# COMP 355
# Advanced Algorithms

**Extensions of Network Flows**
**Section 7.7- 7.12(KT)**

Rhodes College
1848

1

## Applications of Max-Flow

Huge number of applications

- Bipartite Matching
- Perfect Matching
- Disjoint Paths

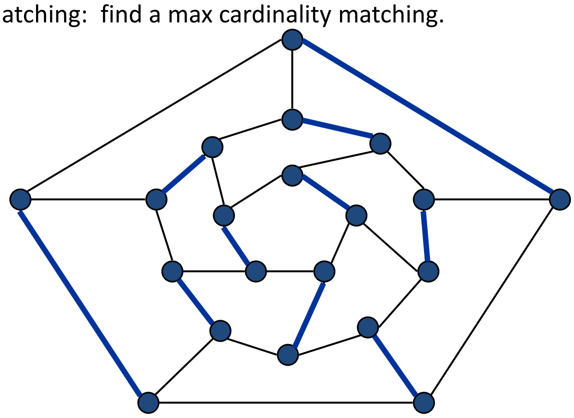
Rhodes College
1848

2

# Matching

Matching.

- Input: undirected graph G = (V, E).
- $M \subseteq E$ is a matching if each node appears in at most one edge in M.
- Max matching: find a max cardinality matching.

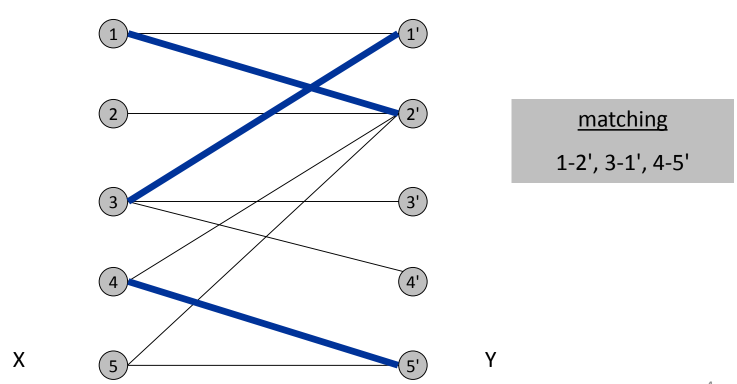




# Bipartite Matching

Bipartite matching.

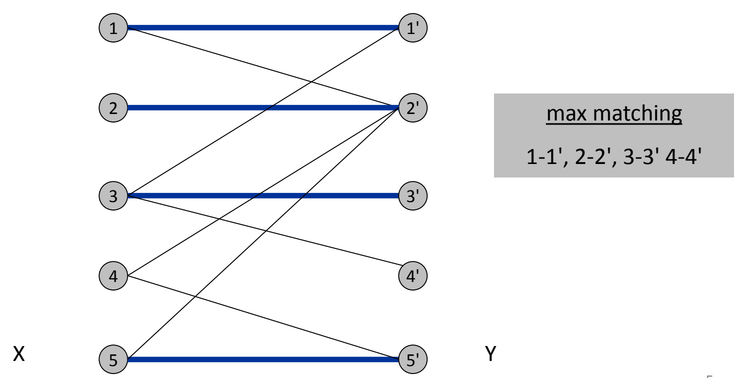
- Input: undirected, bipartite graph $G = (X \cup Y, E)$.
- $M \subseteq E$ is a matching if each node appears in at most one edge in M.
- Max matching: find a max cardinality matching.



# Bipartite Matching

Bipartite matching.

- Input: undirected, bipartite graph G = (X ∪ Y, E).
- M ⊆ E is a matching if each node appears in at most edge in M.
- Max matching: find a max cardinality matching.

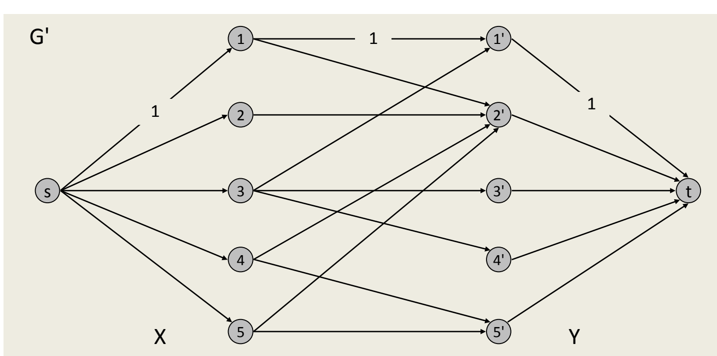



5

# Bipartite Matching

Max flow formulation.

- Create digraph G' = (X ∪ Y ∪ {s, t}, E' ).
- Direct all edges from X to Y, and assign unit (or infinite) capacity.
- Add source s, and unit capacity edges from s to each node in X.
- Add sink t, and unit capacity edges from each node in Y to t.



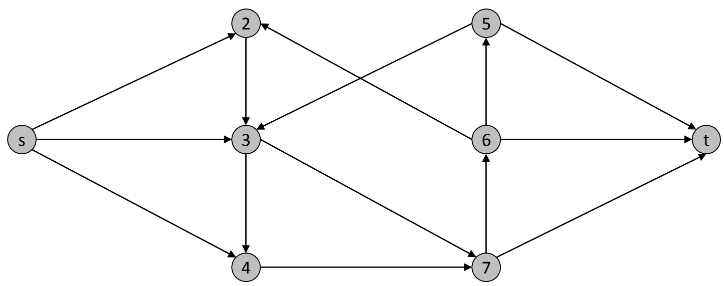

6

# Edge Disjoint Paths

Disjoint path problem. Given a digraph G = (V, E) and two nodes s and t, find the max number of edge-disjoint s-t paths.

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

Ex: communication networks.



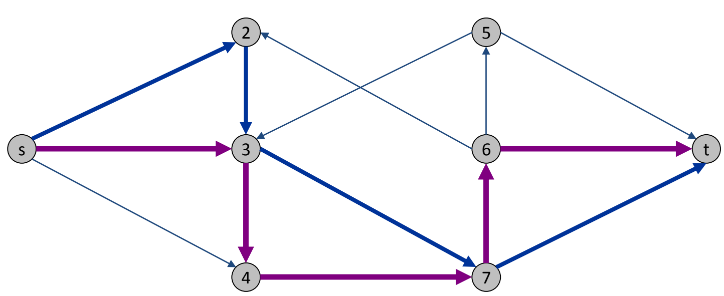

7

Edge Disjoint Paths

Disjoint path problem. Given a digraph G = (V, E) and two nodes s and t, find the max number of edge-disjoint s-t paths.

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

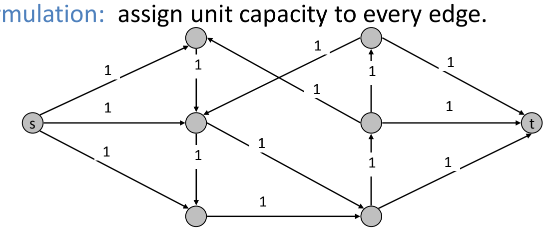
Ex: communication networks.




8

# Edge Disjoint Paths

Max flow formulation: assign unit capacity to every edge.




Theorem. Max number edge-disjoint s-t paths equals max flow value.

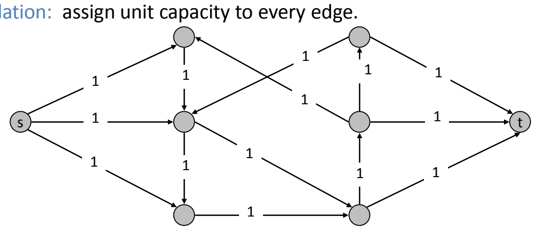
Pf. $\leq$

- Suppose there are k edge-disjoint paths $P_1, \ldots, P_k$.
- Set f(e) = 1 if e participates in some path $P_i$ ; else set f(e) = 0.
- Since paths are edge-disjoint, f is a flow of value k. ▪

9

# Edge Disjoint Paths

Max flow formulation: assign unit capacity to every edge.




Theorem. Max number edge-disjoint s-t paths equals max flow value.

Pf. $\geq$

- Suppose max flow value is k.
- Integrality theorem $\Rightarrow$ there exists 0-1 flow f of value k.
- Consider edge (s, u) with f(s, u) = 1.
  - by conservation, there exists an edge (u, v) with f(u, v) = 1
  - continue until reach t, always choosing a new edge
- Produces k (not necessarily simple) edge-disjoint paths. ▪

can eliminate cycles to get simple paths if desired

10

# Extensions of Network Flow

Network flow - useful in a wide variety of applications

We will discuss two useful extensions to the network flow problem.

- Both can be reduced to network flow
- Single algorithm will solve them both.

Many computational problems that would seem to have little to do with flow of fluids through networks can be expressed as one of these two extended versions.

Rhodes College

11

# Circulation with Demands

Circulation with demands.

- Directed graph G = (V, E).
- Edge capacities c(e), e $\in$ E.
- Node supply and demands d(v), v $\in$ V.

demand if d(v) > 0; supply if d(v) < 0; transshipment if d(v) = 0

Def. A circulation is a function that satisfies:

- For each e $\in$ E: $0 \leq f(e) \leq c(e)$ (capacity)
- For each v $\in$ V: $\sum_{e \text{ in to } v} f(e) - \sum_{e \text{ out of } v} f(e) = d(v)$ (conservation)

Circulation problem: Given (V, E, c, d), does there exist a circulation?

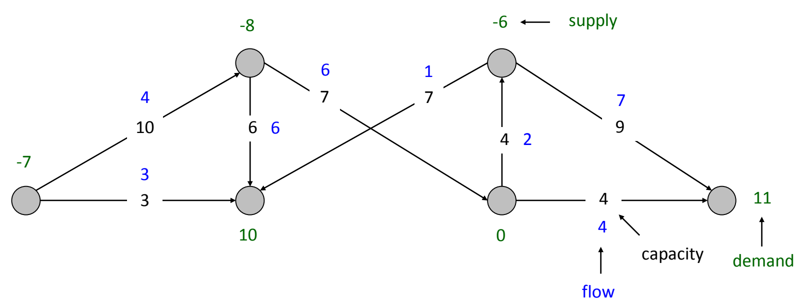
12

# Circulation with Demands

Necessary condition: sum of supplies = sum of demands.

$$\sum_{v\,:\,d(v)>0} d(v) \;=\; \sum_{v\,:\,d(v)<0} -d(v) \;=:\; D$$

Pf. Sum conservation constraints for every demand node v.

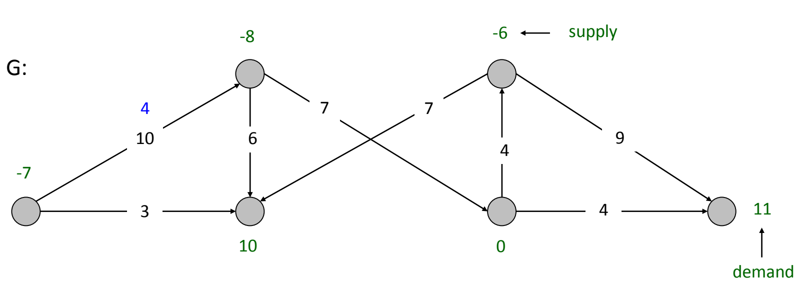



13

# Circulation with Demands

Max flow formulation.

G:



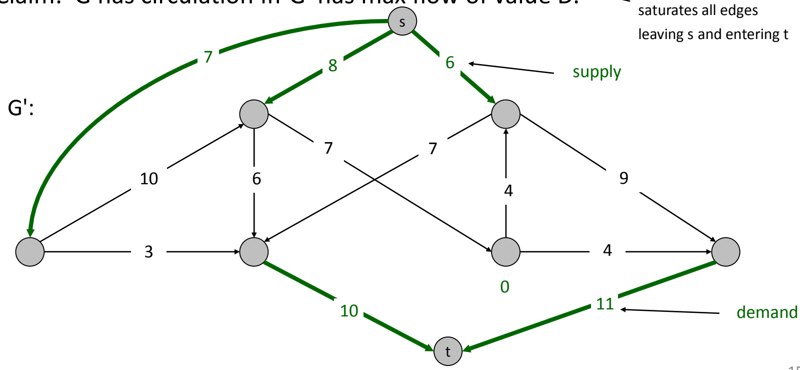

14

# Circulation with Demands

Max flow formulation.

- Add new source s and sink t.
- For each v with d(v) < 0, add edge (s, v) with capacity -d(v).
- For each v with d(v) > 0, add edge (v, t) with capacity d(v).
- Claim: G has circulation iff G' has max flow of value D.

saturates all edges leaving s and entering t




15

# Circulation with Demands

Integrality theorem. If all capacities and demands are integers, and there exists a circulation, then there exists one that is integer-valued.

Pf. Follows from max flow formulation and integrality theorem for max flow.

Characterization. Given (V, E, c, d), there does not exists a circulation iff there exists a node partition (A, B) such that $\Sigma_{v\in B}\, d_v > \text{cap}(A, B)$

Pf idea. Look at min cut in G'.

demand by nodes in B exceeds supply of nodes in B plus max capacity of edges going from A to B

16

# Circulation with Demands and Lower Bounds

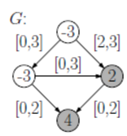
Feasible circulation.

- Directed graph G = (V, E).
- Edge capacities c(e) and lower bounds $\ell$ (e), e $\in$ E.
- Node supply and demands d(v), v $\in$ V.

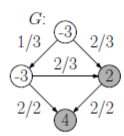
Def. A circulation is a function that satisfies:

- For each e $\in$ E: $\ell(e) \leq f(e) \leq c(e)$ (capacity)
- For each v $\in$ V: $\sum_{e \text{ in to } v} f(e) - \sum_{e \text{ out of } v} f(e) = d(v)$ (conservation)

Circulation problem with lower bounds. Given (V, E, $\ell$, c, d), does there exists a circulation?




Initial network (edges labeled with $[\ell, c]$ bounds)



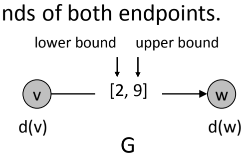

A valid flow

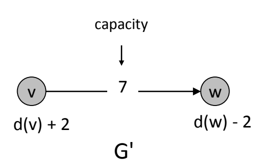


# Circulation with Demands and Lower Bounds

Idea. Model lower bounds with demands.

- Send $\ell$(e) units of flow along edge e.
- Update demands of both endpoints.




Theorem. There exists a circulation in G iff there exists a circulation in G'. If all demands, capacities, and lower bounds in G are integers, then there is a circulation in G that is integer-valued.

Pf sketch. f(e) is a circulation in G iff f'(e) = f(e) - $\ell$(e) is a circulation in G'.

# Survey Design

Survey design.

- Design survey asking $n_1$ consumers about $n_2$ products.
- Can only survey consumer i about a product j if they own it.
- Ask consumer i between $c_i$ and $c_i'$ questions.
- Ask between $p_j$ and $p_j'$ consumers about product j.

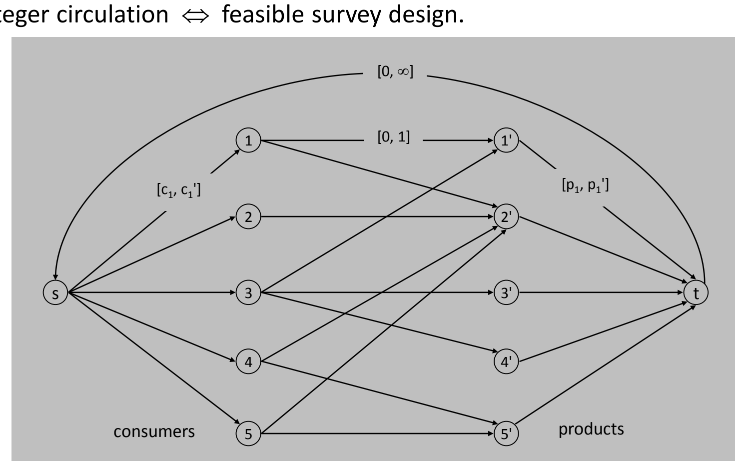
Goal. Design a survey that meets these specs, if possible.



# Survey Design

Algorithm. Formulate as a circulation problem with lower bounds.

- Include an edge (i, j) if customer own product i.
- Integer circulation $\Leftrightarrow$ feasible survey design.



# Practice (Chapter 7 (KT), Exercise 7)

Consider a set of mobile computing clients in a certain town who each need to be connected to one of several possible *base stations*. We'll suppose there are $n$ clients, with the position of each client specified by its $(x, y)$ coordinates in the plane. There are also $k$ base stations; the position of each of these is specified by $(x, y)$ coordinates as well.

For each client, we wish to connect it to exactly one of the base stations. Our choice of connections is constrained in the following ways.

- Range parameter – $r$ – a client can only be connected to a base station that is within distance r.
- Load parameter – $L$ – no more than L clients can be connected to any single base station.

Design a polynomial-time algorithm to determine whether every client can be connected simultaneously to a base station, subject to the range and load conditions above.