A simple primal-dual approximation algorithm for 2-edge-connected spanning subgraphs

Stephan Beyer[1] Markus Chimani[1] Joachim Spoerhase[2]

[1]Osnabrück University, Germany

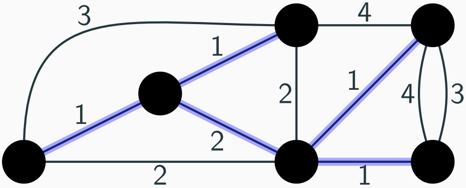
[2]Aalto University, Finland

The minimum spanning tree problem

Application: basic topological network design problem

Given an undirected multigraph $G = (V, E)$
with edge costs $c\colon E \to \mathbb{R}_{\geq 0}$

find edges $E' \subseteq E$ of minimum total cost $c(E') := \sum_{e \in E'} c(e)$
such that $G[E']$ is connected.

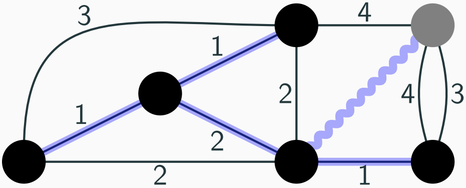


The minimum spanning tree problem

Application: basic topological network design problem

Given an undirected multigraph $G = (V, E)$
with edge costs $c\colon E \to \mathbb{R}_{\geq 0}$

find edges $E' \subseteq E$ of minimum total cost $c(E') := \sum_{e \in E'} c(e)$
such that $G[E']$ is connected.

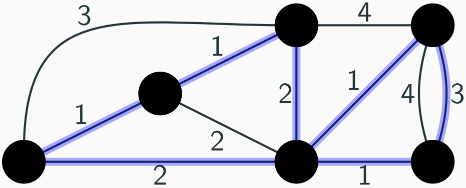


The minimum 2-edge-connected spanning subgraph problem

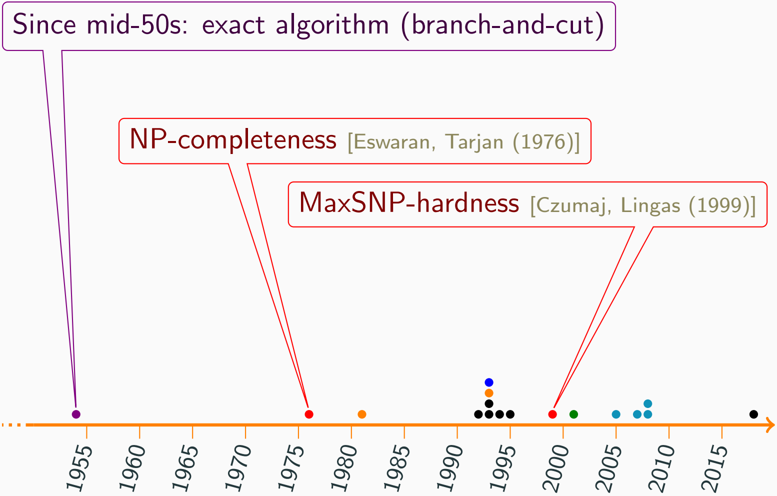
Application: basic topological network design problem
with protection against single link failures

Given an undirected multigraph $G = (V, E)$
with edge costs $c\colon E \to \mathbb{R}_{\geq 0}$

find edges $E' \subseteq E$ of minimum total cost $c(E') := \sum_{e \in E'} c(e)$
such that $G[E']$ is 2-edge-connected.



Timeline

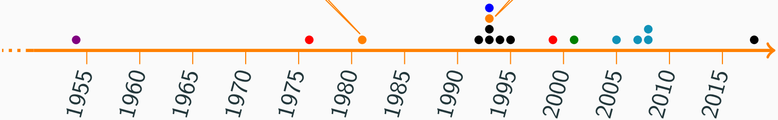


Since mid-50s: exact algorithm (branch-and-cut)
NP-completeness [Eswaran, Tarjan (1976)]
MaxSNP-hardness [Czumaj, Lingas (1999)]
1955
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010
2015

Timeline

3-approximation algorithm by augmenting a minimum spanning tree, in $O(n^2)$ time [Frederickson, JáJá (1981)]

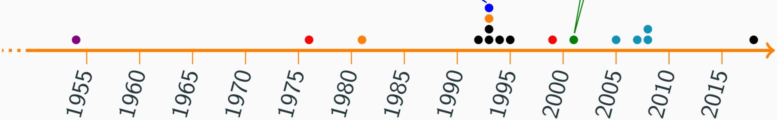
same in $O(m + n \log n)$ time [Khuller, Thurimella (1993)]



Timeline

2-approximation algorithm using a simple reduction to the weighted matroid intersection problem [Khuller, Vishkin (1994)] solvable in $O(n(m + n \log n) \log n)$ time [Gabow (1991)]

2-approximation algorithm using iterative LP rounding [Jain (2001)]



Timeline

90s: a series of primal-dual 3-approximation algorithms

— [Saran, Vazirani, Young (1992)]

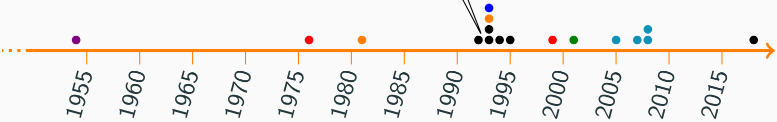
— [Gabow, Goemans, Williamson (1993)]

— [Klein, Ravi (1993)]

— [Goemans, Goldberg, Plotkin, Shmoys, Tardos (1994)]

— [Williamson, Goemans, Mihail, Vazirani (1995)]

best one with time $O(n^2 + n\sqrt{m \log\log n})$

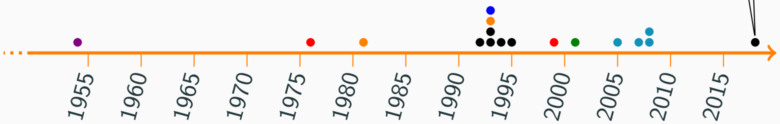


Timeline

Contribution

- only one grow phase (instead of two) to directly establish 2-edge-connectivity
- allows extremely simple implementation

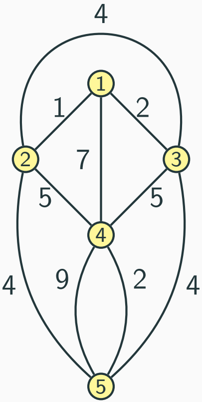
simple primal-dual 3-approximation algorithm with time $O(nm)$ [B., Chimani, Spoerhase (2018)]



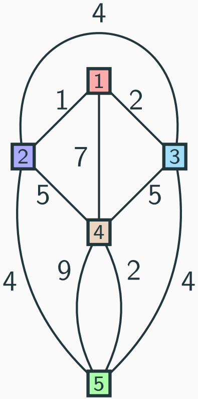
Our algorithm

The grow phase (initialization)

graph *G* with **costs *c***



forest *F*, reduced costs *r*



edges *T*

$\varnothing$

The grow phase

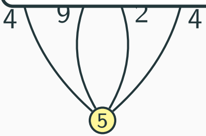
graph *G* with **costs *c***

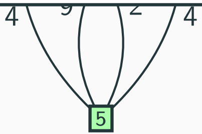
forest *F*, reduced costs *r*

edges *T*

In each grow step:

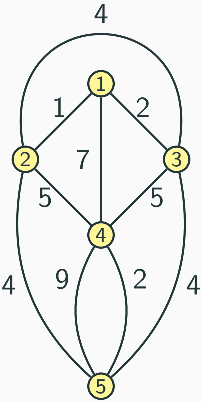
1. each leaf in F spends money Δ to afford incident edges
2. update r: reduce costs of incident edges by Δ
3. update F: add affordable edge and ensure acyclicity
4. update T: add edge



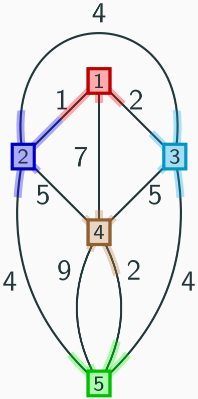


The grow phase (grow step 1)

graph G with **costs c**



forest F, reduced costs r



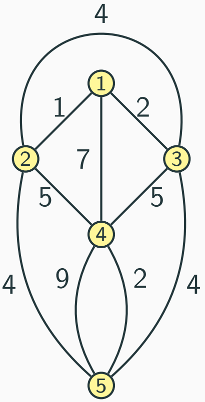
$\Delta = 0.5$

edges T

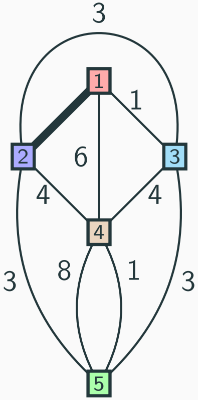
$\varnothing$

The grow phase (grow step 1)

graph *G* with **costs *c***



forest *F*, reduced costs *r*

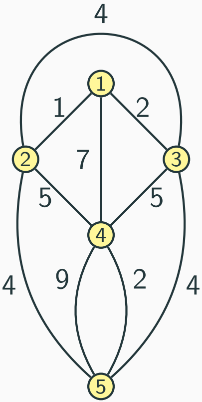


edges *T*

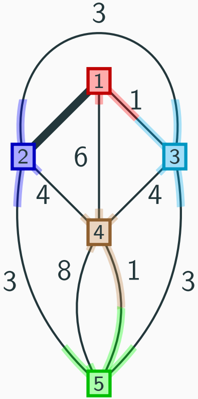
$\{1, 2\}$

The grow phase (grow step 2)

graph *G* with costs *c*



forest *F*, reduced costs *r*



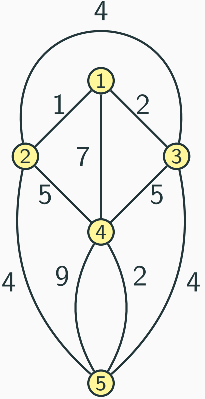
$\Delta = 0.5$

edges *T*

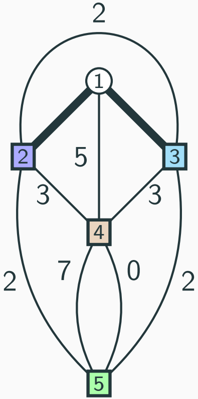
$\{1, 2\}$

The grow phase (grow step 2)

graph *G* with costs *c*



forest *F*, reduced costs *r*

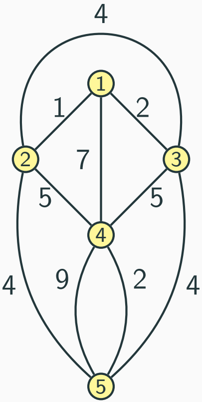


edges *T*

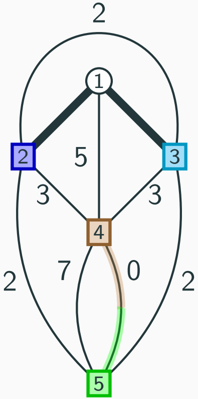
$\{1,2\}$
$\{1,3\}$

The grow phase (grow step 3)

graph *G* with costs *c*



forest *F*, reduced costs *r*



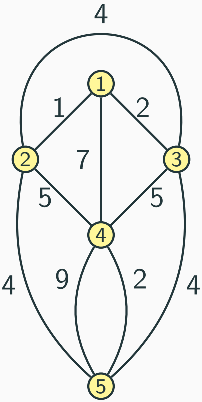
$\Delta = 0.0$

edges *T*

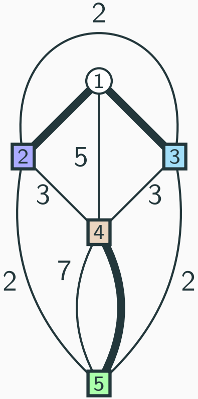
$\{1, 2\}$
$\{1, 3\}$

The grow phase (grow step 3)

graph G with costs c



forest F, reduced costs r

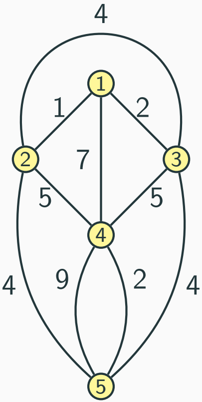


edges T

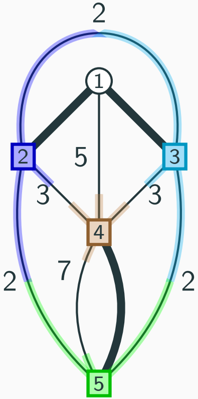
$\{1,2\}$
$\{1,3\}$
$\{4,5\}$

The grow phase (grow step 4)

graph *G* with costs *c*



forest *F*, reduced costs *r*



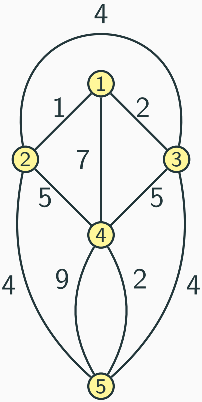
$\Delta = 1.0$

edges *T*

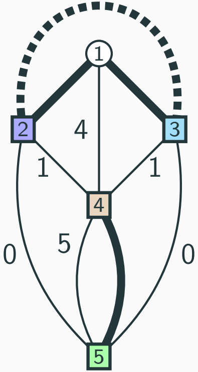
$\{1, 2\}$
$\{1, 3\}$
$\{4, 5\}$

The grow phase (grow step 4)

graph *G* with costs *c*



forest *F*, reduced costs *r*

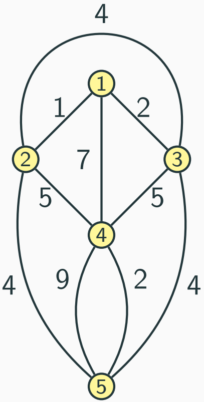


edges *T*

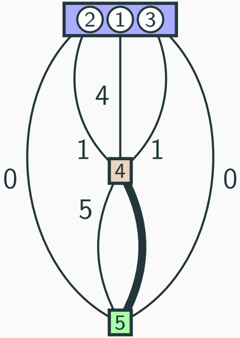
$\{1, 2\}$
$\{1, 3\}$
$\{4, 5\}$
$\{2, 3\}$

The grow phase (grow step 4)

graph *G* with costs *c*



forest *F*, reduced costs *r*

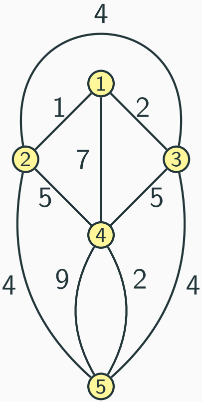


edges *T*

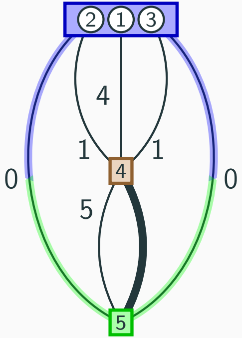
$\{1, 2\}$
$\{1, 3\}$
$\{4, 5\}$
$\{2, 3\}$

The grow phase (grow step 5)

graph *G* with costs *c*



forest *F*, reduced costs *r*



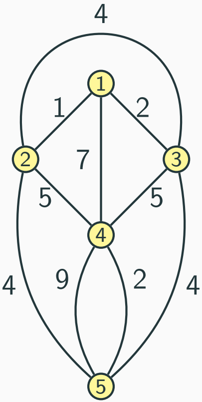
$\Delta = 0.0$

edges *T*

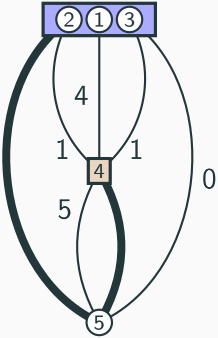
$\{1, 2\}$
$\{1, 3\}$
$\{4, 5\}$
$\{2, 3\}$

The grow phase (grow step 5)

graph G with costs c



forest F, reduced costs r

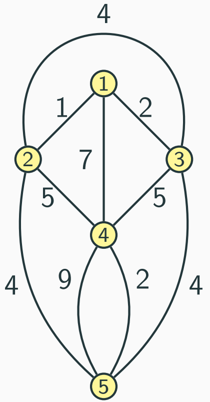


edges T

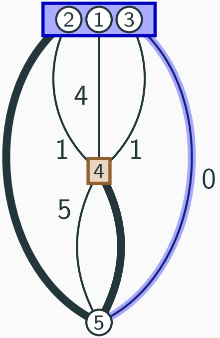
$\{1,2\}$
$\{1,3\}$
$\{4,5\}$
$\{2,3\}$
$\{2,5\}$

The grow phase (grow step 6)

graph G with costs c



forest F, reduced costs r



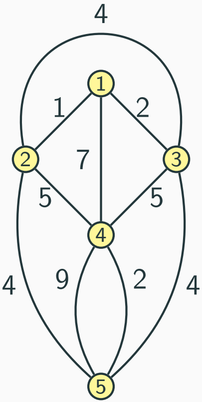
$\Delta = 0.0$

edges T

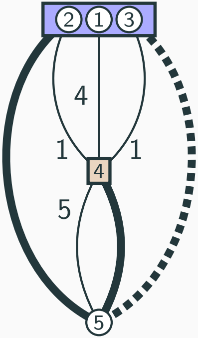
$\{1, 2\}$
$\{1, 3\}$
$\{4, 5\}$
$\{2, 3\}$
$\{2, 5\}$

The grow phase (grow step 6)

graph *G* with costs *c*



forest *F*, reduced costs *r*

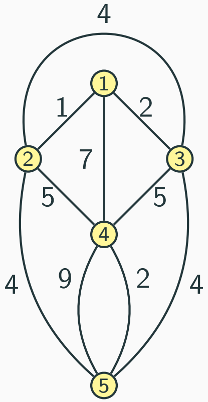


edges *T*

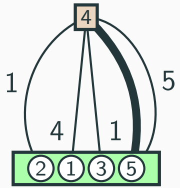
$\{1,2\}$
$\{1,3\}$
$\{4,5\}$
$\{2,3\}$
$\{2,5\}$
$\{3,5\}$

The grow phase (grow step 6)

graph G with costs c



forest F, reduced costs r

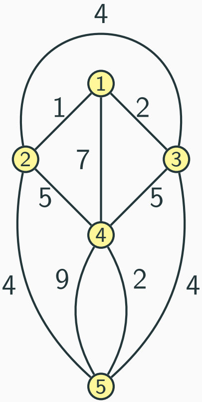


edges T

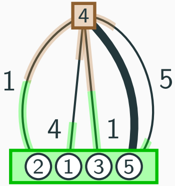
$\{1, 2\}$
$\{1, 3\}$
$\{4, 5\}$
$\{2, 3\}$
$\{2, 5\}$
$\{3, 5\}$

The grow phase (grow step 7)

graph *G* with costs *c*



forest *F*, reduced costs *r*



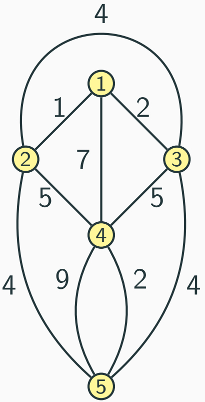
$\Delta = 0.5$

edges *T*

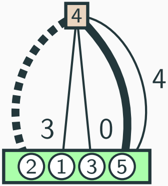
$\{1, 2\}$
$\{1, 3\}$
$\{4, 5\}$
$\{2, 3\}$
$\{2, 5\}$
$\{3, 5\}$

The grow phase (grow step 7)

graph *G* with **costs *c***



forest *F*, reduced costs *r*

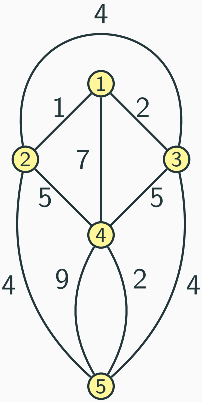


edges *T*

$\{1,2\}$
$\{1,3\}$
$\{4,5\}$
$\{2,3\}$
$\{2,5\}$
$\{3,5\}$
$\{2,4\}$

The grow phase (grow step 7)

graph *G* with costs *c*



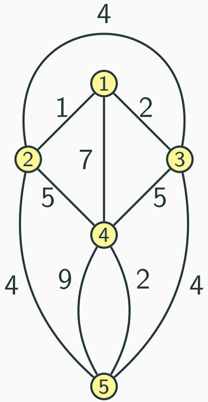
forest *F*, reduced costs *r*

edges *T*

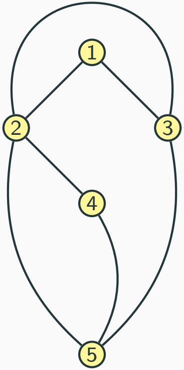
$\{1,2\}$
$\{1,3\}$
$\{4,5\}$
$\{2,3\}$
$\{2,5\}$
$\{3,5\}$
$\{2,4\}$

The cleanup phase

graph G with costs c



$G[T]$

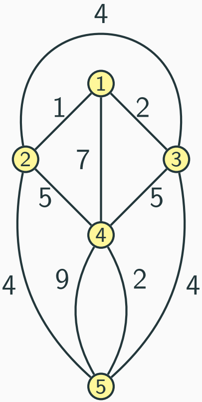


edges T

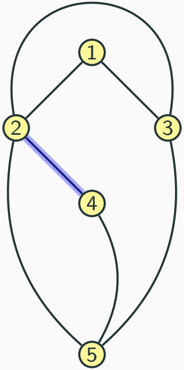
$\{1,2\}$
$\{1,3\}$
$\{4,5\}$
$\{2,3\}$
$\{2,5\}$
$\{3,5\}$
$\{2,4\}$

The cleanup phase

graph G with costs c



$G[T]$

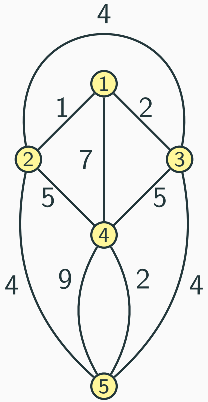


edges T

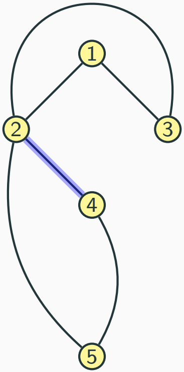
$\{1, 2\}$
$\{1, 3\}$
$\{4, 5\}$
$\{2, 3\}$
$\{2, 5\}$
$\{3, 5\}$
$\{2, 4\}$

The cleanup phase

graph *G* with costs *c*



***G*[*T*]**

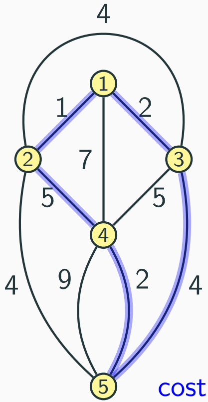


edges *T*

$\{1,2\}$
$\{1,3\}$
$\{4,5\}$
$\{2,3\}$
$\{2,5\}$
$\{3,5\}$
$\{2,4\}$

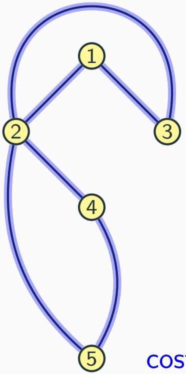
The cleanup phase

graph *G* with costs *c*



cost 14

$G[T]$



cost 18

edges *T*

$\{1,2\}$
$\{1,3\}$
$\{4,5\}$
$\{2,3\}$
$\{2,5\}$
$\{3,5\}$
$\{2,4\}$

Let’s do the math

The linear program

$$\begin{aligned} \textit{minimize} \quad & \sum_{e \in E} c(e)\, x_e \\ x(\delta_G(S)) &\geq 2 \qquad && \forall S \in \mathcal{S} \\ x_e &\in \{0,1\} \qquad && \forall e \in E \end{aligned}$$

all subsets of V except $\varnothing$ and V

edges with exactly one node in S

The linear program

$$\textit{minimize} \quad \sum_{e \in E} c(e)\, x_e$$

$$x(\delta_G(S)) \geq 2 \qquad \forall S \in \mathcal{S}$$

$$0 \leq x_e \leq 1 \qquad \forall e \in E$$

all subsets of V except $\varnothing$ and V

edges with exactly one node in S

The linear program and its dual

$$\begin{aligned} \textit{minimize} \quad & \sum_{e \in E} c(e) x_e \\ x(\delta_G(S)) &\geq 2 && \forall S \in \mathcal{S} \\ -x_e &\geq -1 && \forall e \in E \\ x_e &\geq 0 && \forall e \in E \end{aligned}$$

$$\begin{aligned} \textit{maximize} \quad & 2 \sum_{S \in \mathcal{S}} y_S - 1 \sum_{e \in E} z_e \\ y(\mathcal{S}_e) - z_e &\leq c(e) && \forall e \in E \\ y_S &\geq 0 && \forall S \in \mathcal{S} \\ z_e &\geq 0 && \forall e \in E \end{aligned}$$

$\mathcal{S}_e := \{S \in \mathcal{S} \mid e \in \delta_G(S)\}$

The dual program and the algorithm

- maintain implicit dual solution $(\bar{y}, \bar{z})$ such that $\forall e \in E$:
  $r(e) = c(e) - \bar{y}(\mathcal{S}_e) + \bar{z}_e \geq 0$
  - $\bar{y}_S$ increases by $\Delta \iff S$ corresponds to a leaf
  - $\bar{z}_e$ increases by $\Delta \iff$ increasing $\bar{y}_S$ only violates constraint
- $r(e) = 0 \iff$ constraint for e is tight
- Let $\bar{T}$ be the solution. For each $e \in \bar{T}$, we have $r(e) = 0$.
  Hence $\sum_{e \in \bar{T}} c(e) = \sum_{e \in \bar{T}} \Big(\bar{y}(\mathcal{S}_e) - \bar{z}_e\Big) = \sum_{S \in \mathcal{S}} \deg_{\bar{T}}(S)\bar{y}_S - \bar{z}(\bar{T})$

$$\text{maximize} \quad 2\sum_{S \in \mathcal{S}} y_S - 1\sum_{e \in E} z_e$$

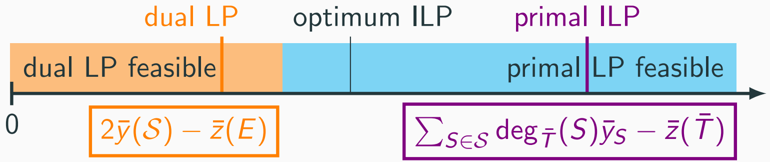
$$y(\mathcal{S}_e) - z_e \leq c(e) \qquad \forall e \in E$$

$$y_S \geq 0 \qquad \forall S \in \mathcal{S}$$

$$z_e \geq 0 \qquad \forall e \in E$$

$\mathcal{S}_e := \{S \in \mathcal{S} \mid e \in \delta_G(S)\}$

The journey to approximation ratio 3 begins...



primal ≤ 3 dual (it follows primal ≤ 3 optimum)

$$\sum_{S\in\mathcal{S}} \deg_{\bar{T}}(S)\bar{y}_S - \bar{z}(\bar{T}) \leq 3(2\bar{y}(\mathcal{S}) - \bar{z}(E))$$

$$\sum_{S\in\mathcal{S}} \deg_{\bar{T}}(S)\bar{y}_S \leq 6\bar{y}(\mathcal{S}) - 3\bar{z}(E) + \bar{z}(\bar{T})$$

$$\sum_{S\in\mathcal{S}} \deg_{\bar{T}}(S)\bar{y}_S \leq 6\bar{y}(\mathcal{S}) - 2\bar{z}(\bar{T}) - 3\bar{z}(E \setminus \bar{T})$$

Induction

Start of algorithm: $\bar{y} = 0, \bar{z} = 0$

The journey to approximation ratio 3 continues...

The i-th grow step:

- $\bar{y}$ and $\bar{z}$ grow only at leaves in the forest F
- Let L be the leaves in F. Partition L into

  L_0 — degree 0 (isolated nodes)

  L_1 — degree 1 and the incident edge is in $\bar{T}$

  L_2 — degree 1 and the incident edge is not in $\bar{T}$
- $\sum_{S \in \mathcal{S}} \deg_{\bar{T}}(S)\bar{y}_S \leq 6\bar{y}(\mathcal{S}) - 2\bar{z}(\bar{T}) - 3\bar{z}(E \setminus \bar{T})$

  $\sum_{v \in L} \deg_{\bar{T}^F}(v)\Delta \leq 6|L|\Delta - 2|L_1|\Delta - 3|L_2|\Delta$

  $\sum_{v \in L} \deg_{\bar{T}^F}(v) \leq 6|L| - 2|L_1| - 3|L_2|$

  if every edge in $\bar{T}^F$ but not in F is necessary in $\bar{T}^F \cup F$

  (otherwise it would have been deleted by the cleanup phase)
- show this graph property and we are set

  (7 of 12 pages in the paper)

Conclusion and open problems

Conclusion

- simple 3-approximation algorithm (tight) for the 2-edge-connected spanning subgraph problem

Open problems

- 2-node-connectivity? generalized edge-connectivity?
- practical evaluation (comparison to complicated algorithms)
- find a primal-dual algorithm with ratio < 3
- find an algorithm with ratio < 2

Thank you!
Questions?