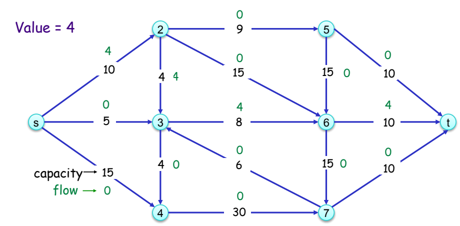
Objectives

- Network Flow Applications
  - Bipartite Matching
  - Circulation
  - Survey Design
  - Airline Scheduling

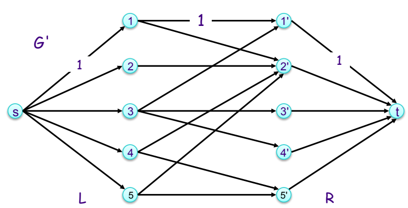
Review: Flows

- The value of a flow f is $v(f) = \sum_{e \text{ out of } s} f(e)$



Bipartite Graph: Max Flow Formulation

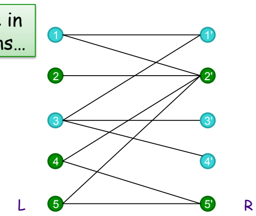
- Create digraph G' = (L ∪ R ∪ {s, t}, E')
- Direct all edges from L to R, and assign unit capacity
- Add source s, and unit capacity edges from s to each node in L
- Add sink t, and unit capacity edges from each node in R to t



Marriage Theorem [Frobenius 1917, Hall 1935]

- Let G = (L ∪ R, E) be a bipartite graph with |L| = |R|. Then, G has a perfect matching iff |Γ (S)| ≥ |S| for *all subsets* S ⊆ L.

Need to prove in both directions…



No perfect matching:
S = { 2, 4, 5 }
Γ(S) = { 2', 5' }.

Review: Power of Max Flow Problem

- Some problems with non-trivial combinatorial searches can be formulated as max flow or min cut in a directed graph

Review: Circulation with Demands

- Circulation with demands
  - Directed graph G = (V, E)
  - Edge capacities c(e), e ∈ E
  - Node supply and demands d(v), v ∈ V

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

Review: Circulation with Demands

- Circulation with demands
  - Directed graph G = (V, E)
  - Edge capacities c(e), e ∈ E
  - Node supply and demands d(v), v ∈ 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 ∈ E: $0 \le f(e) \le c(e)$ (capacity)
  - For each v ∈ V: $\sum_{e \text{ in to } v} f(e) - \sum_{e \text{ out of } v} f(e) = d(v)$ (conservation)

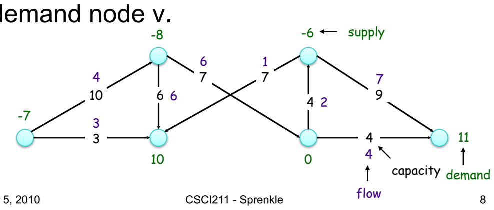
Circulation problem: given (V, E, c, d), does there exist a circulation? (Can we satisfy demand with supply?)

Review: Circulation with Demands

- Necessary condition: sum of supplies = sum of demands

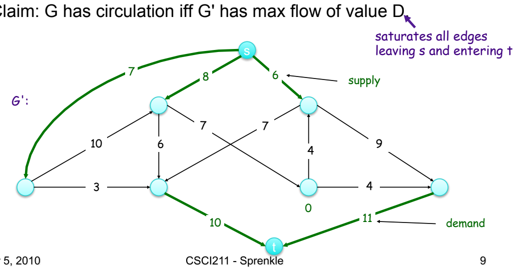
$$\sum_{v:d(v)>0} d(v) = \sum_{v:d(v)<0} -d(v) =: D$$

- Pf. Sum conservation constraints for every demand node v.



Review: 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.



Review: Circulation with Demands: Characterization

- Given (V, E, c, d), there does **not** exist a circulation iff there exists a node partition (A, B) such that

$$\Sigma_{v \in B}\, d_v > \text{cap}(A, B)$$

demand by nodes in B exceeds supply of nodes in B + max capacity of edges going from A → B

- Pf idea. Look at min cut in G'.

Circulation with Demands and Lower Bounds

Force flow to make use of certain edges

- Feasible circulation.
  - Directed graph G = (V, E).
  - Edge capacities c(e) and lower bounds ℓ(e), e ∈ E.
  - Node supply and demands d(v), v ∈ V.
- Def. A *circulation* is a function that satisfies:
  - For each e ∈ E: $0 \le \ell(e) \le f(e) \le c(e)$ (capacity)
  - For each v ∈ 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, ℓ, c, d), does there exists a circulation?

Towards a Solution...

- Start by adding the required minimum flow (f_0) across each edge
  - Could be 0 flow

$f_0^{in}(v) - f_0^{out}(v) = \Sigma_{e \text{ into } v}\, \ell_e - \Sigma_{e \text{ out of } v}\, \ell_e = L_v$

- If $L_v = d_v$, for all v, then satisfied demand
- Otherwise, add amount of flow required to the edge (f_1)

How much capacity is left for f_1? What is the demand of node v?

Towards a Solution…

- Start by adding the required minimum flow (f_0) across each edge
  - Could be 0 flow

$$f_0^{in}(v) - f_0^{out}(v) = \Sigma_{e \text{ into } v} \ell_e - \Sigma_{e \text{ out of } v} \ell_e = L_v$$

- If $L_v = d_v$, for all v, then satisfied demand
- Otherwise, add amount of flow required to the edge (f_1)
  - Capacity: $c_e - \ell_e$
  - Node v's demand: $d_v - L_v$

Circulation with Demands and Lower Bounds

- Model lower bounds with demands
  - Send ℓ(e) units of flow along edge e
  - Update demands of both endpoints
- Simple example:

Circulation with Demands and Lower Bounds

- 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) - ℓ(e) is a circulation in G'.

7.8 SURVEY DESIGN

Survey Design

- Design survey asking consumers about products
- Can only survey a consumer about a product 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.

Sound similar to any earlier problems?

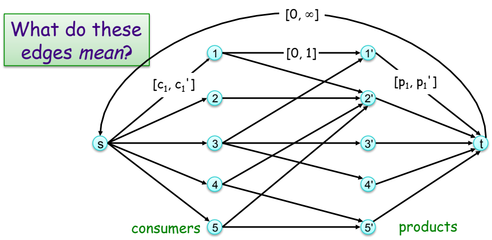
Bipartite Graph

- Nodes are customers and products
- Edge between a customer and product means customer owns product
- For each customer, range of # of products asked about
- For each product, range of how many customers asked about it

- **Bipartite perfect matching:** Special case when $c_i = c_i' = p_i = p_i' = 1$

Survey Design Algorithm

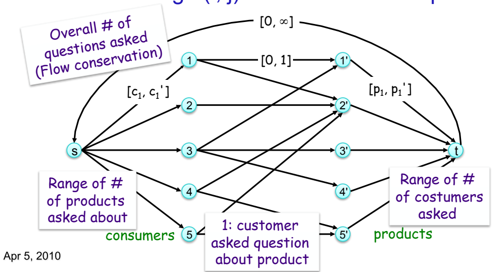
- Formulate as a circulation problem with lower bounds
  - Include an edge (i, j) if customer i owns product j



Survey Design Algorithm

How do we know if we can create a survey? What is the survey?

- Formulate as a circulation problem with lower bounds
  - Include an edge (i, j) if customer i owns product j



Survey Solution

- If a feasible, integer solution, can create the survey
- Customer *i* will be surveyed about product *j* iff the edge (*i*,*j*) carries a unit of flow

7.9 AIRLINE SCHEDULING

Airline Scheduling

- Scheduling goal: efficient in terms of equipment usage, crew allocation, customer satisfaction, …
- Our simplified problem:
  - Flight segment: origin & destination airport, departure & arrival time
  - Use a plane for two flight segments (i, j) if
    - i's destination is same as j's origin & enough time to perform maintenance on plane OR
    - Add a flight segment in between that gets plane to j's origin with adequate time in between

Scheduling Planes

- Maintenance time: 1 hour

Number	Origin	Departure	Destination	Arrival
1	Boston	6 a.m.	DC	7 a.m.
2	Philadelphia	7 a.m.	Pittsburgh	8 a.m.
3	DC	8 a.m.	LAX	11 a.m.
4	Philadelphia	11 a.m.	San Francisco	2 p.m.
5	San Francisco	2:15 p.m.	Seattle	3:15 p.m.
6	Las Vegas	5 p.m.	Seattle	6 p.m.

What is a valid use of one plane for > 1 segment?

Scheduling Planes

- Maintenance time: 1 hour

Number	Origin	Departure	Destination	Arrival
1	Boston	6 a.m.	DC	7 a.m.
2	Philadelphia	7 a.m.	Pittsburgh	8 a.m.
3	DC	8 a.m.	LAX	11 a.m.
4	Philadelphia	11 a.m.	San Francisco	2 p.m.
5	San Francisco	2:15 p.m.	Seattle	3:15 p.m.
6	Las Vegas	5 p.m.	Seattle	6 p.m.

What is a valid use of one plane for > 1 segment?
1→3, 6

Problem Statement

- A flight j is *reachable* from flight i if it is possible to use the same plane for flight j as flight i
- Goal: Determine if it's possible to serve all *m* flights on your original list using at most *k* planes

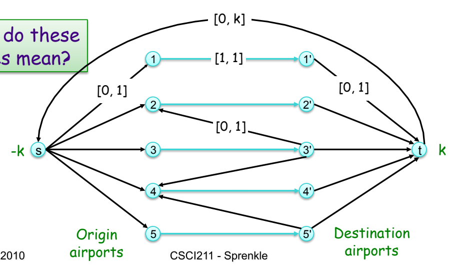
Could we schedule all flights from previous example with only 2 planes?

Ideas about our solution/approach?

Airline Scheduling Algorithm

- Flow: airplanes; Nodes: airports
- Find a feasible circulation

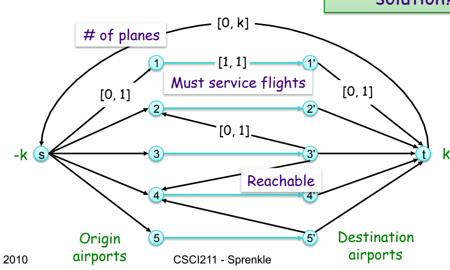
What do these edges mean?



Airline Scheduling Algorithm

- Flow: airplanes; Nodes: airports
- Find a feasible circulation

How do we know if we have a solution? How do we get the solution?



Scheduling Solution

- Construct schedules by following edges from s to origin airports
  - Represents the schedule of one plane