Artificial Intelligence

Max Flow Problem

G. Guérard

Department of Nouvelles Energies
Ecole Supérieur d'Ingénieurs Léonard de Vinci

Lecture 4

Outline

Application

Various problems can be reducted into MAX FLOW PROBLEM, for examples:

- NETWORK CONNECTIVITY: *Edge-disjoint paths, Network reliability.*
- MATCHING: *Bipartite matching, Maximum matchings.*
- VERTEX CAPACITIES: *Data mining.*
- SCHEDULING: *Airline scheduling, Project selection.*
- GRAPH CUTS: *Image processing.*
- ASSIGNEMENT: *Baseball elimination, Distributed computing.*

Min cut problem

Flow network

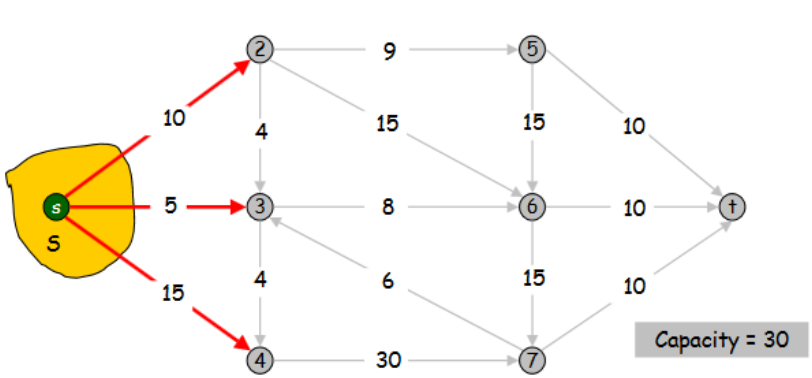
Let $G = (V, E)$ be a directed graph with two distingished nodes s, the source, and t, the sink. And a capacity $c(e)$ for each edge $e \in E$.

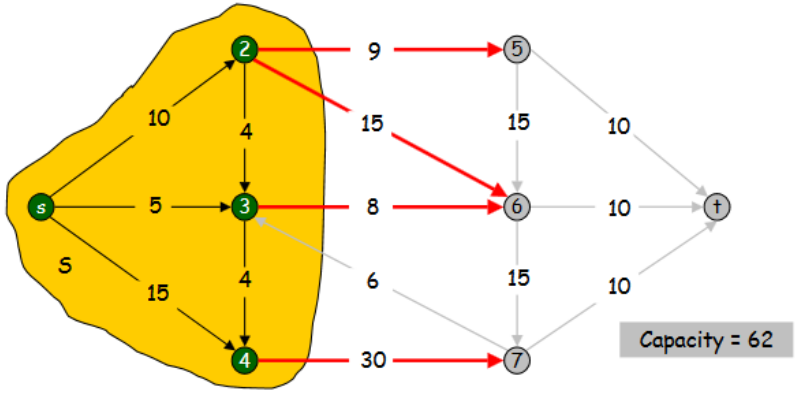
Min cut problem

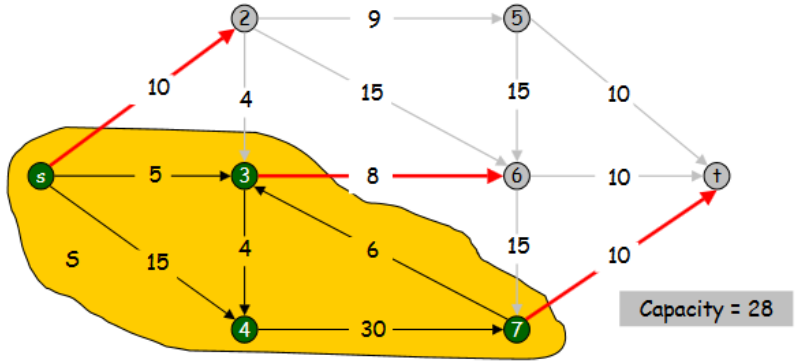
Delete a set of edges to disconnect t from s. Found a set to minimize the sum of edge's capacity. A CUT is a node partition (S, T) such that s is in S and t is in T.

MINIMIZE THE SUM OF WEIGHTS OF EDGES LEAVING S.

Min cut problem







Flows

Flow

A FLOW f is an assignment of weights to edges so that. The flow can't go over the capacity of an edge. *The flow entering in a vertex is equal to the flow leaving it.* An $s-t$ flow is a function that satisfies: for each $e \in E, 0 \leq f(e) \leq c(e)$ (capacity); for each $v \in V - \{s,t\}$, $\sum_{e \in \Gamma^+} f(e) = \sum_{e \in \Gamma^-} f(e)$.

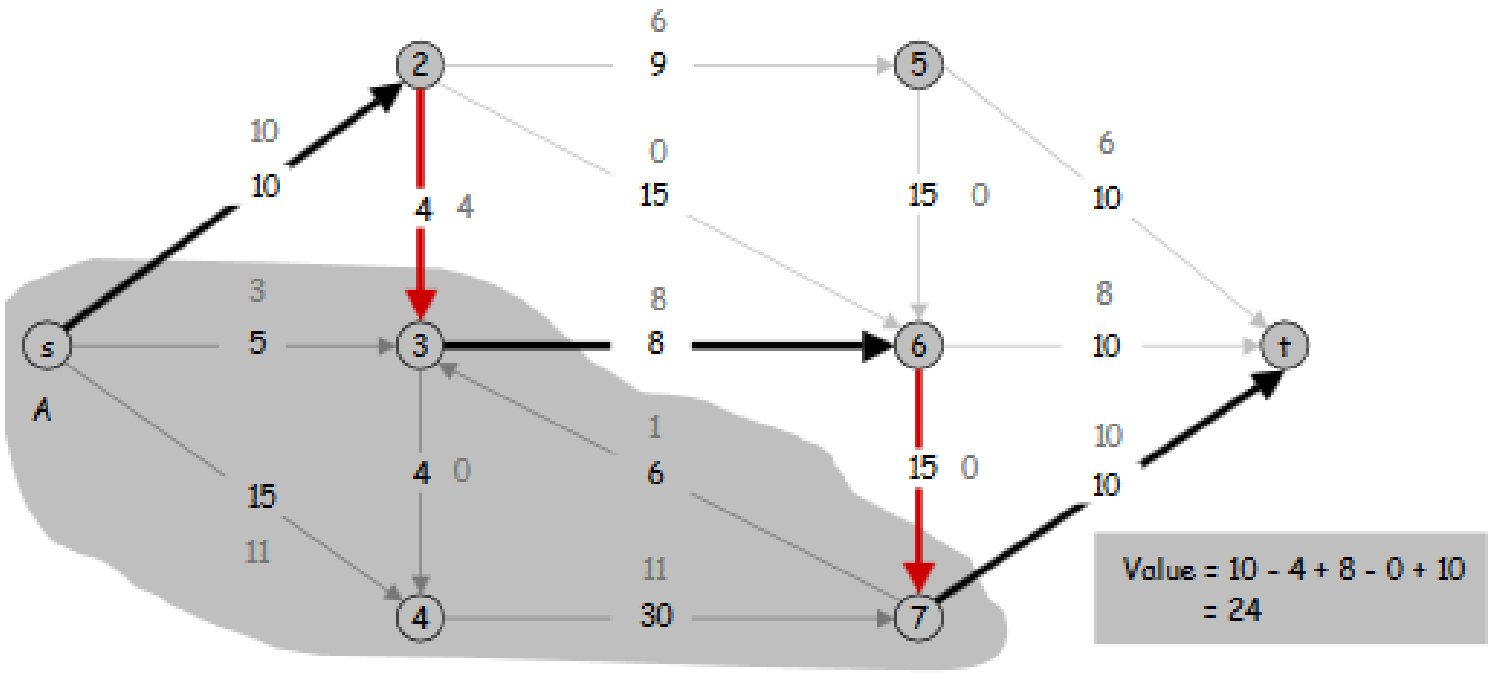
Max flow

Find $s-t$ flow of maximum value $v(f)$. An simple and easy way to know the max flow is to sum flow out of s or to sum flow in to t: $v(f) = \sum_{e \in \Gamma^+(s)} f(e)$.

Flows

Flow value Lemma

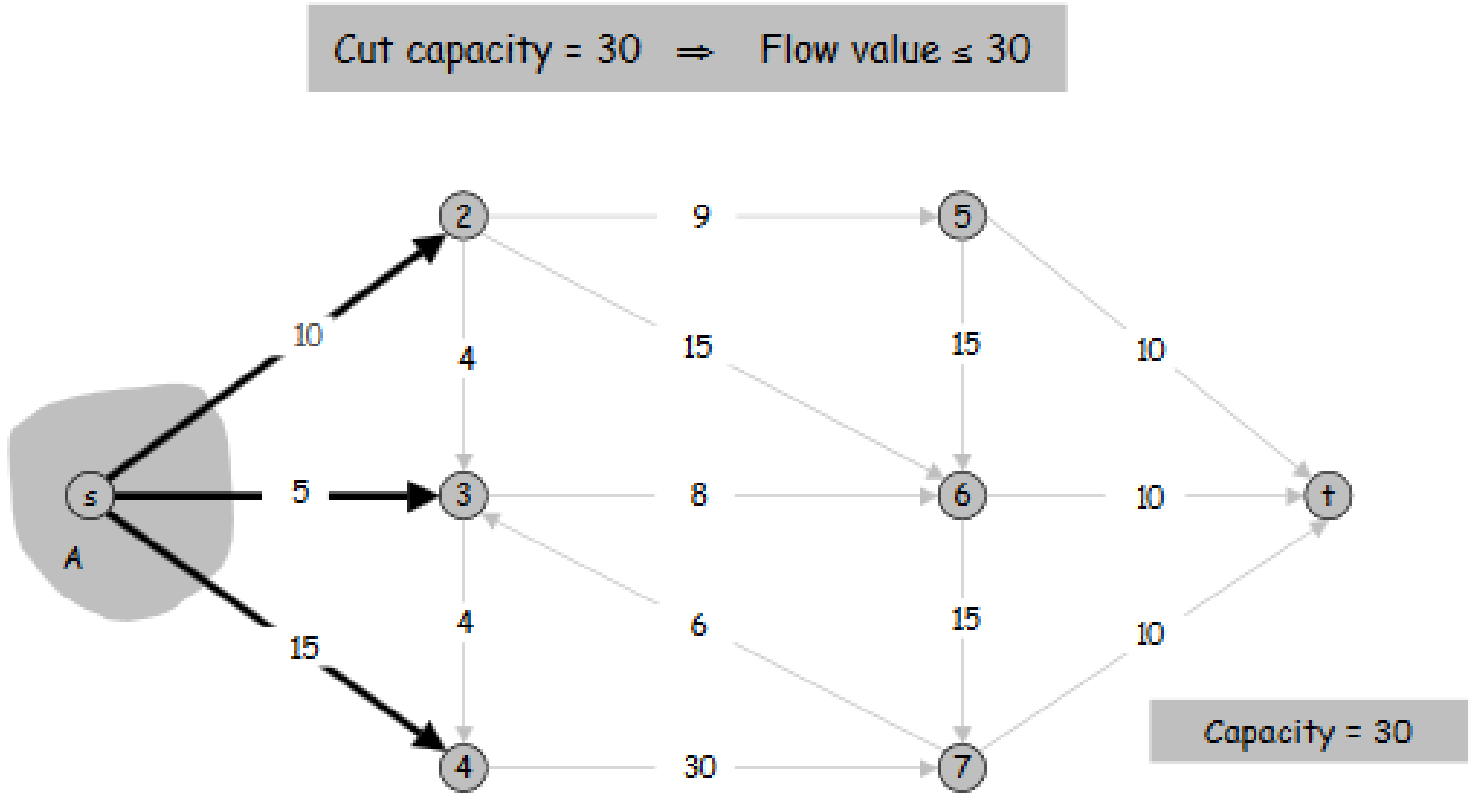
Let f be any flow, and let (A, B) be any cut. Then, the net flow sent across the cut is equal to the amount leaving s: $v(f) = \sum_{e \in \Gamma^+(A)} f(e) - \sum_{e \in \Gamma^-(A)} f(e)$.



Flows and cuts

Weak duality

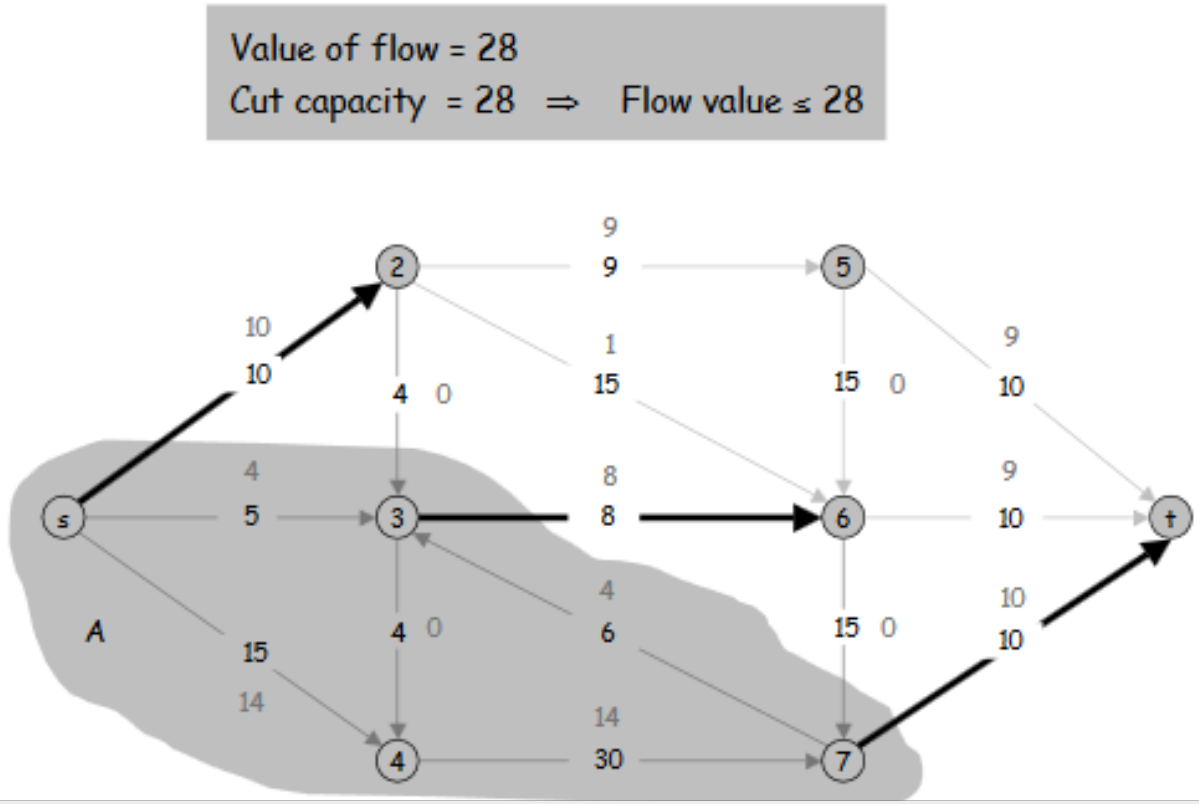
Let f be any flow, and let (A, B) be any cut. Then the value of the flow is at most the capacity of the cut.



Flows and cuts

Corollary

Let f be any flow, and let (A, B) be any cut. If $v(f)$ is equal to the capacity of the cut, then f is a max flow and (A, B) forms a min cut.

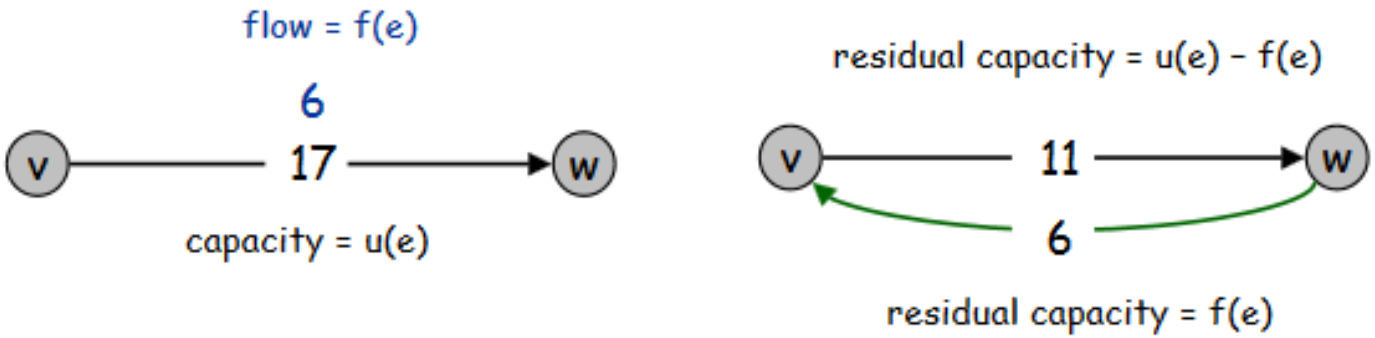


Augmenting path

Find a $s-t$ path where each arc has $f(e) < u(e)$ and augment flow along it by the minimum of $u(e) - f(e)$. A naive algorithm consists to repeat this process until you get stuck. This method do not give an optimal value, we need to be able to backtrack.

Residual graph

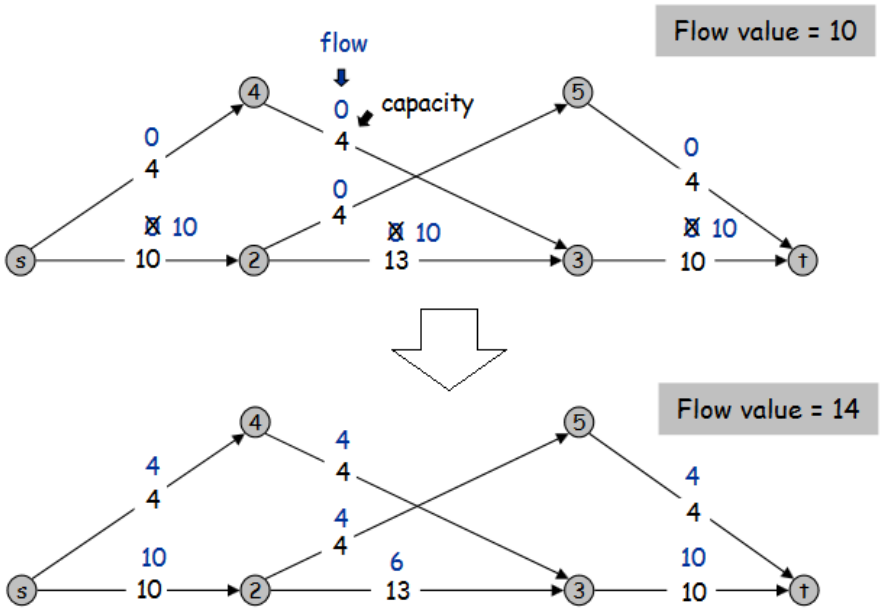
Let e be an edge from v to w, $u(e)$ its capacity and $f(e)$ the flow passing by this edge. A residual graph $G' = (V, E')$ represents each possible evolution of flows in the graph $G = (V, E)$. To construct G' all the arc that have strictly positive flow are duplicate, the new arc go from the end node to the start node, its capacity is equal to the flow. The previous has a capacity equal to $u(e) - f(e)$.

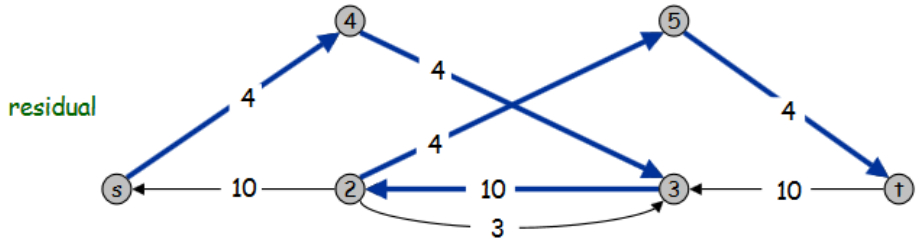


Augmenting path

Augmenting path

Find a $s - t$ path in residual graph, it is called augmenting path. Once the path is selected, increase flow along forward edges, decrease flow along backward edges.





Algorithm

FORD-FULKERSON's algorithm is a generic (greedy) method for solving max flow.

- *While there exists an augmenting path*
  - *find augmenting path P*
  - *compute bottleneck capacity of P*
  - *augment flow along P*

Theorem

A flow f is a max flow iff there are no augmenting paths.

Algorithm

THOSE THREE SENTENCES ARE EQUIVALENT:

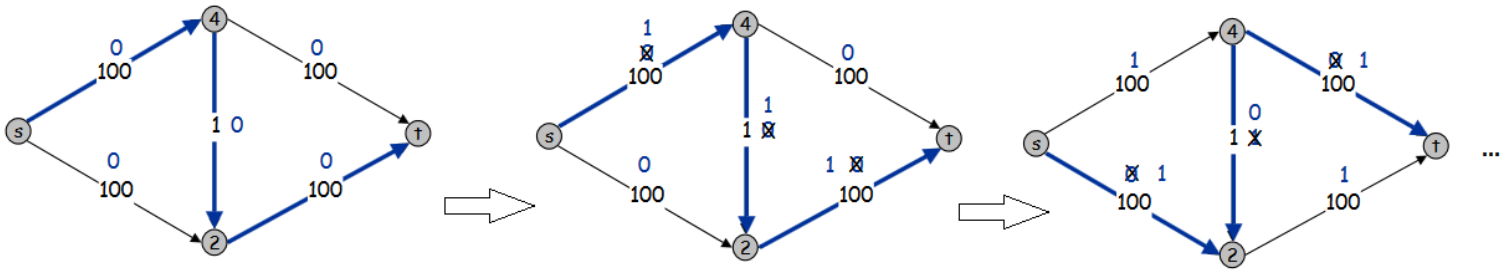
- *f is a max flow.*
- *There is no augmenting path relative to f.*
- *There exists a cut whose capacity equals the value of f.*

SO, THE FOLLOWING PROBLEMS ARE EQUIVALENT:

- *Find a max flow.*
- *Find a min cut.*
- *Use residual graph to find bottlenecks (backward edges without respecting forwward edges).*
- *Use residual graph to find a min cut (minimum value set of backward edges without respecting forward edges).*

Good Augmenting Path

Use care when selecting augmenting paths. Design goal is to choose augmenting paths so that can find augmenting paths efficiently and in few iterations, not like:



Fundamental knowledge

You have to know before the tutorial:

1. *Flow definition and max flow problem.*
2. *Min-cut problem and weak duality.*
3. *Augmenting path definition.*
4. *Ford-Fulkerson's algorithm.*