The maximum edge-disjoint paths problem in bounded treewidth graphs

Chandra Chekuri, *Guyslain Naves*, Bruce Shepherd

Bellairs workshop, April 2011

McGill

Maximum Edge-Disjoint Paths problem

(MEDP for short)

Input:
- a graph G,
- capacities $c : E(G) \to \mathbb{N}$,
- pairs (s_i, t_i) of *commodities*, with weights w_i.

Output:
- $\mathcal{P}$, family of (s_i, t_i)-paths in G,
- at most $c(e)$ paths of $\mathcal{P}$ contain e $(e \in E(G))$.

Goal: Maximize $\sum_{i \in I_{\mathcal{P}}} w_i$,

where $I_{\mathcal{P}} = \{i \ : \ \text{there is an } (s_i, t_i)\text{-path in } \mathcal{P}\}$.

McGill

General results

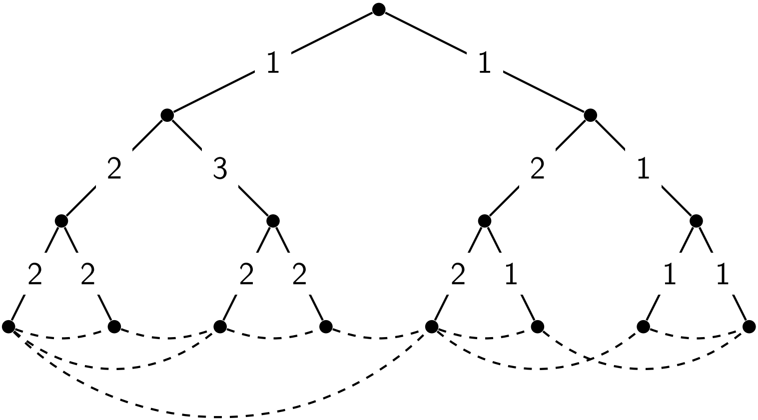
MEDP...

- is APX-hard, even in trees (Garg, Vazirani, Yannakakis, 1997),
- is hard to approximate within $\Omega(m^{\frac{1}{2}-\varepsilon})$ in directed graphs (Guruswami, Khanna, Rajaraman, Shepherd, Yannakakis, 1999),
- is hard to approximate within $\Omega(\log^{1/2-\varepsilon} n)$ in undirected graphs (Andrews, Chuzhoy, Khanna, Zhang, 2005),
- has $\Omega(\sqrt{n})$ integrality gap, for the natural LP (Guruswami,...), $O(\sqrt{n})$ in undirected graphs (Chekuri, Khanna, Shepherd, 2005)
- has approximation ratio $O(\sqrt{m})$ (Kleinberg, 1996).

McGill

2-approximation in trees

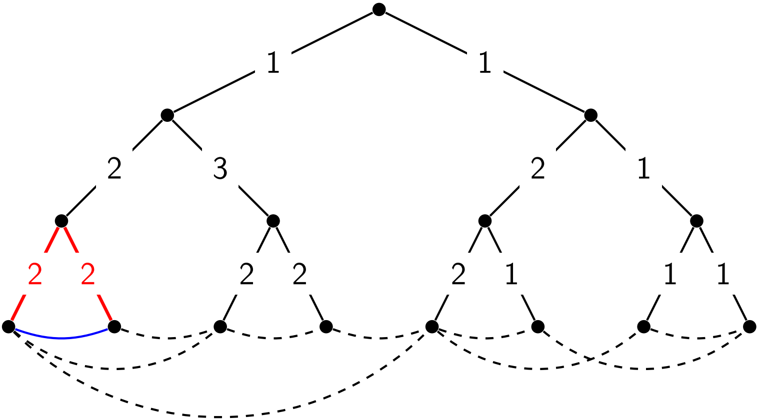
(Garg, Vazirani, Yannakakis, 1997)



Idea: route the deepest possible demand.

2-approximation in trees

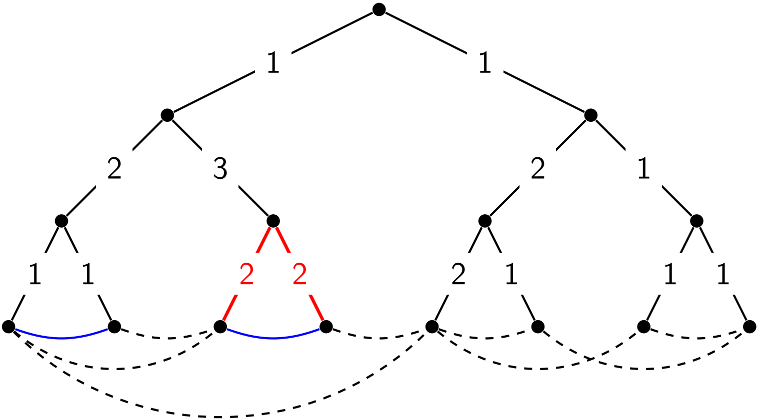
(Garg, Vazirani, Yannakakis, 1997)



Idea: route the deepest possible demand.

2-approximation in trees

(Garg, Vazirani, Yannakakis, 1997)

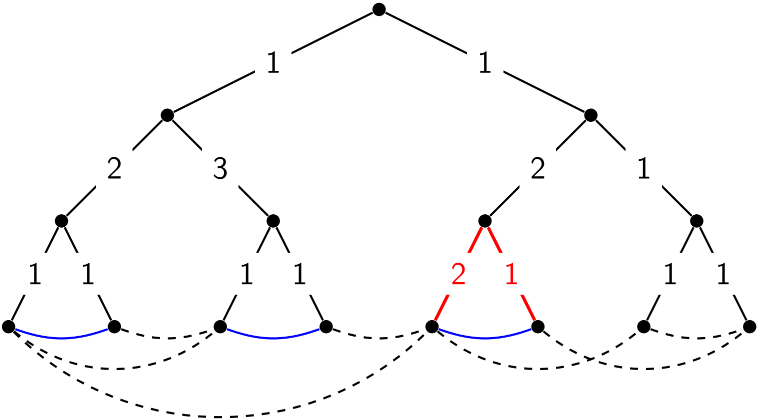


Idea: route the deepest possible demand.

McGill

2-approximation in trees

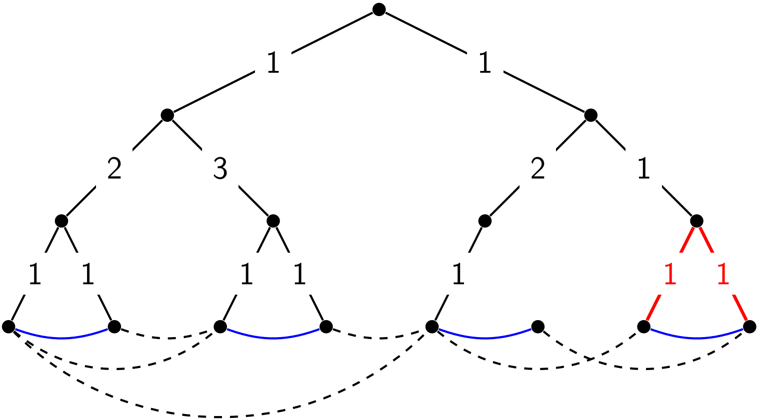
(Garg, Vazirani, Yannakakis, 1997)



Idea: route the deepest possible demand.

2-approximation in trees

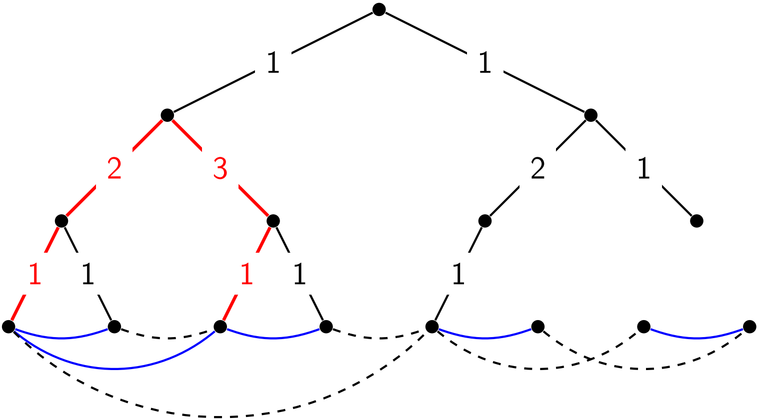
(Garg, Vazirani, Yannakakis, 1997)



Idea: route the deepest possible demand.

2-approximation in trees

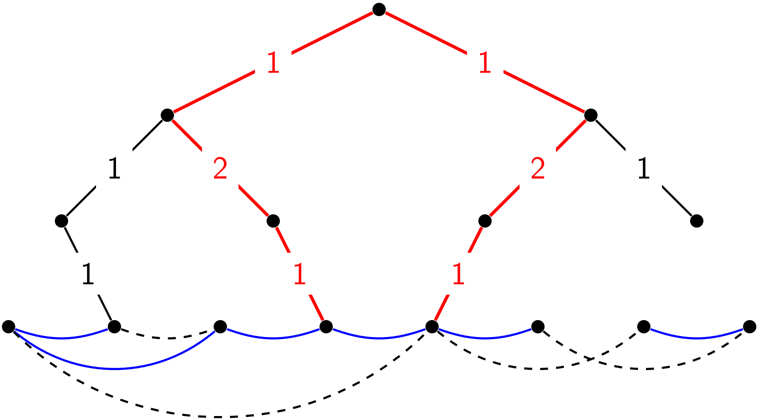
(Garg, Vazirani, Yannakakis, 1997)



Idea: route the deepest possible demand.

2-approximation in trees

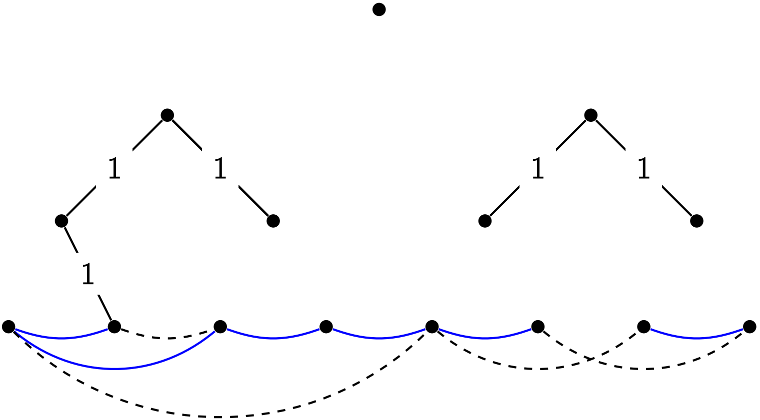
(Garg, Vazirani, Yannakakis, 1997)



Idea: route the deepest possible demand.

2-approximation in trees

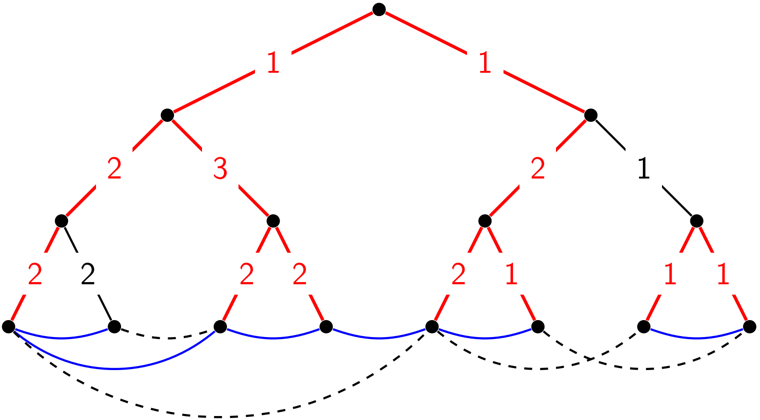
(Garg, Vazirani, Yannakakis, 1997)



Idea: route the deepest possible demand.

2-approximation in trees

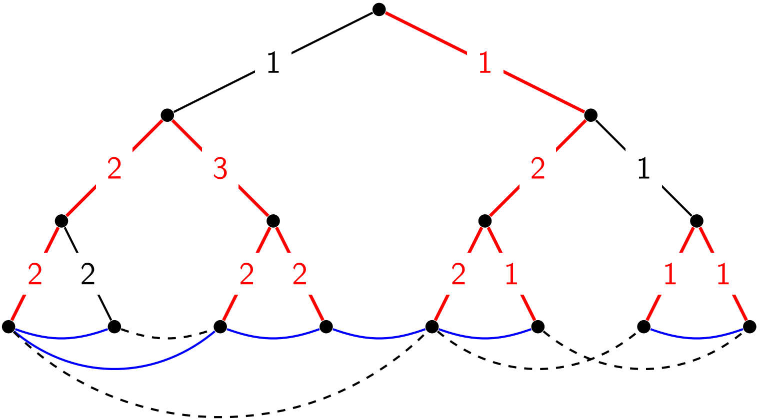
(Garg, Vazirani, Yannakakis, 1997)



Idea: route the deepest possible demand.

2-approximation in trees

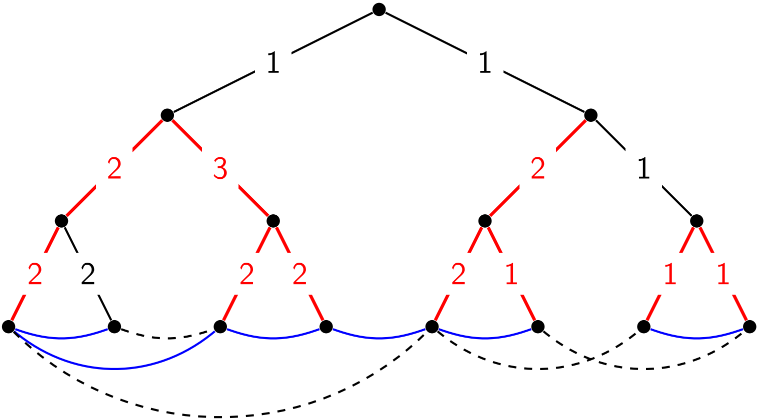
(Garg, Vazirani, Yannakakis, 1997)



Idea: route the deepest possible demand.

2-approximation in trees

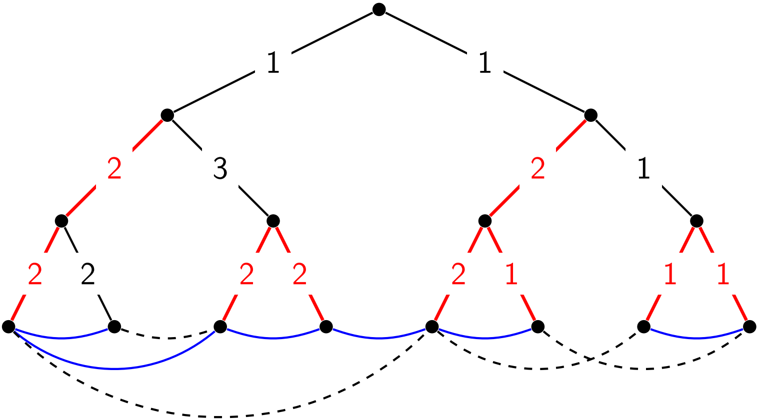
(Garg, Vazirani, Yannakakis, 1997)



Idea: route the deepest possible demand.

2-approximation in trees

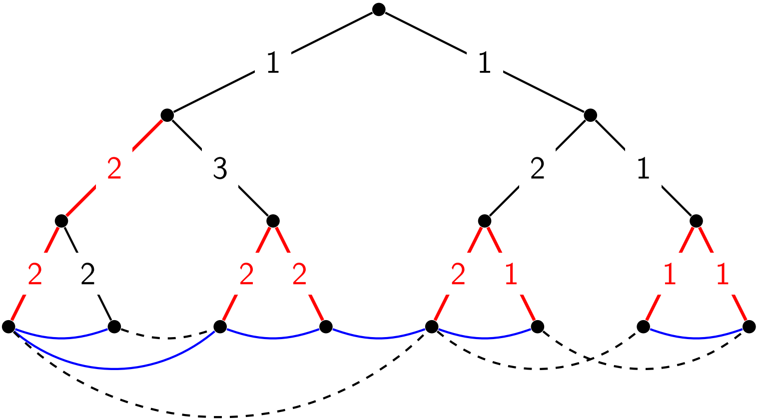
(Garg, Vazirani, Yannakakis, 1997)



Idea: route the deepest possible demand.

2-approximation in trees

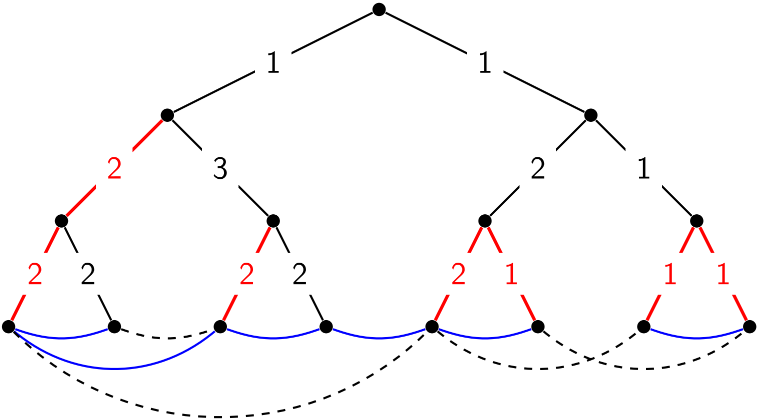
(Garg, Vazirani, Yannakakis, 1997)



Idea: route the deepest possible demand.

2-approximation in trees

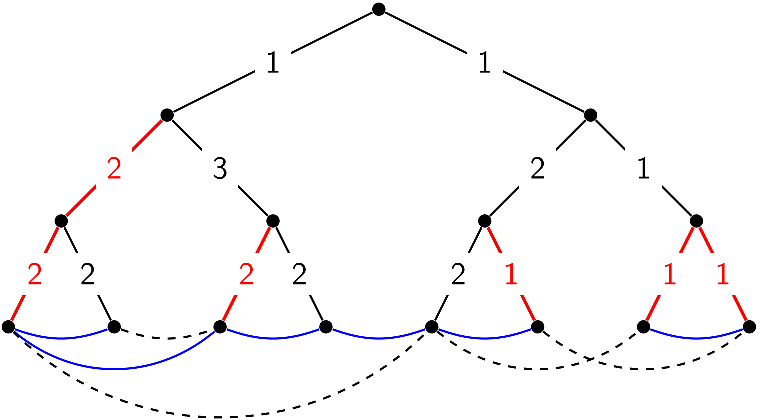
(Garg, Vazirani, Yannakakis, 1997)



Idea: route the deepest possible demand.

2-approximation in trees

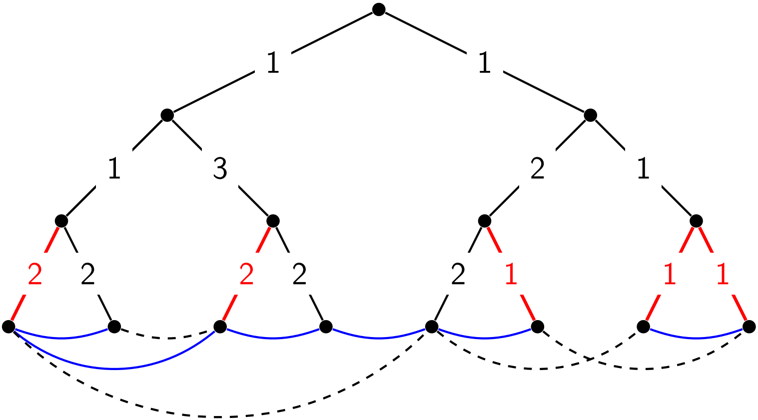
(Garg, Vazirani, Yannakakis, 1997)



Idea: route the deepest possible demand.

2-approximation in trees

(Garg, Vazirani, Yannakakis, 1997)

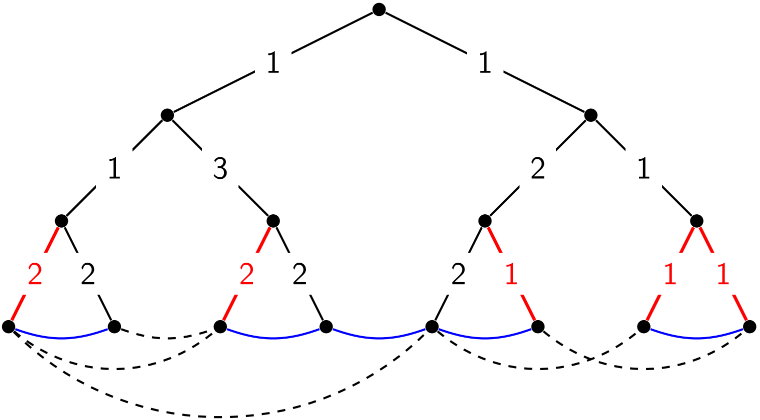


Idea: route the deepest possible demand.

McGill

2-approximation in trees

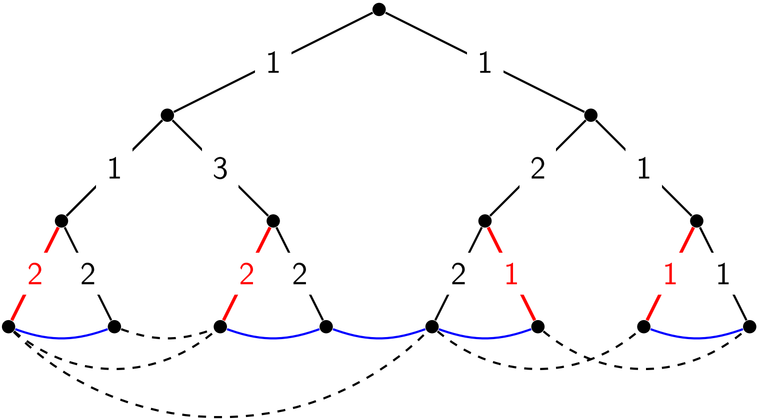
(Garg, Vazirani, Yannakakis, 1997)



Idea: route the deepest possible demand.

2-approximation in trees

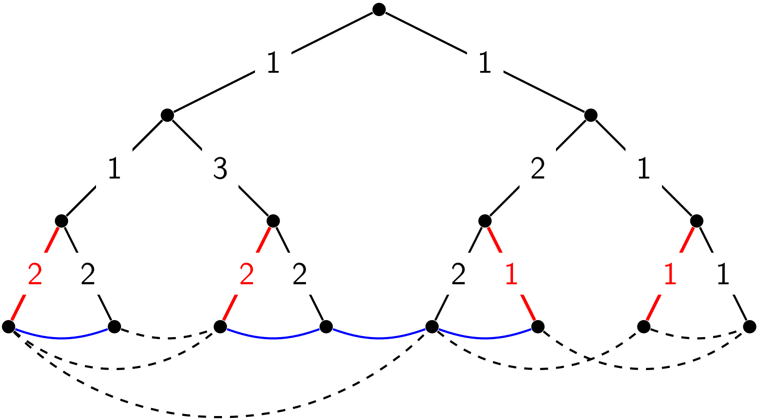
(Garg, Vazirani, Yannakakis, 1997)



Idea: route the deepest possible demand.

2-approximation in trees

(Garg, Vazirani, Yannakakis, 1997)

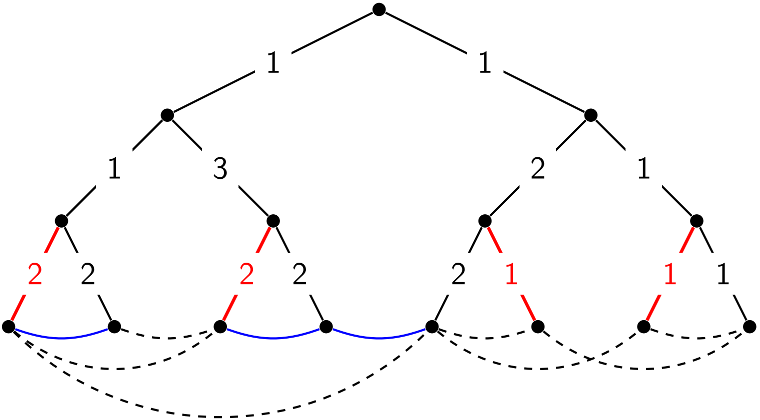


Idea: route the deepest possible demand.

McGill

2-approximation in trees

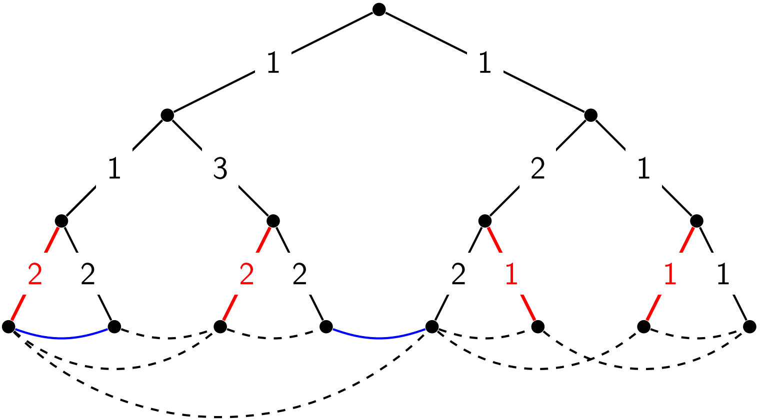
(Garg, Vazirani, Yannakakis, 1997)



Idea: route the deepest possible demand.

2-approximation in trees

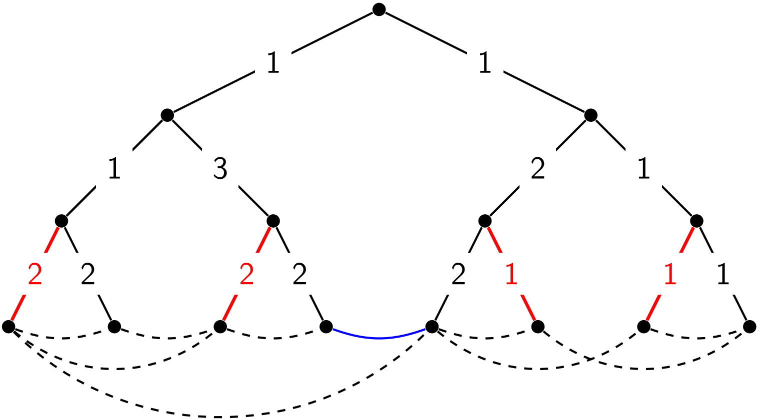
(Garg, Vazirani, Yannakakis, 1997)



Idea: route the deepest possible demand.

2-approximation in trees

(Garg, Vazirani, Yannakakis, 1997)



Idea: route the deepest possible demand.

McGill

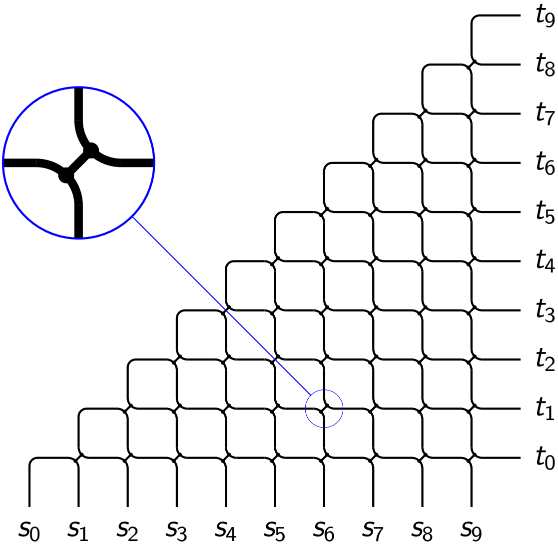
MEDP on trees: results

- APX-hard and
- 2-approximation, no weight, (Garg, Vazirani, Yannakakis, 1997)
- 4-approximation with weight (Chekuri, Mydlarz, Shepherd, 2003).

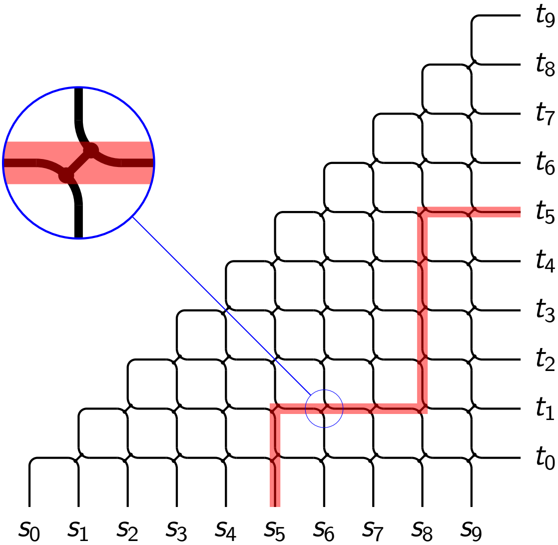
Both algorithms have a bottom-up approach.

Planar graphs

A bad example ($\sqrt{n}$ integrality gap):

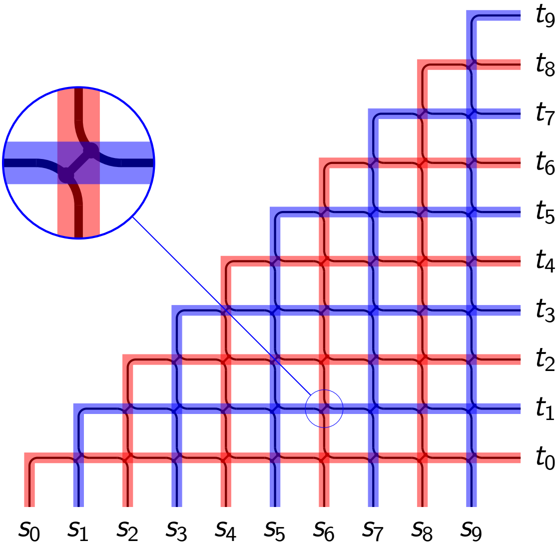


A bad example ($\sqrt{n}$ integrality gap):



Planar graphs

A bad example ($\sqrt{n}$ integrality gap):



McGill

Congestion

In the previous example, multiplying the capacities by 2 leads to an integral solution matching the fractional optimum.

Definition

Congestion: maximum ratio allowed between the number of paths taking an edge and its capacity.

MEDP on planar graphs

Theorem (Chekuri, Khanna, Shepherd, 2006)

$O(1)$-approximation with congestion 4 in planar graphs.

- Find a disc $\mathcal{D}$ with properties:
  - capacity of $\delta(\mathcal{D}) \ll$ flow routed inside $\mathcal{D}$,
  - $\frac{1}{10}$ of the flows routed inside $\mathcal{D}$ can be routed to the boundary of $\mathcal{D}$.
- Charge the flow crossing $\delta(\mathcal{D})$ to $\mathcal{D}$.
- Remove $\mathcal{D}$ and recurse.
- On $\mathcal{D}$, use the routing to the boundary, plus Okamura-Seymour theorem.

McGill

Bounded treewidth graphs

- Trees = graphs of treewidth 1,
- Graphs of treewidth 2 $\subset$ planar graphs,
- $O(k \log k \log n)$-approximation for graphs of treewidth k (Chekuri, Khanna, Shepherd 2006).
- Getting rid of the $\log n$ factor?
- Extending planar result to minor-closed classes of graphs?

McGill

Bounded treewidth graphs

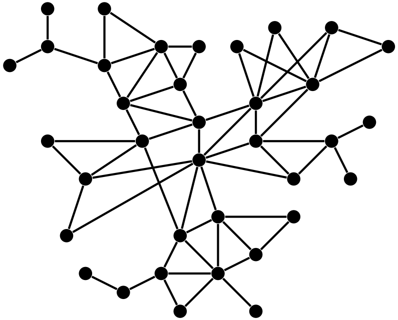
- Trees = graphs of treewidth 1,
- Graphs of treewidth 2 $\subset$ planar graphs,
- $O(k \log k \log n)$-approximation for graphs of treewidth k (Chekuri, Khanna, Shepherd 2006).
- Getting rid of the $\log n$ factor?
- Extending planar result to minor-closed classes of graphs?

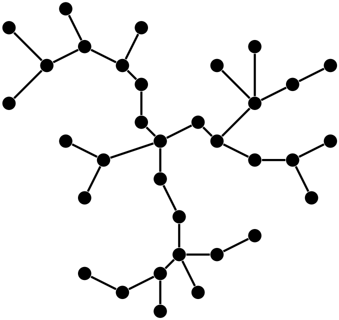
Theorem

For graphs of treewidth k, α_k-approximation with congestion β_k.

McGill

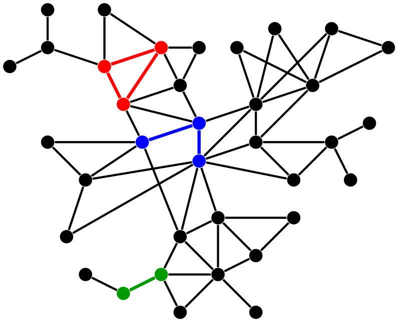
A graph with treewidth 2

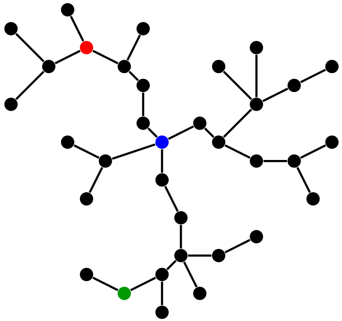




McGill

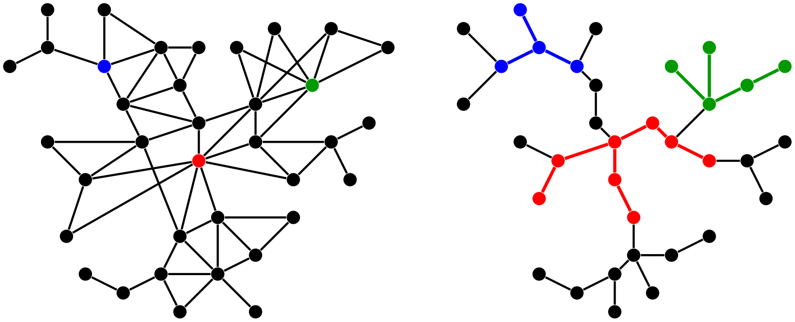
Bags. . .





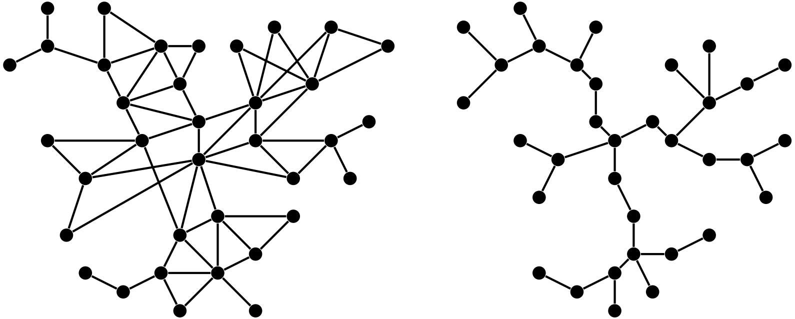
Every bag contains at most $k + 1$ vertices.

. . . and vertices

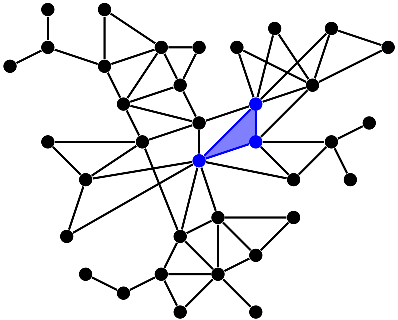


The bags containing a given vertex form a subtree.
Two adjacent vertices have non-disjoint subtrees.

Intersection of adjacent bags

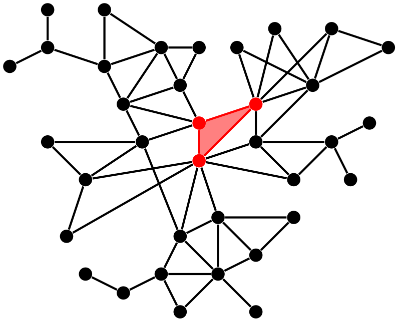


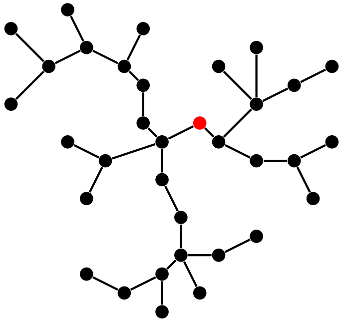
Intersection of adjacent bags



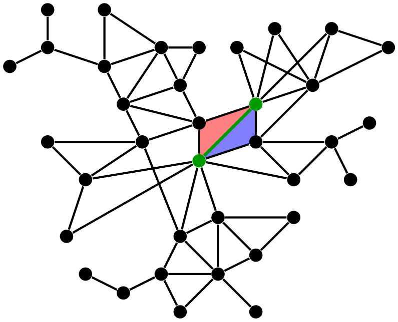
McGill

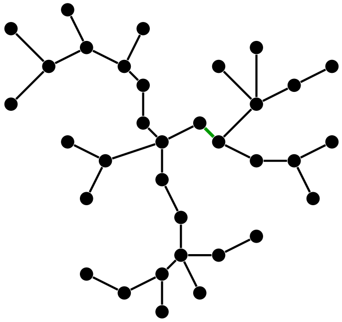
Intersection of adjacent bags





Intersection of adjacent bags





Intersection of adjacent bags $\implies$ cutset of size k

McGill

Proof of $O(1)$-approx, $O(1)$-congestion

Let x be a fractional optimum solution.

Definition

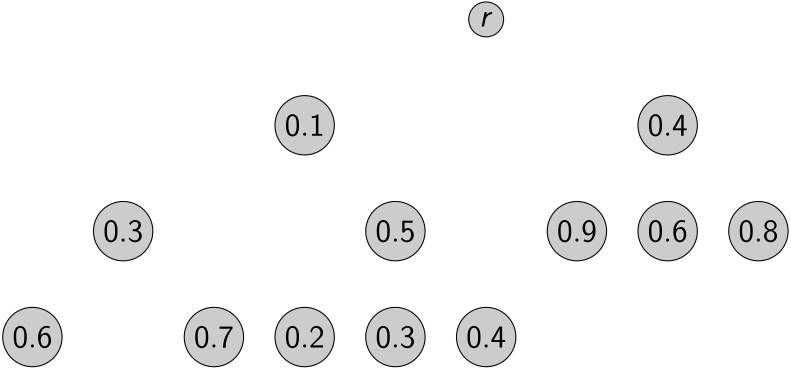
Marginal flow at v: value of the flow paths in x having extremity v.

Main ideas:

- Bottom-up approach,
- Cutting along a sparse cut and charging to the inside,
- Clustering.

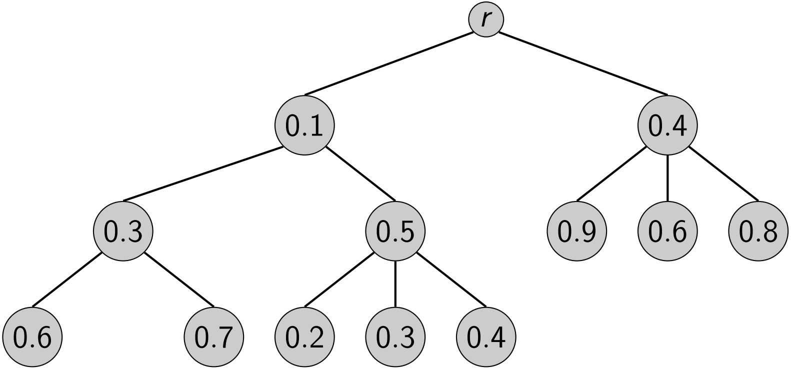
McGill

The clustering tool



Suppose there is a flow to r with these marginal values.

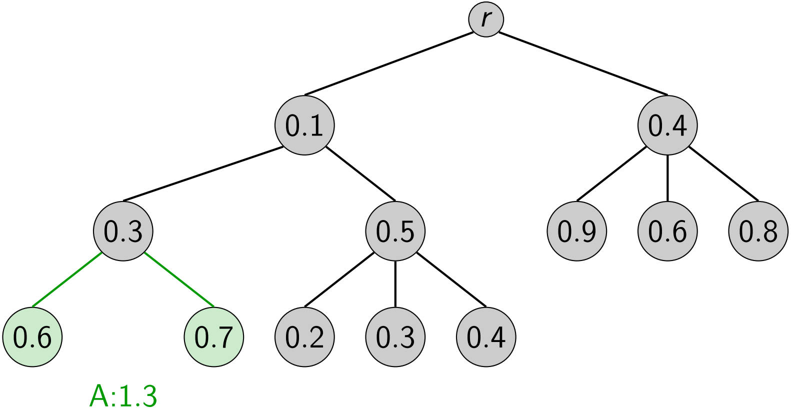
The clustering tool



Take an arbitrary spanning tree.

McGill

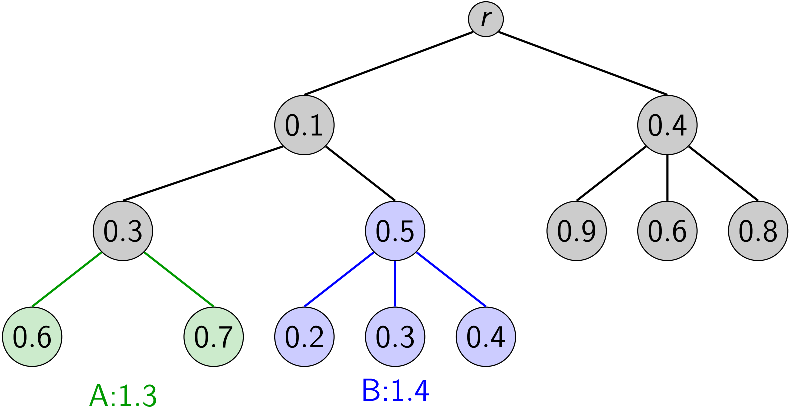
The clustering tool



Find a lowest level node with marginal value ≥ 1.
Take just enough sons to get a value ≥ 1

McGill

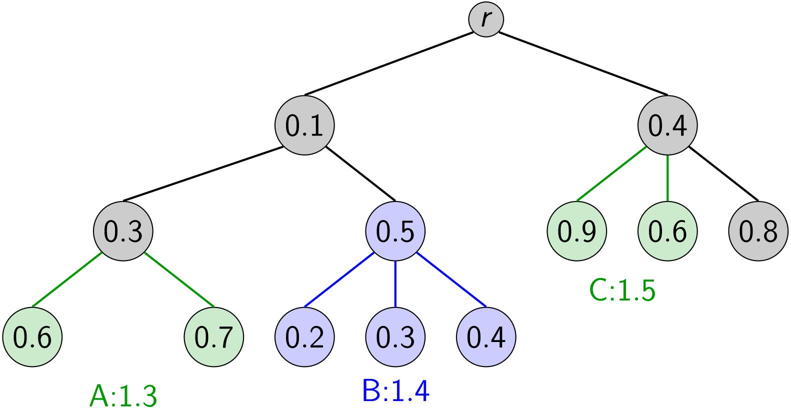
The clustering tool



Find a lowest level node with marginal value ≥ 1.
Take just enough sons to get a value ≥ 1 , repeat.

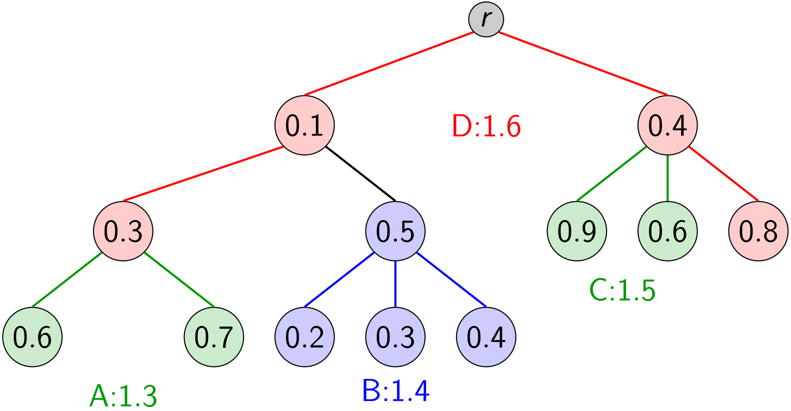
McGill

The clustering tool



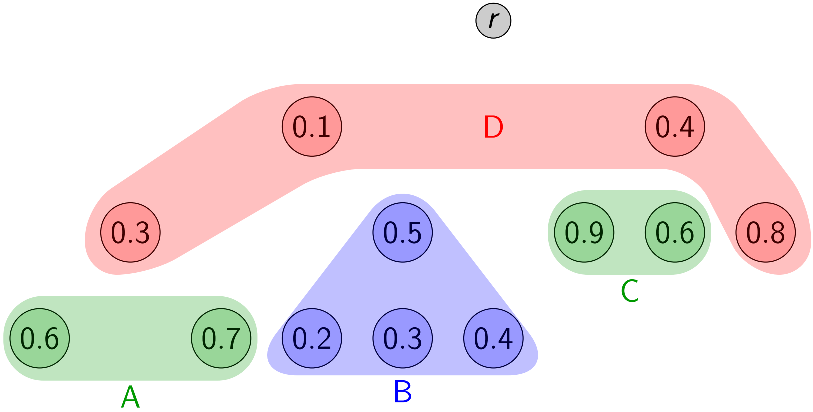
Again. . .

The clustering tool



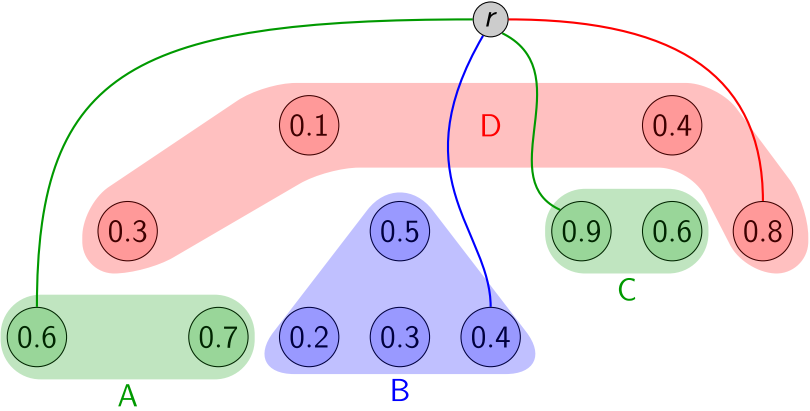
Again… until the remaining marginal value is < 3.

The clustering tool



Clusters send a flow ≥ 1 to the root...

The clustering tool

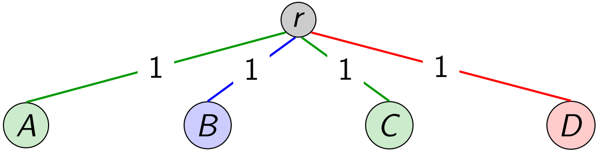


Clusters send a flow ≥ 1 to the root...

...so we can find edge-disjoint paths.

McGill

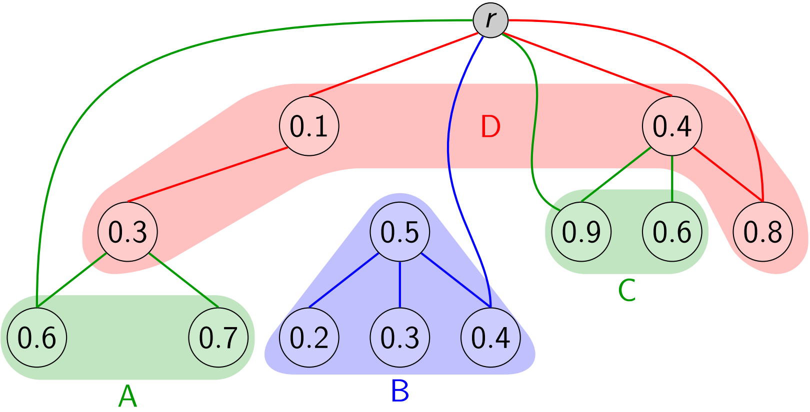
Contracting the clusters



- Replace each cluster by a leaf.
- Also contract the demands.
- Then find an integral routing...
- ...and uncontract the edge-disjoint paths.
- We get a 3-approximation with congestion 2.

McGill

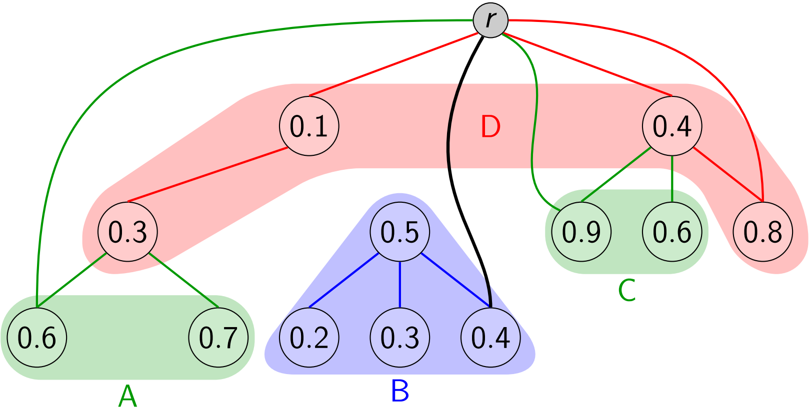
Uncontracting a path



For a path satisfying a demand to the 0.2 blue node.

McGill

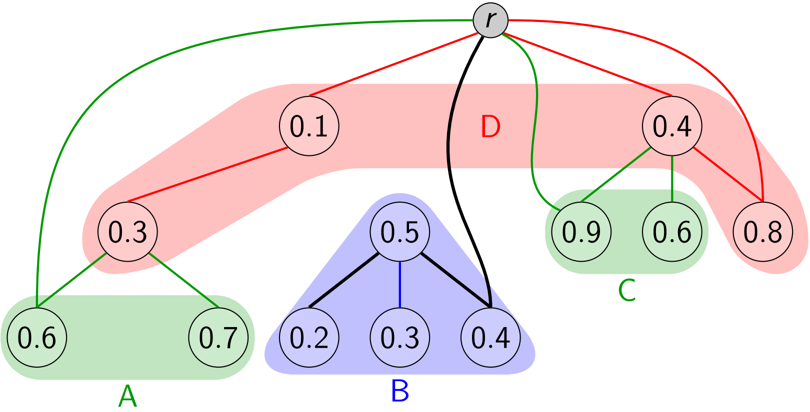
Uncontracting a path



For a path satisfying a demand to the 0.2 blue node.

McGill

Uncontracting a path



For a path satisfying a demand to the 0.2 blue node.

McGill

Clustering: what we get

- If we can route a fraction of the marginal flow to $U \subset V$,
- Then, move the demands to U,
- Up to constant approximation, constant congestion:

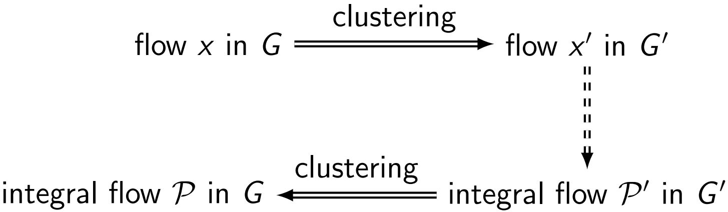
$$\text{flow } x \text{ in } G \xRightarrow{\text{clustering}} \text{flow } x' \text{ in } G'$$

$$\text{integral flow } \mathcal{P} \text{ in } G \xLeftarrow{\text{clustering}} \text{integral flow } \mathcal{P}' \text{ in } G'$$

McGill

Clustering: what we get

- If we can route a fraction of the marginal flow to $U \subset V$,
- Then, move the demands to U,
- Up to constant approximation, constant congestion:

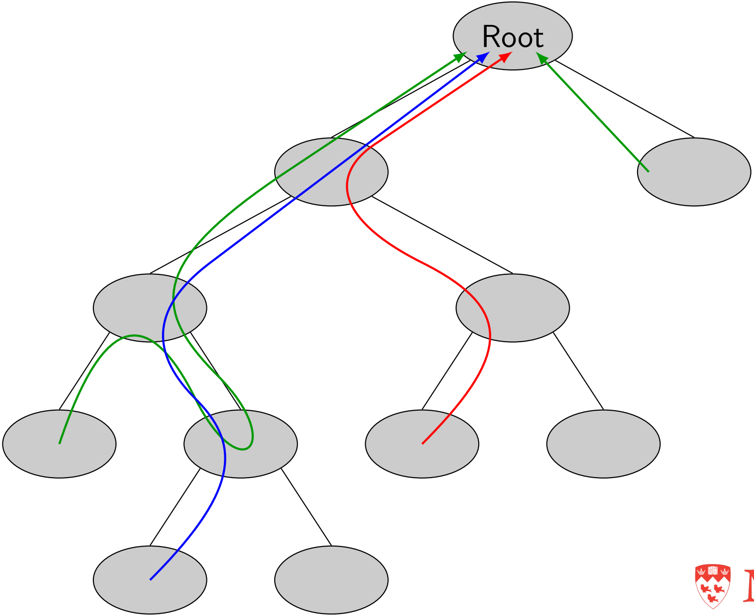


The algorithm

Route the marginal values to the root of the decomposition tree.

- if success, then use clustering to conclude.
- if fail, cut along a sparse cut.

Easy case: a flow to the root



Easy case: solution

There is a flow f routing $\frac{1}{10}$ of the marginal flow to the root.

- Make clusters using this flow $f \Longrightarrow$ fractional flow x'.
- The root has at most $k+1$ vertices, that are the terminals for x'.
- Select the pair (u, v) with maximum fractional flow x' between them.
- Find a packing of $\lceil x'(u, v) \rceil$ disjoint (u, v)-paths, uncontract them.

Easy case: solution

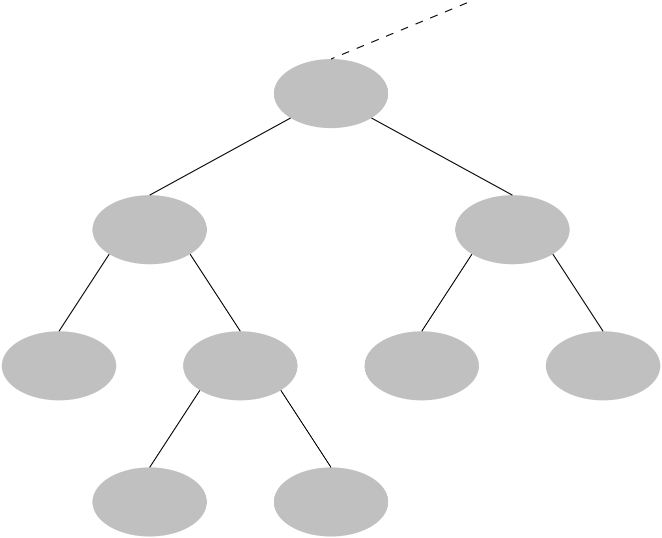
There is a flow f routing $\frac{1}{10}$ of the marginal flow to the root.

- Make clusters using this flow f $\Longrightarrow$ fractional flow x'.
- The root has at most $k+1$ vertices, that are the terminals for x'.
- Select the pair (u, v) with maximum fractional flow x' between them.
- Find a packing of $\lceil x'(u, v) \rceil$ disjoint (u, v)-paths, uncontract them.

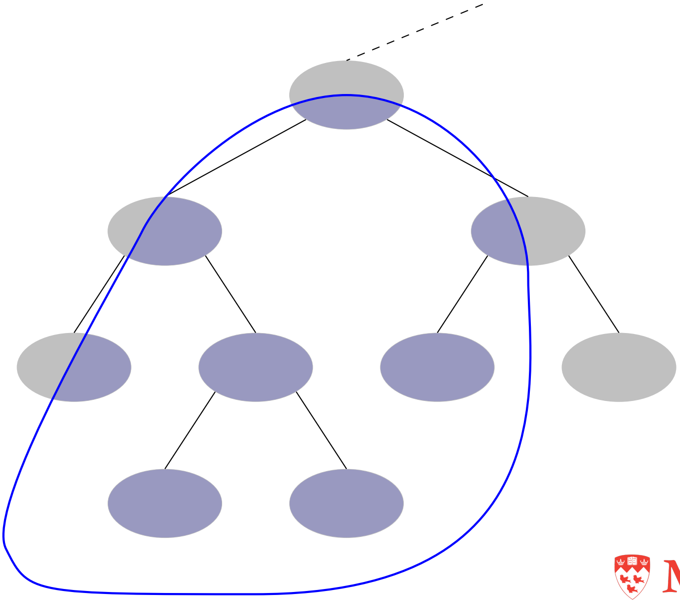
αk^2-approximation with β congestion.

McGill

Hard case: there is a sparse cut



Hard case: there is a sparse cut



Hard case: there is a sparse cut

There is a sparse cut X separating terminals from the root.

- Remove the flow through this cut.

Hard case: there is a sparse cut

There is a sparse cut X separating terminals from the root.

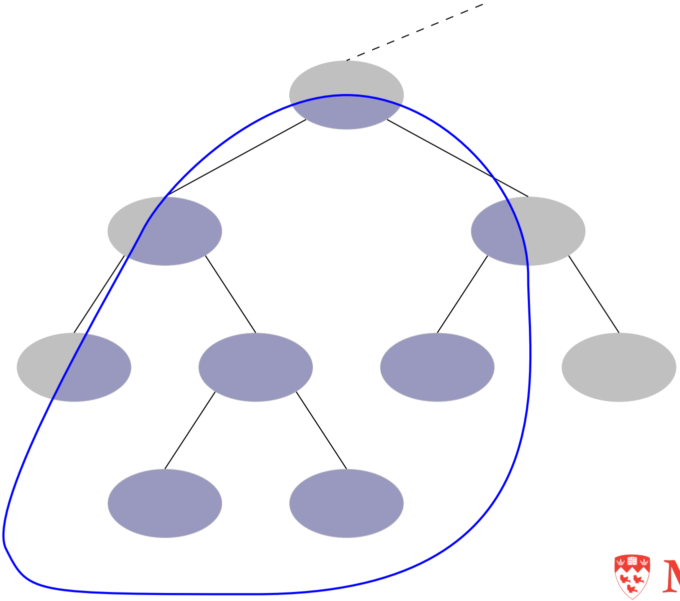
- Remove the flow through this cut.
- Charge the lost flow to the demands inside X.

Hard case: there is a sparse cut

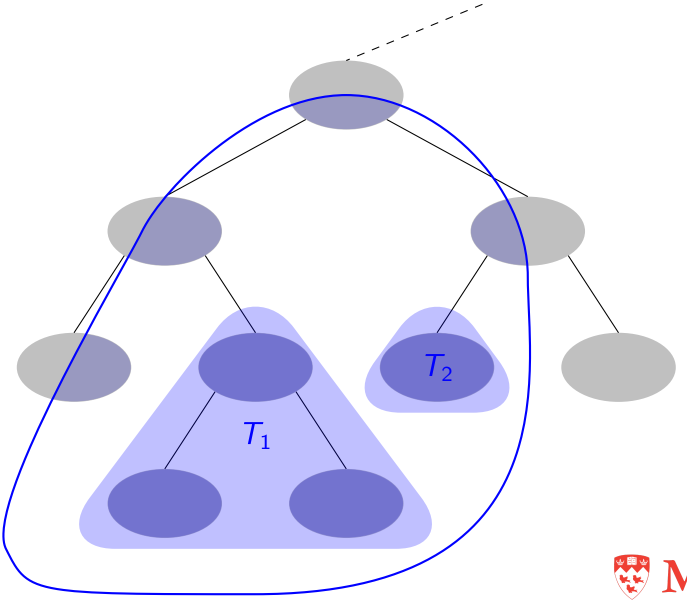
There is a sparse cut X separating terminals from the root.

- Remove the flow through this cut.
- Charge the lost flow to the demands inside X.
- Recurse on $G - X$ (smaller graph of treewidth k).

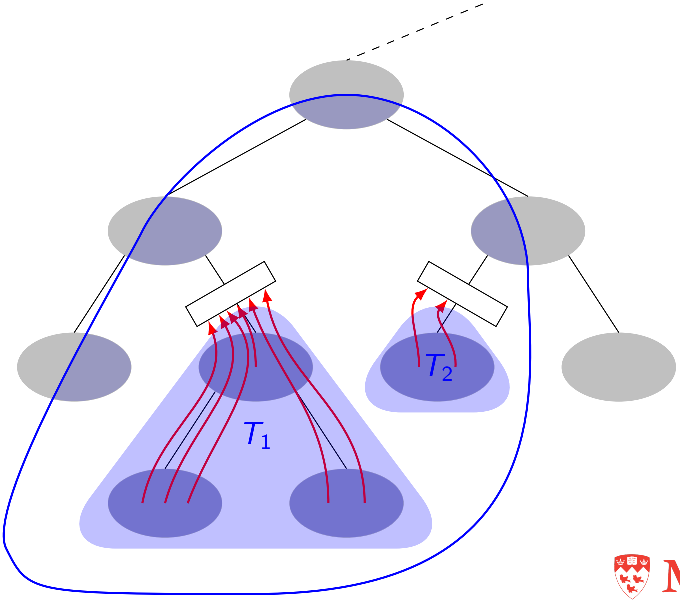
Hard case: there is a sparse cut



Hard case: there is a sparse cut



Hard case: there is a sparse cut



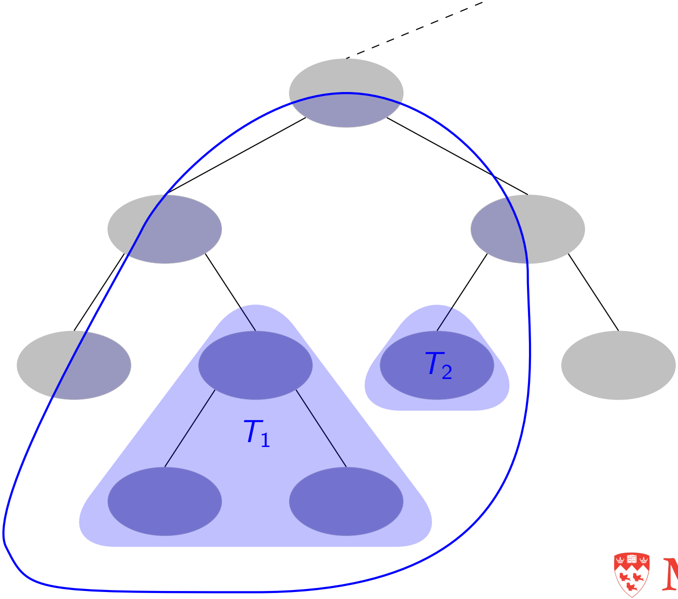
Hard case: there is a sparse cut

There is a sparse cut X separating terminals from the root.

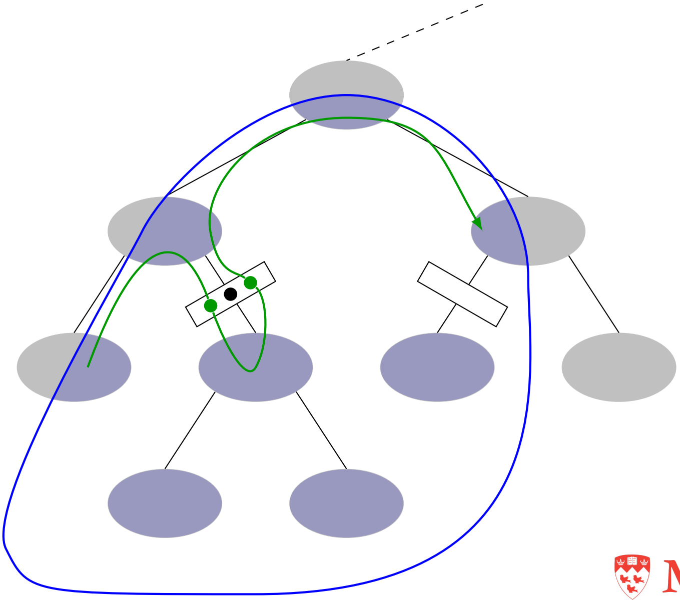
- Remove the flow through this cut.
- Charge the lost flow to the demands inside X.
- Recurse on $G - X$ (smaller graph of treewidth k).
- Apply clustering on the complete subtrees of X.

McGill

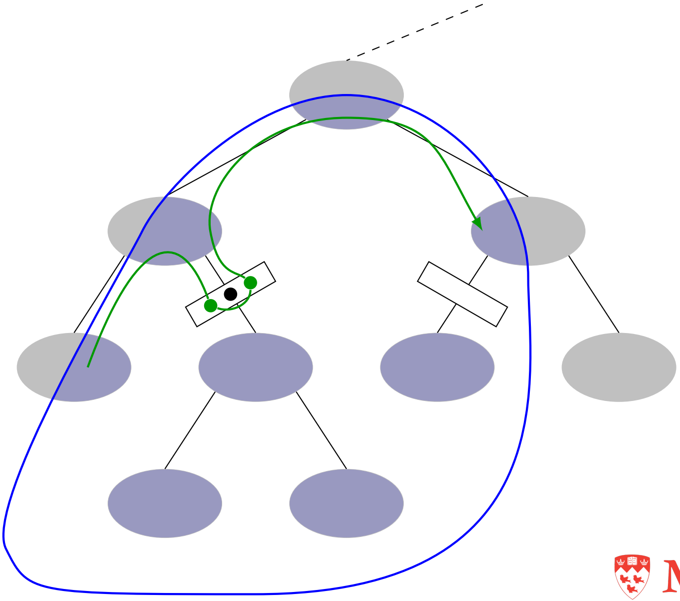
Hard case: there is a sparse cut



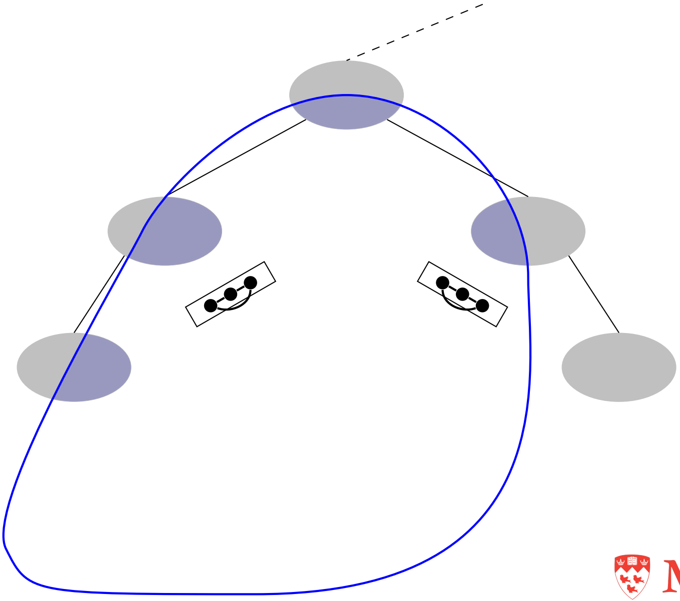
Hard case: there is a sparse cut



Hard case: there is a sparse cut

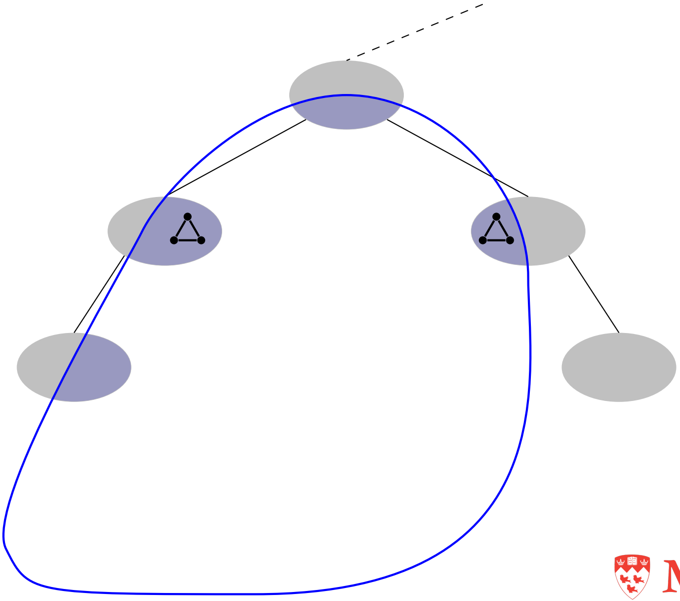


Hard case: there is a sparse cut



McGill

Hard case: there is a sparse cut



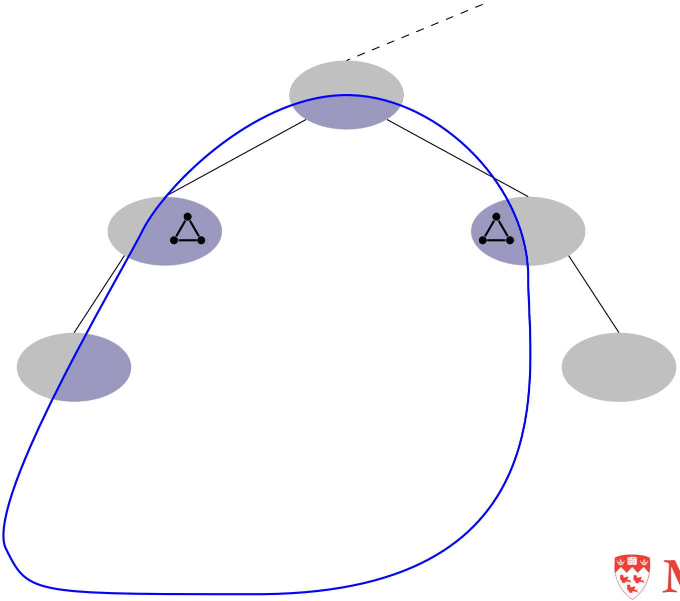
McGill

Hard case: there is a sparse cut

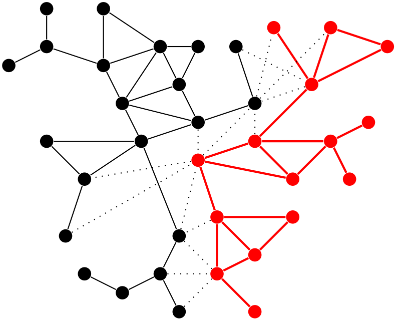
There is a sparse cut X separating terminals from the root.

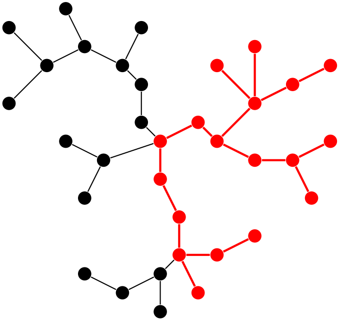
- Remove the flow through this cut.
- Charge the lost flow to the demands inside X.
- Recurse on $G - X$ (smaller graph of treewidth k).
- Apply clustering on the complete subtrees of X.
- Contract the complete subtrees into cliques (congestion k^2).

Hard case: there is a sparse cut

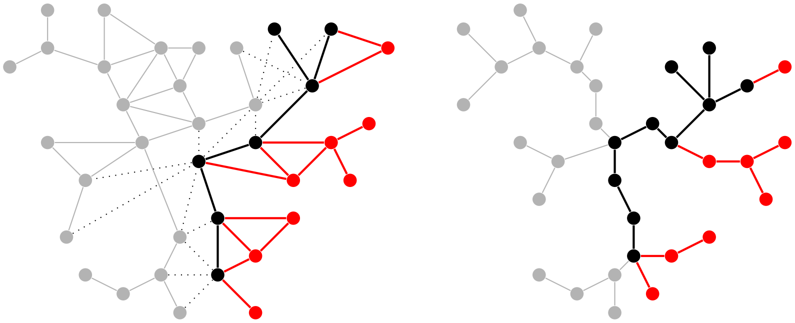


Hard case in action

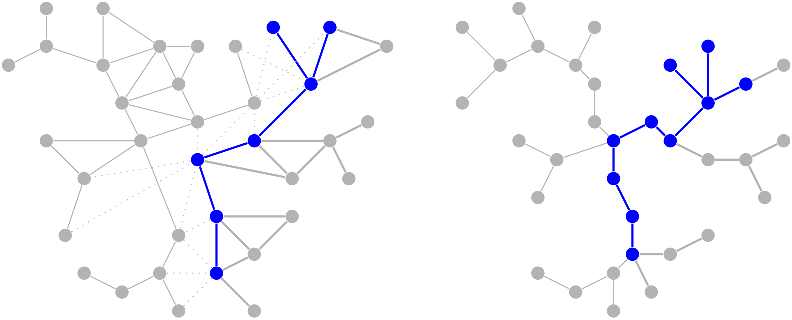




Hard case in action



Hard case in action



Hard case: there is a sparse cut

There is a sparse cut X separating terminals from the root.

- Remove the flow through this cut.
- Charge the lost flow to the demands inside X.
- Recurse on $G - X$ (smaller graph of treewidth k).
- Apply clustering on the complete subtrees of X.
- Contract the complete subtrees into cliques (congestion k^2).
- Apply induction on the contracted graph (treewidth $k - 1$).

McGill

What's next?

- weighted version,
- better bounds for congestion and approximation (exponential in the treewidth now),
- extend it to minor-closed classes of graphs.

The end

Thank you!