Lecture 13: Matchings

Harvard SEAS - Fall 2024 *2024-10-15*

1 Announcements

- Midterm on Thursday
- Problem Set 5 distributed tomorrow, due Wed 10/23.
- Embedded EthiCS Module next Tuesday 10/22. You are expected to attend; there will be material on ps6 building on the module.

Recommended Reading: Cormen–Leiserson-Rivest–Stein, Sec. 25.1.

2 Recap

Definition 2.1. For a graph $G = (V, E)$, a *matching* in G is a subset $M \subseteq E$ such that every vertex $v \in V$ is incident to at most one edge in M.[1] Equivalently, no two edges in M share an endpoint.

If a vertex v is incident to an edge in M, we say v is *matched* by M; otherwise we say it is *unmatched*.

The problem of finding the largest matching in a graph is called Maximum Matching.

Input	: A graph $G = (V, E)$
Output	: A matching $M \subseteq E$ in G of maximum size

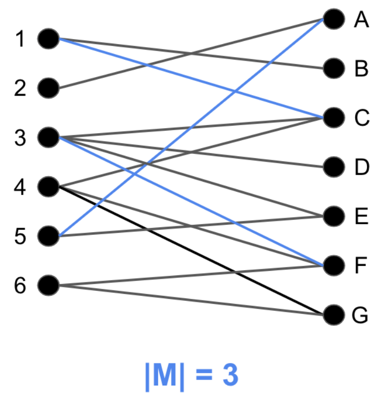
Computational Problem Maximum Matching

Definition 2.2. Let $G = (V, E)$ be a graph, and M be a matching in G. Then:

1. An *alternating walk* W in G with respect to M is a walk $(v_0, v_1, \ldots, v_\ell)$ in G such that for every $i = 1, \ldots, \ell - 1$, $\{v_{i-1}, v_i\} \in M \Leftrightarrow \{v_i, v_{i+1}\} \in E \setminus M$.
2. An *augmenting path* P in G with respect to M is an alternating walk in which v_0 and v_ℓ are respectively unmatched by M, and in which all of the vertices in the walk are distinct, and $\ell \geq 1$.

[1]Saying a vertex v is *incident* to an edge e is another way of saying v is an endpoint of e. It is more symmetric, in that we would also say that e is incident to v.

Example: Consider the following graph with $M = \{\{1, C\}, \{3, F\}, \{5, A\}\}$.



We check whether the following sequences of vertices constitute an alternating walk or an augmenting path.

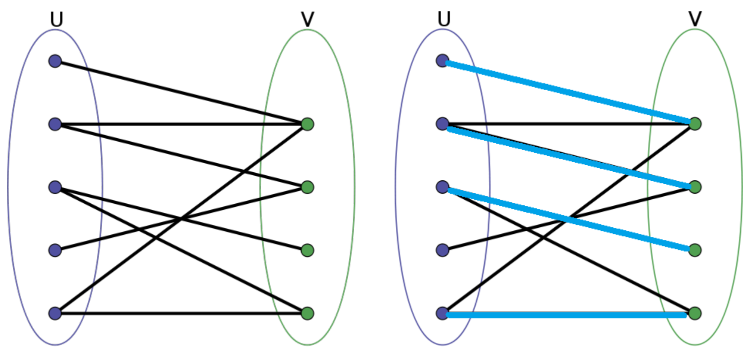
Sequence	Alternating Walk	Augmenting Path
$2, A, 5$	Yes	No
$G, 4$	Yes	Yes
$3, F, 4, C, 3$	No	N/A
$1, C, 6$	No	N/A
$6, F, 3, C, 1, B$	Yes	Yes

Theorem 2.3 (Berge's Theorem)**.** *Let $G = (V, E)$ be a graph, and $M \subseteq E$ be a matching. M is not a maximum-size matching in G if and only if G has an augmenting path with respect to M.*

3 A Motivating Application

Kidney Exchange: In the US, there is a nationwide system for arranging kidney transplants, matching patients (who need a kidney) with donors (who are willing to donate a kidney). Each donor can only donate one kidney (they need their other one to survive!) and only to certain patients (due to blood type and HLA type compatibilities). There over 100,000 patients currently on the kidney waiting list in the US, with a little over 25,000 donations happening per year, and patients spending an average of about 3.6 years on the waiting list. Algorithms like what we will cover play a significant role in saving patients through kidney donations.

Q: If our goal is to maximize the number of kidneys donated, how can we model this as a matching problem?
Create a graph with $u_1, \ldots, u_t$ representation potential donors, and $v_1, \ldots, v_l$ representing potential recipients. Then, for u_i, v_j we place an (undirected) edge connecting the two if they are compatible.



Additional considerations in real-life kidney exchange (to be discussed more in Embedded EthiCS Module next Tue!):

- Priority: We may want to give higher priority to some patients, like those who have been waiting longest, or have more advanced kidney disease, or are organ donors themselves.
- Donor preference: most kidney donors sign up because they have a loved one who needs a kidney, but aren't compatible with each other. So they are only willing to donate their kidney if their loved one receives a kidney from someone else.
- Chains: due to the donor preference above, and laws disallowing contracts around organ donations, a donor may only want to do their donation if it is nearly simultaneous with their loved one receiving a kidney. This leads to simultaneous or near-simultaneous surgeries, with the longest chain to date (Dec 2020) involving 35 donations (70 surgeries) over multiple locations. See the Grey's Anatomy episode *There's no "I" in Team* (season 5, episode 5) for a fictional depiction of such a "domino surgery."
- Location: It is easiest and most cost effective if the exchange happens in the same hospital, though it has become common to ship the kidneys on ice from the donor's hospital to the patient's hospital!

4 Matching vs. Independent Sets

This section was not covered in lecture, and is optional (but recommended) material.

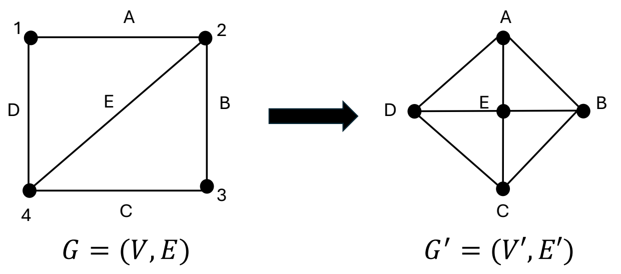
Like IntervalScheduling-Optimization, Maximum Matching can be viewed as a special case of the Independent Set problem we studied last time, i.e. there is a very efficient reduction from Maximum Matching to Independent Set:

Given a matching instance $G = (V, E)$, want to reduce to an independent set instance $G' = (V', E')$. In max matching, we want to find a set of edges that don't conflict, whereas independent set tries to find a set of vertices that don't conflict. Thus, let

$$V' = E \quad \text{and} \quad E' = \{\{e, e'\} : e, e' \text{ share an endpoint}\}.$$

(The graph G' is known as the *line graph* of G.) Then the maximum-size independent sets in G' are exactly the same as the maximum-size matchings in G.

For example:



Unfortunately, the fastest known algorithm for Independent Set runs in time approximately $O(1.2^n)$. However, as we saw last time for IntervalScheduling-Optimization, special cases of IndependentSet can be solved more quickly. Matching is another example!

5 Maximum Matching Algorithm

Like in a greedy strategy, we will try to grow our matching M on step at a time, building a sequence $M_0 = \emptyset, M_1, M_2, \ldots$, with $|M_k| = k$. However, to get M_k from M_{k-1} we will sometimes do more sophisticated operations than just adding an edge. Instead we will rely on Berge's Theorem, which tell us that if our matching is not of maximum size, then there is an augmenting path. We will obtain an algorithm by the following two lemmas:

Lemma 5.1. *Given a graph $G = (V, E)$, a matching M, and an augmenting path P with respect to M, we can construct a matching M' with $|M'| = |M| + 1$ in time $O(n)$.*

Lemma 5.2. *Given a* bipartite *graph $G = (V, E)$ and a matching M that is not of maximum size, we can find an augmenting path with respect to M in time $O(n + m)$.*

Since a matching can be of size at most $n/2$, repeatedly applying these two lemmas gives us an algorithm that runs in time $(n/2) \cdot (O(n) + O(n + m)) = O(n \cdot (n + m))$. Moreover, by eliminating isolated vertices, in time $O(n)$ we can reduce to the case where $n \leq 2m$, giving us a run time of $O(n) + O(n \cdot (2m + m)) = O(nm)$. Thus we have:

Theorem 5.3. *Maximum Matching can be solved in time $O(mn)$ on bipartite graphs with m edges and n vertices.*

Proof of Lemma 5.1.
Let $P = (v_0, \ldots, v_\ell)$ be an augmenting path. We have that $\{v_0, v_1\} \in E \setminus M$ since v_0 is unmatched, and $\{v_{\ell-1}, v_\ell\} \in E \setminus M$ since v_ℓ is unmatched. Thus, let

$$M' = (M - \{\{v_1, v_2\}, \{v_3, v_4\}, \ldots, \{v_{\ell-2}, v_{\ell-1}\}\}) \cup \{\{v_0, v_1\}, \{v_2, v_3\} \ldots, \{v_{\ell-1}, v_\ell\}\}.$$

In words, we "flip" each of the edges in this augmenting path: any edge of the augmenting path originally in the matching M is now not in the matching M', and any edge of the augmenting path not in M is in M'.

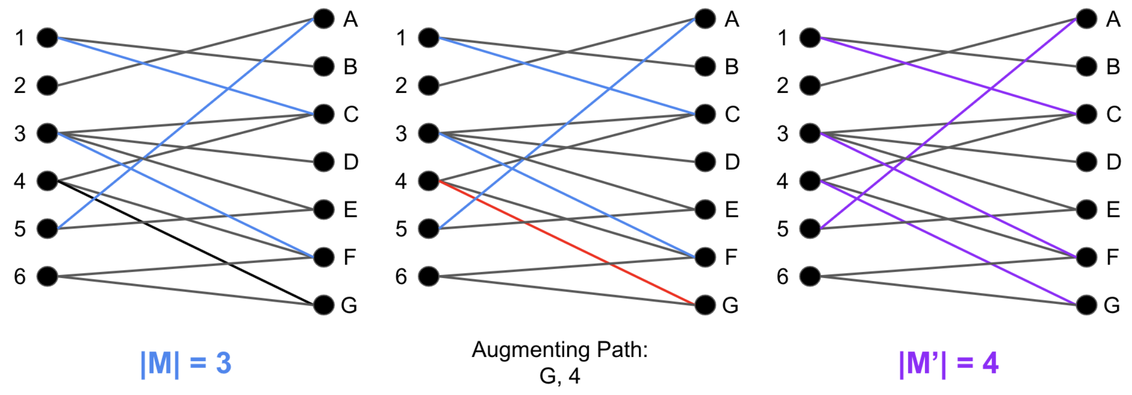
To show M' is a matching, we need to argue that every vertex in G is incident to at most one edge from M'. None of the edges we have added to the matching M are incident any vertices other than $v_0, \ldots, v_\ell$, so we only need to check that the matching property holds for those vertices. v_0

and v_ℓ were previously unmatched, and they are each incident to exactly one of the new edges we have added from P (since paths have no repeated vertices by definition). The vertices $v_1, \dots, v_{\ell-1}$ were each previously incident to exactly one edge of the matching M, which we have removed, and they each are incident to exactly one of the new edges we have added.

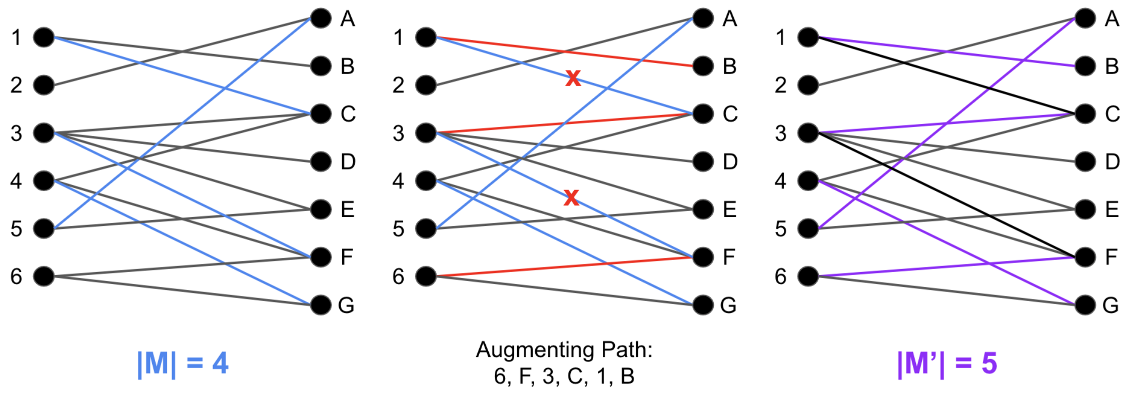
M' matches two more vertices than M, namely v_0 and v_ℓ, so $|M'| = |M| + 1$.

□

Example: We repeatedly apply Lemma 5.1 to our example matching given earlier. We first consider the augmenting path $(G, 4)$ and grow our matching accordingly.



Next, we add the augmenting path $(6, F, 3, C, 1, B)$.



Note that $\{3, F\}$ and $\{1, C\}$ are not in M', since they were part of the original matching M.

In the above example, we gave you the augmenting paths. We still need to show that we can find augmenting paths efficiently (Lemma 5.2). This is the only place we use bipartiteness in the algorithm. The following lemma (whose proof is in the optional Section 6 below) reduces our task to a shortest path problem.

Lemma 5.4. *Let $G = (V_0 \cup V_1, E)$ be bipartite and let M be a matching in G that is not of maximum size. Let U be the vertices that are not matched by M, and $U_0 = V_0 \cap U$ and $U_1 = V_1 \cap U$. Then:*

1. *G has an alternating walk with respect to M that starts in U_0 and ends in U_1.*

2. Every shortest alternating walk from U_0 to U_1 is an augmenting path.

We use this lemma to give the following reduction.

Lemma 5.5. *Finding shortest alternating walks in bipartite graphs reduces to finding shortest paths in directed graphs in time $O(n+m)$.*

Lemma 5.2 follows because Shortest Paths in directed graphs can be found in time $O(n+m)$.

Proof sketch.
Given G, we construct a directed graph G' as follows:

1. Direct edges in $E \setminus M$ to go from V_0 to V_1,
2. Direct edges in M to go from V_1 to V_0,
3. Add a source vertex s with all edges going from s to U_0,
4. Add a target vertex t with all edges going from U_1 to t.

Then walks of length ℓ from s to t in G' correspond to alternating walks of length $\ell - 2$ in G from U_0 to U_1 (just drop s and t from the walk), and thus shortest walks from s to t correspond to shortest alternating walks from U_0 to U_1.

□

Example: If we apply this reduction to second-to-last matching (namely $M = \{\{1, C\}, \{3, F\}, \{4, G\}, \{5, A\}\}$) in our example above and we apply BFS to find a shortest path in G', we explore the vertices of G' as follows:

$$\begin{tikzcd}
 & & & & E \arrow[r] & t \\
 & & A \arrow[r] & 5 \arrow[ur] & & \\
 & 2 \arrow[ur] \arrow[dr] & & & C & \\
s \arrow[ur] \arrow[ddr] & & F \arrow[r] & 3 \arrow[ur] \arrow[dr] & & \\
 & & & & D & \\
 & 6 \arrow[dr] & & & & \\
 & & G \arrow[r] & 4 & &
\end{tikzcd}$$

The edges colored in blue are edges in M, which are directed from right to left. Thus our shortest path from s to t is $s \to 2 \to A \to 5 \to E \to T$, and our augmenting path is $2 \to A \to 5 \to E$ (shorter than the one we gave in the earlier example).

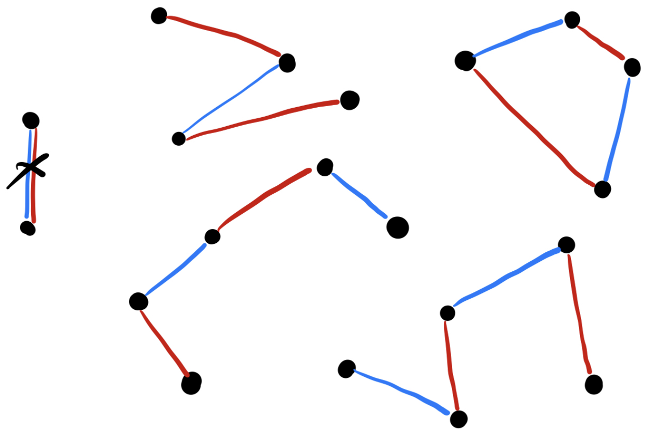
Putting it all together, we have the following algorithm:

```
1  MaxMatchingAugPaths($G$)
   Input  : A bipartite graph $G = (V, E)$
   Output : A maximum-size matching $M \subseteq E$
2  Remove isolated vertices from $G$;
3  Let $V_0, V_1$ be the bipartition (i.e. 2-coloring) of $V$;
4  $M = \emptyset$;
5  repeat
6      Let $U$ be the vertices unmatched by $M$, $U_0 = V_0 \cap U$, $U_1 = V_1 \cap U$;
7      Use BFS to find a shortest alternating walk $P$ that starts in $U_0$ and ends in $U_1$;
8      if $P \neq \perp$ then augment $M$ using $P$ via Lemma 5.1;
9  until $P = \perp$;
10 return $M$
```

6 Omitted Proofs

These proofs were not covered in lecture and are included here as optional reading, according to your interest.

Proof of Theorem 2.3.



Suppose we have a matching M and a larger matching M'. Then consider $G_\Delta = (V, M \Delta M')$ where the symmetric difference $M \Delta M'$ contains the edges in *exactly one* of M and M'. Note that G_Δ has degree at most 2. This is since an edge has to come from M or M', and both of those have degree at most 1. Furthermore, every path or cycle in this graph alternates between edges in M and M'. Finally, at least one path must start and end with an edge in M'. This is because M' is bigger. Every cycle in G_Δ contains an equal number of edges from M and M', so there must be some path with more M' edges than M edges, and this is only possible if both endpoints are in M'. But this path is an augmenting path by definition, so we are done.

The "only if" part of Theorem2.3 follows from Lemma 5.1. $\square$

Proof of Lemma 5.4.

1. By Theorem 2.3, we know that G has an augmenting path P, with vertices $v_0, v_1, \ldots, v_\ell$. Since v_0 and v_ℓ are unmatched, the edges $\{v_0, v_1\}$ and $\{v_{\ell-1}, v_\ell\}$ are not in M. Since the path is alternating between edges from $E - M$ and from M, the path must then be of odd length ℓ. Since paths in a bipartite graph alternate between V_0 and V_1, we have either $v_0 \in U_0$ and $v_\ell \in U_1$ or $v_0 \in U_1$ and $v_\ell \in U_0$. So either P or the reverse of P is an alternating walk starting in U_0 and ending in U_1.

2. Let W be a shortest alternating walk $v_0, v_1, \ldots, v_\ell$ with $v_0 \in U_0$ and $v_\ell \in U_1$. Assume for sake of contradiction that W has a repeated vertex, i.e. $v_i = v_j$ for some $i < j$. Since the graph is bipartite, $j - i$ must be even. Thus if we remove the vertices $v_{i+1}, \ldots, v_j$ from the W, the path W will still be alternating (since the portion we remove will either begin with M and end with $E - M$ or vice-versa), but will be of shorter length. This contradicts our hypothesis that W was a shortest alternating walk. Thus, our assumption that W has a repeated vertex must have been incorrect.

$\square$