15-251: Great Theoretical Ideas In Computer Science

Recitation 9 : NP-Hardness and Approximation Algorithms

More or Less Review

- A problem Y is **NP-hard** if for every problem $X \in \text{NP}$, $X \leq_m^P Y$.
- A problem is **NP-complete** if it is both in NP and NP-hard.
- The goal of an optimization problem is to find the minimum (or maximum) value under some constraints.
- OPT(I) is the value of the optimal solution to an instance I of an optimization problem.
- We say an algorithm $\mathcal{A}$ for an optimization problem is a factor-α approximation if for all instances I of the problem $\mathcal{A}$ outputs a solution that is at least as good as $\alpha \cdot \text{OPT}(I)$.

Edge Cover-Up

Let $G = (V, E)$ be a graph. A vertex covering of G is a set $C \subseteq V$ such that for every edge $\{x, y\} \in E$, either $x \in C$ or $y \in C$ (a set of vertices such that every edge is incident to at least one vertex in the set). An independent set in G is a set $S \subseteq V$ such that for any $u, v \in S$, $\{u, v\} \notin E$ (a set of vertices such that no edge connects two vertices in the set). Define the following languages:

VERTEX-COVER: $\{\langle G, k\rangle : G$ is a graph, $k \in \mathbb{N}^+$, G contain a vertex covering of size $k\}$

IND-SET: $\{\langle G, k\rangle : G$ is a graph, $k \in \mathbb{N}^+$, G contains an independent set of size $k\}$

Show that VERTEX-COVER $\leq_m^P$ IND-SET.

Cut and Dried

We define the Max-Cut problem as follows:

Let $G = (V, E)$ be a graph. Given a coloring of the vertices with 2 colors, we say that an edge $e = \{u, v\}$ is *cut* if u and v are colored differently. In the *Max-Cut problem*, the input is a graph G, and the output is a coloring of the vertices with 2 colors that maximizes the number of cut edges.

Consider the following approximation algorithm for the Max-Cut problem:

```
function MaxCutApprox(G = (V, E))
    color all vertices red          ▷ allowable colors are red and something else (say, blue)
    while there exists v ∈ V s.t. changing v's color increases the number of cut edges do
        change the color of v
    end while
    return coloring
end function
```

(a) Show that this algorithm is poly-time.

(b) Prove that this algorithm is a $\frac{1}{2}$-approximation for Max-Cut.

(c) Show that this algorithm is not a $(\frac{1}{2} + \varepsilon)$-approximation algorithm for Max-Cut for any $\varepsilon > 0$.

Gotta Catch a Lot of 'Em

Consider a set of Pokémon and a set of m trainers each having a subset of these Pokémon. Given an integer k, the problem is to pick k trainers in a way that maximizes the number of distinct Pokémon owned among them. This problem will show that there exists a poly-time $(1 - 1/e)$-approximation by considering the following greedy algorithm:

```
function PokémonApprox((S_1, S_2, ..., S_m), k)
    T ← ∅                          ▷ keeps track of trainers we have already picked
    U ← ∅                          ▷ keeps track of which Pokémon we have already
    for 1 ≤ i ≤ k do
        j ← arg max_j |S_j − U|    ▷ pick the trainer j with the most new Pokémon
        T ← T ∪ {j}
        U ← U ∪ S_j
    end for
    return T
end function
```

(a) Prove that the algorithm runs in poly-time.

(b) Let T^* denote the optimum solution, and let $U^* = \bigcup_{j \in T^*} S_j$. Further, define U_i to be the set U in the algorithm after the i-th iteration of the loop. Prove that $|U^*| - |U_i| \leq (1 - \frac{1}{k})^i |U^*|$.

(c) Using the inequality $1 + x \leq e^x$, deduce that this algorithm is a $(1 - \frac{1}{e})$-approximation.

(Extra) Looping Around

Show that the HALTS is NP-hard.

(Bonus) Hard Cut

On the previous page, we defined Max-Cut as an optimization problem. We can also define the decision version MAX-CUT as follows:

MAX-CUT: $\{\langle G, k \rangle : G$'s vertices may be colored with two colors in a way that cuts at least k edges$\}$.

Prove that MAX-CUT is NP-hard. This is slightly difficult; try reducing from IND-SET.