CCS Discrete III — Professor: Padraic Bartlett

Homework 7: Flow Applications

Due Friday, Week 5 — *UCSB 2015*

Do **three** of the following **five** problems! Have fun!

1. Many students, when they first come across the idea of the max-flow min-cut theorem, actually implement the algorithm as follows:

   i. We start with a network G with source s, sink t, and nonnegative rational-valued capacity function c; we also start with a feasible flow f, that is initially 0 on every edge.

   ii. Take our network G, and attempt to find a path $s = x_0 \to x_1 \to \ldots \to x_n = t$ such that for every i, $f_{x_i x_{i+1}} < c_{x_i x_{i+1}}$.

   iii. If such a path exists, increase f along this path to the minimum of the differences $c_{x_i x_{i+1}} - f_{x_i x_{i+1}}$.
   After doing this, if any edge is saturated — i.e. if $f_{x_i x_{i+1}} = c_{x_i x_{i+1}}$ — mark it as "finished," and don't use it in any future runs of our algorithm.
   Return to (ii.)

   iv. Otherwise, if no such path exists, we have created a maximal flow; halt.

   (a) How does this algorithm differ from the one we presented in class?

   (b) Give an example of a network and sequence of chosen paths where this algorithm fails.

   (c) Show that for any network, there is **some** way in which we can pick paths to run the algorithm above on, so that it **will** generate a maximum flow!

2. In Connor's presentation of the matrix-rounding problem, he assumed (as many textbooks do!) the following claim:

   Theorem. Suppose that G is a network with a capacity function c, such that all of c's values are integers. Also suppose that there is some feasible flow f on G. Then there is an integer-valued maximum flow with value equal to the capacity of the minimum cut.

   We proved this claim for networks where the capacity function takes on nonnegative integer values. However, we did **not** prove this claim for networks where the capacity can take on negative values (i.e. where you can have "minimum" capacities on edges.)

   Prove this here!

   (A hint: By problem 1 on the previous HW, you know that there is some maximum flow f ; you now just want to "round" this flow so that it is a maximum integer-valued flow. Take G, and consider the subgraph H given by the collection of all edges in G where $c_{xy} - f_{xy} \notin \mathbb{Z}$. Come up with a process to repeatedly "shrink" H until it is the empty graph, without changing the overall value of our flow!)

3. Suppose that we want to make a tournament for a two-player game, where $2n$ people are registered to play. We want to have every possible pair of people play each other in the smallest number of rounds possible.

   In class, we claimed that we could do this as follows:

   - Let G be an abelian group of order $2n-1$, say $\mathbb{Z}/(2n-1)\mathbb{Z}$.
   - For each $g \in G$, let $M_g = \{\{g, \infty\} \cup \{a, b\} : a + b = 2g, a \neq b\}$.
   - Identify our teams with the set $G \cup \{\infty\}$, and each of our $2n-1$ rounds with the $2n-1$ distinct M_g's.

   Prove that this design yields a tournament, as claimed.

4. Prove or disprove: there is a tree T with the following properties:

   - The k-th level of T contains k^2 many vertices, for every $k \in \mathbb{N}$.
   - T has no "leaf" vertices: that is, every vertex in T has degree at least 2.
   - A random walker starting at the root of T returns to the root with probability p, for some value $p < 1$.

   (This is the flip side of the "infinite-resistance" quadratic tree problem from last week.)

5. YER A WIZARD, HARRY!

   (ahem) So, yes. You're a wizard! In particular, you are a wizard that has been tasked with writing a column on this year's Quidditch World Cup race. As part of your research, you are trying to determine which teams are still in the running to win their division.

   For each group, you have the following information:

   - Each of the team's current win-loss records.
   - The number of games each team has left versus the other teams in their group.

   For example, the current state of the Oceanic Group could be the following:

Team	Win-Loss	Games Left	NZ	FIJ	AUS	SAM	TON
New Zealand	60-22	36		6	6	4	2
Fiji	58-25	35	6		5	2	5
Australia	17-65	36	6	5		3	9
Samoa	40-41	37	4	2	3		8
Tonga	25-57	36	2	5	9	8	

   From looking at the table, it might seem possible for a team like Tonga to "run the table" and still take first place if they won all of their games, if New Zealand and Fiji were to lose out. However, this isn't actually possible; Fiji and New Zealand have six games against each other, which means that at least one of them will end up with 62 wins, and thereby place ahead of Tonga.

   In general, you have the following information:

   - A list of teams $T_1, \ldots T_n$, each with a number of wins $w(T_i)$ and games remaining $g(T_i)$.

- For any two different teams, we have a number $r(T_i, T_j)$ that tells us the number of games those two teams still have against each other.

Consider the following network:

- For every pair of distinct teams $\{T_i, T_j\}$ make a vertex $v_{i,j}$.
- For every team T_i, make a team vertex t_i. Connect each team vertex to each game it appears in.
- Add a source vertex s and connect it to every game; add a sink vertex t and connect it to every team.

For any team T_i, create a capacity function on this network so that team T_i can win the division if and only if there is a flow that saturates every edge leaving s. (A flow "saturates" an edge (x, y) if and only if $f_{xy} = c_{xy}$.)