CS 170 Homework 12

Due **Monday 4/22/2024, 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 Approximating Independent Set

In the maximum independent set (MIS) problem, we are given a graph $G = (V, E)$, and our goal is to find the largest set of vertices such that no two vertices in the set are connected by an edge. For this problem, we will assume the degree of all vertices in G is bounded by d (i.e. $\forall v \in V,\ \deg(v) \leq d$).

Consider the following greedy algorithm to approximate the maximum independent set:

```
procedure GREEDY-MIS($V, E$)
    $I \leftarrow \emptyset$
    while $G \neq \emptyset$ do
        choose a vertex $v$ in $G$
        $I = I \cup \{v\}$
        remove $v$ and all its neighbors from $G$
    return $I$
```

(a) Provide an example where the greedy approximation does not give the optimal solution.

(b) Provide an approximation ratio for the given greedy algorithm in terms of $|V|$, $|E|$, and/or d. Briefly justify your answer.

3 $\sqrt{n}$ coloring

(a) Let G be a graph of maximum degree δ. Show that G is $(\delta + 1)$-colorable.

(b) Suppose $G = (V, E)$ is a 3-colorable graph. Let v be any vertex in G. Show that the graph induced on the neighborhood of v is 2-colorable.

Note: the graph induced on the neighborhood of v refers to the following subgraph:

$$G' = (V' = \text{neighbors of } v, E' = \text{all edges in } E \text{ with both endpoints in } V').$$

(c) Give a polynomial time algorithm that takes in a 3-colorable n-vertex graph G as input and outputs a valid coloring of its vertices using $O(\sqrt{n})$ colors. Prove that your algorithm is correct.

Hint: think of an algorithm that first assigns colors to "high-degree" vertices and their neighborhoods, and then assigns colors to the rest of the graph. The previous two parts might be useful.

4 Multiway Cut

In the multiway cut problem, we are given a graph $G = (V, E)$ with k special vertices $s_1, s_2, \ldots, s_k$. Our goal is to find the smallest set of edges F which, when removed from the graph, disconnect the graph into at least k components, where each s_i is in a different component. When $k = 2$, this is exactly the min s-t cut problem, but if $k \geq 3$ the problem becomes NP-hard.

Consider the following algorithm: Let F_i be the set of edges in the minimum cut with s_i on one side and all other special vertices on the other side. Output F, the union of all F_i. Note that this is a multiway cut because removing F_i from G isolates s_i in its own component.

(a) Explain how each F_i can be found in polynomial time.

(b) Let F^* be the smallest multiway cut. Consider the components that removing F^* disconnects G into, and let C_i be the set of vertices in the component with s_i. Let F_i^* be the set of edges in F^* with exactly one endpoint in C_i. How many different F_i^* does each edge in F^* appear in? Which is larger: F_i and F_i^*?

(c) Using your answer to the previous part, show that $|F| \leq 2|F^*|$.

(d) **Extra Credit:** how could you modify this algorithm to output F such that $|F| \leq (2 - \frac{2}{k})|F^*|$?

5 Relaxing Integer Linear Programs

As discussed in lecture, Integer Linear Programming (ILP) is NP-complete. In this problem, we discuss attempts to approximate ILPs with Linear Programs and the potential shortcomings of doing so.

Throughout this problem, you may use the fact that the ellipsoid algorithm finds an optimal vertex (and corresponding optimal value) of a linear program in polynomial time.

(a) Suppose that $\vec{x}_0$ is an optimal point for the following arbitrary LP:

$$\begin{aligned} \text{maximize } & c^\top x \\ \text{subject to: } & Ax \leq b \\ & x \geq 0 \end{aligned}$$

Show through examples (i.e. by providing specific canonical-form LPs and optimal points) why we cannot simply (1) round all of the element in $\vec{x}_0$, or (2) take the floor of every element of $\vec{x}_0$ to get good integer approximations.

(b) The Matching problem is defined as follows: given a graph G, determine the size of the largest subset of disjoint edges of the graph (i.e. edges without repeating incident vertices).

Find a function f such that:

$$\begin{aligned} \text{maximize } & f \\ \text{subject to: } & \sum_{e \in E, v \in e} x_e \leq 1 \qquad \forall v \in V \\ & 0 \leq x_e \leq 1 \qquad \forall e \in E \end{aligned}$$

is an LP relaxation of the Matching problem. Note that the ILP version (which directly solves Matching) simply replaces the last constraint with $x_e \in \{0, 1\}$.

(c) It turns out that the polytope of the linear program from part (b) has vertices whose coordinates are all in $\{0, \frac{1}{2}, 1\}$. Using this information, describe an algorithm that approximates Matching and give an approximation ratio with proof.

Hint: round up, then fix constraint violations.

(d) There is a class of linear program constraints whose polytopes have only integral coordinates. Let $\mathcal{P}_{>2,\text{odd}}(V)$ be the set of subsets of the vertices with size that is odd and greater than 2. It turns out that, if we simply add to the LP from part (b) the following constraints:

$$\sum_{e \in E(S)} x_e \leq \frac{|S| - 1}{2} \qquad \forall S \in \mathcal{P}_{>2,\text{odd}}(V),$$

then all vertices of the new polytope are integral. First, interpret this constraint in words and explain why it still describes the Matching problem. Then, explain what this result implies about approximating ILPs with (special) LPs.

(e) Why doesn't the observation in part (d) imply that MATCHING $\in \mathsf{P}$?