Graph Theory: Week 13

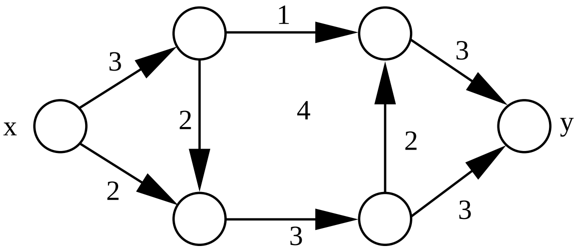
Network flow

John Quinn

20th November, 2007

Networks

- Networks can be used to represent the transportation of some commodity through a system of delivery channels.
- There are *sources* (x) and *sinks* (y).
- The network is a directed graph, where each arc a is associated with a *capacity*, $c(a)$.



Flow

- A *flow* in a network is a set of numbers associated with each arc, $f(a)$.
- This indicates how much of a channel's capacity is being used.
- $0 \leq f(a) \leq c(a)$.
- For a vertex v, the flow into and out of the vertex is denoted by $f^-(v)$ and $f^+(v)$ respectively.
- For intermediate vertices (not sources or sinks) the flow in is the same as the flow out. This is called the *conservation condition*.

Resultant flow

- For some set of vertices S, the resultant flow out of S is given by $f^+(S) - f^-(S)$.
- We are often interested in the resultant flow out of the source x. (Or the set of sources X if there is more than one).
- In particular, we usually want to find a maximum flow, so that as much of the capacity is used as possible in transporting out of the sources to the sinks.
- It is straightforward to extend a network with multiple sources and sinks to one with just one source and sink in order to analyse the maximum flow.

Exercise

- Find all the possible flows and the value of the maximum flow in the example network.

Cuts

- A *cut* is a division of the vertices into two sets S and $\bar{S}$, so that the source is in S and the sink is in $\bar{S}$.
- The capacity of a cut is the sum of all the edges which cross between S and $\bar{S}$.
- How many cuts are possible in a network with ν vertices?
- What are the different cuts of the network on the board, and what are their capacities?

Max-flow min-cut

- In all the examples we have seen, the minimum capacity cut is the same as the maximum flow.
- Intuitively we can think of saturating the bottlenecks.
- To prove, we can show first that max flow $\leq$ min cut (no augmenting paths).
- Then show max flow $\geq$ min cut (removing edges changes capacity).

The Ford-Fulkerson Algorithm

An algorithm for finding the maximum flow in a network.

1. Set the flow to zero for all arcs.
2. Calcuate the residual network G_f. While there is a path p from x to y in G_f:
   2.1 Find $c_f(p) = \min\{c_f(u,v) | (u,v) \in p\}$
   2.2 For each edge in p, add $c_f(p)$ to the flow.
   (Subtract $c_f(p)$ from the flow if the edge is a reverse arc in the network).

   Repeat step 2 until there is no augmenting path.

The residual network G_f has:

- $c_f(u,v) = c(u,v) - f(u,v)$.
- No flow on any edges.

Other problems regarding network flow

- Multi commodity flow: a number of sources produce different products that are to be transported to different sinks using the same network.
- Mimimum cost flow: each arc has an associated cost, and we want to find the cheapest mode of transportation.
- Circulation: there is a lower bound on the flow as well as an upper bound.

Bonus material: Vertex colourings

- Previously we looked at edge colourings of a graph, where no adjoining edges could be the same colour.
- It is also sometimes useful to look at vertex colourings. No adjacent vertices should be the same colour.
- The chromatic number $\chi(G)$ is the fewest vertex colours required for graph G.
- What is the limit on how many colours might be required? Is it the same as for edge colouring?
- Applications?

Exam

- Next Wednesday 28th. 4:00-7:00pm, FST A1.4
- In case of revision difficulties, mail `jquinn@cit.mak.ac.ug` or come to my office on the second floor of the CIT building.
- Assignments are due next week—marks will be submitted to the faculty board shortly after the exams are in.