CSE 421
Algorithms

Richard Anderson

Lecture 24

Network Flow Applications

Today's topics

- Problem Reductions
  - Undirected Flow to Flow
  - Bipartite Matching
  - Disjoint Path Problem
- Circulations
- Lowerbound constraints on flows
- Survey design

Problem Reduction

- Reduce Problem A to Problem B
  - Convert an instance of Problem A to an instance Problem B
  - Use a solution of Problem B to get a solution to Problem A
- Practical
  - Use a program for Problem B to solve Problem A
- Theoretical
  - Show that Problem B is at least as hard as Problem A

Problem Reduction Examples

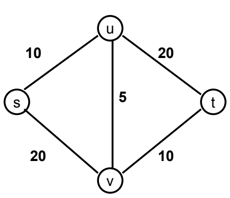
- Reduce the problem of finding the Maximum of a set of integers to finding the Minimum of a set of integers

Find the maximum of: 8, -3, 2, 12, 1, -6

Construct an equivalent minimization problem

Undirected Network Flow

- Undirected graph with edge capacities
- Flow may go either direction along the edges (subject to the capacity constraints)



Construct an equivalent flow problem

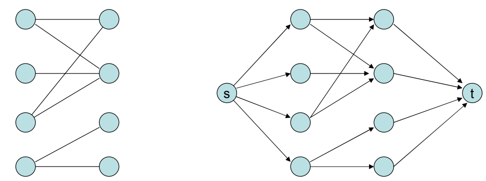
Bipartite Matching

- A graph G=(V,E) is bipartite if the vertices can be partitioned into disjoints sets X,Y
- A matching M is a subset of the edges that does not share any vertices
- Find a matching as large as possible

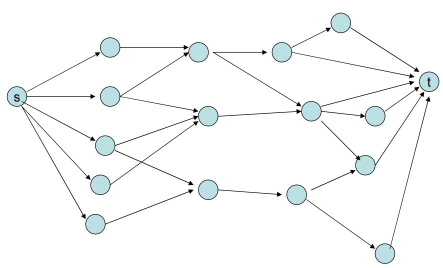
Application

- A collection of teachers
- A collection of courses
- And a graph showing which teachers can teach which courses

Converting Matching to Network Flow



Finding edge disjoint paths



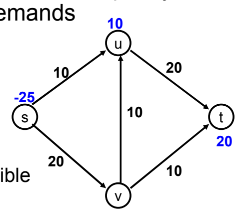
Construct a maximum cardinality set of edge disjoint paths

Theorem

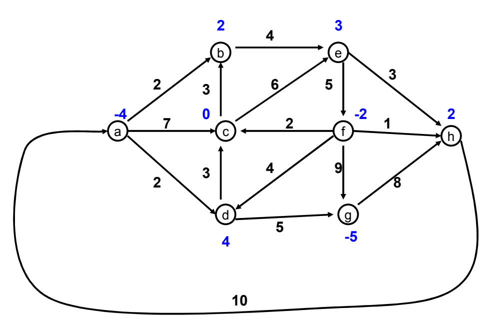
- The maximum number of edge disjoint paths equals the minimum number of edges whose removal separates s from t

Circulation Problem

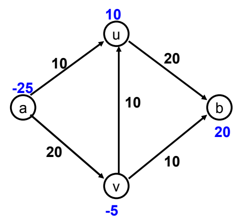
- Directed graph with capacities, c(e) on the edges, and demands d(v) on vertices
- Find a flow function that satisfies the capacity constraints and the vertex demands
  - $0 <= f(e) <= c(e)$
  - $f^{in}(v) – f^{out}(v) = d(v)$
- Circulation facts:
  - Feasibility problem
  - d(v) < 0: source; d(v) > 0: sink
  - Must have $\Sigma_v d(v)=0$ to be feasible



Find a circulation in the following graph



Reducing the circulation problem to Network flow

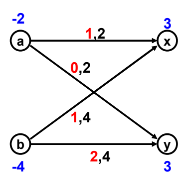


Formal reduction

- Add source node s, and sink node t
- For each node v, with d(v) < 0, add an edge from s to v with capacity -d(v)
- For each node v, with d(v) > 0, add an edge from v to t with capacity d(v)
- Find a maximum s-t flow. If this flow has size $\Sigma_v$cap(s,v) then the flow gives a circulation satisifying the demands

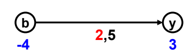
Circulations with lowerbounds on flows on edges

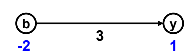
- Each edge has a lowerbound l(e).
  - The flow f must satisfy l(e) <= f(e) <= c(e)



Removing lowerbounds on edges

- Lowerbounds can be shifted to the demands





Formal reduction

- $L_{in}(v)$: sum of lowerbounds on incoming edges
- $L_{out}(v)$: sum of lowerbounds on outgoing edges
- Create new demands d' and capacities c' on vertices and edges
  - $d'(v) = d(v) + l_{out}(v) – l_{in}(v)$
  - $c'(e) = c(e) – l(e)$

Application

- Customized surveys
  - Ask customers about products
    - Only ask customers about products they use
    - Limited number of questions you can ask each customer
    - Need to ask a certain number of customers about each product
    - Information available about which products each customer has used

Details

- Customer $C_1, \ldots, C_n$
- Products $P_1, \ldots, P_m$
- S_i is the set of products used by C_i
- Customer C_i can be asked between c_i and c'_i questions
- Questions about product P_j must be asked on between p_j and p'_j surveys

Circulation construction