CS6841: Approximation Algorithms

Assignment 1

Name:

Roll number:

Instructions

- **Deadline**: 20 Feb (in class during Midsem)
- **References:** Williamson and Shmoys (`https://www.designofapproxalgs.com/book.pdf`), Approximation Algorithms by Vijay Vazirani, CLRS

1. (5 points) **k-suppliers Problem** The input to the problem is a positive integer k, and a set of vertices V. Let $d(u, v)$ represents the shortest path between any two vertices $u, v \in V$ and obey the same properties as in the Facility Location Problem. The vertices are partitioned into suppliers $F \subseteq V$ and customers $C = V \setminus F$.

   The goal is to find k suppliers such that the maximum distance from a supplier to a customer is minimized. In other words, the problem is to find $S \subseteq F$, $|S| \leq k$, that minimizes $\max_{j \in C} d(j, S)$.

   (a) (3 points) Give a 3-approximation algorithm for the k-suppliers problem.

   (b) (3 points) Prove that there is no α-approximation algorithm for $\alpha < 3$ unless P = NP.

   Solution: Write your solution here

2. (5 points) **Job-Scheduling with Precedence** This problem is a variant of the job scheduling (in multiple processors) problem, that was discussed in class. Here, along with the n jobs and m processors, we also have precedence constraint between the jobs.

   Given are n jobs $\mathcal{J} = \{j_1, j_2, \cdots, j_n\}$, with processing times $t_1, t_2, \cdots, t_n$ and m processors $P_1, P_2, \cdots, P_n$. The precedence constraints between jobs is given in the form of a partial order $(\mathcal{J}, <)$ on the jobs i.e. if $j < j'$, then job j has to be completed before job j' has started. The problem is to find the shortest schedule that satisfies the precedence constraints.

   (a) (3 points) Prove that the naive job scheduling algorithm (first algorithm discussed in class) yields a 2-approximation algorithm for this problem.

   (b) (3 points) For any general value of n and m, give an instance to prove that $2.OPT$ is a tight bound. Assume $t_i = 1 \forall i \in [1, n]$.

Solution: Write your solution here

3. (5 points) **Vertex Cover – Highest Degree Selection** Consider Algorithm 1 as the solution for the vertex cover problem. Prove that the algorithm does NOT give a constant factor approximation for the vertex cover problem.

Algorithm 1: Algorithm for question 3

Input: Graph $G = (V, E)$

Output: Vertex Cover $S \subseteq V$

$S \leftarrow \phi$

while $\exists e \in E(G)$ **do**

  Remove the vertex v with the highest degree

  Add it to the vertex cover S

end

return S

Solution: Write your solution here

4. (5 points) Show that the following algorithm is a 2-approximation for unweighted vertex cover: Perform a DFS and return all internal vertices. Is this true for BFS?

5. (5 points) Unlike the unweighted case, show that the maximal matching may not lead to a constant factor approximation for the weighted vertex cover problem.

6. (5 points) Show that the $\log n$-approximation factor is tight for the greedy set cover algorithm by buildling a worst case example.