COMPSCI 330: Design and Analysis of Algorithms **Mar 30, 2020**

Lecture 17: Bipartite Matching, Max Flow

Lecturer: Rong Ge *Scribe: Haoming Li*

Overview

In this lecture, we will discuss two problems, maximum bipartite matching and maximum flow, and the linear programs that solve them. We will also interpret the dual of the LP for maximum flow.

17.1 Maximum Bipartite Matching

Here's a motivating problem: There are a number of courses and classrooms. Because of different requirements (capacity, facility, location etc.) each course can only use a subset of classrooms. For each course, we are given the list of classrooms that it can use. We want to match as many courses to classrooms as we can. Each course only needs one classroom, and each classroom can only hold one course.

This problem can be modeled as a *maximum bipartite matching* problem. In this problem, we are given a *bipartite graph* $G = (U, V, E)$, a graph whose vertices can be divided into two disjoint sets U and V such that every edge connects a vertex in U to one in V. We want to choose as large a subset of edges $M \in E$ as possible that forms a *matching*, a set of edges that do not share any vertices.

17.1.1 LP for Maximum Bipartite Matching

We can use a linear program to solve for a maximum matching. For each edge (i, j), we will have one variable x_{ij} that takes on value 1 if $(i, j) \in M$ or 0 if $(i, j) \notin M$. That is, a pair of course and classroom (i, j) that can be matched ends up either matched or not. The objective function follows immediately:

$$\max \sum_{(i,j)\in E} x_{ij}$$

That is, we want to maximize the number of edges that are in the matching. The constraint we have to respect in this problem is that every course i uses at most 1 classroom and every classroom j holds at most 1 course. That is:

$$\sum_{(i,j)\in E} x_{ij} \leq 1, \ \forall i \in U$$

$$\sum_{(i,j)\in E} x_{ij} \leq 1, \ \forall j \in V$$

Lastly, we may relax the integral constraint on the variable, so that we can have a linear program (LP) as opposed to an integer linear program (ILP). That is:

$$0 \leq x_{ij} \leq 1, \ \forall (i, j) \in E$$

As we will see below, the reason we can do this is that, even though there possibly are fractional optimal solutions that assign fractional values to x_{ij} (which cannot be interpreted as a matching), there always exists an integral optimal solution that can be found by the LP.

17.2 Maximum Flow

The *maximum flow* problem states the following: Given a directed graph $G = (V, E)$, where each edge (i, j) has a capacity c_{ij}, a starting vertex s and an ending vertex t, we want to find a maximum *flow* from s to t, with the constraint that 1) the flow sent on every edge cannot exceeds its capacity, and that 2) for every vertex other than s and t, incoming flow must equal outgoing flow.

17.2.1 Using Maximum Flow to Solve Maximum Bipartite Matching

We can reduce a maximum bipartite matching problem to a maximum flow problem. Given a bipartite graph $G = (U, V, E)$, we 1) add a vertex s and edges (s, u) of capacity 1 to every vertex $u \in U$, 2) add a vertex t and edges (v, t) of capacity 1 from every vertex $v \in V$, and 3) make every edge $(u, v) \in E$ directed and with capacity 1. This creates an instance of maximum flow problem.

The *integrality theorem* states that, if all capacities are integers, then there exists an optimal solution for which the amount of flow sent on every edge is an integer. Such integral optimal solution to the maximum flow problem constructed above corresponds to an optimal solution to the original maximum bipartite matching problem.

17.2.2 LP for Maximum Flow

We can use a linear program to solve for a maximum flow. For each edge (i, j), we will have one variable x_{ij} represents the amount of flow sent on this edge. The objective follows immediately:

$$\max \sum_{(s,j)\in E} x_{s,j} \text{ or } \max \sum_{(i,t)\in E} x_{i,t}$$

The flow sent on every edge cannot exceeds its capacity:

$$x_{ij} \le c_{ij},\ \forall (i, j) \in E$$

At every vertex i, incoming flow must equal outgoing flow:

$$\sum_{(i,j)\in E} x_{ij} = \sum_{(k,i)\in E} x_{ki},\ \forall i \in V \backslash \{s, t\}$$

Lastly, the sign constraint says that the amount of flow sent on edges must be positive:

$$x_{ij} \ge 0,\ \forall (i, j) \in E$$

17.2.3 Dual of the LP for Maximum Flow

Let's take the duel of the above LP and try to interpret it. For every capacity constraint on edge (i, j), we will have a dual variable λ_{ij}. For every flow conservation constraint on vertex $i \notin V \backslash \{s, t\}$, we will have a dual variable y_i. For every variable x_{ij} on edges that are not adjacent to s or t, we have constraint:

$$\lambda_{ij} + y_i - y_j \ge 0,\ \forall (i, j) \in E,\ i, j \ne s, t$$

For variables on edges that are adjacent to s or t (assuming we went with the first of the two equivalent primal objective), we have constraint:

$$\lambda_{sj} - y_j \ge 0,\ \forall (s, j) \in E$$

$$\lambda_{it} + y_i \geq 1, \ \forall (i,t) \in E$$

The dual objective comes from the RHS of the canonical primal constraints:

$$\min \sum_{(i,j)\in E} c_{ij}\lambda_{ij}$$

The sign constraints on the dual variables come from the signs of the canonical primal constraints:

$$\lambda_{ij} \geq 0, \ \forall (i,j) \in E$$

$$y_i \in \mathbb{R}, \ \forall i \in V \backslash \{s,t\}$$

We can then simplify this LP to:

$$\min \sum_{(i,j)\in E} c_{ij}\lambda_{ij}$$

$$\lambda_{ij} + y_i - y_j \geq 0, \ \forall (i,j) \in E$$

$$y_s = 0$$

$$y_t = 1$$

With the same sign constraints. This dual LP in fact solves the *minimum cut* problem.

17.2.4 Minimum Cut

A *cut* is a partition of the vertices of a graph into two disjoint subsets. Given a directed graph $G = (V,E)$ with capacity on edges and vertex s and t, a *minimum cut* partitions the vertices into S and T such that 1) $s \in S$, 2) $t \in T$, and 3) the total capacity of the edges that are directed from S to T is minimize. Such edges are said to be *in the cut.*

Inspecting the dual LP above, we see that an optimal solution will assign $\lambda_{ij} = \max\{y_j - y_i, \ 0\}$. Without loss of generality, we can assume $0 \leq y_i \leq 1$. An integral solution will assign $y_i = 0$ if $i \in S$ and $y_i = 1$ if $i \in T$. Meanwhile, it will assign $\lambda_{ij} = 1$ if (i,j) is in the cut and $\lambda_{ij} = 0$ if not.

The strong duality theorem tells us that the optimal solution to the LP for maximum flow equals to that to its dual, the LP for minimum cut. This is known as the *max-flow min-cut theorem.*