CS 222 Fall 2007 HW 3 Corrected Solutions.

Below is the statement of problem 2, and a modified solution. The main change is what is in bold, although there are other smaller changes.

2. Suppose we have the height and weight of every person in the class, and we organize this information into a two dimensional table T where there are n height intervals, and m weight intervals. In the table, the (i, j) entry $T(i, j)$ indicates how many people fall into height category i and weight category j. Suppose that some of the particular cell entries are kept secret (so they are empty), but the number of people falling into each height category is public, as is the number that fall into each weight category. That is, the n row sums and the m column sums of the table are published, along with some, but not all, of the individual cell values of the table.

Devise an algorithm, based on network flow, that finds a way to assign values to the empty cells so that every row now adds up to its given row total, and every column now adds up to its given column total.

Answer: Set up a bipartite graph $G = (A, B)$ with one node in A for each row and one node in B for each column, and an edge (i, j) between the node for row i and the node for column j, if cell (i, j) is empty. Put infinite capacity on each such edge. Now add a source node s with a directed edge to each node i in A, with capacity equal to the given row total for i, **minus the sum of the non-empty values in row** i. Add a terminal node t with a directed edge to it from each node j in B, with capacity equal to the column total for j, **minus the sum of the non-empty values in column** j. A maximum s-t flow which saturates the edges out of s (and hence also saturates the edges into t) describes valid values to use for the missing cells.

In preparation for other problems, note that every maximum s-t flow in this graph saturates all the edges out of s and all the edges into t, and moreover, for any node x in the network, the total flow into x is the same in every maximum s-t flow.

Problem 4. Note: 4a is the same as the first part of the original problem 4, and the answer is unchanged.

4a. Let f and f' be two different maximum s-t network flows in a graph G, where the total flow into any node v is the same in f as in f' (this may be the case for example in your solution to problem 2). Let $e = (u, v)$ be a directed edge where $f'(e) > f(e)$. Show that in the residual graph G_f for flow f, there must be a directed cycle that includes the forward edge (u, v).

Similarly, in the residual graph $G_{f'}$ for f', there must be a directed cycle that includes the backward edge (v, u).

Answer: For each edge $g = (p, q)$, where $f'(g) \neq f(g)$, define $F(p, q) = f'(g) - f(g)$, which could be positive or negative. Note that for any node $q \neq s$ (even t), $\Sigma_p F(p, q) = 0$, since the total flow into q is the same in f and f'. Similarly for any node $p \neq t$ (even s), $\Sigma_q F(p, q) = 0$, since the total flow out of p is the same in f and f'. Also, note that if $F(p, q) > 0$ then (p, q) is a forward edge in G_f since $f(p, q)$ cannot be at its capacity. In fact, the residual capacity of edge (p, q) in G_f is at least $F(p, q)$. Similarly, if $F(p, q) < 0$ then $f(p, q) > 0$, so edge (q, p) is a backward edge in G_f with residual capacity $f(p, q) \geq f(p, q) - f'(p, q) = |F(p, q)|$. So in G_f, if $F(p, q) > 0$ then assign $F(p, q)$ to the forward edge (p, q); if $F(p, q) < 0$, assign $|F(p, q)|$ to the backward edge (q, p). But don't assign anything to edge (u, v), or to edge (v, u). I claim that the result is a flow from v to u of total value $F(u, v)$. To see this, note that the assigned values obey the capacity constraints on the edges in G_f, and for every node other than v or u, the total of the assigned values into the node equals the total of the assigned values out of the node. This is even true for nodes s and t. The total out of v is greater than the total in by exactly $F(u, v)$ and the total in to u is greater than the total out of u by exactly $F(u, v)$. Thus this assignment satisfies the definition of a flow from v to u, of value $F(u, v)$. Now since $F(u, v) > 0$, there certainly must be a directed path in G_f from v to u, and when we include the edge $e = (u, v)$ (which exists in G_f since $f(u, v) < f'(u, v) \leq c_e$), we have shown that e is in a directed cycle in G_f. Note that this flow in G_f does not put any flow on the backward edge (v, u), so in fact there is a directed cycle in G_f that contains e and that has at least three nodes.

The second part of the original problem 4 asked about flows f'' and f' in a general directed graph. That question was intended as preparation for use in Problem 2 of HW 4, and the algorithm I gave and the answer I presented is correct when applied to networks used in the solution to problem 2 in HW 3. However, it is not true in general. Below I restate the problem and solution correctly. Sorry for any wasted effort on your part in trying to prove the more general result.

4b. Now consider the network G used in the solution to problem 2 of this homework (HW 3), and let f' be a maximum s-t flow on that network. We want to know if there is another maximum s-t flow f'' in G where $f''(e) >$

$f'(e)$, assuming that $f'(e) < C_e$.

We claim we can solve this question with the following method: Let $G_{f'}$ be the residual graph for flow f'. Then remove the edge (v, u) from $G_{f'}$ and find the maximum flow value in that $G_{f'}$, from v to u. (That is, let v be the source node, and u be the terminal node in that flow computation.)

Problem: How does the flow value computed in the above method answer the question of whether such an f'' exist? That is, complete the description of the above method, and explain why the method is correct.

Answer: To finish the algorithm find a maximum flow from v to u in $G_{f'}$. There is the claimed flow f'' if and only if the maximum v to u flow has non-zero value. The proof of this is as follows.

By the answer to problem 4a, if there is such an f'', there is a directed path from v to u in $G_{f'}$, so there is some non-zero $v - u$ flow in $G_{f'}$. Note that this direction of the answer does not even depend on whether the graph G is of the type used in the solution to problem 2 of this homework.

Conversely, if the maximum flow value in $G_{f'}$ is greater than 0, then let b be the minimum non-zero flow on any edge in the flow. Hence there must be some simple directed path P from v to u in $G_{f'}$ with minimum non-zero capacity of b. Now since G is a network used in the solution to problem 2, all edges out of s are saturated in flow f', so there is no edge out of s in $G_{f'}$. Similarly, all edges into t are saturated, so there is no edge into t in $G_{f'}$. Therefore, path P cannot go through s or t; any v to u path touches only nodes in $A \cup B$. Moreover, since every such edge is directed from a node in A to a node in B, any directed path in $G_{f'}$ from v to u (and in particular P) consists of alternating back and forward edges. Therefore, if we increase the flow on any forward edge on P by b, and decrease the flow by b on any edge (i, j) where (j, i) is a backward edge on P, the total flow into any node on P other than v and u will be unchanged. Then if we increase the flow on (u, v) by b, the result will be another $s - t$ maximum flow where the total flow into any node is the same as in f', but where the flow on e is $b > 0$ units larger than in f'. So if there is a non-zero $v - u$ flow in G_f, there is a flow f'' as described above.

NOTE: Because problem 2 of HW 4 gave a hint of reading the answer to problem 4, and the original second part of problem 4 was in error, I have posted on the class website that you do not need to do problem 2 of HW 4 now. We may see it again in some other way later. Sorry for the confusion.

And since you do not need to do problem 2 of HW 4, here is a solution to it. The problem is stated first:

2. In Problem 2 from HW 3, a "legal assignment" of values to the empty cells, is one such that with the assigned values included, every row adds up to its given row total, and every column adds up to its given column total. In HW 3 you solved the problem of finding one legal assignment if there is one, and otherwise determining that there is no legal assignment. For technical simplicity, we assume now that each of the cell values is an integer, and therefore the row and column totals are also integers.

We would now like to know for each empty cell (u, v), what is the *maximum* possible value, call it $v^*(u, v)$, such that there is some legal assignment of values to the empty cells which assigns value $v^*(u, v)$ to cell (u, v).

Show how to use network flow to determine $v^*(u, v)$. A single network flow computation suffices, after determining some legal assignment. Hence if we want to compute $v^*(i, j)$ for every empty cell (i, j), it suffices to compute one flow to determine a first legal assignment, followed by one flow per empty cell.

Answer: Let $e = (u, v)$. First, using the answer to Problem 2 in HW 3, find some legal assignment using a maximum flow f in graph G, and build the residual graph G_f. Note the flow into any node x in G is the same for each maximum s-t flow, since in any legal assignment, every row and colum total is the same. Hence the results from Problem 4a and Problem 4b on HW 3 apply here.

Now compute a maximum flow fr from v to u in G_f. By problem 4a on HW 3, if that flow has value 0, then $v^*(u, v) = f(e)$. That is, the flow in f on e is already as large as possible in any flow corresponding to a legal assignment of values to emtpy cells. By problem 4b (the modified problem and solution above), if the maximum $v - u$ flow value in fr is 0, then there is another maximum flow in G where the flow on e is greater than $f(e)$, and so $v^*(u, v) > f(e)$. But how much greater than $f(e)$ is $v^*(u, v)$? The answer is that it is exactly the value of the v to u flow in G_f. The proof of that is a bit subtle.

Looking again at the given solution to problem 4a, note that f' was only defined as some s-t maximum flow where $f'(e) > f(e)$. But now, examine that proof when we require that f' be a maximum s-t flow in G which maximizes the flow on edge $e = (u, v)$. That is, among all the maximum s-t flows in G, f' must be one which puts the most flow on e. Then define

$F(i,j) = f'(i,j) - f(i,j$ as in the solution to problem 4a, and recall that assigning $F(i,j)$ to any edge where $F(i,j) > 0$, and assigning $|F(i,j)|$ to any backward edge (j,i) where $F(i,j) < 0$, defines a flow (let us call it F^*) in G_f from v to u of value $F(u,v)$, and that this implies there is an s-t maximum flow in G where the flow on edge $e = (u,v)$ is $f(e) + F(u,v)$. But since f' is defined as the s-t flow that puts the most flow on edge $e = (u,v)$, and the v to u flow we compute in G_f only uses information from f (and not any other particular $s - t$ flow), it must be that the computed flow fr has the same v to u value as F^*. Therefore $v^*(u,v)$ must be exactly $f(e)$ plus the $v - u$ flow in fr.