CSORE 4231 ANALYSIS OF ALGORITHMS I Homework 6

Francis Zhang xz3279

September 19, 2024

1. Minimum-cost circulation

CLRS 29-5

In this problem, we consider a variant of the minimum-cost-flow problem from Section 29.2 in which we are not given a demand, a source, or a sink. Instead, we are given, as before, a flow network and edge costs $a(u, v)$. A flow is feasible if it satisfies the capacity constraint on every edge and flow conservation at every vertex. The goal is to find, among all feasible flows, the one of minimum cost. We call this problem the **minimum-cost-circulation problem**.

a. Formulate the minimum-cost-circulation problem as a linear program.

b. Suppose that for all edges $(u, v) \in E$, we have $a(u, v) > 0$. Characterize an optimal solution to the minimum-cost-circulation problem.

c. Formulate the maximum-flow problem as a minimum-cost-circulation problem linear program. That is given a maximum-flow problem instance $G = (V, E)$ with source s, sink t and edge capacities c, create a minimum-cost-circulation problem by giving a (possibly different) network $G' = (V', E')$ with edge capacities c' and edge costs a' such that you can discern a solution to the maximum-flow problem from a solution to the minimum-cost-circulation problem.

d. Formulate the single-source shortest-path problem as a minimum-cost-circulation problem linear program.

2. Integer linear programming

Given an integer $m \times n$ matrix A and an integer vector b of dimension m, the 0-1 integer-programming problem asks whether there exists an integer vector x of dimension n with elements in the set $\{0, 1\}$ such that $Ax \leq b$. Prove that 0-1 integer programming is NP-complete. (Hint: Reduce from 3-SAT)

3. Bonnie and Clyde

CLRS 34-2

Bonnie and Clyde have just robbed a bank. They have a bag of money and want to divide it up. For each of the following scenarios, either give a polynomial-time algorithm, or prove that the problem is NP-complete. The input in each case is a list of the n items in the bag, along with the value of each.

a. The bag contains n coins, but only 2 different denominations: some coins are worth x dollars, and some are worth y dollars. Bonnie and Clyde wish to divide the money exactly evenly.

b. The bag contains n coins, with an arbitrary number of different denominations, but each denomination is a nonnegative integer power of 2, i.e., the possible denominations are 1 dollar, 2 dollars, 4 dollars, etc. Bonnie and Clyde wish to divide the money exactly evenly.

c. The bag contains n checks, which are, in an amazing coincidence, made out to “Bonnie or Clyde.” They wish to divide the checks so that they each get the exact same amount of money.

d. The bag contains n checks as in part (c), but this time Bonnie and Clyde are willing to accept a split in which the difference is no larger than 100 dollars.

4. Vertex Cover approximation

Consider the following algorithm for Vertex Cover. Start with $C = \emptyset$. At each step, choose the vertex v with the highest degree, add it to the cover C, and then delete v and all its incident edges from the graph. Stop when the graph is empty. Does this algorithm give a 2-approximation for vertex cover? Either prove that it does, or show a counterexample where it does not.