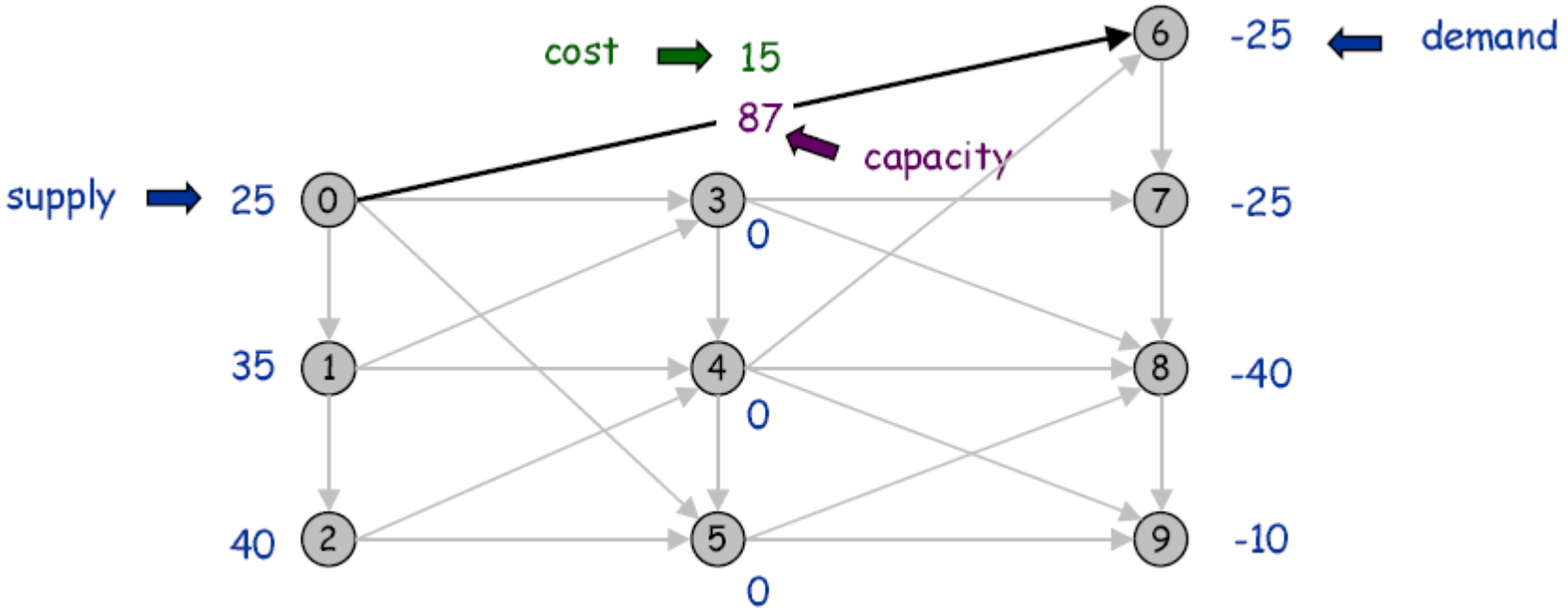
Min-Cost Flow Problem

Min-cost flow problem.

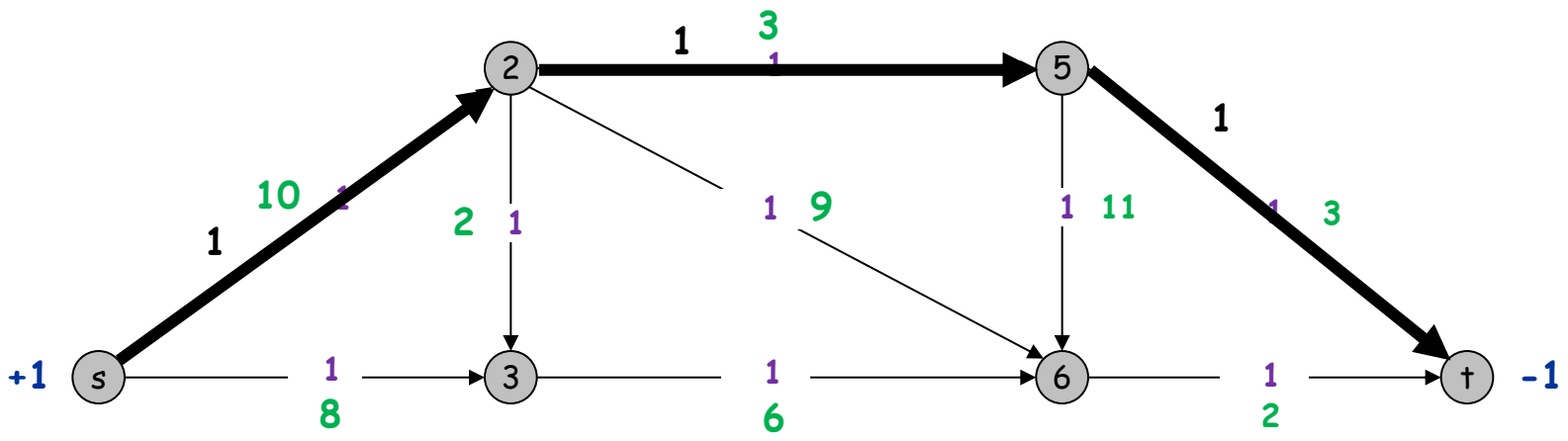
- $G = (V, E)$ = directed graph, no parallel edges.
- s_v = net supply / demand at vertex v (sum of supply = sum of demand)
- c_{vw} = unit shipping cost from v to w
- u_{vw} = capacity of edge v-w.
- Goal: send all supply to demand at minimum cost



Applications: Logistics and many others

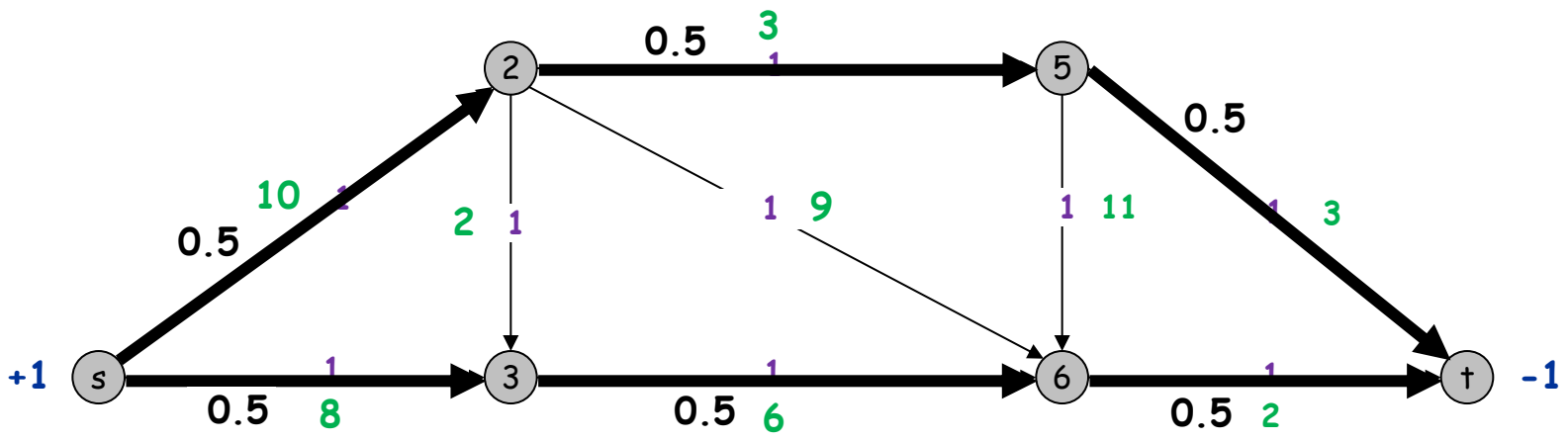
Min-Cost Flow: Quick Applications

Generalizes shortest path problem:



Min-Cost Flow: Quick Applications

Generalizes shortest path problem:



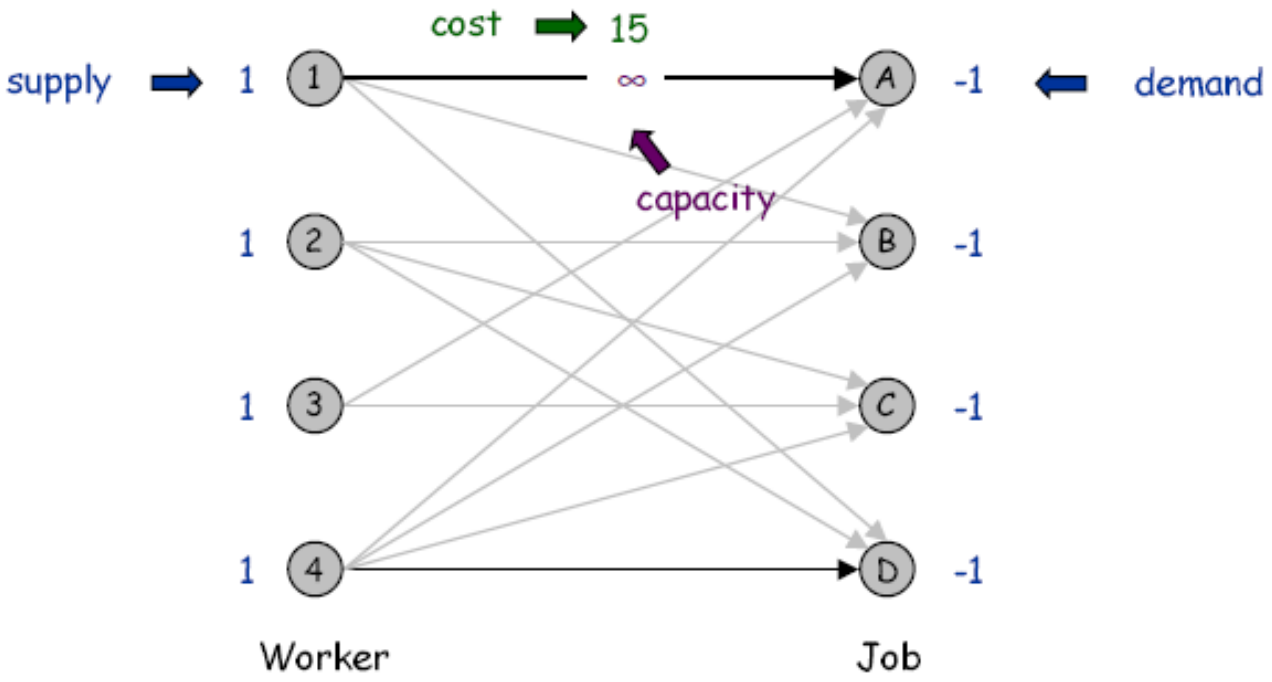
Careful: Technically the flow can be fractional

But we will argue again an Integrality Theorem, where if the capacities and demands are integral, there is a min-cost flow that is integral

Min-Cost Flow: Quick Applications

Assigment problem

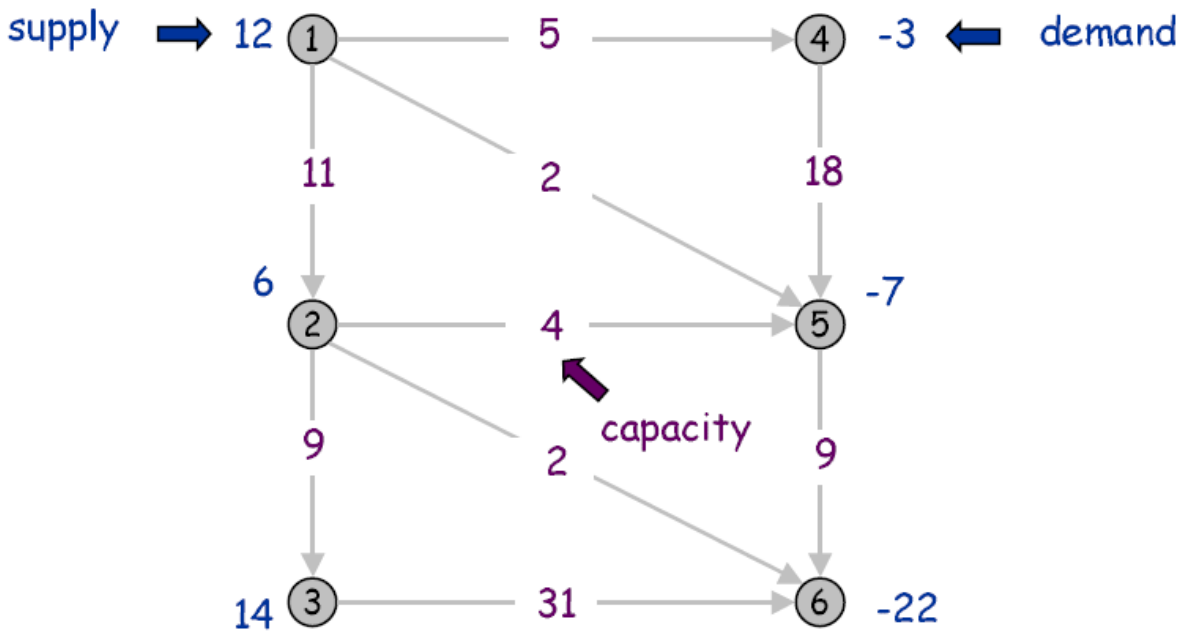
- Assign workers to jobs
- c_{vw} = cost of assigning worker v to job w
- **Goal**: Minimize total cost



Starting to Find Min-Cost Flow

Start by finding a feasible (not necessarily minimum) flow, if one exists

Q: Can we do it using max flow?

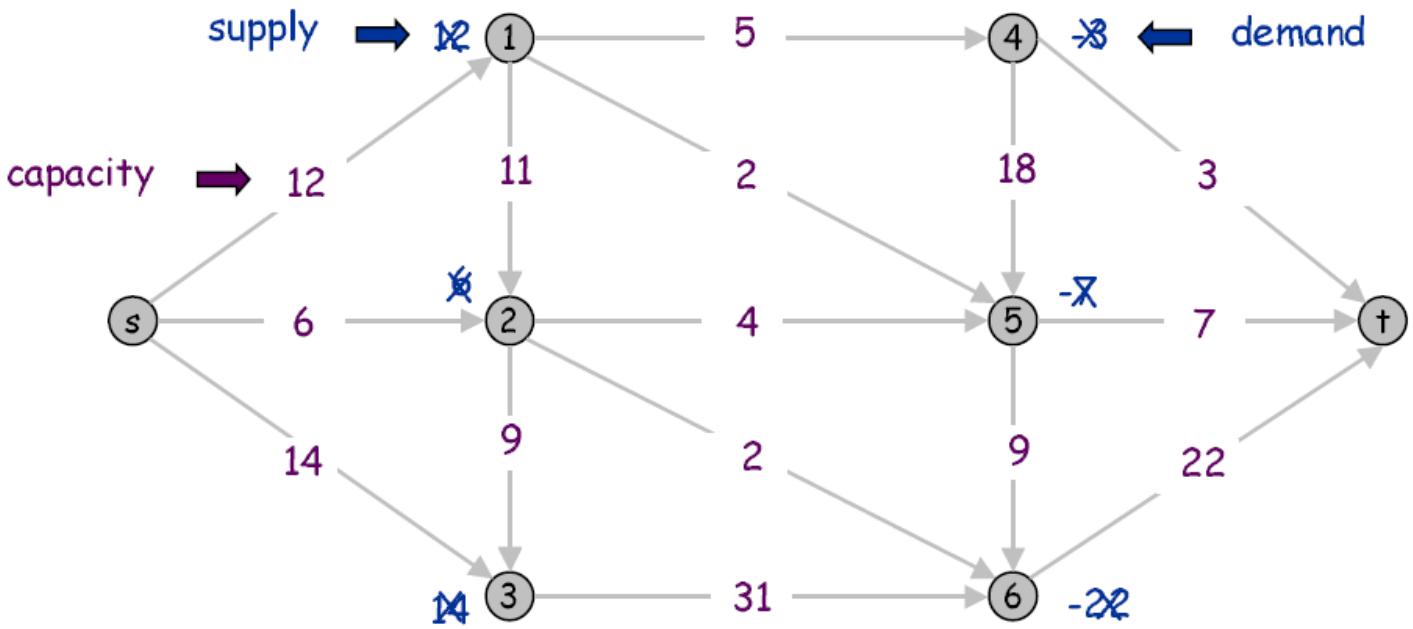


Starting to Find Min-Cost Flow

Start by finding a feasible (not necessarily minimum) flow, if one exists

Q: Can we do it using max flow?

A: Find max flow in a related network. There is feasible flow for the original network iff this maxflow saturates all new edges

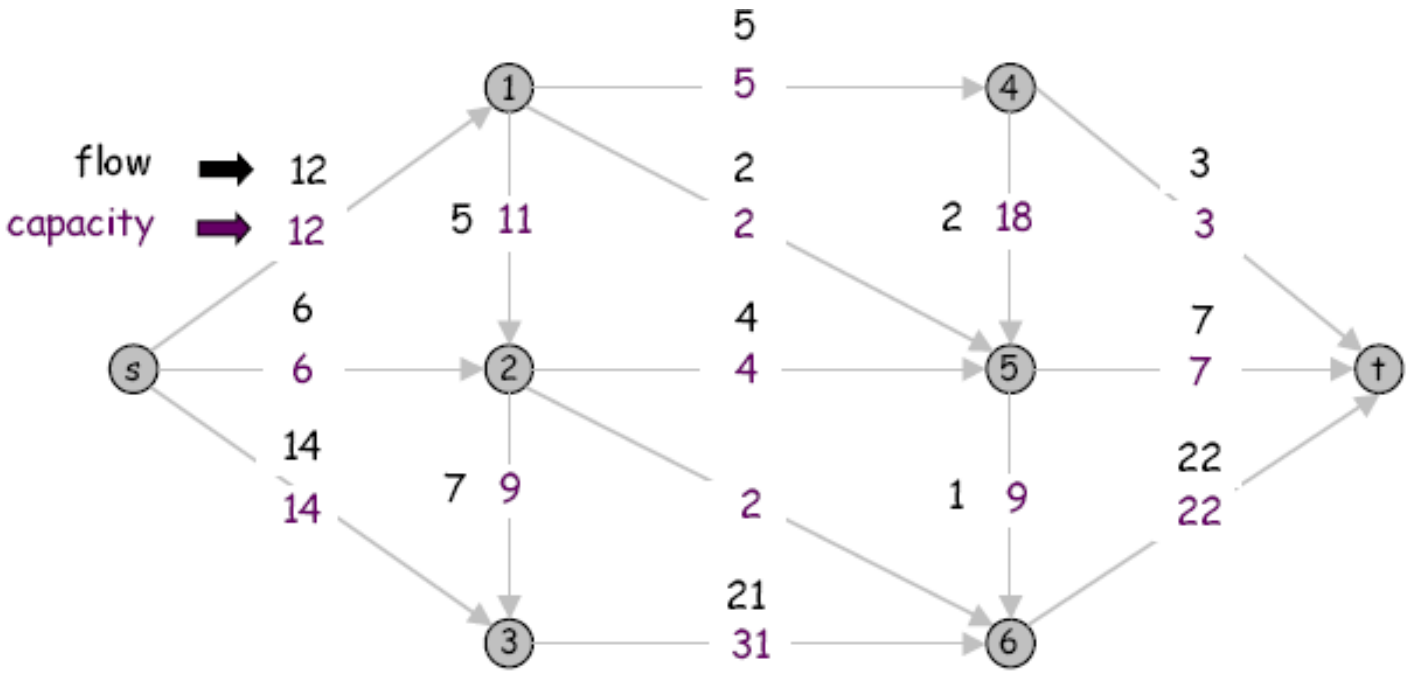


Starting to Find Min-Cost Flow

Start by finding a feasible (not necessarily minimum) flow, if one exists

Q: Can we do it using max flow?

A: Find max flow in a related network. There is feasible flow for the original network iff this maxflow saturates all new edges

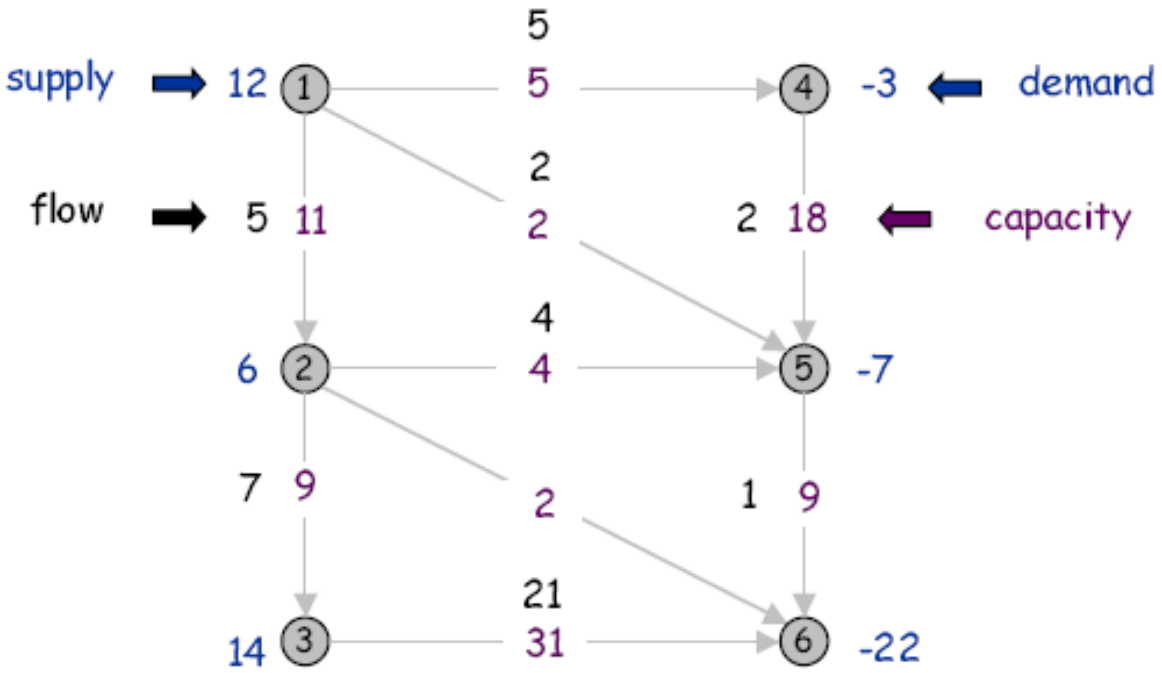


Starting to Find Min-Cost Flow

Start by finding a feasible (not necessarily minimum) flow, if one exists

Q: Can we do it using max flow?

A: Find max flow in a related network. There is feasible flow for the original network iff this maxflow saturates all new edges



Residual Graph

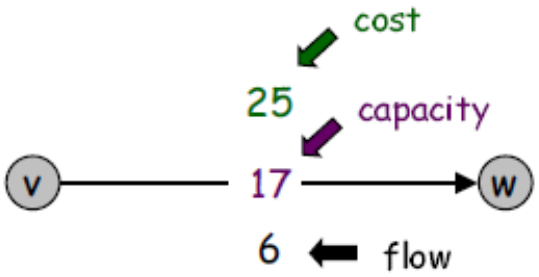
Similar to residual graph in max flow

Q: What should be the cost of reverse edges?

A: Negative of the original edge

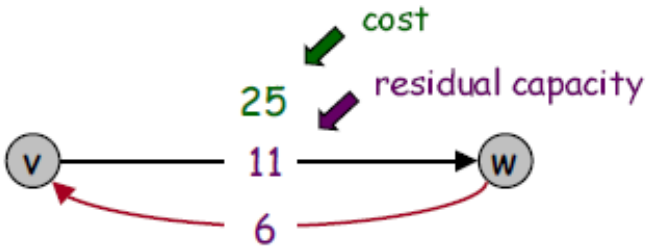
Original graph.

- Flow f(e)
- Arc e = v-w



Residual arcs

- v-w and w-v (allows to undo flow)
- Cost of w-v = -cost of v-w

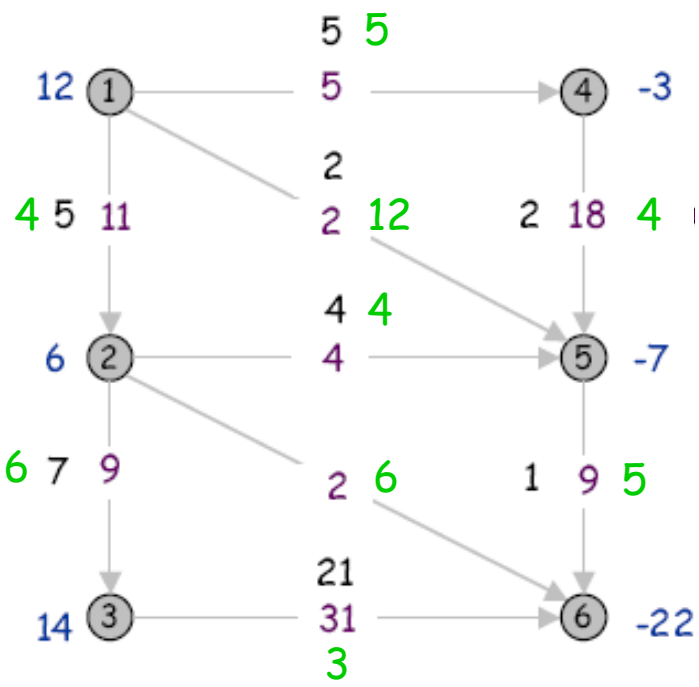


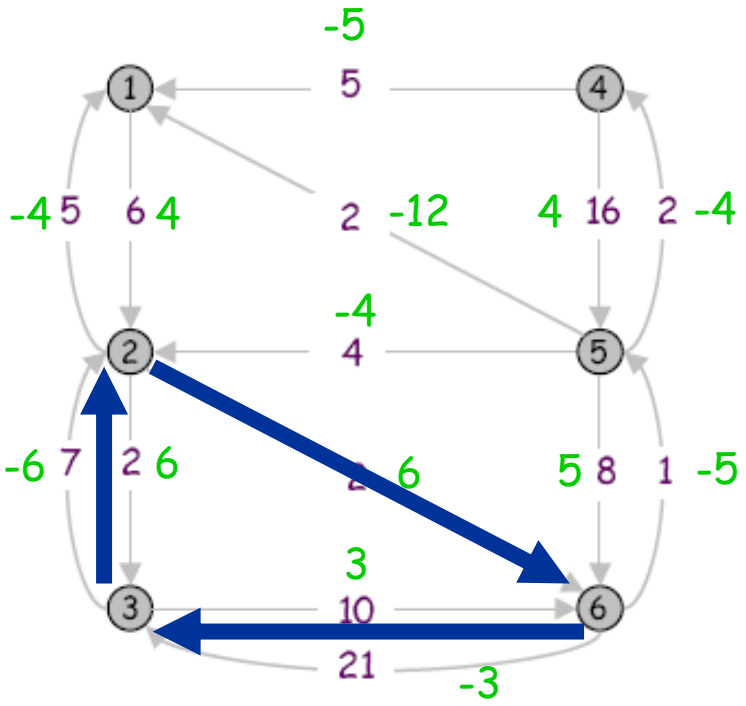
Improving the Flow

We already have a flow satisfying all supply/demands. Need to reduce its cost

Q: What happens if we augment flow along a path in residual graph?
A: Flow does not satisfy supply/demand anymore

Idea: Augment along a cycle (of negative residual cost)





Cycle Canceling Algorithm

Cycle canceling algorithm:

- Try to find a feasible flow via max-flow on a modified network. If cannot find, exit
- Repeat:
  - Try to find negative cost cycle K in residual network. If cannot, exit
  - Augment as much flow as possible on cycle K, subject to residual capacities

Theorem: If there is no negative cost cycle in the residual network, then the current flow is optimal.

So the algorithm always returns a min-cost flow

Q: How to find negative cycle?
A: Use Bellman-Ford

Cycle Canceling Algorithm: Analysis

Assume that all capacities, cost, demands are integral
Assume capacities are from 1,…,U, costs are from 1,…,C

As in the Ford-Fulkerson Algorithm, in the Cycle Canceling Algorithm we always maintain an integral flow, and the residual capacity/costs are also integral

So in each iteration the algorithm augments at least +1 along a cycle of cost <= -1

So the algorithm terminates after MAX-COST <= m.U.C iterations

Theorem: The Cycle Canceling Algorithm takes time $O(m.U.C.m.n) = O(m^2n.U.C)$

Bellman-Ford

Integrality Theorem

Integrality theorem: If capacities and demands are integers, there is always an integral min-cost flow

Again this is very useful when modeling applications (we just saw in the beginning for shortest path application)

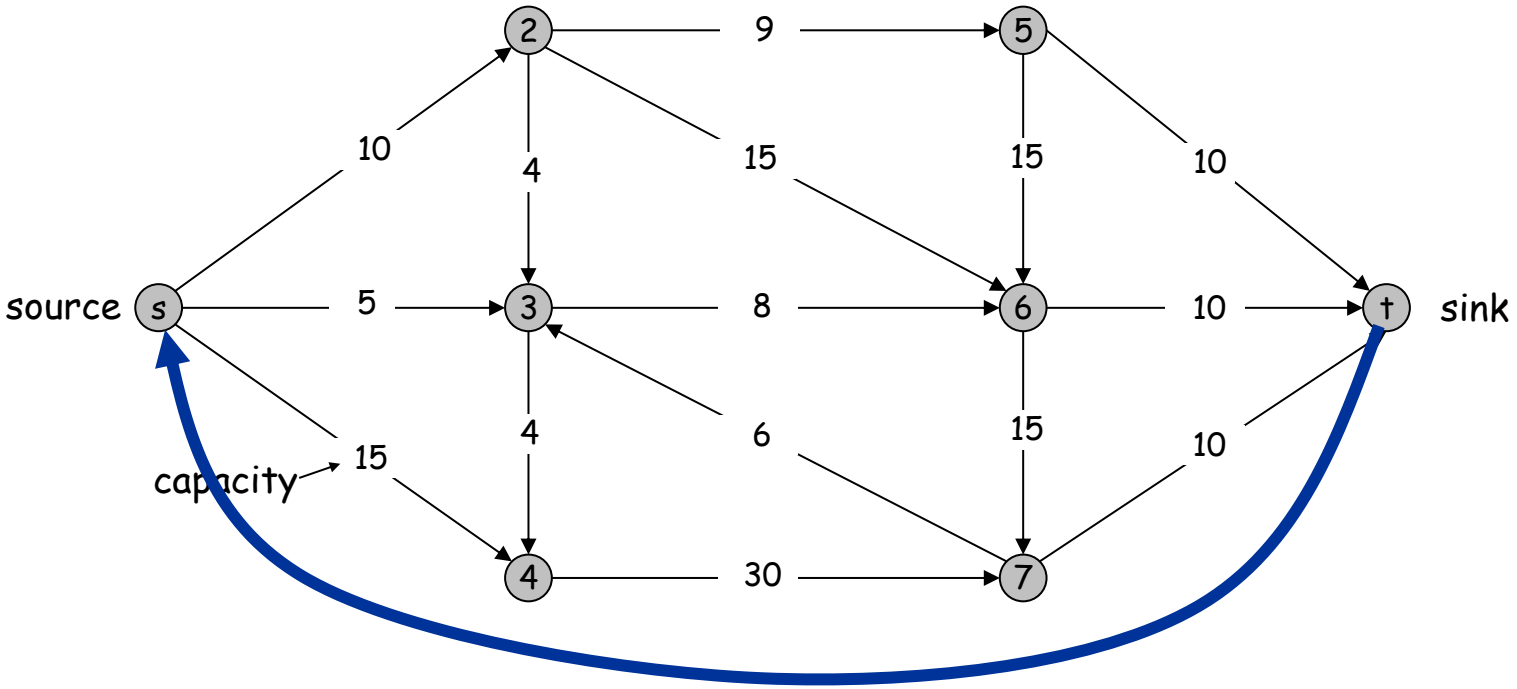
Best Algorithms

Year	Discoverer	Method	Big-Oh ~
1951	Dantzig	Network simplex	$E^2 V^2 U$
1960	Minty, Fulkerson	Out of kilter	$E V U$
1958	Jewell	Successive shortest path	$E V U$
1962	Ford-Fulkerson	Primal dual	$EV^2 U$
1967	Klein	Cycle canceling	$E^2 C U$
1972	Edmonds-Karp, Dinitz	Capacity scaling	$E^2 \log U$
1973	Dinitz-Gabow	Improved capacity scaling	$E V \log U$
1980	Rock, Bland-Jensen	Cost scaling	$E V^2 \log C$
1985	Tardos	ε-optimality	poly(E, V)
1988	Orlin	Enhanced capacity scaling	E^2

Modeling with Min-Cost Flows

Max-flow

Model the max-flow problem as a min-cost flow problem



Add edge from sink to source with negative cost and infinite capacity (put cost 0 on all other edges), all demands/supply equal to 0

Called circulation

Deciding Production Location

Exercise: Each year, Data Corporal produces as many as 400 computers in Boston and 300 computers in Raleigh. Los Angeles costumers must receive 400 computers, and Austin costumers must receive 300 computers.

Producing a computer in Boston costs $800 and in Raleigh costs $900. Shipping costs are the following (computers can be sent through Chicago):

From\to	Chicago	Austin	Los Angeles
Boston	80	220	280
Raleigh	100	140	170
Chicago	-	40	50

Formulate the problem of deciding the company's production to minimize the total cost as a min-cost flow problem, and solve it.