MAX FLOW APPLICATIONS

David Kauchak
CS140 - Fall 2022

1

Admin

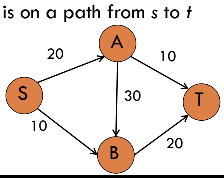
Checkpoint on Thursday

Assignment 10 due Tuesday 11/22

2

Flow graph/networks

Flow network
- directed, weighted graph (V, E)
- positive edge weights indicating the "capacity" (generally, assume integers)
- contains a single source $s \in V$ with no incoming edges
- contains a single sink/target $t \in V$ with no outgoing edges
- every vertex is on a path from s to t



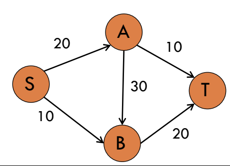
3

Flow constraints

in-flow = out-flow for every vertex (except s, t)

flow along an edge cannot exceed the edge capacity

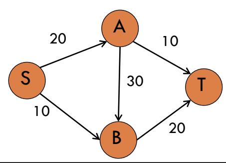
flows are positive



4

Max flow problem

Given a flow network: *what is the maximum flow we can send from s to t that meet the flow constraints?*



5

Network flow properties

If one of these is true then all are true (i.e. each implies the the others):

f is a maximum flow

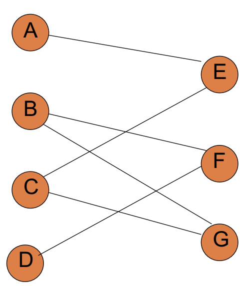
G_f (residual graph) has no paths from s to t

|f| = minimum capacity cut

6

Application: bipartite graph matching

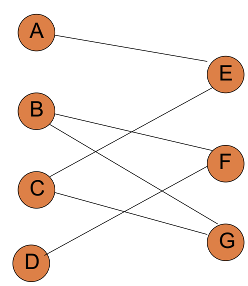
Bipartite graph – a graph where every vertex can be partitioned into two sets X and Y such that all edges connect a vertex u ∈ X and a vertex v ∈ Y



8

Application: bipartite graph matching

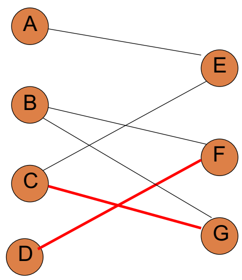
A *matching* M is a subset of edges such that each node occurs **at most once** in M



9

Application: bipartite graph matching

A *matching* M is a subset of edges such that each node occurs **at most once** in M

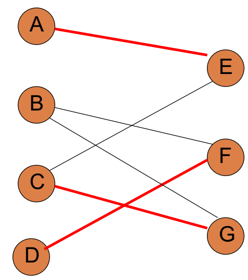


matching

10

Application: bipartite graph matching

A *matching* M is a subset of edges such that each node occurs **at most once** in M

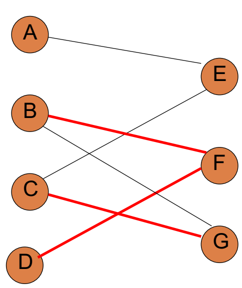


matching

11

Application: bipartite graph matching

A *matching* M is a subset of edges such that each node occurs **at most once** in M

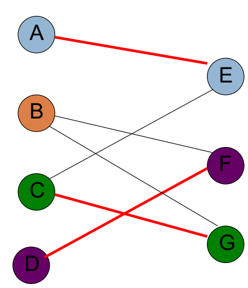


not a matching

12

Application: bipartite graph matching

A *matching* can be thought of as pairing the vertices



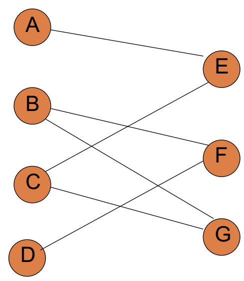
13

Application: bipartite graph matching

Bipartite matching problem: find the *largest* matching in a bipartite graph

Where might this problem come up?

- CS department has n courses and m faculty
- Every instructor can teach *some* of the courses
- What course should each person teach?
- Anytime we want to match n things with m, but not all things can match



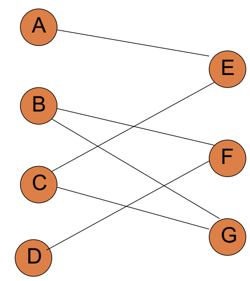
14

Application: bipartite graph matching

Bipartite matching problem: find the *largest* matching in a bipartite graph

ideas?

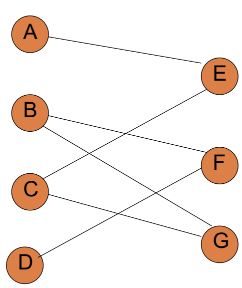
- greedy?
- dynamic programming?



15

Application: bipartite graph matching

Setup as a flow problem:

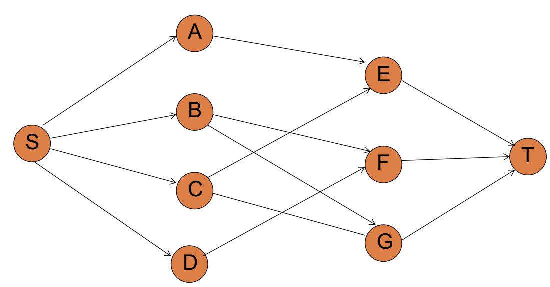


16

Application: bipartite graph matching

Setup as a flow problem:

edge weights?

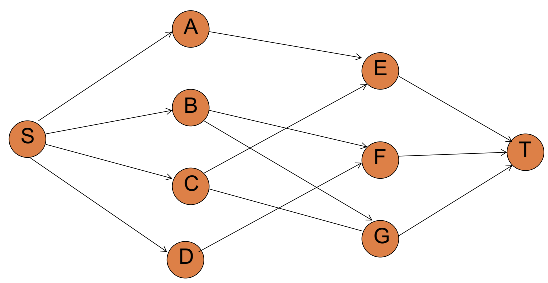


17

Application: bipartite graph matching

Setup as a flow problem:

all edge weights are 1

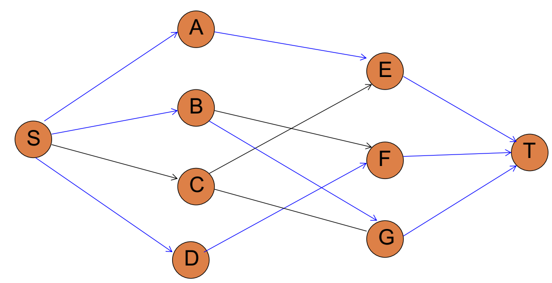


18

Application: bipartite graph matching

Setup as a flow problem:

after we find the flow, how do we find the matching?

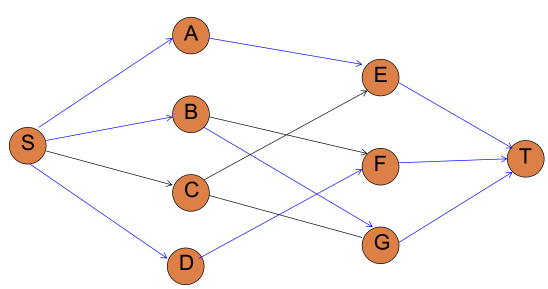


19

Application: bipartite graph matching

Setup as a flow problem:

match those nodes with flow between them



20

Application: bipartite graph matching

Is it correct?

Assume it's not

- there is a better matching
- because of how we setup the graph flow = # of matches
- therefore, the better matching would have a higher flow
- contradiction (max-flow algorithm finds maximal!)

21

Application: bipartite graph matching

Run-time?

Cost to build the flow?

- O(E)
  - each existing edge gets a capacity of 1
  - introduce V new edges (to and from s and t)
  - V is O(E) (for non-degenerate bipartite matching problems)

Max-flow calculation?

- Basic Ford-Fulkerson: O(max-flow * E)
- Edmunds-Karp: $O(V E^2)$
- Preflow-push: $O(V^3)$

22

Application: bipartite graph matching

Run-time?

Cost to build the flow?

- O(E)
  - each existing edge gets a capacity of 1
  - introduce V new edges (to and from s and t)
  - V is O(E) (for non-degenerate bipartite matching problems)

Max-flow calculation?

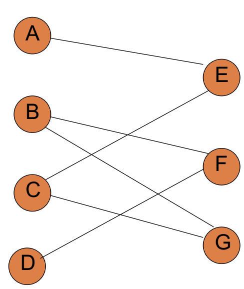
- Basic Ford-Fulkerson: O(max-flow * E)
  - max-flow = O(V)
  - O(V E)

23

Application: bipartite graph matching

Bipartite matching problem: find the *largest* matching in a bipartite graph

- CS department has n courses and m faculty
- Every instructor can teach *some* of the courses
- What course should each person teach?
- Each faculty can teach at most 3 courses a semester?



Change the s edge weights (representing faculty) to 3

24

Survey Design

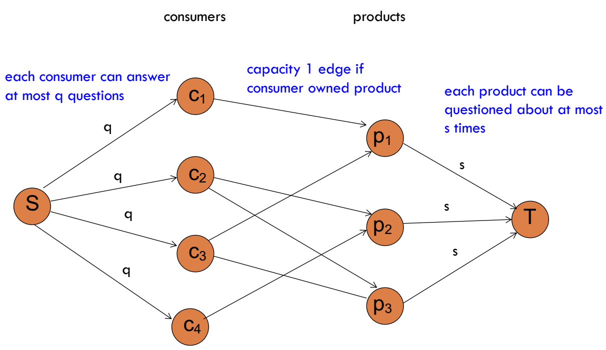
Design a survey with the following requirements:

- Design survey asking *n* consumers about *m* products
- Can only survey consumer about a product if they own it
- Question consumers about at most *q* products
- Each product should be surveyed at most *s* times
- Maximize the number of surveys/questions asked

How can we do this?

25

Survey Design



26

Survey design

Is it correct?

- Each of the comments above the flow graph match the problem constraints
- max-flow finds the maximum matching, given the problem constraints

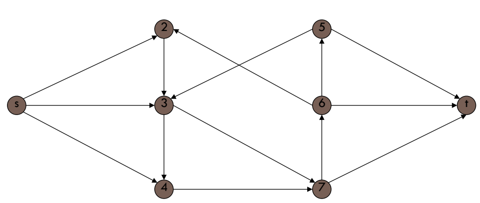
What is the run-time?

- Basic Ford-Fulkerson: O(max-flow * E)
- Edmunds-Karp: $O(V E^2)$
- Preflow-push: $O(V^3)$

27

Edge Disjoint Paths

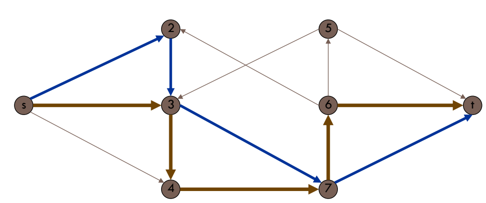
Two paths are edge-disjoint if they have no edge in common



28

Edge Disjoint Paths

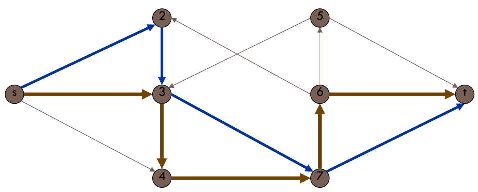
Two paths are edge-disjoint if they have no edge in common



29

Edge Disjoint Paths Problem

Given a directed graph G = (V, E) and two nodes s and t, find the max number of edge-disjoint paths from s to t



Why might this be useful?

30

Edge Disjoint Paths Problem

Given a directed graph G = (V, E) and two nodes s and t, find the max number of edge-disjoint paths from s to t

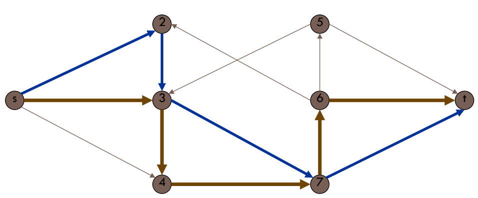
Why might this be useful?

- edges are unique resources (e.g. communications, transportation, etc.)
- how many *concurrent (non-conflicting)* paths do we have from s to t

31

Edge Disjoint Paths

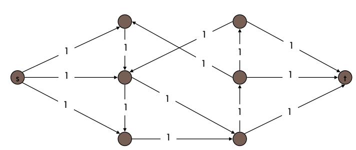
Algorithm ideas?



32

Edge Disjoint Paths

Max flow formulation: assign unit capacity to every edge

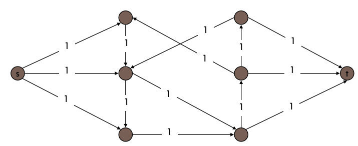


What does the max flow represent?
Why?

33

Edge Disjoint Paths

Max flow formulation: assign unit capacity to every edge

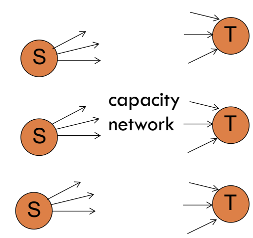


- max-flow = maximum number of disjoint paths
- correctness:
  - each edge can have at most flow = 1, so can only be traversed once
  - therefore, each unit out of s represents a separate path to t

34

Max-flow variations

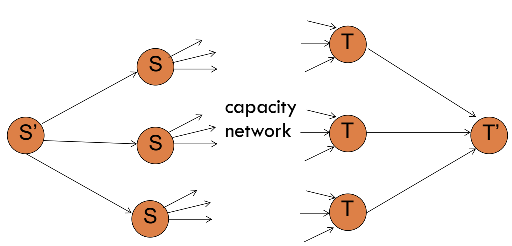
What if we have multiple sources and multiple sinks (e.g. the Russian train problem has multiple sinks)?



35

Max-flow variations

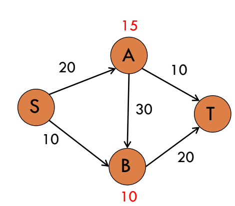
Create a new source and sink and connect up with infinite capacities…



36

Max-flow variations

Vertex capacities: in addition to having edge capacities we can also restrict the amount of flow through each vertex

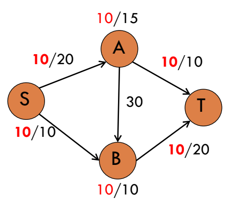


What is the max-flow now?

37

Max-flow variations

Vertex capacities: in addition to having edge capacities we can also restrict the amount of flow through each vertex

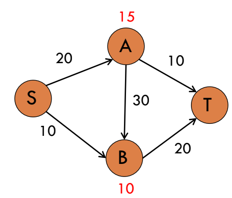


20 units

38

Max-flow variations

Vertex capacities: in addition to having edge capacities we can also restrict the amount of flow through each vertex



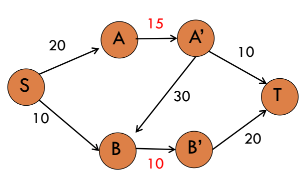
How can we solve this problem?

39

Max-flow variations

For each vertex v

- create a new node v'
- create an edge with the vertex capacity from v to v'
- move all outgoing edges from v to v'



Can you now prove it's correct?

40

Max-flow variations

Proof:

1. show that if a solution exists in the original graph, then a solution exists in the modified graph
2. show that if a solution exists in the modified graph, then a solution exists in the original graph

41

Max-flow variations

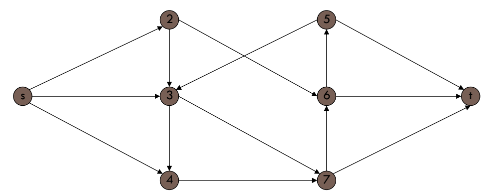
Proof:

- we know that the vertex constraints are satisfied
  - no incoming flow can exceed the vertex capacity since we have a single edge with that capacity from v to v'
- we can obtain the solution, by collapsing each v and v' back to the original v node
  - in-flow = out-flow since there is only a single edge from v to v'
  - because there is only a single edge from v to v' and all the in edges go in to v and out to v', they can be viewed as a single node in the original graph

42

More problems: maximum independent path

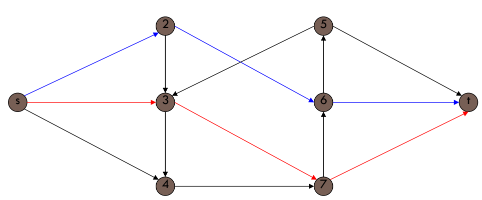
Two paths are independent if they have no *vertices* in common



43

More problems: maximum independent path

Two paths are independent if they have no *vertices* in common

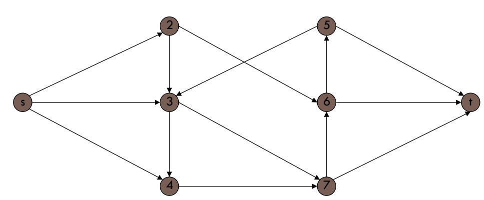


44

More problems: maximum independent path

Find the maximum number of independent paths

Ideas?

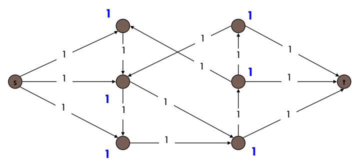


45

maximum independent path

Max flow formulation:

- assign unit capacity to every edge (though any value would work)
- assign unit capacity to every vertex



Same idea as the maximum edge-disjoint paths, but now we also constrain the vertices

46