CS 683 Spring 2001 – Problem Set 2

Due in class Thursday, March 8

(1) It is much easier to describe explicit constructions of bounded-degree expander graphs when we deal with infinite (as opposed to finite) graphs. Specifically, in this question, let $G = (V, E)$ denote a graph whose vertex set is countably infinite. We say that G is *locally finite* if there is a natural number d such that each node of G is incident to at most d edges. We say that G is an *expander* if there is a positive real number α such that for every finite subset of nodes S, we have $|\delta(S)|/|S| \geq \alpha$.

Describe a graph G whose vertex set is countably infinite, such that G is a locally finite expander. Prove that your graph G has these properties.

(2) The *hypercube of order* n is the graph G whose vertex set consists of all n-bit strings, with two strings joined by an edge if they differ in a single position (i.e. if their Hamming distance is 1).

Specify a positive constant α (independent of n) so that for every natural number $n \geq 1$, the hypercube of order n has expansion at least α. Prove that your choice of α has this property.

(3) **(a)** Is the following true for all natural numbers $n \geq 3$?

> *If G is an n-node cycle then the shortest-path metric of G is isometrically embeddable (i.e. embeddable with distortion* 1*) in the L_1 metric.*

Give a proof or a counter-example with justification.

(b) Is the following true for all pairs of natural numbers $m, n \geq 1$?

> *If G is a complete bipartite graph, with m nodes on one side of its bipartition and n nodes on the other, then the shortest-path metric of G is isometrically embeddable (i.e. embeddable with distortion* 1*) in the L_1 metric.*

Give a proof or a counter-example with justification.

(4) Suppose G is a connected planar graph in which every node has even degree; you may assume that G does not contain any node v such that the deletion of v disconnects G.

We saw in class that it is possible to place terminals $s_1, t_1, s_2, t_2, \ldots, s_k, t_k$ on the outer face of G so that, although the cut condition holds, there do not exist edge-disjoint paths joining corresponding pairs of terminals (s_i, t_i). (Although the cut condition holds, the "evenness condition" may fail.)

Supopse we impose a slightly stronger condition, which we'll call the *strict cut condition.* This asserts that for every subset S of the nodes, the number of edges with one end in S and the other in $V \setminus S$ is *strictly greater* than the number of terminal pairs with one end in S and the other in $V \setminus S$. (In other words, $|\delta(S)| > |D(S)|$ for all S.)

Is it the case that if G is as described in the first paragraph, and we place terminals $s_1, t_1, s_2, t_2, \ldots, s_k, t_k$ on the outer face of G so that the strict cut condition holds, then there must exist edge-disjoint paths joining corresponding pairs of terminals (s_i, t_i)? Give a proof or a counter-example with justification.

(5) A major issue in parallel computing is the effective "mapping" of a large computation onto the underlying architecture of a parallel computer. Here's a simplified problem based on some of the issues that arise here.

Suppose we are trying to run a process that is composed of n parallel *threads*. (For reasons that will become clear below, we assume n is a power of 2.) Certain of the threads need to communicate as they run, and we define a graph G whose nodes are the set of threads, and with an (undirected) edge joining each pair that needs to communicate.

We are given a parallel computer whose processors are arranged in a complete binary tree T with n leaves. We must assign each of the n threads to a distinct leaf of T; notationally, we define a bijection f between the set of threads and the set of leaves of T. Now, there are better and worse choices of the bijection f, for the following reason. If i and j are threads such that (i, j) is an edge of G, then a load is placed on each internal node of T that lies on the unique $f(i)$-$f(j)$ path — this internal node will need to be involved in routing communication between i and j throughout the computation. We define the *aggregate load* on an internal node w of T to be the total number of edges (i, j) of G for which w lies on the $f(i)$-$f(j)$ path. We define the *aggregate load* of a bijection f to be the maximum aggregate load it causes on any internal node.

We can now define the *optimal tree mapping* problem as follows: find a bijection f whose aggregate load is as small as possible. Give a polynomial-time approximation algorithm for this problem; for some constant c, it should produce a bijection whose aggregate load is at most $O(\log^c n)$ times the minimum possible.