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

Handout 11
October 26, 1999

Problem Set 6

Due: November 2, 1999.

Problem 1. Suppose that you are given a minimum cost flow problem, but told you need not find a maximum flow. Solve the following two variants efficiently via reduction to standard min-cost flow problems.

(a) Find the minimum cost flow sending at least 90% of the maximum possible amount of flow from s to t.

(b) Find the flow of minimum cost if you are allowed to reduce the flow value below its maximum at a cost of K per unit of reduction.

Problem 2. Given a min-cost flow problem and a solution to it, give an $O(mn)$ time algorithm for deciding whether there is a different solution and outputting one if there is.

Problem 3. Suppose that you have an optimal solution to some min-cost circulation problem and you then change one edge cost by one unit. Explain how you can re-optimize the problem in $O(mn \log n)$ time. Deduce a (truly) cost-scaling algorithm for min-cost flow that requires $O(m^2 n \log n \log C)$ time.

Problem 4. It is visiting day for the new graduate admits to MIT. The day is divided into d "slots" in which one faculty member can meet one new student. Student i is willing to pay a certain amount of money c_{ij} to see faculty member j.

Note: this problem broke; b is trivial; the real question requires Hall's theorem.

(a) Supposing $d = 1$, give an efficient algorithm that lets the department make as much money as possible off the visiting admits.

(b) Do the same if $d > 1$, but it is ok for a students to see the same faculty member multiple times (and pay again).

(c) Do the same if a second (distinct) student (but no more) can meet the same faculty member at the same time, but they only pay half their suggested price (first student still pays full price).

** **(d)** Do the same if each student may meet each faculty member at most once.

This solution was actually used to schedule last years' visit day (except for the part where we took the students' money).

* **Problem 5.** In graphs with unit edge capacities, the blocking flow bounds can be extended to min-cost flows. The goal here is to develop a cost scaling step that transforms an ϵ-optimal flow to an $\epsilon/2$-optimal flow in $O(m^{3/2})$ time. As with standard cost-scaling, we start by saturating all the negative reduced-cost arcs, creating excess that must be routed back to the deficits while adjusting prices to maintain $\epsilon/2$-optimality. Define the *admissible arc graph* to be the set of arcs that have negative cost. We will maintain that this graph is acyclic.

(a) Show that if there is no admissible path from any excess to any deficit, we can "relabel" the excesses and all vertices they reach in the admissible graph, decreasing their prices by $\epsilon/2$, without violating $\epsilon/2$-optimality. Note that this shortens every excess-deficit path by $\epsilon/2$.

(b) We don't really have a layered graph. Nevertheless explain how a blocking flow can be used to eliminate all admissible excess-deficit paths if the admissible graph is acyclic.

(c) Show that after $O(\sqrt{m})$ block/relabel steps, only $O(\sqrt{m})$ units of excess will still be in the graph (**hint:** of the original flow paths used to saturate negative arcs, most will have had their reverses made admissible by the time $O(\sqrt{m})$ blocking steps occur, so excess can return where it came from. Alternatively, consider the difference in cost of the preflow under the starting and current cost functions.)

(d) Argue that after a blocking step, it is always possible to relabel to create at least one new admissible path.

(e) Show that $O(\sqrt{m})$ block/relabel steps suffice to complete a scaling phase. What is the running time of the resulting min-cost flow algorithm?