CS 38 Introduction to Algorithms **Spring 2014**

Solution Set 5

Posted: May 22

If you have not yet turned in the Problem Set, you should not consult these solutions.

1. (a) We have a variable x_e in our LP for each edge in the flow network. The objective function is $\sum_{e \text{ exiting } s} x_e$ and the constraints are
   - $\sum_{e \text{ entering } v} x_e - \sum_{e \text{ exiting } v} x_e = 0$, for all v except s and t,
   - $\sum_{e \text{ incident to } v} x_e \leq d(v)$, for all v, and
   - $0 \leq x_e \leq c(e)$, for all e.

   This is the standard LP formulation of max-flow, with additional constraints representing the vertex-limiting feature.

   (b) We have a variable x_e in our LP for each edge in the flow network. The objective function is $\sum_{e \text{ exiting } s} 2x_e - x_{e^*}$ and the constraints are
   - $\sum_{e \text{ entering } v} x_e - \sum_{e \text{ exiting } v} x_e = 0$, for all v except s and t,
   - $0 \leq x_e \leq c(e)$, for all e.

   This is the standard LP formulation of max-flow, but the objective function is non-standard. We claim that the objective function is maximized when $f = \sum_{e \text{ exiting } s} x_e$ equals the max-flow value, and subject to that, x_{e^*} is minimized. Let $f_{\max}$ be the maximum flow value achievable. Suppose $f < f_{\max}$. Then by Ford-Fulkerson, we can add $\Delta = f_{\max} - f$ units of flow, and this increases the flow across edge e^* by at most Δ units of flow. So the net change in the objective function is $+2\Delta - \Delta$ which is positive, so it must be that $f = f_{\max}$. Then it is clear that subject to that constraint, the objective function is maximized when x_{e^*} is minimized.

2. (a) We have a variable $x_{u,v}$ for each edge in the bipartite graph $G = (V, U, E)$. If $w(u, v)$ are the edge weights, the objective function is $\sum u, vw(u, v)x_{u,v}$. We have constraints:
   - $0 \leq x_{u,v} \leq 1$ for all u, v,
   - $\sum_u x_{u,v} = 1$ for all v,
   - $\sum_v x_{u,v} = 1$ for all u,

   In other words, we require that the variables $x_{u,v}$ are $0/1$ depending on whether edge u, v is included in the matching or not. We also require that every vertex on the left is matched (and no vertex has more than one incident edge in the matching) and similarly for the vertices on the right. We observe that the $x_{u,v} \leq 1$ constraints are redundant, since every $x_{u,v}$ is involved in one of the other constraints that assert that the sum of several variables equals 1, and all variables are non-negative. We need this observation for the constraints to be in standard form so we can argue about total unimodularity.

   Now, we argue that the constraint matrix is totally unimodular. Observe that every column in the constraint matrix contains exactly two ones (since each $x_{u,v}$ appears in exactly exactly two constraints), and in fact by reordering rows (which doesn't affect total-unimodularity), we can assume that one such one is in an odd row and the other is in an even row.

We claim that any square submatrix of such a matrix has determinant 0, 1, or -1. It is clear that any submatrix has *at most* two 1's per columns, and if there are two 1's, one is in an even row and the other is in an odd row. The proof is by induction on the size of the submatrix. If it is 1×1, it clearly has determinant 1 or 0. Now assume the claim holds for $i-1 \times i-1$ matrices. Consider any $i \times i$ submatrix. If it has a column of all zeros, its determinant is zero and we are done. If it has a column with a single one in it, then we can expand the determinant along that column and find that it equals ± 1 times the determinant of an $i-1 \times i-1$ submatrix, which is ± 1 or 0 by induction. Otherwise every column has two ones in it, one in an odd row and the other in an even row. Consider multiplying on the left by the row vector with 1's in the corresponding odd coordinates and -1's in the corresponding even coordinates. The result is the zero vector, so the matrix is singular and the determinant is zero. This concludes the proof of the claim.

(b) We have a variable $x_{i,j}$ for each pair of i-th storage lot and j-th showroom, and the objective function is $\sum i, i - c_{i,j} x_{i,j}$ (since we want to minimize cost). We have constraints:

- $\sum_j x_{i,j} \leq a_i$ for all i,
- $\sum_i x_{i,j} \geq b_j$ for all j, and
- $0 \leq x_{i,j}$ for all i, j

The first set of constraints assert that the total outflow from the i-th storage lot does not exceed the supply and the second set of constraints assert that the total inflow to the j-th showroom is at least the demand b_j for that showroom. To get this LP into standard form, we'll need slack variables s_i and t_j, and the constraints become:

- $\sum_j x_{i,j} + s_i = a_i$ for all i,
- $\sum_i x_{i,j} - t_j = b_j$ for all j,
- $0 \leq x_{i,j}$ for all i, j,
- $0 \leq s_i$ for all i, and
- $0 \leq t_j$ for all j

Now, we argue that the constraint matrix is totally unimodular. The proof is similar to the previous part. Observe that every column in the constraint matrix contains exactly two ones (since each $x_{i,j}$ appears in exactly exactly two constraints) OR it contains exactly one 1 or exactly one -1 (if it corresponds to one of the slack variables. Again, by reordering rows (which doesn't affect total-unimodularity), we can assume that every column with two ones, has one such one in an odd row and the other in an even row.

We claim that any square submatrix of such a matrix has determinant 0, 1, or -1. The proof is by induction on the size of the submatrix. If it is 1×1, it clearly has determinant ± 1 or 0. Now assume the claim holds for $i-1 \times i-1$ matrices. Consider any $i \times i$ submatrix. If it has a column of all zeros, its determinant is zero and we are done. If it has a column with a single 1 or -1 in it, then we can expand the determinant along that column and find that it equals ± 1 times the determinant of an $i-1 \times i-1$ submatrix, which is ± 1 or 0 by induction. Otherwise every column has two ones in it, one in an odd row and the other in an even row. Consider multiplying on the left by the row vector with 1's in the corresponding odd coordinates and -1's in the corresponding even coordinates. The result is the zero vector, so the matrix is singular and the determinant is zero. This concludes the proof of the claim.

3. (a) Given distributions p and q, consider the quantities $Q_j = \sum_i p_i M[i,j]$ and $P_i = \sum_j q_j M[i,j]$, and let $Q_{\max} = \max_j Q_j$ and $P_{\max} = \max_i P_i$. We claim that p and q represent a Nash equilibrium iff $Q_j = Q_{\max}$ for all $j \in T$ and $P_i = P_{\max}$ for all $i \in S$.

   Suppose for some $j \in T$ we have $Q_j < Q_{\max}$, and let j^* be such that $Q_j = Q_{\max}$. Then the column player's expected payoff can be improved by shifting the probability from p_j to p_{j^*}. Similarly, if for some $i \in S$ we have $P_i < P_{\max}$, and i^* is such that $P_i = P_{\max}$, then the row player's expected payoff can be improved by shifting the probability from q_i to q_{i^*}. This proves the forward direction (via the contrapositive).

   Now suppose we have $Q_j = Q_{\max}$ for all $j \in T$ and $P_i = P_{\max}$ for all $i \in S$. Any mixed strategy p' for the row-player achieves an expected payoff that is some convex combination of the Q_j, and thus it can never be larger than $Q_{\max}$, which is achieved by p. Similarly any mixed strategy q' for the column-player achieves an expected payoff that is some convex combination of the P_i, and thus it can never be larger than $P_{\max}$, which is achieved by q. Thus p, q represent a Nash equilibrium.

   Given this characterization, we can produce an LP. The objective function is just 1 (i.e. there is no optimization necessary; we are just interested in a feasible solution). The variables will be the p_i and q_j, P and Q, and we have constraints:

   - $\sum_i p_i M[i,j] \leq Q$ for all j,
   - $\sum_j q_j M[i,j] \leq P$ for all i,
   - $\sum_i p_i M[i,j] = Q$ for all $j \in T$,
   - $\sum_j q_j M[i,j] = P$ for all $i \in S$,
   - $0 \leq p_i, q_j, P, Q$ for all i, j

   In any feasible solution to this LP, the variables P and Q equal $Q_{\max}$ and $P_{\max}$, respectively. By the above characterization, any feasible solution to this LP yields a Nash equilibrium p, q. Since LPs can be solved in polynomial time, this takes only polynomial time given S, T.

   (b) We simply enumerate over all subsets S, T. This takes 2^{2n} time. For each pair, we solve the LP from the previous part. We can check if the solution satisfies the characterization from that part in polynomial time as well. If we obtain a solution (which is guaranteed since Nash equilibria always exist, although you were not expected to know that), we output it and halt. The overall running time is $O(2^{2n}) \cdot \text{poly}(n)$ as required.