Math 3322: Graph Theory[1] Mikhail Lavrov

# Lecture 14: König's theorem

*October 3, 2024* *Kennesaw State University*

## 1 Notation and plans for today

Graph theorists like to use Greek letters for numerical invariants of graphs. We have recently learned two new such invariants, so let's learn the notation for them.

- If $G$ is a graph, then $\alpha'(G)$ denotes the number of edges in a maximum matching of $G$.

  (The $'$ in the notation indicates that it is an "edge version" of a more fundamental invariant. In this case, we will study the "vertex version" later.)

- If $G$ is a graph, then $\beta(G)$ denotes the number of vertices in a minimum vertex cover of $G$.

For example, consider the following claim from the previous lecture:

**Claim 1.1.** *If $M$ is a matching and $U$ is a vertex cover in the same graph, then $|M| \leq |U|$.*

We can rewrite it using our new notation as follows:

**Claim 1.2.** *For any graph $G$, $\alpha'(G) \leq \beta(G)$.*

Actually, it might not be immediately obvious that Claim 1.1 and Claim 1.2 are the same: the first is talking about an arbitrary matching and vertex cover, and the second is only talking about maximum matchings and minimum vertex covers.

However, if Claim 1.1 holds for all $M$ and $U$, then in particular it holds when $M$ is a maximum matching and $U$ is a minimum vertex cover, which proves Claim 1.2. Conversely, if Claim 1.2 holds, then for any matching $M$ and vertex cover $U$, we have $|M| \leq \alpha'(G) \leq \beta(G) \leq |U|$, proving Claim 1.1. (The inequalities $|M| \leq \alpha'(G)$ and $\beta(G) \leq |U|$ come from the definitions of "maximum" and "minimum".)

In this notation, here is König's theorem, our goal for today:

**Theorem 1.3** (König)**.** *For any bipartite graph $G$, $\alpha'(G) = \beta(G)$.*

Our plan for this proof is to actually give an algorithm that does the following. It starts with any matching $M$, and then either constructs a larger matching $M'$, or find a vertex cover $U$ with $|M| = |U|$. (In the second case, we immediately conclude that $M$ is a maximum matching and $U$ is a minimum vertex cover.)

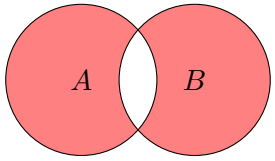
However, we know that we cannot always improve a suboptimal matching simply by adding edges. So the question we'll look at next is: what *does* it take to improve a suboptimal matching?

[1] This document comes from an archive of the Math 3322 course webpage: http://misha.fish/archive/3322-fall-2024

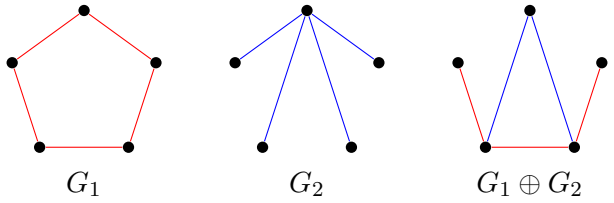
# 2 Augmenting paths

## 2.1 Symmetric difference

Let's back up and talk about an operation called **symmetric difference**. At its most basic, it's defined on sets; if $A, B$ are two sets, then $A \oplus B = (A \cup B) \setminus (A \cap B)$ is the set of all vertices in $A$ or $B$ but not both. Here is a Venn diagram:




We can step this up to graphs. We will only consider graphs $G_1, G_2$ with the same vertex set $V$. Then $G_1 \oplus G_2$ is the graph which also has vertex set $V$, and whose edge set is $E(G_1) \oplus E(G_2)$. That is, $G_1 \oplus G_2$ has every edge which is in exactly one of $G_1$ or $G_2$. Here is an example:

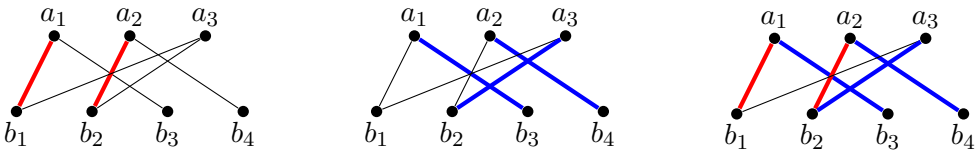



We can think of $G_1 \oplus G_2$ as a summary of the changes that need to be made to $G_1$ to get $G_2$. (If you've ever compared two versions of a Wikipedia article, you've seen a similar idea.) If we start with $G_1$, and "toggle" all the edges of $G_1 \oplus G_2$ (removing them from $G_1$ if they exist there, and adding them if they don't), then we get $G_2$.

## 2.2 Comparing two matchings

So let's use this idea to see what changes need to be made to get from one matching to another.

We are looking for small, incremental changes. So we will compare two matchings $M_1$ and $M_2$ such that $M_2$ only has one more edge than $M_1$. Let's look at the graph from the previous lecture again:




In this example, $M_1 \cap M_2 = \emptyset$, and so $M_1 \oplus M_2 = \{a_1b_1, a_1b_3, a_2b_2, a_2b_4, a_3b_2\}$ is the same as $M_1 \cup M_2$. To go from $M_1$ and $M_2$, we need to remove all the edges of $M_1$, and add all the edges of $M_2$.

The view as sets is not particularly helpful! So think about $M_1 \oplus M_2$ as a graph instead: the subgraph of all the edges that are red or blue in the third diagram.

As a graph, $M_1 \oplus M_2$ has two connected components: $\{a_1, b_1, b_3\}$ and $\{a_2, a_3, b_2, b_4\}$.

- The component on vertices $\{a_1, b_1, b_3\}$ is just a path $(b_1, a_1, b_3)$ with one blue edge and one red edge. This is not an important component: it's telling us that $M_2$ swaps edge $a_1b_1$ for edge $a_1b_3$, but this doesn't really help make $M_2$ any large.
- The component on vertices $\{a_2, a_3, b_2, b_4\}$ is a path $(a_3, b_2, a_2, b_4)$ with **two** blue edges and one red edge. This path is what's helping $M_2$ become bigger! By toggling all the edges of this path, we go from $M_1$ to a matching with 3 edges.

### 2.3 Augmenting paths

In general, if we have a matching $M$ in a graph $G$, an **$M$-augmenting path** is a path in $G$ with the following properties:

- The path begins and ends at two vertices that are not covered by $M$.
- The edges of the path alternate between "edge not in $M$" and "edge in $M$".

In order for both of these to hold, the path needs to have odd length. If it has length $2k+1$, then the $1^{\text{st}}, 3^{\text{rd}}, \ldots, (2k+1)^{\text{th}}$ edges are not part of $M$, while the $2^{\text{nd}}, 4^{\text{th}}, \ldots, (2k)^{\text{th}}$ edges are.

Suppose $M$ is a matching and $P$ is an $M$-augmenting path. Then, thinking of both $M$ and $P$ as subgraphs, $M \oplus P$ is a matching with one more edge than $M$. To prove this, we check that all vertices of $G$ have degree at most 1 in $M \oplus P$:

- The first and last vertices of $P$ were uncovered by $M$: they had degree 0 in $M$. When we go from $M$ to $M \oplus P$, they gain an edge, so now they have degree 1.
- All other intermediate vertices of $P$ both gain and lose an edge when we go from $M$ to $M \oplus P$. They had degree 1 in $M$, and they still have degree 1 in $M \oplus P$.
- All vertices that are not on $P$ are unaffected by the change.

It can be shown that all matching in all graphs are either maximum, or can be improved by some augmenting path. We will not prove this today, but our proof of König's theorem will imply this fact for bipartite graphs.

## 3 Proof of König's theorem

### 3.1 The augmenting path algorithm

Let $G$ be a bipartite graph with bipartition $(A, B)$, and let $M$ be a matching in $G$.

We'll further break down $A$ into $A_0 \cup A_1$: let $A_0$ be the set of vertices of $A$ that are uncovered by $M$ (have degree 0 in $M$) and let $A_1$ be the set of vertices of $A$ that are covered by $M$ (have degree 1 in $M$). Similarly, break down $B$ into $B_0 \cup B_1$.

To search for an augmenting path in $G$, we start exploring from all the vertices of $A_0$. We will not go into the details that computer scientists would care about: they would think about stacks and queues and efficient data structures, and we will not.

To keep track of our progress, let $X_A$ be the set of all **ex**plored vertices in $A$, and let $X_B$ be the set of all **ex**plored vertices in $B$. Initially, $X_A = A_0$, and $X_B = \varnothing$.

We repeat the following steps (illustrated in examples on the next page):

1. **Explore!** From each vertex in $X_A$, we go to all its neighbors and add them to $X_B$ (if they are not already there).
2. **Win?** If there is a vertex in $X_B \cap B_0$ (a vertex in $X_B$ not covered by $M$), stop immediately; we will have found an augmenting path.
3. **Match!** Otherwise, $X_B \subseteq B_1$: all the vertices in $X_B$ are covered by $M$. Take the vertices they're matched to, and add them to $X_A$ (they will not already be there).

We stop whenever we no longer make any progress: there are no new vertices to explore in the **explore** step.

Whenever we add a vertex $a$ to $X_A$, either $a \in A_0$ or we reached it from some $b \in X_B$ by an edge $ab \in M$ in a **match** step. Whenever we add a vertex $b$ to $X_B$, we reached it from some $a \in X_A$ in an **explore** step—and $ab$ cannot be in $M$, since if $a$ is covered at all, it is covered by the edge of $M$ that brought us to $A$.

Therefore for every vertex we explore, we reach it by a path that starts in $A_0$ in which every other edge lies in $M$. This explains how we find an augmenting path in the **win** step, if we ever see a vertex $b \in X_B \cap B_0$. The path that leads to $b$ starts in $A_0$, ends in $B_0$, and every other edge is in $M$: it is an augmenting path!

### 3.2 The vertex cover

To prove König's theorem, we need to see what happens if we stop without finding an augmenting path. In this case, let $U = (A_1 - X_A) \cup X_B$. Two claims finish the proof of Theorem 1.3:

**Claim 3.1.** *When the algorithm stops, $U$ is a vertex cover.*

*Proof.* Take any edge $ab \in E(G)$, where $a \in A$ and $b \in B$.

- If $a \in A_0$, then we added $b$ to $X_B$ in the very first **explore** step, so $b \in U$.
- If $a \in A_1 - X_A$, then $a \in U$.
- If $a \in A_1 \cap X_A$, then either we reached $a$ from $b$ when we **matched**, or else we reached $b$ from $a$ in the next step when we **explored**. Either way, $b \in X_B$, so $b \in U$.

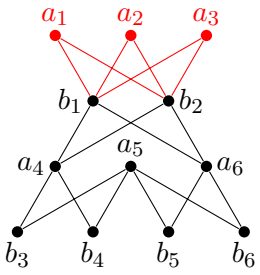
The three cases are exhaustive, and in all of them, $ab$ is covered by $U$. □

**Claim 3.2.** *If we never **win**, then after every **match** step (and in particular, when the algorithm stops) $|U| = |M|$.*

*Proof.* Initially, $|U| = |A_1| = |M|$ (because each edge of $M$ has one endpoint in $A$). If an **explore** step adds $k$ vertices to $X_B$, then $|X_B|$ (and $|U|$) increases by $k$. But if we did not **win**, then all $k$ vertices are in $B_1$, so we will add the $k$ vertices they're matched to $X_A$. This makes $|A_1 - X_A|$ (and $|U|$) decrease by $k$, and we're back to $|U| = |M|$. □

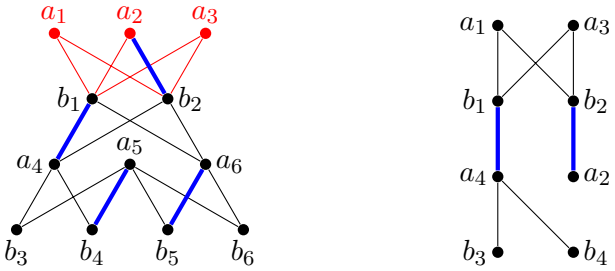
### 3.3 Diagrams of the algorithm

The graph below is a graph I decided to call the **volcano graph**;[2] we will use it as an example on which to test out our algorithm for finding a perfect matching.




This graph is bipartite, with bipartition $A = \{a_1, a_2, a_3, a_4, a_5, a_6\}$ and $B = \{b_1, b_2, b_3, b_4, b_5, b_6\}$, though it is not drawn in a way that makes this maximally obvious. It also has a vertex cover of size 5 which you might be able to spot from the diagram, though we'll also find it in the process of finding the maximum matching.

When starting from an empty matching, the first few steps of the algorithm tend to be boring: we will find an edge of the matching immediately. So we'll skip ahead to a step where we've found a maximal matching to which we can't simply add another edge. The matching we will start with is the matching $M = \{a_1b_2, a_4b_1, a_5b_4, a_6b_5\}$, shown below on the left.



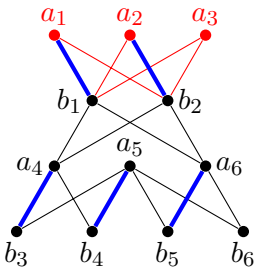

In the diagram on the right, the exploration process is unfolded, with each row corresponding to a separate step.

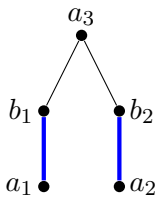
1. We begin by setting $X_A = \{a_1, a_3\}$: this is $A_0$, the two vertices of $A$ not covered by the matching.
2. We **explore** to add $b_1$ and $b_2$ to $X_B$.
3. Both $b_1$ and $b_2$ are in $B_1$: they are covered by the matching. So we don't **win**.
4. We **match** to add $a_2$ (because $a_2b_2 \in M$) and $a_4$ (because $a_4b_1 \in M$) to $X_A$.
5. We **explore** to add $b_3$ and $b_4$ to $X_B$ (both from $a_4$).
6. We realize that $b_3 \in B_0$: it is not covered by the matching! So we **win**.

To find an augmenting path, we look at the graph we've drawn on the right, and find a path from the top to $b_3$, the vertex of $B_0$ we've found.

[2] The red edges are mandatory but purely decorative.

One such path is $(a_1, b_1, a_4, b_3)$, so that's the path we will use to augment our matching. In the diagram below on the left, the new matching $M' = \{a_1b_1, a_2b_2, a_4b_3, a_5b_4, a_6b_5\}$ is highlighted.




The diagram above on the right shows our exploration process, as before, but now it is much shorter.

1. We begin by setting $X_A = \{a_3\}$: this is the only vertex not covered by the matching.
2. We **explore** to add $b_1$ and $b_2$ to $X_B$.
3. Both $b_1$ and $b_2$ are in $B_1$: they are covered by the matching. So we don't **win**.
4. We **match** to add $a_1$ (because $a_1b_1 \in M'$) and $a_2$ (because $a_2b_2 \in M'$) to $X_A$.
5. We try to **explore**, but the only neighbors of $a_1$ and $a_2$ are the already-visited $b_1$ and $b_2$. So we stop, and conclude that we've found a maximum matching.

The matching we've found is obvious: it is $M'$. But what is the optimal vertex cover?

The rule is that we take $U = (A_1 - X_A) \cup X_B$. Here:

- $A_1 = \{a_1, a_2, a_4, a_5, a_6\}$: all the vertices covered by our matching.
- But we've explored $X_A = \{a_1, a_2, a_3\}$. So $A_1 - X_A = \{a_4, a_5, a_6\}$ will be the set of vertices from $A$ in our vertex cover.
- $X_B = \{b_1, b_2\}$: the two vertices from $B$ we've explored. These are the vertices from $B$ in our vertex cover.

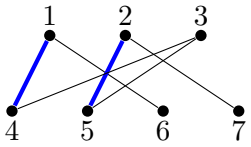
The result is $U = \{a_4, a_5, a_6, b_1, b_2\}$. Looking at the volcano graph, you can tell that this is a vertex cover, because it consists of the middle two "layers" of the volcano. The top layer (the "sparks" of the volcano) only has edges to the second layer, and the bottom layer (the "base" of the volcano) only has edges to the third, so $U$ is indeed a vertex cover.

## 4 Practice problems

1. In our proof of König's theorem, we've assumed throughout that we have a bipartite graph, but a lot of the things we've done could be imitated in any graph.

   Where do we *need* $G$ to be bipartite: what goes wrong when $G$ is not a bipartite graph that means we don't actually prove König's theorem?

2. Take this smaller example from our notes, and use our algorithm to find an augmenting path. There will be two choices; one is the augmenting path we already found for this matching. What is the other?




$M_1 = \{14, 25\}$

3. In preparation for the next lecture, let's think about a few more properties of symmetric differences of matchings

   (a) Let $M_1$ and $M_2$ be two matchings in the same graph $G$. Prove that $M_1 \oplus M_2$ is a subgraph of $G$ with maximum degree 2.

   (b) Describe *all* the connected graphs with maximum degree 2.

   (c) Every connected component of $M_1 \oplus M_2$ should be one of the graphs you listed in part (b). However, there are infinitely many cases of (b) that *cannot* be connected components of $M_1 \oplus M_2$: what are they?

4. Suppose we have a bipartite graph $G$ with bipartition $(A, B)$ which is $r$-regular: every vertex has degree $r$. Let $n = |A| = |B|$ (we have already shown in the past that when $G$ is regular, we must have $|A| = |B|$.)

   (a) How many edges does $G$ have?

   (b) Prove that a set of $k$ vertices in $G$ can cover at most $kr$ of the edges. What does this say about the size of a vertex cover?

   (c) Use König's theorem to prove that $G$ must have a perfect matching.

5. Suppose we have a bipartite graph $G$ with bipartition $(A, B)$ and minimum degree $\delta(G)$.

   (a) It is always true that the set $A$ is a vertex cover, and so is the set $B$. Suppose $U$ is a vertex cover that **does not** contain either $A$ or $B$. (There is some vertex $a \in A$ such that $a \notin U$, and there is some vertex $b \in B$ such that $b \notin U$.)

   Prove that in this case, $|U| \geq 2\delta(G)$.

   (b) When does it follow that $\alpha'(G) \geq 2\delta(G)$, and why?

   (c) In the case $|A| = |B| = 10$ and $\delta(G) = 3$, give an example of a bipartite graph in which this lower bound is true and $\alpha'(G) = 6$.