Adam Ten Hoeve (11/20) adam.tenhoeve@coloradoe.edu

Problem 1 (20 points)

Given a directed graph $G = (V, E)$ with capacity $c(u, v) > 0$ for each edge $(u, v) \in E$ and demand $r(v)$ at each vertex $v \in V$, a routing of flow is a function f such that

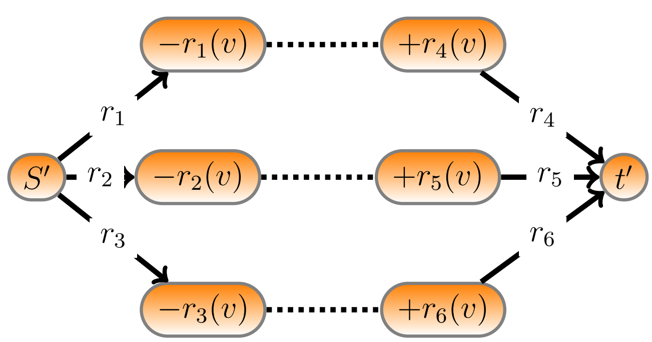
- *for all $(u, v) \in E, 0 \leq f(u, v) \leq c(u, v)$, and*
- *for all $v \in V$,*

$$\sum_{u:(u,v)\in E} f(u, v) - \sum_{u:(v,u)\in E} f(v, u) = r(v),$$

i.e., the total incoming flow minus the total outgoing flow at vertex v is equal to $r(v)$. Notice that the demand $r(v)$ can take positive value, negative value, or zero.

(a) *Show how to find a routing or determine that one does not exist by reducing to a maximum flow problem.*

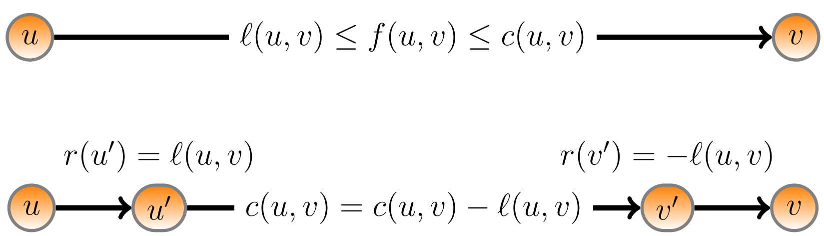
For every "source" vertex in G, the demand $r(v)$ will be negative because flow begins at that vertex and all flow is going out. Every "target" vertex is one that has a positive demand because flow stops at that vertex and does not continue. We can convert graph G into G' by adding a source vertex S' with outgoing edges to all source vertices in G and a target vertex t' with incoming edges from all sink vertices. We then set the capacity of every edge (S', s) as the demand of that source vertex $r(v)$, and do the same thing with the edges (t, t'). An example of the graph G' would look like:



The total capacity of edges (S', s) is D, which is the sum of all of the demands of the sources vertices. If we saturate all edges from S', then all edges going to t' must also be at max capacity because the flow is conserved. Therefor, all demands of all other vertices must have been met. This shows that to find a routing, we can saturate the source vertex and see if the max-flow value is D.

(b) *Suppose that additionally there is a lower bound $\ell(u, v) > 0$ at each edge (u, v), and we are looking for a routing f satisfying $f(u, v) \geq \ell(u, v)$ for all $(u, v) \in E$. Show how to find such a routing or determine that one does not exist by reducing to a maximum flow problem.*

To make the problem into a maximum flow problem, we must again add the source vertex S' and target vertex t'. Then we will have to add more to the graph. For ever edge e between vertex u and v, we will create a vertex u' with an edge (u, u') and another vertex v' with edge (v, v'). We can then set $r(u') = \ell(u, v)$, $r(v') = -\ell(u, v)$, and change the capacity of edge (u, v) to $c(u, v) - \ell(u, v)$. This is demonstrated in the picture bellow:



With this, we have reduced every edge to an edge without lower bound. If the demands for all u' and v' can be met then problem can be solved as a maximum flow problem and a routing can be found.

Problem 2 (20 points extra credit)

Even though in this class we focus on those greedy algorithms that generate optimal solutions, in general a greedy algorithm may not give an optimal solution. So, we are interested in those greedy algorithms that generate a good enough solution, i.e., not too far from the optimal solution. Let us consider one such problem as follows.
Given subsets $S_1, S_2, \cdots, S_n$ of a set S of points and an integer m, a maximum m-cover is a collection of m of the subsets that covers the maximum number of points of S. Finding a maximum m-cover is a computationally hard problem. Give a greedy algorithm that achieves approximation ratio $1-1/e$; i.e., let y be the maximum number of points that can be covered by m subsets and x be the number of points that are covered by the m subsets generated by your algorithm, then give a greedy algorithm such that $x \geq (1-1/e)y$. Here are some useful hints:

- *You may want to use the inequality $(1-1/m)^m \leq 1/e$ for integer $m \geq 1$.*
- *You may want to use induction at some point.*