

## CS 170 Homework 11

Due 4/18/2022, at 10:00 pm (grace period until 11:59pm)

### 1 Study Group

List the names and SIDs of the members in your study group. If you have no collaborators, you must explicitly write “none”.

### 2 Max $k$ -XOR

In the Max  $k$ -XOR problem, we are given  $n$  boolean variables  $x_1, x_2, \dots, x_n$ , a list of  $m$  clauses each of which is the XOR of exactly  $k$  distinct variables (that is, the clause is true if and only if an odd number of the  $k$  variables in the clause are true), and an integer  $r$ . Our goal is to decide if there is some assignment of variables that satisfies at least  $r$  clauses.

- (a) In the Max Cut problem, we are given an undirected unweighted graph and integer  $c$  and want to find a cut  $S$  such that at least  $c$  edges cross this cut (i.e. have exactly one endpoint in  $S$ ). Give and argue correctness of a reduction from Max-Cut to Max 2-XOR.
- (b) Give and argue correctness of a reduction from Max 3-XOR to Max 4-XOR.

### 3 Dominating Set

A dominating set of a graph  $G = (V, E)$  is a subset  $D$  of  $V$ , such that every vertex not in  $D$  is a neighbor of at least one vertex in  $D$ . Let the Minimum Dominating Set problem be the task of determining whether there is a dominating set of size  $\leq k$ . Show that the Minimum Dominating Set problem is NP-Complete. You may assume that  $G$  is connected.

*Hint: Try reducing from Vertex Cover or Set Cover.*

### 4 Independent Set Approximation

In the Max Independent Set problem, we are given a graph  $G = (V, E)$  and asked to find the largest set  $V' \subseteq V$  such that no two vertices in  $V'$  share an edge in  $E$ .

Given an undirected graph  $G = (V, E)$  in which each node has degree  $\leq d$ , give an efficient algorithm that finds an independent set whose size is at least  $1/(d+1)$  times that of the largest independent set. Only the main idea and the proof that the size is at least  $1/(d+1)$  times the largest solution's size are needed.

## 5 Coffee Shops

A rectangular city is divided into a grid of  $m \times n$  blocks. You would like to set up coffee shops so that for every block in the city, either there is a coffee shop within the block or there is one in a neighboring block. (There are up to 4 neighboring blocks for every block). It costs  $r_{ij}$  to rent space for a coffee shop in block  $ij$ .

Write an integer linear program to determine which blocks to set up the coffee shops at, so as to minimize the total rental costs.

- (a) What are your variables, and what do they mean?
- (b) What is the objective function?
- (c) What are the constraints?
- (d) Solving the non-integer version of the linear program gets you a real-valued solution. How would you round the LP solution to obtain an integer solution to the problem? Describe the algorithm in at most two sentences.
- (e) What is the approximation ratio obtained by your algorithm? Briefly justify.

## 6 Orthogonal Vectors (optional)

In the 3-SAT problem, we have  $n$  variables and  $m$  clauses, where each clause is the OR of (at most) three of these variables or their negations. The goal of the problem is to find an assignment of variables that satisfies all the clauses, or correctly declare that none exists.

In the orthogonal vectors problem, we have two sets of vectors  $A, B$ . All vectors are in  $\{0, 1\}^m$ , and  $|A| = |B| = n$ . The goal of the problem is to find two vectors  $a \in A, b \in B$  whose dot product is 0, or correctly declare that none exists. The brute-force solution to this problem takes  $O(n^2m)$  time: We compute all  $|A||B| = n^2$  dot products between two vectors in  $A, B$ , and each dot product takes  $O(m)$  time.

Show that if there is a  $O(n^c m)$ -time algorithm for the orthogonal vectors problem for some  $c \in [1, 2)$ , then there is a  $O(2^{cn/2} m)$ -time algorithm for the 3-SAT problem. For simplicity, you may assume in 3-SAT that the number of variables must be even. *Hint: Try splitting the variables in the 3-SAT problem into two groups.*