Spring 2009, CS 598CSC: Approximation Algorithms
Homework 2
Due: 03/04/2009 in class

Instructions and Policy: You are not allowed to consult any material outside of the textbook and class notes in solving these problems. Each student should write up their own solutions independently. You need to indicate the names of the people you discussed a problem with; ideally you should discuss with no more than two other people.

Please write clearly and concisely - clarity and brevity will be rewarded. Refer to known facts as necessary. Your job is to convince us that you know the solution, as quickly as possible.

Problem 1 [20 pts] Multi-processor scheduling: given n jobs $J_1, \ldots, J_n$ with processing times $p_1, p_2, \ldots, p_n$ and m machines $M_1, M_2, \ldots, M_m$.

1. For identical machines show that greedy list scheduling that orders the jobs in non-increasing sizes has an approximation ratio of $3/2$. **Extra credit:** Show a bound of $4/3$.

2. Now consider the problem where the machines are not identical. Machines M_j has a speed s_j. Job J_i with processing time p_i takes p_i/s_j time to complete on machine M_j. Give a constant factor approximation for scheduling in this setting to minimize makespan (the maximum completion time). (Hint: consider jobs in decreasing sizes. Assuming $p_1 \geq p_2 \geq \ldots \geq p_n$ and $s_1 \geq s_2 \geq \ldots s_m$, show that $OPT \geq \max_{i \leq m}(\sum_{j \leq i} p_j / \sum_{j \leq i} s_j)$.)

Problem 2 [10 pts] Prove that the following algorithm (due to Gonzalez) for the k-CENTER problem is a 2-approximation.

<u>k-CENTER</u>
For ($i \leftarrow 1$ to k)
Pick a center furthest from all currently picked centers
Output the set of k centers picked.

Problem 3 [20 pts] Recall that in the INDEPENDENT SET problem, the goal is to find a largest set of vertices in a given graph G such that no two vertices are adjacent. INDEPENDENT SET is both **NP-Hard** and hard to approximate; unless **P** $=$ **NP**, there is no $|V(G)|^{1-\varepsilon}$-approximation algorithm. However, in special classes of graphs, one can obtain better results:

1. Given a graph G with n vertices and m edges, let $\overline{d} = 2m/n$ denote the average degree of a vertex. Give a deterministic $O(\overline{d})$ approximation for the INDEPENDENT SET problem. (Following the convention with approximation ratios less than 1, this would be an $\Omega(1/\overline{d})$ approximation.)

2. For any planar graph $G(V, E)$, we have $|E| \leq 3|V| - 6$; it follows that the average degree of a planar graph is at most 6. Give a simple deterministic algorithm to find an independent set of size at least $|V|/6$ in any planar graph. (Note that the four-color theorem implies that every planar graph has an independent set of size $|V|/4$.)

3. Give an $O(\Delta)$-approximation to the problem of finding an independent set in the *square* of a planar graph, where Δ denotes the maximum degree in the planar graph. (The square of a graph G is the graph on the same vertex set, in which two vertices are adjacent if the distance between them in the original graph G is at most 2.)

Extra Credit: Consider the following *randomized* algorithm: Pick a random permutation of the vertices, and select a vertex for the independent set if it appears earlier in the permutation than all its neighbors. Prove that the *expected* size of the independent set returned by this algorithm is at least $\frac{n}{d+1}$. (*Hint*: What is the expected contribution of a vertex to the independent set?)

Note: There is a PTAS for INDEPENDENT SET in planar graphs; Baker [2] gives a general technique to find approximation schemes for several problems in planar graphs (and in related families of graphs, such as those forbidding fixed minors).

Problem 4 [30pts] In this problem, we solve INDEPENDENT SET in another family of graphs, the intersection graphs of disks in the Euclidean plane: Given a set of disks in the plane, construct a graph by creating a vertex for each disk, and connecting two vertices by an edge if the corresponding disks intersect.

1. Give a PTAS for the INDEPENDENT SET problem in these graphs, assuming all disks have unit radius.

   Hint 1: Consider a grid of lines spaced $\frac{1}{\varepsilon}$ units apart. If no disks of an optimal solution intersect these grid lines, can you find an *exact* algorithm with running time polynomial in n for any fixed ε?

   Hint 2: Consider a grid with *random* offset: Take a grid of lines spaced $\frac{1}{\varepsilon}$ apart, such that the origin is at the intersection of a horizontal and vertical grid line. Pick a shift/offset L uniformly at random from $[0, \frac{1}{\varepsilon})$, and shift the grid vertically and horizontally by an distance L. (Equivalently, consider the grid of spacing $\frac{1}{\varepsilon}$ such that the point (L, L) is at the intersection of two grid lines.) What is the probability that a disk is intersected by a grid line? Can you give a deterministic approximation scheme?

2. We say that a solution X to the INDEPENDENT SET problem is s-optimal if we cannot get a larger independent set by removing at most s vertices from X and adding at most $s + 1$ vertices from $V - X$.

   Consider the following local search algorithm for INDEPENDENT SET in unit disk graphs: Start with an arbitrary solution, and as long as the current solution is not s-optimal, find a larger independent set by removing at most s vertices and adding at most $s + 1$.

Prove that there is a (small) constant s such that the local search algorithm gives a constant-factor approximation. Try to make the approximation ratio as small as you can.

Note: There is a PTAS for the problem, even if the disks are allowed to have different sizes. Do you see how to obtain a PTAS? For more information about geometric approximation, see the survey by Arora [1], and Chapter 11 in the textbook.

Problem 5 [20 pts] Do part 2 of Exercise 13.4 from the textbook (by Vazirani).

References

[1] Sanjeev Arora. Approximation schemes for NP-hard geometric optimization problems: A survey. *Math. Programming*, 97: 43–69, 2003.

[2] Brenda S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the ACM*, 41(1): 153–180, 1994.