COL751 - Lecture 22

1 Maximum Matching

A *matching* M in an undirected graph $G = (V, E)$ is a set of edges that have no common vertex. Given a matching M, an *alternating path* w.r.t. M is a path whose edges alternates between matched and unmatched edges. An *augmenting path* w.r.t. M is an alternating path that starts and ends at free (unmatched) vertices.

It is easy to see that a matching M having and augmenting path can not be optimum. Berge proved that converse also holds.

Lemma 1 (Berge's Lemma)**.** *A matching M is maximum iff it has no augmenting path.*

Proof: Let M be a matching for a graph G.If there is an M-augmenting path, say P, then we can get a matching of size $1 + |M|$ by switching edges along P. So a matching having an augmenting path cannot be maximum.

Next let us assume M is not maximum. We will prove that there exists an M-augmenting path. Let M_{opt} be a matching in G of maximum size. Consider the graph $H = (V, M \oplus M_{opt})$. Since degree of each vertex in H is at most two, H (after ignoring singleton vertices) is just a union of vertex-disjoint paths and cycles of even length. Moreover, no adjacent edges in H can lie in M or M_{opt}. Since $|M_{opt}| \gneq |M|$, there must exist a path P in H such that $|M_{opt} \cap P| \gneq |M \cap P|$. Such a path P must be an M-augmenting path as its endpoints must be unmatched. This proves the claim. $\square$

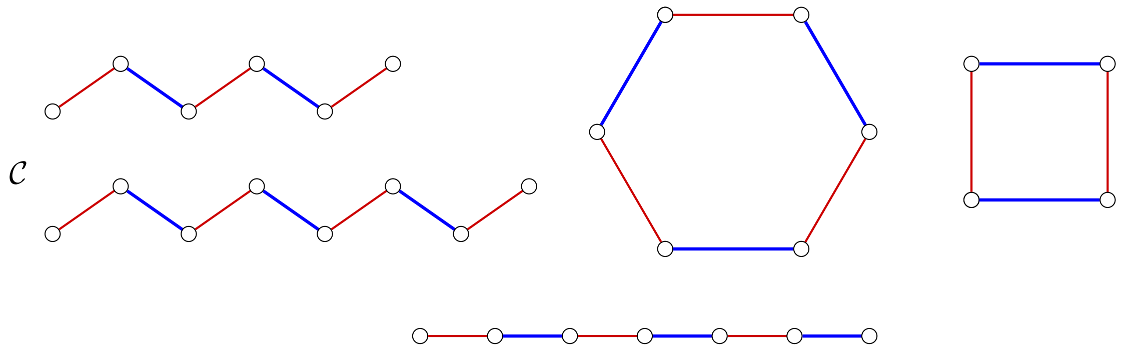

Figure 1: Depiction of paths and cycles in $M \oplus M_{opt}$. Edges in M are shown in blue, and edges in M_{opt} are shown in red.

Remark For any matching M, there exists a set $\mathcal{C}$ of size $(|M_{opt}| - |M|)$ comprising of vertex-disjoint augmenting-paths which when xored with M gives a maximum matching.

Lemma 2. *Let M be a matching satisfying that each M-augmenting-path in G has length at least L, then $(|M_{opt}| - |M|)$ is at most n/L.*

Proof: Let $\mathcal{C}$ be a collection of vertex-disjoint M-augmenting paths whose edges lie in $M_{opt} \oplus M$. See Figure 1. As each M-augmenting path in G has length at least L, we have $|\mathcal{C}| \leqslant n/|L|$. Therefore, $(|M_{opt}| - |M|) = |\mathcal{C}| \leqslant n/|L|$. □

Homework Prove that upper bound in Lemma 2 can be strengthened to $2|M_{opt}|/L$.

2 Maximum Matching in Bipartite graphs

We discussed earlier a max-flow-based approach for computing maximum-matching in bipartite graphs which had $O\big(\text{TIME}(\textit{max-flow})\big) = O(mn)$ time complexity. We will study in this section an improved algorithm for computing maximum matching in bipartite graphs that was given by Hopcroft and Karp (1973).

$M \leftarrow \emptyset$;
while *(M is not maximum)* **do**
    $\mathcal{C} = (P_1, \dots, P_\alpha) \leftarrow$ Blocking set of augmenting paths;
    $M = M \oplus P_1 \oplus \cdots \oplus P_\alpha$;
end
Return M;

Algorithm 1: Hopcraft-Karp(G)

Algorithm 1 presents an outline of the Hopcraft-Karp algorithm. We define below the notion on *blocking set of augmenting paths.*

Definition 1. *Let M be a matching in graph G, and 'k' be length of smallest M-augmenting path. Then a* **blocking set of augmenting paths** *with respect to M is a set $\{P_1, \dots, P_\alpha\}$ of vertex-disjoint augmenting paths of length k that is inclusion maximal.*

Let $H = H(M)$ be an *orientation* of $G = (X, Y, E)$ such that all unmatched edges in G are directed from X to Y, and matched edges are directed from Y to X. The graph H satisfies the following property.

Property 1. *There exists 1-1 correspondence between simple directed paths in H and alternating paths in G.*

Let S be a set comprising of unmatched vertices in X. For $i \geqslant 0$, let L_i be the set of vertices at distance i from S in H. See Figure 2. Further, let H^* be a layered graph with vertex set

$$V^* \;=\; \bigcup_{i<k} L_i \;\; \bigcup \;\; (L_k \cap \text{unmatched-vertices}),$$

and edge-set E^* comprising of those edges in H that lie in $(L_{i-1} \times L_i)$, for some $i \in [1, k]$.

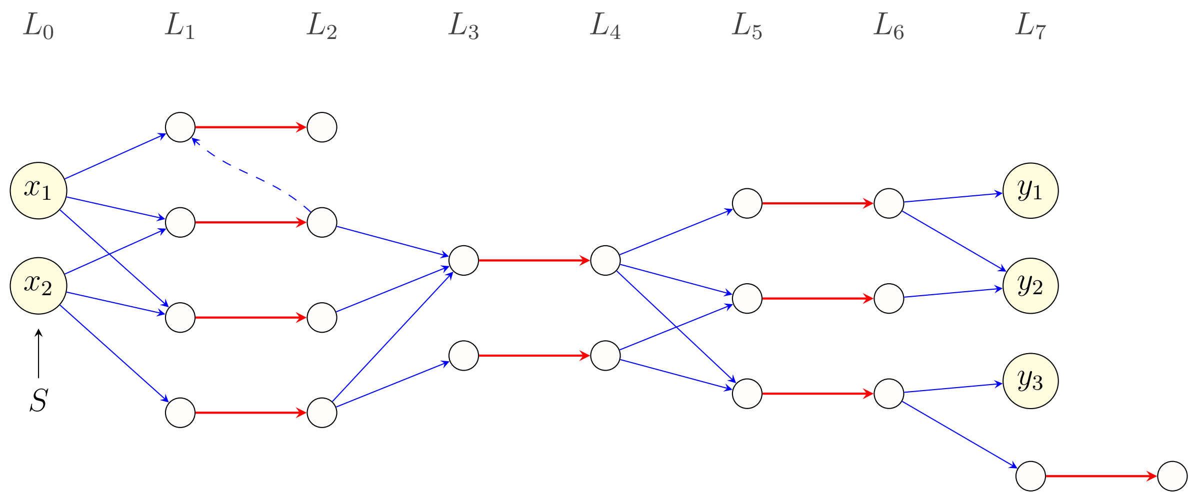

Figure 2: Layered structure obtained from BFS traversal in H from set S as supernode.

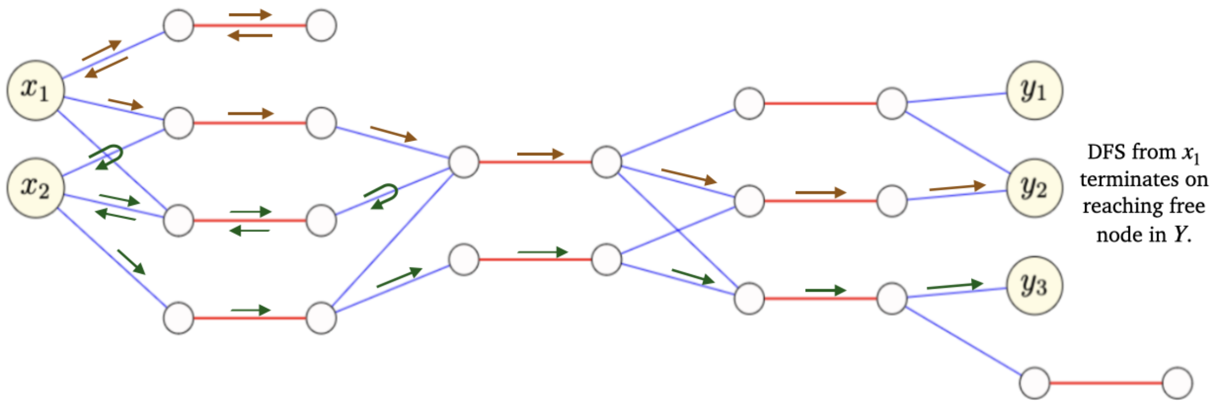

Figure 3: Depiction of DFS traversal from vertices x_1, x_2 lying in S in layered graph H^*. DFS traversal from each $x_i \in S$ terminates on reaching free nodes in Y.

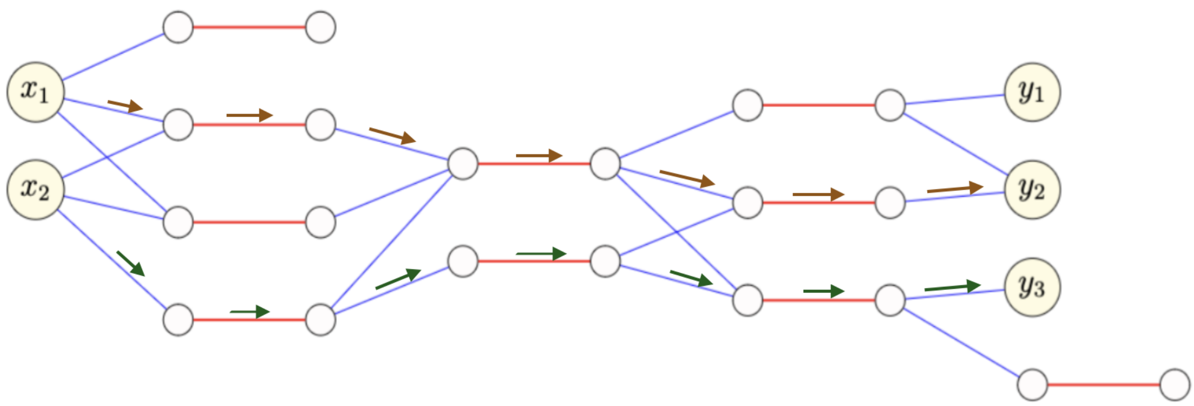

Figure 4: A blocking set of augmenting paths. Note that after updating matching M as $M = M \oplus P_1 \oplus \cdots \oplus P_\alpha$, (i) matched and unmatched edges in $P_1, \ldots, P_\alpha$ will be swapped, (ii) orientation of edges in $P_1, \ldots, P_\alpha$ will thus be reversed in $H = H(M)$.

Lemma 3. *A blocking set of augmenting paths w.r.t. a matching M in a connected bipartite graph $G = (X, Y, E)$ can be computed in $O(m)$ time, where m denotes the number of edges in G.*

Proof: Observe that k, i.e. length of smallest augmenting path in G, can be computed in $O(m)$ time by simply performing a BFS in H from the set S. In particular, $k = \min\{dist_H(S, y) \mid y \in Y \text{ is free}\}$. Similarly, sets $L_0, L_1, \ldots, L_k$, and graph H^* can also be computed in $O(m)$ time.

To compute $\mathcal{C}$, we perform a modified DFS traversal from vertices in S in graph H^* such that traversal from each $x_i \in S$ terminates as soon as it encounters a free vertex in set Y. The time complexity of this step is $O(m)$ as each vertex is visited at most once during the DFS traversal from nodes in S. $\square$

Lemma 4. *Let M be a matching in a bipartite graph $G = (X, Y, E)$, k be length of smallest M-augmenting path, and $\mathcal{C} = (P_1, \ldots, P_\alpha)$ be a blocking set of augmenting paths. Then each augmenting path w.r.t. matching $M^* = (M \oplus P_1 \oplus \cdots \oplus P_\alpha)$ has length larger than k.*

Proof: Let $\mathcal{E}$ be collection of edges lying in $P_1, \ldots, P_\alpha$, and $V_\mathcal{E} = \cup_{i=1}^{\alpha} V(P_i)$ be endpoints of edges in set $\mathcal{E}$. Let Q be a smallest augmenting path with respect to matching M^*. We have following two cases:

Case 1: $Q \cap V_\mathcal{E} = \emptyset$.
In this case length of Q must be at least $k + 1$ as blocking set $\mathcal{C}$ was inclusion maximal.

Case 2: $Q \cap V_\mathcal{E} \neq \emptyset$.
In this case Q must contain an edge matched under M^* lying in set $\mathcal{E}$. This is because all M^*-matched edges incident to vertices in $V_\mathcal{E} = \cup_{i=1}^{\alpha} V(P_i)$ lies in $\mathcal{E}$. Since matched edges lying in $H(M^*)$ are directed from L_i to L_{i-1}, for some $i \in [1, k]$, the length of augmenting path Q must be at least $k + 2$. $\square$

Lemma 5. *The number of iterations of While loop in Algorithm 1 is at most $2\sqrt{n}$.*

Proof: By Lemma 4, after each iteration of While loop in Algorithm 1, the length of smallest augmenting path increases. So, after first $\sqrt{n}$ iterations of While loop, the length of smallest augmenting path is at least $\sqrt{n}$. At this stage, due to Lemma 2, $|M_{opt}|$ and $|M|$ differs by at most $\sqrt{n}$. This proves that the total number of iterations must be at most $2\sqrt{n}$ as each iteration increases the size of matching by at least one. $\square$

We thus have the following result.

Theorem 1 (Hopcraft and Karp, 1973)**.** *There exists an algorithm that given any n vertex, m edges connected bipartite graph G computes a maximum matching of G in $O(m\sqrt{n})$ time.*

Homework Design an algorithm that for any connected bipartite graph $G = (X, Y, E)$ computes in $O(|E|)$ time a matching M of size at least $0.999 \times |M_{opt}|$.