CS 880: Approximation Algorithms	Homework 2
Out: 2/22/07	**Due: 3/15/07**

1. **(Euclidean disk-cover.)** Given n points $p_1, \cdots, p_n$ in an area $I \subset \mathbb{R}^2$, our goal in this problem is to cover all of these points with as few disks of diameter $D > 0$ as possible. In this question you will develop a polynomial-time approximation scheme for this problem.

   (a) Suppose at first that I is a square of side-length ℓD for some constant $\ell > 0$. Give a polynomial-time algorithm that solves these instances exactly. (Remember that ℓ is a constant, so it is okay to have a running time exponential in ℓ.)

   (Hint: Show that you only need $O(\ell^2)$ disks to cover I, and that you only need to consider $O(n^3)$ positions for each disk.)

   (b) For a general I, consider the following partitioning strategy: partition I along the x-axis into strips of width ℓD. Let this partition be $\Pi_1 = \{S_1^1, \cdots, S_k^1\}$. We are going to approximate each of these segments separately. But since this partitioning may not work well for some pathological examples, we will consider ℓ different partitions and pick the best over all of them. The second partition Π_2 is obtained by "shifting" each of the S_j^1's by a distance D to the right (S_k^1 is shifted cyclically—the shifted set S_k^2 covers the leftmost strip of width D that was previously covered by S_1^1). Partitions $\Pi_3, \cdots, \Pi_\ell$ are obtained similarly. Note that $\Pi_{\ell+1} = \Pi_1$. In the following, let $\Pi_i = \{S_1^i, \cdots, S_k^i\}$.

   Now suppose that we are given an algorithm A that gives a ρ-approximation for the problem whenever the length of I is bounded by ℓD along one dimension. For each of the Π_i's, we compute a solution as follows: use A to compute feasible disk-covers $C_1^i, \cdots, C_k^i$ for each of the strips $S_1^i, \cdots, S_k^i$. Then, $\mathsf{APX}^i = C_1^i \cup \cdots \cup C_k^i$. Return

   $$\mathsf{APX} = \mathrm{argmin}_{1 \le i \le \ell} |\mathsf{APX}^i|$$

   Show that $|\mathsf{APX}| \le \rho(1 + 1/\ell)|\mathsf{OPT}|$.

   (c) How do (a) and (b) lead to a PTAS for the Euclidean disk-cover problem?

2. **(Vertex cover in planar graphs.)** It is well-known that planar graphs are 4-colorable. In other words, any planar graph can be partitioned into 4 independent sets. Show how you can use an algorithm for 4-coloring a planar graph to find a $3/2$-approximation to vertex cover in the graph.

   Hint: Use the half-integrality of vertex cover.

3. **(K-median.)** The K-median problem is a variant of facility location in which facilities don't have opening costs, but we can open at most K of them. In particular, given an complete graph $G = (V, E)$ with non-negative distances $d : E \to \mathbb{R}_+$ and a number K, find a set $S \subseteq V$ of size at most K that minimizes the routing cost $C_r(S) = \sum_{v \in V} \min_{s \in S} d(s, v)$.

   Note that the distances d do not necessarily form a metric (that is, they may not satisfy the triangle inequality). We will allow our solution to approximate both the number of medians picked ($|S|$) and the routing cost $C_r(S)$.

   (a) Formulate the K-median problem as an ILP. Let the optimal value of its LP relaxation be C^*.

   (b) For the general (non-metric) case, show how to round this LP solution to an integer solution with at most $O((1 + \epsilon) \log |V| \cdot K)$ medians and routing cost $O((1 + \frac{1}{\epsilon}) \cdot C^*))$ for any $\epsilon > 0$.

   (Hint: Use the filtering technique of Lin-Vitter from class.)

   (c) For the metric case (that is, when the distances d obey the triangle inequality), show how to round this LP solution to an integer solution with at most $O((1 + \epsilon) \cdot K)$ medians and routing cost $O((1 + \frac{1}{\epsilon}) \cdot C^*))$ for any $\epsilon > 0$.

4. **(Multiway Cut revisited.)** We can look at MULTIWAY CUT as a coloring problem: color each node in V with one of k colors such that the terminal t_i is colored with color i, so as to minimize the number of bichromatic edges. (Make sure you believe this!)

   Consider an extension of the problem: we are given a "coloring cost" function for each vertex $v \in V$, $C_v : [k] \to \mathbb{R}_{\geq 0}$, such that the cost of coloring v with color i is $C_v(i)$. Now we want to find a coloring $f : V \to [k]$ so as to minimize the total cost

$$\Phi(f) = \sum_{v \in V} C_v(f(v)) + \text{ number of bichromatic edges in } f\ . \tag{1}$$

   Note that if we set $C_{t_j}(i)$ to be 0 if $i = j$ and ∞ otherwise, and for each non-terminal node v, we set $C_v(i) = 0$ for all colors i, then we get back the MULTIWAY CUT problem.

   (a) Our local search algorithm will make moves of the following form: if we are at coloring f, pick a color i and try to find the *best* coloring f' obtained from f by recoloring some of the vertices by the color i. I.e., f' satisfies the property that either $f'(v) = i$ or $f'(v) = f(v)$, and it is the one with the least cost over all such colorings. Call such a best coloring an *i-move*. (In case of ties, choose one arbitrarily.) *Note that we have not shown how to find such an i-move; we will discuss this issue later.*

   Show that if f is a local optimum with respect to these moves, (i.e., none of the k potential i-moves results in the cost strictly decreasing), then $\Phi(f) \leq 2\Phi(\mathsf{OPT})$. As usual, OPT is the optimal coloring.

   (b) Since it may take a long time to reach a local minimum, we can change the algorithm to make a move from f to f' as long as it decreases the cost by at least $\Phi(f) \times (\epsilon/k)$. Show that if we start from a coloring f_0, then the algorithm takes at most

$$O\left(\frac{\log(\frac{\Phi(f_0)}{\Phi(\mathsf{OPT})})}{-\log(1 - \epsilon/k)} \right) \tag{2}$$

   local improvement steps to reach a solution of cost $2(1 + \epsilon)\Phi(\mathsf{OPT})$.

   (c) Note that the number of steps in the above solution is not *strongly polynomial*: if the coloring costs $C_v(\cdot)$ are very large, the number of rounds may be very large (albeit polynomial in the representation of the instance). One way to fix this is to choose the start state f_0 carefully. Can you show a choice of f_0 so that (2) is at most $\text{poly}(n, k, \epsilon)$?

   What about the case when $k \gg n$? Can you change the algorithm so that the number of steps to reach a near-local-optimum is at most $\text{poly}(n, \epsilon)$?

   (d) Suppose you now wanted to make smaller local-search moves of the form: pick a vertex v and a color i, and paint v with color i if the resulting $\Phi(f)$ decreases. (These moves are called the *Glauber dynamics*.) Note that the new algorithm makes much smaller moves than the one above, and hence may take more time to reach a local optimum.

   Are local minima of this new process also 2-approximate? Give a proof or a counterexample.

   Remark: We did not address the question: given a color i and a coloring f, how can we find the best i-move? Despite the fact that there may be $\Omega(2^n)$ possible i-moves to consider, we can indeed find it efficiently using an s-t min-cut computation in a suitably defined graph! (We'll show how to do this in the answers, or you can think about it.)