The Bandwidth Problem for Graphs and Matrices—A Survey

P. Z. Chinn
HUMBOLDT STATE UNIVERSITY
ARCATA, CA 95521
J. Chvátalová
VANIER COLLEGE
MONTREAL, QUEBEC, CANADA
A. K. Dewdney[1]
UNIVERSITY OF WESTERN ONTARIO
LONDON, ONTARIO, CANADA
N. E. Gibbs[2]
ARIZONA STATE UNIVERSITY
TEMPE, AZ 85281

ABSTRACT

The bandwidth problem for a graph G is to label its n vertices v_i with distinct integers $f(v_i)$ so that the quantity $\max\{|f(v_i) - f(v_j)| : (v_i, v_j) \in E(G)\}$ is minimized. The corresponding problem for a real symmetric matrix M is to find a symmetric permutation M' of M so that the quantity $\max\{|i-j| : m'_{ij} \neq 0\}$ is minimized. This survey describes all the results known to the authors as of approximately August 1981. These results include the effect on bandwidth of local operations such as refinement and contraction of graphs, bounds on bandwidth in terms of other graph invariants, the bandwidth of special classes of graphs, and approximate bandwidth algorithms for graphs and matrices. The survey concludes with a brief discussion of some problems related to bandwidth.

1. INTRODUCTION AND PRELIMINARY RESULTS

There are essentially two different forms of bandwidth problem. For a graph G the problem is to label the n vertices v_i of G with distinct integers $f(v_i)$ from $\{1, 2, \ldots, n\}$ so that the maximum value of

[1]Work supported by NSERC grant A9271.
[2]Work supported by ONR contract N00014-76-C-0673, NR044-459.

Journal of Graph Theory, Vol. 6 (1982) 223–254

CCC 0364-9024/82/030223-32$04.20

$$|f(v_i) - f(v_j)|,$$

taken over all adjacent pairs of vertices, is a minimum. For a real, symmetric matrix M the problem is to find a symmetric permutation M' of M so that the maximum value of

$$|i-j|$$

taken over all nonzero entries m'_{ij} is a minimum.

The equivalence of the two problems is made clear by replacing the nonzero entries of M by 1's and interpreting the result as the adjacency matrix of a graph.

The matrix bandwidth problem seems to have originated in the 1950s when structural engineers first analyzed steel frameworks by computer manipulation of their structural matrices: In order that operations like inversion and finding determinants take as little time as possible, the attempt was made to discover an equivalent matrix in which all the nonzero entries lay within a narrow band about the main diagonal—hence the term "bandwidth."

The bandwidth problem for graphs, meanwhile, originated independently at the Jet Propulsion Laboratory at Pasadena in 1962: single errors in a 6-bit picture code were represented by edge differences in a hypercube whose vertices were words of the code. At JPL, L. H. Harper and A. W. Hales sought codes which minimized the maximum absolute error and the average absolute error. Thus were born the bandwidth and bandwidth sum [57] problems—at least for cubes. Not long after this, R. R. Korfhage [63] began work on the graph bandwidth problem and F. Harary [56] publicized the problem at a conference in Prague.

Since the mid-sixties, there has been a strong interest in the bandwidth problem for graphs, with a steadily growing body of theory developing alongside the continuing search for better bandwidth minimization algorithms for matrices. This survey describes all the results known to the date of writing, including the discovery by Papadimitriou [75] that the bandwidth problem for graphs—and hence for matrices—is NP-complete.

The survey is organized into five sections. Section 2 concentrates on the effect of local operations such as refinement and edge removal on the bandwidth of a graph: what is the maximum amount by which bandwidth can change when such operations are performed? Section 3 includes all the knwon bounds on bandwidth in terms of other graph invariants such as connectivity and maximum degree. Section 4 concentrates on the problem of finding the bandwidth of special classes of graphs, including trees, bipartite graphs, and powers of paths. Section 5 discusses both exact and approximate bandwidth algorithms. A final section discusses problems having a close relationship to graph bandwidth. For those readers wishing closer contact with the subject, a few of the simpler theorems are followed by brief sketches of their proofs.

Let $G = (V, E)$ be a graph on n vertices. A 1-1 mapping $f: V \rightarrow Z$, the integers, will be called a *numbering* of G and if $f(V) = \{1, 2, \ldots, n\}$, then f will be called a *proper numbering* of G. The *bandwidth of a numbering f for G*, denoted by $B_f(G)$, is the number

$$\max\{\,|f(u) - f(v)|\, : (u, v) \in E(G)\},$$

and the *bandwidth of G*, denoted by $B(G)$, is the number

$$\min\{B_f(G) : f \text{ is a numbering of } G\}.$$

This definition could well have been made by restricting f to be a proper numbering of G, since there is a procedure [21, p. 3] for converting a numbering f into a proper numbering f' such that $B_{f'}(G) \leq B_f(G)$. The above definition is useful, however, in proofs which manipulate numberings by various arithmetic and combinatorial means.

Besides the foregoing equivalence of definitions, there are two other preliminary results whose proofs are very easy. At this point, a formal list of theorems is begun.

Theorem 1.1. If H is a subgraph of G, then $B(H) \leq B(G)$.

Theorem 1.2. If G has components $G_1, G_2, \ldots, G_m$, then

$$B(G) = \max\{B(G_1), B(G_2), \ldots, B(G_m)\}.$$

The next theorem, due to Harper [57], is used in the proof of several results appearing in subsequent sections. The symbol ∂S denotes the set of vertices in S adjacent to those in $V(G) - S$.

Theorem 1.3. For any connected graph G,

$$B(G) \geq \max_{k} \min_{|S| = k} |\partial S|\,.$$

This result is more or less immediate on observing that

$$\partial S_k \subseteq f^{-1}\{k, k-1, \ldots, \max\{1, k - B(G) + 1\}\},$$

where $S_k \equiv f^{-1}\{1, 2, \ldots, k\}$.

It is frequently useful to visualize the bandwidth of a graph G as an embedding of G in P_n^k, the kth power of a path on n vertices.

Theorem 1.4. A graph G on n vertices has bandwidth k if and only if k is the smallest integer such that G can be embedded in P_n^k.

For example, any graph on nine vertices having bandwidth three may be embedded in P_9^3, shown in Figure 1.

2. BANDWIDTH AND LOCAL OPERATIONS

The local operations referred to here are elementary refinements, contraction, addition of an edge and merger of two vertices. The work reported in this section is due to Chvátalová [19] and to Chvátalová and Opatrný [23]. Its general nature is to show that under a local operation, the bandwidth of a graph may increase or decrease, and the amount of change is not necessarily small.

2.1. Refinements

An *elementary refinement* of a graph G is obtained by replacing an edge (u, v) of G by a path of length two having endvertices u and v. A *refinement of* G is the result of a finite number of elementary refinements.

Under the operation of elementary refinement, the bandwidth of a graph may either increase or decrease. It cannot increase, however, by more than 1, as shown by Chvátalová [19].

Theorem 2.1.1. If H is an elementary refinement of G, then

$$B(H) \leq B(G) + 1.$$

On the other hand, Chvátalová and Opatrný [23] have shown that it is possible for the bandwidth of a graph to drop by as much as one quarter of its value under an elementary refinement.

Theorem 2.1.2. If H is an elementary refinement of a graph G, then

$$B(H) \geq \frac{3\,B(G) - 1}{4}.$$

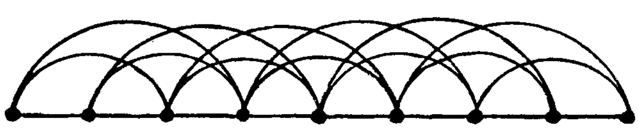

FIGURE 1. The third power of P_8.

Theorem 2.1.3. For every $n > 0$, there exists a graph G with an elementary refinement H of G such that $B(G) > n$ and

$$B(H) = \left\lceil \frac{3B(G) - 1}{4} \right\rceil .$$

The next four results, involving refinements, are due to Chvátalová [19].

Theorem 2.1.4. If H is a refinement of G, then $B(H) \geq \delta(G)$.

Theorem 2.1.5. If H is a refinement of $K_{2,n}$, then $B(H) \geq \lfloor (n + 3)/2 \rfloor$.

Theorem 2.1.6. If H is a refinement of B_n, then $B(H) \geq \lfloor (n + 4)/3 \rfloor /3$.

In the last theorem, B_n dentoes the full binary tree of depth n. In the case of such trees, the lower bound of Theorem 2.1.4 cannot be improved by more than a constant factor.

Theorem 2.1.7. For each $n \geq 1$, there is a refinement H_n of B_n such that $B(H_n) \leq \lfloor n/2 \rfloor + 1$.

2.2. Contractions, Edge-Additions, and Merger of Two Vertices

Given a graph G, denote by $G|_{u,v}$ the *merger of two vertices u, v of G* which results when the two vertices u, v are identified and the possible resulting loop and extra edges are removed from G. If u and v are the vertices of an edge e, $G|_e$ denotes the *contraction of G along e* which is isomorphic to $G|_{u,v}$.

As in elementary refinements, the bandwidth of a graph may either increase or decrease as a result of a merger or a contraction. However, the relative latitudes for such changes in bandwidth are the reverse of those for elementary refinements, since one can undo the effect of a refinement by a contraction.

Theorem 2.2.1 (Chvátalová [19]. If G is any graph and $e \in E(G)$, then

$$B(G|_e) \geq B(G) - 1.$$

This theorem is proved by altering a bandwidth numbering f for $G|_e$ by defining

$$\begin{aligned} f'(w) &= f(w) \text{ when } f(w) \leq f(u), \\ &= f(w) + 1 \text{ when } f(w) > f(u), \text{ and} \\ f'(v) &= f(u) + 1, \end{aligned}$$

where $e = \{u, v\}$. The resulting numbering f' establishes the inequality.

Chvátalová and Opatrný have discovered the following upper bounds for bandwidth increase under contractions and mergers.

Theorem 2.2.2. If G is a graph and $e \in E(G)$, then

$$B(G|_e) \leq \left\lceil \frac{3B(G) - 1}{2} \right\rceil .$$

Theorem 2.2.3. If G is a graph and u, v vertices of G, then

$$B(G) - 1 \leq B(G|_{u,v}) \leq 2B(G).$$

Theorem 2.2.4. The upper and lower bounds in Theorems 2.2.1, 2.2.2, and 2.2.3 cannot be improved.

Denote by $G + e$ a graph obtained from G by *adding an edge e*. Erdös asked whether $B(G + e) - B(G) \leq 1$. A negative answer provided by Chvátalová [19, p. 127] qualified her for a tax-free prize of $5.00.

Theorem 2.2.5. If G is a graph and u, v vertices of G, then

$$B(G + (u, v)) \leq 2B(G).$$

Chvátalová and Opatrný [23] have discovered that while the bound of the foregoing theorem is attained, in general [23], it can be improved when the distance between u and v is taken into account, specifically when $d(u, v) \leq 3$.

Theorem 2.2.6. If G is a graph and u, v are vertices of G, then

$$B(G + (u, v)) \leq \left\lfloor \frac{(d(u, v) + 2)B(G)}{3} \right\rfloor .$$

3. BANDWIDTH OF FINITE AND INFINITE GRAPHS

3.1. Upper and Lower Bounds on the Bandwidth of Finite Graphs

Given some graph-theoretic invariant, say x, it is frequently easier to obtain a lower than an upper bound for $B(G)$ in terms of $x(G)$. This is due to the fact that x frequently restricts the possibilities for bandwidth numberings of G, thereby imposing minimum values on $B(G)$, while x rarely gives good information about how to construct numberings f for which $B_f(G)$ is very close to $B(G)$. In this subsection we examine a number of lower bounds for $B(G)$ and a few upper bounds, all involving simple and well-known invariants x. The number of such bounds, all best possible and all involving different invariants, is somewhat remarkable.

The first two theorems, among the few proved in this section, involve the number p of vertices, number q of edges, and maximum degree Δ of a graph. The first is due to Chvátal [18] and the second to Dewdney [21].

Theorem 3.1.1. For any graph G,

$$B(G) \geq p(G) - \frac{2 - [(2p(G) - 1)^2 - 8q(G)]^{1/2}}{2}.$$

This result may be obtained readily by analyzing G as a subgraph of P_p^k as in Theorem 1.4.

Theorem 3.1.2. For any graph G, $B(G) \geq [\Delta(G)/2]$.

At a vertex v of maximum degree, the set of differences $|f(v) - f(u)|$ with adjacent vertices u must have at least $\Delta(G)/2$ distinct members.

The bounds implied by the two foregoing theorems are attained when $G = K_p$ and $K_{1,p}$ respectively. In the next theorem, due to Chvátal, it is not so clear which graphs realize the bound.

Theorem 3.1.3. For any graph G with degree sequence $d_1, d_2, \ldots, d_p$ satisfying $d_1 \leq d_2 \leq \cdots \leq d_p$,

$$B(G) \geq \max_j \{d_j - \lfloor (j-1)/2 \rfloor, d_j/2\}.$$

This result implies Theorem 3.1.2 but does not seem to imply the next one, which involves only the minimum degree δ.

Theorem 3.1.4. For any graph G containing no triangles

$$B(G) \geq \lfloor (3\delta(G) - 1)/2 \rfloor.$$

This bound is attained when $G = K_{n,n}$.

The next two results, by Dewdney [21], illustrate the remarkable nature of the relationship between bandwidth and other graph invariants such as the connectivity κ and chromatic number χ of a graph G.

Theorem 3.1.5. For any graph G, $B(G) \geq \chi(G) - 1$.

Theorem 3.1.6. For any graph G, $B(G) \geq \kappa(G)$.

In the first of these theorems it will be observed that $B(K_n) = \chi(K_n) - 1$ while in the second, $B(P_n^m) = \kappa(P_n^m)$ for all m and $n \geq m + 1$. It follows that neither inequality can be improved with respect to these invariants.

There are relatively simple upper and lower bounds for the bandwidth of a graph in terms of its vertex-independence number β_0, as shown by Chvátal [18] and Chvátalová [21], respectively.

Theorem 3.1.7. For any graph G

$$\lceil p(G)/\beta_0(G) \rceil - 1 \leq B(G) \leq p(G) - \lfloor \beta_0(G)/2 \rfloor - 1.$$

The first inequality is exact for K_n, P_n and C_n while the second is exact for the join of K_n and $\overline{K}_m$, where $m = \beta_0(G)$ and $n = p(G) - m$.

3.2. Bounds on the Bandwidth of Connected Graphs

When G is required to be connected, several new bounds on the bandwidth of G become possible. Chvátal [18] and Dewdney [21], respectively, bound the bandwidth in terms of the diameter of a graph.

Theorem 3.2.1. If G is a connected graph, then

$$\lceil (p(G) - 1)/D(G) \rceil \leq B(G) \leq p(G) - D(G).$$

The first inequality is proved by observing the independently useful fact that if f is a proper bandwidth numbering of G, then $|f(u) - f(v)| \leq B(G) \cdot d(u, v)$. The second inequality is obtained by selecting two vertices u and v at distance $D(G)$ apart in G, defining classes $V_k = \{w \in V(G) : d(u, w) = k\}$ and numbering the vertices in these classes, from $k = 0$ to $k = D(G)$, level by level.

The lower bound is attained by P_n (the path on n vertices), while the upper bound is attained when G is the graph formed by attaching one end of $P_{D(G)-1}$ to K_n, where $n = p(G) - D(G) + 1$.

Let G be a connected graph. The eccentricity of G, defined by

$$e(G) = \min_u \max_v d(u, v),$$

provides an exponential bound on $B(G)$ as a power of $\Delta(G)$.

Theorem 3.2.2. For any connected graph G

$$B(G) \leq \Delta(G) \cdot (\Delta(G) - 1)^{e(G)-1}.$$

As remarked in [21], this bound is only effective when $\Delta(G$ is small and $e(G)$ itself is bounded.

The final result in this subsection is due to Chvátalová [19] and uses Theorem 1.3 in its proof.

Theorem 3.2.2. For any connected graph G,

$$B(G) \geq \max_{k} \min_{|S|=k} \frac{[1 + 8|\Delta S|]^{1/2} - 1}{2},$$

where ΔS denotes the set of all edges of G with only one end in S.

Although computationally inefficient, this result might be specialized to yield effective bounds by restricting k.

3.3. Bounds on the Bandwidth of a Graph and its Complement

This subsection studies the relationship of $B(G)$ to $B(\overline{G})$ and to various structural properties of $\overline{G}$. Chvátalová [19] found the following equivalent theorems.

Theorem 3.3.1. For any graph G, $B(G) \leq p(G) - 3$ if and only if $\overline{G}$ contains P_4.

Theorem 3.3.2. For any graph G, $B(G) \leq p(G) - 2$ if and only if every component of $\overline{G}$ is a vertex, star, or triangle.

By Theorem 3.2.2, $B(G) \leq p(G) - D(G)$. If it should happen that $D(G) = 2$, the following corollary results.

Corollary 3.3.3. If $D(G) = 2$ then $B(G) = p(G) - 2$ if and only if every component of $\overline{G}$ is a vertex, a star, or a triangle.

The following result by Chinn et al. [17] displays an interesting relationship between the bandwidths of G and of its complement.

Theorem 3.3.4. For any graph G,

$$B(G) + B(\overline{G}) \geq p(G) - 2.$$

The proof depends on showing that $B(\overline{P}_{2m}^{m-1}) = m - 1$, a result also recently found independently by Kahn and Kleitman [60]. The following strong upper bound on $B(G) + B(\overline{G})$ is also established in [17].

Theorem 3.3.5. There is a constant $c > 0$ such that for every p, all graphs on p vertices satisfy

$$\text{(a)}\ B(G) + B(\overline{G}) \leq 2p - c \log p.$$

There is also a constant $c' > 0$ such that for every p, some graph G on p vertices satisfies

$$\text{(b)}\ B(G) + B(\overline{G}) \geq 2p - c' \log p.$$

Obtaining exact values of c and c' appears to depend on knowing the asymptotic behavior of Ramsey numbers, a difficult problem.

3.4. Bandwidth of Infinite Graphs

Much of the emphasis on work in infinite graph bandwidth has been to determine when a denumberably infinite connected graph G has finite bandwidth. In 1973, Erdös conjectured that if there exists a constant c (depending on G) for which the following statement is true, then G has finite bandwidth.

> "For every vertex $v \in V(G)$ and for every positive integer k, $|\{x \in V(G) : d(v, x) \leq k\}| \leq ck$."

This conjecture has been disproved by Chvátalová; the result appears in Sec. 4 where special graphs, including infinite trees, are discussed.

The two remaining theorems of this subsection, both by Chvátalová [19], give sufficient conditions for an infinite graph to have finite bandwidth and infinite bandwidth, respectively.

Theorem 3.4.1. Let G be a denumberable connected graph. If $B(G') \leq k$ for every finite subgraph G' of G, then $B(G) \leq 2k$.

Theorem 3.4.2. If G is a denumerable graph containing infinitely many disjoint infinite paths, then $B(G)$ is infinite.

One may prove the second theorem by showing that the existence of k disjoint infinite paths forces G to have bandwidth $> k$; the latter inequality follows from the existence of a constant M such that each path in the collection has bandwidth values on either side of M.

4. BANDWIDTH OF SPECIAL GRAPHS

Special graphs, as well as providing test cases for conjectures, are the source of a number of interesting and difficult problems. The special graphs discussed in this section include P_n, K_n, K_{mn}, Q_n, trees, (finite and infinite), caterpillars, graph products such as cartesian product, join and corona, as well as some miscellaneous special graphs.

4.1. Indexed Graphs

These graphs include those special graphs uniquely determined by a single integer or small number of integers, given some strong structural feature. The first few results listed here may be derived immediately and are lumped into a single theorem.

Theorem 4.1.1. $B(P_n) = 1$, $B(C_n) = 2$, $B(K_n) = n - 1$, $B(K_{m,n}) = (m-1)/2 + n$, where $m \geq n$.

The only equality which is not immediately obvious is the last: number the smaller set of vertices in $K_{m,n}$ with the numbers $m/2 + 1$ to $m/2 + n$ and those which remain with the unused numbers from $\{1, 2, \ldots, m+n\}$.

A result by Eitner [32], involving complete k-partite graphs, generalizes this last formula.

Theorem 4.1.2. If $n_1 \geq n_2 \geq \cdots \geq n_k$ and p is the number of vertices then

$$B(K_{n_1, n_2, \ldots, n_k}) = p - \left\lceil \frac{n_1 + 1}{2} \right\rceil.$$

Harper [57], using Theorem 1.3, found the following exact formula for the bandwidth of the n-cube Q_n.

Theorem 4.1.3.

$$B(Q_n) = \sum_{k=0}^{n-1} \binom{k}{\lfloor k/2 \rfloor}.$$

4.2. Trees-Finite and Infinite

For a finite tree, there is an upper bound on its bandwidth due to Chvátalová [21] and some implicit lower bounds due to Zak [92]. A special class of trees with bounded degree is investigated and this subsection closes with Chvátalová's refutation of the conjecture by Erdös introduced in Sec. 3.

Theorem 4.2.1. If T is a tree, then $B(T) \leq p(T)/2$ with equality if and only if T is a star and $p(T)$ is even.

In bounding the bandwidth of a graph G below, one may search for a subgraph G' having the property that

(1) $B(G) = B(G')$, and

(2) $B(G') > B(G'')$ for any subgraph G'' of G'.

The same search may be made for a subgraph having a maximum value for any of the functions which bound $B(G)$ below, for example, the one appearing in Theorem 3.2.1:

$$B(G) \geq \lceil (p(G) - 1)/D(G) \rceil .$$

Denoting this lower bound by $h(G)$ we may define a graph G to be *h-critical* if $h(G) > h(H)$ for every proper subgraph H of G. The main results for h-critical trees are due to Zak [92].

Theorem 4.2.2. Let T be an h-critical tree. Then

(1) $p(T) = (h(T) - 1)D(T) + 2$, and

(2) if $D(T) > 2$, then

$$\Delta(T) \leq 2h(T) - 2 \text{ and } P_{ed}(T) \geq 2h(T) - 1,$$

where $P_{ed}(T)$ is the number of vertices in T whose eccentricity is $D(T)$.

It can happen, of course, that an H-critical subtree T' of T will attain a much larger h-value than will T. This improves the lower bound of Theorem 3.2.1 for T and makes H-critical trees worth looking for. The two theorems above are steps toward characterizing H-critical trees. Zak has found all H-critical trees of diameter 2, 3, and 4.

Chinn [16] has recently characterized trees of bandwidth 3 which are critical with respect to deletion of vertices and contraction of edges.

An interesting tree for which the precise bandwidth is currently unknown is denoted by $T_{m,n}$ and defined as a rooted tree in which the root and all vertices at distance $<n$ from the root have degree m. The remaining vertices have degree 1.

Theorem 4.2.3.

$$\left\lceil \frac{m(m-1)^n - 1}{2n(m-2)} \right\rceil \leq B(T_{m,n}) \leq m(m-1)^{n-1}.$$

It is suspected that the lower bound above is closer to the true value of $B(T_{m,n})$ than the upper bound. This class of trees was defined in [21] as a possible example of a class of graphs having bounded degree (m fixed) yet achieving near maximum bandwidth as a function of n.

The next theorem gives sufficient conditions for an infinite tree to have a refinement whose bandwidth is finite. As shown in [22], these conditions are also necessary.

Theorem 4.2.4. Let T be a denumerable tree such that for some constants, k, m, n,

(1) T has degrees bounded by k,

(2) the number of edge-disjoint semi-infinite paths in T is at most n, and

(3) T does not contain a refinement of any binary tree having depth $\geq n$.

Then there exists a refinement T' of T, having finite bandwidth.

The refinement T' cannot be T itself, as the next theorem shows.

Theorem 4.2.5. There exists a denumerable tree which satisfies conditions (1), (2), and (3), and does not have finite bandwidth.

The proof of this theorem uses the construction displayed in Figure 2.

4.3. Bandwidth of Graph Products

Many kinds of graph products may be defined, including cartesian product, corona and composition. Given two graphs G_1 and G_2, the *(cartesian) product of* G_1 *and* G_2, written $G_1 \times G_2$, is the graph whose vertex set is $V(G_1) \times V(G_2)$, two vertices (u_1, u_2) and (v_1, v_2) being adjacent in $G_1 \times G_2$ if either $u_1 = v_1$ and u_2 is adjacent to v_2 or *vice versa*.

The next three results are from Chvátalová [20], [21]. The first relates the bandwidth of a product to the bandwidth of its components, while the remaining two compute the bandwidth of products involving cycles and paths.

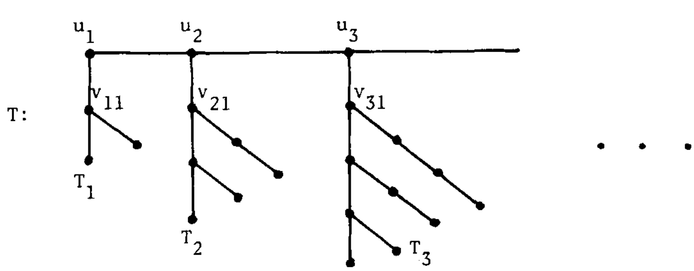

FIGURE 2. A tree satisfying conditions (1), (2), and (3).

Theorem 4.3.1. For any two graphs G_1 and G_2,

$$B(G_1 \times G_2) \leq \min\{p(G_2)\cdot B(G_1), p(G_1)\cdot B(G_2)\}.$$

Corollary 4.3.2. For any n graphs $G_1, \ldots, G_n$,

$$B(G_1 \times \cdots \times G_n) \leq \min\left\{B(G_i)\cdot \prod_{j\neq i} p(G_j) : i = 1, \ldots, n\right\}.$$

Theorem 4.3.3. If $\max\{m, n\} \geq 2$, then $B(P_m \times P_n) = \min\{m, n\}$.

This theorem and the following one employ essentially the same method of proof, making use of Harper's theorem (1.3).

Theorem 4.3.4. If $m \geq 2$ and $k \geq 3$, then $B(P_m \times C_k) = \min(2m, k)$.

Given two graphs G and H, the *corona* $G \circ H$, is the graph whose vertex set is $V(G) \cup V(H_1) \cup \cdots \cup V(H_m)$, where $m = |V(G)|$ and each H_i is a copy of H, one for each vertex in $V(G)$. Two vertices u and v in $G \circ H$ are adjacent if $u = u_i \in V(G)$ for some i and $V \in V(H_i)$. Note that the corona is not commutative. Chinn [14] has proved the following result.

Theorem 4.3.5. For any two graphs G and H,

$$B(G \circ H) \leq B(G)\cdot(p(H) + 1).$$

The bound is attained when $G = C_n$ and $H = K_1$, provided that $n \geq 9$.

Given two graphs G and H, *the composition of G and H*, written $G(H)$, is the graph with vertex set $V(G) \times V(H)$ and (u_1, v_1) is adjacent to (u_2, v_2) if u_1 is adjacent to u_2 in G *or* $u_1 = u_2$ and v_1 is adjacent to v_2 in H.

The composition of two graphs obeys an inequality, similar to that in the foregoing theorem, obtained by Dewdney and Chinn [14]. Equality is attained when $G = K_n$ and $H = K_m$.

Theorem 4.3.6. For any two graphs G and H,

$$B(G(H)) \leq (B(G) + 1)p(H) - 1.$$

4.4. The Bandwidth of Miscellaneous Graphs

The following sorts of graphs are discussed in this subsection: nonplanar graphs, cubic graphs, powers of graphs, multipaths, caterpillars, and Möbius ladders.

Chvátalová [21] observed that if $B(G) < 4$, then G must be a subgraph of the planar graph P_n^3.

Theorem 4.4.1. If G is a nonplanar graph, then $B(G) \geq 4$.

A graph G has *cyclic edge connectivity 3* (CEC3) if 3 is the minimum number of edges whose removal from G results in two graphs, each containing a cycle.

Gibbs [47] has found a procedure for constructing precisely those cubic CEC3 graphs having bandwidth three. Starting with either of two types A or B of graph on six vertices, the construction adds three new edges and two vertices at each stage in either of two ways (A or B), as shown in Figure 3.

Theorem 4.4.2. A graph constructed by a sequence of As and Bs has bandwidth 3 if and only if the sequence contains no two consecutive Bs.

As a corollary to this theorem, one obtains the following enumeration result.

Corollary 4.4.3. The number of bandwidth-3 numbered cubic CEC3 graphs on 2n vertices is F_{n-1}, the $(n-1)$st Fibonacci number.

The next two theorems on cubic CEC3 graphs are due to Korfhage and Gibbs [64].

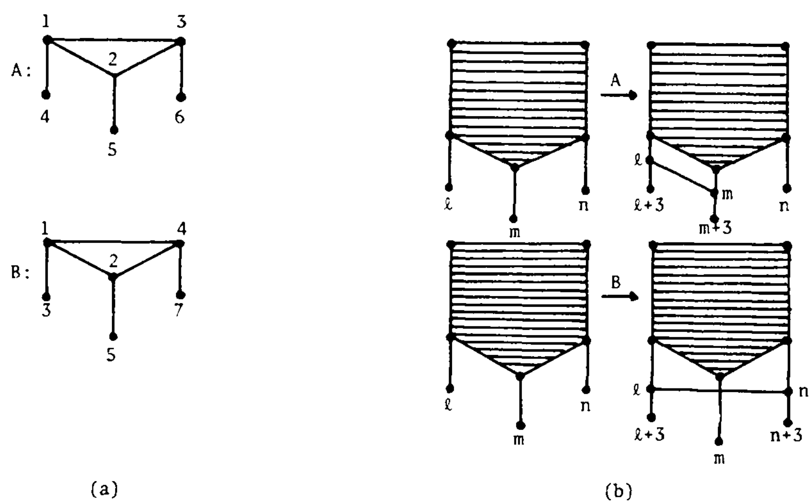

FIGURE 3. Initial graphs (a) and construction steps (b) for cubic CEC3 graphs.

Theorem 4.4.4. If a cubic CEC3 graph G has bandwidth 3, then G is Hamiltonian and 3-edge colorable.

Theorem 4.4.5. The number of nonisomorphic cubic CEC3 graphs on $2n$ vertices and having bandwidth 3 is at most $2F_{n-3}$, the $(n-3)$rd Fibonacci number.

Chvátalová [21] has shown that the bandwidth of G^n, the nth power of G, cannot exceed the bandwidth of G by more than a factor of n. Equality holds when $G = P_m$, $m > n$.

Theorem 4.4.6. For any graph G and any positive integer n,

$$B(G^n) \leq n \cdot B(G).$$

A graph consisting of two vertices u and v along with n disjoint paths of length $m + 1$ connecting u and v will be called an *(n, m)-multipath* and denoted by $P_{n,m}$. It is useful to know the bandwidth of (n, m)-multipaths since they can frequently be found in engineering structures and help provide a lower bound for the bandwidth of the structure in such cases. The following theorems, due to Dewdney and Korfhage [21], represent a partial solution to the problem of determining $B(P_{n,m})$.

Theorem 4.4.7. If $m \geq n$, then $B(P_{n,m}) = n$.

Theorem 4.4.8. If $m \geq \lceil n/2 \rceil$, then $B(P_{n,m}) \geq n - 1$.

It is conjectured [21] that whenever $m \geq \lceil n/k \rceil$, $B(P_{n,m}) \geq n - k + 1$.

One special class of tree is a *caterpillar*, a tree which yields a path when all its pendant vertices are removed. A caterpillar is called *ordered* if the degrees of vertices along this path form a monotonic sequence. Given an ordered caterpillar with nonincreasing degrees $d(u_1) \geq d(u_2) \geq \cdots \geq d(u_m)$, denote by T_i the subcaterpillar induced by $u_1, u_2, \ldots, u_i$ and all vertices adjacent to these. Syslo and Zak [85] have the following exact expression for the bandwidth of an ordered caterpillar, and give a linear algorithm for computing the bandwidth.

Theorem 4.4.9. If T is an ordered caterpillar with m nonpendant vertices, then

$$B(T) = \max\{h(T_i) : i = 1, 2, \ldots, m\},$$

where h is the function defined by $h(G) = \lceil (p(G) - 1)/D(G) \rceil$.

Using this result, Syslo and Zak are able to find the bandwidths of two-stars and combs, a problem posed by Hedetniemi [59]. The *two-star* $S_{m,n}$ is the caterpillar with two nonpendant vertices. These vertices have degrees m and n, taking $m \geq n$ by convention. The *comb* D_n is a caterpillar with n nonpendant vertices each incident with exactly one pendant vertex.

Theorem 4.4.10.

$$B(S_{m,n}) = \max\left\{\left\lceil \frac{m}{2} \right\rceil, \left\lceil \frac{m+n-1}{2} \right\rceil\right\},$$

$$B(D_{2n}) = 2.$$

The *Möbius ladder* M_n is a circuit of length n with all pairs of vertices at distance $\lfloor n/2 \rfloor$ apart in the circuit joined by an edge. As shown by Chinn [15], using many of the inequalities developed in Secs. 2, 3, and 4, the bandwidth of M_n is constant.

Theorem 4.4.11. $B(M_n) = 4$.

5. BANDWIDTH ALGORITHMS

Bandwidth algorithms may be divided into two classes: slow, exact algorithms and reasonably fast, approximate algorithms. The reasons for this seemingly mutually exclusive division lie in the computationally intractable nature of the bandwidth problem. This section is divided accordingly.

5.1. Exact Algorithms

Since the early 1970s there has come into existence a steadily growing class of computational problems called NP-complete [40]: A problem in this class* has the property that if a polynomial-time algorithm is ever found for it, then all the problems in NP (a class which would appear to include almost) all the combinatorial decision problems ever formulated for computer solution) will also have polynomial-time algorithms. Papadimitriou [75] first showed that the bandwidth problem lay in this class. In more formal terms, the *bandwidth problem* consists in discovering, for each graph G and positive integer k as input, whether or not G has a numbering whose bandwidth does not exceed k. A *polynomial-time solution* of this problem is an algorithm which will, in all cases, make such a discovery in time $p(n)$, where p is some fixed polynomial and n is a measure of the size [41] of G and k.

*Not to be confused, as is sometimes done, with the class NP.

Theorem 5.1.1. The bandwidth problem is NP-complete.

This result is taken as a good reason to be pessimistic about ever finding a polynomial-time solution of the bandwidth problem, for no NP-complete problem (and there are now hundreds of these) has ever been shown to have a polynominal-time solution.

Indeed, there are now much sharper theorems concerning the bandwidth problem. These involve laying restrictions on the graphs admitted as instances of the problem. For example, if the graph G in the formulation of the bandwidth problem above is a tree with no degree exceeding three, the problem is still difficult, as shown by Garey, Graham, Johnson, and Knuth [39].

Theorem 5.1.2. The bandwidth problem for trees T with $\Delta(T) = 3$ is NP-complete.

A relatively simple proof of the NP-completeness of bandwidth for trees without the degree restriction may be found in [31].

It might be asked whether fixing the bandwidth to be found results in an easier problem. The *bandwidth-k problem* thus consists in discovering, for each graph G as input, whether or not $B(G) \leq k$. When $k = 1$, Gibbs and Poole [48], as well as Faulkerson and Gross [39], have found polynomial time solutions for the bandwidth-1 problem: When is a graph a disjoint collection of paths?

Theorem 5.1.3. The bandwidth-1 problem has a linear-time solution.

Garey et al. were able to find a polynomial-time solution for the case $k = 2$.

Theorem 5.1.4. The bandwidth-2 problem has a linear-time solution.

Until recently, it was thought that the bandwidth-k problem might be NP-complete for some value of $k \geq 3$. However, Saxe [81] has shown that when k is fixed, the bandwidth-k problem always has a polynomial solution.

Theorem 5.1.5. There exists an algorithm which solves the bandwidth-k problem in time $O(f(k)n^{k+1})$, where n is the number of vertices in the graph and $f(k)$ is a function depending only on k.

5.2. Approximate Algorithms

As mentioned in the introduction, the bandwidth problem for matrices has been important in engineering applications for over two decades. Such applications include not only structural engineering but fluid dynamics and network analysis. Typically, the matrices arising in such applications are sparse, and to solve a corresponding system of equations, it is tempting to find a band-limited permutation of the matrix. Unmodified Gaussian elimination* requires $O(n^3)$ operations to solve an $n \times n$ system, but if a permutation of the matrix which places all nonzero entries with b of the main diagonal can be found, the number of solution steps is reduced to $O(nb^2)$. An algorithm whose purpose is to reduce b as far as possible (whether for a matrix or its equivalent graph) will be called a *bandwidth reduction algorithm*. Such algorithms, in view of the results of the foregoing subsection, are best regarded as approximate in the sense that they are not guaranteed always to discover the actual bandwidth of the matrix. On the other hand, it usually makes little difference in most applications whether one has reduced the bandwidth of a matrix as far as is theoretically possible; *any* reduction in the band-size b of the current matrix is desirable.

In a 1979 survey paper by Everstine [36], 49 reduction algorithms are cited. It is safe to assume, however, that many more† algorithms than this have been designed to date, perhaps up to twice as many. Such production algorithms have been largely heuristic in nature and, no doubt, the vast majority were designed by applications programmers unfamiliar with the NP-completeness of the problem.

In the descriptions of bandwidth reduction algorithms appearing below, the theorem format of the foregoing sections will be dropped and each algorithmic approach will be discussed informally. Because of the applications for which algorithms were intended, matrices will be the focus of some of them.

The first bandwidth reduction algorithm, published by Alway and Martin [5] in 1965, examined a large number of permutations using rather complex criteria to discard unwanted ones. Although the algorithm was effective only in reducing the bandsize of small matrices, it stimulated further research.

The first widely used production algorithm, due to Rosen [80] in 1968, starts with an arbitrary numbering of the matrix and then, by successive stages, attempts to reduce it. At each stage the nonzero entries at the edge of the band are examined and a search is made for symmetric interchange permutations which result in zeros in those locations. When such permutations are made, the new band size is computed. The entire process is iterated until no improvement in the band size is observed. Although the algorithm uses a local strategy in terms of the entire structure of the associated graph and is, therefore, time consuming, it was still being used in its production mode by the late 1970s.

---

*Although out of vogue for a period of time, combined Gaussian elimination-bandwidth reduction algorithms have been used increasingly on vector processing machines such as the Cray and the CDC Star [69].

†See Reference List.

In 1969 the well-known Cuthill–McKee algorithm [29] appeared and was subsequently incorporated into NASTRAN [35]. This algorithm and a closely related one are now described as in [49], using the notation from that paper.

A *level structure of a graph* G, denoted by $L(G)$, is a partition of $V(G)$ into sets $L_1, L_2, \ldots, L_k$ called *levels* which satisfy the following condition:

For $1 < i < k$, all vertices adjacent to vertices in L_i are either in L_{i-1}, L_i, or L_{i+1}, with $L_0 = L_{n+1} = \emptyset$.

A level structure of particular interest is the *level structure rooted at* v, denoted by $L_v(G)$. In this structure

$$L_i = \{u \in V(G) : d(u, v) = i - 1\}.$$

Given a level structure $L(G)$, the *width* $w(L)$ and *depth* $d(L)$ are $\max_i |L_i|$ and the number of levels, respectively. By numbering G arbitrarily, level by level, it is not hard to see that for such a numbering f,

$$B(G) \leq B_f(G) \leq 2w(L) - 1.$$

Moreover, if L is a rooted level structure $L_v(G)$, then

$$B_f(G) \geq w(L_v).$$

The Cuthill–McKee algorithm begins by generating a level structure rooted at each vertex of "low degree" [29]. For each rooted level structure having minimum width and generated in the first stage, the graph is numbered level-by-level with consecutive positive integers, starting with 1 at the root. For each successive level, the vertices adjacent to the lowest-numbered vertex in the previous level are numbered in order of increasing degree, ties being broken arbitrarily. Then un-numbered vertices adjacent to the next lowest-numbered vertex are assigned numbers in the same way. This process continues until all the vertices in the current level are numbered: then the algorithm is applied to the next level in the same manner. In this way at most $|V(G)|$ candidate numberings for a graph G are produced, and one with the minimum bandwidth is selected as the output of the algorithm.

Although the Cuthill–McKee algorithm was the most widely used bandwidth reduction algorithm during the 1970s, it has a number of shortcomings. The first of these is the time consumed by an exhaustive search through the rooted level structures for ones having minimum width. A second shortcoming is that for each level structure of minimum width (and there can be many of these) the entire graph is numbered and the bandwidth of that numbering is computed. A third problem with the Cuthill–McKee algorithm is that the numbering generated has bandwidth exactly equal to the width of the level structure selected: the actual bandwidth of the graph might of course be less than this.

An algorithm described, implemented and tested by Gibbs, Poole, and Stockmeyer [49], [26], [51] in 1976, has largely removed these shortcomings: A starting vertex is selected after generating a relatively few rooted level structures, the graph is renumbered (and the corresponding bandwidth computed) only once, and a more general kind of level structure results in a numbering whose bandwidth is, in general, closer to the bandwidth of the graph inputted. The Gibbs–Poole–Stockmeyer algorithm has three phases:

(1) finding a pseudo-diameter,
(2) minimizing level width,
(3) numbering a level structure.

It was observed that level structures of small width are usually among those of greater depth. In an average sense, increasing the number of levels decreases the number of vertices in each level and reduces the width of the level structure as well. In this sense, it would be ideal to generate level structures which are rooted at endvertices of a diameter of the graph. Although it is possible to find a diameter in polynomial time, no very efficient algorithm for doing this is known. Accordingly, the GPS algorithm discovers a *pseudodiameter* (defined to be the distance between the vertices u and v generated by the corresponding algorithm) which is close to the real diameter on the average. For a large class of graphs, including trees, the pseudodiameter is the diameter.

The algorithm for finding the endvertices of a pseudodiameter is now described:

(1) select a vertex of minimum degree and call it v,
(2) generate L_v and let S_v be the deepest level of L_v,
(3) for each $u \in S_u$ and in order of increasing degree, generate L_u. As soon as an L_u having depth greater than L_v is found, replace v by u and return to step 2,
(4) if no L_u has depth greater than L_v, select the vertex $u \in S$ for which L_u has minimum width. The endvertices of the pseudodiameter are u and v.

The purpose of the next algorithm is to minimize level width. From $L_v = (L_1, L_2, \ldots, L_k)$ and $L_u = (M_1, M_2, \ldots, M_k)$, associate with each vertex w of G the ordered pair (i, j), called the *associated level pair*, where i is the index of the level in L_v that contains w, and $k + 1 - j$ is the index of the level in L_u that contains w. Thus the pair (i, j) is associated with a vertex w if and only if $w \in L_i \cap M_{k+1-j}$. Note that the pair $(1, 1)$ is associated with the vertex v, while the pair (k, k) is associated with u.

Assign the vertices of G to levels in a new level structure $L = (N_1, N_2, \ldots, N_k)$ as follows:

(1) If the associated level pair of a vertex w is of the form (i, i), then vertex

w is placed in N_i. The vertex w and all edges incident to w are removed from the graph. If $V(G)$ is empty, stop.

(2) The graph G now consists of a set of one or more disjoint connected components $C_1, C_2 \ldots, C_t$ ordered so that $|V(C_1)| \geq |V(C_2)| \geq \cdots \geq |V(C_t)|$.

(3) For each connected component (taken in the order $C_1, C_2, \ldots, C_t$) do the following:

(a) Compute the vector $(n_1, n_2, \ldots, n_k)$ where $n_i = |N_i|$.

(b) Compute the vectors $(h_1, h_2, \ldots, h_k)$ and $(l_1, l_2, \ldots, l_k)$, where $h_i = n_i +$ (the number of vertices which would be placed in N_i if the first element of the associated level pairs were used) and $l_i = n_i +$ (the number of vertices which would be placed in N_i if the second element of the associated level pairs were used).

(c) Find $h_0 = \max_i\{h_i : h_1 - n_i > 0\}$ and $l_0 = \max_i\{l_i : l_i - n_i > C\}$.

(i) If $h_0 < l_0$, place all the vertices of the connected component in the levels indicated by the first elements of the associated level pairs.

(ii) If $l_0 < h_0$, use the second elements of the level pairs to place the vertices in the levels.

(iii) If $h_0 = l_0$, then use the elements of the level pairs which arise from the rooted level structure of smaller width. If the widths are equal, use the first elements.

The algorithm terminates when each vertex of G has been assigned a level in the level structure L.

The third and last basic algorithm employed by Gibbs et al. uses the level structure L generated by the foregoing algorithm and, like the Cuthill—McKee algorithm, assigns consecutive positive integers to the vertices of G level by level.

(1) If u has smaller degree than v interchange the labels u and v and reverse the level numbers by setting N_i to N_{k+1-i}, $(i = 1, 2, \ldots, k)$.

(2) Define the numbering f by setting $f(v) = 1$, $N_0 = \emptyset$ and, for $i = 1, 2, \ldots, k$, carrying out the following steps:

(a) Find the $w \in N_{i-1}$ (if it exists) having the lowest f-number over all numbered vertices in N_{i-1} adjacent to un-numbered vertices in N_i. Number the vertices of N_i adjacent to w in order of increasing degree.

(b) Repeat step (a) until all vertices of N_i adjacent to vertices in N_{i-1} are numbered.

(c) Find the $w \in N_i$ (if it exists) having the lowest f-number over all numbered vertices—adjacent to at least one un-numbered vertex in N_i. Number the as yet unnumbered vertices of N_i adjacent to w in order of increasing degree.

(d) Repeat step (c) until all the vertices of N_i have been numbered; in the case of un-numbered vertices which are not adjacent to numbered ones, choose one of minimum degree.

Unlike the Cuthill–McKee numbering algorithm, however, the one above uses the general type of level structure. With such a structure, it is possible to implement several operations which serve to minimize level width.

The Gibbs–Poole–Stockmeyer algorithm has been implemented in FORTRAN [25] and an improved FORTRAN version by Lewis [69] is claimed to run slightly faster.

The worst-case complexity of the Gibbs–Poole–Stockmeyer algorithm is in the order of n^3, where $n = |V(G)|$, the bulk of the time being taken by the pseudodiameter subalgorithm. However, its average case complexity appears to be considerably better than this. Based on a set of gridded rectangles and cylinders, however, the algorithm would appear to be nearly linear in the sense that its average-time complexity on these graphs was approximately $O(n^{1.2})$. Such graphs were also used by George and Liu [45] to evaluate matrix partitioning algorithms.

Gridded rectangles and cylinders have a close resemblance to many engineering structures. In fact, a set of 30 test matrices based on such structures are provided by Everstine [36] and now comprise a standard benchmark for production algorithms. On these matrices, the Gibbs–Poole–Stockmeyer algorithm consistently out performed the Cuthill–McKee algorithm.

6. RELATED PROBLEMS

There are a number of problems related to the bandwidth problem for graphs or matrices, some of which are near relations while others are only distant cousins.

In dealing with bandwidth considerations for a nonsymmetric matrix, it is natural to ask for a permutation which, nevertheless, confines the nonzero entries of the matrix to a symmetrical band about the main diagonal. The corresponding problem of discovering the bandwidth of a directed graph is virtually the same as solving the bandwidth problem for its undirected version with loops and multiple edges removed. Having observed this, Eitner [33] has also shown, for arbitrary matrices M and N, that

$$B(M \cdot N) \leq B(M) + B(N)$$

and that

$$B(M + M^t) \leq B(M),$$

where $\cdot$ and $+$ are the usual matrix multiplication and addition operations.

Another version of the bandwidth problem for directed graphs involves the stipulation that only numberings f which satisfy the following condition are admitted into the minimization process

$$(1) \text{ if } (u, v) \in E(G), \text{ then } f(u) < f(v).$$

The resulting *directed bandwidth problem* appears to be intractable since a simple modification of the construction by Garey et al. [40] yields the following result.

Theorem 6.1. The directed bandwidth problem for trees having no vertices of indegree > 2 or outdegree > 1 is NP-complete.

The notion of *cross-bandwidth* $\widetilde{B}(G)$, due to Kahn and Kleitman [60] is very similar to bandwidth: for a graph G and a numbering f of G find the minimum value of

$$\max\{|f(u) - f(v)| : (u, v) \in E(G)\}$$

subject to the restriction that for each edge (u, v) of G,

$$1 \leq f(u) \leq \lfloor n/2 \rfloor \text{ and } \lfloor n/2 \rfloor + 1 \leq f(v) \leq n.$$

Kahn and Kleitman have displayed a relationship between cross-bandwidth of a graph and the connectivity of certain subcontractions. As well, they have found a useful lower bound for the complement of P_n^k.

Theorem 6.2. If there is a contraction c of G onto a k-connected graph G' for which $|c^{-1}(v)| \leq 2$ for all $v \in V(G')$, then $\widetilde{B}(G) \geq k$.

Theorem 6.3. $\widetilde{B}(P_n^k) \geq n - k - 2$.

Since $B(G) \geq \widetilde{B}(G)$ for any graph G, Theorem 6.2 also holds for the (ordinary) bandwidth of the complement of P_n^k answering a question posed by Chinn et al. [17] concerning the bandwidth of a certain bipartite graph.

Instead of taking the maximum difference $|f(u) - f(v)|$, one may take the sum of such differences over all edges in the graph. The attempt to minimize the resulting number is called the *bandwidth sum* problem. Little is known about this problem beyond the work of Harper [56], who solved the problem for the n-cube. Denote the minimum bandwidth sum for a graph G by $B_{\text{sum}}(G)$.

Theorem 6.4. $B_{\text{sum}}(Q_n) = 2^{n-1}(2^n - 1)$.

Harper's work takes its place in the setting of discrete isoperimetric problems [88] in which one asks for the minimum "boundary" of an m-subset of a discrete set of points.

As shown in Garey and Johnson [41], the bandwidth sum problem is just as hard, computationally speaking, as the bandwidth problem.

Theorem 6.5. The bandwidth sum problem is NP-complete.

If one defines the *directed bandwidth sum problem* as a combination of condition (1) above with a sum of differences rather than the maximum difference, the resulting problem appears to be slightly easier than the directed bandwidth problem: although still NP-complete, as shown by Lawler [65], Adolfson and Hu [1] have found a polynomial time algorithm for the problem when restricted to oriented trees.

There is another sum-minimization problem somewhat related to the bandwidth sum problem. Define the *profile* [48] of a symmetric $n \times n$ matrix $A = (a_{ij})$ as the minimum value of the sum

$$\sum_{u \in V(G)}^{n} (i - \min\{j : a_{ij} \neq 0\}),$$

taken over all symmetric permutations of A, it being assumed that $a_{ii} = 1$, $(i = 1, 2, \ldots, n)$. Profile may be redefined as a graph invariant $P(G)$ by finding a numbering f of G which minimizes the sum

$$\sum_{u \in V(G)} (f(u) - \min\{f(v) : v \cong u\}),$$

where $\cong$ means "is adjacent or equal to." It is easily seen that $P(G) \leq \frac{1}{2}B_{\text{sum}}(G)$.

Since profile reduction is relevant to the speedup of matrix computations—much as bandwidth reduction is—a number of profile reduction algorithms have been developed. The first widely used algorithm for profile reduction was the reverse Cuthill–McKee algorithm based on one developed by George [44]. Other profile reduction algorithms have been developed by King [61], Gibbs [46], and Snay [84], the latter often producing the best profile. Based on average profile reduction, however, these three algorithms are comparable improvements on the reverse Cuthill–McKee algorithm.

If, in the course of finding a minimum bandwidth numbering of a graph G, one is allowed to subdivide edges arbitrarily often, the resulting *topological bandwidth problem* does not appear to be as hard, computationally, as the bandwidth problem. Although it is to be suspected that this problem is NP-complete, Dewdney [30] has found a polynomial-time algorithm for computing the elastic bandwidth of an arbitrary tree. The fact that topological bandwidth for trees is in P means that topological properties alone do not account for the NP-completeness of bandwidth for trees generally.

Finally, a problem which is reminiscent of the bandwidth problem but, nevertheless, rather different, has been studied extensively by Bloom and Golomb [10]: given a graph G with $|V(G)| = m$ and $|E(G)| = n$, assign m of the integers $\{0, 1, \ldots, n\}$ to the vertices of G so that each of the integers 0, $1, \ldots, n$ appears exactly once as a difference $|f(u) - f(v)|$, where f is the assignment function. A graph for which such an assignment is possible is called *graceful* and the discovery of graceful graphs has applications to coding theory as well as other topics. The paper by Bloom and Golomb cited above also contains a useful overview of the numbering problems in general, including both the bandwidth problem and some of its variants mentioned above.

ACKNOWLEDGMENTS

The authors wish to thank Ron Graham for encouragement in enlarging and enhancing the survey [21] previously available only as a departmental report. They also wish to thank the *Graph Theory Newsletter* for its help in publicizing the search for little-known results. Thanks are finally due to the referees who greatly aided in introducing coherence, both in style and content, into the contributions of four separate authors.

References

[1] D. Adolfson and T. Ch Hu, Optimal linear ordering. *SIAM J. Appl. Math.* 25 (1973) 403–423.

[2] G. Akhras and G. Dhatt, An automatic node-relabelling scheme for minimizing a matrix or network bandwidth. *Internat. J. Numer. Methods Engrg.* 10 (1976) 787–797.

[3] F. A. Akyuz, Reply by author to J. Barlow and C. G. Marples. *J. Amer. Inst. Aeronaut. Astronaut.* 7 (1969) 381–382.

[4] F. A. Akyuz and S. Utku, An automatic node-relabelling scheme for bandwidth minimization of stiffness matrices. *J. Amer. Inst. Aeronaut. Astronaut.* 6 (1968) 728–730.

[5] G. G. Alway and D. W. Martin, An algorithm for reducing the bandwidth of a matrix of symmetrical configuration. *Comput. J.* 8 (1965) 264–272.

[6] I. Arany, W. F. Smyth and L. Szôda, An improved method for reducing the bandwidth of sparse symmetric matrices. In *Information Processing* 71 (*Proc. IFIP Congress 71*), North Holland, Amsterdam (1972) 1246–1250.

[7] I. Arany and L. Szôda, Ritka szimmetrikus matrixok savézélesség redukciója. *Informác. Elektron.* 4 (1973) 273–282.
[8] J. Barlow and C. G. Marples, Comment on an automatic node-relabelling scheme for bandwidth minimization of stiffness matrices. *J. Amer. Inst. Aeronaut. Astronaut.* 7 (1969) 380–381.
[9] G. S. Bloom, Numbered undirected graphs and their uses: A survey of a unifying scientific and engineering concept and its use in developing a theory of nonredundant hemometric sets relating to some ambiguities in X-ray diffraction analysis. Ph.D. dissertation, University of Southern California, Los Angeles (1975).
[10] G. S. Bloom and S. W. Golomb, Applications of numbered undirected graphs. *Proc. IEEE* 65 (1977) 562–570.
[11] J. H. Bolstad, G. K. Leaf, A. J. Lindeman and G. H. Kaper, An empirical investigation of reordering and data management for finite element systems of equations. Argonne Nat. Lab. Report #8056, Argonne, IL (1973).
[12] K. Y. Cheng, Minimizing the bandwidth of sparse symmetric matrices. *Computing* 2 (1973) 103–110.
[13] K. Y. Cheng, Note on minimizing the bandwidth of sparse symmetric matrices. *Computing* 2 (1973) 27–30.
[14] P. Z. Chinn, The bandwidth of the corona and composition of two graphs. MS, Department of Mathematics, Humboldt State University, Arcata, California (1980).
[15] P. Z. Chinn, The bandwidth and other invariants of the Möbius Ladder. Technical Report, Department of Mathematics, Humboldt State University (1980).
[16] P. Z. Chinn, Critical bandwidth - 3 trees. Technical Report, Humboldt State University (1980).
[17] P. Z. Chinn, F. R. K. Chung, P. Erdös and R. L. Graham, On the bandwidth of a graph and its complement. *The Theory and Applications of Graphs*, G. Chartrand, Ed. Wiley, New York (1981) 243–253.
[18] V. Chvátal, A remark on a problem of Harary. *Czechoslovak Math. J.* 20 (1970) 109–111.
[19] J. Chvátalová, On the bandwidth problem for graphs. Ph.D. dissertation, Department of Combinatorics and Optimization, University of Waterloo (1980).
[20] J. Chvátalová, Optimal labelling of a product of two paths. *Discrete Math.* 11 (1975) 249–253.
[21] 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, UWO, London, Ontario (1975).

[22] J. Chvátalová and J. Opatrný, Two results on the bandwidth of graphs. *10th S. E. Conference on Combinatorics, Graph Theory and Computing* 1. Utilitas Mathematica, Winnipeg (1979) 263–274.

[23] J. Chvátalová and J. Opatrný, The bandwidth problem and operations on graphs. To appear.

[24] R. J. Collins, Bandwidth reduction by automatic renumbering. *Internat. J. Numer. Methods Engr.* 6 (1973) 345–356.

[25] W. L. Cook, Automated input preparation for NASTRAN. Goddard Space Flight Center Report X-321-69-237 (1969).

[26] H. L. Crane Jr., N. E. Gibbs, W. G. Poole Jr. and P. K. Stockmeyer, Algorithm 508 matrix bandwidth and profile reduction. *ACM Trans. Math. Software* 2 (1976) 375–377.

[27] J. G. Crose, Bandwidth minimization of stiffness matrices. *J. Dech. Div. ASCE* 97 (1971) 163–167.

[28] E. Cuthill, Several strategies for reducing the bandwidth of matrices. In *Sparse Matrices and their Applications*, D. Rose and R. Willoughby, Eds. Plenum, New York (1972) 157–166.

[29] E. Cuthill, J. McKee, Reducing the bandwidth of sparse symmetric matrices. *Proc. 24th Nat. Conf. ACM* (1969) 157–172.

[30] A. K. Dewdney, The bandwidth problem for graphs: some recent results. In *7th S. E. Conference on Combinatorics, Graph Theory and Computing*. Utilitas Mathematica, Winnipeg (1976) 273–288.

[31] A. K. Dewdney, Tree topology and the NP-completeness of tree bandwidth. Research Report #60, Department of Computer Science, UWO, London, Ontario (1980).

[32] P. G. Eitner, The bandwidth of the complete multipartite graph. Presented at the Toledo Symposium on Applications of Graph Theory (1979).

[33] P. G. Eitner, The bandwidth problem for directed graphs. *Abst. Amer. Math. Soc.* 1 (1980) 408.

[34] R. Epp and A. G. Fowler, Efficient code for steady-state flows in networks. *J. Hydraulics Div. Proc. ASCE* 96 (1970) 43–56.

[35] G. C. Everstine, The BANDIT computer program for the reduction of matrix bandwidth for NASTRAN. NSRDC Report 3827, Bethesda, MD (1972).

[36] G. C. Everstine, A comparison of three resequencing algorithms for the reduction of matrix profile and wave-front. *Internat. J. Numer. Methods in Engr.* 14 (1979) 837–863.

[37] G. Everstine, Recent improvements to BANDIT. In *NASTRAN's User Experience* NASA TM X-3278 (1975) 511–521.

[38] C. H. FitzGerald, Optimal indexing of the vertices of a graph. *Math. Comput.* 28 (1974) 825–831.

[39] D. R. Fulkerson and O. A. Gross, Incidence matrices and interval graphs. *Pacific J. Math.* 15 (1965) 835–855.

[40] 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.

[41] M. R. Garey and D. S. Johnson, *Computers and Intractibility: A Guide to the Theory of NP-Completeness.* W. H. Freeman, San Franciso (1979).

[42] M. R. Garey, D. S. Johnson,and R. L. Stockmeyer, Some simplified NP-complete graph problems. *Theoretical Comp. Sci.* 1(1976) 237–267.

[43] M. R. Garey, D. S. Johnson,and R. L. Stockmeyer, Some simplified NP-complete problems. *Proc. 6th Annual ACM Symp. On Theory of Computing* (1974) 47–63.

[44] J. A. George, Computer Implementation of the finite element method. Ph.D. dissertation, Tech. Report STAN-CS-71-208, Computer Science Department, Stanford University, Stanford, CA (1971).

[45] J. A. George and J. W. H. Liu, Algorithms for matrix partitioning and the numerical solution of finite element systems. *SIAM J. Numer. Anal.* 15 (1978) 297–327.

[46] N. E. Gibbs, Algorithm 509 - A hybrid profile reduction algorithm. *ACM Trans. Math. Software* 2 (1976) 378–387.

[47] N. E. Gibbs, The bandwidth of graphs. Ph.D. dissertation, Purdue University, Lafayette, IN (1969).

[48] N. E. Gibbs and W. G. Poole Jr., Tridiagonalization by permutations. *Comm. ACM* 20 (1974) 20–24.

[49] 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) 235–251.

[50] N. E. Gibbs, W. G. Poole Jr.,and P. K. Stockmeyer, An algorithm for reducing the bandwidth and profile of a sparse matrix. *ICASE* (1974).

[51] N. E. Gibbs, W. G. Poole, and P. K. Stockmeyer, A comparison of several bandwidth and profile reduction algorithms. *ACM Trans. Math. Software* 2 (1976) 322–330.

[52] S. W. Golomb, How to number a graph. *Graph Theory and Computing*, R. C. Read, Ed. Academic, New York (1972) 23–37.

[53] R. L. Graham, On primitive graphs and optimal vertex assignments. *Ann. N.Y. Acad. Sci.* 175 (1970) 170–186.

[54] H. R. Grooms, Algorithm for matrix bandwidth reduction. *J. Struct. Div. ASCE* 98 (1972) 203–214.

[55] F. Harary, *Graph Theory.* Addison-Wesley, Reading, MA (1969).

[56] F. Harary, Problem 16, In *Theory of graphs and its applications.* M. Fiedler, Ed., Czechoslovak Academy of Science, Prague (1967).

[57] L. H. Harper, Optimal assignment of numbers to vertices. *J. SIAM* 12 (1964) 131–135.

[58] L. H. Harper, Optimal numberings and isoperimetric problems on graphs. *J. Combinat. Theory* 1 (1966) 385–393.

[59] S. T. Hedetniemi, Some unsolved problems involving combinatorial algorithms. MS, Department of Computer and Information Science, University of Oregon (1980).

[60] J. Kahn and D. J. Kleitman, On cross-bandwidth. *Discrete Math.* 33 (1981) 323–325.

[61] I. P. King, An automatic reordering scheme for simultaneous equations derived from network systems. *Internat. J. Numer. Methods Engr.* 2 (1970) 523–533.

[62] I. Konishi, N. Shiraish, and T. Taniguch, Reducing bandwidth of structural stiffness matrices. *J. Structural Mech.* 4(1976) 197–226.

[63] R. R. Korfhage, Numberings of the vertices of graphs. Computer Science Department Technical Report 5, Purdue University, Lafayette, IN (1966).

[64] R. R. Korfhage and N. E. Gibbs, The bandwidth of cubic graphs. Res. Memo 70-1, School of Industrial Engineering, Purdue University, Lafayette, IN (1970).

[65] E. L. Lawler, Sequencing jobs to minimize total weighted completion time subject to precedence constraints. Submitted.

[66] P. F. Lemieux and P. E. Brunelle, Relabelling algorithm to obtain a band-matrix from a sparse one. Technical Report PFL-3-72, Department of Civil Engineering, University of Sherbrooke, Sherbrooke, Quebec (1972).

[67] R. Levy, Resequencing of the structural stiffness matrix to improve computational efficiency. *Jet Propul. Lab. Quart. Tech. Rev.* 1 (1971) 61–70.

[68] R. Levy, Structural stiffness matrix wavefront resequencing program (WAVEFRONT). JPL Technical Report 32-1526, Vol. XIV (1972) 50–55.

[69] J. G. Lewis, Implementation of the Gibbs-Poole-Stockmeyer algorithm and the Gibbs-King algorithm (algorithms 508 and 509). *ACM Trans. Math. Software*. To appear.

[70] C. C. Lin, Bandwidth reduction for simultaneous equations in matrix analysis. Babcock and Wilcox Co., Report TP-535, Lynchburg, VA (1974).

[71] J. W. Liu, On reducing the profile of sparse symmetric matrices. Ph.D. dissertation, Department of Computer Science, Faculty of Mathematics, University of Waterloo, Waterloo, Ontario (1975).

[72] J. W. H. Liu and A. H. Sherman, Comparative analysis of the Cuthill-McKee and the reverse Cuthill-McKee ordering algorithms for sparse matrices. *SIAM J. Num. Anal.* 12 (1975) 198–213.

[73] R. K. Livesley, The analysis of large structural systems. *Comput. J.* 3 (1960) 34–39.

[74] M. F. Nelson and J. Hidalgo, Discussion of algorithm for matrix bandwidth reduction by H. R. Grooms. *J. Structural Div. ASCE* 98 (1972) 2820–2821.

[75] C. H. Papadimitriou, The NP-completeness of the bandwidth minimization problem. *Computing* 16 (1976) 263–270.

[76] E. Roberts, Relabelling of finite-element meshes using a random process. NASA Technical Memo. NASA TM X-2660 (1972).

[77] J. S. N. Rodrigues, Node numbering optimization in structural analysis. *J. Struct. Div. ASCE* 101 (1975) 361–376.

[78] A. Rosa, On certain valuations of the vertices of a graph. *Theory of Graphs, International Symposium Rome*. Gordon and Breach (1967) 349–355.

[79] D. J. Rose, A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations. *Graph Theory and Computing*, R. C. Read, Ed. Academic, New York (1972) 23–37.

[80] R. Rosen, Matrix bandwidth minimization. *Proc. 23rd Natl. Conf. ACM*. Brandon Systems, Princeton, NJ (1968) 585–595.

[81] J. B. Saxe, Dynamic programming algorithms for recognizing small-bandwidth graphs in polynomial time. *SIAM J. Albegraic Discrete Methods* I (1980) 363–369.

[82] D. A. Sheppard, The factorial representation of balanced, labelled graphs. *Discrete Math.* 15 (1976) 379–388.

[83] W. F. Smith and I. Arany, Another algorithm for reducing bandwidth and profile of a sparse matrix. *Proc. AFIPS 1976 NCC*, AFIPS Press, Montvale, New Jersey (1976) 987–994.

[84] R. A. Snay, Reducing the profile of sparse symmetric matrices. *Bul. Géodésique* 50 (1976) 341–352.

[85] M. M. Syslo and J. Zak, The bandwidth problem for ordered caterpillars. Computer Science Department Report CS-80-065. Washington State University (1980).

[86] R. P. Tewarson, Row column permutation of sparse matrices. *Comput. J.* 10 (1967) 300–305.

[87] D. L. Wang and P. Wang, On the bandwidth ordering and isoperimetric problems on graphs that are products of paths or even cycles. To appear.

[88] D. L. Wang and P. Wang, Discrete isoperimetric problems. *SIAM J. Appl. Math.* 32 (1977) 860–870.

[89] P. T. R. Wang, Bandwidth minimization, reducibility, decomposition, and triangularization of sparse matrices. Ph.D. dissertation, Department of Computer and Information Science, Ohio State University, Columbus, OH (1973).

[90] P. T. R. Wang, D. S. Kerr, and L. J. White, Bandwidth reduction of sparse matrices by row and column permutations. *SIAM Rev.* 17 (1975) 391.

[91] R. A. Willoughby, Ed., IBM Sparse matrix proceedings. IBM Report RAI, No. 11707 (1969).
[92] J. W. Zak, The bandwidth minimization problem for trees. Master's thesis, Institute of Computer Science, University of Wroclaw (1980).
[93] O. C. Zienkiewicz, *The Finite Element Method in Engineering Science*. McGraw-Hill, London (1971).