CS602 Applied Algorithms **2019-20 Sem II**

Lecture 13: February 20

Scribe: Kaveri Kale *Lecturer: Rohit Gurjar*

1 Minimum Weight Perfect Matching Algorithm For Bipartite Graph

Consider a bipartite graph G(V,E) with vertices partitioned into two sets V_1 & V_2. Let $\{w_e\}_{e \in E}$ be the given weights on edges. Consider one variable for each edge $\{x_e\}_{e \in E}$.

Primal LP: As discussed in previous lectures, we can model the min-weight perfect matching problem as a linear optimization problem.

$$\begin{aligned} \min \sum_{e \in E} w_e x_e \quad & \text{s.t.} \\ 0 \leq x_e \quad & \text{for } e \in E, \\ \sum_{e \in \delta(v)} x_e = 1 \quad & \text{for } v \in V. \end{aligned}$$

Dual LP: Dual for this LP can be written as,

$$\begin{aligned} \max \sum_{u \in V} y_u \quad & \text{s.t.} \\ y_u + y_v \leq w_e \quad & \text{for } e = (u,v) \in E, u \in V_1 \text{ and } v \in V_2. \end{aligned}$$

Definition 13.1. *For a given dual solution $y \in \mathbb{R}^V$, an edge e = (u,v) is called tight edge if $w_e = y_u + y_v$.*

1.1 Primal Dual Schema

In the previous lecture, we saw that the **Complementary Slackness Condition** for above LP with a primal solution $x \in \mathbb{R}^E$ and a dual solution $y \in \mathbb{R}^V$, is as follows:

$$\text{for each } e = (u,v) \in E, \text{ we have } x_e(w_e - y_u - y_v) = 0$$

In other words, x_e can be nonzero only for tight edges, that is, those edges that satisfy $y_u + y_v = w_e$. Primal variable should be zero or dual constraint should be tight.

1.2 Application of Primal-Dual schema to find optimal solution

1. Find a feasible dual solution.
2. Try to find a primal feasible integral solution that satisfies complementary slackness condition with the dual solution.
3. If we succeed in getting a primal feasible solution then we are done. If we fail then there is a way to improve dual solution. Go back to step 2 with the improved dual solution.

We explain the steps of this algorithm in more detail with the help of a running example.

Example. Let's consider following graph with edge weights in Figure 1.

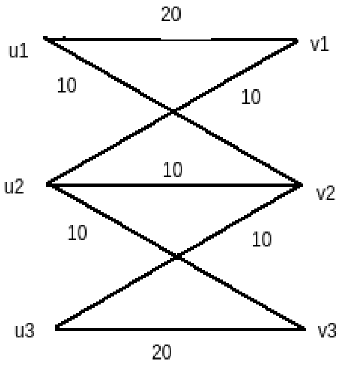

Figure 1: Graph G

Let V_1 and V_2 be the sets of left and right vertices of graph G. In the example, we have $V_1 = \{u_1, u_2, u_3\}$ and $V_2 = \{v_1, v_2, v_3\}$.

1. Start with dual values such that, $y_u = 0$ for all $u \in V_1$ & $y_v = \min\{w_e : e \in \delta(v)\}$ for all $v \in V_2$. It is easy to verify that this is a feasible dual solution. This solution has at least one tight edge for each vertex in V_2. Figure 2 shows the initial dual solution in the example. Red lines represent the tight edges. Value of the dual objective function in the example is

$$\sum_{v \in V} y_v = 30.$$

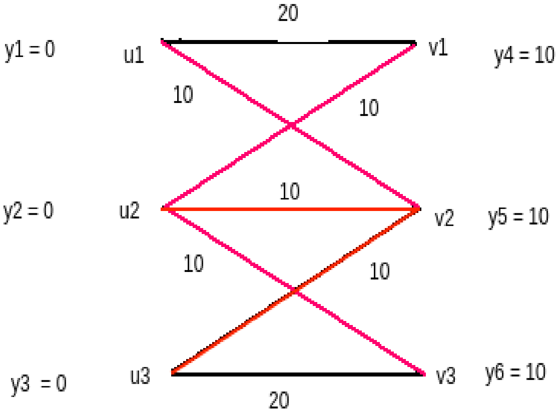

Figure 2: Graph G with tight edges

2. Try to find a primal feasible integral solution that satisfies complementary slackness. This is same as trying to find a perfect matching in the subgraph G' that has only tight edges.

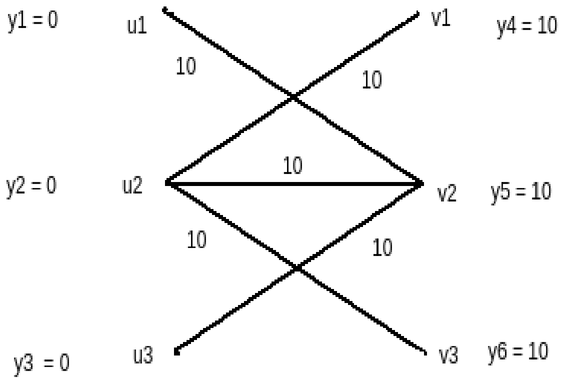

Figure 3: Graph G' with only tight edges

In the example, G' does not have a perfect matching (Figure 3). One way to see this is by observing that $\{u_2, v_2\}$ is a minimum vertex cover for G'. Recall König's theorem that minimum vertex cover size is same as the maximum matching size, which is 2 here.

3. Since we didnt get perfect matching, we will try to improve dual solution y (that is to get a y with higher dual objective). To improve the dual solution, we must increase variables y_v for some vertices v. But note that for any tight edge, if you increase one end point you must decrease the other end point to maintain feasibility. A minimum vertex cover of G' will be useful in improving y, while maintaining feasibility.

   Let $U \subseteq V$ be the vertex cover set with $|U| < n$ and $|V/U| > n$. Such a vertex cover is guaranteed to exist because we do not have a perfect matching (König's theorem). Increase the dual variables for vertices in $|V/U|$ and decrease them for vertices in U, by the same quantity say $\epsilon > 0$.

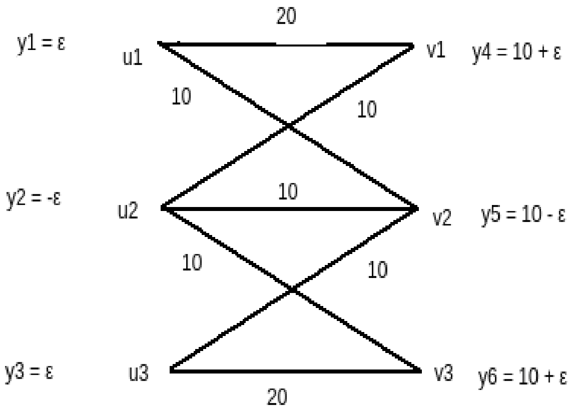

Figure 4: Graph with improved dual values

We argue that the new dual solution is still feasible. Recall that U is a vertex cover of G', the subgraph of tight edges. Hence, every tight edge has at least one end-point in U. This might not be true for non-tight edges. We conclude:

- Every tight edge has at least one end-point that will see $-\epsilon$, thus, the dual constraint for a tight edge is still satisfied.

- Non-tight edges might see $+\epsilon$ on both end points. But, this is not an issue because being non-tight means there was some room to increase the dual variables for the two end points.

That means we can choose some nonzero ϵ. Take it to be the largest possible value such that it maintains the feasibility of all non-tight edge constraints. The total change in the dual value will be $-\epsilon|U| + \epsilon|V \setminus U|$, which is positive. Hence, we get an improved dual solution.

In the running example, we have two non-tight edges (u_1, v_1) and (u_3, v_3) (Figure 4). Each has weight $w_e = 20$. To see what should be the value of ϵ, we consider the dual constraints for these two edges.

$$\begin{aligned} y_1 + y_4 = &\quad \epsilon + 10 + \epsilon &\leq 20 \\ y_3 + y_6 = &\quad \epsilon + 10 + \epsilon &\leq 20 \end{aligned}$$

This means we should choose $\epsilon = 5$. New dual value becomes $30 + 2\epsilon = 40$.

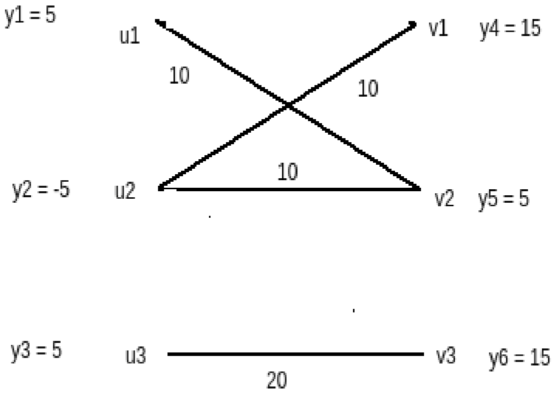

Figure 5: Graph G' with tight edges with respect to new dual

4. Now, with the new dual solution, we go back to step 2. Figure 5 shows the tight edges with respect to the new dual solution. We again try to find a primal integral feasible solution, that is a perfect matching among the tight edges. This time we can find one. Figure 6 shows the perfect matching edges with red colour. By, complementary slackness, this is minimum weight perfect matching (with weight $10 + 10 + 20 = 40$).

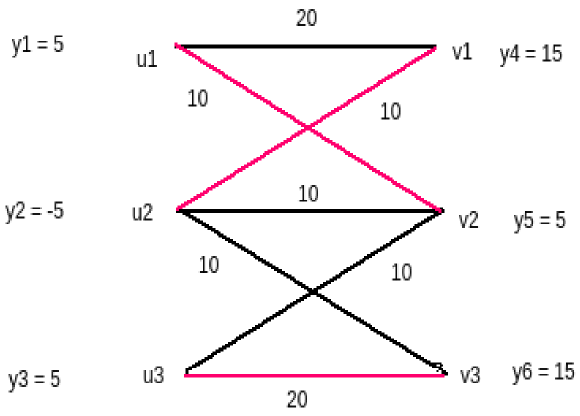

Figure 6: Graph G with matched edges

Bounding the running time. Each iteration of the procedure requires to find a maximum matching/minimum vertex cover with the tight edges. We had discussed how the augmenting path algorithm can be used to find these. There are other steps of changing the dual values, but they do not dominate the running time of finding maximum matching. The naive implementation of augmenting path algorithm takes $O(|E||V|)$ time.

Now, we need to bound the number of iterations. Observe that we can use the dual objective value as the measure of progress since it increases in every iteration. Without loss of generality, we can assume that the given weights are all integral. We claim that the increase in the dual value is also integral in each iteration (proved below). And thus, the number of iterations can be bound by the difference of the final dual value and the initial dual value. This, in turn, can be bounded by the sum of absolute values of all the edge weights (very rough bound).

Note that the weights can have polynomially many bits, and thus, exponentially large value. Then our bound will not be polynomial in the input size. Such bounds are called weakly-polynomial (they are polynomial in the weight-value). A better implementation of the primal-dual scheme exists that can be shown to run in strongly-polynomial time. We will not go into those details, you can see these notes `https://math.mit.edu/~goemans/18433S09/matching-notes.pdf`. We finish with just proving the claim that change in the dual value is integral.

Claim 13.2. *After every iteration, the dual objective remains integral.*

Proof. To show this we just use two claims: (i) for any $u \in V_1$ and $v \in V_2$, the quantity $y_u + y_v$ is always integral and (ii) in every iteration, 2ϵ is integral. In an iteration, the change in the sum $y_u + y_v$ can be -2ϵ, 0 or 2ϵ. Thus, if for each edge (u, v), the sum $y_u + y_v$ is integral and w_e is integral then 2ϵ is integral. In turn, if 2ϵ is integral then the updated sum $y_u + y_v$ remains integral. Thus, the objective value must remain integral: to see this write the dual objective as $\sum_{i=1}^{n}(y_{u_i} + y_{v_i})$, where u_i, v_i are the i-th vertices on the left and right, respectively. □