Massachusetts Institute of Technology
6.854J/18.415J: Advanced Algorithms
David Karger

Handout 7
Wednesday, September 28, 2005

Problem Set 4

Due: Friday, October 7 2005.

Notice that one problem is marked **noncollaborative.** As you might expect, this problem should be done without any collaboration.

NONCOLLABORATIVE Problem 1. Which of the following statements about flows are true and which are false? Justify your answer with a (short) proof or counterexample.

(a) If all directed edges in a network have distinct capacities, then there is a unique maximum flow.

(b) If we replace each directed edge in a network with two directed edges in opposite directions with the same capacity and connecting the same vertices, then the value of the maximum flow remains unchanged.

(c) If we add the same positive number λ to the capacity of every directed edge, then the minimum cut (but not its value) remains unchanged.

Problem 2. In this problem we develop a scaling algorithm for shortest paths that achieves running time $O(m \log C)$ with no data structure more complicated than an array. Recall that when we discussed shortest paths with positive integer edge lengths, we showed how Dial's algorithm uses a single array of buckets to find shortest paths in $O(m + nC)$ time for maximum edge length C.

Consider a solution to a shortest path problem with edge lengths l_{vw} from v to w in which the (shortest path) distance of vertex w from the source s is d_v. Define the *reduced edge lengths* l^d_{vw} by the rule $l^d_{vw} = l_{vw} + d_v - d_w$.

(a) Argue that in fact the running time of Dial's algorithm is $O(m + D)$, where D is the maximum distance, and that the algorithm still works if some edges have length 0.

(b) Show that the reduced edge lengths defined above are all nonnegative integers.

(c) Show that the shortest paths under the reduced length function are the same as those under the original length function. What is the length of shortest paths under l^d?

(d) Devise a scaling algorithm for shortest paths. Use the reduced edge lengths and Dial's bucketing algorithm to keep the task of shifting in a new bit simple.

(e) Is base-2 scaling the best possible, or can you achieve a (slightly) better running time by scaling with a different base?

Problem 3. At lunchtime it is crucial for people to get to the food trucks as quickly as possible. We will show how to use maximum flow to solve this problem. Consider the following model. The building is represented by a graph $G = (V, E)$, where each room, landing, or other location is represented by a vertex and each corridor or stairway is represented by an edge. Each corridor has an associated capacity c, meaning that at most c people can pass through the corridor at once. Traversing a corridor from one end to the other takes one timestep. (Traversing a room takes zero time.)

(a) Suppose all people are initially in a single room s, and that the building has a single exit t. Show how to use maximum flow to find a fastest way to get everyone out of the building. Hint: create an auxiliary graph G' that has vertices to represent each room *at each time step.*

(b) Show that the same technique can be used when people are initially in multiple locations and there are multiple exits.

(c) Generalize the approach to where different corridors have different (integral) transit times.

OPTIONAL (d) Suppose transit times are not integral. Is there still a way to solve the problem?

OPTIONAL (e) The above algorithm is polynomial in the number of people. Can you improve it to be polynomial (in the graph size) regardless of the number of people?

Problem 4. The minimum flow problem is a close relative of the max flow problem with nonnegative lower bounds on edge flows (if an edge (i, j) has a lower bound $l_{ij} \geq 0$, then f_{ij} must satisfy $l_{ij} \leq f_{ij} \leq u_{ij}$). In the minimum flow problem we wish to send a minimum amount of flow from the source s to the destination t while satisfying given lower and upper bounds on edge flows f_{ij}.

(a) Show how to solve the minimum flow problem by using two applications of any maximum flow algorithm that applies to problems with zero lower bounds on edge flows (e.g., the algorithms described in the lecture). **Hint**: first construct any feasible flow and then convert it into a minimum flow.

(b) Prove the following min-flow-max-cut theorem. Let the lower bound on the cut capacity of an s-t cut $(S, T = V \setminus S)$ be defined as $\sum_{(i,j)\in S\times T} l_{ij} - \sum_{(i,j)\in T\times S} u_{ij}$. Show that the minimum value of all feasible flows from node s to node t equals to the maximum lower bound on cut capacity of all s-t cuts.

(c) A group of students wants to minimize their lecture attendance by sending only one of the group to each of n lectures to take notes. Lecture i begins at time a_i and ends at time b_i. It requires r_{ij} time to commute from lecture i to lecture j. Develop a flow-based algorithm for identifying the minimum number of students needed to cover all the lectures. **Hint:** reduce to minimum flow.

OPTIONAL Problem 5. Augmenting paths are guaranteed to terminate in finite time with a maximum flow only if the edge weights are rational.

(a) Give a graph on which some augmenting path algorithm fails to terminate it finite time.

(b) (Harder) Give a graph on which some augmenting path algorithm fails to terminate it finite time, and whose limiting flow is not maximum.