Combinatorial Mathematics

Mong-Jen Kao (高孟駿)

Monday 18:30 – 21:20

Outline

- The Maximum Matching Problem
  - A Generic Algorithm and the Berge's Theorem
  - Solving the Augmenting Path Problem
    - DFS-based & BFS-based Algorithms for Bipartite Graphs
    - The Blossom Algorithm for General Graphs
- Concluding Notes
  - The best algorithms for Maximum Matching

Characterization of Bipartite Graphs

Identify the two partite sets of a bipartite graph when it is not given.

Characterization of Bipartite Graphs

- The following theorem is simple and intuitive to prove.

> <u>**Theorem. (Characterization of Bipartite Graphs)**</u>
>
> A graph $G = (V, E)$ is bipartite if and only if it has a 2-coloring, i.e., a 2-coloring for V such that no edge $e \in E$ is monochromatic.

- Note that, the 2-colorability of G can be tested by a simple DFS.
  - If G has a 2-coloring, then it also corresponds to a valid classification of the two partite sets.

You will need this fact in ProgHW #1.

An Alternative BFS-based Algorithm

An Alternative Algorithm

- Let X_0 be the set of all unmatched vertices in G.
- For any $i = 0, 1, 2, ...$, define
  - X_{2i+1} to be the set of *unvisited* vertices (not in $X_{\leq 2i}$) that can be reached from X_{2i} ***using an edge not in M***.
  - X_{2i+2} to be the set of *unvisited* vertices (not in $X_{\leq 2i+1}$) that can be reached from X_{2i+1} **using an edge in M**.

An Alternative Algorithm

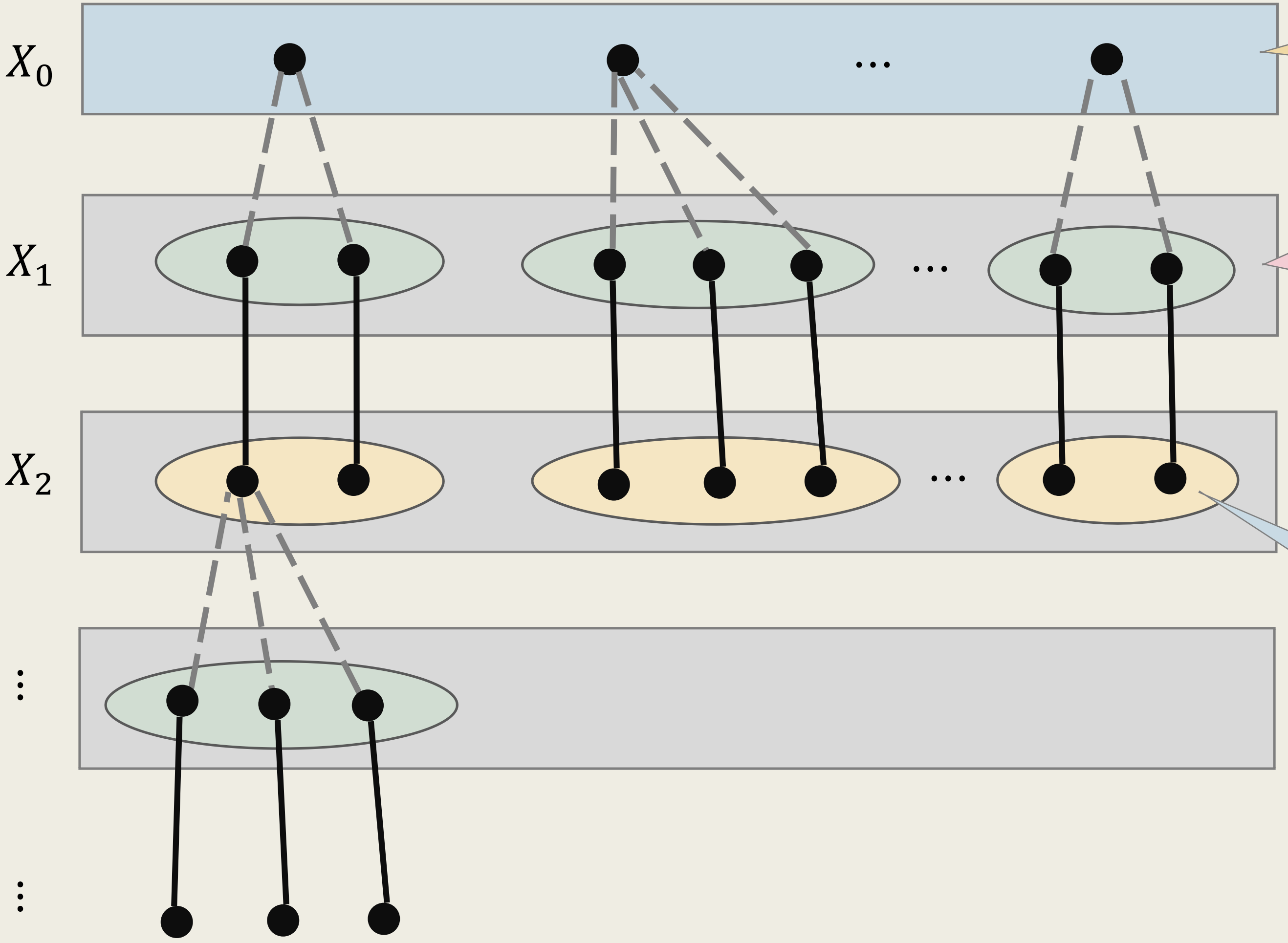
- Let X_0 be the set of all unmatched vertices in G.
- Formally, for any $i = 0, 1, 2, \ldots$, define

$$X_{2i+1} \coloneqq \{ \ v \in V \setminus X_{\leq 2i} \ : \ \exists u \in X_{2i} \ s.t. \ (u, v) \notin M \ \}$$

and

$$X_{2i+2} \coloneqq \{ \ v \in V \setminus X_{\leq 2i+1} \ : \ \exists u \in X_{2i+1} \ s.t. \ (u, v) \in M \ \}.$$

The ***Alternating Forest*** Formed by X_i



The set of unmatched vertices

The vertices in the ***odd levels*** have ***exactly one children*** in the forest.

Each non-root vertex has ***exactly one parent*** in the forest.

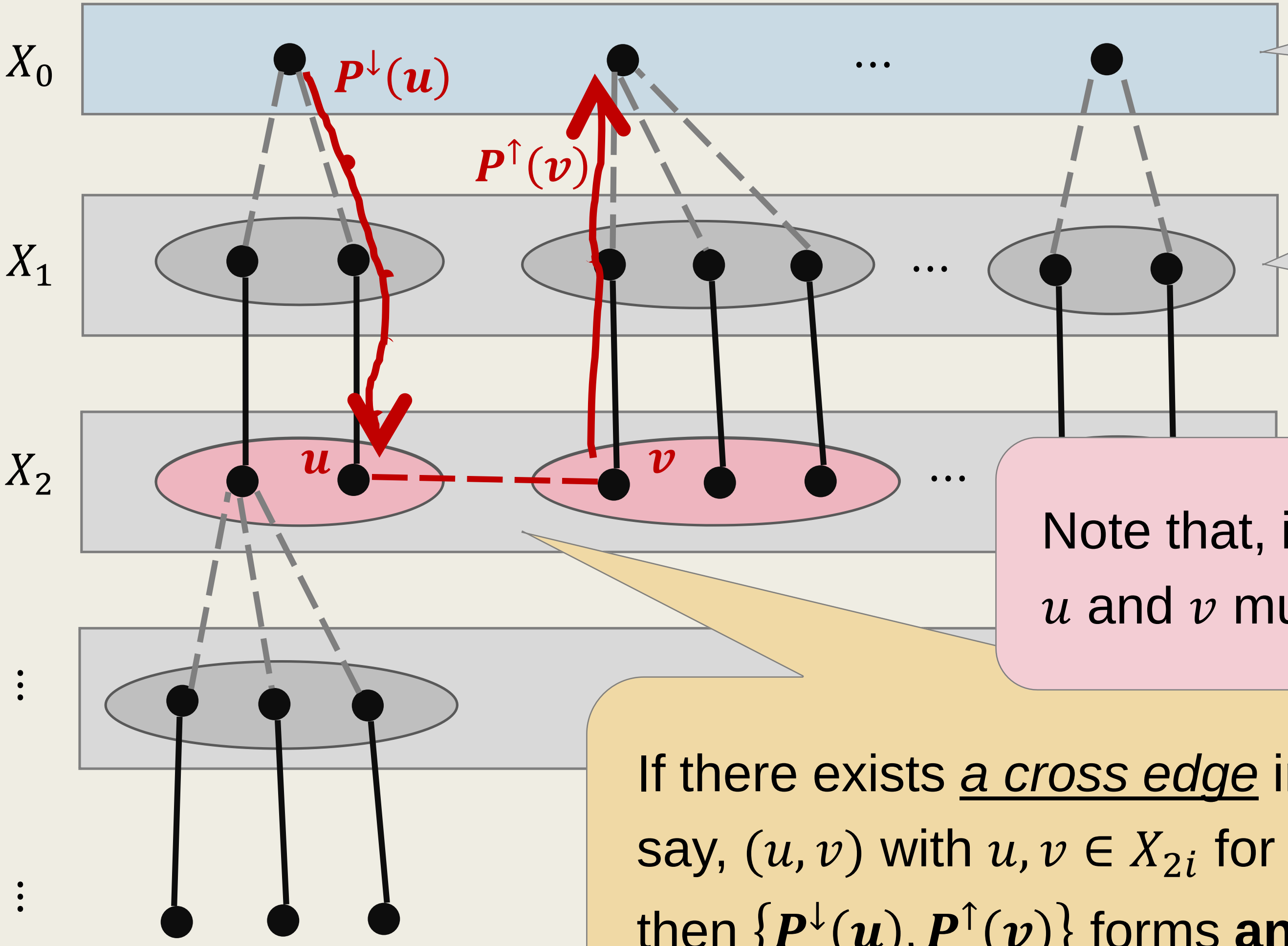
The ***Alternating Forest*** Formed by X_i

- The roots are the unmatched vertices in X_0.
  - Each non-root vertex has exactly one parent in the forest.

- For any vertex $v \in V$,
  - Let $P^{\uparrow}(v)$ be the path from v to its root in the forest.
  - Also, let $P^{\downarrow}(v)$ be the path from its root to v in the forest.

- Note that, $P^{\uparrow}(v)$ and $P^{\downarrow}(v)$ are uniquely defined, and they are M-alternating paths.

The ***Alternating Forest*** Formed by X_i



The set of unmatched vertices

The vertices in the ***odd levels*** have ***exactly one children*** in the forest.

Note that, in this case,

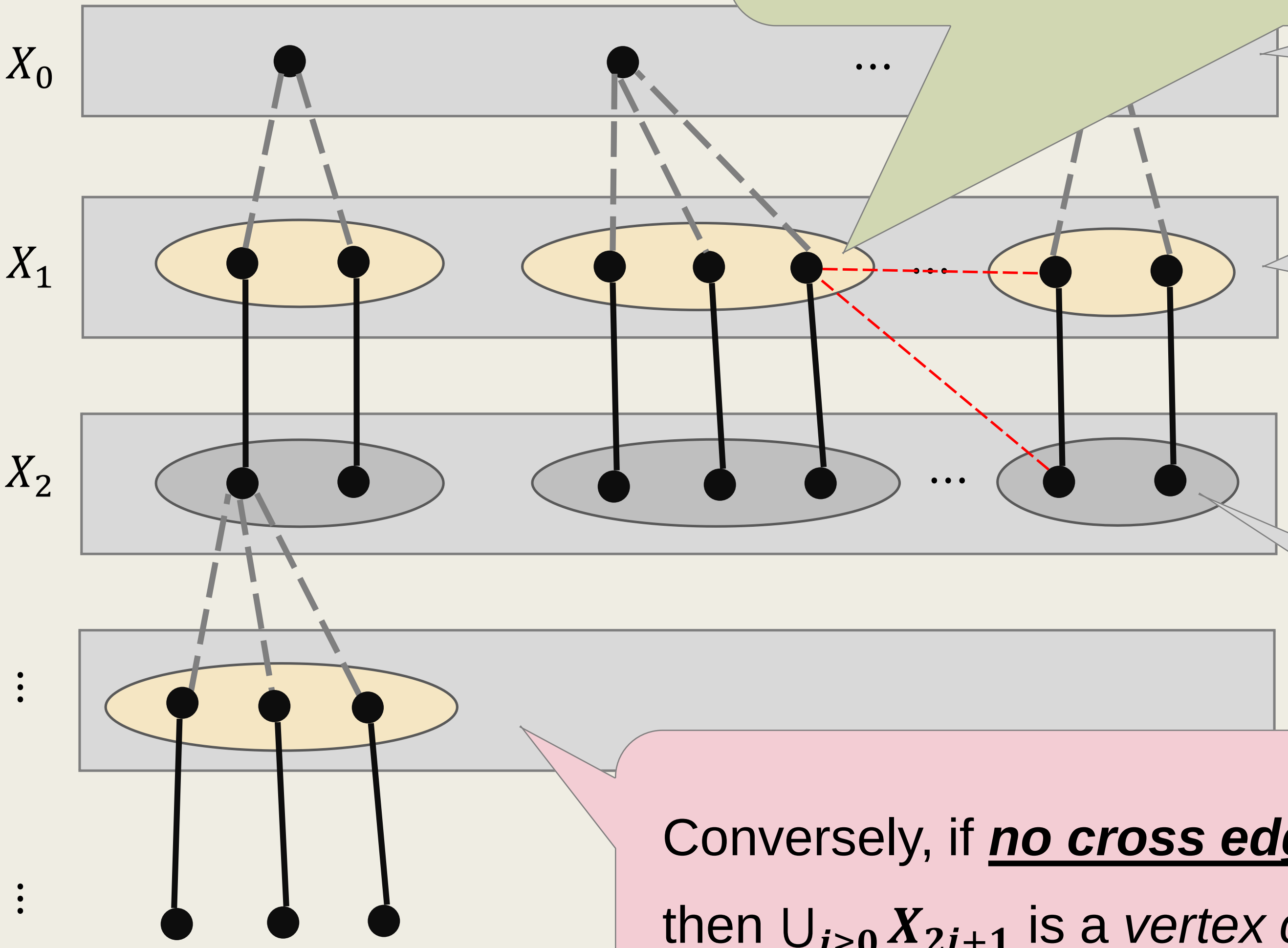
u and v must be in different trees. (Why?)

If there exists *a cross edge* in *the even levels*,

say, (u, v) with $u, v \in X_{2i}$ for some i,

then $\{\boldsymbol{P}^{\downarrow}(\boldsymbol{u}), \boldsymbol{P}^{\uparrow}(\boldsymbol{v})\}$ forms **an $\boldsymbol{M}$-augmenting path**!

The *Alternating Fore*



Note that, there may still be edges between a vertex in the odd level and other vertices, but we don’t care.

The vertices in the ***odd levels*** have ***exactly one children*** in the forest.

Conversely, if ***<u>no cross edge</u>*** exists ***<u>in the even levels</u>***, then $\bigcup_{i\geq 0} \boldsymbol{X_{2i+1}}$ is a *<u>vertex cover</u>* with size $|M|$!

- Let $G = (V, E)$ be a bipartite graph and M be a matching for G.

Another BFS-based Augmenting Path Algorithm (for Bipartite Graphs).

1. Let X_0 be the set of unmatched vertices and $t \leftarrow 0$.
2. Repeat until $X_{\leq 2t} = V$, do
   - If there exists an edge $(u, v) \in E$ for some $u, v \in X_{2t}$,
     then return the path $\{P^{\downarrow}(u), P^{\uparrow}(v)\}$.
   - Otherwise,
     form X_{2t+1} and X_{2t+2} as described and set $t \leftarrow t + 1$.
3. Report $\bigcup_{i\geq 0} X_{2i+1}$ as a *vertex cover* with size $|M|$.

The Augmenting Path Problem in General Graphs

For general graphs,
the augmenting path problem can be solved in $O(nm)$ time *via proper vertex contractions*.

The Augmenting Path Problem in General Graphs

- Let $G = (V, E)$ be a general graph and $\boldsymbol{M}$ be a matching for G.

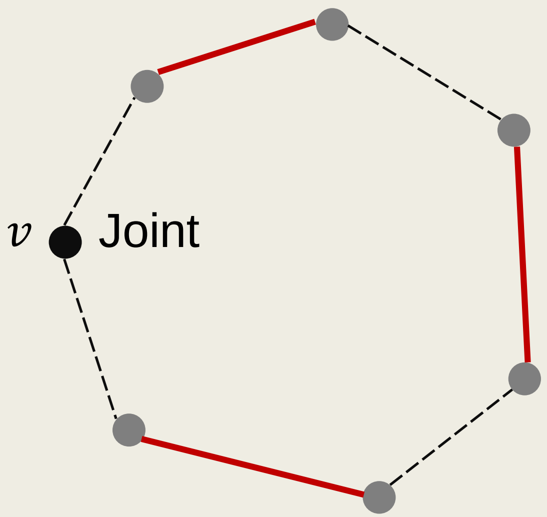
- We introduce an algorithm that computes in $O(nm)$ time either
  - An M-augmenting path for G, or,
  - A *structure* ***(proof)*** *showing that* ***M is maximum***.

    Hence, no M-augmenting path exists in the graph.

Note that, we can *no longer* count on *vertex covers* for this, since the ***strong duality does not hold*** between *matchings and vertex covers* in *general graphs*.

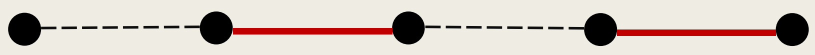
Blossom, Stem, and Flowers

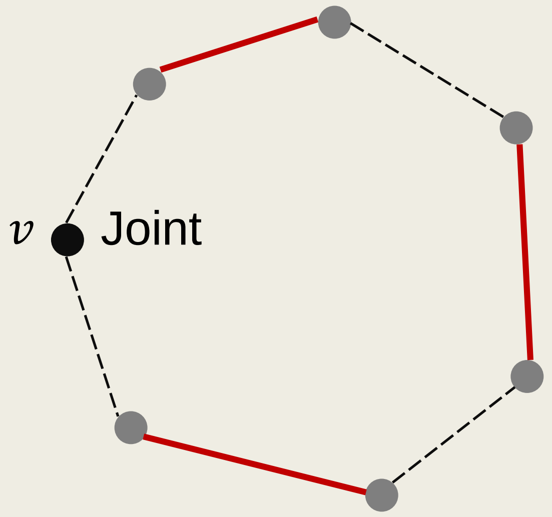
- A ***blossom*** is a cycle C with *an odd length* and $\lfloor |C|/2 \rfloor$ *matched edges in M*.
  - The vertex $v \in C$ that is not incident to any matched edge is called the "***joint***" of the blossom.



Blossom, Stem, and Flowers

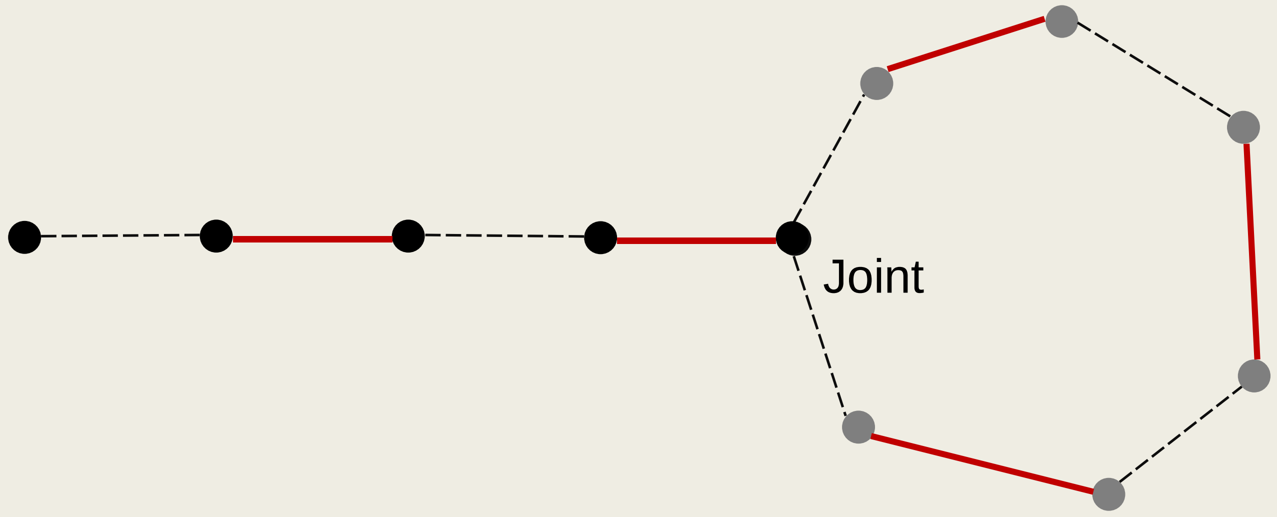
- A ***stem*** is an M-alternating path with *an even length* and *ends at a matched edge in* M.





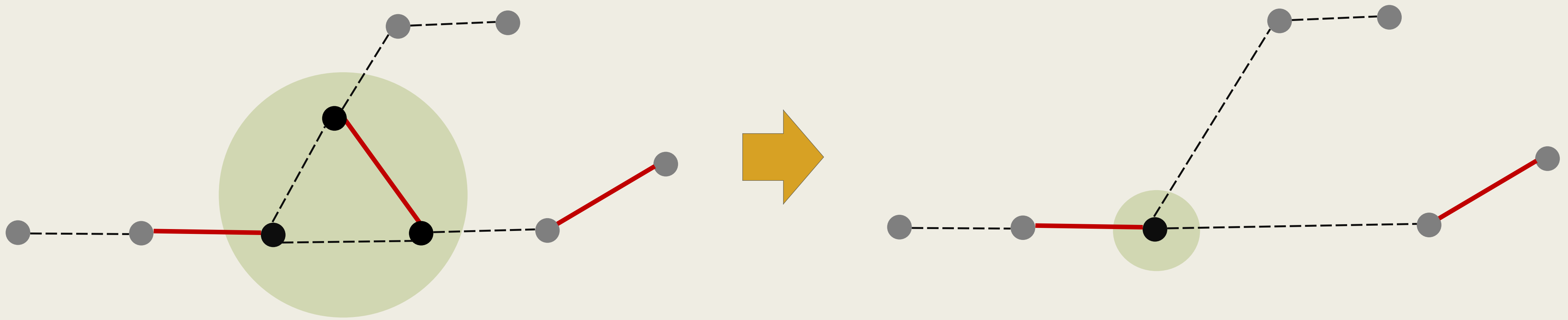
Blossom, Stem, and Flowers

- A ***flower*** is *a stem* and *a blossom* such that the stem ends at the joint of the blossom.



Contracting a Blossom

- Let C be a blossom in G.
  - Define G_C to be the graph obtained by contracting C in G, and M_C' be the remaining set of matched edges.



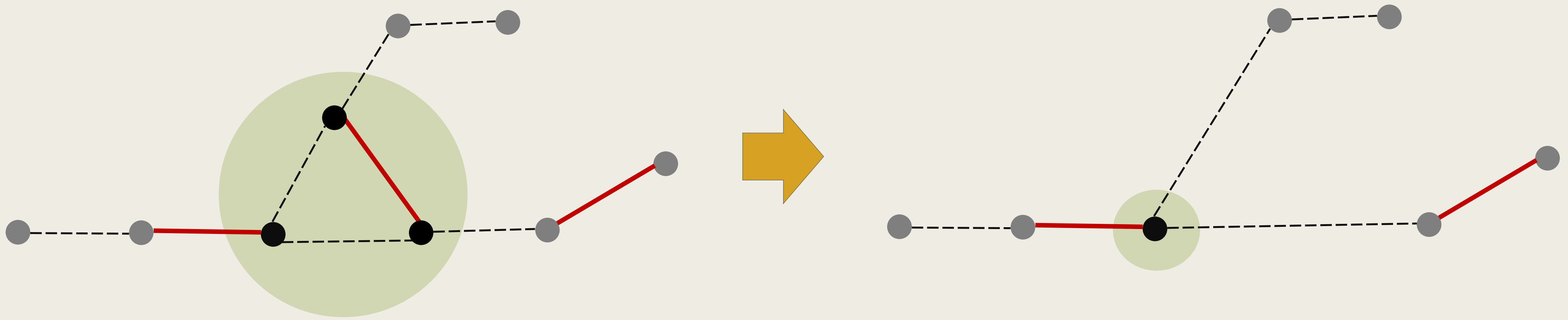
The blossom C

The contracted graph G_C

- Let C be a blossom in G.
  - Define G_C to be the graph obtained by contracting C in G, and M'_C be the remaining set of matched edges.

Lemma. (Blossom Contraction)

G has an M-augmenting path if and only if G_C has an M'_C-augmenting path.



The blossom C

The contracted graph G_C

The Blossom Algorithm (by Jack Edmonds)

- Let X_0 be the set of all unmatched vertices in G.
- For any $i = 0, 1, 2, \ldots$, define
  - X_{2i+1} to be the set of *unvisited* vertices (not in $X_{\leq 2i}$) that can be reached from X_{2i} ***using an edge not in M***.
  - X_{2i+2} to be the set of *unvisited* vertices (not in $X_{\leq 2i+1}$) that can be reached from X_{2i+1} **using an edge in M**.

The Blossom Algorithm (by Jack Edmonds)

- Consider the alternating forest formed by X_i for all $i \geq 0$.

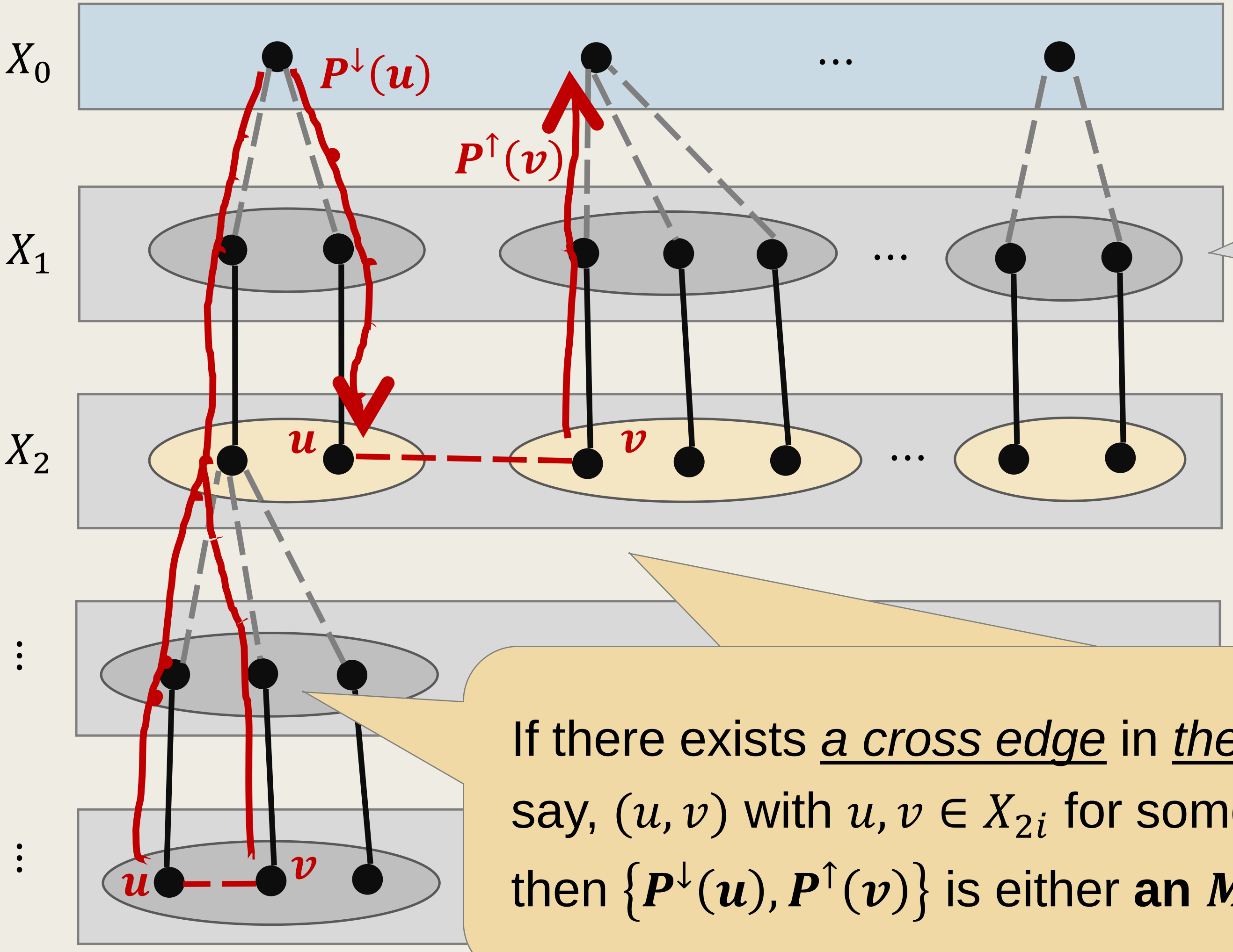
- If there exists ***a cross edge in an even level***,
  i.e., $(u, v) \in E$ for some $u, v \in X_{2i}$ and some $i \geq 0$,

  then $\{P^{\downarrow}(u), P^{\uparrow}(v)\}$ is either *an M-augmenting path* or *a flower*!

  – If $P^{\downarrow}(u) \cap P^{\uparrow}(v) = \emptyset$, then it is an augmenting path.

  – Otherwise,
    it is a flower with the common part being the stem.

The ***Alternating Forest*** Formed by X_i



The vertices in the ***odd levels*** have ***exactly one children*** in the forest.

If there exists a cross edge in the even levels, say, (u, v) with $u, v \in X_{2i}$ for some i, then $\{\boldsymbol{P}^{\downarrow}(\boldsymbol{u}), \boldsymbol{P}^{\uparrow}(\boldsymbol{v})\}$ is either **an $\boldsymbol{M}$-augmenting path or a flower**!

- Let $G = (V, E)$ be a graph and M be a matching for G.

<u>The Blossom Algorithm</u> (by Jack Edmonds).

1. Let X_0 be the set of unmatched vertices and $t \leftarrow 0$.
2. Repeat until $X_{\leq 2t} = V$, do
   - If there exists an edge $(u, v) \in E$ for some $u, v \in X_{2t}$,
     - If $P^{\downarrow}(u) \cap P^{\uparrow}(v) = \emptyset$, then return the path $\{P^{\downarrow}(u), P^{\uparrow}(v)\}$.
     - Otherwise, let $C \leftarrow P^{\downarrow}(u) \,\Delta\, P^{\uparrow}(v)$. Apply the algorithm recursively on G_C and M'_C. <u>*Expand the result*</u> and return it.
   - Otherwise,
     form X_{2t+1} and X_{2t+2} as described and set $t \leftarrow t + 1$.
3. Report $\bigcup_{i \geq 0} X_{2i+1}$ as a <u>***proof***</u>.

The Correctness of

the Blossom Algorithm

Analysis of the Algorithm

- For the correctness of the algorithm,
  - It is clear that, when the blossom algorithm returns an M-augmenting path, it is indeed a valid one.
  - We need to show that, when the algorithm returns a proof (reports "No"), M is indeed a maximum matching.

    For this, we will use the Tutte-Berge Max-Min Theorem.

Lemma. (Tutte-Berge Max-Min Theorem)

Let $G = (V, E)$ be a graph,

$U \subseteq V$ be a vertex subset, and $M \subseteq E$ be a matching.

Then we always have

$$|M| \leq \frac{|V| + |U| - \mathrm{odd}(G \setminus U)}{2},$$

where $\mathrm{odd}(G \setminus U)$ is the number of components with an odd size in $G \setminus U$.

- Later we will see that, the inequality holds with equality for properly chosen M and U when G contains no blossom.

Let $G = (V, E)$ be a graph, $U \subseteq V$ be a vertex subset, and $M \subseteq E$ be a matching. Then we always have

$$|M| \le \frac{|V| + |U| - \mathrm{odd}(G \setminus U)}{2},$$

where $\mathrm{odd}(G \setminus U)$ is the number of odd components in $G \setminus U$.

- Consider the components in U and $G \setminus U$.
  - Each vertex in U is incident with at most one edge in M.
  - For the remaining components K in $G \setminus U$, it contains at most $\left\lfloor \frac{|K|}{2} \right\rfloor$ edges in M.

Since all endpoints of the edges in M are distinct.

We always have $|M| \leq \big(|V| + |U| - \mathrm{odd}(G \setminus U)\big)/2$,

where $\mathrm{odd}(G \setminus U)$ is the number of odd components in $G \setminus U$.

- Consider the components in U and $G \setminus U$.
  - Each vertex in U is incident with at most one edge in M.
  - For the remaining components K in $G \setminus U$, it contains at most $\left\lfloor \frac{|K|}{2} \right\rfloor$ edges in M.

Since all endpoints of the edges in M are distinct.

- Hence,

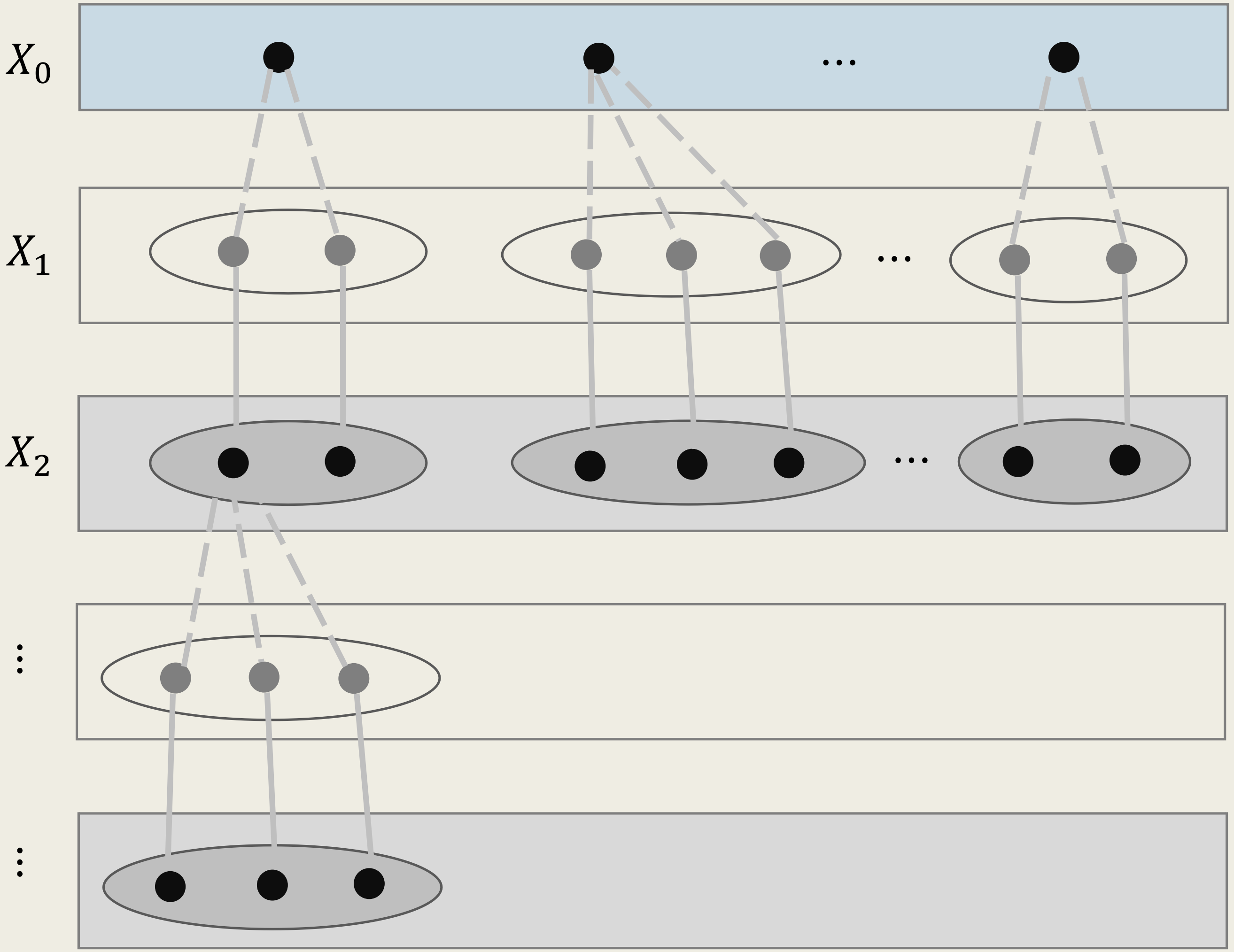
$$|M| \leq |U| + \sum_i \left\lfloor \frac{|K_i|}{2} \right\rfloor = |U| + \frac{|V| - |U|}{2} - \frac{\mathrm{odd}(G \setminus U)}{2}.$$

Analysis of the Algorithm

- Suppose that the Blossom algorithm returns a proof $\bigcup_{i \geq 0} X_{2i+1}$.
- By the Tutte-Berge's inequality, to prove that M is a maximum matching for G, it suffices to show that

> The choice of $U \coloneqq \bigcup_{i \geq 0} X_{2i+1}$ will make
> the Tutte-Berge's inequality hold with equality.

The ***Alternating Forest*** Formed by X_i



There is *no cross edge* in the even levels.

Hence, the vertices in *the even levels* become **isolated**.

The *remaining components* (not included in the forest) are *perfectly matched* by M.

Analysis of the Algorithm

- Let $U \coloneqq \bigcup_{i \geq 0} X_{2i+1}$.
- Then,

$$|M| = |U| + \frac{|V \setminus X_{\geq 0}|}{2} = \frac{|V| + |U| - \left|\bigcup_{i \geq 0} X_{2i}\right|}{2}$$

$$= \frac{|V| + |U| - \mathrm{odd}(G \setminus U)}{2}.$$

- Hence, M is a maximum matching for G.

Concluding Notes

Best Algorithm for the Maximum Bipartite Matching

- In this lecture,
  we have seen an $O(nm) = O(n^3)$ algorithm for this problem.

- The best algorithm for this problem is the Hopcroft-Karp algorithm, which runs in $O\left(\sqrt{n}m\right) = O\left(n^{2.5}\right)$.

The Hopcroft-Karp Algorithm

- The idea is to perform a **BFS** simultaneously from all unmatched vertices in one partite set **to form alternating layers** until some unmatched vertices in the other partite set is met.
- Then a **layer-guided DFS** is used to construct a maximal set of vertex-disjoint shortest augmenting paths.
- It is guaranteed that, only $O\left(\sqrt{n}\right)$ rounds are needed before the maximum matching is computed.

Best Algorithm for the Maximum Bipartite Matching

- This problem is a special case of the max-flow problem.

  A number of flow algorithms are applicable.

  - Practically,
    the most efficient one is the Dinic’s algorithm.

  - Theoretically, the best algorithm is the “Almost linear-time” max-flow algorithm that runs in $m^{1+o(1)}$ time.

Maximum Matching in General Graphs

- For general graphs, we have seen the Edmonds Blossom Algorithm, which runs in $O(n^2 m) = O(n^4)$ time.
- The best (and more complicated) algorithm, due to Micali and Vazirani, solves this problem in $O(\sqrt{n}m) = O(n^{2.5})$ time.