Primal Dual Method

WEIGHTED SET COVER: Ground Set $E = \{e_1 \dots e_n\}$

Subsets $S_1, \dots S_m \subseteq E$, S_j has weight w_j

Find $I \subseteq \{1 \dots m\}$ st $\sum_{j \in I} w_j$ is minimized and $\bigcup_{j \in I} S_j = E$

LP: $\min \sum_{j \in [m]} w_j x_j$

such that $\forall e_i \in E \quad \sum_{\substack{j \in [m] \\ \text{st } e_i \in S_j}} x_j \geq 1$

$x_j \geq 0$

Primal LP

Assume any $e_i \in E$ belongs to at most f subsets.

Primal LP

$$\min \sum_{j \in [m]} w_j x_j$$

such that $\forall e_i \in E \quad \sum_{\substack{j \in [m] \\ \text{st } e_i \in S_j}} x_j \geqslant 1$ $\quad\Big|\quad x_j \geqslant 0$

Dual LP :

Best Lower Bound on Primal Objective as a linear combination of Primal Constraints.

$$\max \sum_{i \in [n]} 1 \cdot y_i \leqslant \sum_{i \in [n]} y_i \Big(\sum_{\substack{j \in [m] \\ e_i \in S_j}} x_j \Big) \leqslant \sum_{j \in [m]} w_j x_j$$

want this

$\Downarrow$

Dual Constraints.

such that $\forall S_j \quad \sum_{\substack{i \in [m] \\ e_i \in S_j}} y_i \leqslant w_j$

and $\quad y_i \geqslant 0$

if not, then don't get a lower bound

f-approx via Primal Dual method

Key advantage:

No need to solve LP optimally

$\Rightarrow$ much faster

- $y \leftarrow 0$, $I \leftarrow \phi$
- while $\exists\, e_i \notin \bigcup_{j \in I} S_j$
  - increase dual variable y_i until some l with $e_i \in S_l$ s.t.

$$\sum_{j \text{ st } e_j \in S_l} y_j = w_l$$

increase dual var until some constraint becomes tight

- $I \leftarrow I \cup \{l\}$

Theorem: Above I is an f-approximation.

Proof: $\sum_{l \in I} w_l = \sum_{l \in I} \sum_{e_i \in S_l} y_i = \sum_{e_i} |\{l \in I \mid e_i \in S_l\}|\, y_i$

at most f subsets contain e_i

$$\leq f \sum_{e_i} y_i$$

$$\leq f\,(\text{Primal OPT})$$

$$\leq f\,(\text{Integral Primal OPT})$$

We rewrite the cost of Primal Sol^n in terms of the dual vars.

- The approx factor of a Primal-Dual algo is also an upperbound on the integrality gap of the primal LP

- Choosing the right dual variable to increase.

UNDIRECTED FEEDBACK VERTEX SET

Given $G(V,E)$ with weight $w_i \geq 0$ for vertex $i \in V$, find min-weight $S \subseteq V$ st every cycle C intersects S.

LP: $\min \sum_{i \in V} w_i x_i$

st $\forall$ cycle C, $\sum_{i \in C} x_i \geq 1$

$x_i \geq 0$

exponentially many constraints.

Dual LP: $\max \sum_{C \in \mathcal{C}} y_C$

set of all cycles in G

st $\forall$ vertex $i \in V$,

$\sum_{C \ni i} y_C \leq w_i$

$y_C \geq 0$

- In applying the Primal-Dual algo, we only need to find a cycle that is not yet hit by S

- If we choose cycle C in $G[V-S]$ arbitrarily

$$\sum_{i \in S} w_i = \sum_{i \in S} \sum_{C \ni i} y_C = \sum_C \underbrace{|S \cap C|}_{\text{could be } O(n)} y_C$$

$\Downarrow$

Choose a short cycle $|C| \leq \alpha$

$\Downarrow$

not always possible BUT

Obs: Any maximal deg-2 path P has ≤ 1 vertex in S

Obs 7.2 (Reading Exercise)

$\Leftarrow$

Choose a cycle minimizing deg ≥ 3 vertices

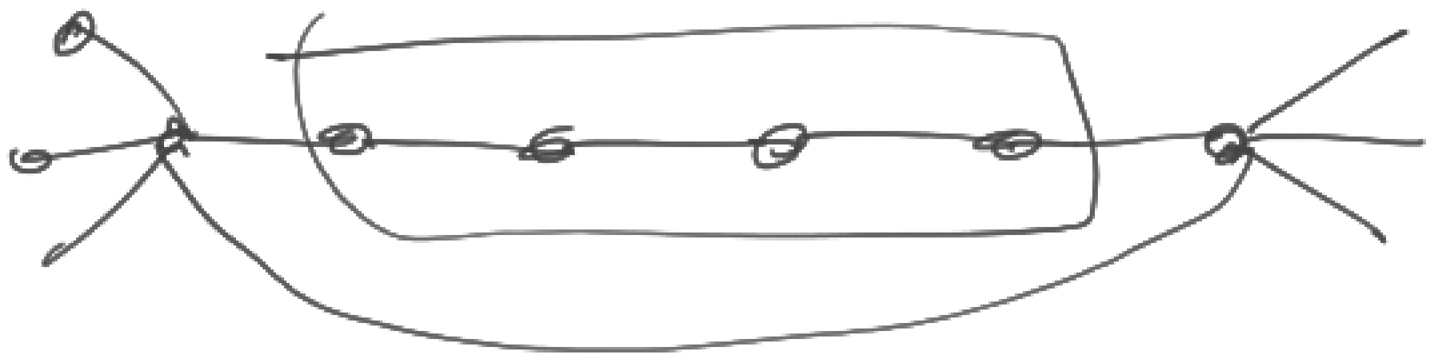
$\Downarrow$

$|C \cap S| = 4 \log n$

Lemma 7.3 (Reading Exercise)

$\Leftarrow$

$4 \log n$ approx



CHEAPEST s-t PATH

- Graph $G(V,E)$, cost c_e $\forall$ edges $e \in E$, $s,t \in V$
- Find a min-cost path from s to t in G

Polytime via Dijkstra

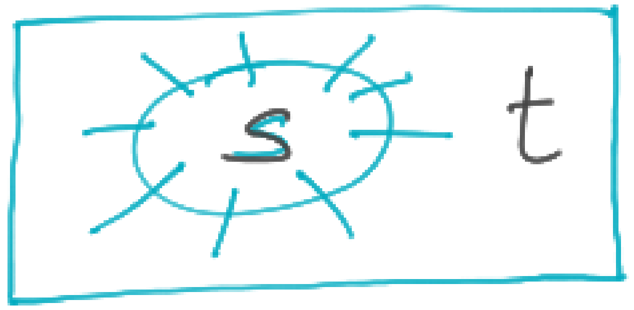
LP: $$\min \sum_{e \in E} c_e x_e$$

st $\forall\, S \subseteq V,\ s \in S,\ t \notin S$

$$\sum_{e \in \delta(S)} x_e \geq 1$$

$\hat{S}$ is the collection of all such $S \subseteq V$

edges with only one endpt in S



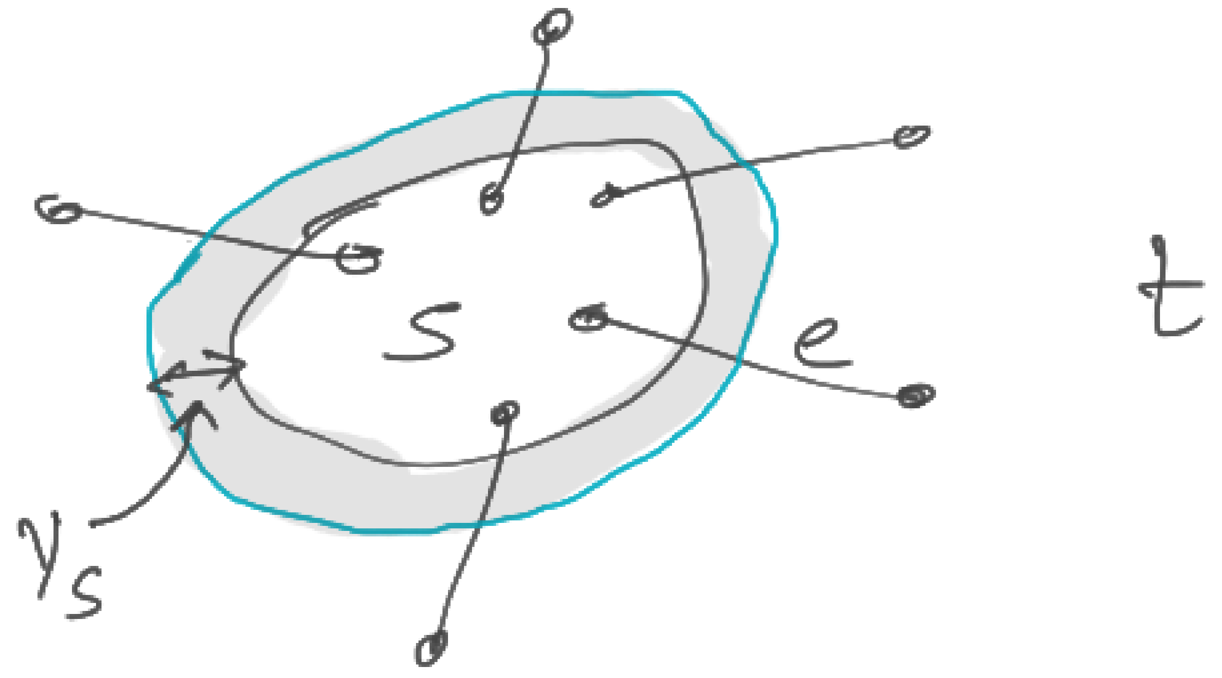
Dual LP:

$$\max \sum_{S \in \hat{S}} y_S \leq \sum_{S} y_S \Big(\sum_{e \in \delta(S)} x_e \Big) \leq \sum_{e} c_e x_e$$

$$\text{st} \sum_{S \mid e \in \delta(S)} y_S \leq c_e \quad \forall e \in E$$

$\boxed{\text{CUT LPs}}$

SHORTEST s-t PATH



- S defines an s-t cut
- y_S is the width of a "moat" around S
- The cost or length of e should be more than total width of all moats it crosses

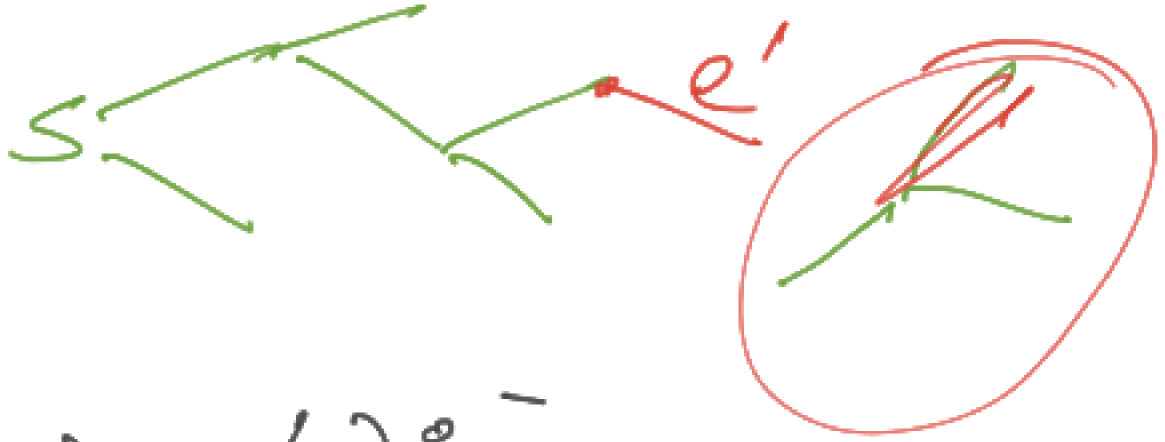
Dual LP:

$$\max \sum_{S \in \hat{S}} y_S \leq \sum_S y_S \Big(\sum_{e \in \delta(S)} x_e\Big) \leq \sum_e c_e x_e$$

$$\text{st} \sum_{S \mid e \in \delta(S)} y_S \leq c_e \quad \forall e \in E$$

$\rightarrow$ Any s-t path crosses all moats and has length $\geq \sum y_S$

Primal Dual Algo:



- $y \leftarrow 0$, $F \leftarrow \emptyset$ ← sol^n s-t path.
- Find a violated cut $C \in \hat{S}$ $(F \cap \delta(C) = \emptyset)$:
  - increase y_C until some dual constraint for $e' \in E$ is tight
  - add e' to F

Let us choose C = vertices reachable from s using only the edges in F

Obs: F is a tree and when we stop it has an s-t path.

Thm: The s-t path $P \subseteq F$ is of min-cost.

Pf: $\sum_{e \in P} c_e = \sum_{e \in P} \sum_{e \in \delta(S)} y_S = \sum_{S \in \hat{S}} |P \cap \delta(S)| y_S = \sum_{S \in \hat{S}} y_S \leq OPT$

Claim: $y_S > 0 \Rightarrow |P \cap \delta(S)| = 1$

S (or C) is chosen as vertices spanned by the current tree F

↳ $y_S > 0$ only for these and only one $e' \in \delta(S)$ is added to F

STEINER FOREST

NP-Hard

- Graph $G(V,E)$, cost c_e $\forall e \in E$, k vertex pairs $\{s_i, t_i\}$.
- Find a min-cost subset of edges that connects every pair
- for a pair $\{s_i, t_i\}$, $\hat{S}_i = \{S \subseteq V \mid s_i \in S,\ t_i \notin S\}$

all $s_i - t_i$ cuts

$$\min \sum_{e \in E} c_e x_e$$

such that

$$\forall i \in [k],\ \forall S \in \hat{S}_i$$

$$\sum_{e \in \delta(S)} x_e \geq 1$$

$$x_e \geq 0$$

$$\max \sum_{S \in \cup \hat{S}_i} y_S$$

"moats"

$$\text{s.t. } \forall e \in E$$

$$\sum_{S \mid e \in \delta(S)} y_S \leq c_e$$

$$y_S \geq 0$$

Primal Dual Algo: Find a cut C st $\delta(C) \cap F = \emptyset$

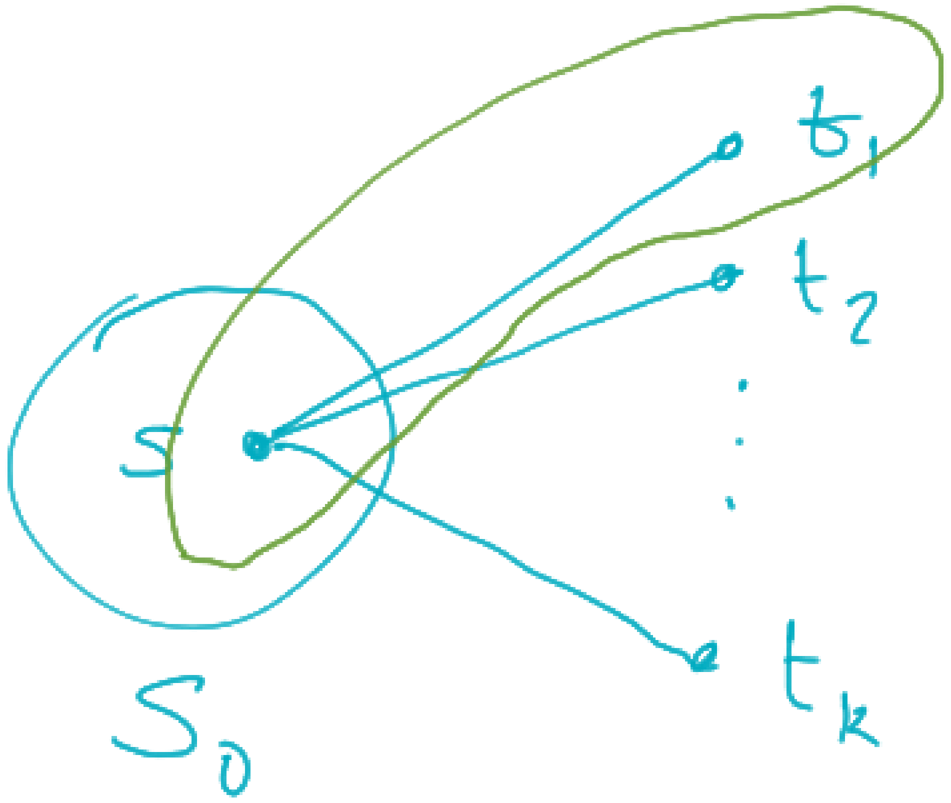
increase y_C until some constraint for $e' \in E$ is tight

add e' to F

- Bound the cost of F as:

$$\sum_{e \in F} c_e = \sum_{e \in F} \sum_{S \mid e \in \delta(S)} y_S = \sum_S |\delta(S) \cap F| \, y_S \leq \alpha \, OPT$$

$$\alpha = \max_S \{ |\delta(S) \cap F| \}$$



It is possible that $\alpha = k$

$$|\delta(S_0) \cap F| = k$$

But $\delta(S_0)$ and $\delta(S_1)$ overlap, so above eqn is overcounting.

- Another Primal Dual Algo:
  - while some $s_i - t_i$ pair is disconnected:
    - $\hat{C}$ = all connected components C of (V, F) such that $|C \cap \{s_i, t_i\}| = 1$ for some $i \in [k]$
    - simultaneously increase $y_C \ \forall \ C \in \hat{C}$ until $c_{e_\ell} = \sum_{S | e_\ell \in \delta(S)} y_S$ for some $e_\ell \in E$
    - add e_ℓ to F and $\ell \leftarrow \ell+1$
- Starting from the newest edge in F, delete unnecessary edges
- As before: $\sum_{e \in F} c_e = \sum_{e \in F} \sum_{S | e \in \delta(S)} y_S = \sum_S |F \cap \delta(S)| y_S \leq 2 \sum_S y_S$ $\leq 2OPT$

Claim: $\sum_S |F \cap \delta(S)| y_S \leq 2 \sum_S y_S$

$$\sum_S |F \cap \delta(S)| y_S \leq 2 \sum_S y_S$$

Proof: Induction on ℓ.

- Initially $\ell = 0$, $y = 0$ so true
- Suppose true till iteration $\ell - 1$
- In the ℓ-th iteration:
  - for every $C \in \hat{C}$ we increase y_C by ε
    - LHS increases by $\varepsilon \sum_{C \in \hat{C}} |F \cap \delta(C)|$
    - RHS increases by $2\varepsilon |\hat{C}|$

- Claim: $\sum_{C \in \hat{C}} |F \cap \delta(C)| \leq 2 |\hat{C}|$ for any iteration

so increase in LHS $\leq$ increase in RHS

$\therefore$ after ℓ-th iter $\sum_S |F \cap \delta(C)| y_S \leq 2 \sum_S y_S$

Proof of claim: $\left(\sum_{C \in \hat{C}} |F \cap \delta(C)| \leq 2|\hat{C}| \right)$

- Obs: F is always a forest. (by our choice of $\hat{C}$ and e_ℓ in every iteration)

- Consider $F_\ell \subseteq F$, $H = F - F_\ell$ $(F_\ell = \{e_1, \ldots e_{\ell-1}\})$
- contract each conn comp of (V, F_ℓ) to one vertex in the forest $(V, F) \longrightarrow$ New Forest
  - some of these vertices correspond to $\hat{C}$ (Red R)
  - other vertices are Blue (B)
  - we want $\sum_{v \in R} \deg(v) \leq 2|R|$ (in the New Forest whose edges are H)

$$\sum_{v \in R} \deg(v) = \sum_{v \in R \cup B} \deg(v) - \sum_{v \in B} \deg(v)$$

$$\leq 2(|R| + |B|) - 2|B| = 2|R|$$

<u>Claim:</u> Any vertex in B has $\deg \geqslant 2$ $\qquad (V, F_\ell)$

<u>Pf:</u> Suppose $v \in B$ has $\deg = 1$

- $C_v \rightarrow$ connected comp of (V, F_ℓ) for v
- some edge $e \in H = F - F_\ell$ is incident on C_v
- Observe that C_v is not an $s_i - t_i$ cut
  OR e is not on any $s_i - t_i$ path in (V, F)

But then we would have deleted e in the final pruning stage

Hence, such a $v \in B$ doesn't exist.