CS 501 / MCS 501 – Computer Algorithms I

Spring 2024

Problem Set 4

Lev Reyzin

Due: 4/5/24 by the beginning of class

Instructions: Atop your answer sheet, write your name and the names of all the students with whom you have collaborated on this problem set. All answers require explanation or proof.

1. [20 pts] For $k > 2$, denote by C_k the cycle on k vertices.

- **a. [5 pts]** What is the value of the MAX-CUT in C_k for every $k > 2$?
- **b. [10 pts]** Show that the value of the SDP for MAX-CUT is at least $k\left(1 - \cos\left(\pi\left(1 - 1/k\right)\right)\right)/2$.
- **c. [5 pts]** What is the largest integrality gap for graphs C_k that you can find?

2. [25 pts] In class, we saw that an LP approach gives a 3/4 approximation to MAX-2-SAT. In this problem, we will improve on this guarantee using semidefinite programming. For a formula $\phi(x_1, \ldots x_n)$ in 2-CNF, we create constraints $y_i \in \{-1, 1\}$, where y_i represents every variable x_i (and $-y_i$ for $\bar{x}_i$). We also create a variable y_0 to represent whether -1 or 1 stands for *true*. For clauses in the form $(x_i \vee x_j)$, we can maximize $\frac{3+y_i y_0+y_j y_0-y_i y_j}{4}$, which will evaluate to 1 if the clause is satisfied (analogous expressions exist for clauses with negations). Summing this over all clauses would solve the problem, but the constraints $y_i \in \{-1, 1\}$ preclude this from being an SDP.

- **a. [5 pts]** Formulate the above as an optimization problem and give an explicit SDP relaxation.
- **b. [10 pts]** Use the above to give a randomized .878 approximation algorithm for MAX-2-SAT.
- **c. [10 pts]** Using the result in part b. (even if you were unable to prove it), improve upon the 3/4 approximation algorithm for MAX-SAT that we obtained from LP-based methods. (*Hint: try combining the result in b. with the "better of two"* 3/4 *approximation algorithm.*)

3. [20 pts] Give polynomial time algorithms for the following problems. Note, these are not questions about semidefinite programming, but rather about problems whose variants are NP hard and have SDP-based approximation algorithms.

- **a. [10 pts]** Deciding whether all clauses in a MAX-2-SAT instance are simultaneously satisfiable.[1] (*Hint: recall that* $(x_i \vee x_j)$ *is equivalent* $(-x_i \rightarrow x_j) \wedge (-x_j \rightarrow x_i)$.)
- **b. [5 pts]** 2-coloring a 2-colorable graph.
- **c. [5 pts]** Coloring a graph with maximum degree Δ using $\Delta + 1$ colors.

[1] While the MAX-2-SAT problem is NP-hard, this variant of it is in P.