Lab11-Approximation

Exercises for Algorithms by Xiaofeng Gao, 2016 Spring Semester

Name:_________ Student ID:_________ Email: _____

1. Explain what the four-tuple $(I, sol, m, goal)$ is for the following problems separately.

   (a) *Load Balancing*: Suppose we have a set of n jobs, which we want to assign to m machines. We are given an array $T[1 \cdots n]$ of non-negative numbers, where $T[j]$ is the running time of job j. We can describe an assignment by an array $A[1 \cdots n]$, where $A[j] = i$ means that job j is assigned to machine i. The makespan of an assignment is the maximum time that any machine is busy:

   $$makespan(A) = \max_i \sum_{A[j]=i} T[j].$$

   The load balancing problem is to compute the assignment with the smallest possible makespan.

   (b) A commander has a scouting force of m soldiers. He once needs to choose a subset S from them to complete as many of a set of n important tasks as possible. Exactly k soldiers are qualified for each task. Each task will be completed if and only if exactly one soldier in S is qualified for that task. How should he choose?

2. Review the *Maximum Independent Set* problem in class.

   (a) In the process of proving the approximation ratio of Greedy Independent Set algorithm, suppose that V^* is the optimal independent set for G, x_i is the vertex chosen at the ith iteration of the algorithm, d_i is the degree of x_i, and k_i is the number of vertices in V^* that are among $d_i + 1$ vertices deleted in the ith iteration. Explain why at each iteration, the number of the deleted edges is at least $\frac{d_i(d_i+1)+k_i(k_i-1)}{2}$.

   (b) Prove that the result value $m_g(x)$ of the solution found by Greedy Independent Set algorithm is at least $\frac{n}{2\delta+1}$, where $\delta = \frac{m}{n}$.

3. Design approximation algorithms with approximation ratio of 2 for the following problems. You should provide both time complexity discussion and approximation factor proof.

   (a) *Linear Arrangement*: Given a directed graph $G = (V, E)$, $|V| = n$, find an ordering $v_1, v_2, \cdots, v_n$ of the vertices that maximizes the number of forward edges. (A directed edge $v_i \to v_j$ is a forward edge if $i < j$.)

   (b) *k-center Clustering*: Given a set $P = \{p_1, p_2, \cdots, p_n\}$ of n points in the plane and an integer k, find a collection of k circles that collectively enclose all the input points, such that the radius of the largest circle is as small as possible. (More formally, you are expected to compute a set $C = \{c_1, c_2, \cdots, c_n\}$ of k center points, such that the following cost function is minimized:

   $$cost(C) := \max_i \min_j |p_i c_j|.$$

   Here, $|p_i c_j|$ denotes the Euclidean distance between input point p_i and center point c_j.)