Approximation Algorithms

Ruben Hoeksma (University of Bremen)

July 2, 2019

The primal-dual method

Set cover

LP Duality (again)

Complementary slackness

Let $(P) = \min\{c^T x \mid Ax \geq b\}$, be the primal and $(D) = \max\{b^T y \mid A^T y \leq c\}$ the corresponding dual. If $x_j > 0$, then $\sum_{i=1}^m a_{ij} y_i = c_j$. If $y_i > 0$, then $\sum_{j=1}^n a_{ij} x_j = b_i$.

Lemma

If x^* is a feasible solution to (P) and y^* is a feasible solution to (D), then x^* and y^* satisfy complementary slackness if and only if both are optimal solutions.

Proof

$$\sum_{j=1}^n c_j x_j \geq \sum_{j=1}^n \left(\sum_{i=1}^m a_{ij} y_i \right) x_j = \sum_{i=1}^m \left(\sum_{j=1}^n a_{ij} x_j \right) y_i \geq \sum_{i=1}^m b_i y_i$$

Apply strong duality.

Primal-dual idea

Construction

Construct primal and dual solutions at the same time:

- Maintain a feasible dual solution throughout the algorithm.
- Algorithm terminates with a feasible primal solution.

Proof

Use "approximate complementary slackness" to prove the approximation guarantee α:

If $x_j > 0$, then $\sum_{i=1}^{m} a_{ij} y_i = c_j$. If $y_i > 0$, then $\sum_{j=1}^{n} a_{ij} x_j \leq \alpha b_i$.

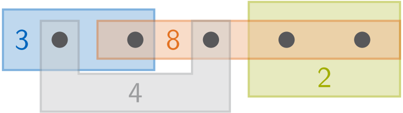
Then,

$$\sum_{j=1}^{n} c_j x_j = \sum_{j=1}^{n} \left(\sum_{i=1}^{m} a_{ij} y_i \right) x_j = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j \right) y_i \leq \alpha \sum_{i=1}^{m} b_i y_i$$

The Set Cover problem

Input: elements E, sets $\mathcal{S} \subseteq 2^E$, weights $w : \mathcal{S} \to \mathbb{R}_+$

Task: find $\mathcal{S}' \subseteq \mathcal{S}$ with $\bigcup_{S \in \mathcal{S}'} S = E$
minimizing $\sum_{S \in \mathcal{S}'} w(S)$



Primal:

$$\begin{aligned} \min \quad & \sum_{S \in \mathcal{S}} w(S) x(S) & \\ \text{s.t.} \quad & \sum_{S : e \in S} x(S) \geq 1 & \forall\, e \in E \\ & x(S) \geq 0 & \forall S \in \mathcal{S} \end{aligned}$$

Dual:

$$\begin{aligned} \max \quad & \sum_{e \in E} y(e) & \\ \text{s.t.} \quad & \sum_{e \in S} y(e) \leq w(S) & \forall\, S \in \mathcal{S} \\ & y(e) \geq 0 & \forall e \in E \end{aligned}$$

Primal-dual for Set Cover

Algorithm:

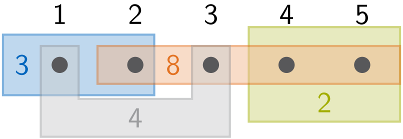
1. Initialize $y(e) = 0$ for all $e \in E$.
2. while ($\exists$ uncovered element e')
   Increase $y(e')$ until a set S with $e' \in S$ becomes tight.
   Add S to $\mathcal{S}'$. (increase can be 0)
3. Return $\mathcal{S}'$.

Theorem

Primal-dual is an f-approximation algorithm for Set Cover.

$$f = \max_i |\{j \in \mathcal{S} : i \in S_j\}|$$

Example



Primal:

$$\begin{aligned} \min \quad & 3 \cdot x_1 + 2 \cdot x_2 + 8 \cdot x_3 + 4 \cdot x_4 \\ \text{s.t.} \quad & x_1 + x_4 \geq 1 \\ & x_1 + x_3 \geq 1 \\ & x_3 + x_4 \geq 1 \\ & x_2 + x_3 \geq 1 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

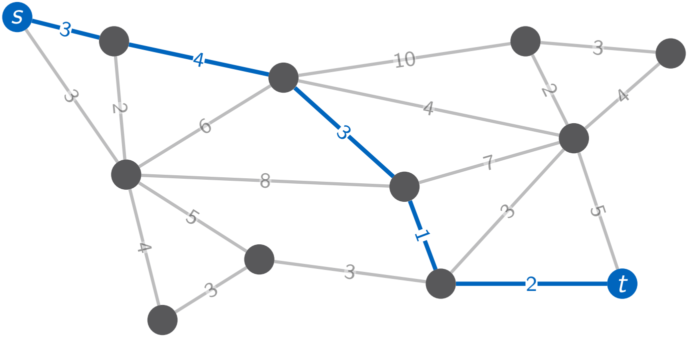
Dual:

$$\begin{aligned} \max \quad & y_1 + y_2 + y_3 + y_4 + y_5 \\ \text{s.t.} \quad & y_1 + y_2 \leq 3 \\ & y_4 + y_5 \leq 2 \\ & y_2 + y_3 + y_4 + y_5 \leq 8 \\ & y_1 + y_3 \leq 4 \\ & y_1, y_2, y_3, y_4, y_5 \geq 0 \end{aligned}$$

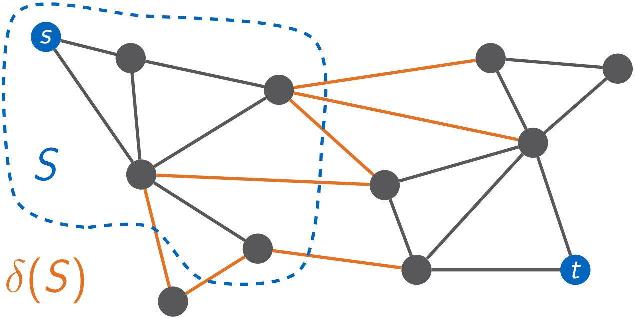
The shortest path problem

Input: graph $G = (V, E)$, weights $w : E \to \mathbb{R}_+$, start $s \in V$, target $t \in V$

Task: find an s-t-path P minimizing $\sum_{e \in P} w_e$



LP relaxation



$$\mathcal{S} := \{S \subseteq V \,:\, s \in S,\ t \notin S\}$$

$$\begin{aligned} \min \quad & \sum_{e \in E} w_e x_e \\ \text{s.t.} \quad & \sum_{e \in \delta(S)} x_e \ \geq\ 1 && \forall\, S \in \mathcal{S} \\ & x_e \ \in \{0,1\} \ \geq\ 0 && \forall\, e \in E \end{aligned}$$

$$\begin{aligned} \max \quad & \sum_{S \subseteq \mathcal{S}} y_S \\ \text{s.t.} \quad & \sum_{S \subseteq \mathcal{S} : e \in \delta(S)} y_S \ \leq\ w_e && \forall\, e \in E \\ & y_S \ \geq\ 0 && \forall\, S \in \mathcal{S} \end{aligned}$$

Primal-dual algorithm

Algorithm

1. $T := \emptyset$, $y := 0$
2. while T does not contain s-t-path
   - Let C be the set of vertices connected to s in (V, T).
   - Increase y_C until there is an $e \in \delta(C)$ with $\sum_{e \in \mathcal{S}_e} y_S = w_e$.
   - $T := T \cup \{e\}$ (increase can be 0)
3. Let P be an s-t-path in T.
4. return P

Theorem

Primal-dual is a 1-approximation algorithm for Shortest Path.