Part II of the course: optimization problems on graphs

**single-source shortest paths**
Find shortest path from source vertex to all other vertices

**all-pairs shortest paths**
Find shortest paths between all pairs of vertices

**maximum flow**
Find maximum flow from source vertex to target vertex

**maximum bipartite matching**
Find maximum number of disjoint pairs of vertices with edge between them
Single-Source Shortest Paths

Bellman-Ford algorithm
- can handle negative edge weights, works even for negative-weight cycles
- running time $\Theta(|V| \cdot |E|)$

Dijkstra’s algorithm
- works only for non-negative edge weights
- running time $\Theta(|V| \log |V| + |E|)$

All-Pairs Shortest Paths

Dynamic-programming algorithms
- connection to matrix-multiplication
- improved version (repeated squaring) runs in $\Theta(|V|^3 \log |V|)$ time
- different subproblems give simple $\Theta(|V|^3)$-time algorithm (Floyd-Warshall)

Johnson’s algorithm: reweighting
- modify graph to make all edge-weights non-negative
- then run Dijkstra’s algorithm $|V|$ times
- running time $\Theta(|V|^2 \log |V| + |V| \cdot |E|)$
The maximum-flow problem

How much flow we can push through the network?

- edges are directed and have a maximum capacity
- source is generating the flow, sink is consuming it
- all flow arriving at a node must also leave the node, except at sink
Flow network: directed graph \( G = (V, E) \), where

- each edge \((u,v)\) in \(E\) has a capacity \(c(u,v) \geq 0\)
  - define \(c(u,v)=0\) if \((u,v)\) not in \(E\)
- there are two special nodes in \(V\): the source \(s\) and the sink \(t\)
- if \((u,v)\) in \(E\) then \((v,u)\) not in \(E\)  
  
  **trick:** insert node on edge
- Assume that any node \(u\) can be reached from \(s\) and can reach \(t\)

**Note:** \(G\) can have cycles
Flow: function $f : V \times V \rightarrow \mathbb{R}$ satisfying

- **capacity constraint**: $0 \leq f(u,v) \leq c(u,v)$ for all nodes $u,v$
  
  (Hence, $f(u,v) = 0$ if edge $(u,v)$ does not exist.)

- **flow conservation**: for all nodes $u \neq s, t$ we have $\text{flow in} = \text{flow out}$:
  
  $$\sum_{v \in V} f(v,u) = \sum_{v \in V} f(u,v)$$

value of flow: $|f| = \sum_{v \in V} f(s,v) - \sum_{v \in V} f(v,s)$
Flow: function \( f : V \times V \rightarrow \mathbb{R} \) satisfying

- capacity constraint: \( 0 \leq f(u,v) \leq c(u,v) \) for all nodes \( u, v \)
  (Hence, \( f(u,v) = 0 \) if edge \((u,v)\) does not exist.)

- flow conservation: for all nodes \( u \neq s, t \) we have flow in = flow out:
  \[
  \sum_{v \in V} f(v,u) = \sum_{v \in V} f(u,v)
  \]

value of flow: \( |f| = \sum_{v \in V} f(s,v) - \sum_{v \in V} f(v,s) \)

Can there be flow if all outgoing edges from the source have zero flow?

\[
\begin{array}{c}
\text{s} & \overset{0/2}{\rightarrow} & \overset{2/2}{\rightarrow} & \overset{0/3}{\rightarrow} & \text{t} \\
\overset{2/3}{\leftarrow} & \overset{2/5}{\rightarrow} & \overset{2/5}{\rightarrow}
\end{array}
\]
Idea for computing max flow incrementally

- start with zero flow
- repeat until stuck
  - find path from $s$ to $t$ along which we can increase the flow
  - increase flow along the path

Total flow: 0
Idea for computing max flow incrementally

- start with zero flow
- repeat until stuck
  - find path from s to t along which we can increase the flow
  - increase flow along the path

Total flow: 0 + 2
Idea for computing max flow incrementally

- start with zero flow
- repeat until stuck
  - find path from $s$ to $t$ along which we can increase the flow
  - increase flow along the path

Total flow: $0 + 2$
Idea for computing max flow incrementally

- start with zero flow
- repeat until stuck
  - find path from $s$ to $t$ along which we can increase the flow
  - increase flow along the path

Total flow: $0 + 2 + 5$
Idea for computing max flow incrementally

- start with zero flow
- repeat until stuck
  - find path from $s$ to $t$ along which we can increase the flow
  - increase flow along the path

```
0 + 2 + 5
```
Idea for computing max flow incrementally

- start with zero flow
- repeat until stuck
  - find path from $s$ to $t$ along which we can increase the flow
  - increase flow along the path

**trick:** decrease flow along edge = send flow along reverse edge
Idea for computing max flow incrementally

- start with zero flow
- repeat until stuck
  - find path from $s$ to $t$ along which we can increase the flow
  - increase flow along the path

**Trick:** decrease flow along edge = send flow along reverse edge

Total flow: $0 + 2 + 5 + 1$
The Ford-Fulkerson method

work with residual network $G_f$

- original edges
- reverse edges: sending flow along such an edge
  
  = decreasing flow along original edge

**Ford-Fulkerson-Method** ($G, s, t$)

1. Initialize flow: set $f(u,v) = 0$ for each pair $(u,v)$ in $V \times V$
2. **while** there is an augmenting path $p$ in the residual network $G_f$
3. **do** increase flow by augmenting flow along $p$
4. **return** $f$
- \( G = (V,E) \) is flow network with source \( s \) and sink \( t \)
- \( f = \) flow on \( G \)

**Residual capacity** of pair of vertices \( u,v \) in \( G \) (for the given flow \( f \))

\[
c_f(u,v) = \begin{cases} 
  c(u,v) - f(u,v) & \text{if } (u,v) \text{ in } E \quad \text{// original edge} \\
  f(v,u) & \text{if } (v,u) \text{ in } E \quad \text{// reverse edge} \\
  0 & \text{otherwise}
\end{cases}
\]

well defined because \( (u,v) \) and \( (v,u) \) cannot both be in \( E \)

![Diagram](image-url)
Residual network of network $G$ with given flow $f$

network $G_f = (V, E_f)$ where $E_f = \{ (u,v) \in V \times V : c_f(u,v) > 0 \}$

- $(u,v)$ in $E$: $E_f$ can contain $(u,v)$ and/or $(v,u)$
- $(u,v)$ not in $E$ and $(v,u)$ not in $E$: $(u,v)$ and $(v,u)$ not in $E_f$ either

![Diagram of residual network]
Let’s look at earlier example, where we had to decrease flow along some edge.
flow in residual network $G_f$

- should satisfy residual capacity constraints
- for each vertex $\neq s,t$: flow in = flow out

We will only use flows along **augmenting path** $p$ = simple $s$-to-$t$ path in $G_f$

residual capacity $c_f(p)$ of $p$ = amount of flow we can push along $p$

$= \min \{ c_f(u,v): (u,v) \text{ is an edge of } p \}$
Augmenting flow \( f \) by flow along simple \( s \)-to-\( t \) path \( p \) in \( G_f \)

\[
\text{for each edge } (u,v) \text{ on the augmenting path } p \\
\text{do } \begin{cases} \text{if } (u,v) \in E & \text{then increase } f(u,v) \text{ by } c_f(p) \ \\
\text{else} & \text{decrease } f(v,u) \text{ by } c_f(p) \end{cases}
\]
Lemma: Augmenting flow along simple s-to-t path $p$ gives valid flow in $G$ whose value is $(\text{old flow value}) + \text{residual capacity } c_f(p)$.

Proof. Must show:
- new flow satisfies capacity constraint: $0 \leq f(u,v) \leq c(u,v)$
- new flow satisfies: flow in = flow out
- new flow value = $(\text{old flow value}) + \text{residual capacity } c_f(p)$. 

$$c_f(u,v) = \begin{cases} 
  c(u,v) - f(u,v) & \text{if } (u,v) \text{ in } E \\
  f(v,u) & \text{if } (v,u) \text{ in } E \\
  0 & \text{otherwise}
\end{cases}$$
Ford-Fulkerson (G, s, t)

1. Initialize flow: set $f(u,v) = 0$ for each pair $(u,v)$ in $V \times V$
2. Construct residual network $G_f$
3. while there is an augmenting path $p$ in the residual network $G_f$
4. do // increase flow by augmenting flow along $p$
5. $c_f(p) \leftarrow$ residual capacity of the path $p$
6. for each edge $(u,v)$ on the path $p$
7. do if $(u,v)$ in $E$
8. then $f(u,v) \leftarrow f(u,v) + c_f(p)$
9. else $f(v,u) \leftarrow f(v,u) - c_f(p)$
10. Update the residual network $G_f$
11. return $f$

need algorithm to find path between two given vertices; for the moment we assume we just find any path
Properties of Ford-Fulkerson:

- Invariant: flow is valid
- Flow increases at each iteration

Questions:

- Are we sure it always terminates?
  
  not guaranteed if capacities are irrational and augmenting paths are chosen in the “wrong” way

- How many iterations before we get stuck (no augmenting path) ?
  
  more next lecture

- Are we sure we have max flow when we get stuck ?
  
  want to prove: no augmenting path $\rightarrow$ max flow
Cuts of flow network $G = (V,E)$

$\text{cut (}S, T\text{)} = \text{partitioning of } V \text{ into subsets } S \text{ and } T, \text{ with } s \text{ in } S \text{ and } t \text{ in } T$

flow $f(S, T)$ across cut $(S, T) = \sum_{u \in S} \sum_{v \in T} f(u, v) - \sum_{u \in S} \sum_{v \in T} f(v, u)$

$= (2 + 2 + 3 + 1) - (1 + 1) = 6$

capacity $c(S, T)$ of cut $(S, T) = \text{max flow across the cut}$

$= \sum_{u \in S} \sum_{v \in T} c(u, v) = (2 + 3 + 3 + 5) = 13$
Lemma: Flow across any cut is the same, and equals the value of the flow.

Proof. See book.
Lemma: Flow across any cut is the same, and equals the value of the flow.

Note: capacity of cuts is not necessarily the same

minimum cut = cut whose capacity is minimum

Corollary: Maximum flow cannot be more than capacity of minimum cut.

Proof. Consider flow $f$ of maximum value

value of $f$ = flow of $f$ across any cut

= flow of $f$ across minimum cut

$\leq$ capacity of minimum cut
Corollary: Maximum flow cannot be more than capacity of minimum cut.

Max-flow min-cut Theorem: Let $f$ be a flow in a flow network $G$. Then the following conditions are equivalent:

(i) $f$ is a maximum flow in $G$

(ii) residual network $G_f$ contains no augmenting path

(iii) there is a cut $(S,T)$ with $|f| = c(S,T)$

Combine (iii) with corollary: maximum flow = capacity of minimum cut
Max-flow min-cut Theorem: Let $f$ be a flow in a flow network $G$. Then the following conditions are equivalent:

(i) $f$ is maximum flow in $G$

(ii) residual network $G_f$ contains no augmenting path

(iii) there is a cut $(S,T)$ with $|f| = c(S,T)$

Proof.

(i) $\iff$ (ii):

If there were an augmenting path, we could increase the flow.
Max-flow min-cut Theorem: Let $f$ be a flow in a flow network $G$. Then the following conditions are equivalent:

(i) $f$ is maximum flow in $G$
(ii) residual network $G_f$ contains no augmenting path
(iii) there is a cut $(S,T)$ with $|f| = c(S,T)$

Proof.
(iii) $\implies$ (ii):
Define: $S =$ set of all vertices that can be reached from $s$ in $G_f$
$T = V - S$

- Consider edge $(u,v)$ from $S$ to $T$:
  $$f(u,v) = c(u,v) \text{ otherwise } (u,v) \in E_f \text{ and } v \text{ is reachable}$$
- Consider edge $(v,u)$ from $T$ to $S$:
  $$f(v,u) = 0 \text{ otherwise } (u,v) \in E_f \text{ and } v \text{ is reachable}$$

Hence, $|f| = c(S,T)$
Max-flow min-cut Theorem: Let $f$ be a flow in a flow network $G$. Then the following conditions are equivalent:

(i) $f$ is maximum flow in $G$

(ii) residual network $G_f$ contains no augmenting path

(iii) there is a cut $(S, T)$ with $|f| = c(S, T)$

Proof.

(iii) $\iff$ (i): follows from Corollary
Max-Flow Summary

Flow network
- directed graph with source and sink, and capacities on the edges
- if $(u,v)$ in $E$ then we cannot have $(v,u)$ in $E$

Flow in a network
- must satisfy capacity constraints and “flow in = flow out”

Ford-Fulkerson method
- iteratively increase flow using augmenting paths in residual graph

Max flow = min cut

more on flows next lecture