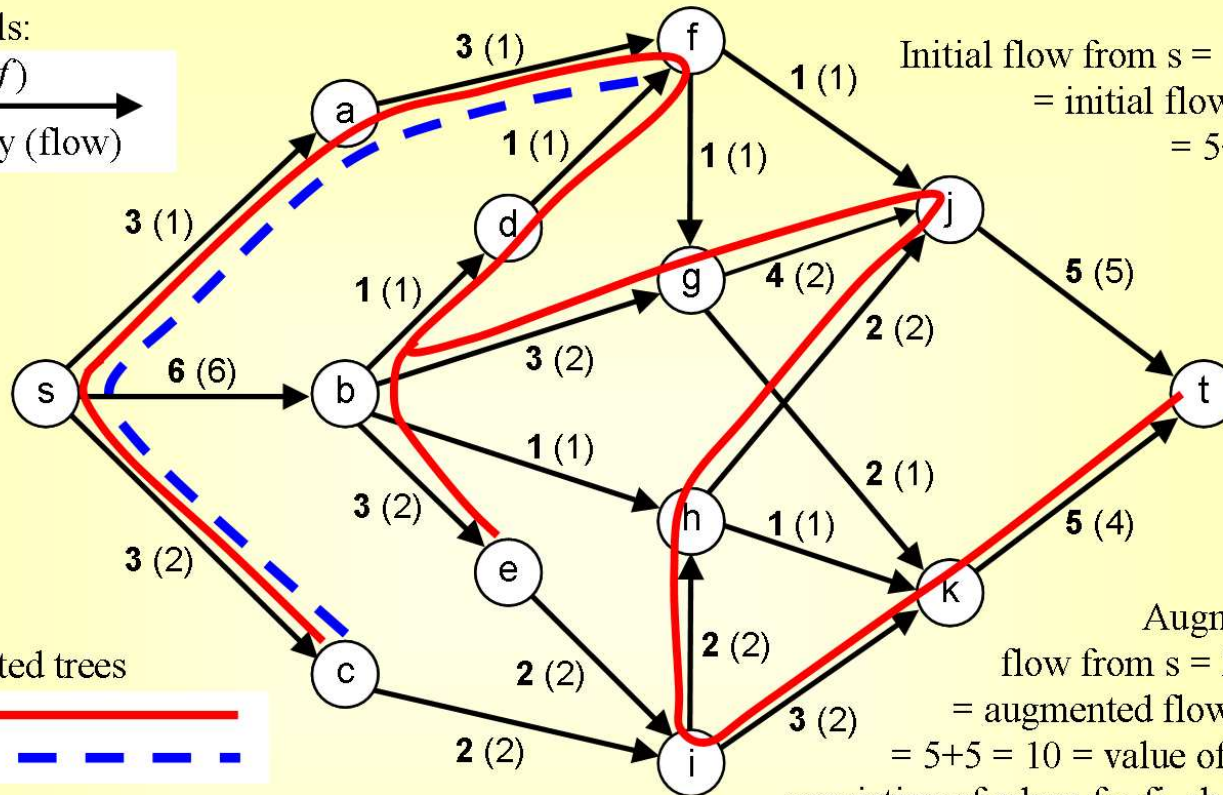
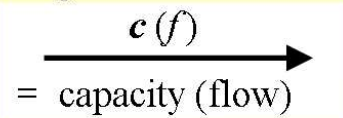




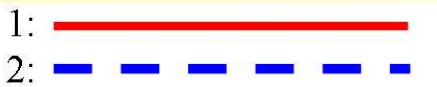
THEOREM OF THE DAY

The Max-Flow Min-Cut Theorem Let $N = (V, E, s, t)$ be an st -network with vertex set V and edge set E , and with distinguished vertices s and t . Then for any capacity function $c : E \rightarrow \mathbb{R}^{\geq 0}$ on the edges of N , the maximum value of an st -flow is equal to the minimum value of an st -cut.

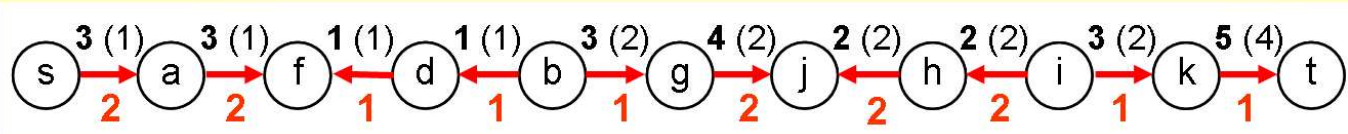
Edge labels:



f -unsaturated trees



f -unsaturated st -path in tree 1:



slack in path = min slack on path edges = 1 (value by which f may be increased/decreased)

Initial flow from $s = 1+6+2 =$ initial flow into $t = 5+4 = 9$

Augmented flow from $s = 2+6+2 =$ augmented flow into $t = 5+5 = 10 =$ value of st -cut consisting of edges fg, fj, sb and ci

In the slightly simplified version illustrated here, an st -network is a directed graph in which no edges enter s nor exit t . An st -flow is a function which, like the capacity function, maps each edge to a nonnegative real number. Additionally it must satisfy:

1. $f(e) \leq c(e)$ for all edges e ;
2. the total flow into any vertex $v \neq s, t$ must equal the total flow leaving it.

Under condition (2), the total flow into t will equal the total flow leaving s ; this total is called the *value* of the flow. Condition (1) bounds the flow value by the total capacity of any st -cut: a set of edges whose removal separates s from t .

On the left, flow value is augmented from 9 to 10 using the Ford-Fulkerson Algorithm: this searches, breadth-first, for an undirected path from s to t in which forward edges have $f(e) < c(e)$ and backward edges have $f(e) > 0$. Along such a so-called f -unsaturated path, f may be increased (decreased) on forward (backward) edges. If the search terminates without reaching t then the set of vertices reached identifies a minimum st -cut: and by the theorem, flow has been maximised.

This theorem characterising optimal transportation in capacity constrained networks was published independently in 1956 by: L.R. Ford Jr and D.R. Fulkerson; by P. Elias, A. Feinstein and C.E. Shannon; and, restricted as in our example to integer capacities, by A. Kotzig.

Web link: www.cse.buffalo.edu/~hungngo/classes/2004/594/notes/Flow-intro.pdf; and see homepages.cwi.nl/~lex/files/histrpclean.pdf for some fascinating prehistory.

Further reading: *Combinatorial Optimization: Algorithms and Complexity* by C. Papadimitriou and K. Steiglitz, Dover Publications, 2000, chapter 6.

