CS787: Advanced Algorithms	**Scribe:** Jing-Yao Chen
Lecture 10: Primal Dual Algorithms	**Date:** 2/26/18

10.1 Primal Dual Formulation

Primal (P):

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax \geq b \\ & x \geq 0. \end{aligned}$$

Dual (D):

$$\begin{aligned} \text{maximize} \quad & b^T y \\ \text{subject to} \quad & y \geq 0 \\ & A^T y \leq c. \end{aligned}$$

$c, x \in \mathbb{R}^{n \times 1}$; $b, y \in \mathbb{R}^{m \times 1}$; $A \in \mathbb{R}^{m \times n}$.

Dual feasible set ——●—— Primal feasible set

P-OPT=D-OPT

The dual feasible set gives a lower bound for primal-OPT.

10.2 Complementary Slackness

1. x^* and y^* are optimal solutions to P and D
2. Slackness
   (a) For all $j \in [m]$, either $y_j^* = 0$ or $(Ax^*)_j = b_j$
   (b) For all $i \in [n]$, either $x_i^* = 0$ or $(A^T y^*)_i = c_j$

Both statements are equivalent. From complementary slackness, it follow that

$$c^T x \geq y^T A x \geq y^T b = b^T y \tag{10.2.1}$$

10.3 Approximation Complementary Slackness

Claim 10.3.1 *Let $\alpha, \beta \geq 1$ be the complementary slackness constraints. If x and y are feasible solutions to P and D, respectively, and satisfy (α, β)-approximate complementary slackness, then the solutions are $\alpha\beta$ approximately optimal for P and D, respectively.*

Proof: The complementary slackness can be relaxed using α and β to the following forms,

- For all $j \in [m]$, either $y_j = 0$ or $(Ax)_j \leq \alpha b_j$

- For all $i \in [n]$, either $x_i = 0$ or $(A^T y)_i \geq c_i/\beta$

As a results, Equation 10.2 can be rewritten as

$$\frac{1}{\beta} c^T x \leq (y^T A)x = y^T (Ax) \leq \alpha y^T b \tag{10.3.2}$$

$$c^T x \leq \alpha\beta(y^T b) \tag{10.3.3}$$

x and y are $\alpha\beta$ approximately optimal solutions to P and D. ■

10.4 Weighted Vertex Cover Primal Dual Formulation

Primal (P):

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_v w_v x_v & \\ \text{subject to} & x_u + x_v \geq 1 & \forall (u,v) \in E \\ & x_v \geq 0 & \forall v \in V \end{array}$$

Dual (D):

$$\begin{array}{lll} \text{maximize} & \displaystyle\sum_{(u,v)\in E} y_{uv} & \\ \text{subject to} & y_{uv} \geq 0 & \forall (u,v) \in E \\ & \displaystyle\sum_{u:(u,v)\in E} y_{uv} \leq w_v & \forall v \in V. \end{array}$$

The integral version of the dual problem is the optimal matching.

10.5 Algorithm For Finding Candidate Solutions

- Start with $x = 0$ and $y = 0$
  x is infeasible, y is feasible
- Until x becomes feasible
  - Consider some constraint j in P that is not satisfied
  - Raise y_j until some constraint i in D becomes tight
  - Raise x_i so that all its P constraints get satisfied

The invariables are: y is feasible and $\forall i$ $x_i = 0$ or $(A^T y)_i = c_i$.

10.6 Weighted Vertex Cover Candidate Solution Algorithm

The goal of the algorithm is to find candidate set from both primal and dual problems. The gap between the candidates $\geq$ approximation ratio.

- Start with $x_v = 0$ $\forall v$ and $y_{uv} = 0$ $\forall (u,v)$
- While there exists uncovered edge (u,v)
  - Raise y_{uv} until either $\sum_{a:(a,v)\in E} y_{av} = w_v$ or $\sum_{a:(a,u)\in E} y_{au} = w_u$

- Set $x_v = 1$ (or $x_u = 1$ or both)
- Freeze all y's incident on v (or u)

Claim 10.6.1 *x and y are both feasible at the end*

Proof: The invariant of the algorithm is that y_{uv} is always feasible since setting y_{uv} satisfies the constraints that $y_{uv} \geq 0$ and $\sum_{u:(u,v)\in E} y_{uv} \leq w_v \quad \forall v \in V$. x is also feasible at the end since the algorithm only terminates when there are no more uncovered edges. ∎

Claim 10.6.2 *The complementary slackness condition always holds.*

Proof: y_{uv} is raised so that the constraint becomes tight. Then the corresponding constraint in x_v is set to 1; thereby, fulfilling the constraints in x_v. ∎

Claim 10.6.3 *For every $(u, v) \in E$, $x_u + x_v \leq 2$.*

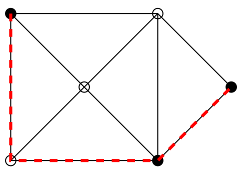
Proof: For an edge (u, v), at most two vertices are selected in the vertex cover. Therefore, α is 2. ∎

Since the second complementary slackness $(A^T y)_i = c_i$ always holds in the algorithm, $\beta = 1$. This is a 2-approximation algorithm.

10.7 Steiner Tree Problem

10.7.1 Steiner Tree Problem

Given a graph with edge weights $w_e \geq 0$ and a set of terminials $T \subseteq V$. The goal is to find the cheapest connected network spanning T. The solutions are always a tree. The Steiner vertices refer to vertices in the solution but not in T.



To write down the primal and dual problems, the constraints are obtained from the observation that any cut in the graph that separates at least one node in T should have non-zero flow between the two separated subsets. Let each cut be defined by one of the two subsets as a result of the cut. Let $\mathcal{S} = \{S : S \subseteq V, 1 \leq |S \cap T| < |T|\}$ denote the group of subsets that separate the terminal. Let $\delta(S) = \{e = \{u, v\} : u \in S, v \in V/S\}$ denote the edges that crosses the cut.

Primal (P):

$$\begin{array}{lll} \text{minimize} & \sum_{e} w_e x_e & \\ \text{subject to} & x_e \geq 0 & \forall e \in E \\ & \sum_{e\in\delta(S)} x_e \geq 1 & \forall S \in \mathcal{S} \end{array}$$

Dual (D):

$$\begin{array}{lll} \text{maximize} & \sum_{S\in\mathcal{S}} y_S & \\ \text{subject to} & y_S \geq 0 & \forall S \in \mathcal{S} \\ & \sum_{S:\delta(S)\ni e} y_S \leq w_e & \forall e \in E \end{array}$$

10.7.2 Steiner Forest Problem

Given a graph with edge weights $w_e \geq 0$ and a set of terminial pairs $T \subseteq V \times V$. The goal is to tfind the cheapest network such that every pair in T is connected in network.