Math 262B Lecture Note 1

Prof. Fan Chung Graham

Compiled by Lei Wu

1 Basic Settings

$G(V, E)$ is a graph with vertex set V and edge set E. A *walk* is a sequence of vertices $v_1, \ldots, v_t$ such that $v_i \sim v_{i+1}$, (i.e. $(v_i, v_{i+1}) \in E$). A *path* is a walk with distinct vertices, i.e. $v_i \neq v_j$ for $i \neq j$. For $u, v \in V$, the *distance* $d(u, v)$ is the length of the shortest path from u to v. The *diameter* of graph G is $D(G) = max_{u,v \in V} d(u, v)$. For example, in Figure 1, $d(v_1, v_3) = 2, D(G) = 3$.

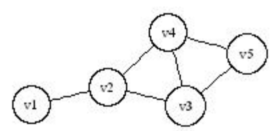

Figure 1: an example of graph G

2 Graph Embedding

Definition 2.1 *Q_n(n-cube) is a graph with $V(Q_n) = \{0, 1\}^n$, i.e. $V(Q_n) = \{a : a$ is a binary string of length $n\}$, and $a \sim b$ if and only if they differ in exactly one coordinate.*

Definition 2.2 *Hamming distance between u and v, $d_{Q_n}(u, v)$, is the number of coordinates in which u and v differ in Q_n.*

Now we want to discuss the embedding f of graph G into the hosting graph H ($f : V(G) \to V(H)$). There are two types of embeddings:

Type-1 Embedding (edge-preserving):
If $u \sim v$ in G, then $f(u) \sim f(v)$ in H.

Remark 2.3 *For Type-1 Embedding, G is embedded in H as a subgraph.*

Type-2 Embedding (distance-preserving):
$d_G(u, v) = d_H(f(u), f(v))$.

Remark 2.4 *Type-2 Embedding is stronger than Type-1 Embedding as distance-preserving is necessarily edge-preserving.*

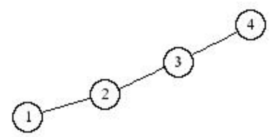

Figure 2: G'

For example, we embed graph G' (see Figure 2) in G in two different ways. We define $f, f' : G' \to G$, respectively, such that $f(1) = v_1, f(2) = v_2, f(3) = v_4, f(4) = v_3$ and $f'(1) = v_1, f'(2) = v_2, f'(3) = v_4, f'(4) = v_5$ then we can see that f is a Type-1 Embedding while f' is a Type-2 Embedding.

3 An addressing problem ('71 Graham-Pollak)

For graph $G = (V, E)$, we want to assign 'addresses', which are binary strings of length k, to vertices of G so that the Hamming distance of the addresses of two vertices is equal to the distance of the two vertices in G. And we call this assignment of addresses a *good assignment*. In another word, we want to find a Type-2 Embedding of G in the n-cube.

Remark 3.1 *If these addresses exist, it will be an excellent scheme for routing. But we know that a lot of graphs(e.g. K_3) are not embeddable in hypercubes. Therefore, we need modifications of the problem and settings.*

Definition 3.2 *Addresses are strings $a = (a_1, a_2, \ldots, a_t)$ of 0's, 1's and *'s with* distance

$$d(a, b) = |\{i : \{a_i, b_i\} = \{0, 1\}\}|.$$

Thus, we can define Q_n^* and the Hamming distance in it similarly as in Definition 2.2.

Under this modification, we can ask the question "Does a good assignment of addresses defined above always exist?"

Theorem 3.3 *For any graph G, there is always a good assignment of addresses to it.*

Given Theorem 3.3, let $q(G) = \min\{k : \text{there is a good assignment } f : V \to \{0,1,*\}^k\}$, we want to find $q(G)$ for any graph G.

Before proving Theorem 3.3, we look at the graph in Figure 1. We can assign addresses to the vertices $v_1, \ldots, v_5$ and write them in matrix form with the ith row corresponding to the address of v_i:

v_1	1	1	1	*	*
v_2	1	0	*	1	*
v_3	*	0	0	0	1
v_4	0	0	1	*	*
v_5	0	0	0	0	0

This is apparently a good assignment of addresses to graph G.

We now show the correspondence between addressing of a graph and quadratic forms. (This is the 3rd way of looking at addressing problem!) To the first column of the addresses above, we associate the product $(x_1 + x_2)(x_4 + x_5)$. Here x_i is in the first, respectively second, factor if the address of v_i has a 1, respectively a 0, in the first column. If we do the same thing for each column and add the terms, we obtain a quadratic form

$$\sum d_{ij}x_ix_j = (x_1 + x_2)(x_4 + x_5) + x_1(x_2 + x_3 + x_4 + x_5) + (x_1 + x_4)(x_3 + x_5) + x_2(x_3 + x_5) + x_3x_5.$$

Here d_{ij} is the distance of the vertices v_i and v_j in G. Thus, in general, an addressing of G corresponds to writing the quadratic form $\sum d_{ij}x_ix_j$ as a sum of k products where k is the length of the addresses. Trivially, we can see that for any graph G there are at most $D(G)n(n-1)/2$ products. Hence, we've proved 3.3.

We can save a factor of $n/2$ if we write

$$x_1(x_2 + \cdots + x_n) + \sum_{i=2}^{D(G)} x_1(x_{i_1} + \cdots + x_{i_{j_i}})$$

where $x_{i_1}, \ldots, x_{i_{j_i}}$ are all those vertices for which $d_{1,i_t} \geq i$. We then repeat for $x_2(x_3 + \cdots + x_n) + \cdots$, and so on up to $d_{n-1,n}$ copies of $x_{n_1}x_n$. In this way, we have at most $(n-1)D(G)$ products in the quadratic form $\sum d_{ij}x_ix_j$. Thus, we proved the following theorem:

Theorem 3.4 $q(G) \le (n-1)D(G)$.

In fact, we can prove that $q(G) \le n-1$('83 Winkler) and show that $n-1$ is also the lower bound for $q(G)$ for some graphs G. We first show this for trees. To prove this, we need the following theorem:

Theorem 3.5 *Let n_+, respectively n_-, be the number of positive, respectively negative, eigenvalues of the distance matrix $M = (d_{ij})$ of the graph G. Then $q(G) \ge \max\{n_+, n_-\}$.*

Theorem 3.6 *If T is a tree on n vertices, then $q(T) = n-1$.*

Proof of Theorem 3.6: We first show $q(T) \le n-1$ by induction on n. For $n = 2$, we have a trivial addressing with length 1, namely $v_1 \to 0$ and $v_2 \to 1$. Now for T with n vertices, suppose v is a leaf and $(v, w) \in E(T)$. Suppose there is a good assignment f for $T' = T \setminus \{v\}$, we define

$$g(u) = \begin{cases} (f(u), 0) & \text{if } u \ne v \\ (f(u), 1) & \text{if } u = v \end{cases}$$

Then, g is a good assignment for T for

$$d(g(u), g(u')) = d((f(u), 0), (f(u'), 0)) = d(f(u), f(u')) = d_T(u, u')$$

for any $u, u' \in V(T)$ and obviously

$$d(g(v), g(w)) = d((f(v), 1), (f(w), 0)) = d(f(v), f(w)) + 1 = d_T(v, w),$$

and g is of length $n-2+1 = n-1$ by the induction hypothesis.

To show $q(T) \ge n-1$, we first calculate the determinant of distance matrix $M = (d_{ij})$ of T. We number the vertices $v_1, \dots, v_n$ in such a way that v_n is a leaf adjacent to v_{n-1}. In the distance matrix, we subtract row $n-1$ from row n, and subtract column $n-1$ from n. Then all the entries in the new last row and column are 1 except for the diagonal element which is equal to -2. Now renumber the vertices $v_1, \dots, v_{n-1}$ in such a way that the new vertex v_{n-1} is a leaf of $T \setminus \{v_n\}$ adjacent to v_{n-2}. Repeat the procedure for the rows and columns with numbers $n-1$ and $n-2$. After $n-1$ steps, we have the determinant:

$$\begin{vmatrix} 0 & 1 & 1 & \cdots & 1 \\ 1 & -2 & 0 & \cdots & 0 \\ 1 & 0 & -2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & -2 \end{vmatrix}.$$

Thus we find that the determinant of the distance matrix of a tree on n vertices is $(-1)^{n-1}(n-1)2^{n-2}$, i.e. it depends only on n. From this fact and Theorem 3.5, we can prove that $q(G) \geq n-1$ which is left as an exercise.□

For complete graph K_n, the distance between any two distinct vertices is 1. Therefore, we can take the identity matrix of size $n-1$, replace the zeros above the diagonal by *'s and add a row of 0's:

$$\begin{pmatrix} 1 & * & \cdots & * & * \\ 0 & 1 & \cdots & * & * \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix}.$$

Any two rows now have distance 1 and hence $q(K_n) \leq n-1$.