7
Labelings of Graphs

F. R. K. CHUNG

1. Introduction

In this chapter we discuss several interrelated graph labeling problems. These labeling problems have been studied in the past in various different formulations. Typically, the problems can be described as follows: *for a given graph, find the optimal way of labeling the vertices with distinct integers, k-tuples of integers, or group elements subject to certain objectives.* These problems often come up in connection with applications in network addressing, circuit layout or code design.

For example, suppose that we consider labeling the vertices of a graph G by distinct integers. (This can be viewed as arranging the vertices into a line or a linear array.) If we want to find the labeling which minimizes the maximum 'stretch' $b(G)$ over all the edges, we have the so-called *bandwidth problem*. If we want to find the labeling which minimizes the total 'length' sum $s(G)$ of the edges, we have the *minimum-sum problem*. If we want to find the labeling which minimizes the maximum 'overlap' $c(G)$, we have the *cutwidth problem*. (In Fig. 1 we illustrate a tree together with several optimal labelings.) These problems will be rigorously defined in Section 2.

These problems can be considered in the following general framework: label the vertices of a graph G by distinct vertices of a 'host graph' H, and

GRAPH THEORY, 3
ISBN 0–12–086203–4

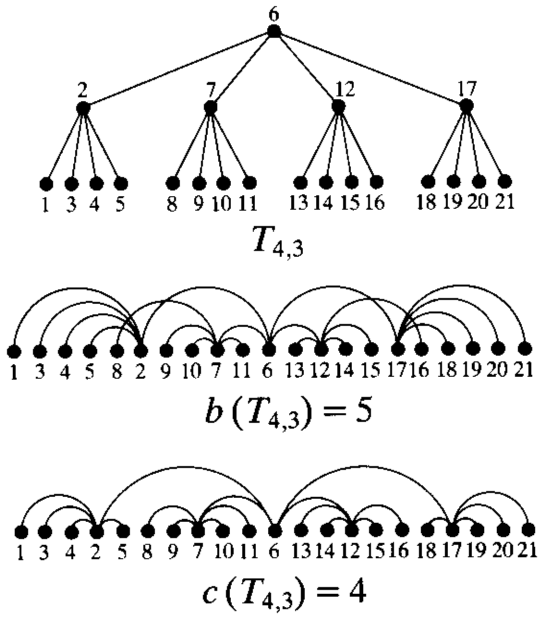

Fig. 1

embed the edges of G into paths of H, subject to any one of the following conditions:

(*i*) the maximum distance in H between adjacent vertices in G is minimized;

(*ii*) the total sum of distances in H between adjacent vertices in G is minimized;

(*iii*) the 'frequency' of edges in H appearing in paths joining adjacent vertices in G is minimized.

For the case of labeling vertices by distinct integers, the host graph is taken to be a path. In the case of labeling vertices by pairs of integers, the host graph is just the grid graph in the plane. When vertices are labeled by binary k-tuples, the labeling provides an embedding into a k-cube as the host graph. Of course, most of the known results in the literature take a path as the host graph.

We shall survey results on various optimal graph labeling problems, and study the properties of these optimal labelings, the relationship with other graph-theoretic parameters, and the algorithmic complexity in determining these optimal labelings for graphs or some special classes of graphs.

2. Definitions and Examples

Let G be a graph with vertex-set $V(G)$ and edge-set $E(G)$. A **labeling** π of G in the host graph H is a one-to-one mapping π from $V(G)$ to $V(H)$. Such a labeling can also be viewed as a placement of the vertices of G into vertices of the fixed host graph H. The following list gives some labelings of interest:

(*i*) The **bandwidth** $b_\pi(G)$ of a labeling π is defined by

$$b_\pi(G) = \max\{d_H(\pi(v), \pi(w)): vw \in E(G)\},$$

where $d_H(v', w')$ denotes the distance between v' and w' in H.

The **bandwidth** $b(G)$ of G is the minimum of $b_\pi(G)$ over all labelings π; that is,

$$b(G) = \min\{b_\pi(G): \pi \text{ is a labeling of } G \text{ in } H\}.$$

(*ii*) A graph G' is said to be a **refinement** of G if G' is obtained from G by a finite number of edge-subdivisions. (For example, in Fig. 2 T' is a refinement of T.) The **topological bandwidth** $b^*(G)$ of G is defined by

$$b^*(G) = \min\{b(G'): G' \text{ is a refinement of } G\}.$$

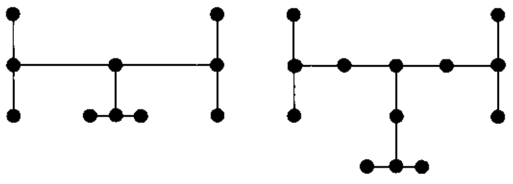

Fig. 2

(*iii*) The **min-sum** $s(G)$ of a labeling π is

$$s(G) = \min\left\{\sum_{vw \in E(G)} d_H(\pi(v), \pi(w)): \pi \text{ is a labeling of } G \text{ in } H\right\}.$$

(*iv*) Suppose we assign a set P_π of paths p in H joining $\pi(v)$ to $\pi(w)$ for all $vw \in E(G)$—that is, $P_\pi = \{p(\pi(v), \pi(w)): vw \in E(G)\}$. We define the **cutwidth** $c_\pi(G)$ of a labeling π to be the maximum number of appearances of an edge e of H in the paths in P_π. In other words,

$$c_\pi(G) = \max_{e \in E(H)} |\{vw \in E(G): e \in E(p(\pi(v), \pi(w))), p(\pi(v), \pi(w)) \in P_\pi\}|.$$

The **cutwidth** $c(G)$ of G is then defined to be

$$c(G) = \min\{c_\pi(G): \pi \text{ is a labeling of } G \text{ in } H\}.$$

When the host graph H is a path with vertices $\{1, 2, \dots\}$, the labeling of G is a mapping from $V(G)$ into the set of positive integers. Such a labeling is often called a **numbering** of G. The distance between two numbers is just the absolute value of the difference. The bandwidth of a numbering π can therefore be written as

$$b_\pi(G) = \max\{|\pi(v) - \pi(w)|: vw \in E(G)\}.$$

Also the cutwidth of a numbering π can be written as

$$c_\pi(G) = \max_i |\{vw \in E(G): \pi(v) \leqslant i < \pi(w)\}|.$$

We conclude this section with a table showing various optimal numberings for paths P_p, circuits C_p, stars $K_{1,p}$, and complete graphs K_p (see [13]):

	b	b^*	s	c
P_p	1	1	$p-1$	1
C_p	2	2	$2(p-1)$	2
$K_{1,p}$	$\lfloor \frac{1}{2}p \rfloor$	$\lfloor \frac{1}{2}p \rfloor$	$\lfloor \frac{1}{4}p^2 \rfloor$	$\lfloor \frac{1}{2}p \rfloor$
K_p	$p-1$	$p-1$	$p(p^2-1)$	$\lfloor \frac{1}{4}p^2 \rfloor$

3. The Bandwidth Numbering

Among all graph labeling problems, bandwidth numberings of graphs have attracted the most attention in the literature. The bandwidth problem originated in the 1950s in the form of finding a matrix equivalent to a given matrix so that all the non-zero entries lie within a narrow band about the main diagonal. Harper [29] investigated the bandwidth numberings for n-cubes; these are related to the design of error-correcting codes subject to minimizing the maximum absolute error. Since then there have been a large number of papers on this subject, references for most of which can be found in a recent survey [5] on bandwidth numberings.

One of the interesting problems on bandwidth is to characterize graphs with large or small bandwidth. What does it take to force a large bandwidth in a graph? There are two known factors which increase the bandwidth of a graph — density (roughly, the ratio of the total number of vertices to the maximum distance between pairs of vertices), and subtrees which are refinements of complete binary trees. As we shall see, either of these two conditions is sufficient but not necessary for forcing up the bandwidth of a graph. We list some known results as well as some obvious (but unpublished) facts on lower bounds for bandwidth numberings.

The following two density lower bounds can be derived in a straightforward way. The first appeared in the early paper of Harper [29], whereas the second one was a folklore theorem discovered by many people independently (see [11]):

Theorem 3.1. *Let ∂S denote the set of vertices in S adjacent to some vertices in $V(G) - S$. Then*

$$b(G) \geqslant \max_k \min_{|S| = k} |\partial S|. \quad \|$$

Theorem 3.2. *Let $D(G)$ denote the diameter of G. Then*

$$b(G) \geqslant (|V(G)| - 1)/D(G). \quad \|$$

If G is a graph with p vertices and diameter D, we define the **density** $ds(G)$ to be $(p - 1)/D$. The **local density** of G is defined to be the maximum density of all subgraphs of G, and can sometimes provide better lower bounds for the bandwidth of G.

Theorem 3.3. $b(G) \geqslant \max\{ds(G')$: *$G'$ is a subgraph of G*$\}$. $\|$

From Theorem 3.3 we can easily deduce the following result, by taking G' to be a star graph (see [11]).

Theorem 3.4. *If Δ denotes the maximum degree in G, then*

$$b(G) \geqslant \tfrac{1}{2}\Delta. \quad \|$$

Sysło and Zaks [43] proved that the local density lower bound in Theorem 3.3 determines the bandwidth of a special class of trees called *caterpillars*. A **caterpillar** is a tree in which the removal of all end-vertices leaves a path P called the **spine**. The end-edges are called **leaves**.

Theorem 3.5. *Let G be a caterpillar whose spine consists of the vertices $u_1, \ldots, u_m$. If the vertex-degrees $\rho(u_1), \ldots, \rho(u_m)$ form a monotone sequence, then*

$$b(G) = \max_i \{\lceil (|V(G_i)| - 1)/D(G_i) \rceil\},$$

where G_i is the subcaterpillar formed by $u_1, \ldots, u_i$ and all vertices adjacent to these u_j, for $i = 1, \ldots, m$. $\|$

In fact, the bandwidths of all caterpillars depend only on the local density:

Theorem 3.6. *If G is a caterpillar, then*

$$b(G) = \max_{G'} \{\lceil (|V(G')| - 1)/D(G') \rceil\},$$

where G' consists of a subpath P' of the spine and all vertices adjacent to vertices in P'.

Proof. Suppose that π is a numbering of G with $b_\pi(G) = b(G) = b$. We can normalize π so that

(*i*) all vertices u_i, $1 \leqslant i \leqslant m$, on the spine are mapped to $bi + 1$;

(*ii*) the leaves adjacent to u_i occupy all the numbers from $\pi(u_i) - r_i$ to $\pi(u_i) + s_i$, except $\pi(u_i)$, with u_i on the spine;

(*iii*) the number of r_i with maximum value is minimized—the maximum value is $b - 1$, except for r_1 which has value b;

(*iv*) some r_i has maximum value.

Now we start from u_j with r_j of maximum value and search for the first k ($>j$) such that $s_k + r_{k+1} < b - 1$ or $s_k = b - 1$. If no such k exists, we take $k = m$. If k satisfies $s_k + r_{k+1} < b - 1$ or $k = m$ and $s_m < b$, then we can modify π by increasing r_t and s_t by 1 for $j \leqslant t \leqslant k$ and thus decrease the number of maximum r_i, contradicting assumption (*iii*). We may assume that $s_k = b - 1$ (or $s_m = b$) and $s_t + r_{t+1} = b - 1$ for $j \leq t \leq k$. Thus, by taking G' with the spine P' consisting of $u_j, \ldots, u_k$, we have

$$(|V(G')| - 1)/D(G') \geqslant \lceil (b(k - j + 1) + 1)/(k - j + 2) \rceil = b,$$

so $b(G) \leqslant \lceil (|V(G')| - 1)/D(G') \rceil$ for some G'. Combining this with Theorem 3.3, we deduce Theorem 3.6. ||

Let $T_{2,k}$ denote the k-level complete binary tree in which the ith level consists of 2^{i-1} vertices and each vertex in level $i < k$ has two 'sons' at level $i + 1$. Then the bandwidth of $T_{2,k}$ depends on k, as follows:

Theorem 3.7. $b(T_{2,k}) = \lceil (2^{k-1} - 1)/(k - 1) \rceil$.

Proof. By Theorem 3.2 we have $b(T_{2,k}) \geqslant x$, where $x = \lceil (2^{k-1} - 1)/(k - 1) \rceil$. It suffices to find a numbering π with $b_\pi(T_{2,k}) = x$. We define π by specifying in increasing order the vertices chosen from the left half (descendants of the left son) of $T_{2,k}$—the first one chosen is labeled 1, and so on; the right half is then chosen in the reverse order of the left half. We first choose x vertices in the left-most part of the bottom (kth) level. Then we choose the y_1 fathers of these x vertices and then choose the $x - y_1$ left-most (unchosen) vertices of the bottom level. We continue to choose x vertices at each step, in such a way that the y_i fathers of previously chosen ones are chosen first, and then the bottom level left-most (unchosen) ones. It is not hard to check that, for $i > 2$, $y_i < \frac{1}{2}x + i$. We have $y_i < x$ if $i \leqslant \frac{1}{2}x$, which is true for $k > 5$. (For $k \leqslant 5$, the bandwidth can be verified directly.) So we can proceed until at most x vertices are left. The remaining vertices are chosen at the final step. ||

The complete k-level t-ary tree $T_{t,k}$ is also determined by its density; the proof is quite similar to that of Theorem 3.7 and will not be included here.

Theorem 3.8. $b(T_{t,k}) = \left\lceil \dfrac{t(t^{k-1} - 1)}{2(k - 1)(t - 1)} \right\rceil$. ||

For trees with bounded degree we have an upper bound which is within a constant factor of the worst-case density lower bounds:

Theorem 3.9. *Suppose that T is a tree on p vertices with bounded degree ρ. Then*

$$b(T) \leqslant 5 \frac{p}{\log_\rho p}.$$

Proof. This upper bound is established by a numbering scheme which uses a result in **[6]**—namely, for a given tree T and any positive value k we can find a vertex v such that there is a subforest F, formed by removing v from T, with $k \leqslant |V(F)| < 2k$. Now we first find a separator set S consisting of $\lceil \log_\rho p - \log_\rho \log_\rho p + 1 \rceil = y$ vertices whose removal separates T into at most y forests $F_1, F_2, \ldots, F_y$, such that $p/y \leqslant |V(F_i)| < 2p/y$, for all i. Furthermore, we may assume that the forests F_i are arranged in increasing order of the number of edges from F_i to T. Now we partition the integers into $3y$ (increasing) blocks B_j, each consisting of $\lceil p/y \rceil$ consecutive numbers. The labeling can be described recursively as follows:

—the numbers in B_1 and B_2 are used to label F_1;

—B_3 is used for labeling the vertices adjacent to vertices in F_1;

—B_4 and B_5 are used to label the (so-far unlabeled) vertices of F_2;

—B_6 is used for labeling the unlabeled vertices adjacent to vertices of F_2 and unlabeled vertices adjacent to vertices labeled by numbers in B_3.

In general, B_{3i+1} and B_{3i+2} are used for labeling the so-far unlabeled vertices of F_{i+1}, and B_{3i+3} is used for labeling the (unlabeled) neighbors of F_{i+1} and the (unlabeled) neighbors of vertices of B_{3i}. Since $\bigcup_{i \leqslant j} F_i$ has at most j vertices in the separating set S, the number of neighbors that B_{3j+3} used for labeling is at most

$$\rho^{j-1} + \rho^{j-2} + \ldots + 1 = \frac{\rho^j - 1}{\rho - 1} \leqslant \rho^y \leqslant \frac{p}{y}.$$

This labeling obviously has bandwidth $4\dfrac{p}{y} \leqslant 5\dfrac{p}{\log_\rho p}$. ||

The density or local density is not an upper bound for graphs in general. There are graphs with large bandwidth but small density. The following result comes from [**12**]:

Theorem 3.10. *For each integer k, there is a tree with local density* 1 *or* 2 *and bandwidth at least k.*

Proof. We consider a refinement T of the complete binary tree $T_{2,2k}$ with $2k$ levels. An edge joining a vertex of the ith level and a vertex of the $(i + 1)$th level is replaced by a path of length 3^{2k-i}. It can be easily checked that T has local density at most 2. The bandwidth of T is not less than $b^*(T_{2,k})$, which is k (see [**12**] and also Section 6). ||

Theorem 3.11. *Any graph which contains a refinement of $T_{2,k}$ must have bandwidth at least $\frac{1}{2}k$.* ||

Of course, there are graphs with large bandwidth which do not contain large complete binary trees. This leads to the following interesting problem:

Problem 3.1. Suppose that the local density of G is at most c_1, and that G does not contain any refinement of the complete binary tree with c_2 levels. Is the bandwidth of G bounded above by a constant depending only on c_1 and c_2?

Note added in proof. This problem was recently answered in the negative by Chung and Seymour.

Chvátalová and Opatrný [**14**] proved a somewhat weaker result:

Theorem 3.12. *Suppose that T is an infinite countable tree such that*

(*i*) *the maximum degree satisfies $\Delta \leqslant c_1$;*

(*ii*) *the number of edge-disjoint semi-infinite paths in T is at most c_2;*

(*iii*) *T does not contain (as a subgraph) a refinement of a complete binary tree of c_3 levels. Then T has a refinement T' with finite bandwidth depending only on c_1, c_2 and c_3.* ||

Another useful observation for dealing with bandwidth involves the graph P_n^k, the kth power of the path P_n, in which two vertices v, w are adjacent if and only if $0 < |v - w| \leqslant k$. It follows immediately that $b(G) \leqslant k$ if and only if $G \subseteq P_n^k$. Many relationships between the bandwidth and other graph invariants were derived using this observation (see [**11**]); for example:

Theorem 3.13. *Suppose that G has p vertices, q edges, connectivity κ, independence number β_0, and degree-sequence $\rho_1 \leqslant \rho_2 \leqslant \ldots \leqslant \rho_p$. Then*:

$$b(G) \geqslant p - \tfrac{1}{2}(1 + ((2p - 1)^2 - 8q)^{\frac{1}{2}});$$
$$b(G) \geqslant \kappa;$$

$$b(G) \geqslant \frac{p}{\beta_0} - 1;$$

$$b(G) \geqslant \max_j \max\{\rho_j - \lfloor\tfrac{1}{2}(j-1)\rfloor, \tfrac{1}{2}\rho_j\}.$$

Proof. Observe that $|V(G_1)| = |V(G_2)|$, and that $G_1 \subseteq G_2$ implies $|E(G_1)| \leqslant |E(G_2)|$, $\kappa(G_1) \leqslant \kappa(G_2)$, $\beta_0(G_1) \geqslant \beta_0(G_2)$ and $\rho_j(G_1) \leqslant \rho_j(G_2)$, for each j. By noting that for $k = b(G)$ we have $G \subseteq P_p^k$ and $|E(P_p^k)| = \frac{1}{2}k(2p - k - 1)$, we get $\kappa(P_p^k) = k$, $\beta_0(P_p^k) = \lceil (p-1)/k \rceil$, $\rho_j(P_p^k) = \min\{p - 1, k + \lfloor\frac{1}{2}(j-1)\rfloor, 2k\}$. ||

We remark that parts of Theorem 3.13 were also proved by A. K. Dewdney [**5**], [**13**]. For completeness, we include here the bandwidths of a few special graphs (see [**11**], [**18**] and [**29**]):

Theorem 3.14. (*i*) $b(K_{r,s}) = \frac{1}{2}(r-1) + s$, *for* $r \geqslant s$;

(*ii*) $b(K_{r_1, r_2, \ldots, r_k}) = p - \lceil\frac{1}{2}(r_1 + 1)\rceil$;

(*iii*) $b(Q_n) = \sum_{k+1}^{n-1} \binom{k}{\lfloor\frac{1}{2}k\rfloor}$, *where* Q_n *is the n-cube.* ||

4. Bandwidth Algorithms

Before we discuss the algorithmic aspects of the bandwidth problem, we shall give a very brief introduction. In general, algorithms are step-by-step procedures for producing solutions for problems. A **polynomial-time algorithm** is an algorithm which always generates a solution in time $p(n)$, for some polynomial function p, where n denotes the input length. A problem is considered to be **intractable** if it is so hard that no polynomial algorithm can possibly solve it. Many problems that are not known to be either provably intractable or provably polynomial turn out to be so-called ***NP*-complete** (non-deterministic polynomial-time complete), which were first introduced in the early 1970s (see [**20**]). A problem in the *NP*-complete class has the property that if a polynomial-time algorithm is ever found for it, then all the problems in this class must also have polynomial-time algorithms. This *NP*-complete class of problems includes many 'classical' problems, such as the traveling salesman problem, the Hamiltonian circuit problem, integer linear programming, and many others. The question of whether or not the *NP*-complete problems are intractable is now considered to be one of the foremost open questions in theoretical computer science.

Papadimitriou [**36**] first proved that the bandwidth problem is *NP*-complete, by showing that it is polynomial transformable from the 3-partition problem. The **3-partition problem** is one of the basic *NP*-complete problems, and can be described as follows:

Suppose that A is a set of $3m$ elements with total weight $\sum_{a \in A} \omega(a) = mB$. Can A be partitioned into m disjoint 3-sets $A_i (i = 1, \ldots, m)$ such that $\sum_{a \in A_i} \omega(a) = B$ for each i?

Theorem 4.1. *The bandwidth problem is NP-complete.* ||

Garey *et al.* [**19**] proved that the bandwidth problem remains *NP*-complete for trees with maximum degree 3. The bandwidth of a graph G is 1 if and only if each connected component of G is an isolated vertex or a path. In [**19**] there is a linear algorithm for testing if $b(G) = 1$ or 2. Saxe [**39**] has shown that, for fixed k, the problem of determining $b(G) \leq k$ can be solved in polynomial time.

There are several approximation algorithms for the bandwidth problem, such as the Cuthill–McKee algorithm [**16**] and the Gibbs–Poole–Stockmeyer algorithm [**23**]. One way to judge the approximation algorithms is to investigate the worst-case performance bounds (that is, the bounds for the ratio of the bandwidth generated by the algorithm and the 'optimum' bandwidth of the graph), or the average performance bounds. However, for these approximation algorithms neither the worst-case performance nor the average performance bound has been extensively analyzed.

5. Bandwidth Labeling in Higher Dimensions

The (two-dimensional) **grid graph** is the graph whose vertex-set consists of pairs of integers, and whose edge-set consists of $\{(i, j), (i + 1, j)\}$ and $\{(i, j), (i, j + 1)\}$, for all integers i, j. Graph labeling problems whose grid graph is the host graph often arise in the formulation of circuit layout models involved in VLSI design or optimization (see [**2**], [**3**]). In particular, the bandwidth of the layout is directly related to the performance of the circuit if the length of the edge is proportional to the propagation delay through the wire, which must be smaller than the period of the system clock. In models with higher-dimensional grid graphs as the host graph, bandwidth problems correspond to layout problems in multi-layer circuits.

Before we proceed to discuss bandwidth results in grid graphs, we remark that the distance of two vertices (a, b) and (c, d) in the grid graph is defined to be $|a - c| + |b - d|$. The *k-dimensional grid graph* G_i has vertex-set consisting of all k-tuples of integers, and edge-set consisting of $\{(a_1, \ldots, a_i, \ldots, a_k), (a_1, \ldots, a_i + 1, \ldots, a_k)\}$, for all a_i and i.

Analogous to the bandwidth numbering problem, we have the following density lower bound:

Theorem 5.1. *If G is a graph with p vertices and diameter D, then the bandwidth of G in the 2-dimensional grid graph is bounded below by $(p^{\frac{1}{2}} - 1)/D$.*

Proof. Let f denote the bandwidth labeling from $V(G)$ to the grid graph G_2. Consider the lines

$$\begin{aligned} L_1&: x + y = \min\{a: (a, b) = f(v), \quad v \in V(G)\} = p_1; \\ L_2&: x + y = \max\{a: (a, b) = f(v), \quad v \in V(G)\} = p_2; \\ L_3&: x - y = \min\{b: (a, b) = f(v), \quad v \in V(G)\} = q_1; \\ L_4&: x - y = \max\{b: (a, b) = f(v), \quad v \in V(G)\} = q_2. \end{aligned}$$

Clearly all vertices of G are embedded (in G_2) in the rectangle bounded by these four lines. This rectangle contains at most $(|p_1 - p_2| + 1)(|q_1 - q_2| + 1)$ vertices. Without loss of generality, we may assume that $|p_1 - p_2| \geqslant |q_1 - q_2|$. It is not hard to check that there are two vertices of G embedded in G_2 with distance $|p_1 - p_2|$ in G. Therefore $bD \geqslant |p_1 - p_2|$, where b denotes the bandwidth. Since

$$(|p_1 - p_2| + 1)^2 \geqslant (|p_1 - p_2| + 1)(|q_1 - q_2| + 1) \geqslant p,$$

we have $b \geqslant (p^{\frac{1}{2}} - 1)/D$. ||

We also have the following theorem:

Theorem 5.2. *Let $b_2(G)$ denote the bandwidth of G in the 2-dimensional grid graph. Then*

$$b_2(G) \geqslant \max\left\{ \left\lceil \frac{|V(G')|^{\frac{1}{2}} - 1}{D(G')} \right\rceil : G' \text{ is a subgraph of } G \right\}. \ ||$$

The higher-dimensional cases can be proved in a similar way.

Theorem 5.3. *If G is a graph with p vertices and diameter D, then the bandwidth of G in the k-dimensional grid graph is bounded below by $(p^{1/k} - 1)/D$.* ||

Theorem 5.4. *Let $b_k(G)$ denote the bandwidth of G in the k-dimensional grid graph. Then*

$$b_k(G) \geqslant \max\left\{ \left\lceil \frac{|V(G')|^{1/k} - 1}{D(G')} \right\rceil : G' \textit{ is a subgraph of } G \right\}. \ ||$$

In the case of embedding complete binary trees in the 2-dimensional grid graph, the density lower bound is essentially $n^{\frac{1}{2}}/\log n$, where n is the input length. Paterson *et al.* [37] proved that the bandwidths b_2 of such trees are within a constant factor of the density lower bounds:

Theorem 5.5. *If* $n = 2^{k+1} - 1$, *then* $b_2(T_{2,k}) = \Omega(n^{\frac{1}{2}}/\log n)$; *in other words,* $b_2(T_{2,k})$ *is bounded above and below by expressions of the form* $cn^{\frac{1}{2}}/\log n$. ||

Problem 5.1. Evaluate $b_k(T_{i,j})$, for general k.

Bhatt and Leighton [**3**] used a general technique to obtain upper bounds for the bandwidth b_2 for binary trees (not necessarily complete) and planar graphs:

Theorem 5.6. *If T is a binary tree* (*that is, a tree with maximum degree* 3), *then*

$$b_2(T) \leqslant \Omega(n^{\frac{1}{2}}/\log n). \ \|$$

Theorem 5.7. *If G is a planar graph, then*

$$b_2(G) \leqslant O\left(\frac{n^{\frac{1}{2}}\log n}{\log\log n}\right). \ \|$$

Problem 5.2. Find good bounds for $b_k(G)$, when G is planar or of bounded genus.

The algorithmic problem of finding optimal labelings in grid graphs turns out to be considerably harder than the case of embedding into a line, as the following result by Bhatt and Cosmadakis [**2**] shows:

Theorem 5.8. *Given a binary tree T, the problem of deciding whether* $b_2(T) = 1$ *is NP-complete.* ||

Unless otherwise specified, in the remainder of the chapter we discuss only labelings with a path as the host graph.

6. Topological Bandwidth

We recall that the topological bandwidth $b^*(G)$ of a graph G is the minimum bandwidth $b^*(G')$ over all refinements G' of G. It follows from the definition that $b^*(G') \geqslant b^*(G)$ for a refinement G' of G. Strict inequality can occur, as shown by Fig. 2.

The topological bandwidth problem can be related to optimization problems arising in some models of VLSI design in which vertices of degree 2 (interpreted as 'drivers' or 'repeaters') are inserted to help minimize the length of the edges. There is also a sparse-matrix version of the topological bandwidth problem [**34**]. Let **A** be a matrix arising from a linear system **Ax** = **b**. The bandwidth problem is just the problem of finding a permutation matrix **P** such that $\mathbf{PAP}^{\mathrm{T}}$ has all of its non-zero entries close to the main diagonal. The topological bandwidth problem can then be viewed as the problem of further narrowing the 'distance' to the main diagonal by repeatedly using the following operations: replace a term $a_{ij}x_j$ by a new variable y and add a new equation $a_{ij}x_j = y$.

There are fewer results on the topological bandwidth than on the bandwidth of graphs. The following results are taken from [**12**], [**35**] and [**42**].

Theorem 6.1. (*i*) $b^*(G) \geqslant \frac{1}{2}\Delta$;

(*ii*) $b^*(G) \geqslant \min\limits_{v \in V(G)} \deg(v)$. ||

Theorem 6.2. (*i*) *For the k-level complete binary tree* $T_{2,k}$, $b^*(T_{2,k}) = \lceil \frac{1}{2}k \rceil$;

(*ii*) $b^*(K_{2,s}) \geqslant \lfloor \frac{1}{2}(s + 3) \rfloor$. ||

Theorem 6.3. (*i*) *There is an* $O(n \log n)$ *algorithm for computing the topological bandwidth of binary trees*;

(*ii*) *the problem of determining the topological bandwidth of a graph is NP-complete*. ||

We conclude this section with the following problem:

Problem 6.1. Determine the computational complexity of the topological bandwidth for trees.

Partial results in this direction can be found in Dewdney [**17**].

7. The Minimum Sum Problem

Instead of minimizing the maximum 'stretch' over all the edges, the minimum sum problem is to search for a labeling which minimizes the total sum of the stretches of the edges. In the case of labeling vertices by integers (or when the host graph is a path), the minimum sum problem is just that of minimizing $s(G) = \sum\limits_{vw \in E(G)} |\pi(v) - \pi(w)|$ over all numberings π. The minimum sum problem was first investigated by Harper [**27**], who determined the optimal labelings for the class of n-cubes which are associated with designing error-correcting codes with minimum average absolute errors:

Theorem 7.1. *For the n-cube* Q_n, $s(Q_n) = 2^{n-1}(2^n - 1)$. ||

Seidvasser [**40**] studied the maximum min-sum $s(T)$ over all trees with p vertices and maximum degree ρ. His results were further improved by Iordanskiĭ [**30**]:

Theorem 7.2. *For any tree T on p vertices and maximum degree ρ, we have*

$$s(T) \leqslant c\,\frac{\rho p \log p}{\log \rho}, \text{ for some constant } c,$$

and there exist trees for which $s(T)$ lies within a constant factor of this upper bound, for all p and ρ. ||

Chung [6] answered a question of Cahit [4] in determining $s(T_{2,k})$ for the complete k-level binary tree $T_{2,k}$:

Theorem 7.3. $s(T_{2,k}) = 2^k(\frac{1}{3}k + \frac{5}{18}) + (-1)^k\frac{2}{9} - 2$ *for* $k \geqslant 2$. ||

For general t $(t > 2)$, it is much more difficult to derive an explicit expression for $T_{t,k}$. A recurrence relation for $s(T_{3,k})$ can be described as follows (see [6]):

Theorem 7.4. *Let $k \geqslant 3$, and let $f(k)$ be the integer l which satisfies $(l-1)3^{l-2} + l + 1 \leqslant k < l3^{l-1} + l$. Then*

$$s(T_{3,k}) = 3^{k-2}(2k - \tfrac{1}{2}) - \tfrac{1}{2} + k - f(k) + s(T_{3,k-1}). \ ||$$

As to the algorithmic problems, Garey *et al.* [22] have proved the *NP*-completeness of the min-sum problem:

Theorem 7.5. *The problem of determining the minimum sum labeling for a general graph is NP-complete.* ||

In the special case of a tree, Goldberg and Klipker [24] have an $O(p^3)$ algorithm for determining a min-sum labeling for a tree on p vertices. Shiloach [41] improved the algorithm to one which has $O(p^{2.2})$ in running time. This was further improved by Chung in [8]:

Theorem 7.6. *If $\lambda = \log 3/\log 2 \approx 1.6$, there is an $O(p^\lambda)$ algorithm for finding the min-sum labeling for trees on p vertices.* ||

8. The Cutwidth of Graphs

The cutwidth problem deals with the number of edges passing over a vertex, when all vertices are arranged in a line; note that $c(G) = \min\limits_{\pi} \max\limits_{i} |\{vw \in E(G)\colon \pi(v) \leqslant i < \pi(w)\}|$. The cutwidth often corresponds to the area of the layout in VLSI design.

Lengauer [32] evaluated the cutwidths for the complete k-level t-ary trees $T_{t,k}$:

Theorem 8.1. $c(T_{t,k}) = \lceil\frac{1}{2}(k-1)(t-1)\rceil + 1$, *for* $k \geqslant 3$. ||

The values of cutwidths for trees are at least as large as their topological bandwidths (see [9], [34]):

Theorem 8.2. $b^*(G) \leqslant c(G)$. ||

For general graphs, $b^*(G)$ and $c(G)$ can be quite different; for example, $b^*(K_p) = p - 1$ and $c(K_p) = \lfloor \frac{1}{4}p^2 \rfloor$. However, for a tree T it was proved in [9] that the values of $b^*(T)$ and $c(T)$ are quite close:

Theorem 8.3. $b^*(T) \leqslant c(T) \leqslant b^*(T) + \log_2 b^*(T) + 2$. ||

These bounds are 'almost' best possible in the sense that, for each n, there exists a tree T with $b^*(T) = n$ and $c(T) \geqslant n + \log_2 n - 1$, and there exists a tree T' with $b^*(T') = c(T') = n$.

Stockmeyer (see [20]) proved the NP-completeness of the cutwidth problem:

Theorem 8.4. *The cutwidth problem for general graphs is NP-complete.* ||

Recently, Makedon *et al.* [35] proved that the cutwidth problem remains NP-complete when restricted to graphs with maximum degree 3. Gurari and Sudborough [26] proved that, for fixed k, the problem of determining whether the cutwidth is at most k is polynomial:

Theorem 8.5. *There is an $O(n^k)$ algorithm for deciding whether $c(G) \leqslant k$ for a graph G.* ||

Yannakakis [44] recently resolved the complexity problem for determining the cutwidth for trees:

Theorem 8.6. *There is an $O(n \log n)$ algorithm for determining the cutwidth for a tree.* ||

There are several results involving the relations of cutwidth with other graph invariants, such as the search numbers and graph pebblings (see [33], [38], [42]), which we shall not discuss here.

9. Concluding Remarks

Graph labeling is an active area within graph theory, having connections with a variety of application-oriented areas such as VLSI optimization, data structure and data representation. In the past few years, many new directions and new results have been developed while many questions remain unresolved. Here we mention several general directions for future research.

(*i*) Find good approximation algorithms for the NP-complete labeling problems mentioned in Sections 3–8. Such algorithms are very desirable because of the advancement in integrated circuit technology and the need in design automation.

(*ii*) Characterize graphs with good labelings—for example, the problem posed in Section 3 (does bounded density and no large complete binary subtree imply small bandwidth?) is one of the problems in characterizing graphs with small bandwidth.

(*iii*) Until now, most of the work has been concentrated on the case in which the host graph is a path. Results concerning the grid graphs as host graphs are quite recent and are known only for the bandwidth labelings. Most labeling problems using grid graphs as the host graphs have not been studied. Many of these problems are of particular interest in rectilinear network layout designs. There are many other good candidates for host graphs, such as trees and circuits; in taking a circuit as host graph, we deal with integers modulo n as labels. Taking various trees as the host graph is associated with problems in data structures, and has a special appeal of its own.

References

1. D. Adolfson and T. C. Hu, Optimal linear ordering, *SIAM J. Appl. Math.* **25** (1973), 403–423; *MR***49**#10349.
2. S. N. Bhatt and S. Cosmadakis, The complexity of minimizing wire lengths in VLSI layouts, preprint.
3. S. N. Bhatt and F. T. Leighton, A framework for solving VLSI graph layout problems, *J. Comput. System Sci.* **28** (1984), 300–343; *MR***86g**:68139.
4. I. Cahit, A conjectured minimum valuation tree, *SIAM Review* **19** (1977), 164.
5. P. Z. Chinn, J. Chvátalová, A. K. Dewdney and N. E. Gibbs, The bandwidth problem for graphs and matrices—a survey, *J. Graph Theory* **6** (1982), 223–254; *MR***84g**:05100.
6. F. R. K. Chung, A conjectured minimum valuation tree, Problems and solutions, *SIAM Review* **20** (1978), 601–604.
7. F. R. K. Chung, Some problems and results in labelings of graphs, *The Theory and Applications of Graphs* (ed. G. Chartrand *et al.*), John Wiley, New York, 1981, pp. 255–263.
8. F. R. K. Chung, On optimal linear arrangements of trees, *Computers and Math. with Applications* **10** (1984), 43–60; *MR***85b**:05065.
9. F. R. K. Chung, On the cutwidth and the topological bandwidth of a tree, *SIAM J. Discrete Alg. Methods* **6** (1985), 268–277; *MR***86k**:05036.
10. F. R. K. Chung and R. L. Graham, On graphs which contain all small trees, *J. Combinatorial Theory (B)* **24** (1978), 14–23; *MR***58**#21808.
11. V. Chvátal, A remark on a problem of Harary, *Czech. Math. J.* **20** (1970), 109–111; *MR***42**#1694.
12. J. Chvátalová, On the bandwidth problem for graphs, Ph.D. Thesis, Department of Combinatorics and Optimization, University of Waterloo, 1980.
13. J. Chvátalová, A. K. Dewdney, N. E. Gibbs and R. R. Korfhage, The bandwidth problem for graphs: a collection of recent results, Research Report #24, Department of Computer Science, University of Western Ontario, London, Ontario, 1975.
14. J. Chvátalová and J. Opatrný, Two results on the bandwidth of graphs, *Proc. Tenth Southeastern Conf. on Combinatorics, Graph Theory and Computing I, Congressus Numerantium XXIII*, Utilitas Math., Winnipeg, 1979, pp. 263–273; *MR***81f**#05064.

15. J. Chvátalová and J. Opatrný, The bandwidth problem and operations on graphs, preprint.
16. E. Cuthill and J. McKee, Reducing the bandwidth of sparse symmetric matrices, *Proc. 24th Natl. Conf. ACM*, 1969, pp. 157–172.
17. A. K. Dewdney, Tree topology and the *NP*-completeness of tree bandwidth, Research Report no. 60, Department of Computer Science, University of Western Ontario, London, Ontario, 1980.
18. P. G. Eitner, The bandwidth of the complete multipartite graph, Presented at *Toledo Symposium on Applications of Graph Theory*, 1979.
19. M. R. Garey, R. L. Graham, D. S. Johnson and D. E. Knuth, Complexity results for bandwidth minimization, *SIAM J. Appl. Math.* **34** (1978), 477–495; *MR***57**#18220.
20. M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W. H. Freeman, San Francisco, 1979; *MR***80g**:68056.
21. M. R. Garey, D. S. Johnson and R. L. Stockmeyer, Some simplified *NP*-complete graph problems, *Theoretical Comput. Sci.* **1** (1976), 237–267; *MR***53**#14978.
22. M. R. Garey, D. S. Johnson and R. L. Stockmeyer, Some simplified *NP*-complete problems, *Proc. 6th ACM Symposium on Theory of Computing*, 1974, pp. 47–63; *MR***54**#6549.
23. N. E. Gibbs, W. G. Poole, Jr. and P. K. Stockmeyer, An algorithm for reducing the bandwidth and profile of a sparse matrix, *SIAM J. Numer. Anal.* **13** (1976), 236–250; *MR***58**#19061.
24. M. K. Goldberg and I. A. Klipker, Minimal placing of trees on a line (Russian), Technical report, Physico-Technical Institute of Low Temperatures, Academy of Sciences of Ukranian SSR, USSR, 1976.
25. S. W. Golomb, How to number a graph, *Graph Theory and Computing* (ed. R. C. Read), Academic Press, New York, 1972, pp. 23–37; *MR***49**#4863.
26. E. M. Gurari and I. H. Sudborough, Improved dynamic programming algorithms for the bandwidth minimization problem and the min cut linear arrangement problem, Technical Report, Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, Ill., 1982.
27. F. Harary, *Graph Theory*, Addison-Wesley, Reading, Mass., 1969; *MR***41**#1566.
28. L. H. Harper, Optimal assignments of numbers to vertices, *J. Soc. Indust. Appl. Math.* **12** (1964), 131–135; *MR***29**#41.
29. L. H. Harper, Optimal numberings and isoperimetric problems on graphs, *J. Combinatorial Theory* **1** (1966), 385–393; *MR***34**#91.
30. M. A. Iordanskiĭ, Minimal numberings of the vertices of trees, *Soviet Math. Dokl.* **15** (1974), 1311–1315; *MR***50**#9649.
31. F. T. Leighton, New lower bound techniques for VLSI, *Proc. 22nd IEEE Symposium on Foundations of Computer Science*, 1981, pp. 1–12.
32. T. Lengauer, Upper and lower bounds on the complexity of the min-cut linear arrangement problem on trees, *SIAM J. Alg. Discrete Methods* **3** (1982), 99–113; *MR***83a**:68081.
33. N. Megiddo, L. Hakimi, M. R. Garey, D. S. Johnson and C. H. Papadimitriou, The complexity of searching a graph, *Proc. 22nd IEEE Symposium on Foundations of Computer Science*, 1981, pp. 376–385.
34. F. Makedon, Layout problems and their complexity, Ph.D. Thesis, Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, Ill., 1982.
35. F. Makedon, C. H. Papadimitriou and I. H. Sudborough, Topological bandwidth, preprint.
36. C. H. Papadimitriou, The *NP*-completeness of the bandwidth minimization problem, *Computing* **16** (1976), 263–270; *MR***53**#14981.
37. M. Paterson, W. Ruzzo and L. Snyder, Bounds on minimax edge length for complete binary trees, *Proc. 13th ACM Symposium on Theory of Computing*, 1981, pp. 293–299.
38. A. L. Rosenberg and I. H. Sudborough, Bandwidth and pebbling, preprint.
39. J. B. Saxe, Dynamic programming algorithms for recognizing small-bandwidth graphs in polynomial time, *SIAM J. Alg. Discrete Methods* **1** (1980), 363–369; *MR***82a**:68086.

40. M. A. Seidvasser, The optimal numbering of the vertices of a tree, *Diskret. Analiz* **17** (1970), 56–74; *MR*45#105.
41. Y. Shiloach, A minimum linear arrangement algorithm for undirected trees, *SIAM J. Comput.* **8** (1979), 15–32; *MR*80c:68034.
42. I. H. Sudborough and J. Turner, On computing the width and black/white pebbles demands of a tree, preprint.
43. M. M. Sysło and J. Zaks, The bandwidth problem for ordered caterpillars. Computer Science Department Report CS-80-065, Washington State University, 1980.
44. M. Yannakakis, A polynomial algorithm for the min-cut linear arrangement of trees, *Proc. 24th IEEE Symposium on Foundation of Computer Science*, 1983, pp. 274–281.