Assignment 4

Due April 8 2022, 11:59pm

Q1 [25 Points] Approximate Vertex Cover

Recall the vertex-cover problem. Given a graph $G = (V, E)$, a vertex cover is a subset of a vertices $S \subseteq V$ so that every edge has at least one endpoint in S.
Let us write a solution to the minimum vertex cover problem as an ILP created by assigning a variable x_u for each vertex u and solving the following linear program.

$$\begin{aligned} &\min 0.5 * \sum_{v \in V} x_v \\ &x_u + x_v \geq 2 \text{ for each edge } (u, v) \\ &x_u \geq 0 \text{ for each vertex} \end{aligned} \tag{1}$$

Recall also that a 4-colouring of a graph G is a function $\kappa : V \mapsto \{0, 1, 2, 3\}$ so that vertices connected by an edge get different colours. In this question, assume that the graph G has a 4-colouring and let V_i be the number of vertices coloured i. Without loss of generality, assume that $|V_3| \geq |V_2| \geq |V_1| \geq |V_0|$. (A famous result in mathematics shows that 4-colourings of many graphs can be found efficiently).

(a) [1 Points] What are the possible values that any variable x_v can take in an optimal solution to the above ILP and why?

(b) [10 Points] Consider the following algorithm to obtain a vertex covering of a 4-colorable graph G. Instead of solving the ILP shown,we solve it as an LP, where each vertex v in the optimal solution has value x_v^*. Then we return a set S with all vertices with $x_v^* \geq 1$ except when $x_v^* = 1$ and $v \in V_3$. Argue that set S returned by this algorithm is indeed a vertex cover of G.

(c) [10 Points] Let x^* be the optimal obective value of the LP solution. Show that for the vertex cover S computed in part (b), $|S| \leq \frac{3}{2}x^*$ and that the algorithm gives a $\frac{3}{2}$-approximation to the minimum vertex cover for 4-colorable graphs.

(d) [4 Points] Adapt the algorithm of part (b) to provide a vertex cover, given a graph G and a k-coloring of G. What is the worst case approximation ratio of this vertex cover to the minimum vertex cover for G?

Q2 [20 Points] Nearly-SAT

Nearly-SAT is the problem described as follows:

- Input: A CNF Formula $\mathcal{F}$ over n variables $x_1, \dots, x_n$ and having m clauses
- Output: "Yes" if there is an assignment of the variables $x_1, \dots, x_n$ that satisfies exactly $m-1$ clauses, "No" otherwise

(a) [5 Points] Show that Nearly-SAT is in NP.

(b) [7 Points] Given a CNF formula $\mathcal{F}$ over the variables $x_1, \dots, x_n$ and which has m clauses, construct a CNF formula $\mathcal{F}'$ over the same set of variables and which has two additional clauses (so $m+2$ in all) such that an assignment to the variables $x_1, \dots, x_n$ that satisfies all m clauses of $\mathcal{F}$ satisfies precisely $m+1$ clauses of $\mathcal{F}'$.

(c) [8 Points] Use part (b) to give a polynomial time reduction from SAT to Nearly-SAT and show that Nearly-SAT is NP-hard.

Q3 [20 Points] Tile Covering

An architect is given a set of rectangular tiles t_i, each with some length l_i and width w_i ($l_i \geq w_i$). There is a wall with length l and width w, and the architect would like to see if there is a way to cover the wall with tiles so that each tile is placed horizontally or vertically, there are no overlaps between the tiles or gaps on the wall, and furthermore each tile is placed on the wall. Assume that all dimensions are given as rational numbers.

(a) [5 Points]The **Architect** problem is defined as the following decision problem: given the tiles and the dimensions of the wall, is there a way to cover the wall according to constraints? Show that the Architect problem is in NP.

(b) [15 Points] Show that the Architect problem is NP-Complete. You may assume that the following problem is NP-Complete:
Partition: Given a set of positive integers S that sum to n, is there a subset $S_1 \subseteq S$ that sums to $\frac{n}{2}$?