Maximum Flow Problems

John E. Mitchell
http://www.rpi.edu/~mitchj

RPI

November 5, 2017
The Ford-Fulkerson Algorithm

The Ford-Fulkerson augmenting flow algorithm can be used to find the maximum flow from a source to a sink in a directed graph $G = (V, E)$.

Each arc $(i, j) \in E$ has a capacity of $u_{ij}$.

We find paths from the source to the sink along which the flow can be increased.

The paths might include arcs facing in the reverse direction from the path; flow is decreased on these arcs.
Example

This graph has a source $s$, a destination or sink $t$, and three transshipment nodes. The capacities of the edges are listed in the figure.

There is no initial flow. One augmenting path is $s \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow t$. 10 units can be pushed along this arc, at which point arc $(2, 3)$ is saturated.
The labels on each arc represent (capacity, flow).

There are now augmenting paths in the graph that result in a reduction of flow on arc (1,2). One possible augmenting path is \( s \rightarrow 2 \rightarrow 1 \rightarrow t \).
Augmenting path

The path is indicated by green forward arcs and red reverse arcs and the flow on the path is denoted $f$.

The maximum possible value for the flow is $f = 10$. 
There is another augmenting path in the graph, $s \rightarrow 1 \rightarrow t$, with both arcs used in the forward direction.
Augmenting path

The path is indicated by green forward arcs and the flow on the path is denoted $f$.

The maximum possible value for the flow is $f = 5$. 
If we try to augment flow further, we cannot push flow along the arc \((s, 1)\). We can push flow along \((s, 2)\), but no further: arc \((2, 3)\) is saturated, and the arc \((1, 2)\) entering node 2 is empty.

So this flow is **optimal**.
Cuts

We can divide the graph into two sets based on the flow:

- Nodes to which we can push flow from \( s \): \( s \) itself, and also node 2. Let \( S_s := \{ s, 2 \} \).
- Nodes to which we cannot push flow from \( s \): nodes 1, 3, \( t \). Let \( S_t := \{ 1, 3, t \} \).

The arcs between \( S_s \) and \( S_t \) constitute an \((s, t)\)-cut in the graph: the source is on one side of the cut and the sink is on the other.

All flow from \( s \) to \( t \) has to flow across this cut.

There is no way to increase the flow further:

- all the arcs in the cut pointing from \( S_s \) to \( S_t \) are saturated,
- and all the arcs pointing in the reverse direction are empty.

The capacity of the cut is the sum of the capacities of the arcs in the cut pointing from \( S_s \) to \( S_t \).

It is a fundamental result that \textbf{Max Flow} = \textbf{Min Cut}. 
Cuts

We can divide the graph into two sets based on the flow:

- Nodes to which we can push flow from $s$: $s$ itself, and also node 2. Let $S_s := \{s, 2\}$.
- Nodes to which we cannot push flow from $s$: nodes 1, 3, $t$. Let $S_t := \{1, 3, t\}$.

The arcs between $S_s$ and $S_t$ constitute an $(s, t)$-cut in the graph: the source is on one side of the cut and the sink is on the other.

All flow from $s$ to $t$ has to flow across this cut.

There is no way to increase the flow further:
- all the arcs in the cut pointing from $S_s$ to $S_t$ are saturated,
- and all the arcs pointing in the reverse direction are empty.

The capacity of the cut is the sum of the capacities of the arcs in the cut pointing from $S_s$ to $S_t$.

It is a fundamental result that $\textbf{Max Flow} = \textbf{Min Cut}$.
Max Flow equals Min Cut

This result can be proved using LP duality. In our example problem, the max flow problem can be written as the following linear program, using a variable $x_{ts}$ to represent the total flow from $s$ to $t$:

$$\max_x \quad x_{ts}$$

s.t.

$$x_{s1} + x_{s2} - x_{ts} = 0$$
$$-x_{s1} + x_{12} + x_{13} + x_{14} = 0$$
$$-x_{s2} - x_{12} + x_{23} = 0$$
$$-x_{13} - x_{23} + x_{3t} = 0$$
$$-x_{1t} - x_{3t} + x_{ts} = 0$$

$$0 \leq x_{s1} \leq 15, \quad 0 \leq x_{1t} \leq 16$$
$$0 \leq x_{s2} \leq 17, \quad 0 \leq x_{23} \leq 10$$
$$0 \leq x_{12} \leq 12, \quad 0 \leq x_{3t} \leq 13$$
$$0 \leq x_{13} \leq 8$$
In the dual LP, we have variables $y_i$ for each vertex $i$, and variables $w_{ij}$ corresponding to the upper bounds on each flow $x_{ij}$:

$$\min_{y,w} \quad 15w_{s1} + 17w_{s2} + 12w_{12} + 8w_{13} + 16w_{1t} + 10w_{23} + 13w_{3t}$$

s.t.

$$y_s - y_1 + w_{s1} \geq 0$$
$$y_s - y_2 + w_{s2} \geq 0$$
$$y_1 - y_2 + w_{12} \geq 0$$
$$y_1 - y_3 + w_{13} \geq 0$$
$$y_1 - y_t + w_{1t} \geq 0$$
$$y_2 - y_3 + w_{23} \geq 0$$
$$y_3 - y_t + w_{32} \geq 0$$
$$y_t - y_s = 1$$

$w_{ij} \geq 0$ for all arcs (i,j)
Cuts and dual solutions


text content
The optimal cut and dual solution

\[ y_s = 0 \]
\[ y_1 = 1 \]
\[ y_2 = 0 \]
\[ y_3 = 1 \]
\[ y_t = 1 \]