**18.433 Combinatorial Optimization**

# The Primal-dual Algorithm

*October 28* *Lecturer: Santosh Vempala*

In this lecture, we introduce the *complementary slackness* conditions and use them to obtain a primal-dual method for solving linear programming.

## 1 Complementary Slackness

As we have seen before, using strong duality, we know that the optimum value for the following two linear programming are equal, i.e. $u = w$, if they are both feasible.

$$u = max\{c^T x : Ax \leq b, x \geq 0\} \quad (P)$$

$$w = min\{b^T y : A^T y \geq c, y \geq 0\} \quad (D)$$

Using the above result, we can check the optimality of a primal and/or a dual solution.

**Theorem 1.** *Suppose $x$ and $y$ are feasible solutions to $(P)$ and $(D)$. Then $x$ and $y$ are optimal if and only if the following conditions are satisfied:*

$$\forall i \ (b_i - \sum_j a_{ij}x_j)y_i = 0;$$

$$\forall j \ (\sum_i a_{ij}y_i - c_j)x_j = 0.$$

*Proof.* First, we note that since $x$ and $y$ are feasible $(b_i - \sum_j a_{ij}x_j)y_i \geq 0$ and $(\sum_i a_{ij}y_i - c_j)x_j \geq 0$. By summing over $i$ and $j$, we have:

$$\sum_i (b_i - \sum_j a_{ij}x_j)y_i \geq 0 \tag{1}$$

$$\sum_j (\sum_i a_{ij}y_i - c_j)x_j \geq 0 \tag{2}$$

By adding 1 and 2 and using the strong duality theorem

$$\sum_i b_i y_i - \sum_{i,j} a_{ij}x_j y_i + \sum_{j,i} a_{ij} y_i x_j - \sum_j c_j x_j = \sum_i b_i y_i - \sum_j c_j x_j = 0.$$

Therefore, all our inequalities must be equalities and we obtain the desired result. □

## 2 Primal-dual algorithm

The main implication of Theorem 1 is that if $x$ and $y$ are feasible and satisfy the complementary slackness conditions, then they are optimal. This result leads us to the primal-dual algorithm in which we start with a feasible solution $x$ and $y$ and try to satisfy the conditions more and more.

For the sake of convenience, we consider the primal and dual programs as follows:

$$min\{c^T x : Ax = b, x \geq 0\} \quad (P)$$

$$max\{b^T y : A^T y \leq c\} \quad (D)$$

In this form, the complementary slackness conditions that we need to satisfy are reduced to:

$$\forall j \ (c_j - \sum_i a_{ij} y_i) x_j = 0. \tag{3}$$

The steps of the primal-dual algorithm are as follows:

1. Start with a feasible solution $y$ for $(D)$. Obtaining such feasible solution $y$ is easier than solving the linear program in many cases.

   Let $J = \{j : \sum_i a_{ij} y_i = c_j\}$.

   Now using 3, we need to obtain a solution $x$ for $(P)$ such that $\forall j \notin J, x_j = 0$. So the question is whether there is a feasible solution $x$ with this property.

2. Formulate the restricted primal (RP) as follows:

$$\begin{aligned} min \sum_{i=1}^{m} X_i \\ \forall i \quad \sum_{j \in J} a_{ij} x_j + X_i = b_i \\ X_i, x_j \geq 0 \\ \forall j \notin J, x_j = 0 \end{aligned}$$

   In fact, $(RP)$ formulates the problem of finding feasible solution $x$ with the aforementioned property. Here variables $X_i$'s are artificial variables and if $min \sum_{i=1}^{m} X_i$ is equal to zero, then $x_j$'s are optimal solutions to $(P)$.

3. If $Opt(RP) = 0$ then $x$ and $y$ are optimal. Otherwise $Opt(RP) > 0$ and we write the dual of (RP), namely (DRP), for which we get solution $\overline{y}$.

$$\begin{aligned} & max \sum_{i=1}^{m} b_i y_i \\ \forall j \in J \quad & \sum_i a_{ij} y_i \leq 0 \\ & y_i \leq 1 \end{aligned}$$

4. Improve the solution to $(D)$ by setting $y' = y + \epsilon\overline{y}$. Here we determine $\epsilon$ such that $y'$ is feasible and $\sum_i b_i y'_i > \sum_i b_i y_i$. For feasibility, we must satisfy the condition $\forall j \ \sum_i a_{ij} y'_i \leq c_j$. For $j \in J$, we must have $\sum_i a_{ij} y_i + \epsilon \sum_i a_{ij} \overline{y}_i \leq c_j$. Since $\forall j \in J \ \sum_i a_{ij} \overline{y}_i \leq 0$, $\epsilon$ can be arbitrary positive for $j \in J$.

   Thus by taking
   $$\epsilon = min_{\{j \notin J \ s.t. \sum_i a_{ij}\overline{y}_i > 0\}} \frac{c_j - \sum_i a_{ij} y_i}{\sum_i a_{ij} \overline{y}_i}$$
   we obtain our $\epsilon > 0$ such that $y'$ is feasible.

   Also since $Opt(DRP) = Opt(RP) > 0$ and $\epsilon > 0$,

   $$\sum_i b_i y'_i = \sum_i b_i y_i + \epsilon \sum_i b_i \overline{y}_i > \sum_i b_i y_i.$$

We note that in the above primal-dual algorithm, solving $(DRP)$ is usually easier than solving $(P)$ or $(D)$. In fact, in this approach, programs $(P)$ and $(RP)$ are temporary programs and we want to solve $(D)$. To this end, we first solve $(DRP)$ and then use the solution to improve $y$ iteratively.

## 2.1 Example

Consider the following formulation of the max-flow problem:

$$
\begin{aligned}
& max\ f \\
\sum_j x_{sj} - \sum_j x_{js} - f &\le 0 \\
f - \sum_j x_{jt} + \sum_j x_{tj} &\le 0 \\
\forall i \ne s,t \quad \sum_j x_{ij} - \sum_j x_{ji} &\le 0 \\
x_{ij} &\le u_{ij} \\
-x_{ij} &\le 0
\end{aligned}
$$

It is worth mentioning that in the original max-flow formulation, the first three sets of constraints are equalities. However in our new formulation by summing these three sets of inequalities, we get $0 \le 0$ and thus these weaker sets of inequalities imply the equalities.

Now, we consider the above formulation as $(D)$. One feasible solution to $(D)$ can be obtained by taking $x$ as a zero vector. Now if we go directly to $(DRP)$ we have:

$$
\begin{aligned}
& maxf \\
\sum_j x_{sj} - \sum_j x_{js} - f &\le 0 \\
f - \sum_j x_{jt} + \sum_j x_{tj} &\le 0 \\
\forall i \ne s,t \quad \sum_j x_{ij} - \sum_j x_{ji} &\le 0 \\
x_{ij} \le 0 \ \ \forall i,j \ where\ x_{ij} = u_{ij}\ in\ (D) \\
-x_{ij} \le 0\ \forall i,j\ where\ x_{ij} = 0\ in\ (D) \\
x_{ij} &\le 1 \\
f &\le 1
\end{aligned}
$$

We can observe that $(DRP)$ has the following interpretation. Find a path from $s$ to $t$ (with a flow of value 1) that uses only the following arcs in the following ways: saturated arcs in the backward direction; arcs with zero flow in the forward direction; and other arcs in either direction. In other words, we need to find a path in the residual graph. This observation shows that the max-flow algorithm is in fact a primal-dual algorithm.

Finally, we note that primal-dual algorithms do not have polynomial running time guarantees.