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

Handout 9
Monday, September 29, 2003

Problem Set 5

Due: Monday, October 6, 2003.

NONCOLLABORATIVE Problem 1. Critical edges for maximum flow.

(a) An edge is *upward critical* if increasing the capacity of this edge increases the maximum flow value. Does every network have an upward-critical edge? Describe an algorithm for identifying all upward-critical edges in a network. The worst-case complexity of your algorithm should be substantially better than that of solving m maximum-flow problems.

(b) An edge is *downward critical* if decreasing the capacity of this edge (by any amount) decreases the maximum flow value. Is the set of upward-critical edges the same as the set of downward-critical edges? If not, describe an algorithm for identifying all downward-critical edges, and analyze your algorithm's worst-case complexity.

Problem 2. You are TAing a class with s students and r recitations. The students tell you which recitations they can fit into their schedules. However, each recitation has capacity for only k students. (Conversely, each student has capacity for only one recitation.) Your goal is to assign the most students to recitations within these constraints.

(a) Suppose first that $k = 1$ (personal tutors!). How can you easily solve this problem in $O(m\sqrt{n})$ time?

(b) Now suppose k is arbitrary. How can you solve this problem with maximum flow?

(c) Give a short self-contained argument that our improvements to the analysis of blocking flow for unit-capacity networks can be applied to your network from part (b) to obtain a time bound of $O(m^{3/2})$.

(d) Give a longer argument that, in fact, an $O(m\sqrt{n})$ time bound can be achieved for your network from part (b), as in unit-capacity networks.

(e) Suppose each student i bids $\$_{ij}$ dollars to get into recitation j. Give an efficient algorithm to assign students to recitations to maximize the professor's profits. (You no longer need to maximize the number of successful assignments of students to recitations.)

Problem 3. Cost-scaling algorithm for minimum-cost flow.

(a) Suppose that you have an optimal solution to some minimum-cost circulation problem and you then change one edge cost by one unit. Show how you can re-optimize the solution faster than solving the minimum-cost circulation problem from scratch.

(b) Deduce a cost-scaling algorithm for minimum-cost flow that makes $O(m \log C)$ calls to your solution to part (a).

OPTIONAL (c) Design a cost-scaling algorithm for minimum-cost flow that makes $O(n \log C)$ calls to your solution to part (a).

Note: The resulting time bounds are worse than the capacity-scaling algorithm we saw in lecture, but a more careful algorithm along these lines attains a better time bound of $O(mn \log\log U \log_n C)$.

Problem 4. Suppose that you are given a network with capacities and costs. In lecture, we saw how to find a flow of either maximum flow or minimum cost, both in polynomial time. In general, we have a *bicriterion optimization problem* where we would like to simultaneously maximize flow value and minimize cost. Show how to solve the following problems in polynomial time via a reduction to minimum-cost flow.

(a) Find the minimum-cost flow subject to the constraint that the flow value is at least 90% of the maximum flow value.

(b) Find the flow that minimizes the cost of the flow plus K times the difference between the flow value and the maximum flow value, for a given weight K.