# 1 March 28th, 2019

## 1.1 Bipartite Matching

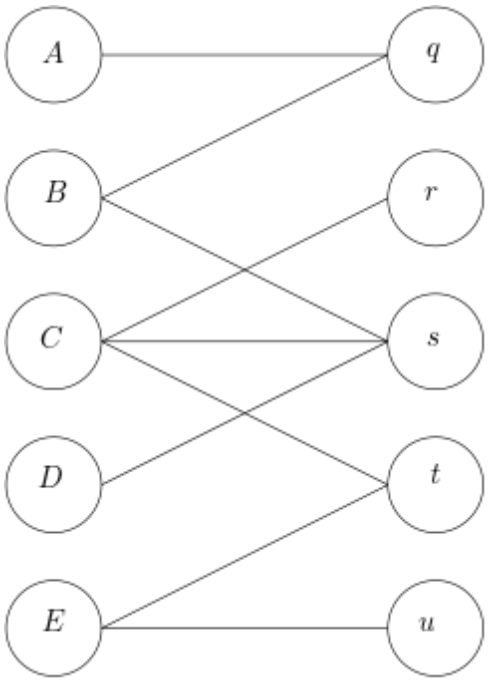


Figure 1: Example of Bipartite Graph

**Definition 1.1.** A **bipartite graph** is a graph with a bipartition, with all edges only crossing the bipartition.

- An edge represents a person willing to do a job
- Each person can be assigned to at most one job
- Each job can be occupied by at most one person
- QUESTION: is it possible to assign the employees such that every employee gets one job and each job is filled?

**Remark 1.2 —** A **maximal matching** is a matching for which if you add another edge, it would not be a matching. A **maximum matching** is the matching with maximum cardinality. A maximum matching is always a maximal, but a maximal is not always a maximum.

**Definition 1.3.** A **neighborhood** of a subset of vertices, $A$, is all the vertices they are connected to a vertex in $A$ by an edge in $E$. e.g.

$$N(\{A, B, D\}) = \{q, s\}.$$

**Remark 1.4 —** The following are necessary conditions to have a perfect matching:

- The size of each side in this partition is the same.
- The size of the neighborhood of any subset is greater than or equal to the size of the subset. e.g. in the example above

$$|N(S)| \geq |S|, \ \forall S \subseteq L.$$

These will be proven to be sufficient later

From weak duality, we have that for any graph, the size of the max matching is less than or equal to the size of the min vertex cover.

## 1.2 Bipartite Matching as a Max Flow Problem

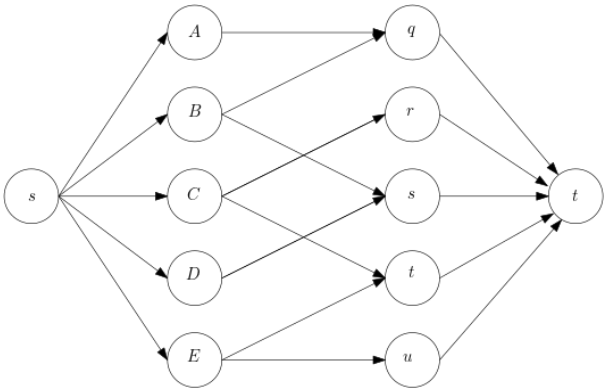


Figure 2: Matching as a Max Flow Problem (all $c(s, v)$ and $c(t, v)$ is 1)

Easy observations:

1. Suppose you have an integral flow of value $k$ in $G'$, the flow network. Then you have a matching of size $k$ in $G$.
2. Suppose you have a matching os size $k$ is $G$. Then you have a flow of value $k$.

**Algorithm 1.5**

We can solve the problem of finding a max matching in a bipartite graph in polynomial time by reducing it to max flow as follows:

- Let $L, R$ denote the vertices on the left and right side of the bipartite graph respectively. Direct edges of the graph from left to right.
- Add 2 new nodes $s$ and $t$, and add direct edges from $s$ to all vertices in $L$ and from all vertices in $R$ to $t$.
- Set $c(s, v) = c(v, t) = 1$ and $\infty$ for edges from $L$ to $R$.
- Find the max flow in the network, assuming that the flow is integral.
- Return the edges of the graph in which flow is 1.

This algorithm outputs a maximum matching.

*Proof.* Follows from (1) and (2) □

**Remark 1.6 —** Note that we can use any FF method, since the number of iteration is bounded by $|V|$. Since each iteration takes $O(|E|)$ time, we have a $O(|V||E|)$ algorithm.

We can run the FF method to find the maximum matching, as all capacities are integral values (all edges have capacity 1). If we have a flow, we will have a matching equal to $|f|$.

**Remark 1.7 —** This algorithm only works for bipartite graphs, but there is another polynomial time algorithm that can find the maximum matching of arbitrary graphs (albeit more complicated).

## 1.3 Perfect Matching in Bipartite Graphs

**Definition 1.8.** A **perfect matching** is a matching which "covers" all the vertices.

**Theorem 1.9** (Hall's Theorem)

A bipartite graph $G(V, E)$ with bipartition $(L, R)$ has a perfect matching if and only if:

1. $|L| = |R|$ ; and
2. $\forall A \subseteq L$, we have
$$|A| \leq |N(A)|.$$

These properties have been shown to be necessary conditions, but they are also sufficient conditions (using max-flow min-cut).

*Proof.* Suppose that $G$ does not have a perfect matching and $|L| = |LR|$. Then we will prove that $\exists A \subseteq L$ such that $|A| > |N(A)|$, i.e. (2) will be violated. We will use the same reduction to the max-flow problem. Recall the flow network $G'$ as before.

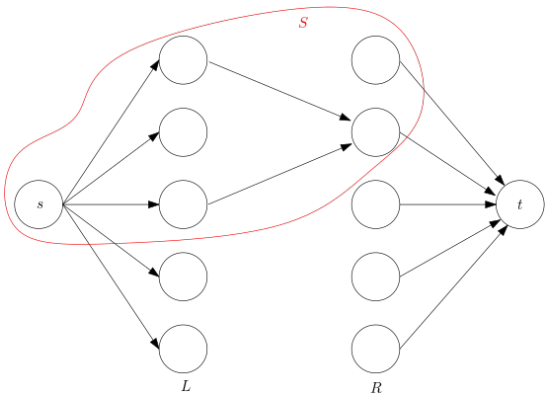


Figure 3: Example of a cut $S$

Consider the $s-t$ cut $S$ of minimum capacity. The capacity of this cut is

$$|L - S| + |R \cap S|.$$

Note that there is no edges from $L \cap S$ that goes to $R \cap (V - S)$, since the capacity of the cup would be infinite, which would violate the max flow min cut thoerem, since the max flow is clearly finite. This means that

$$N(L \cap S) \subseteq R \cap S.$$

Since $G$ does not have a perfect matching, if we let $|L| = |R| = n$

$$\text{Size of max matching} < n \implies \text{Value of max flow } < n \implies \text{Capacity of min cut } < n.$$

Note that

$$|L - S| + |R \cap S| < n = |L - S| + |L \cap S|,$$

which means

$$|L \cap S| > |R \cap S|.$$

□