15-451 Algorithms, Spring 2016
Recitation #13 Worksheet

Primal and Duals and Matching Markets

Consider the following linear program to find a max-weight perfect matching in bipartite graphs $G = (L, R, E)$. Assume that $|L| = |R|$, and $E = L \times R$, i.e., all edges are present, and weights v_{ij} are non-negative.

$$\begin{aligned} \max \textstyle\sum_{ij} v_{ij} x_{ij} & \\ \textstyle\sum_{j \in R} x_{ij} = 1 & \qquad \forall i \in L \\ \textstyle\sum_{i \in L} x_{ij} = 1 & \qquad \forall j \in R \\ x_{ij} \geq 0 & \end{aligned}$$

Let's write the dual of this program (observing that if the constraints are equalities then the dual variables are not constrained to be positive).

$$\begin{aligned} \min \textstyle\sum_i p_i + \sum_j u_j & \\ p_i + u_j \geq v_{ij} & \qquad \forall (i,j) \in E \end{aligned}$$

Think of L as items, and R as buyers, with v_{ij} the value of item i for buyer j.

1. Suppose the preference graph at prices $\{p_i\}$ admits a perfect matching M. (I.e., the prices are "market-clearing".) Define $u_j := \max_i\{v_{ij} - p_i\}$. Show that these values p_i and u_j form a feasible dual solution with value $\sum_{(i,j) \in M} v_{ij}$.

   Solution: For any i, j, $p_i + u_j = p_i + \max_i(v_{ij} - p_i) \geq p_i + (v_{ij} - p_i) = v_{ij}$. So the dual is feasible. This did not need the market-clearing property.

   Observe that for any edge in the preference graph, $u_j = v_{ij} - p_i$. Hence

   $$\sum_i p_i + \sum_j u_j = \sum_{(i,j) \in M} (p_i + u_j) = \sum_{(i,j) \in M} v_{ij}.$$

2. Since we have an integer primal solution M (a matching), and a feasible dual solution, both having the same value, infer that this matching must be social-optimal matching.

   Solution: Consider the solution where $x_{ij} = 1$ for all $(i,j) \in M$ and $x_{ij} = 0$ otherwise. This is a feasible primal solution.

   Moreover, the dual solution is feasible (from above) and has value $\sum_{i,j} v_{ij} x_{ij} = \sum_{(i,j) \in M} x_{ij}$, which is the value of the primal. As we know, if we have feasible primal and dual solutions, and they have the same value, they must be both optimal. Hence M is a max-value matching, which is the social-optimum.

Online Algorithms

Searching for your Keys. You are standing in the middle of a long street (which you model by the infinite line, where you're at the origin). You have lost your keys at some unknown *(positive or negative)* integer location X. To find your keys, you have to walk around until you find them. The competitive ratio is the ratio of the distance you travel until you reach X, divided by $|X|$ (which is the distance from the origin to location X, the best you could have done had you known X).

1. Consider the strategy that starts at 0 and walks to $1, -1, 2, -2, 3, -3, 4, -4, \ldots,$ until you find the keys. What is the competitive ratio?

   Solution: The competitive ratio is unbounded. Note that if the keys are at $X > 0$, we will travel at least $1 + 2 + \ldots + X = \Omega(X^2)$, so the competitive ratio is $\frac{\Omega(X^2)}{X} = \Omega(X)$. And by making X large, we can make the competitive ratio as large as we want.

2. Consider the strategy that starts at 0 and walks to $1, -1, 2, -2, 4, -4, 8, -8, \ldots,$ etc. until you find the keys. What is the competitive ratio?

   Solution: Suppose $X > 0$. Then let $X = 2^i + y$, where $y < 2^i$. Now the algorithm can be viewed as starting from 0, going to 1 and returning to 0, going to -1 and returning to 0, going to 2 and back, -2 and back, $\ldots$, 2^i and back, -2^i and back, and then finally reaching X. So the total distance traveled by the algorithm is

   $$2(1 + 1 + 2 + 2 + 4 + 4 + \ldots + 2^i + 2^i) + X = 4(2^{i+1} - 1) + X.$$

   If $X = -(2^i + y) < 0$, then we get an additional trip to 2^{i+1} and back on the positive side before reaching the (negative) X, so the total travel is $4(2^{i+1} - 1) + 2 \cdot 2^{i+1} + |X| = 12 \cdot 2^i + |X| - 4$.

   To get an upper bound on the comp.ratio, we want the algorithm to travel the most for the same value of $|X|$, so the worst-case input will choose a negative X. This gives us a comp.ratio of $12\frac{2^i}{|X|} + 1 - \frac{4}{|X|}$. And to maximize this, we should make $y = |X| - 2^i$ as small as possible, so set $y = 1$, and hence we get the comp.ratio $13 - o(1)$.

3. Consider the strategy that starts at 0 and walks to $1, -2, 4, -8, 16, -32$ etc. until you find the keys. What is the competitive ratio?

   Solution: Again, suppose $X = 2^i + y > 0$. This time we can think of us traveling to $1, 0, -2, 0, 4, 0, \ldots, 2^i, 0, -2^{i+1}, 0, X$. So the total distance is

   $$2(1 + 2 + \ldots + 2^{i+1}) + X = 2(2^{i+2} - 1) + X = 8 \cdot 2^i + X - 2.$$

   And if you do the calculation for negative Xs, you get a similar distance. So the competitive ratio this time is $\frac{8 \cdot 2^i + |X| - 2}{|X|} = 9 - o(1)$.

   Note: it is possible to show that any deterministic strategy cannot give a competitive ratio better than $9 - o(1)$. The proof requires some case analysis, and we're not doing it in this course.