Chapter 4

Zachary Campbell

April 3, 2018

In this chapter we will take a step back and look at primal-dual algorithms in more generality. The goal will be to describe a method of solving a set of primal-dual algorithms for *network design problems*. Again, we will be restricting our attention to bipartite graphs. In network design problems we are given a graph $G = (U, V, E)$ and a cost c_{uv} for each edge $(u, v) \in E$, and the goal is to find a minimum/maximum-cost subset $E^{'} \subset E$ that satisfies some criteria. Our maximum-cost matching problem is an example of this. There are other common examples that we will explore later on, but for now it suffices to just think of these problems as choosing subsets of our graph according to some stipulations. Throughout, we will be looking at undirected graphs.

1 The Classical Primal-Dual Method

We begin by looking at what's known as the "classical" primal-dual method, which is concerned with linear programs for polynomial-time solvable optimization problems. This will allow us to build up a framework for a more general primal-dual method that we can use for approximation algorithms - i.e. for problems that are known to be NP-hard. **(Not sure if this is within the scope of the thesis – discuss with Jim)

Let's consider the linear program

$$\text{minimize } \mathbf{c}^T\mathbf{x} \tag{1}$$

$$\text{subject to } \; A\mathbf{x} \geq \mathbf{b} \tag{2}$$

$$\mathbf{x} \geq 0 \tag{3}$$

and its dual

$$\text{maximize } \mathbf{b}^T\mathbf{y} \tag{4}$$

$$\text{subject to } A^T\mathbf{y} \leq \mathbf{c} \tag{5}$$

$$\mathbf{y} \geq 0. \tag{6}$$

We first define a concept that we will use throughout the rest of this thesis.

Definition (Complementary slackness). Given two linear programs in the form above, the *primal complementary slackness conditions* are the conditions which, given primal solution $\mathbf{x}$, are necessary for a dual solution $\mathbf{y}$:

$$x_j > 0 \implies A^j\mathbf{y} = c_j,$$

where A^j is the jth column of A. Similarly, the *dual complementary slackness conditions* are the conditions which, given dual solution $\mathbf{y}$, are necessary for a primal solution $\mathbf{x}$:

$$y_i > 0 \implies A_i\mathbf{x} = b_i,$$

where A_i is the ith row of A. Together, these conditions give us necessary and sufficient conditions for solving the primal-dual system, which we will prove. The (maximization) primal slackness variables are given by $\mathbf{s} = \mathbf{b} - A\mathbf{x}$. The dual slackness variables are given by $\mathbf{t} = A^T\mathbf{y} - \mathbf{c}$.

Theorem . [CITE THIS THEOREM] Let $\mathbf{x}$ be a primal feasible solution, and $\mathbf{y}$ a dual feasible solution. Let $\mathbf{s}$ and $\mathbf{t}$ be the corresponding slackness variables. Then $\mathbf{x}$ and $\mathbf{y}$ are optimal solutions if and only if the following two conditions hold:

$$x_j t_j = 0 \quad \forall j \tag{7}$$

$$y_i s_i = 0 \quad \forall i. \tag{8}$$

Proof. Let $u_i = y_i s_i$ and $v_j = x_j t_j$, and $\mathbf{u} = \sum_i u_i$, $\mathbf{v} = \sum_j v_j$. Then $\mathbf{u} = 0$ and $\mathbf{v} = 0$ if and only if (7) and (8) hold. Also,

$$\begin{aligned}
\mathbf{u}+\mathbf{v} &= \sum y_i s_i + \sum x_j t_j \\
&= \sum y_i (b_i - A_i x_i) + \sum x_j (A_j^T y_j - c_j) \\
&= \sum b_i y_i - \sum c_j x_j,
\end{aligned}$$

so we get that $c^T\mathbf{x} = b^T\mathbf{y}$ if and only if $u+v=0$, which proves the statement. $\square$

The general "tug-of-war" between the primal and dual suggests an economic interpretation of slackness conditions. We can think of our primal (maximization) problem as concerned with profit given some constraints on resources, i.e. a resource allocation problem. The dual can be interpreted as a valuation of the resources – it tells us the availability of a resource, and its price. So if we have optimal $\mathbf{x}$ and $\mathbf{y}$, we can interpret slackness as follows: if there is slack in a constrained primal resource i ($s_u > 0$), then additional units of that resource must have no value ($y_u = 0$); if there is slack in the dual price constraint ($t_v > 0$) there must be a shortage of that resource ($x_v = 0$).

We now give an example of complementary slackness in action. Let's look back to our maximum weight matching problem.Recall the primal linear program for maximum-weight matching:

$$\begin{aligned}
\text{maximize} \quad & \sum_{u,v} c_{uv} x_{uv} && (9) \\
\text{subject to} \quad & \sum_{v} x_{uv} \leq 1, \quad \forall u \in U, && (10) \\
& \sum_{u} x_{uv} \leq 1, \quad \forall v \in V, && (11) \\
& x_{uv} \geq 0. && (12)
\end{aligned}$$

and its dual

$$\begin{aligned}
\text{minimize} \quad & \sum_{u} y_u + \sum_{v} y_v && (13) \\
\text{subject to} \quad & y_u + y_v \geq c_{uv} \quad \forall u \in U,\ v \in V, && (14) \\
& y_u, y_v \geq 0. && (15)
\end{aligned}$$

The format here is a little different, since our primal is a maximization problem and the dual is a minimization, but it's easy enough to reverse the roles. It's easy to see our corresponding primal complementary slackness conditions are

$$x_{uv} > 0 \implies y_u + y_v = c_{uv}. \tag{16}$$

The dual complementary slackness conditions are

$$y_u > 0 \implies \sum_v x_{uv} = 1, \tag{17}$$

$$y_v > 0 \implies \sum_u x_{uv} = 1. \tag{18}$$

In general, the slackness conditions guide us in our algorithm – they tells us how, given a solution to one of the problems, we should augment the solution to the other. For example, the algorithm we presented for maximum-weight matching/minimum vertex cover intializes with a solution to both the primal and dual that satisfies conditions (8) and (10); the algorithm then at each step works to decrease the number of conditions in (9) that are unsatisfied, while maintaining satisfiability of (8) and (10). This method is not unique to the Hungarian algorithm. In fact, the Hungarian algorithm paved the way for this general method, which we describe presently.

Looking back at the original linear programs at the beginning of this chapter, suppose we have a dual feasible solution $\mathbf{y}$. We can then state the problem of finding a feasible primal solution $\mathbf{x}$ that obeys our complementary slackness conditions as another *restricted* linear program. Define the sets $A = \{j \mid A^j\mathbf{y} = c_j\}$ and $B = \{i \mid y_i = 0\}$. So A tells us which dual constraints (5) are tight, given the solution $\mathbf{y}$, and B tells us which y_i are 0. What we want to do is give a linear program to find a solution $\mathbf{x}$ that minimizes the "violation" of the complementary slackness conditions and the primal constraints, and to do so we will index our variables by these sets. We will have slack variables s_i which will describe the difference between $A_i\mathbf{x}$ and b_i for $i \notin A$. We do this because we want to look at all $y_i > 0$ where the we do not have that $A_i\mathbf{x} = b_i$. So part of our objective function will be to minimize the sum of these s_i. We also want to minimize the sum over variables x_j where $j \notin A$. This is because we want to see if there are any x_j such that $A^j\mathbf{y} \neq c_j$. So we give the following restricted primal linear program:

$$\text{minimize} \sum_{i \notin B} s_i + \sum_{j \notin A} x_j \tag{19}$$

$$\text{subject to} \quad A_i \mathbf{x} \geq b_i \quad i \in B, \tag{20}$$

$$A_i \mathbf{x} - b_i = s_i \quad i \notin B, \tag{21}$$

$$\mathbf{x} \geq 0, \tag{22}$$

$$\mathbf{s} \geq 0. \tag{23}$$

Observe that if this restricted primal has a feasible solution $(\mathbf{x}, \mathbf{s})$ such that the objective function is 0, then $\mathbf{x}$ is a feasible primal solution that satisfies the complementary slackness conditions for the dual solution $\mathbf{y}$. This means that $\mathbf{x}$ and $\mathbf{y}$ are optimal primal and dual solutions. If, however, the optimal solution to this restricted primal has value greater than 0, more work is required. We can consider the dual of the restricted primal:

$$\text{maximize } \mathbf{b}^T \mathbf{w} \tag{24}$$

$$\text{subject to} \quad A^j \mathbf{w} \leq 0 \qquad j \in A, \tag{25}$$

$$A^j \mathbf{w} \leq 1 \qquad j \notin A, \tag{26}$$

$$w_i^{'} \geq -1 \quad i \notin B, \tag{27}$$

$$w_i^{'} \geq 0 \qquad i \in B. \tag{28}$$

What we want here is to improve our dual solution. By assumption, the optimal solution to this linear program's primal is greater than 0, so we know that this dual has a solution $\mathbf{w}$ such that $\mathbf{b}^T \mathbf{w} > 0$. What we want is the existence of some $\epsilon > 0$ such that $\mathbf{y}^{'} = \mathbf{y} + \epsilon \mathbf{w}$ is a feasible dual solution. In particular, a solution of this form will be an improvement on our original solution $\mathbf{y}$. We can calculate bounds on ϵ as follows. The two conditions we must satisfy in order to maintain dual feasibility are that $y^{'} \geq 0$ and $A^T y^{'} \leq c$. This means that we need

$$y_i + \epsilon w_i \geq 0 \tag{29}$$

$$A_j^T y + A_j^T \epsilon w \leq c_j. \tag{30}$$

Let's consider the first one. When $w_i > 0$, we are fine; we need to be careful when $w_i < 0$ since this could potentially violate the inequality. Solving in this way, we get a first bound on ϵ:

$$\epsilon \leq \min_{i \in B: w_i < 0} (-y_i / w_i).$$

Now let's address the second inequality. When $A_j^T w \leq 0$, we are defintely okay. We need to be careful about violating the constraint when $A^T w > 0$. Thus, we can calculate a second bound on ϵ:

$$\epsilon \leq \min_{j \in A: A_j^T w > 0} \frac{c_j - A_j^T y}{A_j^T w}.$$

If we choose the lower of these two ϵ values, we obtain a new feasible dual solution that has greater objective value. We can then work by reiterating the procedure, with the hope that we find an optimal primal solution.

It's not immediately clear why reducing our original linear programs to a series of linear programs is heplful. However, note that the vector **c** has totally disappeared in the restricted primal and its dual. Recall that in the original linear program, **c** gave us the edge-costs on our graph. So this method reworks our original weighted problem into unweighted parts, which are easier to solve. Oftentimes, it is the case that we can interpret these unweighted problems as purely combinatorial problems, which means that instead of actually solving the problem with linear programming, we can solve it by combinatorial methods. Using a combinatorial algorithm to find a solution **x** that obeys the complementary slackness conditions, or to find an improved dual solution **y**, is oftentimes more efficient.

2 Primal-dual method for weighted matchings

Let us now look at an example of this method. We will look at a weighted matching problem, as in the previous chapter, but this time we will look at *minimizing* the cost of the matching, instead of maximizing. We do this mainly because it illustrates something important about the underlying structure of these matching problems. It will be easy to see how the same method can be used for the case in which we want a maximimum matching. So the primal linear program for a minimum weight perfect matching on a bipartite graph is given as follows.

$$\begin{aligned}
\text{minimize} \quad & \sum_{u,v} c_{uv} x_{uv} && (31)\\
\text{subject to} \quad & \sum_{v} x_{uv} \geq 1, \quad \forall u \in U, && (32)\\
& \sum_{u} x_{uv} \geq 1, \quad \forall v \in V, && (33)\\
& x_{uv} \geq 0. && (34)
\end{aligned}$$

Its dual is

$$\begin{aligned}
\text{maximize} \quad & \sum_{u} y_u + \sum_{v} y_v && (35)\\
\text{subject to} \quad & y_u + y_v \leq c_{uv} \quad \forall u \in U,\ v \in V, && (36)\\
& y_u, y_v \geq 0. && (37)
\end{aligned}$$

We need to start with a dual feasible solution, and try to find a primal solution that minimizes the violation of the constraints and slackness conditions. We can start with the trivial dual solution of $y_u, y_v = 0$ for all u, v. Let's now think about our primal complementary slackness. The set A is given by $\{(u,v) \in E \ : \ y_u + y_v = c_{uv}\}$. We know that these are the edges we want to include in our matching, and since we know our linear program has integer solutions at extreme points of the polyhedron, let's specify that $x_{uv} = 0$ for $(u,v) \notin a$. Now, our other slackness variables s_u, s_v look like

$$\sum_{v:(u,v)\in E} x_{uv} - s_u = 1$$

$$\sum_{u:(u,v)\in E} x_{uv} - s_v = 1.$$

So we want to minimize over the sum of s_u and s_v. Note that at this point, our set B consists of all vertices. So our restricted primal linear program is

$$\text{minimize} \sum_{u \in U} s_u + \sum_{v \in v} s_v \tag{38}$$

$$\text{subject to} \quad \sum_{v} x_{uv} - s_u = 1 \qquad \forall u, \tag{39}$$

$$\sum_{u} x_{uv} - s_v = 1 \qquad \forall v, \tag{40}$$

$$x_{uv} = 0 \quad (u, v) \notin A, \tag{41}$$

$$x_{uv} \geq 0 \quad (u, v) \in A, \tag{42}$$

$$\mathbf{s} \geq 0. \tag{43}$$

Let's first observe that all components of this restricted primal take on values 0 or 1, as in the original primal. Moreover, note that we have turned a weighted problem into an unweighted combinatorial problem. We've specified that we are not including any edge $(u, v) \notin A$ in our matching, and we are trying to include as many $(u, v) \in A$ as possible in our matching by minimizing the slackness variables. Note that the graph $G^{'} = (U, V, J)$ is exactly the equality subgraph as defined in the previous chapter! In our Hungarian algorithm we repeatedly sought to find maximum cardinality matchings within this subgraph, which is exactly what this restricted primal is having us do. This tells us that the problems of maximum weight matching and minimum weight matching only differ in the labeling we are specifying. The underlying procedure for solving both of the problems is essentially the same. So if we find a perfect matching in $G^{'}$, we will have found an $\mathbf{x}$ that obeys the complementary slackness conditions, i.e. $\sum_u s_u + \sum_v s_v = 0$. Moreover, this implies that the dual solution $\sum_u y_u + \sum_v y_v$ must be optimal as well.

Now, if the solution $\sum_u s_u + \sum_v s_v > 0$, we do not have an optimal $\mathbf{x}$, so we need to adjust our dual.

We look at this now, in the dual linear program of the restricted primal.

$$\text{maximize} \sum_{u \in U} w_u + \sum_{v \in V} w_v \tag{44}$$

$$\text{subject to} \qquad w_u + w_v \leq 0 \qquad \forall (u,v) \in A, \tag{45}$$

$$w_u + w_v \leq 1 \qquad \forall (u,v) \notin A, \tag{46}$$

$$w_u, w_v \geq -1 \qquad u, v \notin B, \tag{47}$$

$$w_u, w_v \geq 0 \qquad u, v \in B, \tag{48}$$

$$\mathbf{w} \geq 0. \tag{49}$$

We now want to find an ϵ such that the solution $z = \sum_u y_u + \sum_v y_v + \epsilon(\sum_u w_u + \sum_v w_v)$ is (1) feasible and (2) an improvement of the dual objective. First of all, we know that since the restricted primal has solution ≥ 0, the solution to this dual will also be ≥ 0. So we just need to worry about the condition

$$y_u + y_v + \epsilon(w_u + w_v) \leq c_{uv}.$$

So we get that we at least need that $\epsilon \leq \min_{(u,v) \notin A : w_u + w_v > 0} \frac{c_{uv} - y_u - y_v}{w_u + w_v}$. We can refine this by noting that since $0 < w_u + w_v \leq 1$ for $(u,v) \notin A$, we have $\epsilon = \min_{(u,v) \notin A}(c_{uv} - y_u - y_v)$. Note that the negative of this is exactly the quantity we modify our labeling by in the Hungarian algorithm in the previous chapter. Thus we've found an ϵ that maintains dual feasibility, and increases the objectie function. We can use this solution and revisit the restricted primal in order to look for an improved feasible primal solution.

3 Auction algorithms

We now look at a cool application of weighted bipartite matching. Let's imagine our assignment problem is an auction. We will motivate this through an economic lens. Suppose that a good g has a price p_g, and a buyer b must pay p_g to receive this item. Suppose that the amount b is willing to pay for g, or b's valuation of b, is c_{bg}. Then the net value of this item to b is $c_{bg} - p_g$. Assuming that each bidder b is a rational agent acting in their own best interest, each b would want to be assigned a good g that maximizes their net value, i.e. for all b with good g, we have that

$$c_{bg} - p_g = \max_g \{c_{bg} - p_g\}.$$

This would give us an economic system in equilibrium; no bidder would have an incentive to seek/trade for a different good.

The reason this is of interest to us is that an equilibrium assignment maximizes total profit (i.e. it solves the primal linear program for maximum weight matchings), while the corresponding set of prices solves the associated dual problem.

Our goal is to maximize the total amount earned in the auction – i.e. to maximize $\sum_{(b,g)} c_{bg}$, under the constraint that no bidder gets more than one good, and no good is purchased by more than one bidder.

At a general step in our algorithm, we will want to consider a bidder b that is currently not the owner of any good, and find a good g that maximizes $c_{bg} - p_g$. We will then need to do to things: (1), replace the current owner of g (put them back into the set of unassigned bidders) with b; (2) bump the price of g by some amount δ. Step (2) is tricky to figure out. Bertsekas [CITE] describes in detail the process of finding a good δ and, more importantly, what values of δ to avoid. Here we leave out the details of finding a decent δ. We first present the algorithm with a given δ, and then analyze why it is a good choice.

ALG 1

For each $g \in G$, set $p_g := 0$ and $owner_g := null$.
Queue $Q := B$.
Set $\delta = 1/(|G| + 1)$.
while $Q \neq \emptyset$
    $b = Q.deque()$
    Find $g \in G$ that maximizes $c_{bg} - p_g$
    if $c_{bg} - p_g \geq 0$
        $Q.enqueue(owner_g)$
        $owner_g = b$
        $p_g = p_g + \delta$
return $(owner_g, g)$ for all g.

Definition . We say that a bidder b is δ-happy if one of the following is true:

1. b is the owner of some good g, and for all other goods $g^{'}$, we have that $\delta + c_{bg} - p_g \geq$

$c_{bg'} - p_{g'}$, i.e. our b has the good g that maximizes the value of their contribution.

2. for no good g does it hold that $owner_g = b$ and for all goods g $w_{bg} \leq p_g$.

What we will show is that the algorithm reaches an equilibrium in which all bidders are "happy," as defined above. The loop invariant is that all bidders not in Q are δ- happy. This is clearly true at the beginning, as all bidders start in our queue. When a bidder b is dequeued, line 6 in our loop chooses good g that maximizes $w_{bg} - p_g$, which means it chooses a good that makes b δ-happy, if such a good exists. We need to confirm that this step does not hurt the invariant for any other bidder b'. Well, an increase in p_g by δ for any g not owned by b' does not violate the inequality $\delta + w_{b'g'} - p_{g'} \geq w_{b'g} - p_g$. On the other hand, if b' did own g, this means that b' has been thrown back into Q, so b' no longer owns anything.

Lemma . If all bidders are δ-happy then for every matching M' we have that

$$n\delta + \sum_{b=owner_g} w_{bg} \geq \sum_{(b,g)\in M'} w_{bg},$$

where n is the number of bidders.

First, note that this lemma implies the correctness of our algorithm. Since $n\delta < 1$ and all of our weights are integral, proving this inequality will show that the matching we output is a maximum matching.

Proof. Fix a bidder b, and let good g be such that $b = owner_g$. Let g' be the good assigned to b in M'. (Note: these could be *null*, in which case we adopt the convention of assigning their weights and prices to be zero.) Since b is δ- happy, we have that $\delta + w_{bg} - p_g \geq w_{bg} - p_g$. Now let's sum over all b to get

$$\sum_{b=owner_g} (\delta + w_{bg} - p_g) \geq \sum_{(b,g')\in M'} (w_{bg'} - p_g).$$

We know that our algorithm gives us a matching, as does M', which means that each good g can only appear at most once on each side of this inequality. So if we subtract $\sum_g p_g$ from each side, and rearrange a bit, we get

$$\sum_{b=owner_g} \delta + \sum_{b=owner_g} w_{bg} \geq \sum_{(b,g')\in M'} (w_{bg'}).$$

But $\sum_{b=owner_g} \delta \leq n\delta$, which gives us what we want. $\square$

Let's think about how this algorithm relates to the corresponding primal and dual linear programs. First, note that this algorithm maintains primal feasibility at all times, as each good has at most one owner, and each bidder owns at most one good. However, at no point are we maintaining dual feasibility. We can define the "price" on bidders b as $p_b = 0$ for all b. Furthermore, the prices on goods p_g never exceed $\max_b\{w_{bg}\}$. We can think of these p_b and p_g as the corresponding dual variables in the dual linear program. Throughout the course of this algorithm, we never have that $p_b + p_g \geq w_{bg}$, which means that we are violating our primal complementary slackness conditions. Note however that we do maintain the dual complementary slackness conditions. It is still useful to think about this algorithm in terms of the linear programs since, although infeasible as a dual solution, our final price vector $\mathbf{p}$ is the pointwise minimum such that all bidders are δ-happy. So the algorithm is still performing a minimization over $\sum_b p_b + \sum_g p_g = \sum_g p_g$, but relaxing the constraint that $p_b + p_g = p_g \geq w_{bg}$.