CS201

Mathematics for Computer Science I

Lecture 18

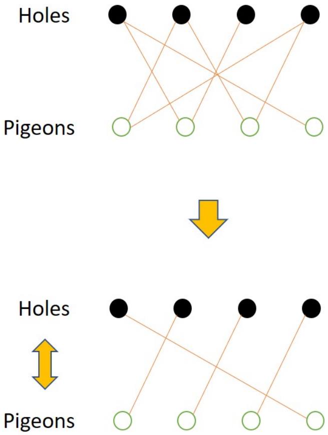
Matching Pigeons to Holes

- Suppose n pigeons are to be put in n holes.
- Further, for each pigeon, there is a set of holes that it can comfortably live in.
- How does one find a mapping of pigeons to holes so that every pigeon lives comfortably?

Matching Pigeons to Holes

- Formulate as graph problem:
  - Consider a bipartite graph $G = (V_P, V_H, E)$ with vertex set $V = V_P \cup V_H$, $|V_P| = |V_H| = n$.
  - V_P represents n pigeons and V_H n holes.
  - Connect $u \in V_P$ with $v \in V_H$ if pigeon u is comfortable in hole v.
  - Need to find a subgraph on V such that every vertex of the subgraph has degree one. It is called a perfect matching.

Example



Existence of Perfect Matchings

Neighbor Set

Let $G = (V, E)$ be a graph. For any $U \subseteq V$, neighbor set of U is defined as:

$$N(U) = \{v \mid v \in V \text{ and for some } u \in U, (u, v) \in E\}.$$

Theorem

A bipartite graph $G = (V_1, V_2, E)$ has a perfect matching if and only if $|V_1| = |V_2|$, and for every $U \subseteq V_1$, $|N(U)| \geq |U|$.

Proof

- Suppose G has a perfect matching.
- Then we must have $|V_1| = |V_2|$ and $|N(U)| \geq |U|$ for any $U \subseteq V_1$.
- Suppose $|V_1| = |V_2|$ and $|N(U)| \geq |U|$ for every $U \subseteq V_1$.
- The proof is by induction on $|V_1| = n$.
- Base case of $n = 1$ is true since $|N(V_1)| \geq 1$ implying there is an edge connecting vertex in V_1 to V_2.
- Assume for $|V_1| \leq n - 1$, and consider the case $|V_1| = n$.

Proof

- Call a $U \subseteq V_1$ critical if $|N(U)| = |U|$.
- Suppose V_1 does not have any critical subset.
- Take any vertex $u \in V_1$.
- There will be an edge incident on u since $|N(\{u\})| \geq 1$.
- Fix any such edge and match u to the other endpoint of the edge.
- Remove u and the matched vertex from G to get induced subgraph $H = (V_1', V_2', E')$.
- For any $U \subseteq V_1'$, consider U as subset of V_1 in G. Since U is not critical, $|N(U)| \geq |U| + 1$ in G.
- Therefore, in H:

$$|N(U)| \geq |U|.$$

Proof

- Since $|V_1'| = n - 1$, applying induction hypothesis on H, we get a perfect matching of H.
- Add to this the removed pair of vertices and edge between them to get a perfect matching of G.
- Now suppose V_1 has a critical subset U.
- Consider any subset $U' \subseteq V_1 \backslash U$.
- Since $|N(U' \cup U)| \geq |U' \cup U| = |U'| + |U| = |U'| + |N(U)|$, and $|N(U' \cup U)| = |N(U') \backslash N(U)| + |N(U)|$,

$$|N(U') \backslash N(U)| \geq |U'|.$$

- Defining H to be induced subgraph on $(V_1 \backslash U, V_2 \backslash N(U))$, we have that for any $U' \subset V_1 \backslash U$, $|N(U')| \geq |U'|$ in H.

Proof

- Therefore, by induction hypothesis, H has a perfect matching.
- Let H' be induced subgraph of G on $(U, N(U))$.
- It is clear that for any subset $U' \subseteq U$, $|N(U')| \geq |U'|$ both in G and H'.
- Hence, H' also has a perfect matching by induction hypothesis.
- Combining the two perfect matchings of H and H' gives a perfect matching of G.

Spanning Trees of Graphs

Given a connected graph $G = (V, E)$, a spanning tree of G is a subgraph of G on V that is a tree.

- Useful in finding paths between vertices.
- Can be extended to spanning forest for any graph.

Finding a Spanning Tree

- Pick any vertex of a connected graph G, and call it root, and mark it reachable.
- For any vertex marked reachable:
  - Remove all edges with other endpoint also marked reachable.
  - For remaining edges, mark the other endpoints reachable.
- The resulting subgraph of G is a spanning tree.
- For a graph with multiple connected components, the same algorithm can be repeated to identify a spanning forest.

Finding a Perfect Matching

- Let $G = (V_1, V_2, E)$ be a bipartite graph with $|V_1| = |V_2| = n$.
- Represent a perfect matching as a bijection $\pi : V_1 \mapsto V_2$.
- Suppose π has been defined for a subset U of V_1.
- Take any vertex $u \in V_1 \backslash U$.
- If there is a $v \in V_2 \backslash \pi(U)$ such that $(u, v) \in E$, then let $\pi(u) = v$, and add u to U, extending π.
- Suppose $N(\{u\}) \subseteq \pi(U)$.

Finding a Perfect Matching

- Construct a graph $H = (V_1, E_H)$ as follows:
  - Edge $(u_1, \pi^{-1}(u_2)) \in E_H$ iff $(u_1, u_2) \in E$.
  - In other words, u_1, u_2, $\pi^{-1}(u_2)$ is a path of length two in G.
- Compute a spanning forest of H, and consider a tree T of this forest rooted at u.
- We have $|N(T) \cap \pi(U)| \leq |T| - 1$ in G:
  - For $v \in N(T) \cap \pi(U)$, $\pi^{-1}(v) \in T$ and $\pi^{-1}(v) \neq u$ since $u \notin U$.
- If $N(T) \subseteq \pi(U)$ then $|N(T)| \leq |T| - 1$ and there is no perfect matching.
- Otherwise, let vertex u' in T such that $N(\{u'\}) \not\subseteq \pi(U)$ and $v' \in N(\{u'\} \backslash \pi(U)$.

Finding a Perfect Matching

- Let $v_0 = u$, v_1, v_2, …, $v_k = u'$ be the path in T from u to u'.
- This path corresponds to the path $v_0 = u$, w_1, v_1, w_2, v_2, …, w_k, $v_k = u'$ in graph G, with $v_i = \pi^{-1}(w_i)$ for $1 \leq i \leq k$.
- Redefine π as:
    - $\pi(u) = w_1$, $\pi(v_1) = w_2$, …, $\pi(v_{k-1}) = w_k$, and $\pi(u') = v'$.
- This extends π to one more vertex increasing the size of U.
- Repeating this eventually makes π a perfect matching.
- Observe that this algorithm also gives an alternative proof of theorem on existence of perfect matchings.