Notes on Primal-Dual Method For Approximation and Network Design

Xilin Yu

July 27, 2017

This is a note on Chapter 4 THE PRIMAL-DUAL METHOD FOR APPROXIMATION ALGORITHMS AND ITS APPLICATION TO NETWORK DESIGN PROBLEMS by Michel X. Goemans and David P. William.

1 General IP model for Network Design Problems

Given a graph $G = (V, E)$, nonnegative costs on edges $c : E \to \mathbb{R}_+$, and a set function $f : 2^V \to \mathbb{N}$. A network design problem can be modeled with the following IP:

$$
\begin{aligned}
\text{(IP)} \qquad & \min \sum_{e \in E} c_e x_e \\
& \text{subject to:} \\
& \sum_{e \in \delta(S)} x_e \geq f(S) && \emptyset \neq S \subset V \\
& x_e \in \{0, 1\} && e \in E.
\end{aligned}
$$

2 LP and Dual LP

We focus on 0-1 functions and have the following relaxed primal and dual LP:

$$
\begin{aligned}
\text{(LP)} \qquad & \min \sum_{e \in E} c_e x_e \\
& \text{subject to:} \\
& \sum_{e \in \delta(S)} x_e \geq 1 && S : f(S) = 1 \\
& x_e \geq 0 && e \in E.
\end{aligned}
$$

$$
\begin{aligned}
\text{(Dual LP)} \qquad & \max \sum_{S : f(S) = 1} y_S \\
& \text{subject to:} \\
& \sum_{S : e \in \delta(S)} y_S \leq c_e && e \in E \\
& y_S \geq 0 && S : f(S) = 1.
\end{aligned}
$$

Primal complementary slackness condition:

$$x_e > 0 \implies \sum_{S: e \in \delta(S)} y_S = c_e$$

Dual complementary slackness condition:

$$y_S > 0 \implies \sum_{e \in \delta(S)} x_e = 1$$

3 Algorithm

Primal Dual Method for Approximation: We maintain a feasible dual solution and a integral primal solution such that the primal complementary slackness condition is satisfied (while the dual complementary slackness condition may be relaxed). The primal solution is likely to be infeasibel at first, so we increase the dual solution value to decrease the infeasibility of the primal solution. In the end we obtain a feasible integral primal solution with a higher valued feasible dual solution. We note that the solutions we obtain may not satisfy the dual complementary slackness solution as the LP may not be integral, so our integral primal solution is only an approximate solution to the original network design problem.

Algorithm 1 Primal Dual Algorithm for Network Design Problems

$y \leftarrow 0$
$A \leftarrow \emptyset$
$l \leftarrow 0$
while A is infeasible **do**
  $l \leftarrow l + 1$
  $\mathcal{V}_l \leftarrow \{\text{minimal violated sets} S\}$
  Increase y_S uniformly for all $S \in \mathcal{V}_l$ until $\exists e_l \in \delta(S)$, $\delta(S) \in \mathcal{V}_l : \sum_{S: e_l \in \delta(S)} y_S = c_{e_l}$
  $A \leftarrow A \cup \{e_l\}$
for $j \leftarrow l$ down to 1 **do**
  if $A - \{e_l\}$ is feasible **then** $A \leftarrow A - \{e_l\}$
Output A and y

3.1 Maximality Property

We focus on 0-1 functions that satisfies the maximality property.

Definition A set function $f : 2^V \to \mathbb{N}$ satiesfies the maximality property if for any two disjoint sets A and B, $f(A \cup B) \leq \max\{f(A), f(B)\}$.

Lemma 3.1. *Let f be a 0-1 function that satisfies the maximality property. Let A be any edge set. Then.*

1. *A is feasible for f iff every connected component C of (V, A) has $f(C) = 0$,*
2. *the minimal violated sets of A are the connected components C of (V, A) for which $f(C) = 1$.*

3.2 Implementation

Instead of keeping track of the dual variables, we keep track of $d(i) = \sum_{S:i\in S} y_S$ for each $i \in V$. So each time $d(i)$ increases by ϵ everytime the dual variable for the component i is in increases by ϵ. We can maintain the collection of connected components through a union find structure. To find the e_l with tight dual constraint, we minimize all $e_l = (i, j)$ with $i \in C_p$, $j \in C_q$, $C_p \neq C_q$ and find

$$\operatorname*{argmin}_{e_l} \frac{c_{e_l} - d(i) - d(j)}{f(C_p) + f(C_q)}.$$

For a pair of components, only the edge crossing the pair with the smallest above value will be maintained and the value will be used as the key for the edge in a priority queue so that we can find the argmin easily.

Algorithm 2 Detailed Implementation for the Primal Dual Algorithm

$A \leftarrow \emptyset$
$LB \leftarrow 0$
$l \leftarrow 0$
$\mathcal{C} \leftarrow \{\{v\} : v \in V\}$
for $i \in V$ **do** $d(i) \leftarrow 0$
while $\exists C \in \mathcal{C} : f(C) = 1$ **do**
  $l \leftarrow l + 1$
  Find edge $e_l = (i, j)$ with $i \in C_p \in \mathcal{C}$, $j \in C_q \in \mathcal{C}$, $C_p \neq C_q$ that minimizes $\epsilon = \dfrac{c_{e_l} - d(i) - d(j)}{f(C_p) + f(C_q)}$
  $A \leftarrow A \cup \{e_l\}$
  for all $k \in C \in \mathcal{C}$ **do**
    $d(k) \leftarrow d(k) + \epsilon \cdot f(C)$
  $LB \leftarrow LB + \epsilon \cdot \sum_{C\in\mathcal{C}} f(C)$
  $\mathcal{C} \leftarrow \mathcal{C} \cup \{C_p \cup C_q\} - \{C_p\} - \{C - 1\}$
for $j \leftarrow l$ down to 1 **do**
  if all components C of $A - \{e_j\}$ satisfy $f(C) = 0$ **then** $A \leftarrow A - \{e_j\}$
Output A and LB

3.3 Analysis

Let A^* be the final output of the algorithm. We know that the cost of output is

$$\begin{aligned}
c(A^*) &= \sum_{e\in A^*} c_e \\
&= \sum_{e\in A^*} \sum_{S:e\in\delta(S)} y_S && \text{(the dual constraint is tight for all } e \in A^*) \\
&= \sum_{e\in A^*} \sum_{S:e\in\delta(S)} \sum_{i:S\in\mathcal{V}_i} \epsilon_i && (y_S \text{ is increased by } \epsilon_i \text{ when it is in } \mathcal{V}_i) \\
&= \sum_{i=1}^{l} \sum_{S\in\mathcal{V}_i} \sum_{e\in\delta(S)\cap A^*} \epsilon_i && \text{(switch summation order by counting } \epsilon_i \text{ differently)} \\
&= \sum_{i=1}^{l} \epsilon_i \sum_{S\in\mathcal{V}_i} |\delta(S) \cap A^*|.
\end{aligned}$$

Let $A_1 = \emptyset$ and $A_i = \{e_1, e_2, \ldots, e_{i-1}\}$ for $i \in [l] \setminus \{1\}$. So A_i is the set A in the algorithm right before adding the edge e_i. Let B_i be a minimal augmentation of A_i. We note that $A^* \subseteq B_i$ for all $i \in [l]$ by the reverse delete order. Therefore, continuing from above, we have

$$\begin{aligned} c(A^*) &= \sum_{i=1}^{l} \epsilon_i \sum_{S \in \mathcal{V}_i} |\delta(S) \cap A^*| \\ &\leq \sum_{i=1}^{l} \epsilon_i \sum_{S \in \mathcal{V}_i} |\delta(S) \cap B_i|. \end{aligned}$$

Construct a graph H_i by taking graph (V, B_i) and shrinking each connected component of (V, A_i) into a vertex. For each $S \in \mathcal{V}_i$, it is a minimally violated set, so it induces a connected component in (V, A_i) and corresponds to a vertex v_S in H_i. Thus, $|\delta(S) \cap B_i| = \deg_{H_i}(v_S)$. Thus,

$$\begin{aligned} c(A^*) &\leq \sum_{i=1}^{l} \epsilon_i \sum_{S \in \mathcal{V}_i} |\delta(S) \cap B_i| \\ &= \sum_{i=1}^{l} \epsilon_i \sum_{S \in \mathcal{V}_i} \deg_{H_i}(v_S) \end{aligned}$$

On the other hand, the dual solution value is a lower bound of our optimal primal solution value:

$$OPT \geq \sum_{S: f(S)=1} y_S = \sum_{i=1}^{l} |\mathcal{V}_i| \epsilon_i.$$

Therefore, if we can show that there exists γ such that for all $i \in [l]$

$$\sum_{S \in \mathcal{V}_i} \deg_{H_i}(v_S) \leq \gamma |\mathcal{V}_i|,$$

then we have $c(A^*) \leq \gamma OPT$, which implies that the primal dual algorithm is an γ-approximate algorithm.

Theorem 3.1. *If there exists γ such that for any infeasible solution A and any minimal augmentation B,*

$$\sum_{S \in \mathcal{V}(A)} \deg_H(v_S) \leq \gamma |\mathcal{V}(A)|,$$

where H is the graph obtained by taking (V, B) and shrinking each connected component S of (V, A) into a vertex v_S, then the primal dual algorithm is an γ-approximate algorithm.

Proof. This theorem follows easily from our previous analysis as the condition of the theorem is even stronger than the above analysis requires. □

4 Proper function and Metric Min Cost Perfect Matching

Definition A function $f : 2^V \rightarrow \{0, 1\}$ is proper if it satiesfies the following three conditions: i) $f(V) = 0$, ii) f is symmetric, iii) f satiesfies the maximality property.

Lemma 4.1. *For a proper function f, if $f(S) = f(A) = 0$ for $A \subseteq S$ then $f(S - A) = 0$.*

For Metric Min Cost Perfect Matching problem, we define f such that $f(S) = 1$ if $|S|$ is odd and $f(S) = 0$ otherwise. It's easy to see that f is proper in this case.

Theorem 4.1. *For any proper function f, the primal dual algorithm gives a 2-approximation.*

Proof. Let us use the notations from Theorem 3.1. We only need to show the condition for Theorem 3.1 with $\gamma = 2$. Let W be the set of vertices in H that were shrunk from connected components C in (V, A) with $f(C) = 1$. So there is a one-to-one correspondence between vertices in W and sets in $\mathcal{V}(A)$. Therefore, we only need to show that $\sum_{v \in W} \deg_H(v) \le 2|W|$.

We note that H is a forest because B is a minimal augmentation. We claim that every leaf of H belongs to W. If the claim is true, we have

$$\sum_{v \in W} \deg(v) = \sum_{v \in V(H)} \deg(v) - \sum_{v \in V(H) \setminus W} \deg(v) \le 2(|V(H)| - 1) - 2(|V(H)| - |W|) = 2|W| - 2,$$

since H is a forest and all vertices in $V(H) \setminus W$ have degree at least 2. So we only need to prove the claim now. Suppose for contradiction there is a leaf v_S of H such that it corresponds to S such that $f(S) = 0$. Let e be the edge incident to v in H, let C be the connected components of (V, B) that contains S. Since B is feasible, $f(C) = 0$. Then by Lemma 4.1, we have $f(C - S) = 0$. Since B is minimal, $B - \{e\}$ is not feasible, so either $f(S) = 1$ or $f(C - S) = 1$. This is a contradiction. So the claim is proved, and this concludes the proof of the theorem. □