CS 38 Introduction to Algorithms **Spring 2014**

Solution Set 7

Posted: June 3

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

1. Our LP will have variables x_i, one for each set, and the objective function is to maximize $\sum_i x_i$, subject to the following constraints:

   - $\sum_{i:u\in S_i} x_i \geq 1$ for each $u \in U$, and
   - $0 \leq x_i \leq 1$ for all i.

   After solving this LP, our solution will be to include set i for which $x_i \geq 1/k$.

   Let OPT be the value of the optimum set cover. Our solution is a set cover, because each constraint is the sum of at most k of the x_i, and thus at least one must be $\geq 1/k$. Moreover, its value is

   $$\sum_{i:x_i\geq 1/k} 1 \leq k \sum_{i:x_i\geq 1/k} x_i \leq k \sum_i x_i \leq k \cdot OPT.$$

   Where the last inequality uses the fact that the LP represents a relaxation of the set cover problem, so its optimium can only be smaller than the true OPT. Thus we have obtained a k-approximation as required.

2. Our LP will have variables x_v, one for each vertex v and $y_{u,v}$, one for each edge (u, v). If these variables were integer 0/1 variables, we would want each $y_{u,v}$ to be 1 iff edge (u, v) crossed the cut and the x_v values to encode which vertices appear on which side of the cut. Our objective function is thus to maximize $\sum_{(u,v)\in E} y_{u,v}$, subject to the following constraints:

   - $y_{u,v} \leq x_u + x_v$ for all edges $(u, v) \in E$, and
   - $y_{u,v} \leq (2 - x_u - x_v)$ for all edges $(u, v) \in E$, and
   - $0 \leq x_v \leq 1$ for all $v \in V$, and
   - $0 \leq y_{u,v} \leq 1$ for all $(u, v) \in E$.

   We solve the LP and now round to an integeral solution as follows. For each vertex v, we flip a coin r_v that comes up heads with probability x_v, and include vertex v on one side of the cut if the coin flip comes up heads and the other if it comes up tails.

   Let us compute the expected size of the cut obtained in this fashion:

   $$E[\text{size of cut}] = \sum_{(u,v)\in E} \Pr[r_u \neq r_v] = \sum_{(u,v)\in E} x_u(1 - x_v) + x_v(1 - x_u).$$

   We now claim that

   $$x_u(1 - x_v) + x_v(1 - x_u) \geq 1/2 \cdot \min\{x_u + x_v, 2 - x_u - x_v\}.$$

We have several cases: if $x_u, x_v \geq 1/2$, then

$$x_u(1 - x_v) + x_v(1 - x_u) \geq 1/2 \cdot (2 - x_u - x_v).$$

If $x_u, x_v \leq 1/2$, then

$$x_u(1 - x_v) + x_v(1 - x_u) \geq 1/2 \cdot (x_u + x_v).$$

Otherwise, we have $x_u \geq 1/2$ and $x_v \leq 1/2$ (and the other case when they are reversed is symmetric). We observe that $\min\{x_u + x_v, 2 - x_u - x_v\} \leq 1$, and that

$$x_u(1 - x_v) + x_v(1 - x_u) = x_u(1 - 2x_v) + x_v$$

is minimized when $x_u = 1/2$ and $x_v = 0$ (since $1 - 2x_v \geq 0$, we should always minimize x_u, and then once $x_u = 1/2$ it is best to set $x_v = 0$). Thus

$$x_u(1 - x_v) + x_v(1 - x_u) \geq 1/2 = 1/2 \cdot 1 \geq 1/2 \cdot \min\{x_u + x_v, 2 - x_u - x_v\}.$$

as desired.

Note also that the optimum value of the LP satisfies $y_{u,v} = \min\{x_u + x_v, 2 - x_u - x_v\}$ since there is nothing preventing us from increasing $y_{u,v}$ until one of its two constraints are tight. If OPT denotes the optimum size of a cut, we have

$$\begin{aligned} E[\text{size of cut}] &= \sum_{(u,v)\in E} x_u(1 - x_v) + x_v(1 - x_u) \geq 1/2 \cdot \sum_{(u,v)\in E} \min\{x_u + x_v, 2 - x_u - x_v\} \\ &= 1/2 \cdot \sum_{(u,v)\in E} y_{u,v} \geq 1/2 \cdot OPT. \end{aligned}$$

3. (a) We simply run through the edges of the graph in the order they appear in the adjacency list. We maintain a bit for each vertex indicating whether one of its incident edges belongs to the matching so far. Each time we encounter a new edge, we check the bits associated with the two endpoints. If neither are set, we add the edge, and set both bits, otherwise, we skip the edge. This required constant work per edge, and so it runs in time $O(E)$.

   (b) Observe that if M is a maximal matching, then its endpoints constitute a vertex cover (for if there was an edge not touched by this set of vertices, we could have added it to M). Thus there exists a vertex cover of cardinality twice the number of edges in M. And if v is the cardinality of the minimum vertex cover then $2|M| \geq v$. At the same time we claim v is an upper bound on the cardinality of a maximum matching. Since each vertex in a vertex cover can touch only one edge of a matching, the maximum matching can be no larger than v. We conclude that $2|M|$ is at least the cardinality of a maximum matching, as required.

4. Set $A = \sum_i a_i$. Observe that the number of bins used must be at least $\lceil A \rceil$. Thus OPT$\geq A$.

   We claim that our approximation algorithm fills all but the possibly one bin at least to 1/2 capacity. For if bin $i < j$ are both filled to at most 1/2 capacity, at the point that the first item was placed in j, it should have been placed in bin i (by our "first-fit" rule, and since its value is certainly at most 1/2).

Thus if the algorithm uses k bins, as we have noted all but one uses more than 1/2 of its capacity, and so we have $A > 1/2(k-1)$. We conclude that

$$k - 1 < 2A \leq 2 \cdot OPT$$

Since both k and OPT are integers, the strict inequality implies $k \leq 2 \cdot OPT$, as required.