Approximation Algorithms **Homework 3**

Date: Sept 28, 2005 **Due:** Oct 5, 2005

Note: In all these problems: if we ask you to show an approximation ratio of ρ, you should tell us about the best approximation ratio you did prove, be it better or worse than ρ.

1. **(K-Center Problem Revisited)** Recall the K-center problem: We are given an undirected *complete* graph $G = (V, E)$ with nonnegative *metric* distances d on the edges; i.e., $d_{ij} + d_{jk} \geq d_{ik}$ for all $i, j, k \in V$. Given G and a positive integer K, the K-center problem asks for a subset $C \subseteq V$ of vertices of size $|C| = K$ that minimizes

$$\max_{v \in V} d(v, C). \tag{1}$$

Let D^* be this minimum value when $|C| = K$.

Now, we consider a variant: In this, we insist on a solution such that the distance from every node to its closest center is at most some specified distance D^*, and want minimize the number of centers K' to locate in order to achieve this. (I.e., to defeat the hardness of the problem, instead of relaxing the distance guarantee, we are instead allowing ourselves to place more centers.)

Design an algorithm for this problem that places a set of centers C with $|C| = O(K \log |V|)$, and ensures that (1) is at most D^*.

2. **(Max Leaf Spanning Tree Revisited)** The *locality gap* of an algorithm for this problem is the largest number $\alpha \geq 1$ for which you can show a LOT T such that $\ell(T) \leq \ell(T^*)/\alpha$, where T^* is a max-leaf spanning tree.

The algorithm in class showed that α can never be more than 10. By exhibiting a suitable family of examples, give the best lower bound you can for the locality gap of the algorithm given in class.

3. **(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 s_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: here each vertex $v \in V$ has an associated coloring cost function $C_v : [k] \to \mathbb{R}_{\geq 0}$ such that the cost of coloring v with color i is $C_v(i)$. Now we want find a coloring $f : V \to [k]$ so as to minimize the total cost

$$\Phi(f) = \sum_{v \in V} C_v(f(v)) + \text{ number of bichromatic edges in } f \ . \tag{2}$$

Note that if we set $C_{s_j}(i)$ 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. 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 most $\Phi(f) \times (\epsilon/k)$. Show that if we start from a coloring f_0, then the algorithm takes at most

$$O\left(\frac{\log(\frac{\Phi(f_0)}{\Phi(\mathsf{OPT})})}{-\log(1-\epsilon/k)}\right) \tag{3}$$

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 (3) is at most $\mathrm{poly}(n, k, \epsilon)$?

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 $\mathrm{poly}(n, \epsilon)$?

(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.

All local minima of this new process are 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.)