Surviving in Directed Graphs: A Quasi-Polynomial-Time Poly-logarithmic Approximation for 2-Connected Directed Steiner Tree (STOC'17)

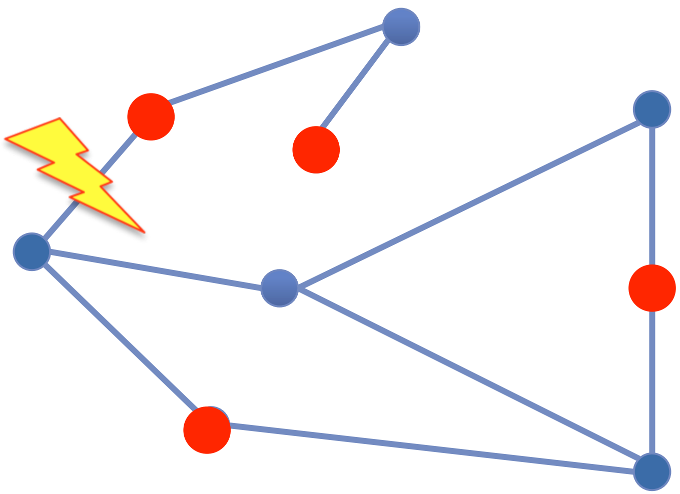
Bundit Laekhanukit
Weizmann Institute of Science

Fabrizio Grandoni
IDSIA, University of Lugano
fabrizio@idsia.ch

Survivable Network Design

Prob: Design (cheap) networks that satisfy given connectivity requirements (between pairs or groups of nodes) **despite** a few **edge/vertex failures**

E.g.: Connect red nodes



Many SND problems are **NP-hard**, we will focus on **approximation algorithms**

Surviving in Directed Graphs?

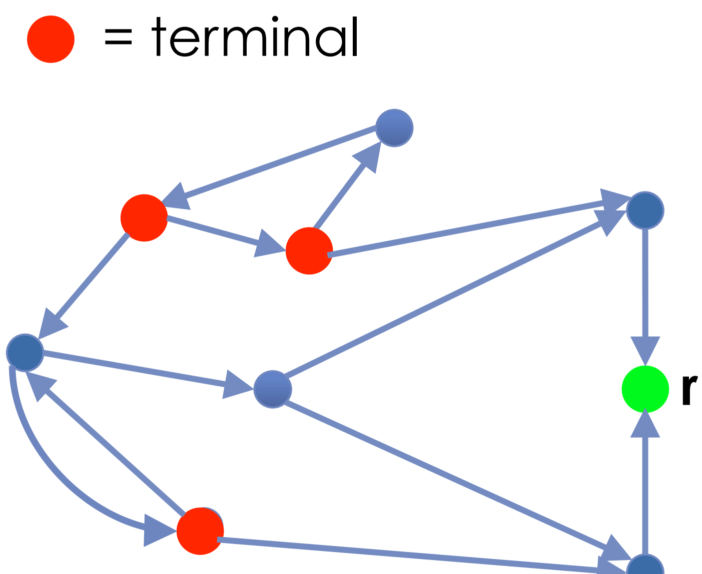
Many approximation algorithms are known for SND problems on **undirected graphs**

Steiner Network Problem: edge-connectivity r(u,v) between every pair of nodes (u,v)	[Jain'01] 2-apx
k-Vertex Connected Steiner Tree: k-vertex connectivity from terminals to a root	[Fleisher et al.'06] 2-apx, k=2 [Nutov'12] O(k log k)-apx
k-Vertex Connected Steiner Subgraph: k-vertex connectivity between terminals	[Fleisher et al.'06] 2-apx, k=2 [Nutov'12] O(k $\log^2$ k)-apx [Cheriyan,Vetta'07] O(1)-apx, metric edge costs
k-Vertex Connected Spanning Subgraph: k-vertex connectivity between all nodes	[Nutov'14] O(log (n/(n-k))log k)-apx, n=# nodes [Cheriyan,Vegh'14] 6-apx, $n \geq 2k^3$

Prob: What about directed graph?

Directed Steiner Tree (DST)

Def: In the **Directed Steiner Tree** problem (**DST**) we are given an **n**-node **directed edge-weighted** graph G, a **root** r, and a set of **terminals** S={1,...,**h**}. Our goal is to compute a min-cost subgraph H that contains a directed path from each terminal to r



Thr [Zelikosky'97, Charikar et al.'99]: For any D>0 in $n^{O(D)}$ time one can compute a $\mathbf{O(D h^{1/D} \log^2 h)}$ approximation for DST

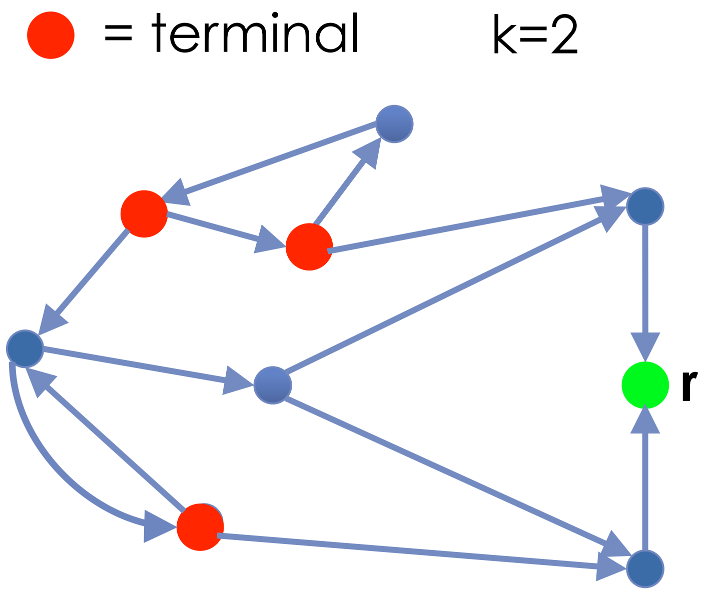
Cor1: For fixed ε>0, $\mathbf{O(h^{\varepsilon})}$ approximation in poly-time

Cor2: $\mathbf{O(\log^3 h)}$ approximation in quasi-polynomial-time $2^{\mathrm{polylog}(n)}$ (**QPT**)

Thr [Halpering,Krauthgamer'03]: DST is $\mathbf{O(\log^{2-\varepsilon} n)}$ hard to approximate

k-Connected Directed Steiner Tree (k-DST)

Def: the **k-(Edge)-Connected Directed Steiner Tree** problem (**k-DST**) is the generalization of DST where one wants **k edge-disjoint paths** from each terminal to the root r



k-Connected Directed Steiner Tree (k-DST)

Def: the **k-(Edge)-Connected Directed Steiner Tree** problem (**k-DST**) is the generalization of DST where one wants **k edge-disjoint paths** from each terminal to the root r

[Cheriyan, Laekhanukit, Naves, Vetta '14]	$2^{\log^{1-\varepsilon} n}$ hard to apx. k^{δ} hard to apx.
[Laekhanukit '14]	$k^{1/2-\varepsilon}$ hard to apx.
[Laekhanukit '16]	O($k^{D-1} D \log n$)-apx, **D-shallow instances** (directed paths of hop-length at most D)

Prob [Feldam, Kortsarz, Nutov '12]:

- Can we get any non-trivial approximation for the general case?
- Possibly analogous to the DST case?
- Say for k=O(1)? Even just for k=2?

k-Connected Directed Steiner Tree (k-DST)

Def: the **k-(Edge)-Connected Directed Steiner Tree** problem (**k-DST**) is the generalization of DST where one wants **k edge-disjoint paths** from each terminal to the root r

Thr [G.,Laekhanukit'17]: For any D>0 in $n^{O(D)}$ time one can compute a **$O(D^3 \log D \, h^{2/D} \log n)$** approximation for **2-DST**

Cor1: For ε>0, **$O(h^{\varepsilon})$** apx

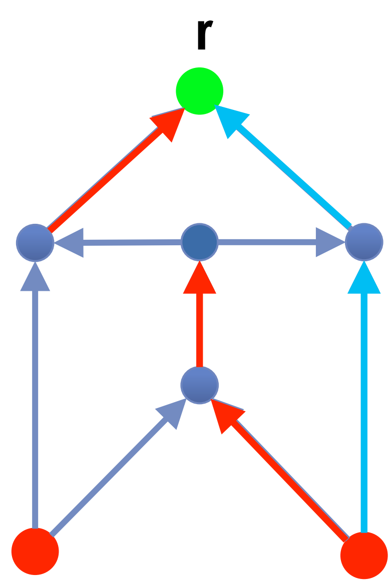
Cor2: **$O(\log^3 h \log n \log\log h)$** apx in QPT

- Complex LP where we combine:
  - Zelikowsky's **height reduction**
  - **Divergent Steiner trees**
  - **Embedding into shallow trees** [Laekhanukit'16]
  - **Group-Steiner-Tree** (**GST**) LP
- LP rounding where we combine:
  - **GKR rounding** for GST [Garg,Kojevod,Ravi'00]
  - **Random path mapping**
  - **Cut-based** connectivity analysis [Chalermsook,G.,Laekhanukit'15]

Divergent Steiner Trees

Prob: Can we decompose a 2-DST solution into 2 edge disjoint DST solutions?

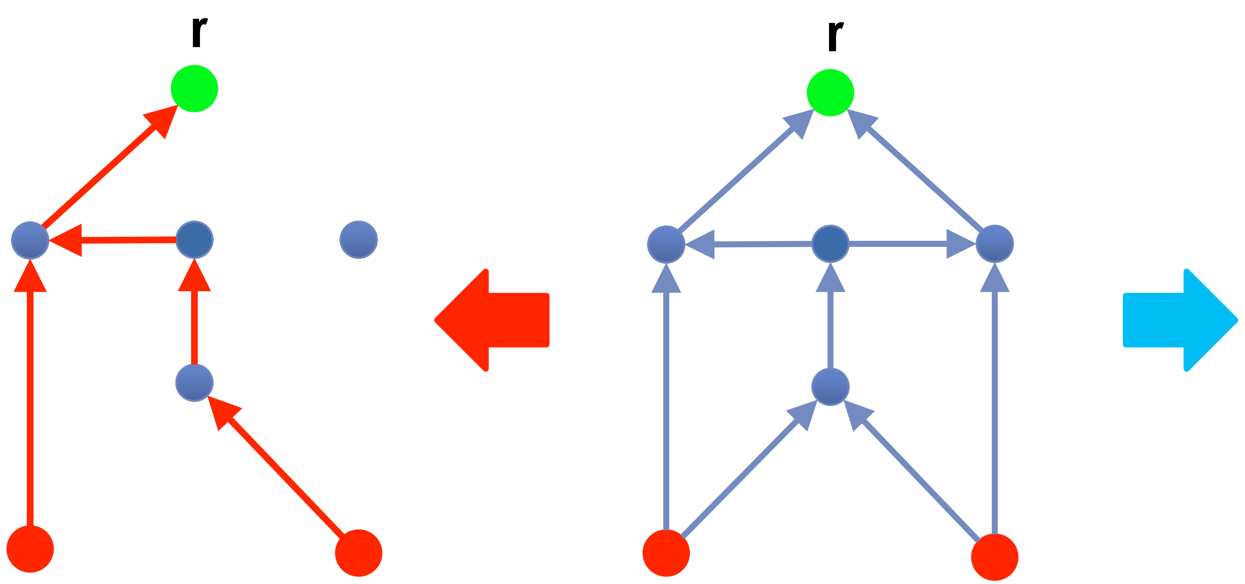
NO!



Divergent Steiner Trees

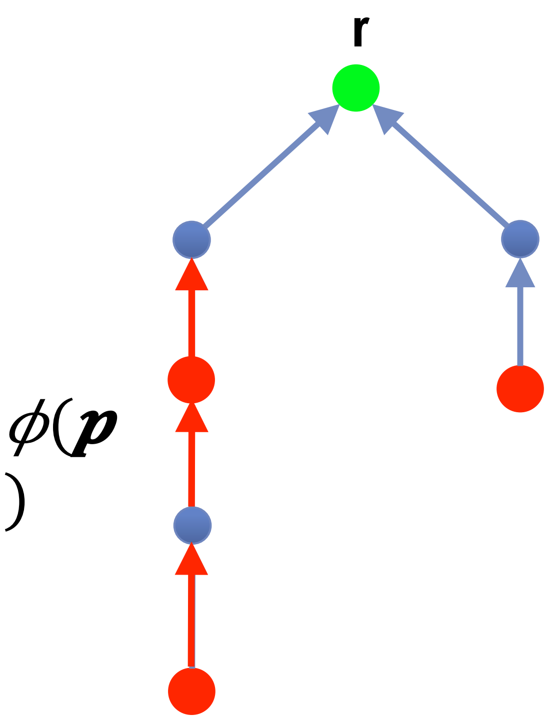
Def: two (possibly not edge disjoint) DST solutions T_1 and T_2 are **divergent** if for any terminal t, the t-r path in T_1 and T_2 are edge disjoint

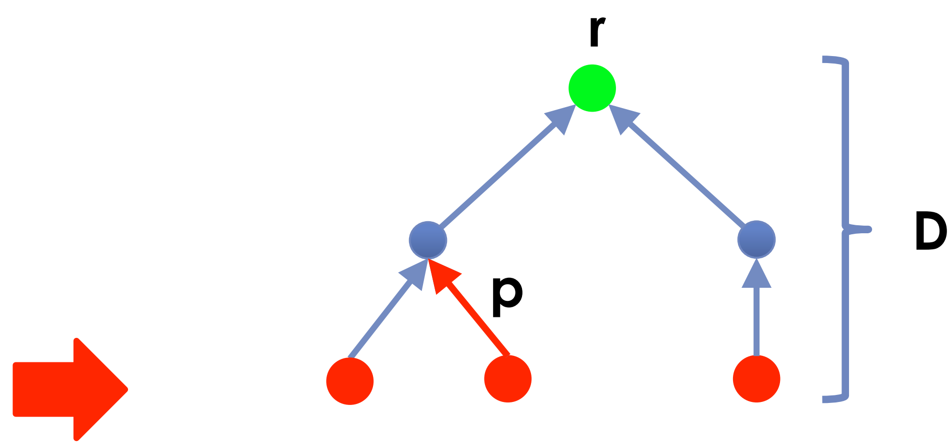
Thr [Georgiadis,Tarjan'05;Berczi,Kovacz'11]: any 2-DST solution can be "decomposed" into 2 divergent Directed Steiner trees



Height Reduction

Thr [Zelikovsky'97]: for any D>0 and DST T, there exists a DST T in the metric closure of T of **depth ≤D** and cost **w(T)≤O(D$h^{1/D}$)w(T)**



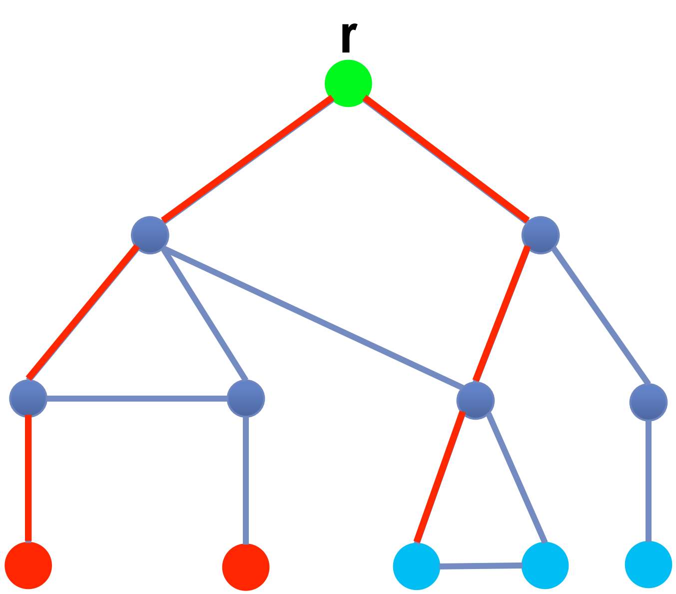


Rem: for DST apx. (**T=OPT**) one can consider the **metric closure** of the overall graph **G**

Rem: the proof defines a **mapping** ϕ from edges of T to paths of T (between the same endpoints) where each edge of T is used $\leq \beta = O(D h^{1/D})$ times altogether (**bounded congestion**)

Group Steiner Tree (GST)

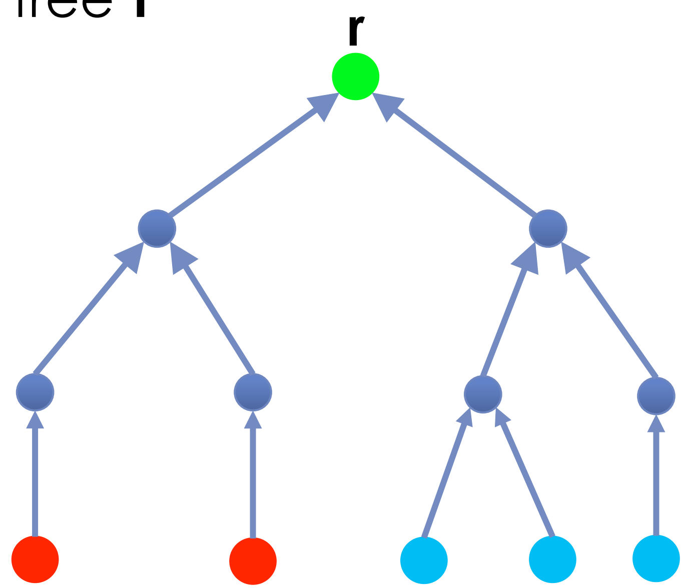
Def: in the **Group Steiner Tree** problem (**GST**) we are given an undirected edge-weighted graph G, a root r, and **h** subsets of nodes $G_1, \ldots, G_h$ (**groups**). The goal is to compute the cheapest tree that contains r and **at least one node per group**



Rem: We will consider the **2-GST** generalization, with connectivity 2 between each group and the root

Group Steiner Tree (GST)

Thr [Garg,Kojevod,Ravi'00]: there is a **$O(\log^2 h)$-apx** for **GST** on a tree **T**



- Solve GST LP
- Apply **GKR rounding** $O(\log^2 h)$ times:
  - select edge p incident to r independently with probability y_p
  - otherwise select p with probability y_p / y_e if parent edge e of p was selected before

$$
\begin{aligned}
\min \quad & \sum_p w(p)\, y_p && \text{(GST LP)} \\
\text{s.t.} \quad & f_p^i \le y_p && \forall i\in[h]\,\forall p\in E(T) \\
& \sum_{p\in\delta^{in}(w)} f_p^i = \sum_{p\in\delta^{out}(w)} f_p^i && \forall i\in[h]\,\forall w\in V(T)\setminus(\{r\}\cup G_i) \\
& \sum_{p\in\delta^{out}(G_i)} f_p^i \ge 1 && \forall i\in[h] \\
& y_p, f_p^i \in [0,1] && \forall i\in[h]\,\forall p\in E(T)
\end{aligned}
$$

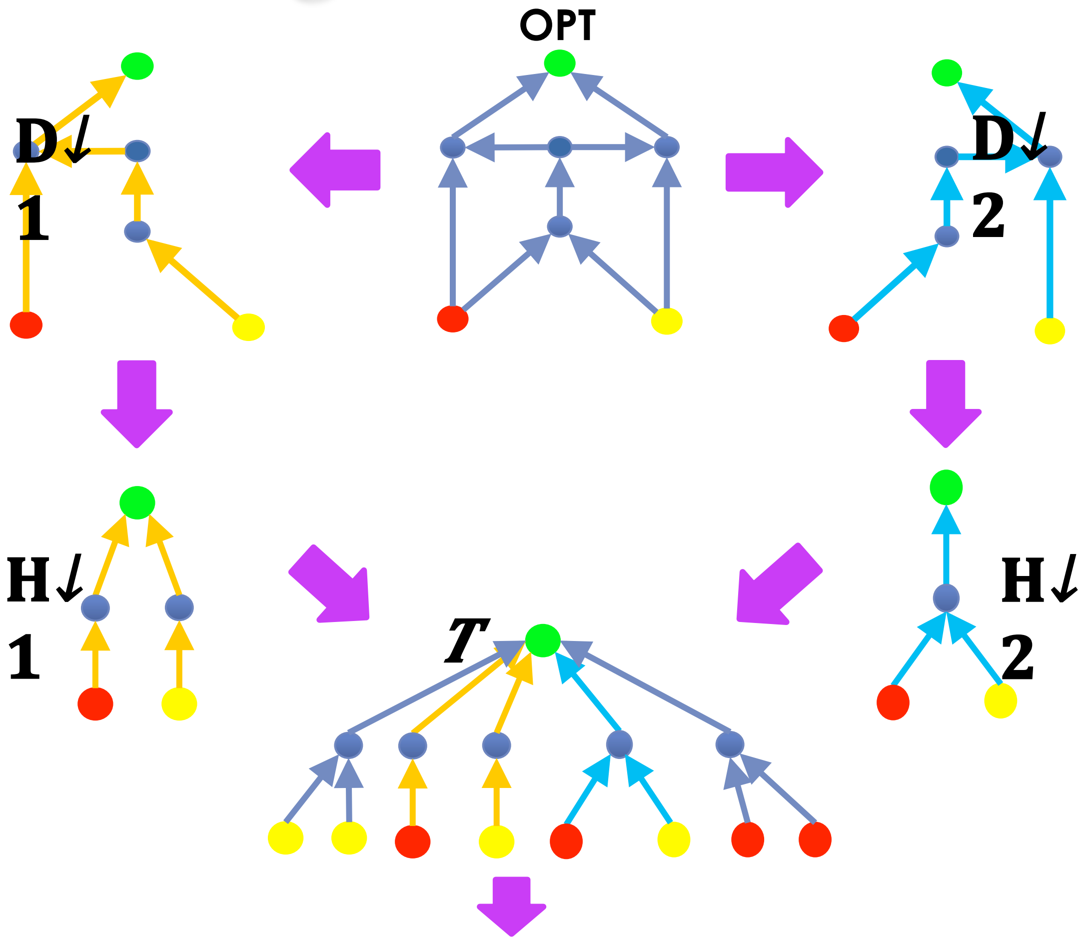
Rem: If the tree has height D, **O(D log h)** rounds are sufficient

Rem: The same works if the flow leaving G_i is $\Omega(1)$

Rem: We will use a similar LP for 2-GST…

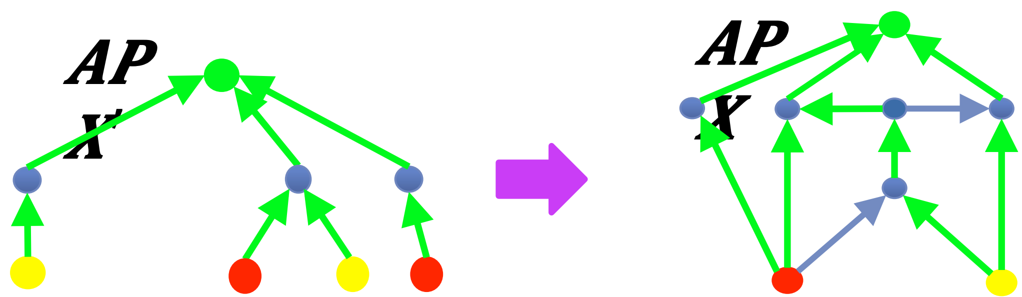
The Big Plan

1. Divergent Steiner Trees
2. D-Height Reduction
3. Embedding into 2-GST instance in a D-Shallow Tree
4. GKR Rounding for 2-GST LP
5. Map back to G



Prob: Differently from DST, cannot use metric closure in Height Reduction

$f_p^i \le y_p$	$\forall i \in [h] \forall p \in E(T)$
$\sum_{p \in \delta^{in}(w)} f_p^i = \sum_{p \in \delta^{out}}$	$\forall i \in [h]$ $\forall w \in V(T) \setminus (\{r\} \cup G_i)$



Problem Fixing

Prob: Cannot use metric closure in Height Reduction (we would lose connectivity properties of original graph)

Idea: Let an LP create the mapping!

- Define a (u,v)-flow $f_{p,e}$ of value y_p in G for each $p=(u,v) \in E(T)$
- Enforce **bounded congestion** (to keep cost under control)

$$f_{p,e} \le x_e \qquad \forall p \in E(T) \forall e \in E(G)$$

$$\sum_{e \in \delta^{out}(u)} f_{p,e} = y_p \qquad \forall p=(u,v) \in E(T)$$

$$\sum_{e \in \delta^{in}(w)} f_{p,e} = \sum_{e \in \delta^{out}(w)} f_{p,e} \qquad \forall p=(u,v) \in E(T) \; \forall w \in V(G) \setminus \{u,v\}$$

$$\sum_{p} f_{p,e} \le 2\beta x_e \in O(D h^{1/D}) x_e \qquad \forall e \in E(G)$$

Rem: x_e choice variable for $e \in E(G)$

Rem: We will interpret this flow as a **distribution over paths**

Problem Fixing

Prob: Cannot use metric closure in Height Reduction (we would lose connectivity properties of original graph)

Idea: Let an LP create the mapping!

- Define a similar flow for each terminal i
- Enforce **divergency** (useful for connectivity analysis)

$f_{p,e}^{i} \le f_{p,e}$	$\forall p \in E(T)$ $\forall e \in E(G)$ $\forall i \in [h]$
$\sum_{e \in \delta^{out}(u)} f_{p,e}^{i} = f_{p}^{i}$	$\forall p=(u,v) \in E(T)$ $\forall i \in [h]$
$\sum_{e \in \delta^{in}(w)} f_{p,e}^{i} = \sum_{e \in \delta^{out}(w)} f_{p,e}^{i}$	$\forall p=(u,v) \in E(T)$ $\forall i \in [h]$ $\forall w \in V(G) \setminus \{u,v\}$
$\sum_{p} f_{p,e}^{i} \le x_{e}$	$\forall e \in E(G)$ $\forall i \in [h]$

The LP

min	$\sum_{e} w(e)x_e$	(2-DST LP)	
s.t.	$f_p^i \le y_p$	$\forall p \in E(T)\ \forall i \in [h]$	2-GST LP
	$\sum_{p\in\delta^{in}(w)} f_p^i = \sum_{p\in\delta^{out}(w)} f_p^i$	$\forall i \in [h]\ \forall w \in V(T)\setminus(\{r\}\cup G_i)$	2-GST LP
	$\sum_{p\in\delta^{out}(G_i)} f_p^i \ge 2$	$\forall i \in [h]$	2-GST LP
	$f_{p,e} \le x_e$	$\forall p \in E(T)\ \forall e \in E(G)$	Path mapping
	$\sum_{e\in\delta^{out}(u)} f_{p,e} = y_p$	$\forall p=(u,v) \in E(T)$	Path mapping
	$\sum_{e\in\delta^{in}(w)} f_{p,e} = \sum_{e\in\delta^{out}(w)} f_{p,e}$	$\forall p=(u,v)\in E(T)\ \forall w \in V(G)\setminus\{u,v\}$	Path mapping
	$\sum_{p} f_{p,e} \le 2\beta x_e$	$\forall e \in E(G)$	Divergency
	$f_{p,e}^i \le f_{p,e}$	$\forall p \in E(T)\ \forall e \in E(G)\ \forall i \in [h]$	Divergency
	$\sum_{e\in\delta^{out}(u)} f_{p,e}^i = f_p^i$	$\forall p=(u,v)\in E(T)\ \forall i \in [h]$	Divergency
	$\sum_{e\in\delta^{in}(w)} f_{p,e}^i = \sum_{e\in\delta^{out}(w)} f_{p,e}^i$	$\forall p=(u,v)\in E(T)\ \forall i \in [h]\ \forall w \in V(G)\setminus\{u,v\}$	Divergency
	$\sum_{p} f_{p,e}^i \le x_e$	$\forall e \in E(G)\ \forall i \in [h]$	Divergency

The Algorithm

1. Solve 2-DST LP $\Rightarrow$ ($x_e, y_p, f_p^i, f_{p,e}, f_{p,e}^i$)
2. For j=1,…, **$O(D \log n)$**
    I. Round $\{y_p\}$ with GKR rounding $\Rightarrow T_j \subseteq T$
    II. For q=1,…,**$O(D h^{1/D} \log D)$**
        a) For each p=(u,v)$\in T_j$, sample (u,v)-path $P_{p,q}$ "from" $f_{p,e}/y$
    III. Let $H_j = \cup P_{p,q} \subseteq G$
3. Return $H = \cup H_j$

Lem: the expected cost is **$O(D^3 \log D\, h^{2/D} \log n)$** times the LP value

- Using bounded congestion, in each execution of step a) each edge $e \in G$ belongs to **$O(\beta) = O(D h^{1/D})$** paths $P_{p,q}$ in expectation

The Algorithm

1. Solve 2-DST LP $\Rightarrow (x_e, y_p, f_p^i, f_{p,e}, f_{p,e}^i)$
2. For j=1,...., **O(D log n)**
   I. Round $\{y_p\}$ with GKR rounding $\Rightarrow T_j \subseteq T$
   II. For q=1,...,**O($Dh^{1/D}$ log D)**
      a) For each p=(u,v)$\in T_j$, sample (u,v)-path $P_{p,q}$ "from" $f_{p,e}/y$
   III. Let $H_j = \cup P_{p,q} \subseteq G$
3. Return $H = \cup H_j$

Lem: w.h.p., for each terminal i and edge e, H\{e} contains an i-r path

Rem: inspired by [Chalermsook, G., Laekhanukit '15] for k-GST

- We discard "bad" edges p∊T such that $P_{p,q}$ has "large" probability to contain e
- Using divergency and bounded congestion, we show that remaining "good" edges support flow ≥1/2 from G_i to r
- Hence H_j\{e} has "large enough" probability to connect i to r

Open Problems

Prob: Obtaining similar approximation for k-DST (say, up to a factor f(k)polylog(n))

Rem: the divergency theorem doesn't hold for k≥3

Idea: our approach would still work with a weakened form of the divergency theorem where:

- We decompose OPT into **f(k)polylog(n) trees** T_i (rather than k)
- For any i and set F of k-1 edges, **at least one T_i \F** connects i with r


Faleminderit!
ευχαριστώ
Ačiū
ТИ БЛАГОДАРАМ!!!
Grazie!
BEDANKT!
TANȦN
Спасибо
TAKK!
D̂AKUJEM
KiiTOS
kisiselbasari.com
Teşekkür Ederim
MULTUMESC!
MERCI
THANK YOU
תודה.
Hvala!
GRACIAS
KÖSZÖNÖM
Děkuji
ДЯКУЮ
DANKE
DZIĘKUJĘ!
tak
OBRIGADO
GRAZZI
TACK!!!
paldies!