CS:5350 Homework 2, Fall 2019

Due in class on Tue, Oct 8th

Notes: (a) It is possible that solutions to some of these problems are available to you via text-books, on-line lecture notes, etc. If you use any such sources even partially, please acknowledge these in your homework fully *and* present your solutions in your own words. You will benefit most from the homework, if you seriously attempt each problem on your own first, before seeking other sources. (b) It is okay to form groups of **three** in solving and submitting homework solutions. (The syllabus says groups of at most two are allowed, but given the size of the class and the fact that the class has no TA, I am modifying this requirement to allow groups of three.) But, my advice from (a) still applies: you will benefit most from the homework, if you seriously attempt each problem on your own first, before seeking help from your group partner(s). (c) Unless you have a documented accomodation, no late submissions are permitted. You will receive no points for your submission if your submission is not turned in at the beginning of class on the due date. (d) Your submissions will be evaluated on correctness *and* clarity. Correctness is of course crucial, but how clearly you communicate your ideas is also quite important.

---

1. Let $G = (V, E)$ be an undirected graph. Let x_e be a variable for each edge $e \in E$. Consider the following "mystery" integer program:

$$\max \sum_{e \in E} x_e$$

subject to

$$\sum_{e : v \in e} x_e \leq 1 \qquad \text{for each } v \in V,$$

and

$$x_e \in \{0, 1\} \qquad \text{for each } e \in E.$$

   (a) This "mystery" integer program (in short, MIP) models a combinatorial optimization problem we have discussed in class. What problem is that?

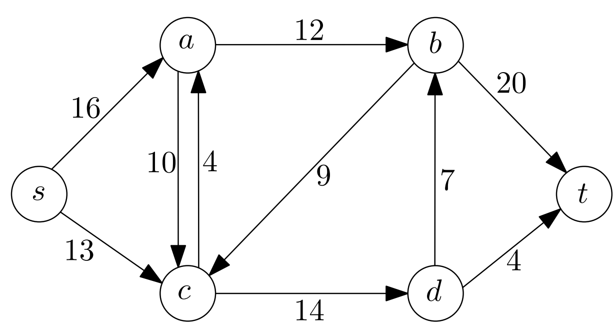
   (b) Obtain the LP relaxation of MIP by replacing the constraint $x_e \in \{0, 1\}$ by the non-negativity constraint $x_e \geq 0$, for all $e \in E$. Let us call this the "mystery" LP (in short, MLP). Draw a graph for which $OPT(MLP) > OPT(MIP)$ and show optimal solutions to MLP and MIP.

   (c) Prove that if G is a bipartite graph, then $OPT(MLP) = OPT(MIP)$.

   (d) Write down the dual linear program of MLP. Let us call this the dual "mystery" LP (in short, DMLP).

   (e) Replace the non-negativity constraints in DMLP by the integrality constraint that requires every dual variable to be in $\{0, 1\}$. Let us call the program we obtain the dual "mystery" IP (DMIP). Precisely state the combinatorial problem modeled by DMIP.

2. Consider the directed graph G given below, with source s and target t identified.



(a) For this graph, consider the *dual* of the flow LP. (This was discussed in class on Sep 19th and Sep 24th.) How many variables does this dual LP have? And how many constraints, exluding the non-negativity constraints, does it have?

(b) Describe a feasible solution to this LP with $y_e \in \{0, 1/3\}$ for each edge $e \in E$. What is the cost of your solution? Is your solution optimal? How do you know?

(c) Describe a solution to this LP that is (i) optimal and (ii) assigns a fractional value to at least one variable y_e *or* argue that such a solution is not possible.

3. Suppose you are given a set of n wireless devices, with positions $p_i = (x_i, y_i)$ for $i = 1, 2, \ldots, n$. The devices are cheap and the environment which they are placed in is hostile and so the devices are prone to failure. As a result, for each device v we want to find and assign $k \geq 1$ other "backup" devices that are within distance d meters of v. (One of the "backup" devices will do the work for v in case v fails. Also, we may want more than one "backup" device for each device v for redundancy in case v fails and its "backup" devices also start failing.) The one problem with this idea is that it is possible for a device to get overloaded if it serves as a "backup" for too many other devices. So for some positive integer parameter b, we want to ensure that no device serves as a "backup" for more than b other devices.

   Here is the problem. Given p_i, $1 \leq i \leq n$ and d, k, and b, describe a polynomial-time algorithm that determines if it is possible to assign "backup" device sets for all of the n devices. If this is indeed possible, your algorithm should return an assignment of "backup" device sets for each device.

4. The *edge connectivity* of an undirected graph is the minimum number k of edges that must be removed to disconnect the graph. For example, the edge connectivity of a tree is 1 and the edge connectivity of a cycle is 2. Show how the edge connectivity of an undirected graph $G = (V, E)$ can be determined by running a maximum-flow algorithm on at most $|V|$ flow networks, each having $O(|V|)$ vertices and $O(|E|)$ edges.

5. Problem 14 from Chapter 10 in Prof. Erickson's notes. Try to make your algorithms as efficient as possible and state their running times.
