

## Lecture 12: February 13

Scribe: Prachi Singh

Lecturer: Rohit Gurjar

## 1 Bipartite Matching

Bipartite matching is one of first problems that inspired application of linear programming and in particular, primal-dual scheme to combinatorial optimization. In this lecture, we will apply primal-dual scheme to weighted bipartite matching.

**Definition 12.1 (Bipartite Graph)** A Graph  $G$  is bipartite if the vertex set  $V$  can be partitioned into two sets  $A$  and  $B$  such that no edge in  $E$  has both endpoints in the same set of the bipartition.

**Definition 12.2 (Matching)** A Matching  $M \subseteq E$  is a set of edges such that every vertex  $V$  is incident to at most one edge of  $M$ . In Fig.??, shaded edges form a matching.

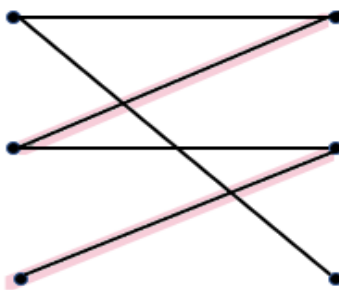


Figure 1: Example of Matching

**Definition 12.3 (Bipartite Matching)** A matching in Bipartite Graph is called Bipartite Matching.

**Goal:** Find the maximum size matching i.e. Maximum cardinality matching problem.

### 1.1 Augmenting Path Algorithm

1. Start with empty matching i.e.,  $M = \phi$ .
2. At any stage, find an augmenting path with respect to  $M$ .
3. Augment current matching.

**Definition 12.4 (Augmenting Path)** An Augmenting path is a path that alternates between matching and non-matching edges, starting and ending with free vertices. In Fig.??, dotted edges do not belong to the matching, and solid edges belong to the matching.

**Claim 12.5** If a matching  $M$  is not maximum, then there exists an augmenting path with respect to  $M$ .

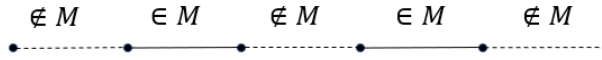


Figure 2: Example of Augmenting Path

The proof of the claim was left as an exercise. One way to prove the claim is to consider the union of matching  $M$  with some maximum size matching  $M'$  and argue that it consists of a set of disjoint alternating paths and cycles. Once we have established that an augmenting path always exists, the next question is:

**Que.** How to find an augmenting path? It is similar to finding an ordinary path from a source to a destination. To ensure that a path alternates between matching and non-matching edges one can direct all the matching edges from left to right and direct all the other edges from right to left. To get an augmenting path, we should find a directed path in this graph from any free vertex on the left to a neighbor of any free vertex on the right. Any BFS/DFS algorithm is good enough for this.

## 2 Minimum weight perfect matching

The bipartite matching problem has various versions, some of which have weights on the edges. Such variants include finding maximum weight matching, maximum/minimum weight perfect matching etc.

**Definition 12.6 (Perfect matching)** A matching is perfect if it covers all vertices. Fig.?? shows an example of perfect matching as all vertices are covered in the matching.

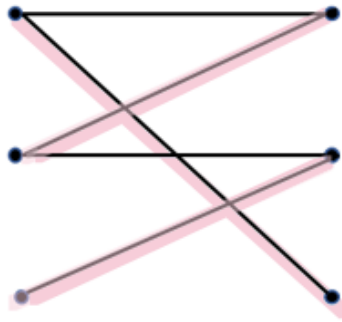


Figure 3: Example of Perfect Matching

These variants turn out to be equivalent. One can reduce the first two to minimum weight perfect matching as follows.

- We can convert maximum weight matching problem to maximum weight perfect matching by converting the given graph into a complete graph by adding 0 weighted edges. As the weight of newly added edges is 0, the sum of weights will be the same as before. In Fig.??, the dotted edges with weight 0 are added to the graph to make the Bipartite Graph Complete. Moreover, if the both sides don't have equal number of vertices then one can add new vertices on the smaller side again with zero weighted edges.

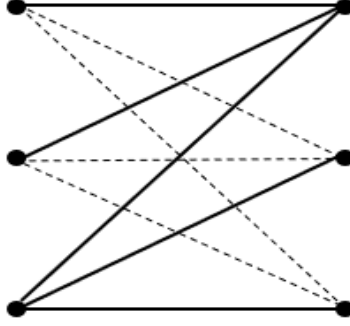


Figure 4: Bipartite Graph to Complete Bipartite Graph by adding edges

- We can also convert the maximum weight perfect matching problem into minimum weight perfect matching by just multiplying the weights on the edges by -1.

If a bipartite graph is complete and balanced, then a perfect matching exists. Next, the plan is to solve this weighted version of matching using linear programming. We will also need to use the augmenting path algorithm to find a maximum size matching as a subroutine. In a sense, it will be a reduction from weighted to unweighted matching. Let us first see the LP formulation of minimum weight perfect matching.

## 2.1 Linear program for min weight perfect matching

Let  $\{w_e\}_{e \in E}$  be the given weights on the edges. As usual, we will have one variable for each edge, that is,  $\{x_e\}_{e \in E}$ . We can write the following LP relaxation for minimum weight perfect matching.

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

**Homework:** Argue that integral optimal value is equals to the fractional optimal value. In other words, there is always an integral optimal solution for this LP. And thus, it is a exact LP formulation for minimum weight bipartite matching.

**Dual LP.** The following will be the dual linear program for the above LP.

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

## 2.2 Primal Dual Schema

For the above Primal and Dual LP the **Complementary Slackness Condition** 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  $y_u + y_v = w_e$ . We can apply the Primal Dual Schema to the problem as follows:

1. First we find some feasible dual solution. This can be done by first taking the dual variables for the left vertices to be zero and then choosing the dual variables for right vertices according to the constraints. To be specific, let  $L, R$  be sets of left and right vertices respectively. Set  $y_u = 0$  for  $u \in L$  and for  $v \in R$ , set  $y_v = \min\{w_e \mid e \text{ is incident on } v\}$ . In Fig. ??, we have constructed a feasible dual solution by giving the vertices on the left value zero and assigned values to the vertices on the right appropriately.

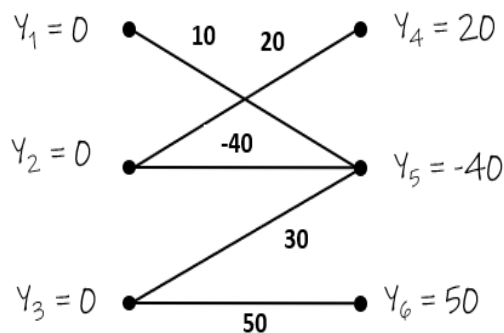


Figure 5: Example of weighted graph with a dual feasible solution.

2. The next step is to try and find a Primal feasible solution that satisfies complementary slackness.

**Tight Edges:** which satisfy  $w_e = y_u + y_v$ . If dual constraints corresponding to  $e$  is not tight, then  $x_e$  must be 0. Let  $G'$  be a subgraph having only the tight edges. Now, try to find a perfect matching in  $G'$ . If we find one then that will be a primal feasible solution satisfying complementary slackness.

The rest of the primal-dual scheme will be covered in the next lecture.