Bipartite Matching

Frank

November 15, 2021

1 Bipartite Matching

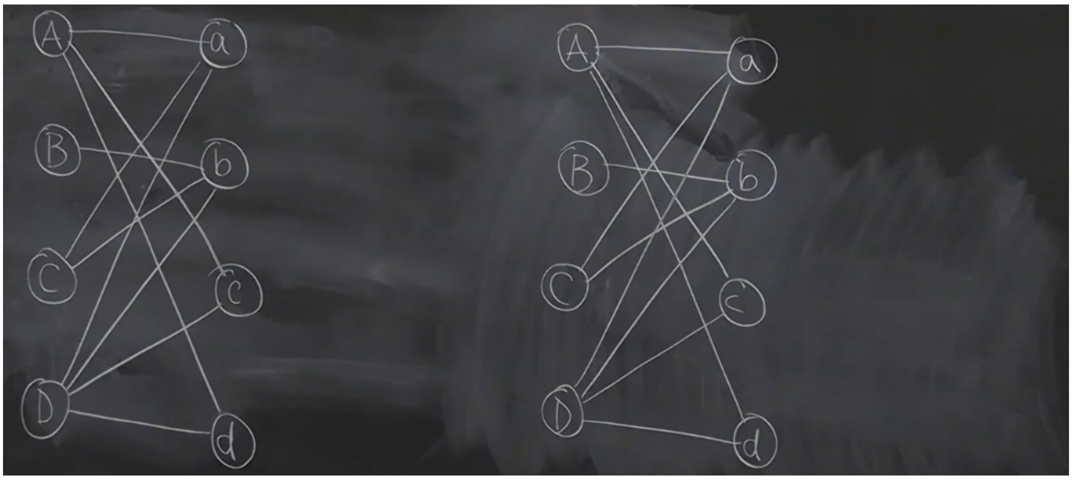
1.1 Bipartite Graph

- Graphs that is separated into 2 parts, where nodes of each of the parts do not share any edges.

  For example, this could be a graph where one side is people, and the other side is tasks.

- Formally, a Bipartite graph is $G = (V, E)$: undirected graph s.t. V can be partitioned into X and Y s.t. each edge connects $x \in X$ to $y \in Y$ but never connects within its partition

  Thus, we have $G = ((X, Y), E)$



- Matching M in G : $M \subseteq E$ s.t. no node appears in two edges in M

  For example, $\{\{A, a\}, \{C, b\}\}$ is a matching

  And $\{\{A, a\}, \{C, b\}, \{D, c\}\}$ is a maximal matching (cannot be extended)

  $\{\{A, c\}, \{B, b\}, \{C, a\}, \{D, d\}\}$ is a maximum matching (Max cardinality)

  maximum matching $\subseteq$ maximal matching

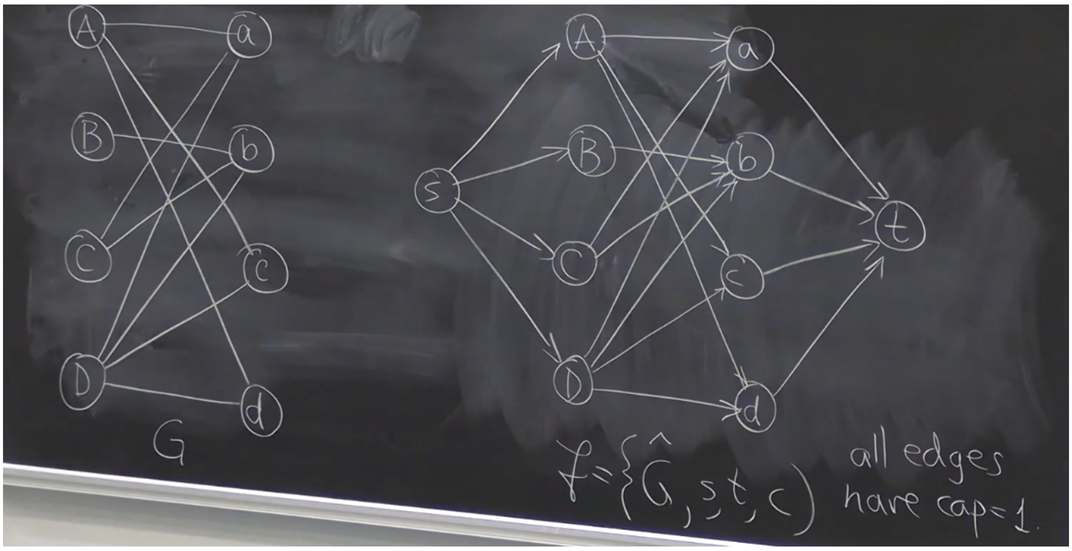
  We are interested in only maximum matching

1.2 Problem

- **Input:** Bipartite $G = ((X, Y), E)$
- **Output:** A max matching in G. Max assignment of each $x \in X$ to each $y \in Y$ such that no x is assigned to 2 y and no y is assigned to 2 x

1.3 Solution Using Max Flow

- Using directed edges, create s and connect s to each $x \in X$, and create t and connect each $y \in Y$ to t. Finally, for each edge from X to Y, convert it into a directed edge and assign it capacity of 1.



Note that Integral Flow = traffic on every edge is integer, and by integrality principle, if we have an integral flow, then we must have a max flow where every flow is an integer.

In F, every integral flow gives $f(e) = 0$ or 1 for every edge $e \in E$

We cannot have a non-integer flow, because we cannot assign half of a task to someone in this problem

1.4 Algorithm

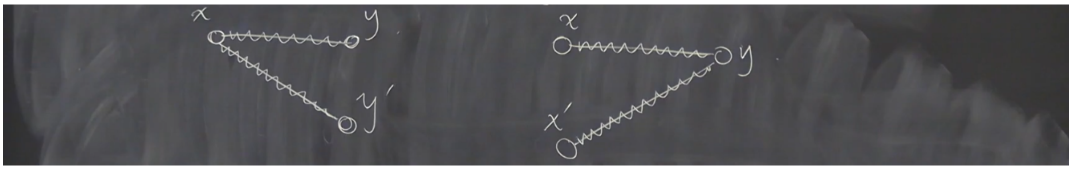
procedure BIPARTITE MATCH(G)
    Construct flow network F from G as shown ▷ $O(n + m)$
    Use FF to get $f \leftarrow MaxFlow(F)$ ▷ $O(mC) = O(mn)$
    $M = \{\{x, y\} : x \in X, y \in Y, f(x, y) = 1\}$ ▷ $O(m)$
    return M

Running time is thus $O(mn)$

1.5 Correctness

Want to prove: Integral flow of value k in $F \iff$ Matching of size k in G $(\star)$

- $\implies$ Given integral flow $f \in F$, find matching M s.t. $|M| = V(f)$
  - $M = \{\{x, y\} : x \in X, y \in Y, f(x, y) = 1\}$
    To verify M is matching, first consider the invalid cases



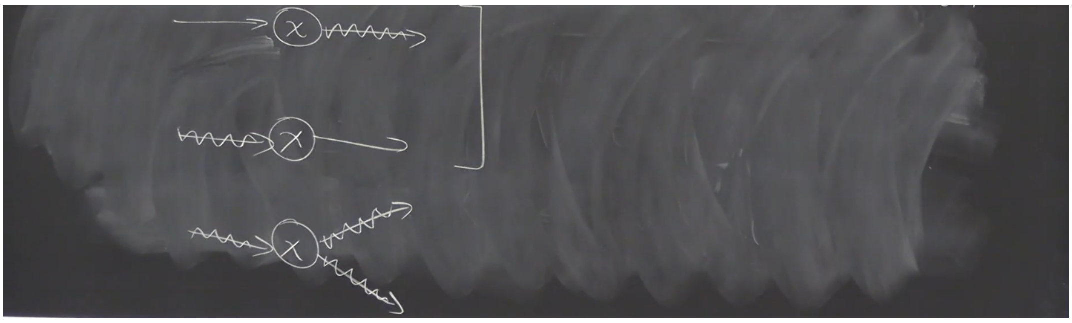
    But this cannot happen, because we required that the flows be integral and every capacity is 1. Thus M is indeed a matching
  - By lemma 2,

$$\begin{aligned} V(f) &= \sum_{e \in out(X) \cap in(Y)} f(e) - \sum_{e \in out(Y) \cap in(X)} f(e) \\ &= \sum_{e \in out(X) \cap in(Y)} f(e) - \ 0 \\ &= |\{\{x, y\} : x \in X, y \in Y, s.t.\ f(x, y) = 1\}| \\ &= |M| \end{aligned}$$

- $\impliedby$ Given matching M, construct integral flow f s.t. $V(f) = |M|$
  - Define

$$f(e) = \begin{cases} 1 & \text{if } e \text{ is on } s \to t \text{ path that contains edge in } M \\ 0 & \text{otherwise} \end{cases}$$

    Verify that f is a flow
    * **Capacity:** Trivial
    * **Conservation:** Consider bad cases:



      Note that the first and second cases are impossible by construction. The third case also follows from the fact that M is a matching

    Then consider that $V(f) = |M|$, then were done.

- **Correctness:**
  - By ($\star$), M in line 4 of algorithm is a matching of size equal to max flow, and there cannot be a flow greater than max flow, thus no matching greater than M

2 Mathematical Reduction

- The technique to transforming a problem into a different problem, then using the solution of that problem to solve our original problem.
- Reduction of bipartite matching to max flow