CSCI 3210:
Computational Game Theory

Network Flow

(Applications to Market)

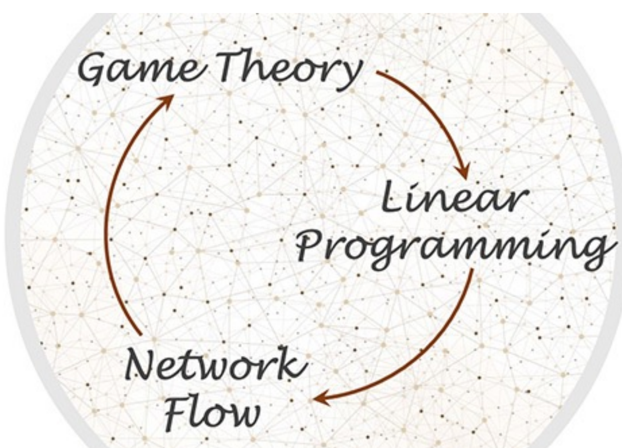
Ref: Ch 7 [Kleinberg-Tardos]
Ch 5 [AGT]

Mohammad T. Irfan

Web: www.bowdoin.edu/~mirfan

Many of the slides are adapted from Vazirani's and
Kleinberg-Tardos' textbooks.

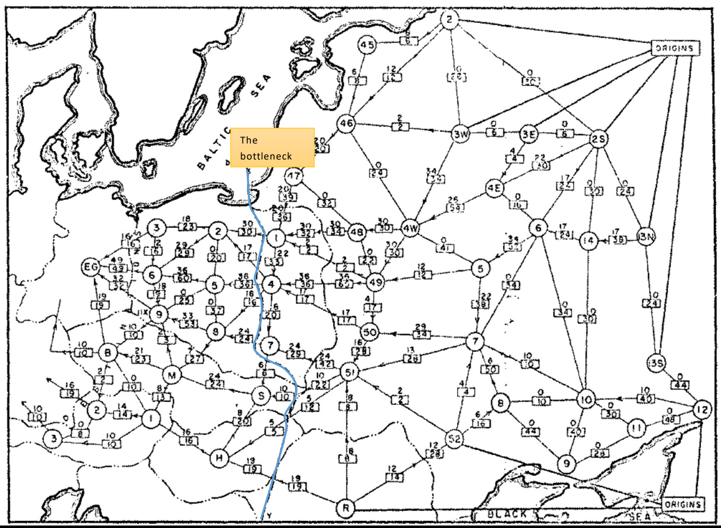
1



2

History: Schrijver (2002)

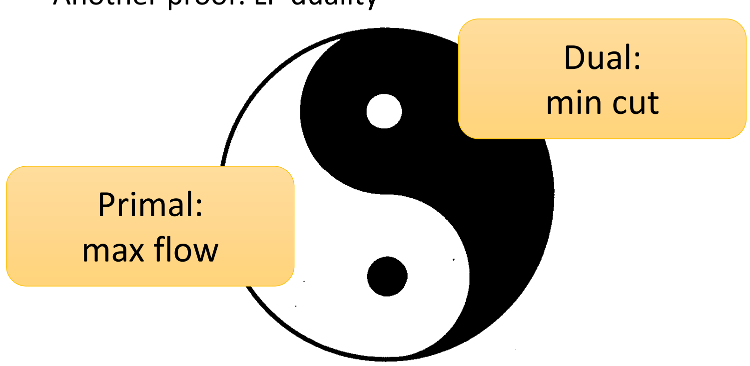
- http://homepages.cwi.nl/~lex/files/histtrpclean.pdf
- Soviet rail network: Tolstoy [1930] vs Harris and Ross [1955] (declassified 1999)



4

Big picture

- Tolstoy (1930): Find max flow
- Harris & Ross (1955): Find min cut
- Ford & Fulkerson (1956): They are the same
  - Their proof: combinatorial
  - Another proof: LP duality



5

Applications

Network	Nodes	Arcs	Flow
communication	telephone exchanges, computers, satellites	cables, fiber optics, microwave relays	voice, video, packets
circuits	gates, registers, processors	wires	current
mechanical	joints	rods, beams, springs	heat, energy
hydraulic	reservoirs, pumping stations, lakes	pipelines	fluid, oil
financial	stocks, currency	transactions	money
transportation	airports, rail yards, street intersections	highways, railbeds, airway routes	freight, vehicles, passengers
chemical	sites	bonds	energy

6

Applications

- **Fisher market**
- Network connectivity
- Bipartite matching
- Data mining
- Open-pit mining
- Airline scheduling
- Image processing
- Project selection
- Baseball elimination
- Network reliability
- Security of statistical data
- Distributed computing
- Egalitarian stable matching
- Distributed computing
- Many many more . . .

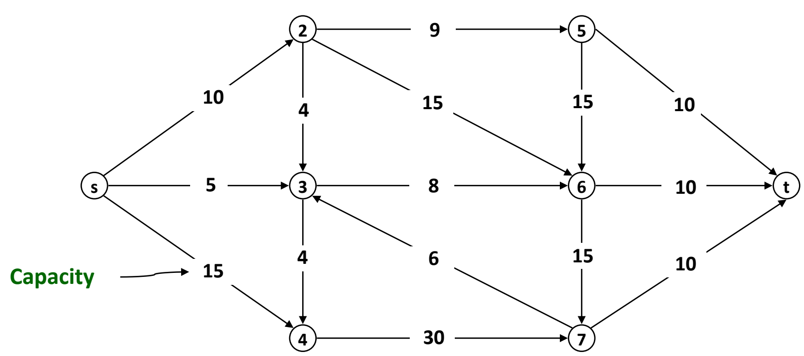
7

Max flow problem

8

Max flow network

- Directed graph (may have cycles)
- Two distinguished nodes: s = source, t = sink
- c(e) = capacity of arc e (integer)



9

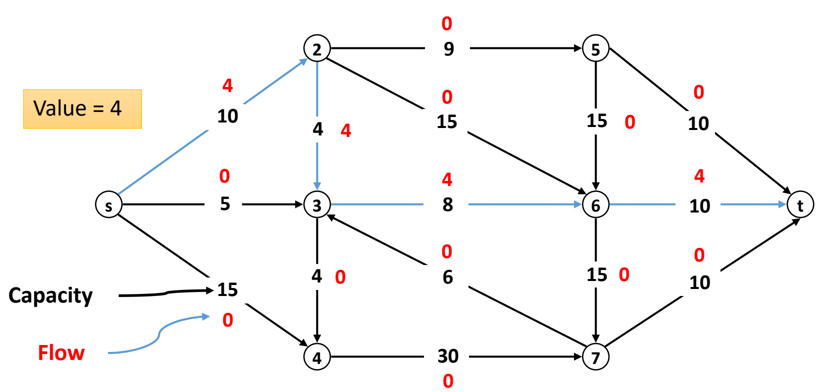
Definition: s-t flow

- Flow is an integer number >= 0 on each arc
- (Capacity)
  Flow on an arc can't exceed the arc's capacity
- (Conservation)
  flow in = flow out at any node ≠ s, t

10

Definition: flow value

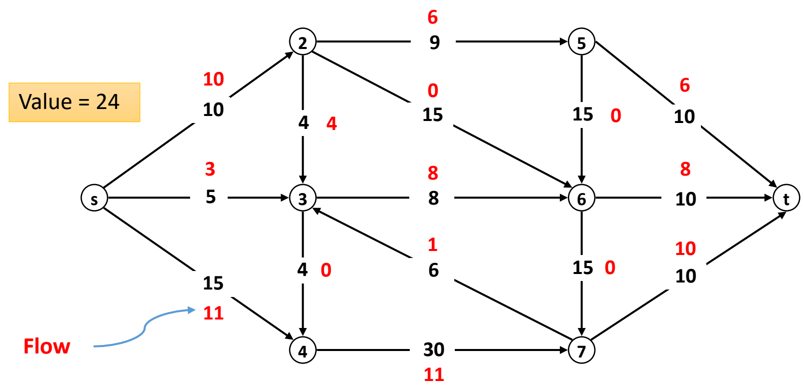
Flow value
= total flow into t = total flow out of s



11

Can we increase the flow value?

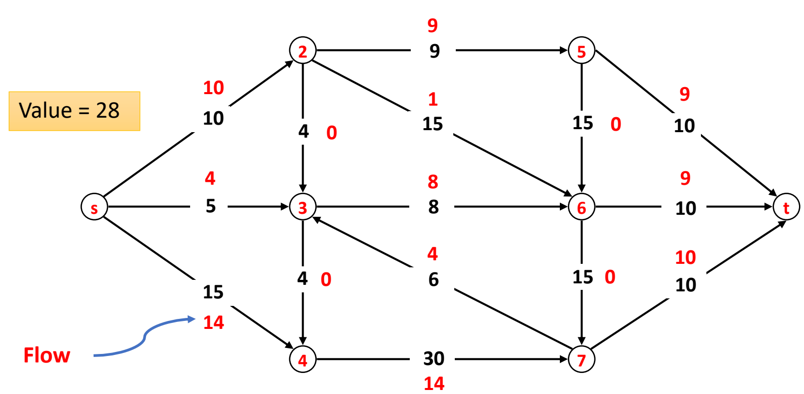
- Capacity and flow conservation constraints are satisfied
- Further increase in flow value?



12

Max flow problem

Compute an s-t flow of maximum flow value



13

Algorithms for max flow

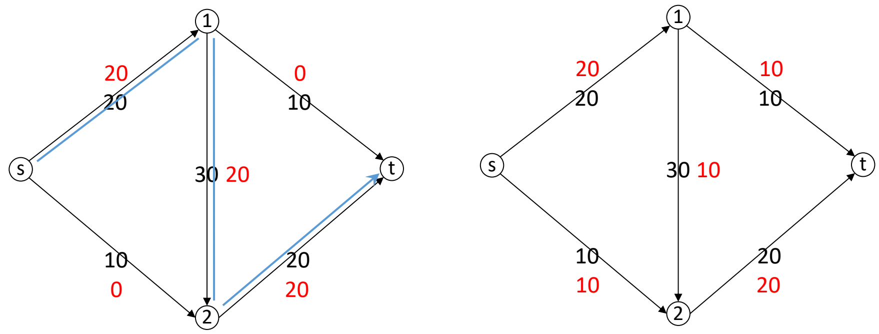
14

First try: greedy

- Start with f(e) = 0 for all arcs e
- Repeat until stuck:
  - Find an s-t path where each edge has f(e) **<** c(e)
  - Push more flow along that path

15

Greedy doesn't work– why?



greedy = 20

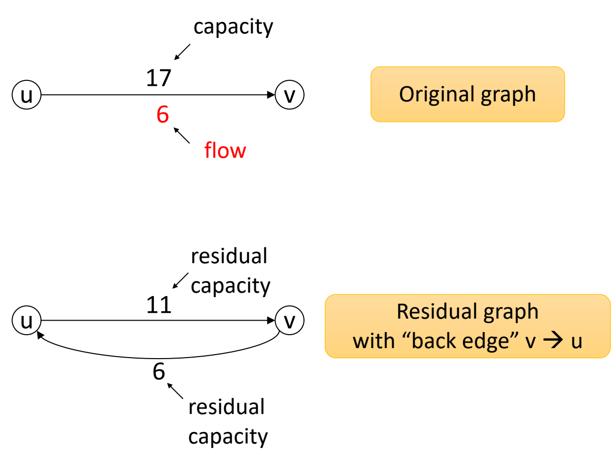
Then stuck

optimal = 30

16

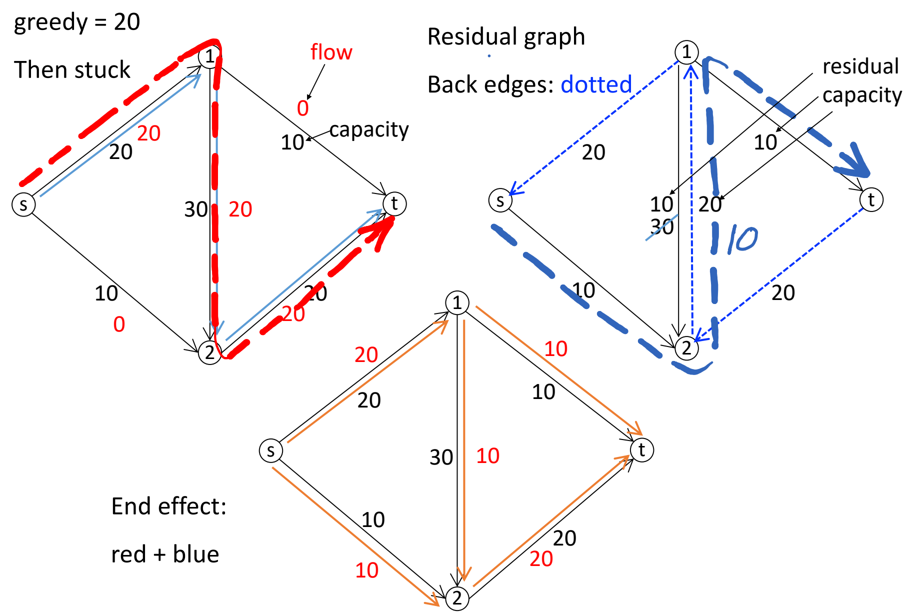
Fix: residual graph

A way of undoing previous flows



17

Residual graph: interpretation



18

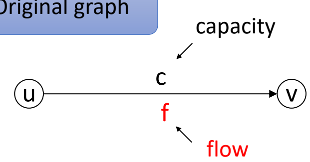
Ford-Fulkerson algorithm

- Iteratively find s-t paths that admit more flow in the residual graph
  - Such s-t paths: augmenting paths
- Push more flow along augmenting paths
  - Prove: It's a valid flow
- No further augmenting path?
  - Optimal solution!

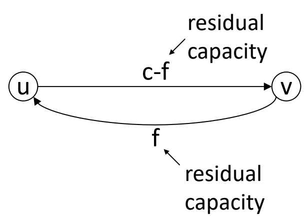
19

Prove: augmenting path gives valid flow

Original graph



Residual graph
with “back edge” v->u



2 cases:

1. Augmenting path uses u→v:
   - The original capacity is not violated: new flow on u→v ≤ c
   - Flow in = flow out at u and v

2. Augmenting path uses back edge v→u:
   - New flow on u→v will not be < 0
   - New flow on u→v ≤ c
   - Flow in = flow out at u and v

20

Ford-Fulkerson Demo

21

Running time of Ford-Fulkerson

n = # of nodes

m = # of edges

C = max capacity of any edge

- At most nC iterations
- Total running time: O(mnC)
- Not strongly polynomial
  - There are strongly polynomial algorithms

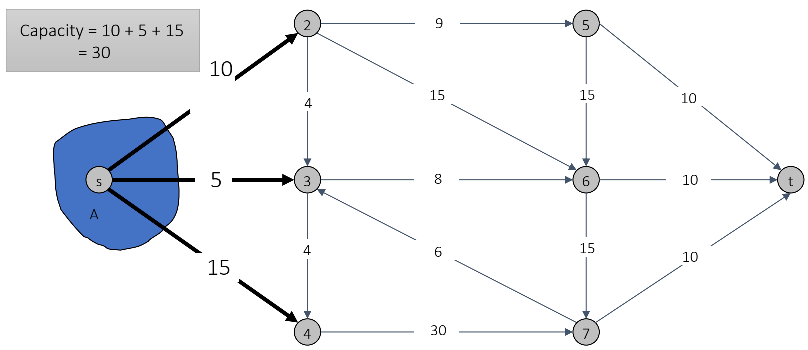
22

Min cut problem

23

s-t cut

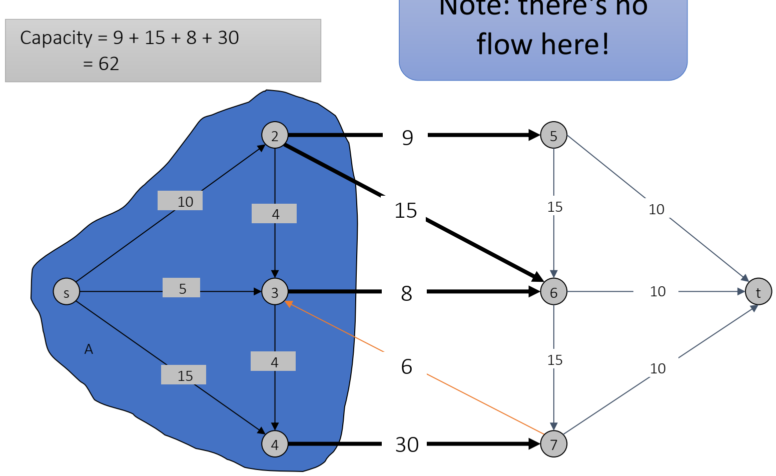
- Partition the nodes into two sets A and B such that s is in A and t is in B
- (A, B) is called an s-t cut
- Capacity of s-t cut (A, B)
  cap(A, B) = sum of capacities of arcs out of A



24

s-t cut: more example

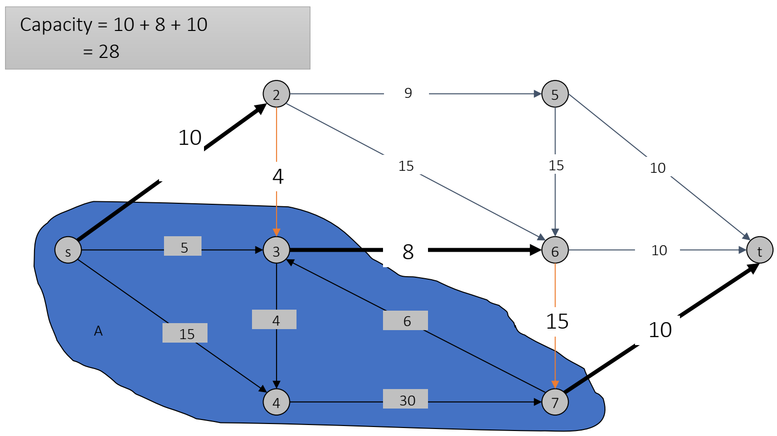
Note: there's no flow here!



25

Min cut problem

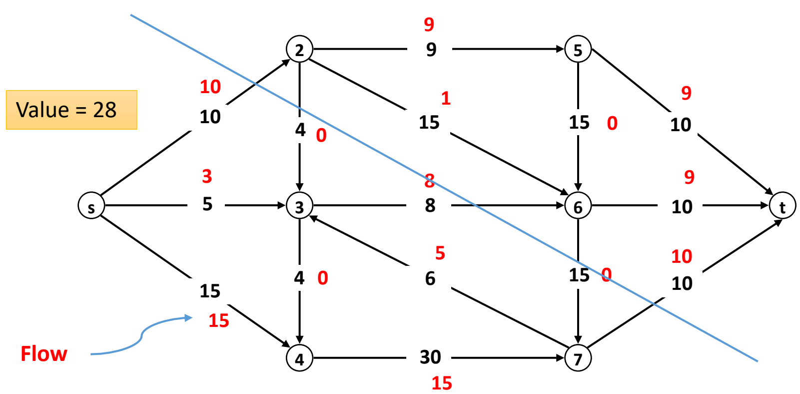
Find an s-t cut of minimum capacity



26

Max flow solution

- Max flow value is also 28!



27

Max flow vs. min cut

28

LP formulation: max flow

flow

Maximize $$v(f) = \sum_{e \text{ out of } s} f(e)$$

capacity

Subject to

$$0 \leq f(e) \leq c(e), \quad \forall e$$

$$\sum_{e \text{ into } v} f(e) = \sum_{e \text{ out of } v} f(e), \quad \forall v \text{ except } s, t$$

Integrality theorem: if all capacities are integers, then there exists a max flow with all integer flows.

29

LP formulation: min cut

- Dual of max flow
- Weak duality: any flow <= any cut capacity
  - Proof without using LP duality:
  - Consider any cut (A, B)

    Any flow
    = sum of flows out of A – sum of flows into A [why?]
    <= sum of flows out of A
    <= sum of edge capacities out of A
    = cut capacity of (A, B)
- Strong duality: max flow = min cut capacity
  - Ford-Fulkerson's proof without using LP duality

30

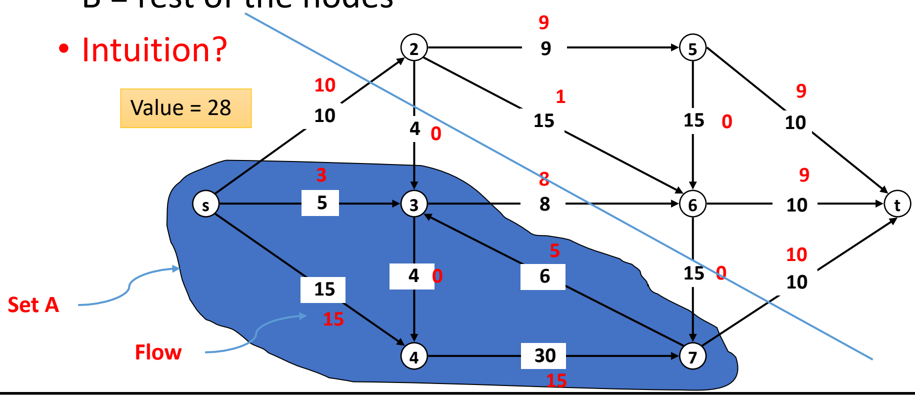
Max-flow min-cut theorem

- Ford & Fulkerson (1956)
- In any network, the value of the max flow is equal to the value of the min cut.

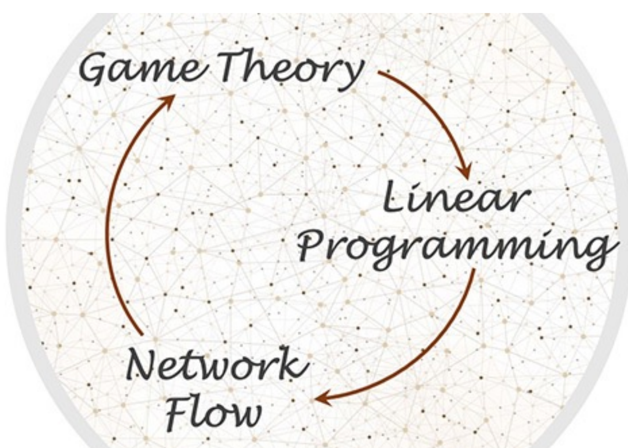
32

How to: max flow → min cut

- Given max flow, find a cut of min capacity
- A = {s and all nodes reachable from s in the final residual graph }
- B = rest of the nodes
- Intuition?



33



34