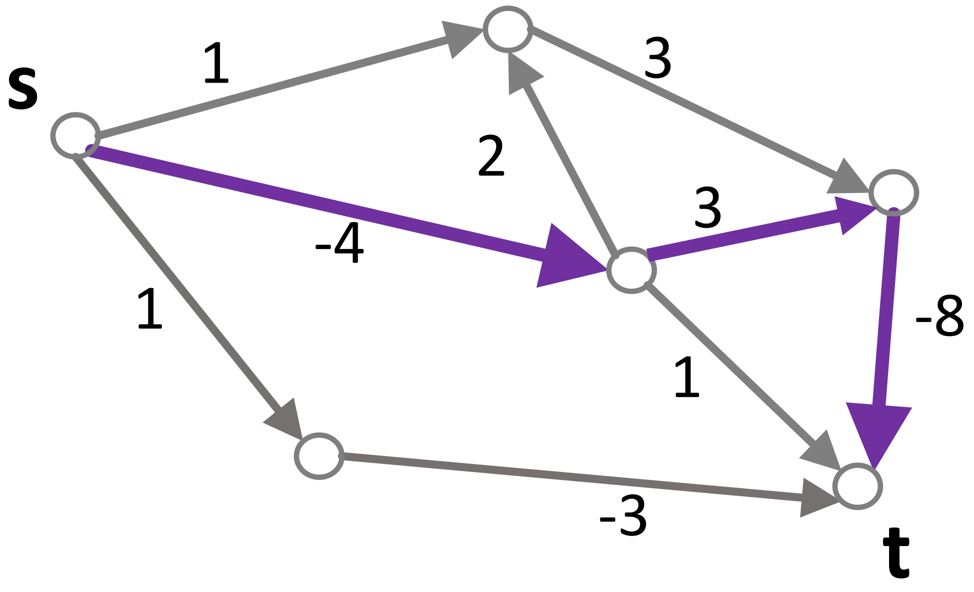
Edge insertions: Flow stays feasible

- Add an edge, put 0 flow on it
- Flow stays feasible



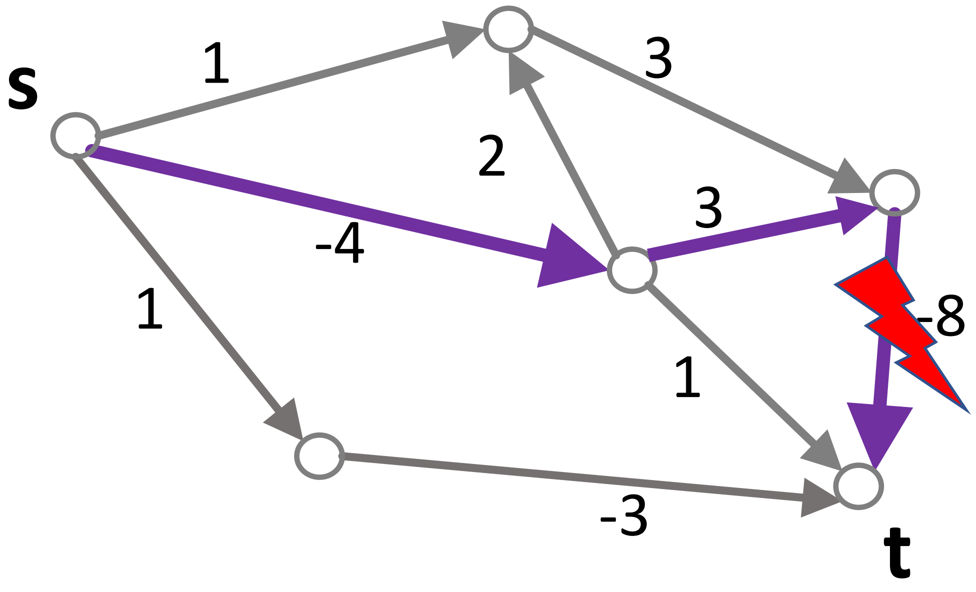
Edge insertions: Flow stays feasible

- Add an edge, put 0 flow on it
- Flow stays feasible
- Then move towards smaller cost
- Incremental min-cost flow in almost-linear time
  - [C-Kyng-Liu-Meierhans-Probst Gutenberg '24]
  - Maintain a primal solution under relaxing updates



Edge deletions: Flow no longer feasible

- Not an st-path anymore
- Edge deletion does not relax the primal problem
- How about the dual?



Transshipment Dual LP

$$\max_{y \in R^V} d^T y$$

For any edge $e = (u, v)$ $\quad c_e \geq y_u - y_v$

Triangle Inequality
Non-negative Slack

Transshipment Dual LP

$$\max_{y \in R^V} d^T y$$

For any edge $e = (u, v)$ $\qquad c - By \geq 0$

Triangle Inequality
Non-negative Slack

- Remove an edge -> remove one constraint
- Current dual solution y stays feasible
- Move towards larger $d^T y$

Decremental Thresholded Transshipment

Theorem [Brand-C-Kyng-Liu-Meierhans-Probst Gutenberg-Sachdeva]

Given a decremental graph G, threshold D^*

After each edge deletion, either

- Certify that $\max_{c-By\geq 0} d^T y < D^*$, or
- Output a feasible dual potential y s.t. $d^T y \geq D^*$

in $m^{1+o(1)}$ total time deterministically

Decremental Thresholded Transshipment

Theorem [Brand-C-Kyng-Liu-Meierhans-Probst Gutenberg-Sachdeva]

Given a decremental graph G, threshold D^*

After each edge deletion, either

- Certify that $\max_{c-By\geq 0} d^T y < D^*$, or
- Output a feasible dual potential y s.t. $d^T y \geq D^*$

in $m^{1+o(1)}$ total time deterministically

Application: Decremental…

1. max flow
2. min-cost flow
3. st-shortest path
4. single-source reachability
5. SCC maintenance
6. …

Dual L1 IPM

[Karmarkar '84]

Maximize $d^T y$

Enforce $\mathrm{c} - \mathrm{B}y \geq 0$

$$\min_{y \in R^V} \Phi(y) = 100m \log(D^* - d^T y) + \sum_e -\log(c_e - (y_u - y_v))$$

- If $\Phi(y) \leq -1000m \log m$, $d^T y \geq D^* - m^{-10}$
- Start at $y^{(0)}$ with $\Phi\left(y^{(0)}\right) = O(m \log m)$
- Iteratively, $\Phi\left(y^{(t+1)}\right) \leq \Phi\left(y^{(t)}\right) - m^{-o(1)}$
  - After $T = m^{1+o(1)}$ iterations, $\Phi\left(y^{(T)}\right) \leq -1000m \log m$

Dual L1 IPM

[Karmarkar '84]

$\boxed{\text{Maximize } d^T y}$ $\boxed{\text{Enforce } \mathrm{c} - \mathrm{B}y \geq 0}$

$$\min_{y \in R^V} \Phi(y) = 100m \log(D^* - d^T y) + \sum_e -\log(c_e - (y_u - y_v))$$

- $\Phi(y + \Delta) \approx \Phi(y) + \langle \nabla\Phi, \Delta \rangle + \|UB\Delta\|_1{}^2, \ \mathrm{U} = \mathrm{diag}(\mathrm{c} - \mathrm{By})^{-1} = \sqrt{\nabla^2\Phi}$
- If $\max_{c - By \geq 0} d^T y \geq D^*, \ \min_{\Delta \in R^V} \frac{\langle \nabla\Phi, \Delta \rangle}{\|UB\Delta\|_1} \leq -0.1$ $\boxed{\text{Min-Ratio Cut}}$
- If $\frac{\langle \nabla\Phi, \Delta \rangle}{\|UB\Delta\|_1} \leq -\kappa, \Phi(y + \eta\Delta) \leq \Phi(y) - \kappa^2$ $\boxed{\langle 1, \nabla\Phi \rangle = 0}$

Decremental Transshipment Algorithm

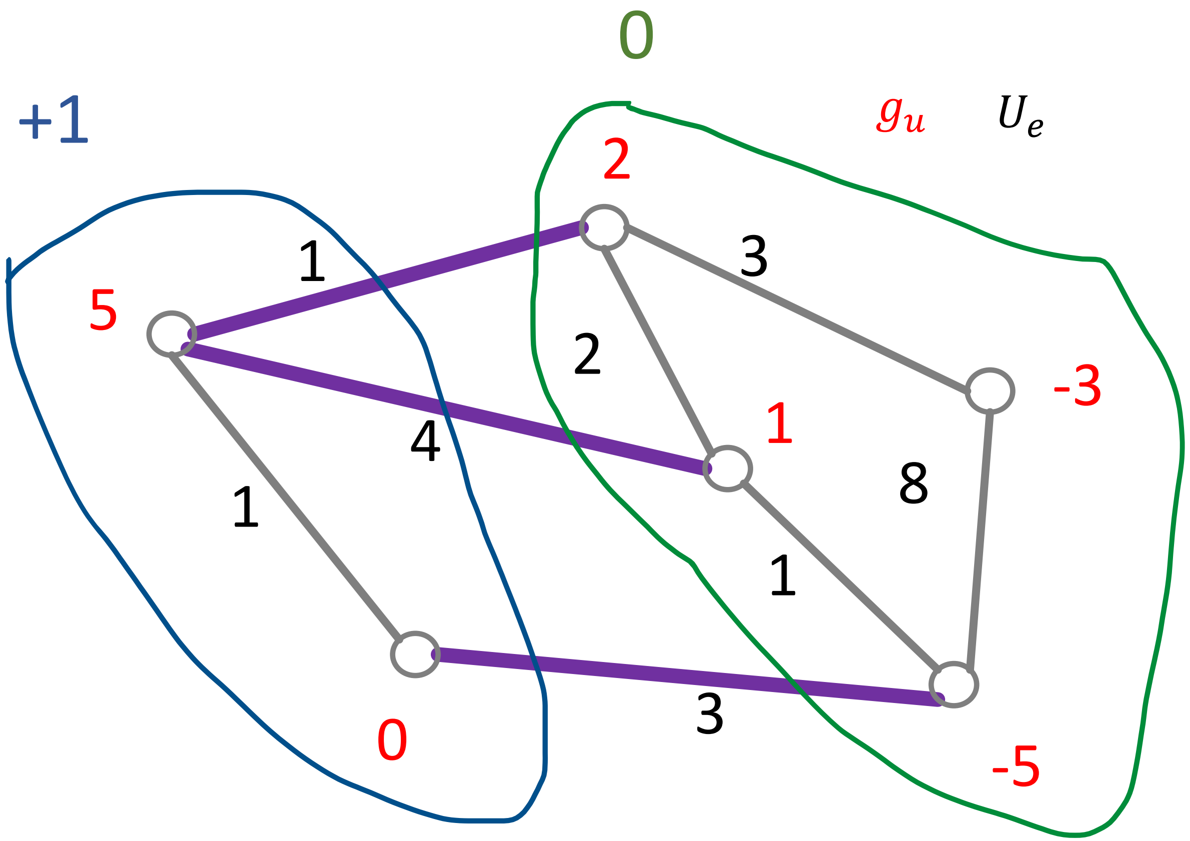
- Start at $y^{(0)}$ with $\Phi\left(y^{(0)}\right) = O(m \log m)$
- For $t = 0, 1, \dots, T = m^{1+o(1)}$, do
  - $m^{o(1)}$-approximately minimize $\min_{\Delta \in R^V} \frac{\langle \nabla\Phi, \Delta \rangle}{\|UB\Delta\|_1}$
  - If the ratio is larger than $-1/m^{o(1)}$, **certify** $\max\langle d, y\rangle < D^*$
  - Otherwise, update $y^{(t+1)} = y^{(t)} + \eta\Delta$
- Output $y^{(T)}$

can view it as a L1 trust region Newton method

Min-Ratio Cut

$$\langle 1, g \rangle = 0$$

$$\min_{\Delta \in R^V} \frac{\langle g, \Delta \rangle}{\|UB\Delta\|_1}$$

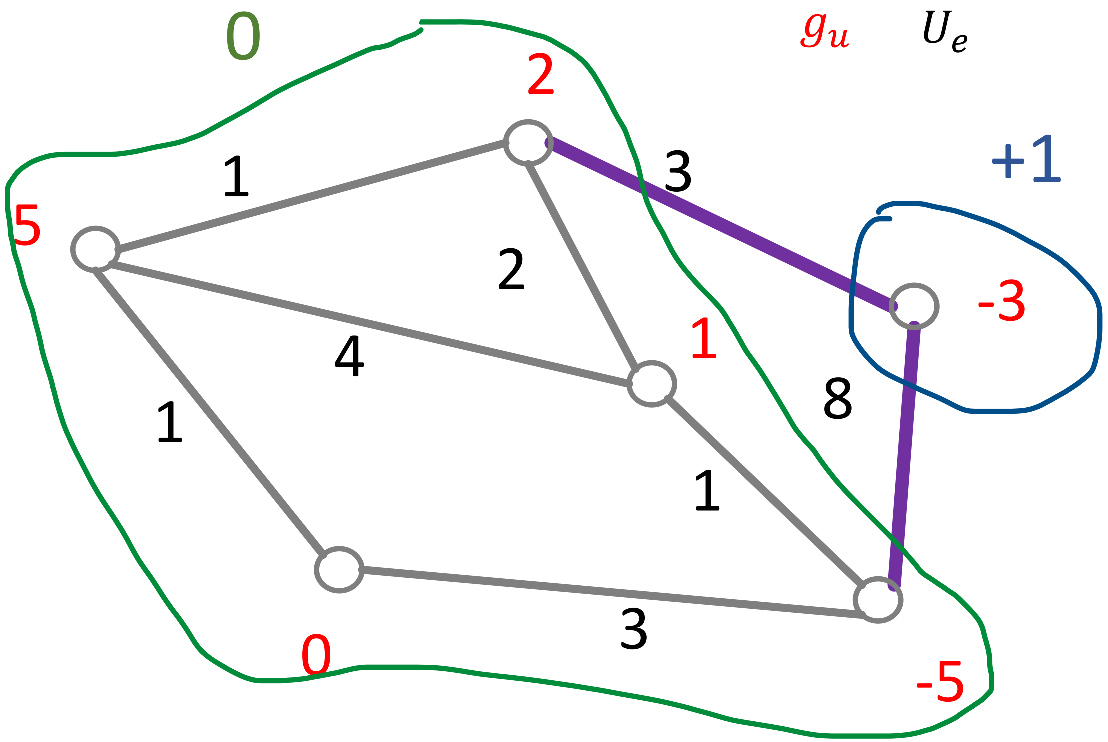


$$\langle g, \Delta \rangle = 5 + 0 = 5$$
$$\|UB\Delta\|_1 = 1 + 4 + 3 = 8$$

Min-Ratio Cut

$$\boxed{\langle 1, g \rangle = 0}$$

$$\min_{\Delta \in R^V} \frac{\langle g, \Delta \rangle}{\|UB\Delta\|_1}$$



$$\langle g, \Delta \rangle = -3$$

$$\|UB\Delta\|_1 = 3 + 8 = 11$$

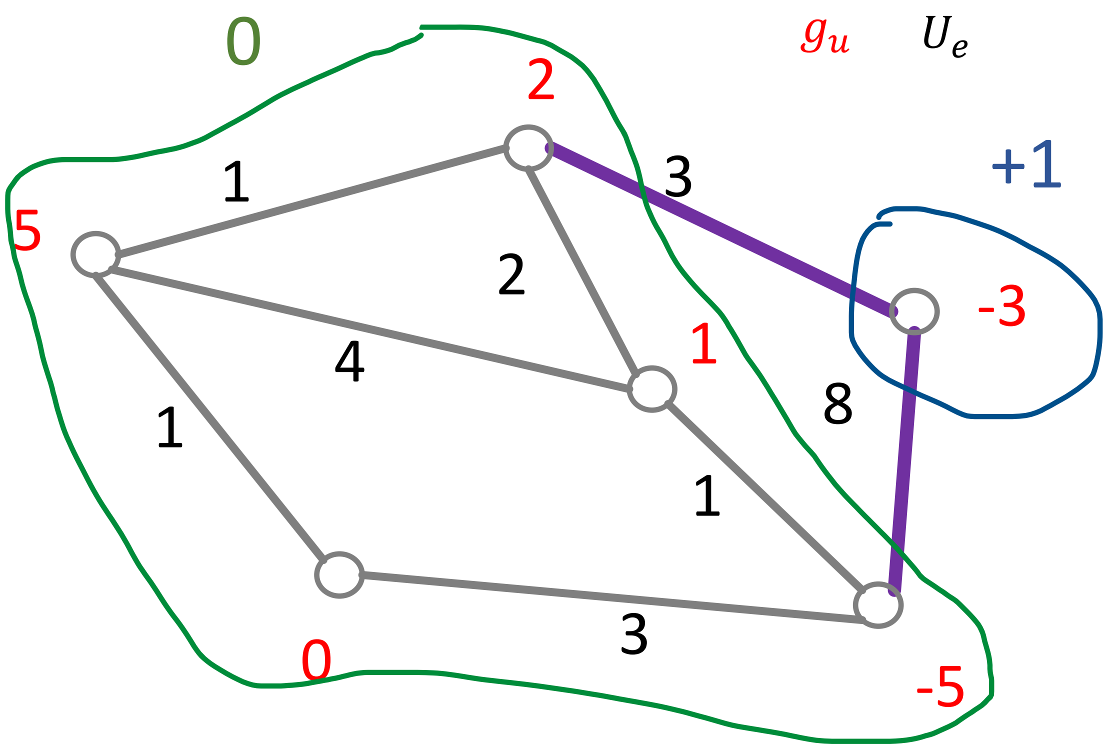
Min-Ratio Cut

$$\boxed{\langle 1, g \rangle = 0}$$

$$\min_{\Delta \in R^V} \frac{\langle g, \Delta \rangle}{\|UB\Delta\|_1}$$

Optimal solution is a cut, i.e.,

$$\Delta \in \{0,1\}^V$$



$$\langle g, \Delta \rangle = -3$$

$$\|UB\Delta\|_1 = 3 + 8 = 11$$

Fully-Dynamic Min-Ratio Cut

Theorem [Brand-C-Kyng-Liu-Meierhans-Probst Gutenberg-Sachdeva]

Given fully dynamic graph G, there's a **deterministic** data structure that

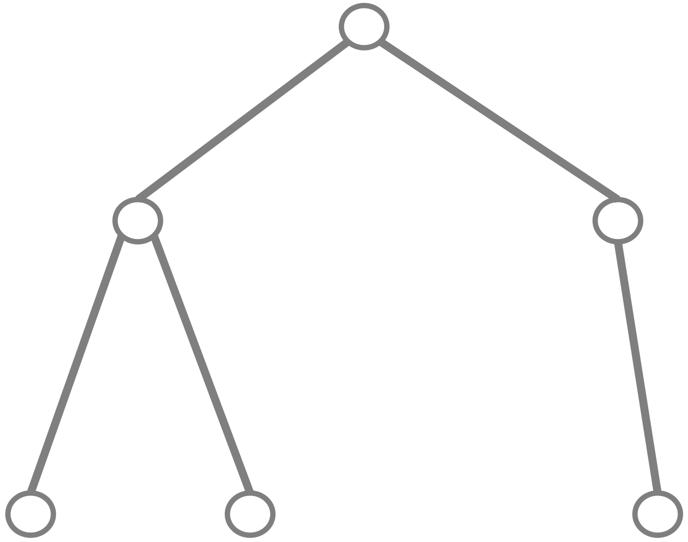
- $m^{o(1)}$-approximately minimize $\min_{\Delta \in R^V} \frac{\langle \nabla\Phi, \Delta \rangle}{\|UB\Delta\|_1}$, and
- Update $y := y + \Delta$, gradients $\nabla\Phi$ and weights $\mathrm{U} = \mathrm{diag}(\mathrm{c} - \mathrm{By})^{-1}$

In $m^{o(1)}$ time per operation

Approx Min-Ratio Cuts via Tree Cut Sparsifiers

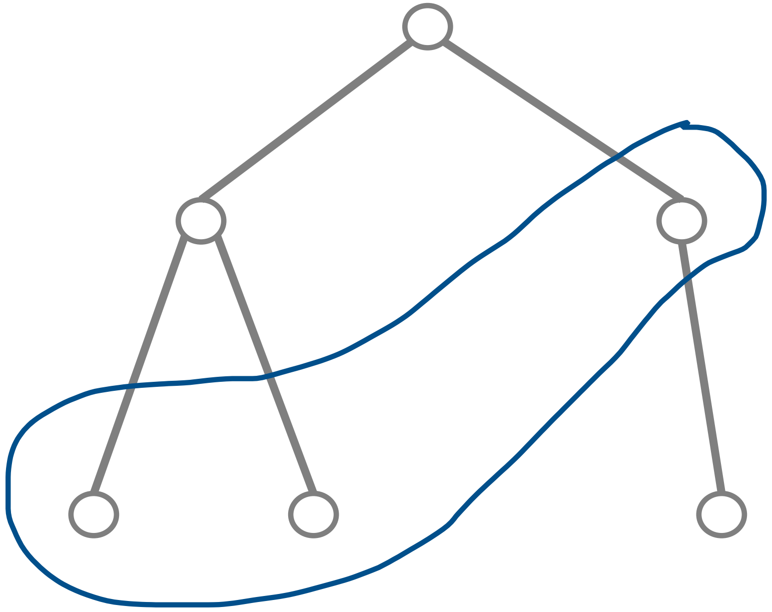
- Approximate the cut structure of G by a tree T
  - [Räcke ‘08, Mądry ‘10, Räcke-Shah-Täubig ‘14, Goranci-Räcke-Saranurak-Tan ‘21]
- Solve Min-Ratio Cut on T
  - Claim: it cuts one tree edge

Min-ratio cut on a tree cuts one edge



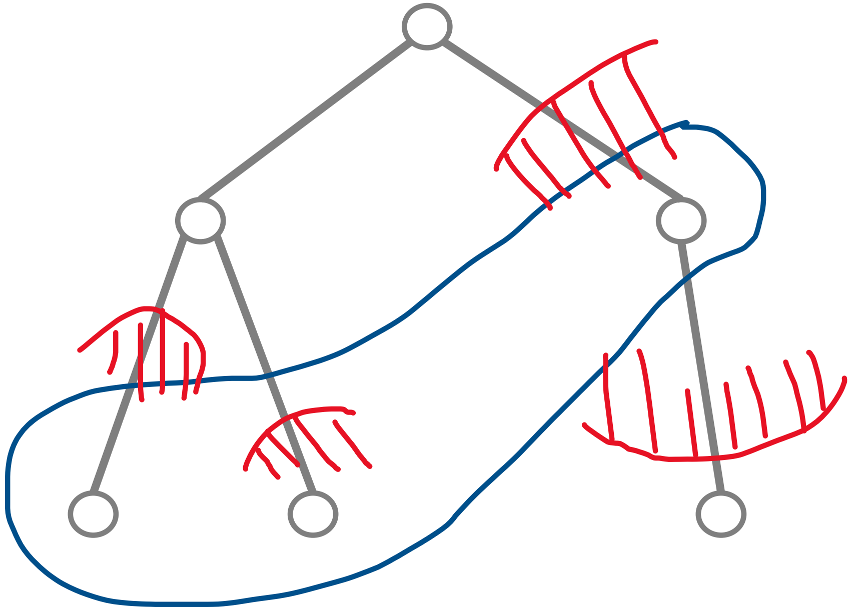
Min-ratio cut on a tree cuts one edge

Min-ratio cut Δ^*



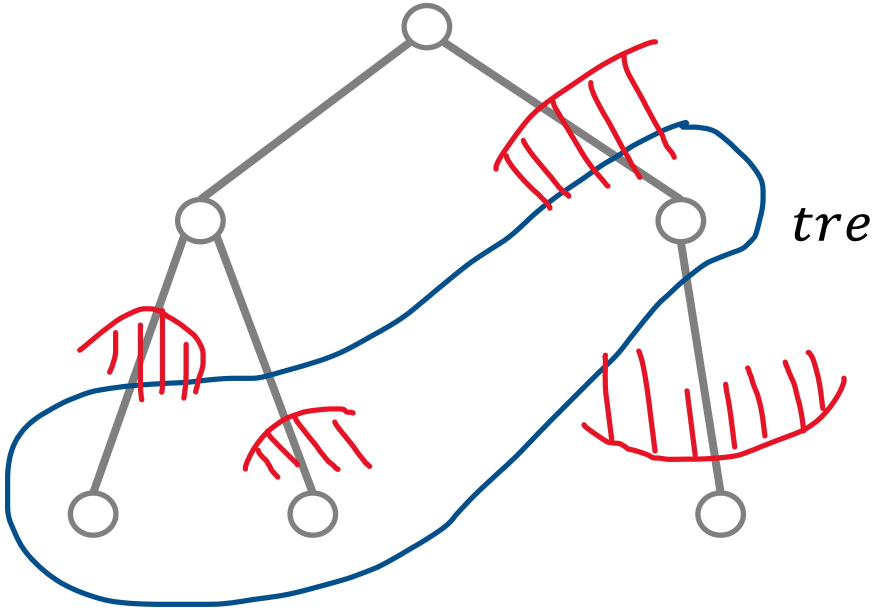
Min-ratio cut on a tree cuts one edge

Min-ratio cut Δ^*



$$\|UB\Delta^*\|_1 = \sum_{tree\ cuts\ C} \|UBC\|_1$$

Min-ratio cut on a tree cuts one edge



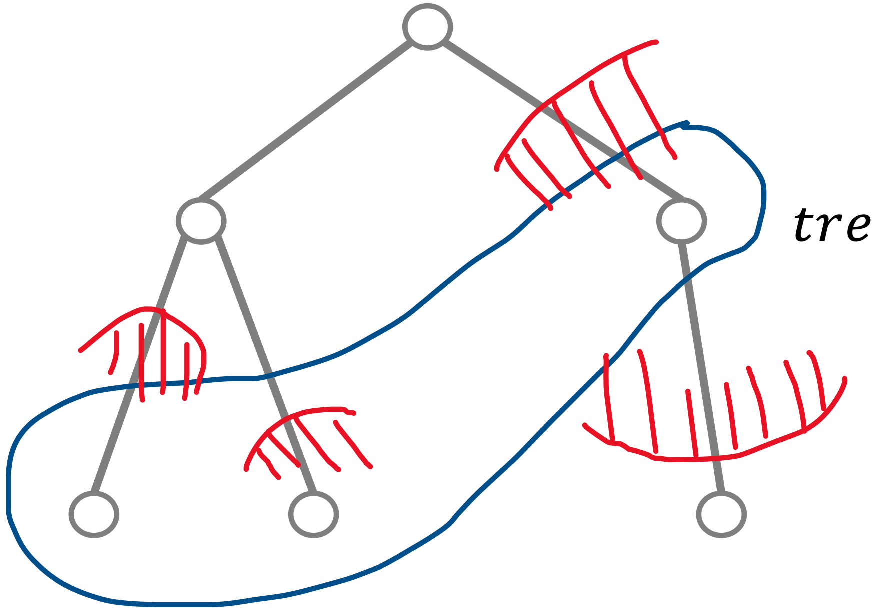
Min-ratio cut Δ^*

$$\sum_{\mathit{tree\ cuts\ C}} \langle g, C \rangle = \left\langle g, \sum_{\mathit{tree\ cuts}} C \right\rangle$$

$$\|UB\Delta^*\|_1 = \sum_{\mathit{tree\ cuts\ C}} \|UBC\|_1$$

Min-ratio cut on a tree cuts one edge

Min-ratio cut Δ^*



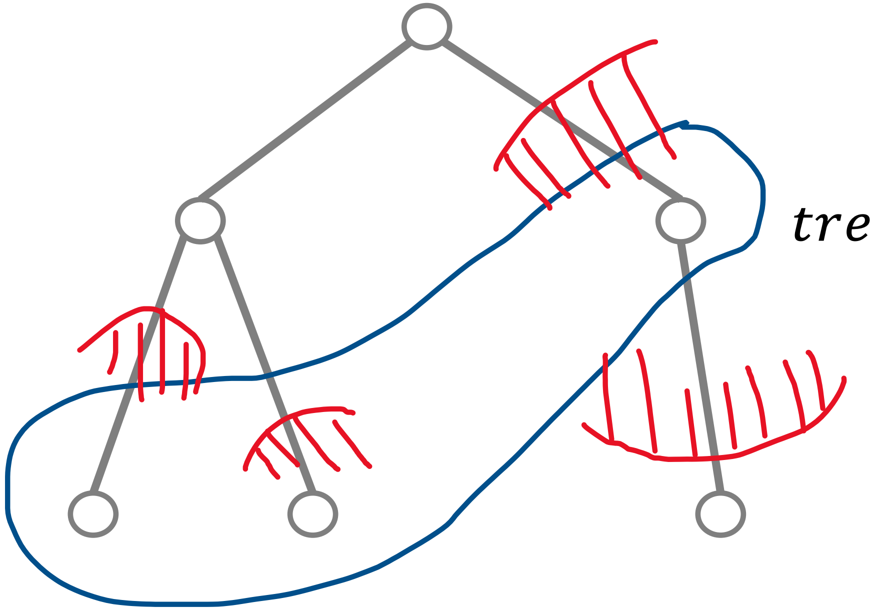
$$\sum_{tree\ cuts\ C} \langle g, C \rangle = \left\langle g, \sum_{tree\ cuts} C \right\rangle$$

$$= \langle g, \Delta^* \rangle$$

$$\|UB\Delta^*\|_1 = \sum_{tree\ cuts\ C} \|UBC\|_1$$

Min-ratio cut on a tree cuts one edge

Min-ratio cut Δ^*



$$\sum_{tree\ cuts\ C} \langle g, C \rangle = \left\langle g, \sum_{tree\ cuts} C \right\rangle$$

$$= \langle g, \Delta^* \rangle$$

$$\|UB\Delta^*\|_1 = \sum_{tree\ cuts\ C} \|UBC\|_1$$

One of is as good as Δ^*

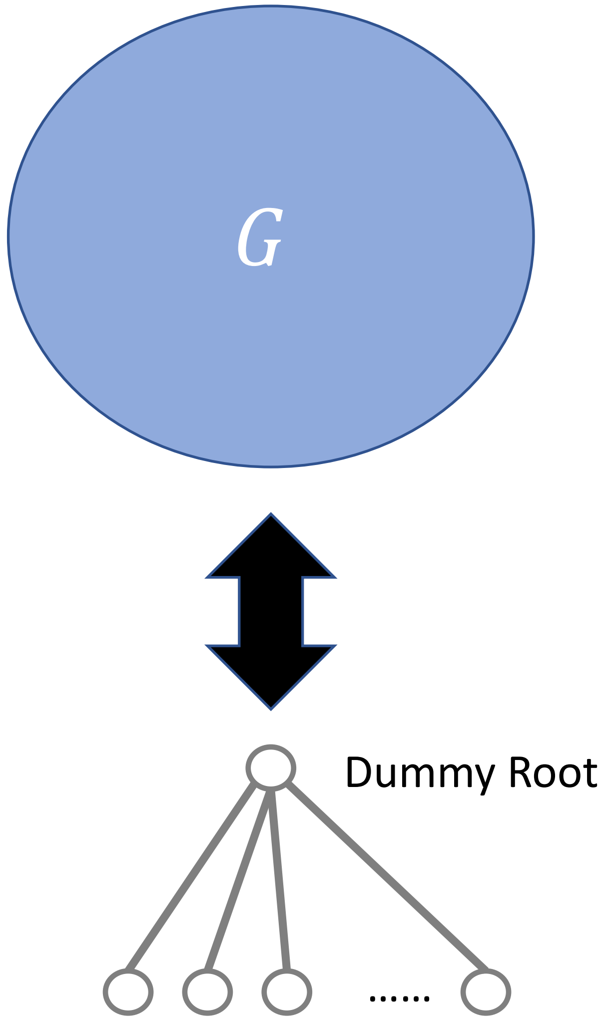
G is not always a tree

- G is a (weighted) ϕ-expander if

$$\frac{U(S, V \backslash S)}{vol_G(S)} \geq \phi, \forall\, S \subseteq V$$

- One of the singleton cut is $(1/\phi)$-approx.

$$\phi \sum_{u \in S} \deg(u) \leq U(S, V \backslash S) \leq \sum_{u \in S} \deg(u)$$

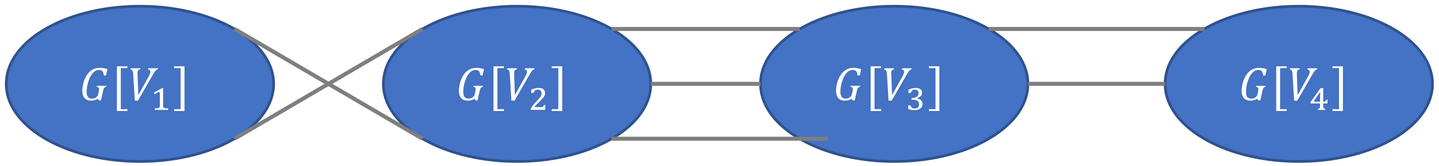


Tree Cut Sparsifier via Expander Hierarchy

[Goranci-Räcke-Saranurak-Tan '21]

- Weighted Expander Decomposition
  - Decompose G into $m^{o(1)}$-expanders $G[V_1], G[V_2]$, ... such that

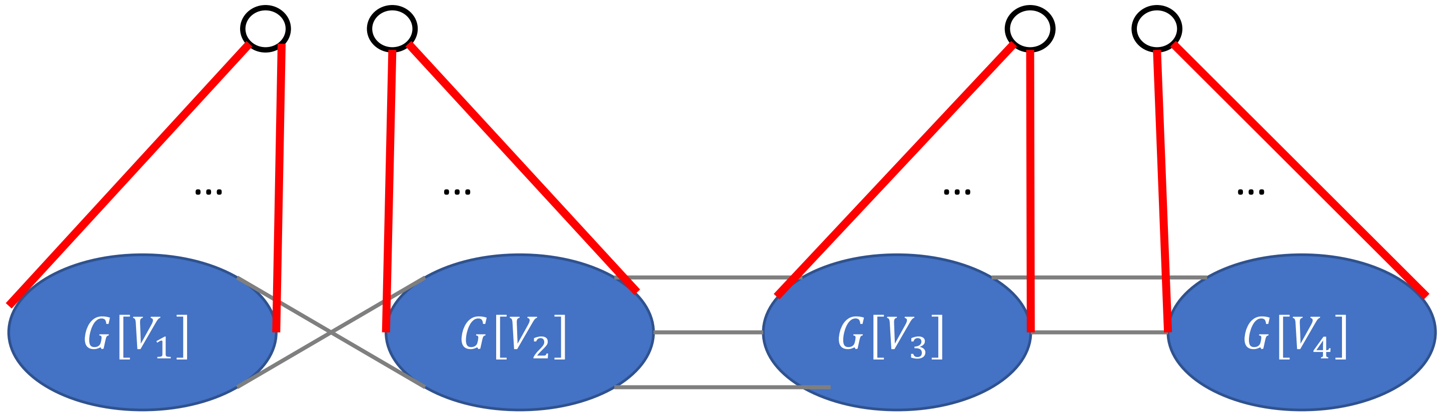
    Inter-expander edge total weight $\leq \phi \cdot m^{o(1)} \cdot total\ edge\ weight$



Tree Cut Sparsifier via Expander Hierarchy

[Goranci-Räcke-Saranurak-Tan ‘21]

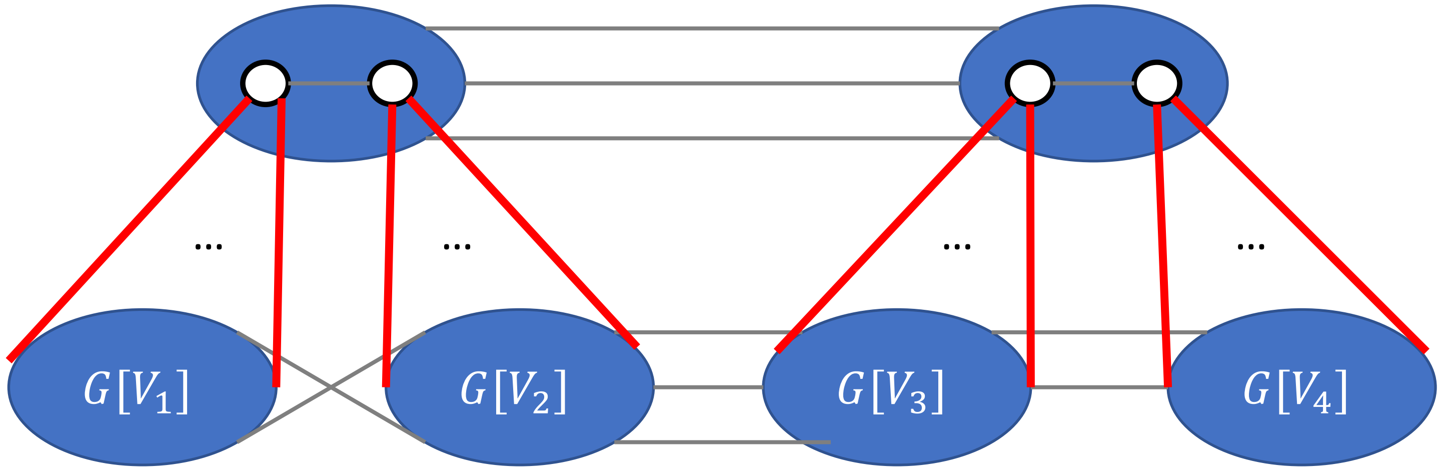
- Weighted Expander Decomposition
- Approx each expander by a star



Tree Cut Sparsifier via Expander Hierarchy

[Goranci-Räcke-Saranurak-Tan '21]

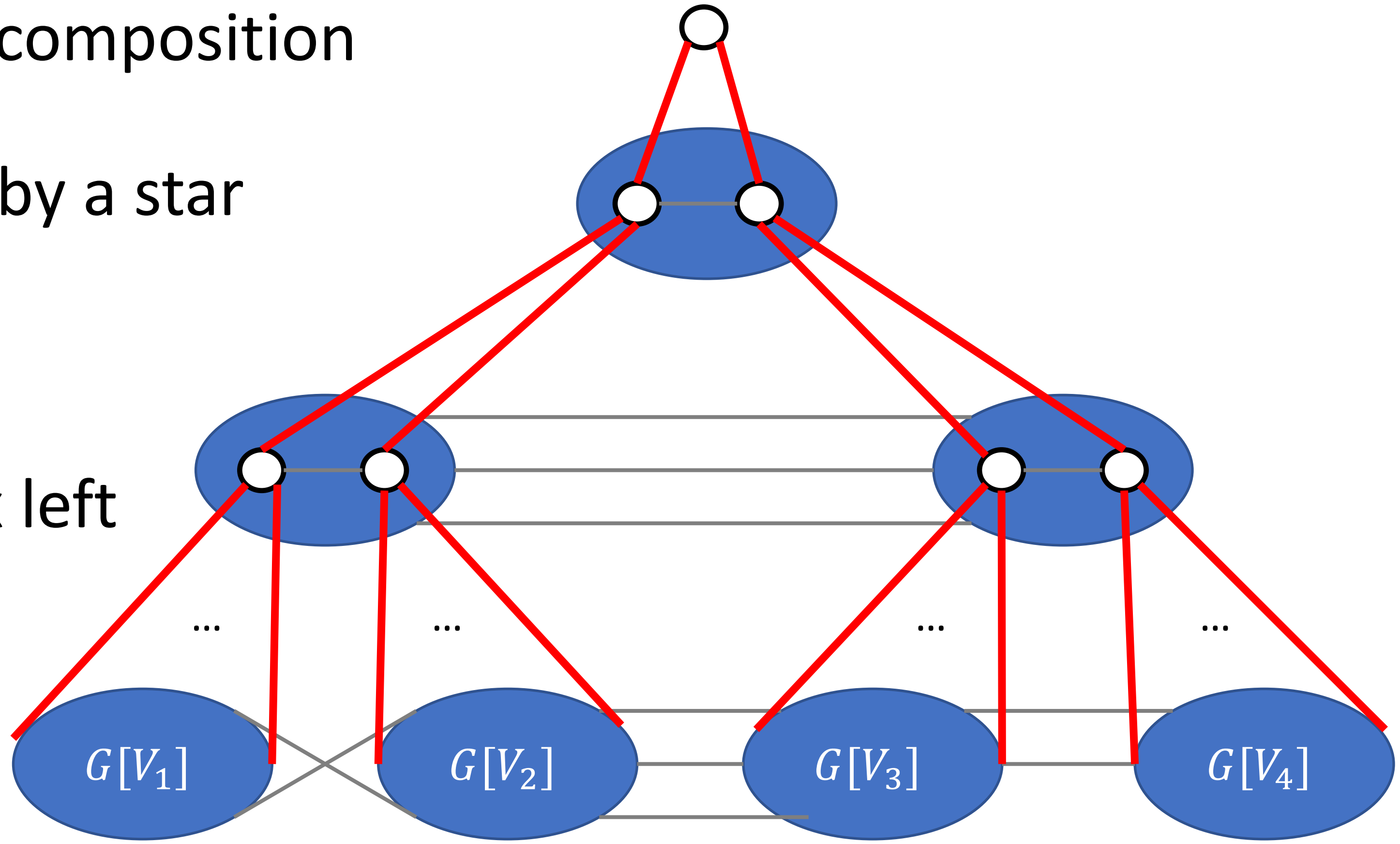
- Weighted Expander Decomposition
- Approx each expander by a star
- Contract



Tree Cut Sparsifier via Expander Hierarchy

[Goranci-Räcke-Saranurak-Tan ‘21]

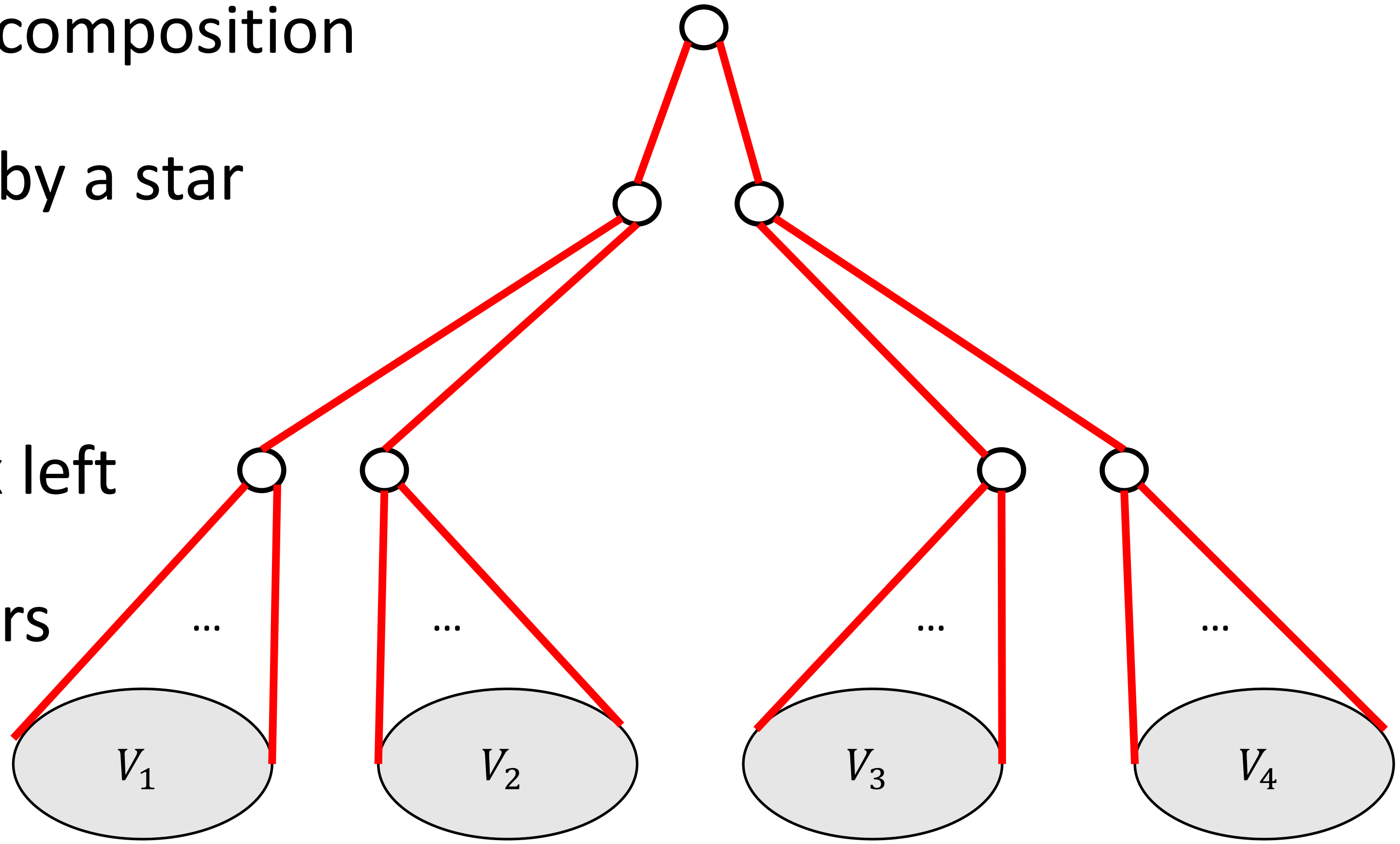
- Weighted Expander Decomposition
- Approx each expander by a star
- Contract
- Repeat until one vertex left



Tree Cut Sparsifier via Expander Hierarchy

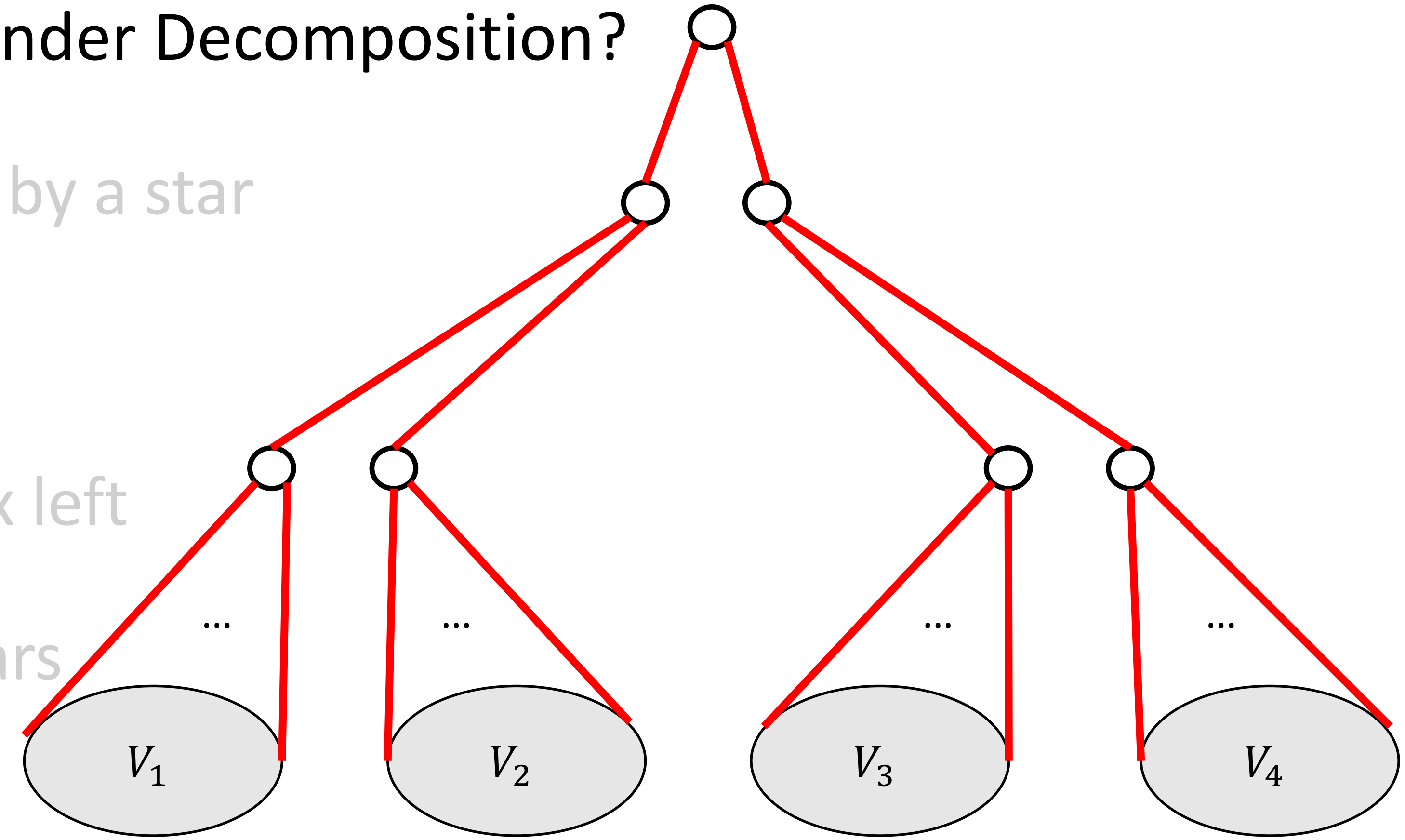
[Goranci-Räcke-Saranurak-Tan '21]

- Weighted Expander Decomposition
- Approx each expander by a star
- Contract
- Repeat until one vertex left
- Output T = union of stars
  - $m^{o(1)}$-approx



Issue

- How to Weighted Expander Decomposition?
- Approx each expander by a star
- Contract
- Repeat until one vertex left
- Output T = union of stars



Weighted Expander Decomposition

- Previous approach is hard to dynamize [Li-Saranurak '21]
- Reduce it to unweighted case via rounding

Weighted Expander Decomposition

- Initialize $G' = (V, \emptyset)$
- Let U^{total} be the total edge weight.
- For any edge e with $u(e) \geq \frac{\phi U^{total}}{m}$, add $\left\lfloor \frac{u(e) \cdot m}{\phi U^{total}} \right\rfloor$ parallel edges to G'
- For each vertex u, add $\left\lceil \frac{\deg(u) \cdot m}{\phi U^{total}} \right\rceil$ self-loops to G'
- Run unweighted expander decomposition on G' and output the partition

Weighted Expander Decomposition

Can be made dynamic

- Initialize $G' = (V, \emptyset)$
- Let U^{total} be the total edge weight.
- For any edge e with $u(e) \geq \frac{\phi U^{total}}{m}$, add $\left\lfloor \frac{u(e) \cdot m}{\phi U^{total}} \right\rfloor$ parallel edges to G'
- For each vertex u, add $\left\lceil \frac{\deg(u) \cdot m}{\phi U^{total}} \right\rceil$ self-loops to G'
- Run unweighted expander decomposition on G' and output the partition

Fully-Dynamic Weighted Expander Hierarchy

Theorem [Brand-C-Kyng-Liu-Meierhans-Probst Gutenberg-Sachdeva]

Given fully dynamic weighted graph G, there is a deterministic algorithm that maintains an expander hierarchy with $m^{o(1)}$ update time.

Fully-dynamic $m^{o(1)}$-approx

1. All-pair maxflow/mincut
2. Sparsest cuts
3. K-comm flows
4. Vertex cut sparsifier
5. …

Summary

- Decremental Min-cost flow via maintaining dual solution
- Use L1 IPM to gradually improve/certify the current solution
- Per step, solve a min-ratio cut problem
- Weighted expander hierarchy: tree approximation of graphs w.r.t. cuts

Future direction

- $T_{maxflow}/m$: $\exp(log^{7/8}m)$ -> $\exp(log^{3/4}m)$ -> polylog?
  - Deterministic: $\exp(log^{17/18}m)$ -> $\exp(log^{5/6}m)$ -> ??
- Decremental $(1+\varepsilon)$-approx. directed SSSP?
- Implication on general graph matching?

- $T_{maxflow}/m$: $\exp(log^{7/8}m)$ -> $\exp(log^{3/4}m)$ -> polylog?
  - Deterministic: $\exp(log^{17/18}m)$ -> $\exp(log^{5/6}m)$ -> ??
- Decremental $(1+\varepsilon)$-approx. directed SSSP?
- Implication on general graph matching?

Thanks!!

Jan van den Brand
Georgia Tech

Rasmus Kyng
ETH

Yang P. Liu
IAS

Simon Meierhans
ETH

Maximilian Probst Gutenberg
ETH

Sushant Sachdeva
U. Toronto