CSE421: Design and Analysis of Algorithms November 26, 2021

Homework 7

Anup Rao Due: December 3, 2021

Read the fine print[1]. Each problem is worth 10 points:

1. Write down the dual of the following linear program:

$$\begin{aligned}&\text{maximize } a - b + c\\&\text{subject to}\\&\quad 5a + 2b \leq 3\\&\quad c - a \leq -2\\&\quad b + c \leq 0\\&\quad a, b, c \geq 0\end{aligned}$$

Solution:

$$\begin{aligned}&\text{minimize } 3x - 2y\\&\text{subject to}\\&\quad 5x - y \geq 1\\&\quad 2x + z \geq -1\\&\quad y + z \geq 1\\&\quad x, y, z \geq 0\end{aligned}$$

2. You are given the following points in the plane: $(1,3), (2,5), (3,7), (5,11), (7,14), (8,15), (10,19)$. You want to find a line $y = ax + b$ that approximately passes through these points (no line is a perfect fit). Write a linear program (you do not need to solve it) to find the line that minimizes the maximum absolute error,

$$\max_{1 \leq i \leq 7} |y_i - ax_i - b|$$

Solution:

[1] In solving the problem sets, you are allowed to collaborate with fellow students taking the class, but **each submission can have at most one author**. If you do collaborate in any way, you must acknowledge, for each problem, the people you worked with on that problem. The problems have been carefully chosen for their pedagogical value, and hence might be similar to those given in past offerings of this course at UW, or similar to other courses at other schools. Using any pre-existing solutions from these sources, for from the web, constitutes a violation of the academic integrity you are expected to exemplify, and is strictly prohibited. Most of the problems only require one or two key ideas for their solution. It will help you a lot to spell out these main ideas so that you can get most of the credit for a problem even if you err on the finer details. Please justify all answers. Some other guidelines for writing good solutions are here: `http://www.cs.washington.edu/education/courses/cse421/08wi/guidelines.pdf`.

$$
\begin{aligned}
&\text{minimize } c\\
&\text{subject to}\\
&\text{for all } i = 1, \ldots, 7,\\
&\qquad c \geq y_i - ax_i - b\\
&\qquad c \geq -y_i + ax_i + b
\end{aligned}
$$

In this linear program, c will be set to $\max_{1 \leq i \leq 7} |y_i - ax_i - b|$, since after fixing a, b, c the constraints force to be larger than all of the errors. So, the value of a, b in the above program gives the description of the best line.

3. You are running a truck business and need to fill a truck that can carry a total weight of 100 tons and volume 30 cubic meters. You can put three types of materials into the truck.

   (a) Item 1 has density 2 tons per cubic meter, maximum available amount is 40 cubic meters and the revenue associated with it is 1000 dollars per cubic meter.

   (b) Item 2 has density 5 tons per cubic meter, maximum available amount is 20 cubic meters and the revenue associated with it is 2000 dollars per cubic meter.

   (c) Item 3 has density 7 tons per cubic meter, maximum available amount is 15 cubic meters and the revenue associated with it is 1500 dollars per cubic meter.

   Write a linear program to calculate how much of each amount the truck should carry to maximize profits (no need to solve it).

   Solution: In the following linear program, a, b, c denote the volume of each of the materials that can be carried.

$$
\begin{aligned}
&\text{maximize } 1000a + 2000b + 1500c\\
&\text{subject to}\\
2a + 5b + 7c &\leq 100\\
a &\leq 40\\
b &\leq 20\\
c &\leq 15\\
a + b + c &\leq 30\\
a, b, c &\geq 0
\end{aligned}
$$

4. We all love vertex covers. The reason is that they are subsets of the vertices that touch every edge of an undirected graph. Give an algorithm to find a vertex cover of smallest size in a bipartite graph. Hints:

   (a) Construct a flow network from the input bipartite graph just as in the maximum matching algorithm.

(b) Show that a "nice" min-cut in this flow network gives a vertex cover. Namely, if the graph has bipartitions A, B and X, Y are the corresponding components of the "nice" min-cut, show that $(A - X) \cup (B - Y)$ must be a vertex cover of smallest size.

(c) Write down the algorithm and prove that it works.

Solution: The flow network will have vertices s, t as well as all the vertices of the bipartite graph. For every edge u on the left, the network has an edge (s, u) with capacity 1. For every edge v on the right, the network has the edge (v, t) of capacity 1 for every edge (u, v) from left to right, the network has the edge (u, v) with capacity ∞.

Now, as in the hint, we prove that every min-cut X, Y corresponds to a vertex cover $(A - X) \cup (B - Y)$ of smallest possible size. To see this, first we prove that given any vertex cover V of the original graph, we obtain an s, t cut in the flow network whose capacity is $|V|$. The s, t cut is given by the partition

$$\{s\} \cup (A - V) \cup (B \cap V), \{t\} \cup (A \cap V) \cup (B - V).$$

This is clearly a valid partition of the vertices of the flow network. The fact that V is a vertex cover implies that no edge of infinite capacity is cut. Such an edge would have to go from $A - V$ to $B - V$, and there are no such edges because V is a vertex cover. So, the capacity of this cut is finite. Moreover the number of edges from s that are cut is exactly $|A \cap V|$, and the number of edges into t that are cut is exactly $|B \cap V|$. So, the capacity of the cut is $|A \cap V| + |B \cap V| = |V|$. This proves that every vertex cover gives a cut whose capacity is the size of the vertex cover.

Finally, we show that every min-cut gives a vertex cover whose size is the capacity of the cut. Given any cut (X, Y) consider the set of vertices $(A - X) \cup (B - Y)$. We claim that this is a vertex cover. Indeed, since this is a min-cut, it cannot cut any edge of infinite capacity. This means that for every edge (u, v) from A to B, either $u \notin X$ or $v \notin Y$. Thus, (u, v) is covered by $(A - X) \cup (B - Y)$. The size of this vertex cover is $|A - X| + |B - Y|$, which exactly the same as the capacity of the cut (X, Y). So, the capacity of the min-cut is the same as the size of the smallest vertex cover.

The final algorithm is to run the capacity scaling algorithm on this flow network.