Lecture 18: Max-Flow & Min-Cut
Applications

Rafael Oliveira

University of Waterloo
Cheriton School of Computer Science
rafael.oliveira.teaching@gmail.com

November 14, 2023
Overview

- Applications of Max-Flow & Min-Cut
  - Maximum Bipartite Matching
  - Minimum Vertex Cover
  - Edge-disjoint Paths
  - Vertex-disjoint Paths

- Further Remarks

- Acknowledgements
Matchings

- Given an undirected graph $G(V, E)$ a matching $M$ is a subset of $E$ such that all edges in $M$ are pairwise vertex disjoint (i.e., no two edges share a common vertex).
- A matching $M \subset E$ is called a perfect matching if every vertex in the graph is matched.
Maximum Bipartite Matching

- **Input:** A bipartite graph $G(L \sqcup R, E)$
- **Output:** A maximum cardinality matching $M \subseteq E$
Maximum Bipartite Matching

- **Input:** A bipartite graph \( G(L \sqcup R, E) \)
- **Output:** A maximum cardinality matching \( M \subset E \)
- Consider directed graph \( H(\{s, t\} \sqcup L \sqcup R, F, c) \) given by

\[
\begin{cases}
\{u, v\} \in E, \ u \in L, \ v \in R \iff (u, v) \in F, \ c(u, v) = \infty \\
(s, u) \in F, \ c(s, u) = 1 \ \forall \ u \in L \\
(v, t) \in F, \ c(v, t) = 1 \ \forall \ v \in R
\end{cases}
\]

in picture:
Maximum Bipartite Matching

- **Input:** A bipartite graph \( G(L \sqcup R, E) \)
- **Output:** A maximum cardinality matching \( M \subseteq E \)
- Consider directed graph \( H(\{s, t\} \sqcup L \sqcup R, F, c) \) given by

\[
\begin{align*}
\{u, v\} \in E, \ u \in L, \ v \in R & \iff (u, v) \in F, \ c(u, v) = \infty \\
(s, u) \in F, \ c(s, u) = 1 & \forall u \in L \\
(v, t) \in F, \ c(v, t) = 1 & \forall v \in R
\end{align*}
\]

in picture:
- **Claim:** there is matching of size \( k \) in \( G \) \( \iff \) there is an \( s \rightarrow t \) flow of value \( k \) in \( H \)
Maximum Bipartite Matching

**Claim:** there is matching of size $k$ in $G \iff$ there is an $s \rightarrow t$ flow of value $k$ in $H$

- ($\Rightarrow$) from matching $M = \{\{u_i, v_i\}\}_{i=1}^k$ we get flow $f(s, u_i) = f(u_i, v_i) = f(v_i, t) = 1$ of value $k$
Maximum Bipartite Matching

- **Claim:** there is matching of size $k$ in $G \iff$ there is an $s \to t$ flow of value $k$ in $H$

  - $(\iff)$ from (integral) flow of value $k$ (exists by Ford-Fulkerson), use flow decomposition lemma (note that $H$ is a DAG) to get $k$ $s \to t$ paths $P_1, \ldots, P_k$, where
    
    $$P_i = (s, u_i, v_i, t)$$

    Path decomposition lemma says that $(s, u_i)$'s and $(v_i, t)$'s must be distinct, since
    
    $$0 < f(s, u_i) \leq c(s, u_i) = 1 \Rightarrow f(s, u_i) = 1$$
    
    (same for $(v_i, t)$).

    Moreover, $\{u_i, v_i\} \in E$ for $i \in [k]$, by construction of $H$.

    Thus, $M = \{\{u_i, v_i\}\}_{i=1}^k$ must be a matching in $G$.  

Maximum Bipartite Matching

- **Claim:** there is matching of size $k$ in $G \iff$ there is an $s \to t$ flow of value $k$ in $H$

- $(\Leftarrow)$ from (integral) flow of value $k$ (exists by Ford-Fulkerson), use flow decomposition lemma (note that $H$ is a DAG) to get $k$ $s \to t$ paths $P_1, \ldots, P_k$, where

  $$P_i = (s, u_i, v_i, t)$$

Path decomposition lemma says that $(s, u_i)$’s and $(v_i, t)$’s must be distinct, since

$$0 < f(s, u_i) \leq c(s, u_i) = 1 \Rightarrow f(s, u_i) = 1$$

(same for $(v_i, t)$).

Moreover, $\{u_i, v_i\} \in E$ for $i \in [k]$, by construction of $H$.

Thus, $M = \{\{u_i, v_i\}\}_{i=1}^{k}$ must be a matching in $G$.

- Ford-Fulkerson gives algorithm with running time $O(|V| \cdot |E|)$ for maximum bipartite matching.
Applications of Max-Flow & Min-Cut
- Maximum Bipartite Matching
- Minimum Vertex Cover
- Edge-disjoint Paths
- Vertex-disjoint Paths

Further Remarks

Acknowledgements
Minimum Vertex Cover

**Definition:** given graph \( G(V, E) \), a subset \( S \subseteq V \) is a vertex cover if for every edge \( \{u, v\} \in E \), we have \( \{u, v\} \cap S \neq \emptyset \)
Minimum Vertex Cover

- **Input:** Bipartite graph $G(L \sqcup R, E)$
- **Output:** Minimum cardinality vertex cover
Minimum Vertex Cover

- **Input:** Bipartite graph \( G(L \sqcup R, E) \)
- **Output:** Minimum cardinality vertex cover
- **König’s Theorem:**

**Theorem (König’s Theorem)**

*In a bipartite graph, the maximum size of a matching equals the minimum size of a vertex cover.*
Minimum Vertex Cover

- **Input**: Bipartite graph $G(L \sqcup R, E)$
- **Output**: Minimum cardinality vertex cover
- **König’s Theorem**: 

**Theorem (König’s Theorem)**

*In a bipartite graph, the maximum size of a matching equals the minimum size of a vertex cover.*

- Ford-Fulkerson finds a min-cut in the modified graph $H$ from the previous slides, and from it we will obtain a vertex cover. (we’ll see this in the next slide)
Proof of König’s theorem

- Let $G(L \uplus R, E)$ be our bipartite graph and $k$ be the maximum size of a matching in it.
- Let $H(\{s, t\} \uplus L \uplus R, F)$ be constructed as before. By our previous result, the max-flow in $H$ has value $k$. 
Proof of König’s theorem

- Let $G(L \sqcup R, E)$ be our bipartite graph and $k$ be the maximum size of a matching in it.
- Let $H(\{s, t\} \sqcup L \sqcup R, F)$ be constructed as before. By our previous result, the max-flow in $H$ has value $k$.
- By the max-flow min-cut theorem, let $S$ be an $s – t$ cut in $H$ with $s \in S \& \text{C}_{\text{out}}(S) = k$. (Ford-Fulkerson finds us such cut)
Proof of König’s theorem

- Let $G(L \sqcup R, E)$ be our bipartite graph and $k$ be the maximum size of a matching in it.
- Let $H(\{s, t\} \sqcup L \sqcup R, F)$ be constructed as before. By our previous result, the max-flow in $H$ has value $k$.
- By the max-flow min-cut theorem, let $S$ be an $s - t$ cut in $H$ with $s \in S$ & $\text{Cout}(S) = k$. (Ford-Fulkerson finds us such cut)
- **Claim 1**: $|(L \setminus S) \cup (S \cap R)| = k$
Proof of König’s theorem

- Let $G(L \sqcup R, E)$ be our bipartite graph and $k$ be the maximum size of a matching in it.
- Let $H(\{s, t\} \sqcup L \sqcup R, F)$ be constructed as before. By our previous result, the max-flow in $H$ has value $k$.
- By the max-flow min-cut theorem, let $S$ be an $s - t$ cut in $H$ with $s \in S$ & $C_{out}(S) = k$. (Ford-Fulkerson finds us such cut)
- **Claim 1:** $|(L \setminus S) \cup (S \cap R)| = k$
  - $s$ has edge of capacity 1 to each vertex in $L \setminus S$
  - $t$ has edge of capacity 1 from each vertex in $S \cap R$
Proof of König’s theorem

- Let \( G(L \sqcup R, E) \) be our bipartite graph and \( k \) be the maximum size of a matching in it.

- Let \( H(\{s, t\} \sqcup L \sqcup R, F) \) be constructed as before. By our previous result, the max-flow in \( H \) has value \( k \).

- By the max-flow min-cut theorem, let \( S \) be an \( s-t \) cut in \( H \) with \( s \in S \) & \( C_{\text{out}}(S) = k \). (Ford-Fulkerson finds us such cut)

- **Claim 1:** \(|(L \setminus S) \cup (S \cap R)| = k\)
  - \( s \) has edge of capacity 1 to each vertex in \( L \setminus S \)
  - \( t \) has edge of capacity 1 from each vertex in \( S \cap R \)
  - These edges are in \( \delta_{\text{out}}(S) \)
Proof of König’s theorem

- Let $G(L \sqcup R, E)$ be our bipartite graph and $k$ be the maximum size of a matching in it.
- Let $H(\{s, t\} \sqcup L \sqcup R, F)$ be constructed as before. By our previous result, the max-flow in $H$ has value $k$.
- By the max-flow min-cut theorem, let $S$ be an $s - t$ cut in $H$ with $s \in S$ & $C_{\text{out}}(S) = k$. (Ford-Fulkerson finds us such cut)

**Claim 1:** $|(L \setminus S) \cup (S \cap R)| = k$

- $s$ has edge of capacity 1 to each vertex in $L \setminus S$
- $t$ has edge of capacity 1 from each vertex in $S \cap R$
- These edges are in $\delta_{\text{out}}(S)$
- Note that $\delta_{\text{out}}(S)$ cannot contain edge from $L$ to $R$ (as these have $\infty$ capacity), so the edges above are the only ones in $\delta_{\text{out}}(S)$. 
Proof of König’s theorem

- Let $G(L \sqcup R, E)$ be our bipartite graph and $k$ be the maximum size of a matching in it.

- Let $H(\{s, t\} \sqcup L \sqcup R, F)$ be constructed as before. By our previous result, the max-flow in $H$ has value $k$.

- By the max-flow min-cut theorem, let $S$ be an $s-t$ cut in $H$ with $s \in S$ & $C_{out}(S) = k$. (Ford-Fulkerson finds us such cut)

  **Claim 1:** $|(L \setminus S) \cup (S \cap R)| = k$
  - $s$ has edge of capacity 1 to each vertex in $L \setminus S$
  - $t$ has edge of capacity 1 from each vertex in $S \cap R$
  - These edges are in $\delta_{out}(S)$
  - Note that $\delta_{out}(S)$ cannot contain edge from $L$ to $R$ (as these have $\infty$ capacity), so the edges above are the only ones in $\delta_{out}(S)$.

- **Claim 2:** $(L \setminus S) \cup (S \cap R)$ is a vertex cover of $G$
Proof of König’s theorem

- Let $G(L \sqcup R, E)$ be our bipartite graph and $k$ be the maximum size of a matching in it.
- Let $H(\{s, t\} \sqcup L \sqcup R, F)$ be constructed as before. By our previous result, the max-flow in $H$ has value $k$.
- By the max-flow min-cut theorem, let $S$ be an $s - t$ cut in $H$ with $s \in S$ & $C_{out}(S) = k$. (Ford-Fulkerson finds us such cut)
- **Claim 1:** $|(L \setminus S) \cup (S \cap R)| = k$
  - $s$ has edge of capacity 1 to each vertex in $L \setminus S$
  - $t$ has edge of capacity 1 from each vertex in $S \cap R$
  - These edges are in $\delta_{out}(S)$
  - Note that $\delta_{out}(S)$ cannot contain edge from $L$ to $R$ (as these have infinite capacity), so the edges above are the only ones in $\delta_{out}(S)$.
- **Claim 2:** $(L \setminus S) \cup (S \cap R)$ is a vertex cover of $G$
  - Note that $\delta_{out}(S)$ cannot contain edge from $L$ to $R$ (as these have infinite capacity).
Proof of König’s theorem

- Let \( G(L \sqcup R, E) \) be our bipartite graph and \( k \) be the maximum size of a matching in it.
- Let \( H(\{s, t\} \sqcup L \sqcup R, F) \) be constructed as before. By our previous result, the max-flow in \( H \) has value \( k \).
- By the max-flow min-cut theorem, let \( S \) be an \( s - t \) cut in \( H \) with \( s \in S \) & \( \text{Cut}(S) = k \). (Ford-Fulkerson finds us such cut)
- **Claim 1:** \( |(L \setminus S) \cup (S \cap R)| = k \)
  - \( s \) has edge of capacity 1 to each vertex in \( L \setminus S \)
  - \( t \) has edge of capacity 1 from each vertex in \( S \cap R \)
  - These edges are in \( \delta_{out}(S) \)
  - Note that \( \delta_{out}(S) \) cannot contain edge from \( L \) to \( R \) (as these have \( \infty \) capacity), so the edges above are the only ones in \( \delta_{out}(S) \).
- **Claim 2:** \( (L \setminus S) \cup (S \cap R) \) is a vertex cover of \( G \)
  - Note that \( \delta_{out}(S) \) cannot contain edge from \( L \) to \( R \) (as these have \( \infty \) capacity).
  - Thus, every edge in \( G \) must be from \( L \setminus S \) or to \( S \cap R \) \( \Rightarrow \) vertex cover
Hall’s Theorem

Theorem (Hall’s Theorem)

A bipartite graph $G(L \sqcup R, E)$ with $|L| = |R| = n$ has a perfect matching if and only if for every subset $S \subseteq L$, it holds that $|N(S)| \geq |S|$. 
Hall’s Theorem

**Theorem (Hall’s Theorem)**

A bipartite graph \( G(L \sqcup R, E) \) with \( |L| = |R| = n \) has a perfect matching if and only if for every subset \( S \subset L \), it holds that \( |N(S)| \geq |S| \).

- Proof of this theorem can be derived from König’s theorem.
- **Hint:** can we have a vertex cover of size \(< n\) when the neighborhood constraints hold?
Applications of Max-Flow & Min-Cut
- Maximum Bipartite Matching
- Minimum Vertex Cover
- Edge-disjoint Paths
- Vertex-disjoint Paths

Further Remarks

Acknowledgements
Edge-Disjoint Paths

- **Input:** Directed (unweighted) graph $G(V, E)$, vertices $s, t \in V$
- **Output:** Maximum subset of edge-disjoint $s \rightarrow t$ paths
Edge-Disjoint Paths

- **Input:** Directed (unweighted) graph $G(V, E)$, vertices $s, t \in V$
- **Output:** Maximum subset of edge-disjoint $s \rightarrow t$ paths
- Simply set capacity of each edge to be 1, and run the max-flow algorithm for it.
**Edge-Disjoint Paths**

- **Input:** Directed (unweighted) graph $G(V, E)$, vertices $s, t \in V$
- **Output:** Maximum subset of edge-disjoint $s \to t$ paths

Simply set capacity of each edge to be 1, and run the max-flow algorithm for it.

**Claim 3:** there are $k$ edge-disjoint $s \to t$ paths iff there is $s \to t$ flow of value $k$

- $(\Rightarrow)$ given $k$ edge disjoint paths $P_1, \ldots, P_k$, we can simply get a flow of value $k$ by “adding” the paths $P_i$, that is, set the flow value to be 1 for each edge in one of the paths, and all other edges get 0 capacity
- $(\Leftarrow)$ given flow of value $k$, by flow decomposition theorem we have $k$ paths $P_1, \ldots, P_k$, and these must be edge disjoint, since for any $e \in E$, we have $0 \leq f(e) \leq c(e) = 1$. 
Edge-Disjoint Paths

- **Input:** Directed (unweighted) graph $G(V, E)$, vertices $s, t \in V$
- **Output:** Maximum subset of edge-disjoint $s \to t$ paths
- Simply set capacity of each edge to be 1, and run the max-flow algorithm for it.

**Claim 3:** there are $k$ edge-disjoint $s \to t$ paths iff there is $s \to t$ flow of value $k$

- $(\Rightarrow)$ given $k$ edge disjoint paths $P_1, \ldots, P_k$, we can simply get a flow of value $k$ by “adding” the paths $P_i$, that is, set the flow value to be 1 for each edge in one of the paths, and all other edges get 0 capacity
- $(\Leftarrow)$ given flow of value $k$, by flow decomposition theorem we have $k$ paths $P_1, \ldots, P_k$, and these must be edge disjoint, since for any $e \in E$, we have $0 \leq f(e) \leq c(e) = 1.$

- **Runtime:** Ford-Fulkerson takes $O(|V| \cdot |E|)$ time
Edge-Disjoint Paths

- **Input:** Directed (unweighted) graph $G(V, E)$, vertices $s, t \in V$
- **Output:** Maximum subset of edge-disjoint $s \rightarrow t$ paths
- Simply set capacity of each edge to be 1, and run the max-flow algorithm for it.
- **Claim 3:** there are $k$ edge-disjoint $s \rightarrow t$ paths iff there is $s \rightarrow t$ flow of value $k$
  - $(\Rightarrow)$ given $k$ edge disjoint paths $P_1, \ldots, P_k$, we can simply get a flow of value $k$ by “adding” the paths $P_i$, that is, set the flow value to be 1 for each edge in one of the paths, and all other edges get 0 capacity
  - $(\Leftarrow)$ given flow of value $k$, by flow decomposition theorem we have $k$ paths $P_1, \ldots, P_k$, and these must be edge disjoint, since for any $e \in E$, we have $0 \leq f(e) \leq c(e) = 1$.
- **Runtime:** Ford-Fulkerson takes $O(|V| \cdot |E|)$ time
- By the max-flow min-cut theorem, can prove:
  The maximum number of edge-disjoint $s \rightarrow t$ paths equals the minimum number of edges whose removal disconnects $s$ and $t$ (i.e., no $s \rightarrow t$ paths).
Applications of Max-Flow & Min-Cut

- Maximum Bipartite Matching
- Minimum Vertex Cover
- Edge-disjoint Paths
- Vertex-disjoint Paths

Further Remarks

Acknowledgements
**Vertex-Disjoint Paths**

- **Input:** Directed (unweighted) graph $G(V, E)$, vertices $s, t \in V$
- **Output:** Maximum subset of vertex-disjoint $s \rightarrow t$ paths
Vertex-Disjoint Paths

- **Input:** Directed (unweighted) graph $G(V, E)$, vertices $s, t \in V$
- **Output:** Maximum subset of vertex-disjoint $s \rightarrow t$ paths
- Reduce this problem to the edge-disjoint paths problem!
Vertex-Disjoint Paths

- **Input:** Directed (unweighted) graph \( G(V, E) \), vertices \( s, t \in V \)
- **Output:** Maximum subset of vertex-disjoint \( s \rightarrow t \) paths
- Reduce this problem to the edge-disjoint paths problem!
- For each \( u \in V \setminus \{ s, t \} \), replace it by two vertices \( u_1, u_2 \) and edges

\[
\begin{align*}
(u_1, u_2) \\
(w, u_1), & \quad \forall \ w \in N_{in}(u) \\
(u_2, v), & \quad \forall \ v \in N_{out}(u)
\end{align*}
\]
Vertex-Disjoint Paths

- **Input:** Directed (unweighted) graph $G(V, E)$, vertices $s, t \in V$
- **Output:** Maximum subset of vertex-disjoint $s \rightarrow t$ paths

Reduce this problem to the edge-disjoint paths problem!

For each $u \in V \setminus \{s, t\}$, replace it by two vertices $u_1, u_2$ and edges

$$
\begin{cases}
(u_1, u_2) \\
(w, u_1), \ \forall \ w \in N_{in}(u) \\
(u_2, v), \ \forall \ v \in N_{out}(u)
\end{cases}
$$

- **Claim 4:** There are $k$ vertex-disjoint $s \rightarrow t$ paths in $G \iff$ there are $k$ edge-disjoint $s \rightarrow t$ paths in the new graph.
Vertex-Disjoint Paths

- **Input**: Directed (unweighted) graph \( G(V, E) \), vertices \( s, t \in V \)
- **Output**: Maximum subset of vertex-disjoint \( s \to t \) paths

Reduce this problem to the edge-disjoint paths problem!

For each \( u \in V \setminus \{s, t\} \), replace it by two vertices \( u_1, u_2 \) and edges

\[
\begin{align*}
(u_1, u_2) \\
(w, u_1), \quad \forall \ w \in N_{in}(u) \\
(u_2, v), \quad \forall \ v \in N_{out}(u)
\end{align*}
\]

- **Claim 4**: There are \( k \) vertex-disjoint \( s \to t \) paths in \( G \) \iff there are \( k \) edge-disjoint \( s \to t \) paths in the new graph.

In this case, Ford-Fulkerson also gives us a \( O(|V| \cdot |E|) \) time algorithm.
Applications of Max-Flow & Min-Cut
- Maximum Bipartite Matching
- Minimum Vertex Cover
- Edge-disjoint Paths
- Vertex-disjoint Paths

Further Remarks

Acknowledgements
It may at first seem a little magic that vertex cover and matching are dual problems.

In fact several combinatorial optimization problems have very natural dual problems, and the knowledge of such duality is a powerful algorithmic tool!

Most (efficient) combinatorial optimization problems captured by *Linear Programming*

one of the most powerful framework for efficient computation.

Most of the dual statements seen here can be derived from *Linear Program Duality*

For more on this topic we encourage you all to take some courses in C&O about it.
Acknowledgement

Based on

- Prof. Lau’s Lecture 16
- Jeff Erickson’s book, Chapter 11
Cormen, Thomas and Leiserson, Charles and Rivest, Ronald and Stein, Clifford. (2009) 
MIT Press

Kleinberg, Jon and Tardos, Eva (2006) 
Algorithm Design. 
Addison Wesley