A Deterministic Almost-Linear Time Algorithm for Minimum-Cost Flow

Li Chen (Georgia Tech -> CMU)

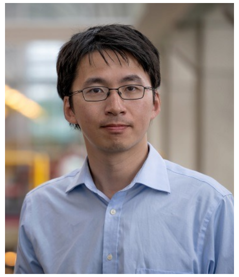
FOCS 2023

Joint work with

Jan van den Brand
Georgia Tech

Rasmus Kyng
ETH

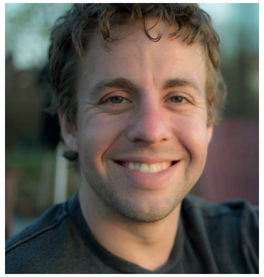
Yang P. Liu
Stanford -> IAS



Richard Peng
U Waterloo -> CMU

Maximilian Probst Gutenberg
ETH

Sushant Sachdeva
U. Toronto

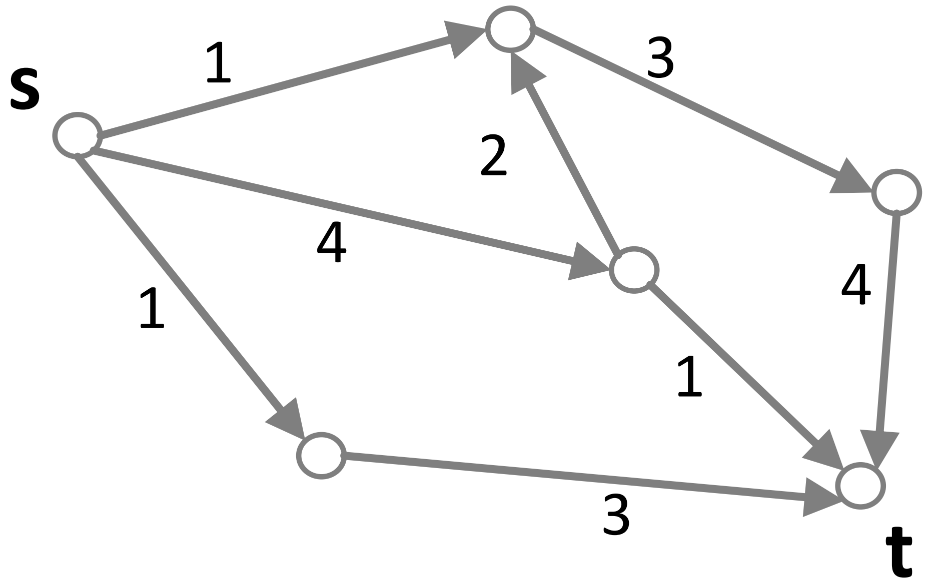


Aaron Sidford
Stanford

Maximum Flow

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

edge *capacities* $u_e \geq 0$, integer in $[0, U]$, where $U = m^{O(1)}$

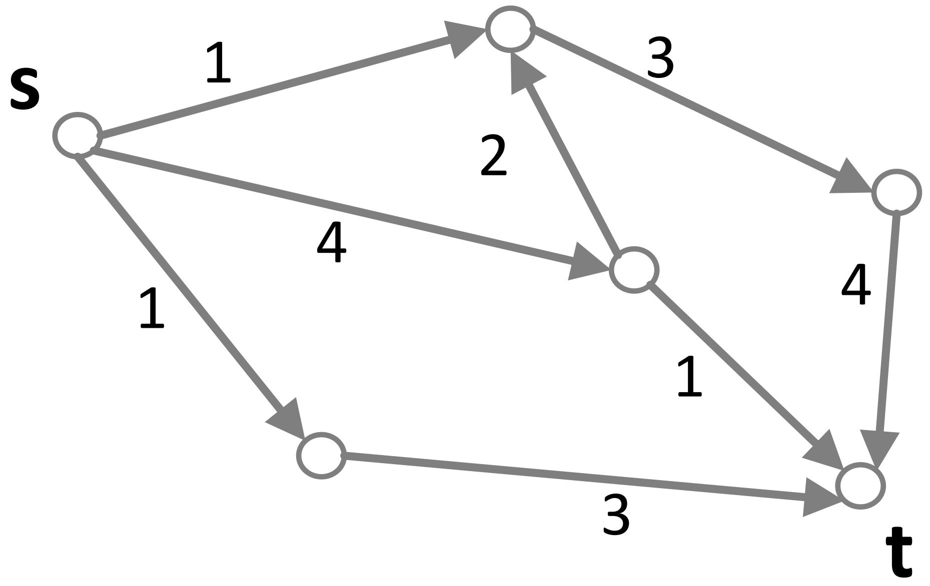


Maximum Flow

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

edge *capacities* $u_e \geq 0$, integer in $[0, U]$, where $U = m^{O(1)}$

Goal: Route maximum flow from $s \to t$, Subject to capacities u_e

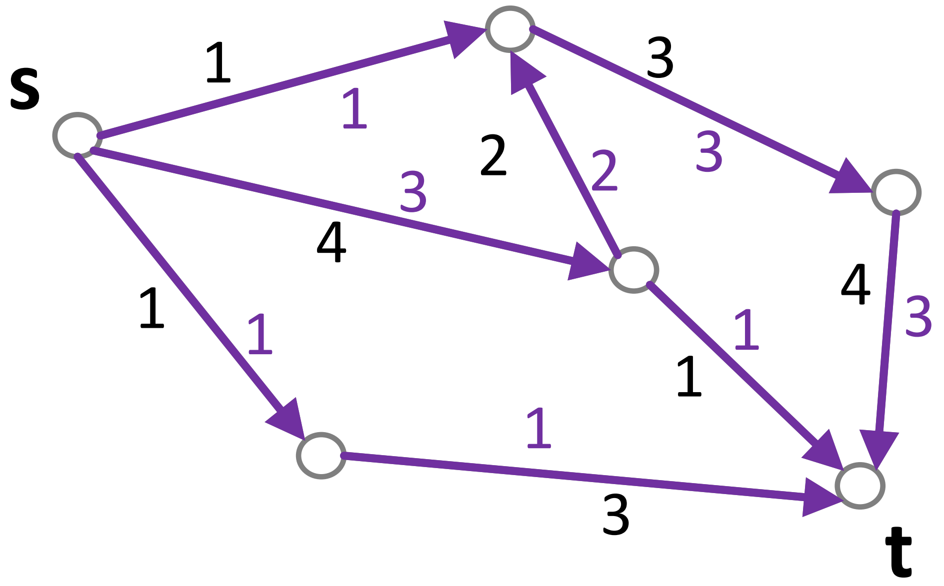


Maximum Flow

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

edge *capacities* $u_e \geq 0$, integer in $[0, U]$, where $U = m^{O(1)}$

Goal: Route maximum flow from $s \rightarrow t$, Subject to capacities u_e



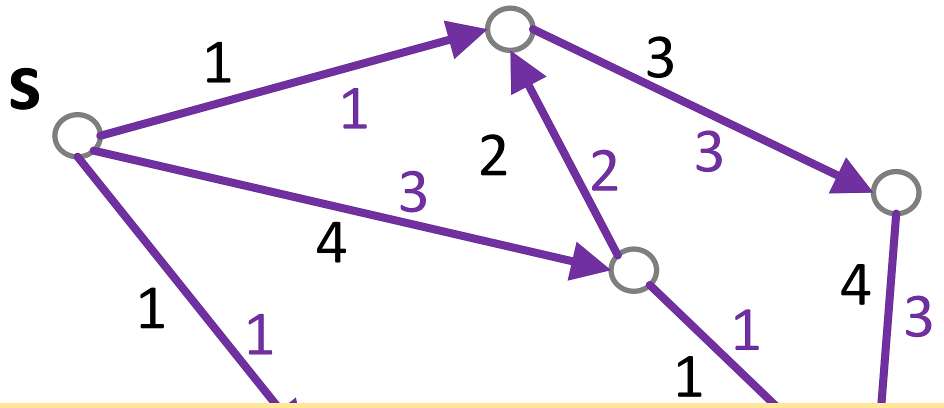
Capacity constraint:
$0 \leq f_e \leq \mathrm{u_e}$

Maximum Flow

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

edge *capacities* $u_e \geq 0$, integer in $[0, U]$, where $U = m^{O(1)}$

Goal: Route maximum flow from $s \to t$, Subject to capacities u_e



[Brand-C-Kyng-Liu-Peng-Probst Gutenberg-Sachdeva-Sidford]

Can solve max-flow in $m^{1+o(1)}$ time **deterministically**

General Convex Flow Program

$$\min_f \sum_e cost_e(f_e)$$ Flow Cost

For all edges e $$0 \leq f_e \leq u_e$$ Direction and Capacity constraints

For all vertices x $$B^T f = d$$ Net flow constraints

[Brand-C-Kyng-Liu-Peng-Probst Gutenberg-Sachdeva-Sidford]

Can solve general convex* flows in $m^{1+o(1)}$ time **deterministically**

*(assuming costs are specified as efficient self-concordant functions)

Previous Works

[Ford-Fulkerson`54]
$\tilde{O}(m^2 U)$

[Galil-Naamad`80, Sleator-Tarjan`83]
$\tilde{O}(mn \log n)$

[Goldberg-Rao`98]
$\tilde{O}(m^{3/2})$

[Daitch-Spielman`08]
$\tilde{O}(m^{3/2})$

[Lee-Sidford`14]
$\tilde{O}(mn^{1/2})$

[BLNPSSSW `20, BLLSSSW `21]
$\tilde{O}(m + n^{3/2})$

[Gao-Liu-Peng`21, BGJLLPS`21]
$\tilde{O}(m^{3/2-1/58})$

[CKLPPS`22]
$m^{1+o(1)}$

[This work]
$m^{1+o(1)}$

- Combinatorial outer and inner algorithms
- Deterministic

- IPM based outer algorithm
- Randomized

Derandomize [CKLPPS`22] with Deterministic Dynamic "Stable" Min-ratio cycle

OPEN: derandomize these

- Derandomize via Deterministic Laplacian Solver [CGLNPS`20]

L1 IPM

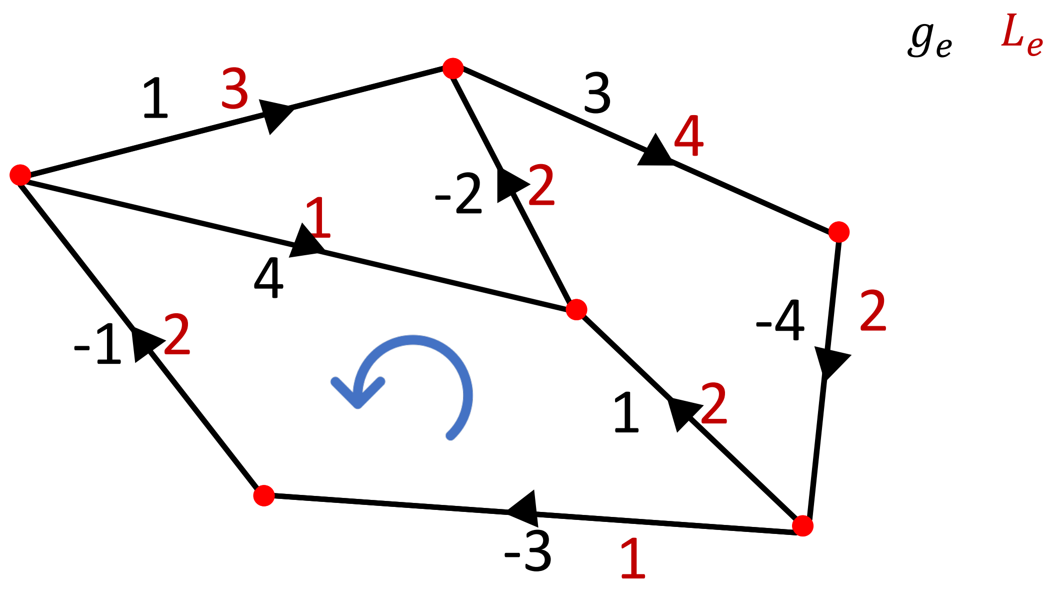
[C-Kyng-Liu-Peng-Probst Gutenberg-Sachdeva `22]

There is an Interior Point Method (IPM) for max-flow such that

1. $m^{1+o(1)}$ iterations, each subproblem a min-ratio cycle $\min_{B^\top \Delta = 0} \frac{g^\top \Delta}{\|L\Delta\|_1}$
2. a $m^{o(1)}$-approximate solution suffices at each iteration
3. At most $m^{1+o(1)}$ total changes to g_e, L_e over all edges e
4. $f - f^*$ has a small ratio and $|L(f - f^*)|$ changes slowly.

Min-ratio Cycle

$$\min_{B^\top \Delta = 0} \frac{g^\top \Delta}{\|L\Delta\|_1}$$

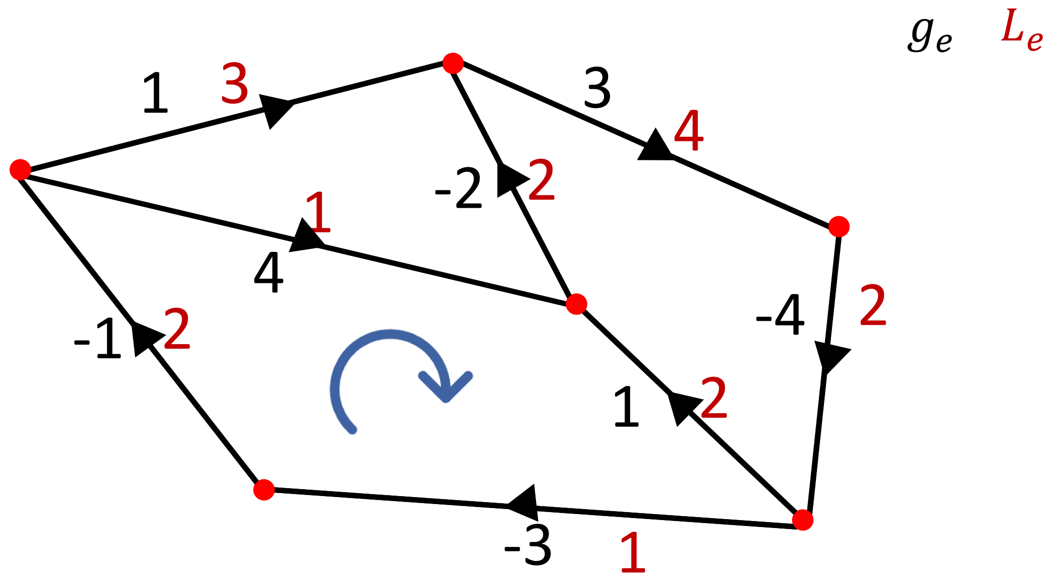


$$\|L\Delta\|_1 = 1 + 2 + 1 + 2 = 6$$

$$g^\top \Delta = -4 + 1 + 3 + 1 = 1$$

Min-ratio Cycle

$$\min_{B^\top \Delta = 0} \frac{g^\top \Delta}{\|L\Delta\|_1}$$



Edges and lengths are undirected

Gradient has a direction

$$\|L\Delta\|_1 = 1 + 2 + 1 + 2 = 6$$

$$g^\top \Delta = 4 - 1 - 3 - 1 = -1$$

Optimal solution is a simple cycle with ratio < 0

Deterministic Dynamic Stable Min-ratio Cycle

[Brand-C-Kyng-Liu-Peng-Probst Gutenberg-Sachdeva-Sidford]

Assuming the min-ratio cycles change slowly,

A deterministic data-structure that supports in $m^{o(1)}$ amortized time

1. Update g_e, L_e for an edge e
2. Return a $m^{o(1)}$-approximate min-ratio cycle
3. Route flow along such a cycle

Overall Algorithm

Initialize a dynamic stable min-ratio cycle data structure

For $t \leftarrow 1, \ldots, m^{1+o(1)}$ iterations

Update gradients g_e and lengths L_e

Update the data structure

Identify a circulation Δ approximately minimizing $\frac{g^\top \Delta}{\|L\Delta\|_1}$,

$f^{(t)} \leftarrow f^{(t-1)} + \alpha\Delta$

Output final flow $f^{(final)}$

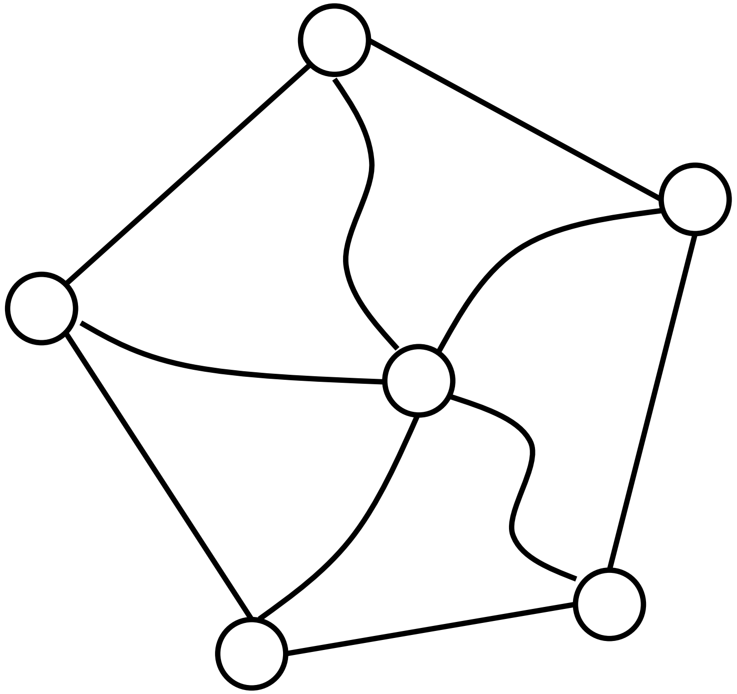
Approx min-ratio cycle via tree embeddings

Goal: Approximately solve $\min_{B^\top \Delta = 0} \frac{g^\top \Delta}{\|L\Delta\|_1}$

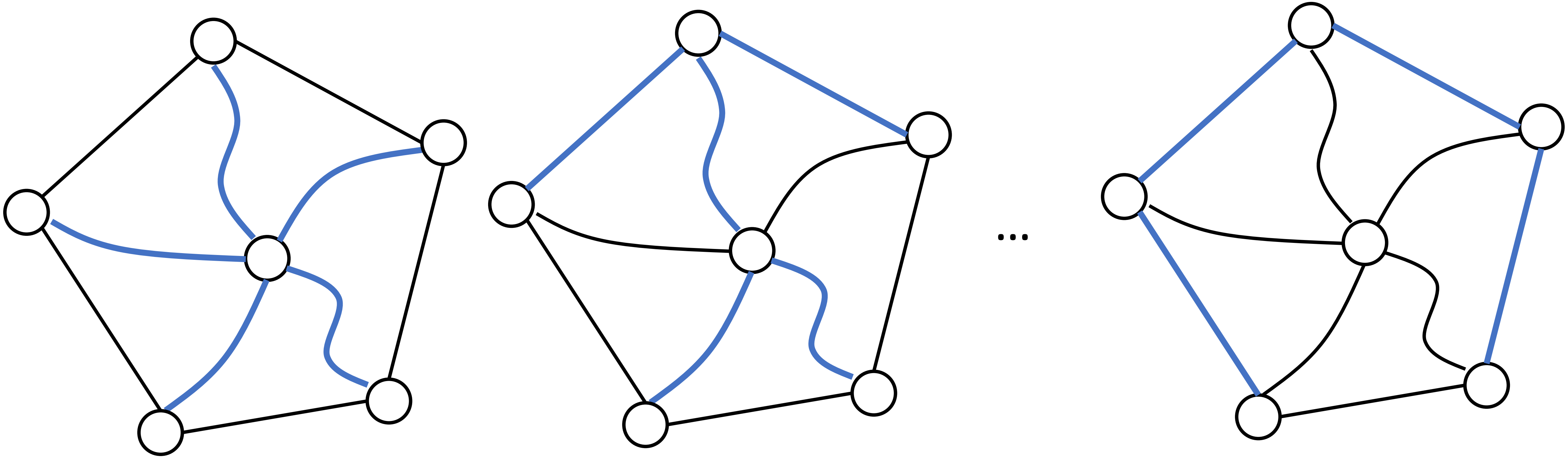
Algorithm:

1. Build m "low stretch spanning trees" $T_1, T_2, \ldots, T_m$ s.t. every edge in G has average stretch $\tilde{O}(1)$ — $\tilde{O}(m^2)$ time
   [Räcke`08, Abraham-Neiman `19]
2. Return the best "tree cycle" among $T_1, T_2, \ldots, T_m$ (one off-tree edge + tree path) — $\tilde{O}(m^2)$ time
   a.k.a. fundamental cycles
   Denoted $\mathrm{cycle}_{T_i}(e)$

Claim: Some $\mathrm{cycle}_{T_i}(e)$ is an $\tilde{\mathrm{O}}(1)$-approx

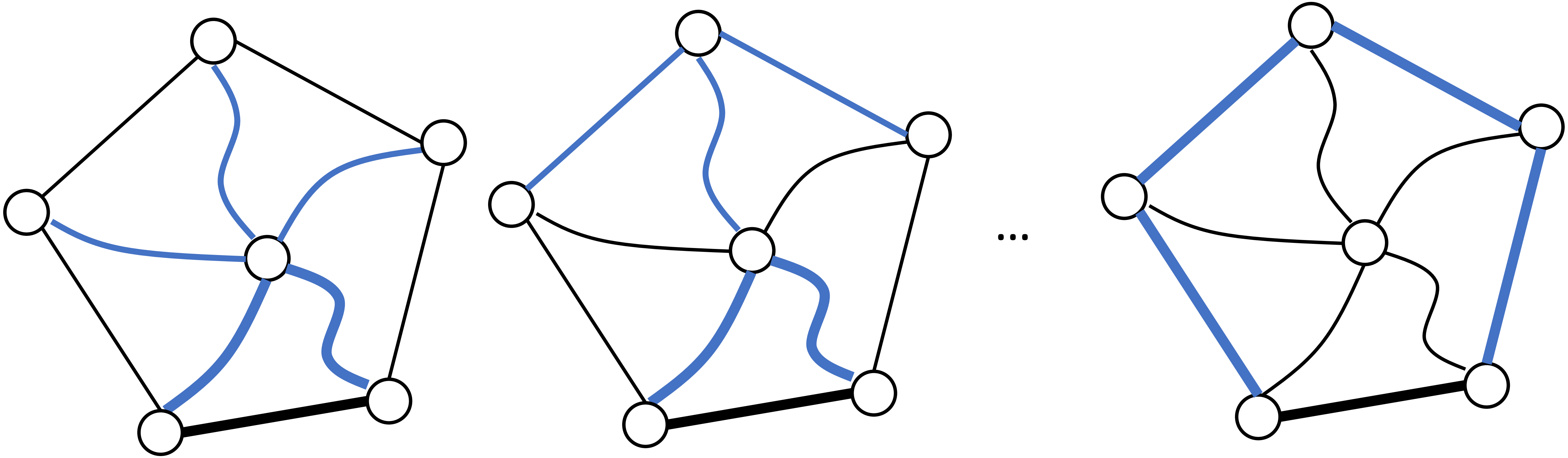


Claim: Some $\mathrm{cycle}_{T_i}(e)$ is an $\tilde{O}(1)$-approx



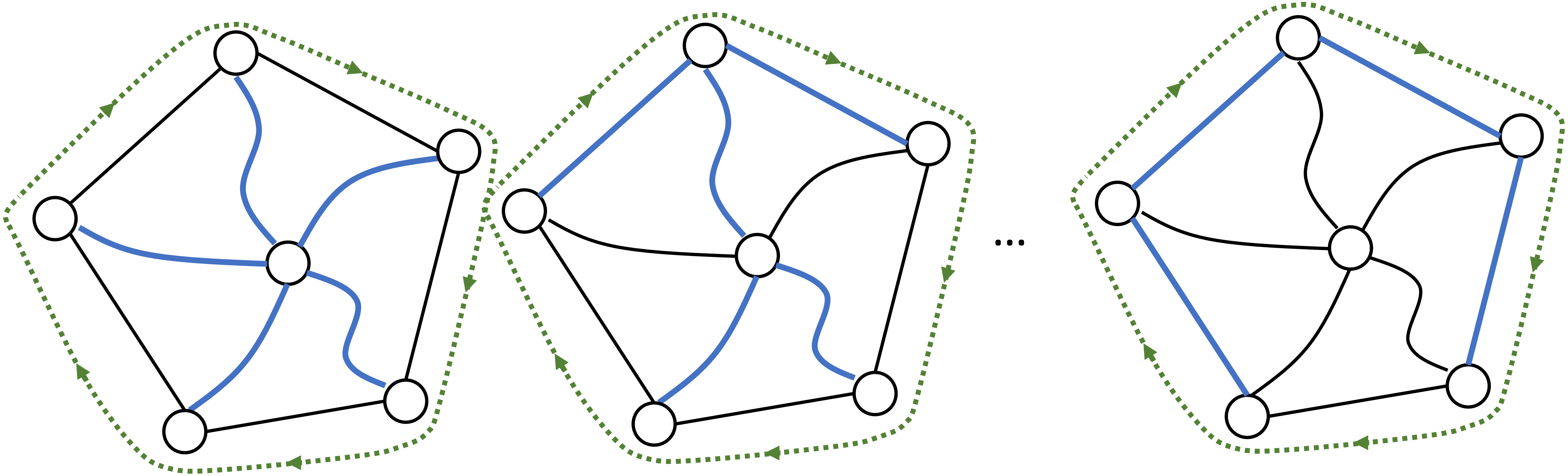
Build m low stretch trees $T_1, T_2, \ldots, T_m$

Claim: Some $\mathrm{cycle}_{T_i}(e)$ is an $\tilde{O}(1)$-approx



$$\frac{1}{m}\sum_i L(\mathrm{cycle}_{T_i}(e)) \leq \tilde{O}(1) L_e$$

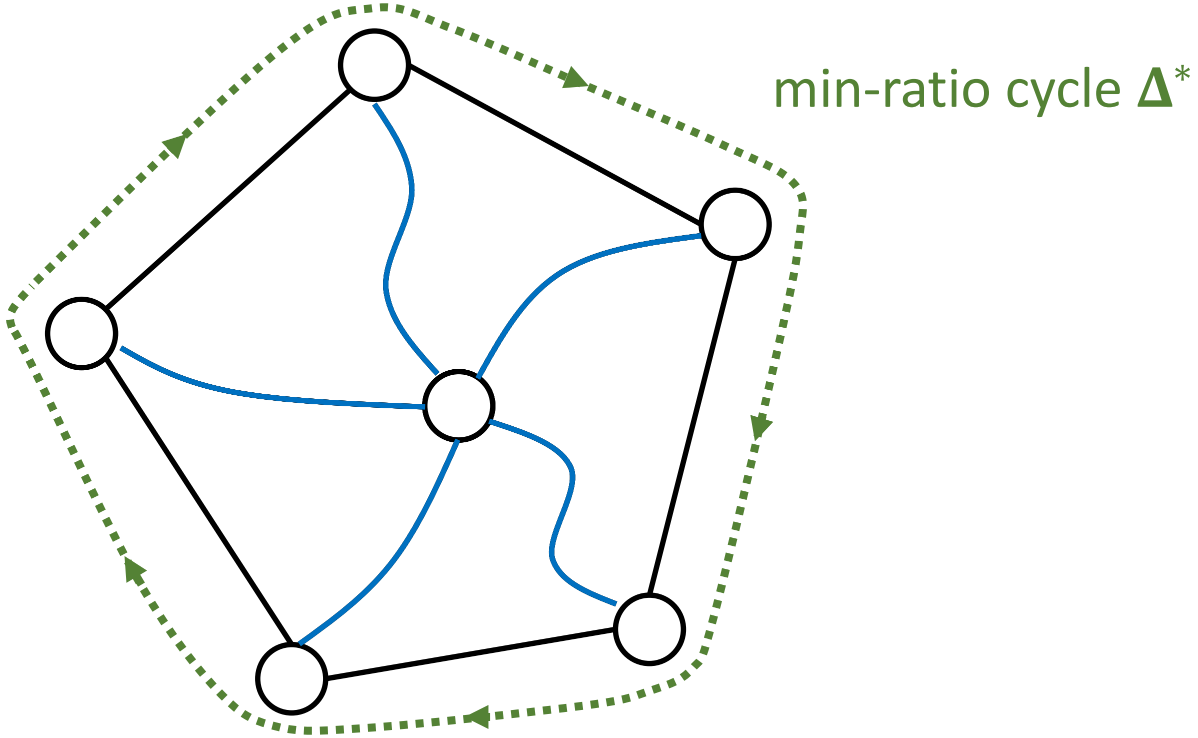
Claim: Some $\mathbf{cycle}_{T_i}(e)$ is an $\tilde{\mathbf{O}}(\mathbf{1})$-approx



min-ratio cycle $\boldsymbol{\Delta}^*$

$$\frac{1}{m}\sum_i \sum_{e\in\Delta^*} L(\text{cycle}_{T_i}(e)) \le \tilde{O}(1)\cdot \|L\Delta^\star\|_1$$

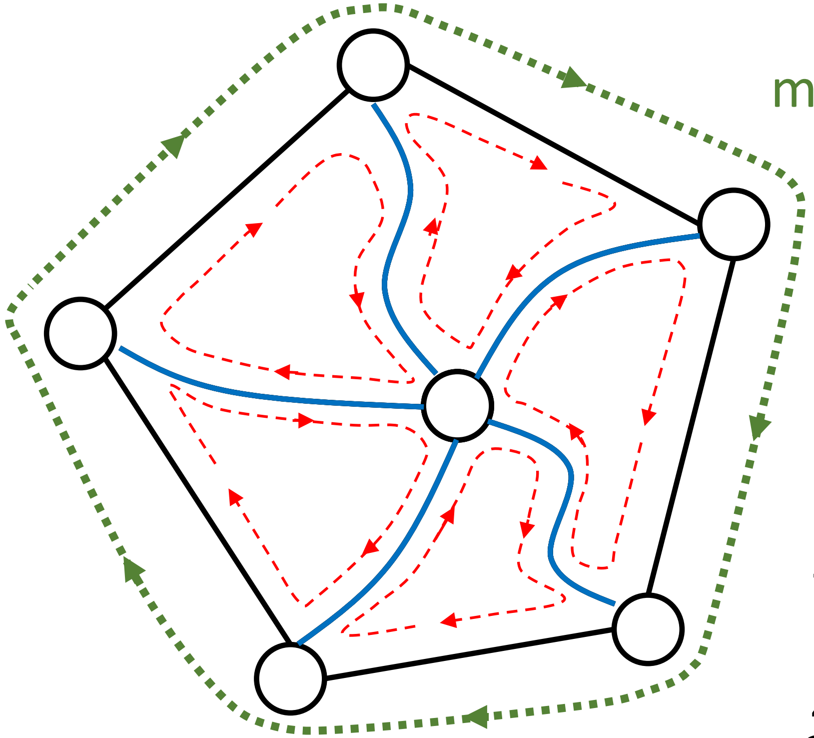
Claim: Some $\mathrm{cycle}_{T_i}(e)$ is an $\tilde{O}(1)$-approx



One of the tree T_i
(actually, half of them)

$$\sum_{e \in \Delta^\star} L\left(\mathrm{cycle}_{T_i}(e)\right) \leq \tilde{O}(1) \cdot \|L\Delta^\star\|_1$$

Claim: Some $\text{cycle}_{T_i}(e)$ is an $\tilde{O}(1)$-approx

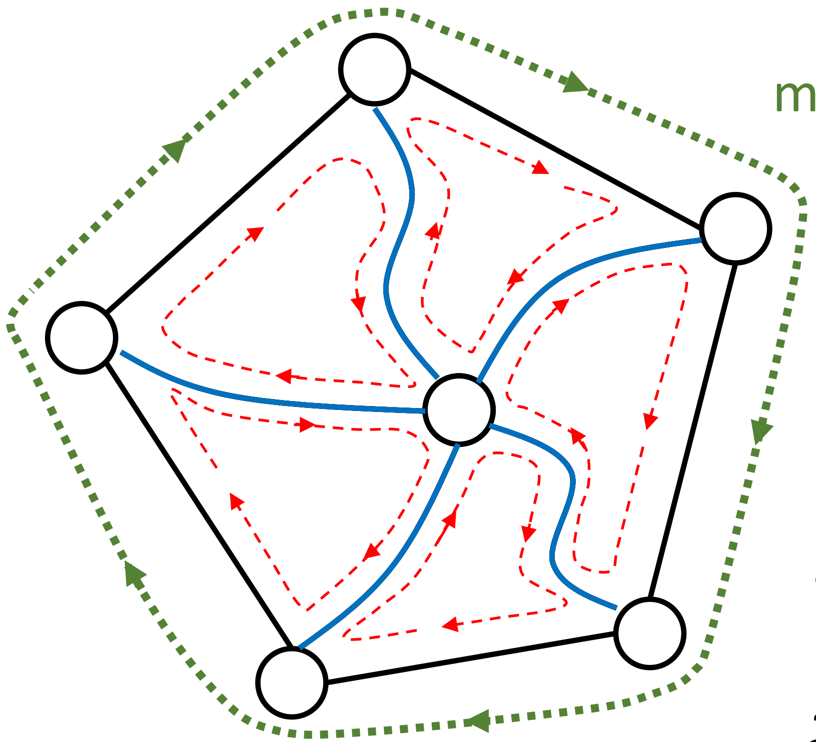


$$\sum_{e \in \Delta^\star} g^\top \text{cycle}_{T_i}(e)$$

One of the tree T_i
(actually, half of them)

$$\sum_{e \in \Delta^\star} L\left(\text{cycle}_{T_i}(e)\right) \leq \tilde{O}(1) \cdot \|L\Delta^\star\|_1$$

Claim: Some $\text{cycle}_{T_i}(e)$ is an $\tilde{O}(1)$-approx



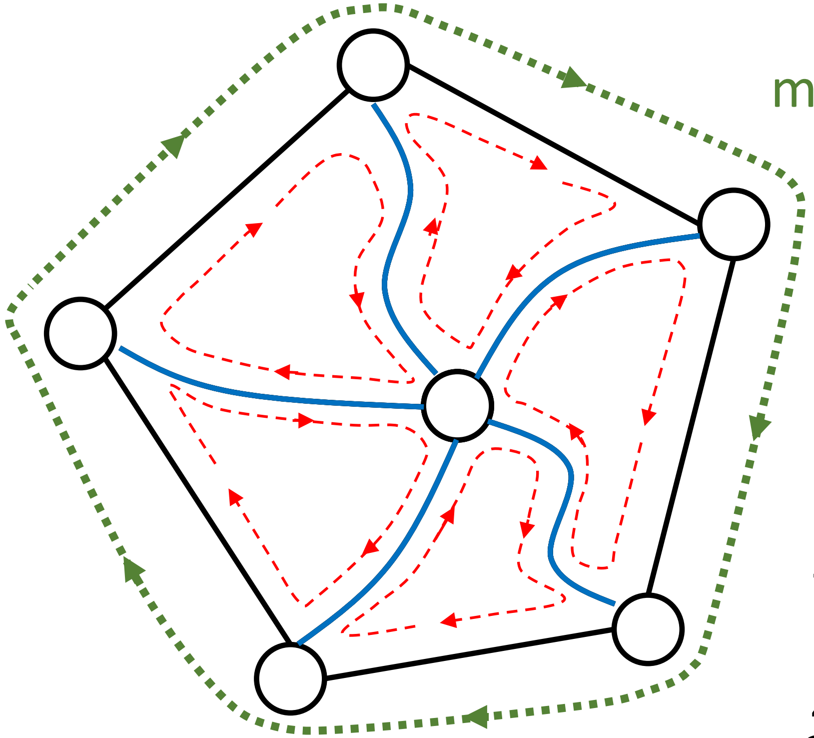
One of the tree T_i
(actually, half of them)

$$\sum_{e \in \Delta^\star} L\left(\text{cycle}_{T_i}(e)\right) \leq \tilde{O}(1) \cdot \|L\Delta^\star\|_1$$

$$\sum_{e \in \Delta^\star} g^\top \text{cycle}_{T_i}(e)$$

$$= g^\top \sum_{e \in \Delta^\star} \text{cycle}_{T_i}(e)$$

Claim: Some $\text{cycle}_{T_i}(e)$ is an $\tilde{O}(1)$-approx



One of the tree T_i
(actually, half of them)

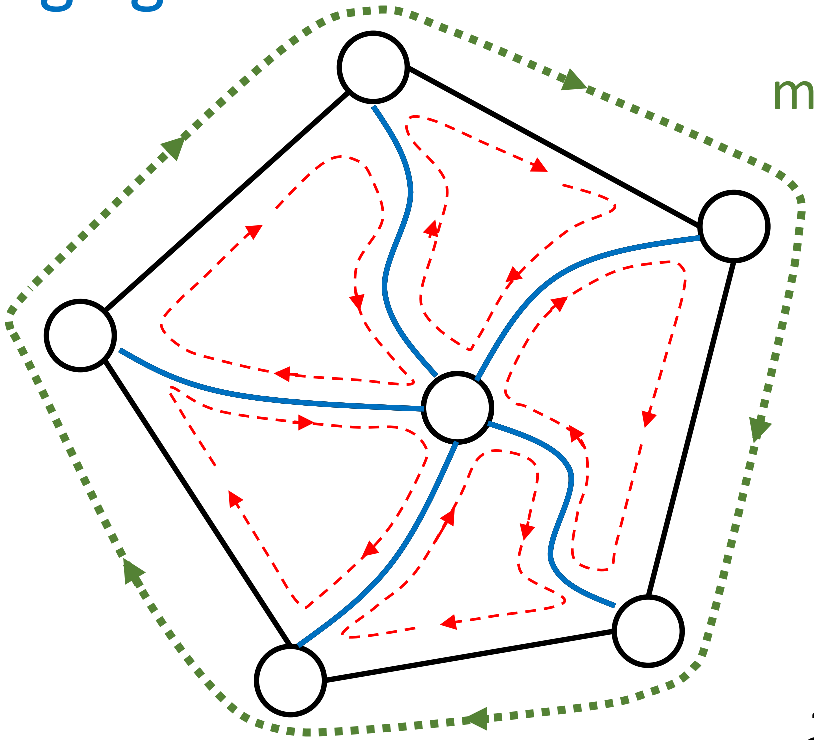
$$\sum_{e \in \Delta^\star} L\left(\text{cycle}_{T_i}(e)\right) \leq \tilde{O}(1) \cdot \|L\Delta^\star\|_1$$

$$\sum_{e \in \Delta^\star} g^\top \text{cycle}_{T_i}(e)$$

$$= g^\top \sum_{e \in \Delta^\star} \text{cycle}_{T_i}(e) = g^\top \Delta^*$$

Claim: Some $\text{cycle}_{T_i}(e)$ is an $\tilde{O}(1)$-approx

Claim follows by averaging.



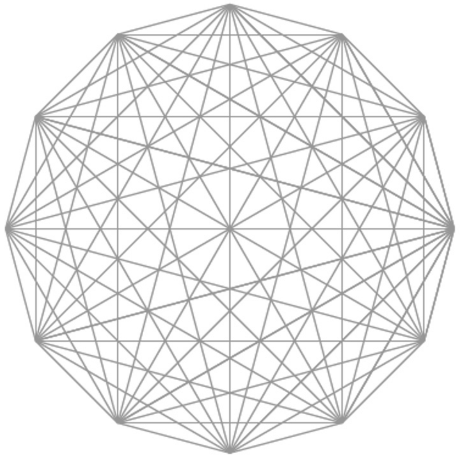
One of the tree T_i
(actually, half of them)

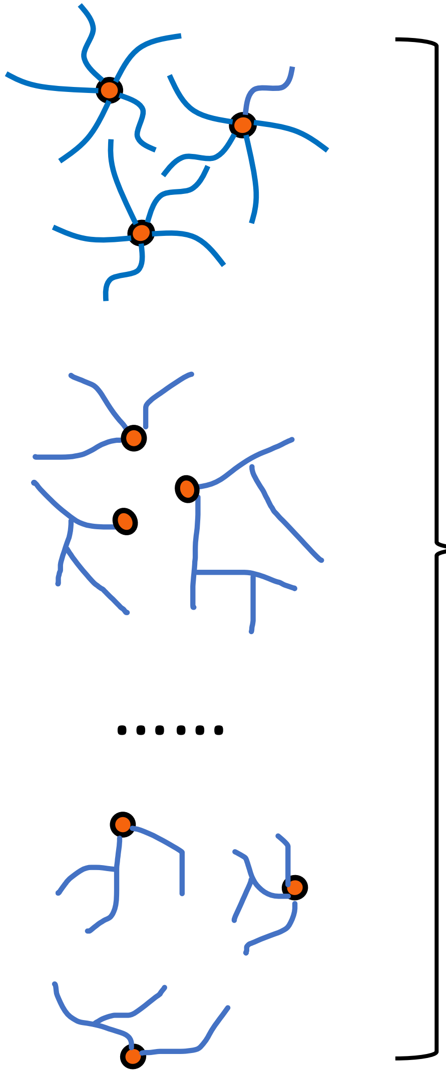
$$\sum_{e \in \Delta^\star} L\left(\text{cycle}_{T_i}(e)\right) \leq \tilde{O}(1) \cdot \|L\Delta^\star\|_1$$

$$\sum_{e \in \Delta^\star} g^\top \text{cycle}_{T_i}(e)$$

$$= g^\top \sum_{e \in \Delta^\star} \text{cycle}_{T_i}(e) = g^\top \Delta^*$$

Dynamic Min Ratio Cycle

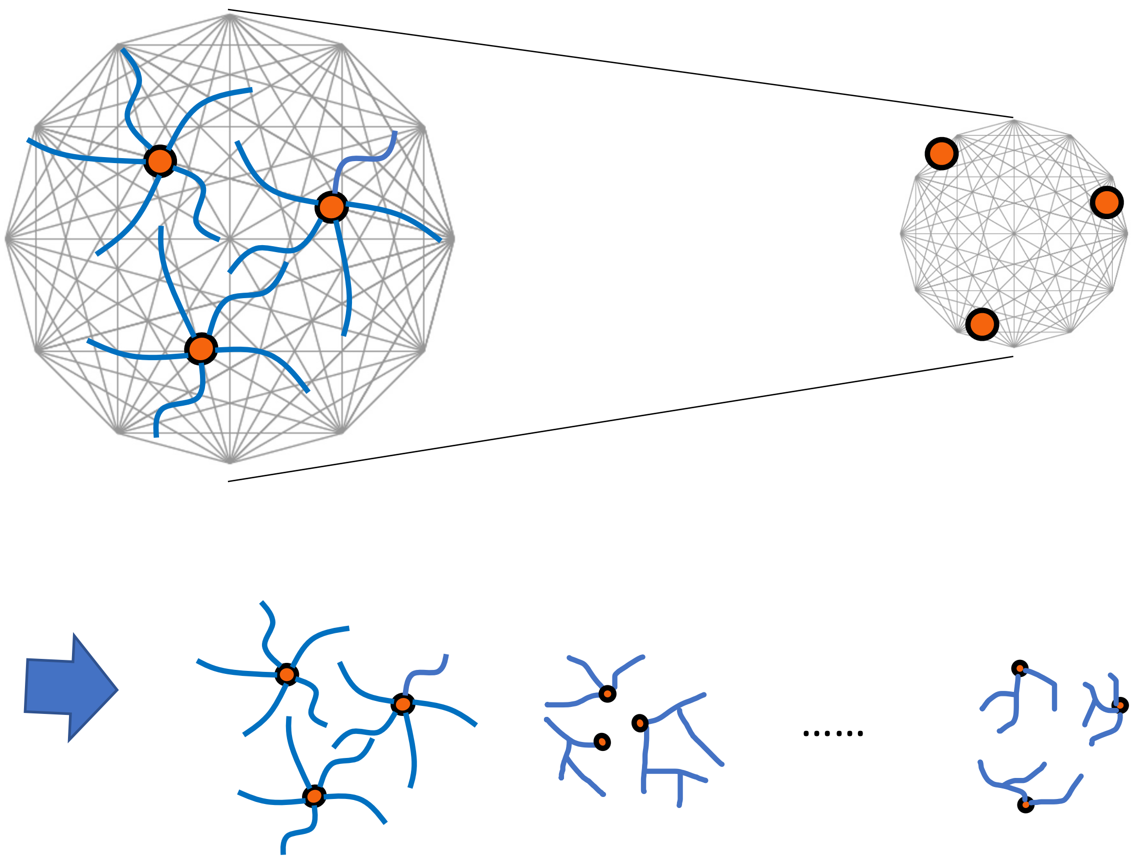




	"Full Trees"
• K partial trees	m
• Partial tree on $\mathrm{m} - \mathrm{m}/K$ edges	m
• Recurse on the rest m/K vertices	1
• Maintain up to m/K upd, then rebuild	

Rooted Forest F
≈"partial tree"

Dynamic Min Ratio Cycle

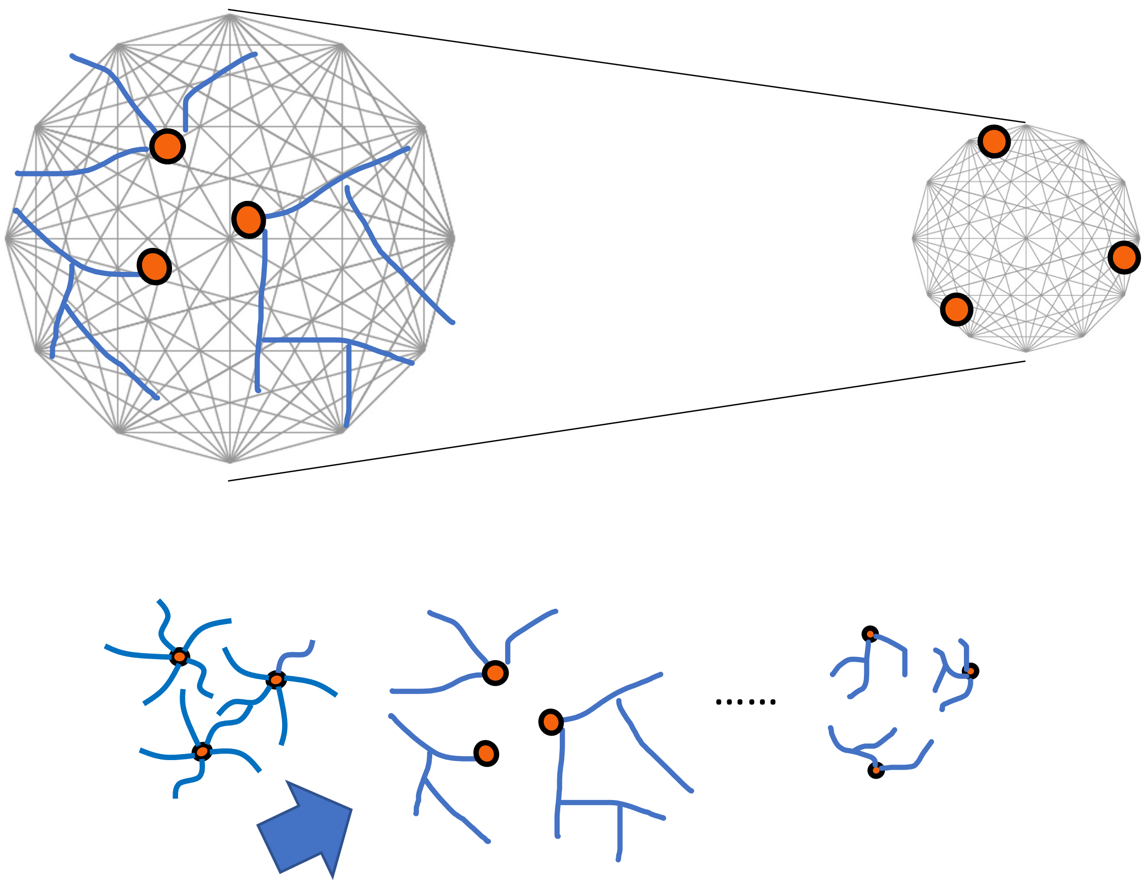


- Pick 1 to recursively build the DS

Rooted Forest F

≈"partial tree"

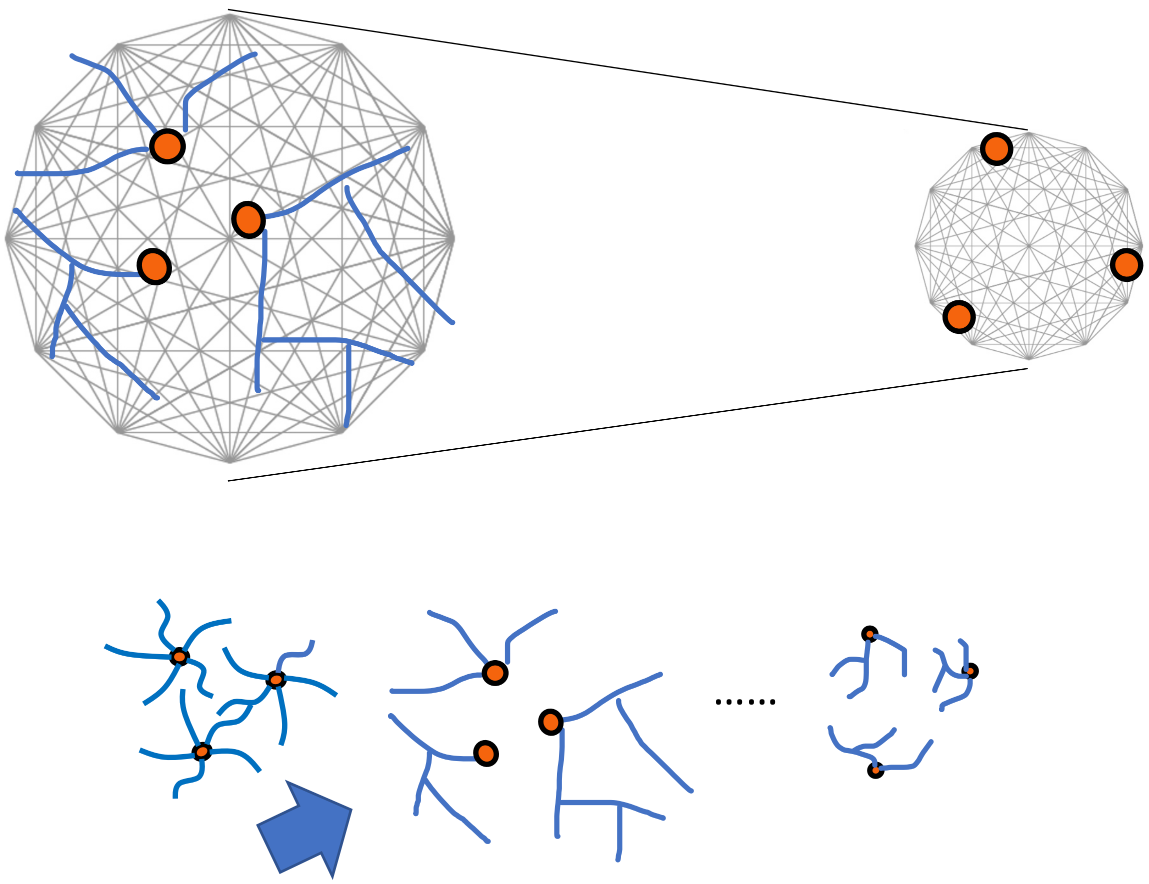
Dynamic Min Ratio Cycle



- Pick 1 to recursively build the DS
- If fail, switch to the next partial tree and rebuild

Rooted Forest F
≈"partial tree"

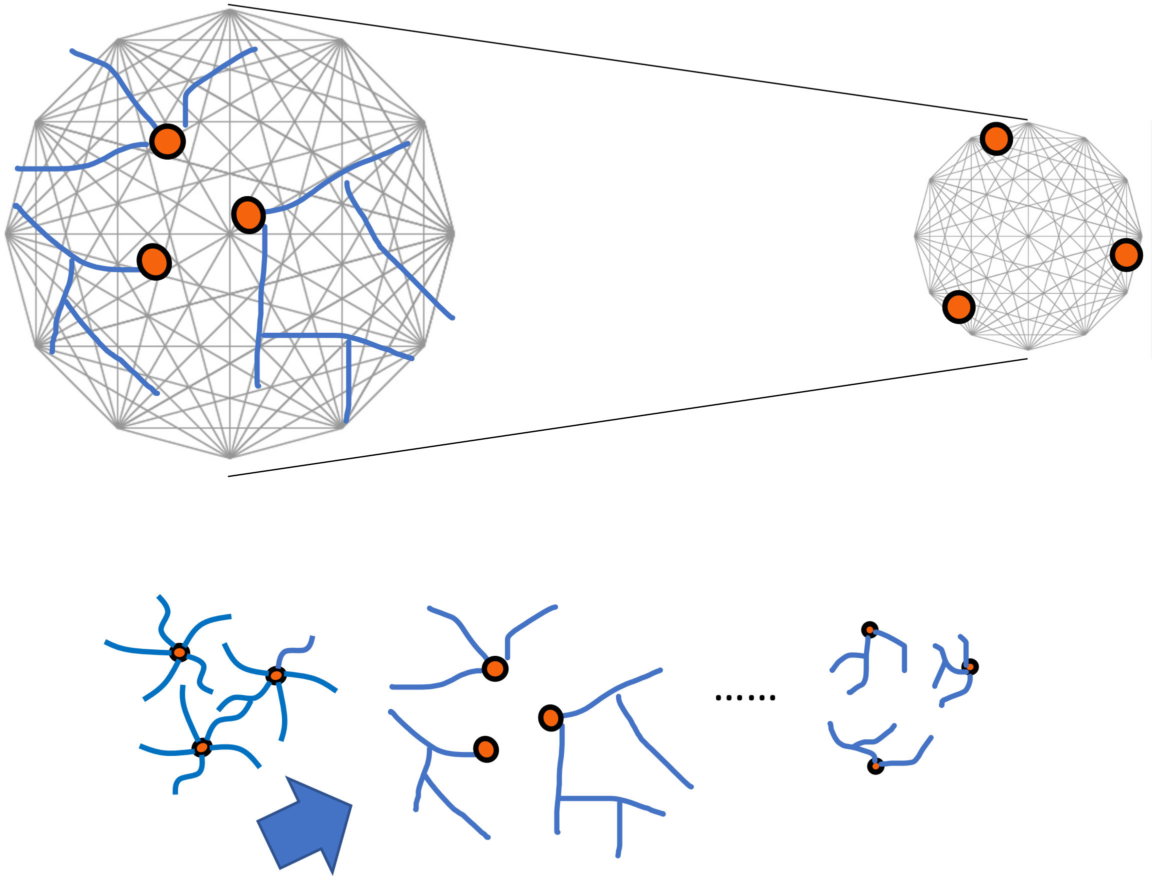
Dynamic Min Ratio Cycle



- Pick 1 to recursively build the DS
- If fail, switch to the next partial tree and rebuild
- What's the overall cost?

Rooted Forest F
≈"partial tree"

Handling Partial Tree Failures



- One of the K partial trees works
  # of switches ≤ K
- m iterations, $\Omega(Km)$ switches
  $\Omega(Km^2)$ run time
- Stability of IPM ensures
  $\tilde{O}(K)$ total switches
- $m^{1+o(1)}$ runtime by $K = m^{o(1)}$

Conclusion

- Maxflow, Min-cost flow in deterministic $m^{1+o(1)}$-time
- Replace sampling by total search
- Low cost due to IPM stability
- Deterministic dynamic spanner
- Deterministic dynamic low stretch tree

Open Questions

- Deterministic Dynamic Min-Ratio Cycle?
- $m^{1+o(1)}$-time to $m\ polylog(m)$–time?
- Can we improve k-commodity flow?
- General Graph Matching in n^2 Time?
- Dynamic maxflow? Incremental/decremental?

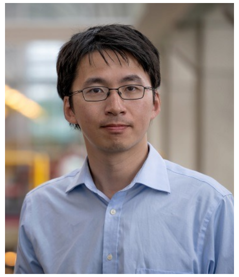
- Deterministic Dynamic Min-Ratio Cycle?
- $m^{1+o(1)}$-time to $m\, polylog(m)$–time?
- Can we improve k-commodity flow?
- General Graph Matching in n^2 Time?
- Dynamic maxflow? Incremental/decremental?

Thanks!!

Jan van den Brand
Georgia Tech

Rasmus Kyng
ETH

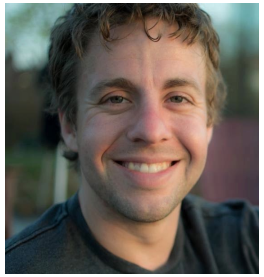
Yang P. Liu
Stanford -> IAS



Richard Peng
U Waterloo -> CMU

Maximilian Probst Gutenberg
ETH

Sushant Sachdeva
U. Toronto



Aaron Sidford
Stanford