Some Advances in the Planar Directed Steiner Tree Problem

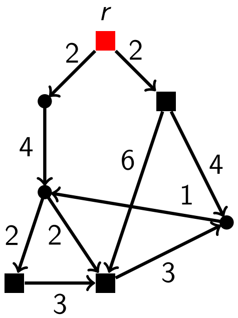
Zachary Friggstad and Ramin Mousavi

University of Alberta

Alberta-Montana Combinatorics and Algorithms Workshop 2022

Definition

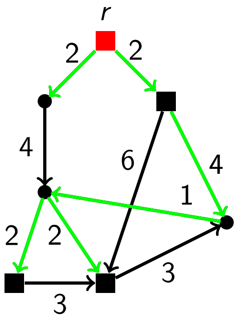
- Given a directed graph $G = (V, E)$, cost on edges, root node r, and a set of terminals $X \subseteq V - r$. The rest of vertices are called Steiner nodes
- Find a min cost subgraph such that every terminal is reachable from the root



- Special cases: Arborescence where $X = V$ or s, t-shortest path where $r = s$ and $X = \{t\}$

Definition

- Given a directed graph $G = (V, E)$, cost on edges, root node r, and a set of terminals $X \subseteq V - r$. The rest of vertices are called Steiner nodes
- Find a min cost subgraph such that every terminal is reachable from the root



- Special cases: Arborescence where $X = V$ or s, t-shortest path where $r = s$ and $X = \{t\}$

What's known?

α-approx for a minimization problem means a polytime alg that finds a solution with cost at most $\alpha \cdot \mathrm{OPT}$.

What's known?

α-approx for a minimization problem means a polytime alg that finds a solution with cost at most $\alpha \cdot \mathrm{OPT}$.

- DST is a generalization of some fundamental combinatorial problems, e.g., Set Cover and Group Steiner Tree!

What's known?

α-approx for a minimization problem means a polytime alg that finds a solution with cost at most $\alpha \cdot \mathrm{OPT}$.

- DST is a generalization of some fundamental combinatorial problems, e.g., Set Cover and Group Steiner Tree!
- Can't do "significantly" better than $\log^{2-\delta} n$ for some fixed $\delta > 0$ Halperin and Krauthgamer - 2003

What's known?

α-approx for a minimization problem means a polytime alg that finds a solution with cost at most $\alpha \cdot \mathrm{OPT}$.

- DST is a generalization of some fundamental combinatorial problems, e.g., Set Cover and Group Steiner Tree!
- Can't do "significantly" better than $\log^{2-\delta} n$ for some fixed $\delta > 0$ Halperin and Krauthgamer - 2003
- Roughly $|X|^{\epsilon}$-approximation for any $\epsilon > 0$ and this the best known Charikar et al. - 1999

What's known?

α-approx for a minimization problem means a polytime alg that finds a solution with cost at most $\alpha \cdot \mathrm{OPT}$.

- DST is a generalization of some fundamental combinatorial problems, e.g., Set Cover and Group Steiner Tree!
- Can't do "significantly" better than $\log^{2-\delta} n$ for some fixed $\delta > 0$ Halperin and Krauthgamer - 2003
- Roughly $|X|^{\epsilon}$-approximation for any $\epsilon > 0$ and this the best known Charikar et al. - 1999
- There is roughly a $\frac{\log^2 |X|}{\log \log |X|}$-approx in quasi-polytime and can't do better under some complexity assumptions Grandoni et al. - 2019

What's known?

α-approx for a minimization problem means a polytime alg that finds a solution with cost at most $\alpha \cdot \mathrm{OPT}$.

- DST is a generalization of some fundamental combinatorial problems, e.g., Set Cover and Group Steiner Tree!
- Can't do "significantly" better than $\log^{2-\delta} n$ for some fixed $\delta > 0$ Halperin and Krauthgamer - 2003
- Roughly $|X|^{\epsilon}$-approximation for any $\epsilon > 0$ and this the best known Charikar et al. - 1999
- There is roughly a $\frac{\log^2 |X|}{\log \log |X|}$-approx in quasi-polytime and can't do better under some complexity assumptions Grandoni et al. - 2019
- Unidrected version has ≈ 1.39-approx Byrka et al. - 2013

DST is hard so what now?!

- Consider restricted instances like planar or quasi-bipartite instances. Quasi-bipartite means no edge between any two Steiner nodes.

DST is hard so what now?!

- Consider restricted instances like planar or quasi-bipartite instances. Quasi-bipartite means no edge between any two Steiner nodes.
- Quasi-bipartite DST admits $\log |X|$-approx (with the same integrality gap) and this is tight Hibi and Fujito - 2012 and Friggstad et al. - 2016

DST is hard so what now?!

- Consider restricted instances like planar or quasi-bipartite instances. Quasi-bipartite means no edge between any two Steiner nodes.
- Quasi-bipartite DST admits $\log |X|$-approx (with the same integrality gap) and this is tight Hibi and Fujito - 2012 and Friggstad et al. - 2016
- Planar and quasi-bipartite instances in the undirected version of DST have a rich literature, e.g. PTAS for planar instances Borradaile et al. - 2009, and ≈ 1.22-approx for quasi-bipartite instances Goemans et al. - 2012

DST is hard so what now?!

- Consider restricted instances like planar or quasi-bipartite instances. Quasi-bipartite means no edge between any two Steiner nodes.
- Quasi-bipartite DST admits $\log |X|$-approx (with the same integrality gap) and this is tight Hibi and Fujito - 2012 and Friggstad et al. - 2016
- Planar and quasi-bipartite instances in the undirected version of DST have a rich literature, e.g. PTAS for planar instances Borradaile et al. - 2009, and ≈ 1.22-approx for quasi-bipartite instances Goemans et al. - 2012

Theorem (Friggstad-M.)

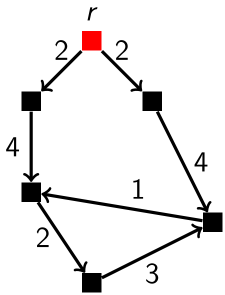
There is a 20-approx for DST on quasi-bipartite, planar instances. We can generalize it to any graph that excludes a fixed minor.

Toolbox: primal-dual

- Primal-dual algorithm is rare in the directed network design problems. One use of this is in Arborescence (more on this in the next slide).
- In contrast, primal-dual algorithm is used in the undirected network design abundantly, e.g. Guha et al. - 1999, Könemann et al. - 2013, Moldenhauer 2013, and Demaine et al. - 2014
- Why primal-dual algorithm is preferred? Can be viewed as combinatorial algorithm and usually fast and easy to implement!

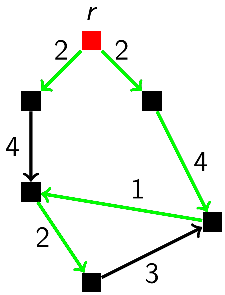
Warm up (Arborescence)

Given a directed graph, edge cost and a root node r. Find a cheapest subgraph such that every node is reachable from r



Warm up (Arborescence)

Given a directed graph, edge cost and a root node r. Find a cheapest subgraph such that every node is reachable from r



Warm up (Arborescence)

$\delta^{in}(S)$ is the set of edges entering S and $x(\delta^{in}(S)) := \sum_{e:e\in\delta^{in}(S)} x_e$

Primal LP

$$\min \sum_{e} c_e \cdot x_e$$
$$x(\delta^{in}(S)) \geq 1, \ \forall S \subseteq V - r$$
$$x \geq 0$$

Dual LP

$$\max \sum_{S} y_S$$
$$\sum_{S:e\in\delta^{in}(S)} y_S \leq c_e, \ \forall e$$
$$y \geq 0$$

Warm up (Arborescence)

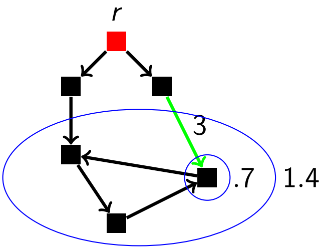
$\delta^{in}(S)$ is the set of edges entering S and $x(\delta^{in}(S)) := \sum_{e:e\in\delta^{in}(S)} x_e$

Primal LP

$$\min \sum_e c_e \cdot x_e$$
$$x(\delta^{in}(S)) \geq 1,\ \forall S \subseteq V - r$$
$$x \geq 0$$

Dual LP

$$\max \sum_S y_S$$
$$\sum_{S:e\in\delta^{in}(S)} y_S \leq c_e,\ \forall e$$
$$y \geq 0$$



Warm up (Arborescence)

$\delta^{in}(S)$ is the set of edges entering S and $x(\delta^{in}(S)) := \sum_{e: e \in \delta^{in}(S)} x_e$

Primal LP

$$\min \sum_{e} c_e \cdot x_e$$
$$x(\delta^{in}(S)) \geq 1, \ \forall S \subseteq V - r$$
$$x \geq 0$$

Dual LP

$$\max \sum_{S} y_S$$
$$\sum_{S: e \in \delta^{in}(S)} y_S \leq c_e, \ \forall e$$
$$y \geq 0$$

- Find a subgraph F

Warm up (Arborescence)

$\delta^{in}(S)$ is the set of edges entering S and $x(\delta^{in}(S)) := \sum_{e: e \in \delta^{in}(S)} x_e$

Primal LP

$$\min \sum_e c_e \cdot x_e$$
$$x(\delta^{in}(S)) \geq 1, \ \forall S \subseteq V - r$$
$$x \geq 0$$

Dual LP

$$\max \sum_S y_S$$
$$\sum_{S: e \in \delta^{in}(S)} y_S \leq c_e, \ \forall e$$
$$y \geq 0$$

- Find a subgraph F
- Find a (fractional) solution $\bar{y}$ for the Dual LP such that:

Warm up (Arborescence)

$\delta^{in}(S)$ is the set of edges entering S and $x(\delta^{in}(S)) := \sum_{e: e \in \delta^{in}(S)} x_e$

Primal LP

$$\min \sum_e c_e \cdot x_e$$
$$x(\delta^{in}(S)) \geq 1, \ \forall S \subseteq V - r$$
$$x \geq 0$$

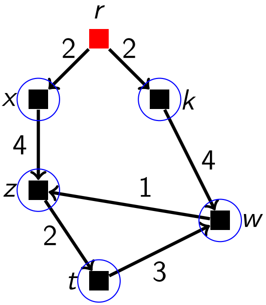
Dual LP

$$\max \sum_S y_S$$
$$\sum_{S: e \in \delta^{in}(S)} y_S \leq c_e, \ \forall e$$
$$y \geq 0$$

- Find a subgraph F
- Find a (fractional) solution $\bar{y}$ for the Dual LP such that:
- $\mathrm{cost}(F) \leq \mathrm{cost}(\bar{y}) \leq$ Dual LP $\leq$ Primal LP $\leq \mathrm{OPT}$

Primal-dual in action!

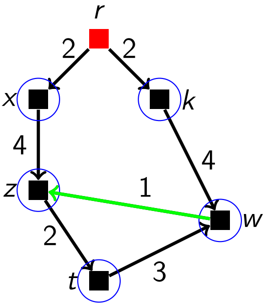
Recall the dual constraint for edge e is $\sum_{S : e \in \delta^{in}(S)} y_S \leq c_e$



- At time 0, every node except r is active set. Increase $y_{\{x\}}, y_{\{k\}}, \ldots$

Primal-dual in action!

Recall the dual constraint for edge e is $\sum_{S:e\in\delta^{in}(S)} y_S \leq c_e$

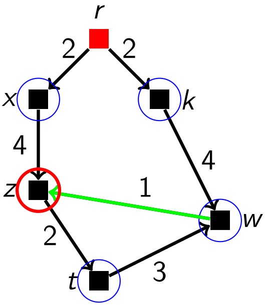


- At time 0, every node except r is active set. Increase $y_{\{x\}}, y_{\{k\}}, ...$
- At time 1, wz is bought and the red set is inactive now.

  $y_{\{x\}} = y_{\{k\}} = y_{\{z\}} = ... = 1$.

Primal-dual in action!

Recall the dual constraint for edge e is $\sum_{S:e\in\delta^{in}(S)} y_S \le c_e$

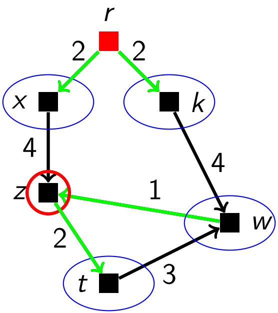


- At time 0, every node except r is active set. Increase $y_{\{x\}}, y_{\{k\}}, \ldots$
- At time 1, wz is bought and the red set is inactive now.

  $y_{\{x\}} = y_{\{k\}} = y_{\{z\}} = \ldots = 1$.

Primal-dual in action!

Recall the dual constraint for edge e is $\sum_{S: e \in \delta^{in}(S)} y_S \leq c_e$



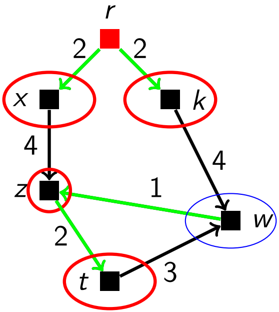
- At time 0, every node except r is active set. Increase $y_{\{x\}}, y_{\{k\}}, \ldots$
- At time 1, wz is bought and the red set is inactive now.

  $y_{\{x\}} = y_{\{k\}} = y_{\{z\}} = \ldots = 1$.
- At time 2, rx, rk and zt are bought and the only active set is the blue one.

  $y_{\{x\}} = y_{\{k\}} = y_{\{w\}} = y_{\{t\}} = 2$, $y_{\{z\}} = 1$.

Primal-dual in action!

Recall the dual constraint for edge e is $\sum_{S: e \in \delta^{in}(S)} y_S \leq c_e$



- At time 0, every node except r is <u>active</u> set. Increase $y_{\{x\}}, y_{\{k\}}, \ldots$
- At time 1, wz is bought and the red set is inactive now.

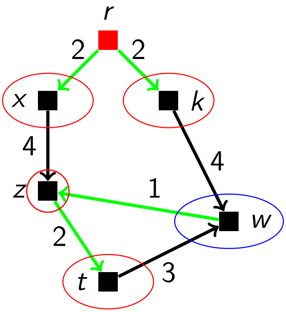
  $y_{\{x\}} = y_{\{k\}} = y_{\{z\}} = \ldots = 1$.
- At time 2, rx, rk and zt are bought and the only active set is the blue one.

  $y_{\{x\}} = y_{\{k\}} = y_{\{w\}} = y_{\{t\}} = 2$, $y_{\{z\}} = 1$.

Primal-dual in action, continued!

Recall the dual constraint for edge e is $\sum_{S:e\in\delta^{in}(S)} y_S \leq c_e$

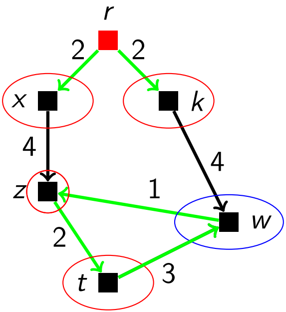
- At time 3, tw is bought
  $y_{\{x\}} = y_{\{k\}} = y_{\{t\}} = 2$,
  $y_{\{z\}} = 1$, $y_{\{w\}} = 3$.



Primal-dual in action, continued!

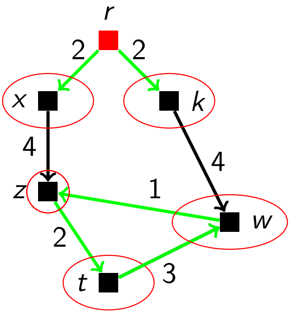
Recall the dual constraint for edge e is $\sum_{S: e \in \delta^{in}(S)} y_S \leq c_e$

- At time 3, tw is bought
  $y_{\{x\}} = y_{\{k\}} = y_{\{t\}} = 2$,
  $y_{\{z\}} = 1$, $y_{\{w\}} = 3$.



Primal-dual in action, continued!

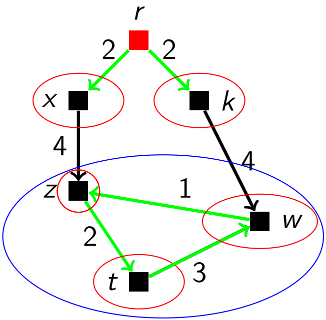
Recall the dual constraint for edge e is $\sum_{S:e\in\delta^{in}(S)} y_S \leq c_e$



- At time 3, tw is bought $y_{\{x\}} = y_{\{k\}} = y_{\{t\}} = 2$, $y_{\{z\}} = 1$, $y_{\{w\}} = 3$.

Primal-dual in action, continued!

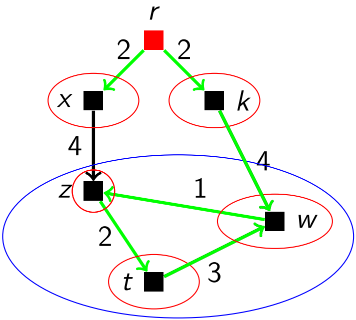
Recall the dual constraint for edge e is $\sum_{S: e \in \delta^{in}(S)} y_S \leq c_e$



- At time 3, tw is bought $y_{\{x\}} = y_{\{k\}} = y_{\{t\}} = 2$, $y_{\{z\}} = 1$, $y_{\{w\}} = 3$.
- z, t, w are strongly connected component, form a bigger active set $\{w, z, t\}$ and $y_{\{w,z,t\}} = 0$

Primal-dual in action, continued!

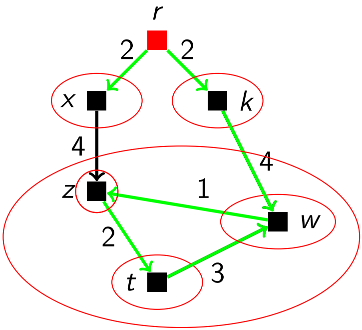
Recall the dual constraint for edge e is $\sum_{S:e\in\delta^{in}(S)} y_S \le c_e$



- At time 3, tw is bought $y_{\{x\}} = y_{\{k\}} = y_{\{t\}} = 2$, $y_{\{z\}} = 1$, $y_{\{w\}} = 3$.
- z, t, w are strongly connected component, form a bigger active set $\{w, z, t\}$ and $y_{\{w,z,t\}} = 0$
- At time 4, kw is bought and there is no active set left. So we terminate. $\{w, z, t\}$. $y_{\{x\}} = y_{\{k\}} = y_{\{t\}} = 2$, $y_{\{z\}} = 1$, $y_{\{w\}} = 3$, $y_{\{w,z,t\}} = 1$.

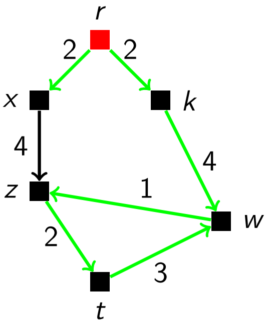
Primal-dual in action, continued!

Recall the dual constraint for edge e is $\sum_{S:e\in\delta^{in}(S)} y_S \leq c_e$



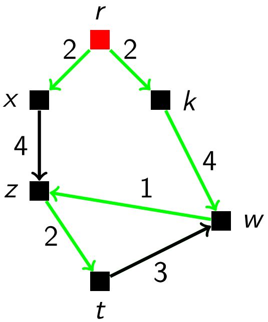
- At time 3, tw is bought $y_{\{x\}} = y_{\{k\}} = y_{\{t\}} = 2$, $y_{\{z\}} = 1$, $y_{\{w\}} = 3$.
- z, t, w are strongly connected component, form a bigger active set $\{w, z, t\}$ and $y_{\{w,z,t\}} = 0$
- At time 4, kw is bought and there is no active set left. So we terminate. $\{w, z, t\}$. $y_{\{x\}} = y_{\{k\}} = y_{\{t\}} = 2$, $y_{\{z\}} = 1$, $y_{\{w\}} = 3$, $y_{\{w,z,t\}} = 1$.

Primal-dual in action, continued!



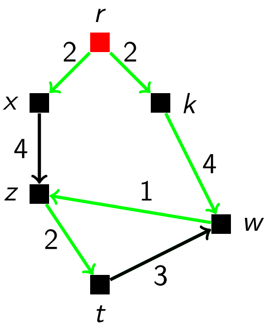
- Edges are bought in the order: wz, rx, rk, zt, tw, and kw. Consider edges in the reverse order they have been added and remove them if not need it!

Primal-dual in action, continued!



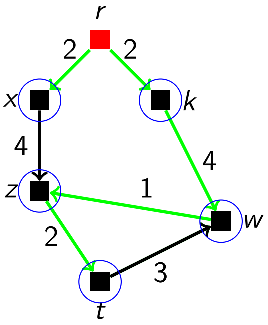
- Edges are bought in the order: wz, rx, rk, zt, tw, and kw. Consider edges in the reverse order they have been added and remove them if not need it!
- Remove tw

Primal-dual in action, continued!



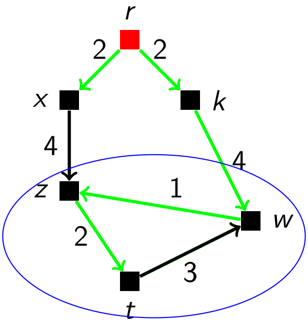
- Edges are bought in the order: wz, rx, rk, zt, tw, and kw. Consider edges in the reverse order they have been added and remove them if not need it!
- Remove tw
- Total dual increased is $y_{\{x\}} + ... + y_{\{z,t,w\}} = 11$. Total cost of green edges is 11 too!

Primal-dual in action, continued!



- Edges are bought in the order: wz, rx, rk, zt, tw, and kw. Consider edges in the reverse order they have been added and remove them if not need it!
- Remove tw
- Total dual increased is $y_{\{x\}} + ... + y_{\{z,t,w\}} = 11$. Total cost of green edges is 11 too!
- Consider an iteration. Every active set has at most one edge entering

Primal-dual in action, continued!



- Edges are bought in the order: wz, rx, rk, zt, tw, and kw. Consider edges in the reverse order they have been added and remove them if not need it!
- Remove tw
- Total dual increased is $y_{\{x\}} + ... + y_{\{z,t,w\}} = 11$. Total cost of green edges is 11 too!
- Consider an iteration. Every active set has at most one edge entering

Back to DST!

Primal LP

$$\min \sum_e c_e \cdot x_e$$
$$x(\delta^{in}(S)) \geq 1, \ \forall S \subseteq V - r, S \cap X \neq \emptyset$$
$$x \geq 0$$

Dual LP

$$\max \sum_S y_S$$
$$\sum_{S : e \in \delta^{in}(S)} y_S \leq c_e, \ \forall e$$
$$y \geq 0$$

Back to DST!

Primal LP

$$\min \sum_e c_e \cdot x_e$$
$$x(\delta^{in}(S)) \geq 1, \ \forall S \subseteq V - r, S \cap X \neq \emptyset$$
$$x \geq 0$$

Dual LP

$$\max \sum_S y_S$$
$$\sum_{S: e \in \delta^{in}(S)} y_S \leq c_e, \ \forall e$$
$$y \geq 0$$

- Find a subgraph F

Back to DST!

Primal LP

$$\min \sum_{e} c_e \cdot x_e$$
$$x(\delta^{in}(S)) \geq 1, \ \forall S \subseteq V - r, S \cap X \neq \emptyset$$
$$x \geq 0$$

Dual LP

$$\max \sum_{S} y_S$$
$$\sum_{S: e \in \delta^{in}(S)} y_S \leq c_e, \ \forall e$$
$$y \geq 0$$

- Find a subgraph F
- Find a (fractional) solution $\bar{y}$ for the Dual LP such that

Back to DST!

Primal LP

$$\min \sum_{e} c_e \cdot x_e$$
$$x(\delta^{in}(S)) \geq 1, \ \forall S \subseteq V - r, S \cap X \neq \emptyset$$
$$x \geq 0$$

Dual LP

$$\max \sum_{S} y_S$$
$$\sum_{S:e \in \delta^{in}(S)} y_S \leq c_e, \ \forall e$$
$$y \geq 0$$

- Find a subgraph F
- Find a (fractional) solution $\bar{y}$ for the Dual LP such that
- $\mathrm{cost}(F) \leq 20 \cdot \mathrm{cost}(\bar{y}) \leq 20 \cdot \mathrm{OPT}$

Back to DST!

Primal LP

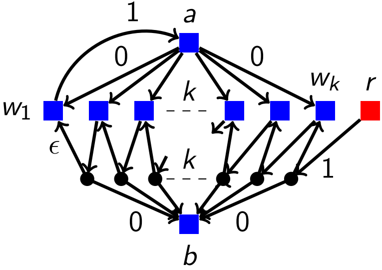
$$\min \sum_e c_e \cdot x_e$$
$$x(\delta^{in}(S)) \geq 1, \ \forall S \subseteq V - r, S \cap X \neq \emptyset$$
$$x \geq 0$$

Dual LP

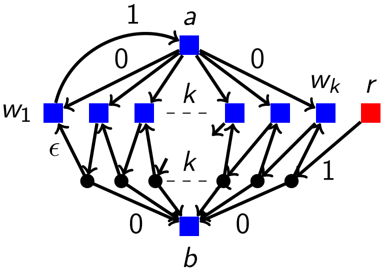
$$\max \sum_S y_S$$
$$\sum_{S : e \in \delta^{in}(S)} y_S \leq c_e, \ \forall e$$
$$y \geq 0$$

- Find a subgraph F
- Find a (fractional) solution $\bar{y}$ for the Dual LP such that
- $\mathrm{cost}(F) \leq 20 \cdot \mathrm{cost}(\bar{y}) \leq 20 \cdot \mathrm{OPT}$
- Natural thing to try is to use the "same" primal-dual algorithm for Arborescence here!

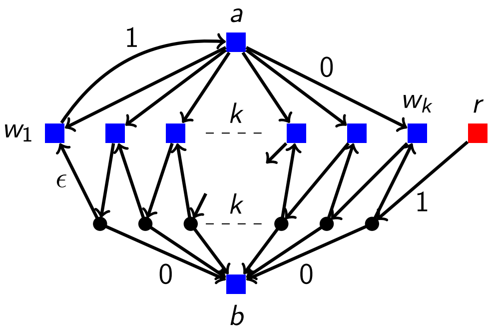
There is always an obstacle!



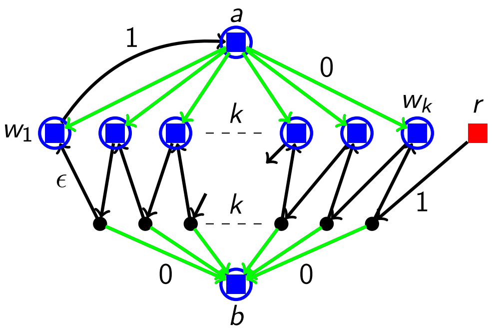
There is always an obstacle!



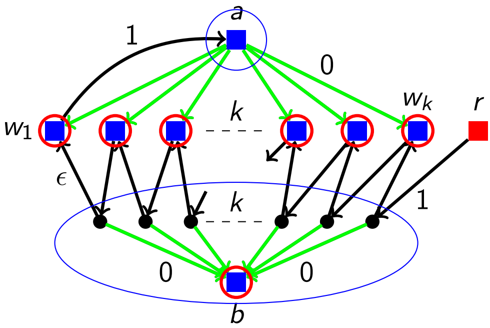
There is always an obstacle!



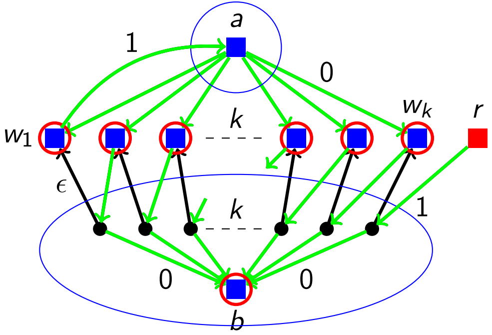
There is always an obstacle!



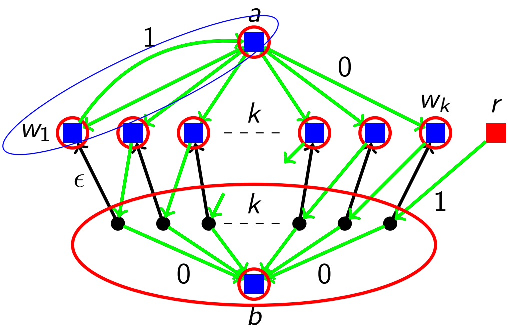
There is always an obstacle!



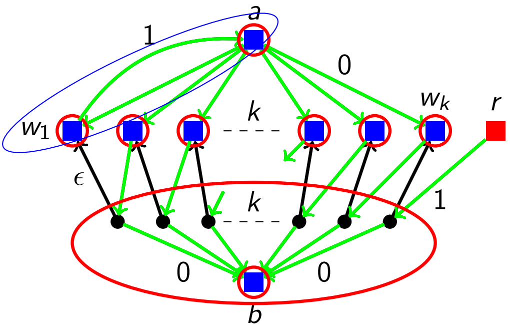
There is always an obstacle!



There is always an obstacle!

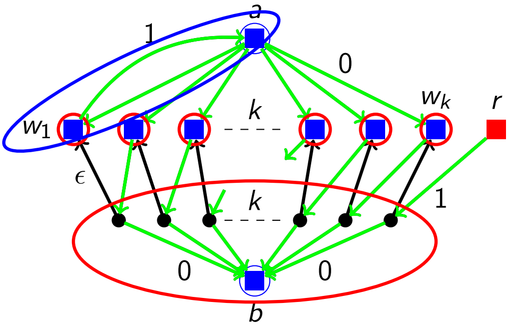


There is always an obstacle!



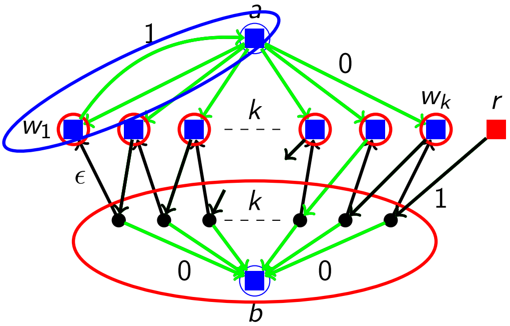
- Bottom set bought too many edges that aren't used for its connectivity

But sometimes there is a bypass!



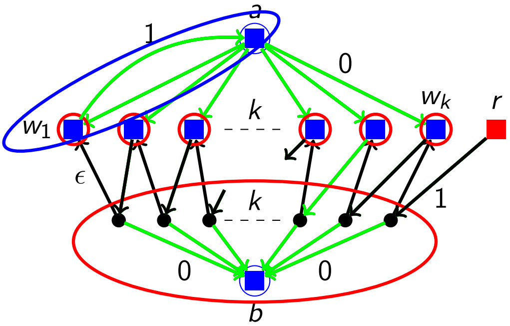
- Make the bottom set to purchase only one of the downward edges.

But sometimes there is a bypass!



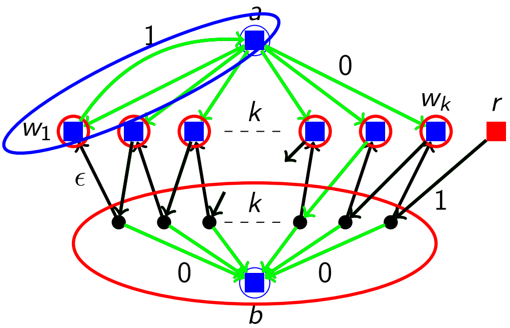
- Make the bottom set to purchase only one of the downward edges.
- The top active set will buy the rest of the downward edges as it grows.

But sometimes there is a bypass!



- Make the bottom set to purchase only one of the downward edges.
- The top active set will buy the rest of the downward edges as it grows.
- How to formalize this?

But sometimes there is a bypass!



- Make the bottom set to purchase only one of the downward edges.
- The top active set will buy the rest of the downward edges as it grows.
- How to formalize this?
- Introducing two buckets: expansion and killer!

But sometimes there is a bypass!

- Where does planarity is used?!

But sometimes there is a bypass!

- Where does planarity is used?!
  - The average degree of active sets w.r.t. final solution is constant

But sometimes there is a bypass!

- Where does planarity is used?!
  - The average degree of active sets w.r.t. final solution is constant
  - We actually need every minor of the graph also has a constant average degree

Open problems

- Planar DST?
- Planar DAG DST?
- Other applications of multiple buckets typed primal-dual algorithm?

Open problems

- Planar DST?
- Planar DAG DST?
- Other applications of multiple buckets typed primal-dual algorithm?

Thank You

Bonus! The analysis

- Fix an iteration

Bonus! The analysis

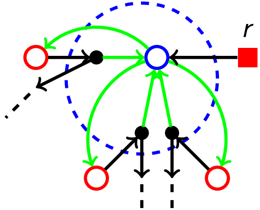
- Fix an iteration
- Every active set has at most one killer edge entering it

Bonus! The analysis

- Fix an iteration
- Every active set has at most one killer edge entering it
- What about expansion edges?

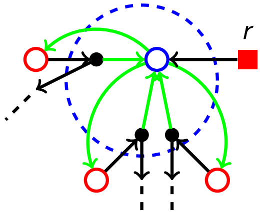
Bonus! The analysis

- Fix an iteration
- Every active set has at most one killer edge entering it
- What about expansion edges?



Bonus! The analysis

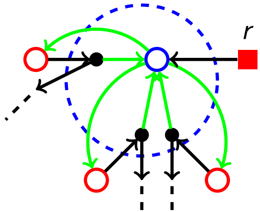
- Fix an iteration
- Every active set has at most one killer edge entering it
- What about expansion edges?



- Charge the expansion edges to an active set down the road!

Bonus! The analysis

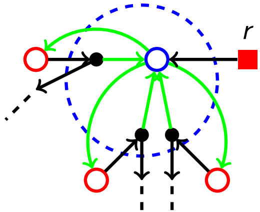
- Fix an iteration
- Every active set has at most one killer edge entering it
- What about expansion edges?



- Charge the expansion edges to an active set down the road!
- Where does planarity is used?!

Bonus! The analysis

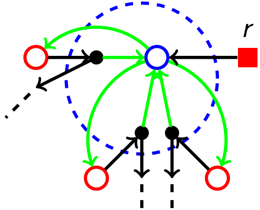
- Fix an iteration
- Every active set has at most one killer edge entering it
- What about expansion edges?



- Charge the expansion edges to an active set down the road!
- Where does planarity is used?!
  - Recall relation between average degree of active sets and performance guarantee

Bonus! The analysis

- Fix an iteration
- Every active set has at most one killer edge entering it
- What about expansion edges?



- Charge the expansion edges to an active set down the road!
- Where does planarity is used?!
  - Recall relation between average degree of active sets and performance guarantee
  - We actually need every minor of the graph has constant average degree