CMPSC 565: Design and Analysis of Algorithm (**Due 2024-10-25 10:00pm**)

Problem Set 3

Instructor: Dr. Antonio Blanca TAs: Qiyu Chen, Jianqiang Li

Notice: Type your answers using LaTeX and make sure to upload the answer file on Gradescope before the deadline. Recall that for any problem or part of a problem, you can use the "I'll take 20%" option. For more details and the instructions read the syllabus.

Problem 1. Standard Form

Write the following linear program in the standard form given in the lecture notes.

$$\begin{aligned} \max\ x + y \\ 2x + y \leq 3 \\ x + 3y \leq 5 \\ x \geq 0 \end{aligned}$$

Problem 2 Duality

(1) Write the dual of the following linear program.

$$\begin{aligned} \max\ x + y \\ 2x + y \leq 3 \\ x + 3y \leq 5 \\ x, y \geq 0 \end{aligned}$$

(2) Find the optimal value and vertex of both the primal and dual LPs.

Problem 3. Simplex Algorithm

Given the following linear program, compute the optimum value via the Simplex Algorithm. Use the variant from class and choose the axis with the highest positive coefficient each iteration. Show your work by demonstrating the whole transformed linear system at each intermediate vertex.

$$\begin{aligned} \max\ 3x_1 - 2x_2 - x_4 \\ x_1 + 3x_3 - 2x_4 \leq 6 \\ 2x_2 - x_3 + x_4 \leq 4 \\ x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

Problem 4. Coverage Gap

Given a graph $G = (V, E)$, a *vertex cover* of G is a subset $S \subseteq V$ of nodes such that every edge in E is incident to *at least* one node of S. (S "covers" the edges of E.) The size of the vertex cover is $|S|$.

(1) Give a linear program such that the optimal integral solution (that is, a solution where all the variables are assigned integer values) gives a vertex cover of minimum size.

(2) If G is a general graph, the optimal solution to the LP is not necessarily integral, which means that it may not correspond to a vertex cover in G. Give an example of a graph G where the optimal solution to the LP you give in part (a) is not integral, and give the optimal solution. (This phenomenon is called the integrability gap)

Problem 5. Oxygen Included

A spaceship uses some *oxidizer* units that produce oxygen for three different compartments. However, these units have some failure probabilities. Because of differing requirements for the three compartments, the units needed for each have somewhat different characteristics.

A decision must now be made on just *how many* units to provide for each compartment, taking into account design limitations on the *total* amount of *space*, *weight* and *cost* that can be allocated to these units for the entire ship. Specifically, the total space for all units in the spaceship should not exceed 500 cubic inches, the total weight should not exceed 200 lbs and the total cost should not exceed $400,000$ dollars.

The following table summarizes the characteristics of units for each compartment and also the total limitation:

	Space (cu in.)	Weight (lb)	Cost ($)	Probability of failure
Units for compartment 1	40	15	30,000	0.30
Units for compartment 2	50	20	35,000	0.40
Units for compartment 3	30	10	25,000	0.20
Limitation	500	200	400,000	

The objective is to *minimize the probability* of all units failing in all three compartments, subject to the above limitations and the further restriction that each compartment have a probability of no more than 0.05 that all its units fail.

Formulate the *linear programming model* for this problem.

Problem 6. Generalization of the maximum flow problem

You are given a flow network $G = (V, E)$ with edge capacities c_e for each $e \in E$. You are given multiple pairs $(s_1, t_1), (s_2, t_2), \cdots, (s_k, t_k)$, where the s_i's are sources of G and the t_i's are sinks of G. You are also given k demands $d_1, \ldots, d_k$. The goal is to find k flows $f^{(1)}, \cdot, f^{(k)}$ such that:

- Each $f^{(i)}$ is a valid flow from s_i to t_i.
- For each edge e, the total flow through the edge $\sum_{i=1}^{k} f^{(i)}(e)$ does not exceed the capacity $c(e)$.
- The value of each flow $f^{(i)}$ is at least the demand d_i.
- The value of the total flow (the sum of the flows) is as large as possible.

Write a Linear Program to solve this problem. Explain you choice of variables, constraints, and objective function.

Problem 7. Brute Force

Show that for any problem $\Pi \in$ NP, there is an algorithm which solves Π in time $O(2^{p(n)})$, where n is the size of the input instance and $p(n)$ is a polynomial (which may depend on Π).

Problem 8. 4SAT

Assume that you are given a set of clauses where each of which is a disjunction of exactly 4 literals, and such that each variable occurs at most once in each clause. The goal is to find a satisfying assignment (if one exists). Prove that this problem is NP-complete.

Hint: Reduce 3SAT to this problem. Think about how to convert a clause from a 3SAT problem to an equivalent clause in this problem.

Problem 9. Dominating set

In an undirected graph $G = (V, E)$, we say $D \subseteq V$ is a dominating set if every $v \in V$ is either in D or adjacent to at least one member of D. In the DOMINATING SET problem, the input is a graph and a budget k', and the aim is to find a dominating set in the graph of size at most k', if one exists. Prove that this problem is NP-complete.

Hint: This problem is similar to the vertex cover problem.

Problem 10. That's Bing

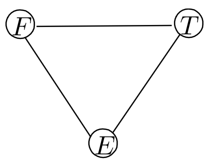
Suppose you have a procedure which runs in polynomial time and tells you whether or not a graph has a Hamiltonian path (given a graph $G = (V, E)$, whether there exists any $s, t \in V$ s.t. there exists a Hamiltonian-(s, t)-path in G). Show that you can use it to develop a polynomial-time algorithm for the Hamiltonian Path (which returns the actual path, if it exists).

Problem 11. Tricolor

The k-coloring problem is the problem of deciding whether a graph $G = (V, E)$ is k-colorable – whether there exists an assignment c of colors $\{1, \ldots, k\}$ to the vertices V such that $c[u] \neq c[v]$ $\forall (u, v) \in E$. That is, in a proper coloring, the endpoints of every edge are assigned different colors. Please follow the steps below to prove that 3COLOR, the 3-coloring problem, is NP-complete by reducing 3SAT to it.

(1) Prove that 3COLOR is in NP.

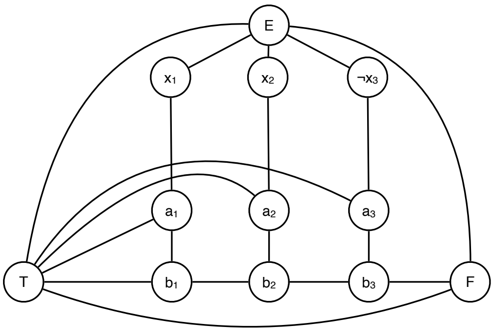
(2) Consider the following labeled 3-clique, denoted K_3:



This graph has 3 nodes, and each node must be colored with a different color. We can use it to label the colors $\{1, 2, 3\}$ in a 3-coloring of K_3 as (T)rue, (F)alse, or (E)xcluded; that is, if vertex F is assigned color 2, then we associate color 2 with False.

Given a set of n boolean variables $x_1 \ldots x_n$, construct a graph G and a bijection between valid 3-colorings of G and all 2^n possible assignments of the boolean variables.

(3) Connecting One OR Gadget: Consider the following 12-node graph.



i . Argue that T, F, E all have different colors and that $x_1, x_2, \bar{x}_3$ must be colored with T or F.

ii . Prove that there is no valid 3-coloring of the graph where $\{x_1, x_2, \bar{x}_3\}$ all have the color F.

iii . Prove that for each other coloring of $\{x_1, x_2, \bar{x}_3\}$ (with T or F) there exists a compatible 3-coloring of the graph.

(4) Given a set of n boolean variables $x_1 \ldots x_n$ and a set of m disjunctive clauses $c_1 \ldots c_m$ with three literals, describe a poly-time procedure that produces a graph which is 3-colorable if and only if all m clauses are satisfiable. Hint: connect multiple OR gadgets from Part (3) to the graph you made in Part (2).

(5) Prove the correctness of your reduction and prove that it is a poly-time reduction. You should use the results from Part (3) in your proof.