CSC 226: Summer 2018: Lab 9

July 17, 2018

1 Bipartite Matching

Problem: We have 6 workers and 8 task to assign to them under the condition that a person can be assigned at most one task. But not every worker qualify for every job. Figure 1 shows the qualification of the workers as a graph. We have a set of vertices $a, \ldots, f$ representing the workers and another set of vertices $1, \ldots, 8$ representing the tasks. An edge between a worker and a task means that that worker qualify for that task, and an absence of an edge means that a person does not qualify for that task. For example, Worker a qualifies for Tasks $1, 4, 5$. Now, how can we assign tasks to the workers so that maximum number of tasks are accomplished?

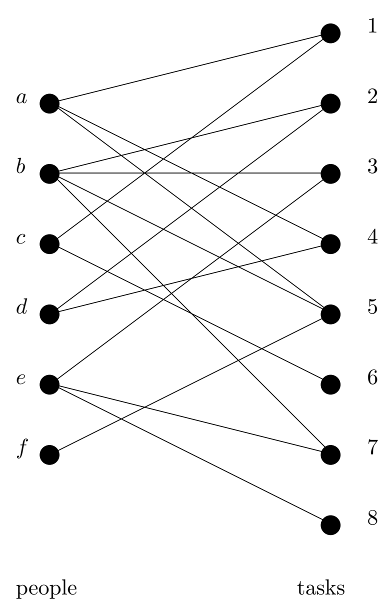

Figure 1: A bipartitte graph representing qualifications of a set of workers for a set of tasks.

The above problem is an example of *maximum bipartite matching*. A *bipartite* graph is a graph where we can partition the vertex set into two sets such that there is no edge between the vertices in the same set. A *matching* is a set of edges in a graph such that no two edges share an endpoint. A *maximum matching* is a matching in a graph with maximum number of edges.

Now try to find a maximum matching in the bipartite graph in Figure 1. How many edges does it have? Is there another matching with the same number of edges?

We will try to find an algorithm for this problem. Can we reduce this problem to a network flow problem?

2 Exercise

Find the maximum flow and a minimum cut in the graph in Figure 2.

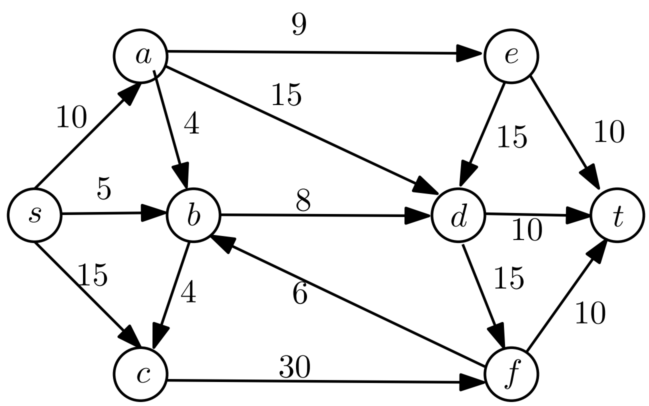

Figure 2: A flow network.