cs473: Algorithms   Assigned: *Wed., Nov. 13, 2019*

Problem Set #9

Prof. Michael A. Forbes
Prof. Chandra Chekuri   Due: *Wed., Nov. 20, 2019 (10:00am)*

For problems that ask for a linear-programming formulation of some problem, a full-credit solution requires the following components:

- A list of variables, along with a brief in-words description of what each variable represents.

- A linear objective function (expressed either as minimization or maximization, whichever is more appropriate), along with a brief in-words description of its meaning.

- A sequence of linear inequalities (expressed using $\leq$, $=$, or $\geq$, whichever is more appropriate), along with a brief in-words description of what each constraint represents.

- A proof that your linear programming formulation is correct, meaning that the optimal solution to the original problem can always be obtained from the optimal solution to the linear program. This may be very short.

It is **not** necessary to express the linear program in canonical form, or even in matrix form. Clarity is much more important than formality.

All (non-optional) problems are of equal value.

1. Consider the following scheduling problem. There are n jobs that need to be scheduled on m identical machines, where each job takes one unit of time to complete. Each job j has a release time r_j and a deadline d_j both of which are integers. A schedule assigns each job to a time slot $[t, t+1]$ for some integer t and to some machine such that no two jobs are assigned to the same slot on the same machine. If job j is completed by its deadline there is no penalty; if it completes at time $C_j > d_j$ then it incurs a penalty of $C_j - d_j$. A job cannot be scheduled before it is release time. The goal is find a schedule for all of the jobs on the given machines so as to minimize the total penalty.

   Describe a polynomial time algorithm for this problem via reduction to minimum-cost max-flow. Be sure to pay attention to the fact that your algorithm runs in polynomial time by examining carefully the size of the input.

2. Flows are typically defined as a function on the edges, but the path-based definition is useful and necessary in various applications. Let $G = (V, E)$ be a directed graph with non-negative capacities $c : E \to \mathbb{R}_{\geq 0}$. Given distinct nodes $s, t \in V$ let $P_{s,t}$ denote the set of all simple paths between s and t.

   (a) Write the maximum (s,t)-flow problem as a linear programming problem with one variable for each path $p \in P_{s,t}$. Note that the primal can have an exponential (in $|V|$) number of variables. Write its dual.

(b) Suppose we are given an assignment of values to the variables in the dual. Show that, even though there are possibly exponentially-many constraints, one can use a shortest path algorithm to efficiently check that the values satisfy all the constraints of the dual.

3. The facility location problem is the following. There is a set of n facilities F and m clients C. The cost of opening a facility i is f_i. There is a cost $c(i, j)$ to connect client j to facility i. The goal is to open a subset of the facilities and connect each client to an open facility. The objective function is to minimize the sum of the facility opening costs and the connection costs.

Write an integer linear programming formulation for this problem using two sets of decision variables, one set for opening facilities, and one set for assigning clients to open facilities. Prove that it is sufficient to constrain only the facility opening variables to be integer valued. Such a problem is called a mixed-integer linear programming problem.