On Approximation Intractability of the Bandwidth Problem

Gunter Blache* Marek Karpinski† Jürgen Wirtgen‡

Abstract

The *bandwidth problem* is the problem of enumerating the vertices of a given graph G such that the maximum difference between the numbers of adjacent vertices is *minimal*. The problem has a long history and a number of applications. There was not much known though on approximation hardness of this problem, till recently. Karpinski and Wirtgen [KW 97] showed that there are no polynomial time approximation algorithms with an absolute error guarantee of $n^{1-\epsilon}$ for any $\epsilon > 0$ unless $P = NP$.

In this paper we show, that there is no $PTAS$ for the bandwidth problem unless $P = NP$, even for trees. More precisely we show that there are no polynomial time approximation algorithms for general graphs with an approximation ratio better than 1.5, and for the trees with an approximation ratio better than $4/3 \approx 1.332$.

*Dept. of Computer Science, University of Bonn, 53117 Bonn. Email: blache@cs.bonn.edu.

†Dept. of Computer Science, University of Bonn, 53117 Bonn, and International Computer Science Institute, Berkeley, California. Research partially supported by DFG Grant KA 673/4-1, by the ESPRIT BR Grants 7097 and EC-US 030, and by the Max-Planck Research Prize. Email: marek@cs.bonn.edu.

‡Dept. of Computer Science, University of Bonn, 53117 Bonn. Research partially supported by the ESPRIT BR Grants 7097 and EC-US 030. Email: wirtgen@cs.bonn.edu

1 Introduction

The bandwidth problem on graphs has a very long and interesting history cf. [CCDG 82].

Formally the bandwidth minimization problem is defined as follows. Let $G = (V, E)$ be a simple graph on n vertices. A numbering (or layout) of G is a one-to-one mapping $f : V \to \{1, ..., n\}$. The bandwidth $B(f, G)$ of this numbering is defined by

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

the greatest distance between adjacent vertices in G corresponding to f. The bandwidth $B(G)$ is then

$$B(G) = \min_{f \text{ is a numbering of } G} \{B(f, G)\}$$

Clearly the bandwidth of G is the greatest bandwidth of its components. Therefore, we assume without loss of generality that the input graph is connected.

The problem of constructing the bandwidth of a graph is NP-hard [Pa 76], even for trees with maximum degree 3 [GGJK 78]. There are only few cases for which we can construct the optimal layout in polynomial time [GGJK 78], [Sa 80], [Ch 88], [Sm 95].

To date there was not much known about approximating the bandwidth. Recently Feige [Fe 97] constructed an approximation algorithm constructing a layout within a polylogarithmic factor of the optimum. The algorithm (cf. [Fe 97]) is based on volume respecting embeddings, which are natural extensions of small distortion embeddings of Linial, London and Rabinovich [LLR 95].

Also for special graph classes, like caterpillars [HMM 91] found a polynomial time $\log n$-approximation algorithm. A caterpillar is a special kind of a tree consisting of a simple chain, the body, with an arbitrary number of simple chains, the hairs, attached to the body by coalescing an endpoint of the added chain with a vertex of the body. For this special class of trees the bandwidth problem was also shown to be NP-hard [Mo 86]. Karpinski, Wirtgen and Zelikovsky [KWZ 97] constructed a 3-approximation algorithm for δ-dense graphs. A graph G is δ-dense, if the minimum degree $\delta(G)$ is at least δn. We call it everywhere dense, if it is δ-dense for some $\delta > 0$.

The design of approximation algorithms for NP-hard optimization problems became an important field of research in the last decade. In the best of situations we are able to find approximation algorithms which work in polynomial time and approximate optimal solutions within an arbitrary given constant. Such meta-algorithms are called polynomial time approximation schemes ($PTAS$s), cf.eg., [Ho 97]. For the dense instances of MAX-SNP problems [PY 91], the existence of $PTAS$ has been proven by Arora, Karger and Karpinski [AKK 95].

Most of the above algorithms have one thing in common, namely their running times are bounded by $n^{O(f(1/\epsilon))}$ where the approximation ratio is $r = 1 + \epsilon$. The algorithms are becoming more practical if their running times are functions of a kind $g(1/\epsilon)n^{O(1)}$. These algorithms are called efficient polynomial approximation schemes ($EPTAS$s). There has been recently some progress in this direction. Fernandez de la Vega [Fe 96] designed a randomized algorithm for the MAX-CUT

problem, which runs in $2^{(1/\epsilon)^{O(1)}} n^{O(1)}$ time (removing dependence on ϵ in the exponent of n). Frieze and Kannan [FK 96] obtained similar bounds for dense instances of some MAX-SNP-hard problems using an algorithmic version of Szemeredi's regularity lemma. Another improvement was given in Goldreich, Goldwasser and Ron [GGR 96], and in Frieze and Kannan [FK 97].

In [KW 97] Karpinski and Wirtgen relate the parameterized complexity theory [DF 92] to the notion of $EPTAS$s to show, that there are no $EPTAS$s for the bandwidth problem, under certain natural conditions.

Another open problem was the question whether there exist absolute approximation algorithms for the bandwidth problem. We say, a solution S is a absolute r-approximation to the optimum OPT, if $S \leq OPT + r$ (in the case of minimization problems). For some graph parameters like the treewidth, or vertex separator, it is known, that there are no absolute approximations [BGHK 95], [BJ 92]. In [KW 97] the bandwidth was related to the treewidth. It was shown, that there are no absolute $n^{1-\epsilon}$-approximations for the bandwidth problem, unless $P = NP$.

This paper is organized as follows. In Section 2 we prove, that it is NP-hard to find a $PTAS$ for trees. Section 2.1 shows that we get better hardness bounds for general graphs.

2 NP-Hardness of Bandwidth Approximation

We show a reduction from the $3SAT$ problem to the bandwidth problem restricted on trees. For simplicity, we can assume that each clause contains exactly 3 literals (cf. [Pa 76]). Let be $\phi(x) = \bigwedge_{i=1}^{m} c_i$ an instance of $3SAT$ on n variables. We will construct in polynomial time a tree $T = T_\phi$, such that $\phi \in 3SAT$ iff $B(T) \leq b$, b will be specified later. We use parameters $p, s, o, o_{1..3}^{S,U,V}$ which will be chosen suitable in $n^{O(1)}$. For simplicity reasons, we will define $d = 3 + n + 2m$. Later we will deduce all parameters from s and o. For the proof of NP-hardness of the decision problem, the two may be chosen freely from $n^{O(1)}$, as long as they satisfy

$$o + s \quad \geq \quad 3n + 6m \qquad (1)$$

For the proof of the approximation hardness, we will set them explicitly.

The construction of T starts with a center vertex c. There are two subgraphs S and U, one subgraph L_y for each literal y and for each clause c_j four subgraphs graphs $C_j^1, \ldots, C_j^4$, which are all attached to c. At the outer ends of S and U are another two subgraphs V_S and V_U attached (see Figure 1).

The subgraphs L_{x_i} and $L_{\overline{x}_i}$ consist of a line of $m + n$ components. Every component has a line of $2d$ nodes and a star of size s attached to the node with number $d + 3 + i$. The $d + 1$-th node of the $m + i$-th component has a star of size s. Moreover, the $d + 1$-th node of every j-th component has a star of size s attached through an intermediate node iff x_i (or $\overline{x}_i$) satisfies the clause c_j (Figure 2).

The four lines for every clause consist of $m + n$ components as well. Every component has a star attached to every node with number $d + 3 + n + j$. The lines C_j^1 and C_j^2 receive one star at the $d + 1$-th node of the j-th component (Figure 3). The remaining lines C_j^3 and C_j^4 (Figure 4) are kept for consistency.

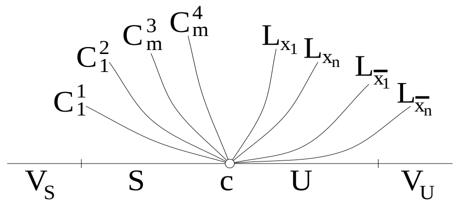

Figure 1: Overview of the structure of T

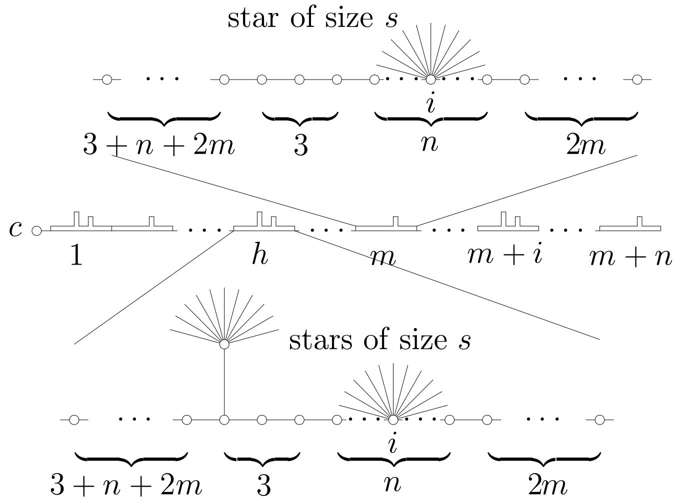

Figure 2: Line of x_i, occurring in clause 1 and h

The component of the subgraphs S,U and V is a graph with a backbone of length $2d$. The first d nodes have stars of size p (these are called *barriers*), the following 3 nodes have a star of size o_1,o_2 and o_3 respectively (called *pockets*). The remaining n nodes have a star of size o, the last $2m$ nodes a star of size $o-1$ attached (these are referred to as *buckets*). When built into the subgraphs, the o_i are replaced by $o_i^{S,U,V}$ respectively.

The two subgraphs S and U are constructed from m components and are joined directly with the center node c (Figure 5). The subgraph V is made of n components plus an additional barrier at the end (Figure 6). Two copies are built into the graph: one at the end of S (subsequently called V_S) and U (called V_U).

We define b to be

$$b = n + 2m + o + s + 1 \tag{2}$$

The barriers should have space for exactly $n+2m$ nodes, so that $n+2m$ lines may be layed through them. The three pockets of S, the first two of U and the first of V_S and V_U will need extra space for s nodes. It is easy to see, that we can choose our parameters such that in each of this parts the bandwidth will be b:

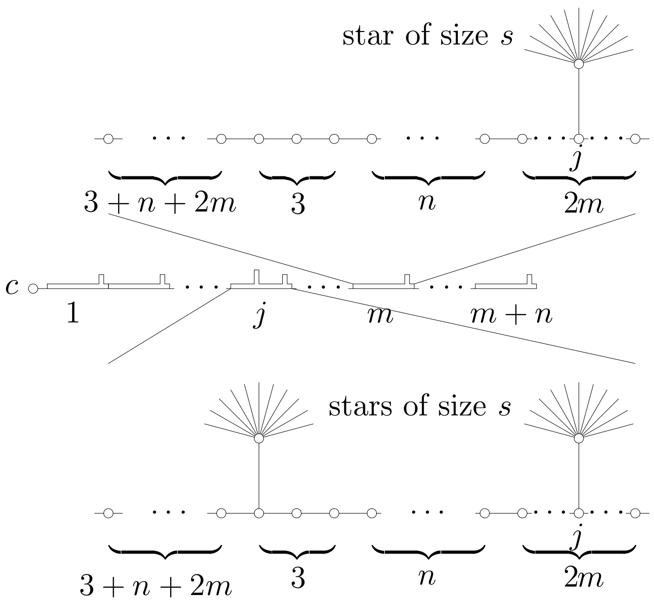

Figure 3: Line C_j^1 and C_j^2

$$p = o + s$$

$$\begin{array}{llllll} o_1^S & = o-2 & o_2^S & = o-1 & o_3^S & = o \\ o_1^U & = o-2 & o_2^U & = o & o_3^U & = o+s \\ o_1^V & = o-1 & o_2^V & = o+s & o_3^V & = o+s \end{array}$$

This construction is polynomially bounded by n. There are one central node, $4d(n+m)+2d$ nodes of the backbone, $2(n+m)$ barriers of dp nodes, $m(o_1^S+o_2^S+o_3^S)$ nodes in the pockets of S, $m(o_1^U+o_2^U+o_3^U)$ in those of U and $n(o_1^V+o_2^V+o_3^V)$ in each of V_S and V_U. $2(m+n)n$ stars of size o and $2(m+n)2m$ stars of size $o-1$ are in the buckets of S,U,V_S and V_U and $d(b-1)$ in the ending barriers. The $2n$ lines for the literals each have $2d(m+n)$ nodes in the line, a star of size s connected through an intermediate node and $(m+n)s$ in the stars for the buckets. Together they contain $3m$ stars of size s (since it is $3SAT$) plus the additional intermediate node. The $4m$ lines for the clauses each consist of $2d(m+n)$ nodes and $m+n$ stars of size s plus the additional node, $2m$ of them have an additional star of size s attached to an intermediate node. Adding these together and using the above equations, it turns out that the tree has $1+(4d(m+n)+2d)b$ nodes. Since the diameter is $4d(m+n)+2d$, b is a strict lower bound for the bandwidth of T.

Lemma 2.1 *For every $3SAT$-formula ϕ, the tree T_ϕ has a minimum layout f with $B(f,T_\phi) \leq b$ iff ϕ is satisfiable and a minimum layout g with $B(g,T_\phi) \geq b \cdot min\{\frac{3}{2}, 1+\lceil\frac{s}{3b}\rceil\}$, iff ϕ is not satisfiable.*

Proof: Be ϕ satisfiable. Then there is an assignment a, such that at most two literals of every clause are not satisfied. The layout is given as follows: S, V_S and

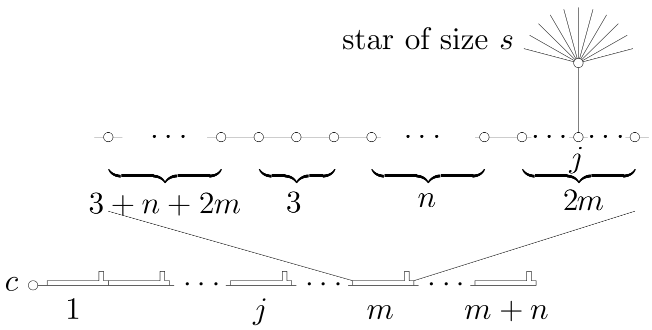

Figure 4: Line C_j^3 and C_j^4

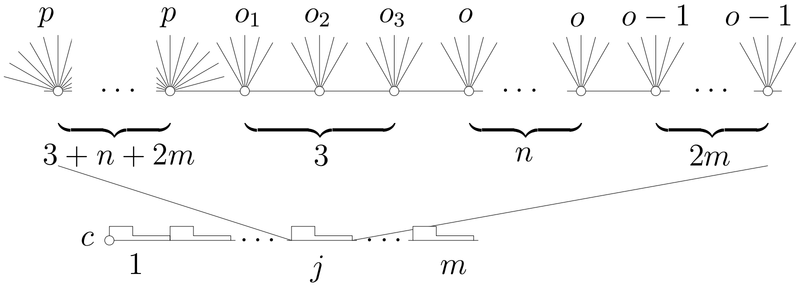

Figure 5: structure of S and U

the lines of the satisfied literals are folded to the left and U, V_U and the lines of the unsatisfied literals are folded to the right of c. C_1^j and C_2^j are folded to the left, if two literals in c_j are not satisfied, one to the left and one to the right, if one literal is not satisfied and both to the right, if all are satisfied. The other two are folded to the opposite side. This layout has bandwidth b, because every pocket in S has exactly 3 stars of size s, every pocket in U has exactly 2 stars of size s and every pocket in both V_S and V_U has exactly one star of size s. And because $n+2m$ lines are folded to the same side, the buckets have $n+2m$ stars of size s. The values of $o, o_{1..3}^{S,U,V}$ were chosen, so that all these stars have enough room.

No suppose ϕ is not satisfiable. Then there are four possibilities to layout the nodes:

1. Both S and U (and with them V_S and V_U) are folded to the same side. Then $2p+2$ is a lower bound for the bandwidth. To distinguish this case from the satisfied one, we required (1). Thus

$$
\begin{aligned}
s+o &\geq 3n+6m \\
4s+4o+4 &> 3(n+2m+o+s+1) \\
4p+4 &> 3b
\end{aligned}
$$

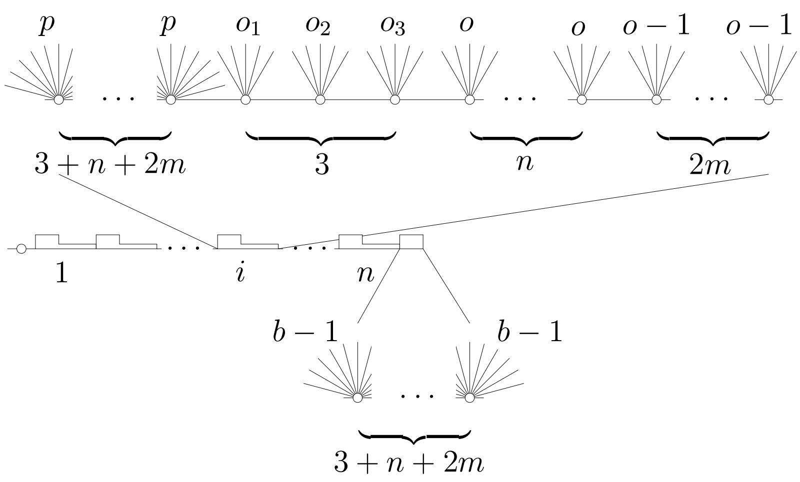

Figure 6: structure of V_S and V_U

$$2p + 2 \quad > \quad \frac{3}{2}b$$

2. The assignment on which the layout is based, is not valid, i.e. $\exists i : x_i = \overline{x}_i$. Then two stars of size s will need to be squeezed into the i-th pocket (either in V_S or V_U). The best way to do this is to spread the nodes of the star in the three neighboring sections. So we have

$$B(T) \geq b + \left\lceil \frac{s}{3} \right\rceil \tag{3}$$

3. The layout is based on a valid assignment s, but at least one clause is not satisfied : $\exists j : c_j(a) = 0$. The pockets of the unsatisfied clause have therefore three stars of size s. But because there is only space for two stars, the third one is spread over the three neighboring sections. So we have

$$B(T) \geq b + \left\lceil \frac{s}{3} \right\rceil \tag{4}$$

4. S and U are folded to different sides (wlog S to the left and U to the right) and no pocket has more stars than it should, but the lines are stretched or squeezed to achieve this. E.g. c_j is not satisfied, but the line of one literal is stretched to place the superfluous star into the pocket for c_{j+1}, where only one unsatisfied literal is placed. The line C_1^{j+1} for the clause is folded to the other side and squeezed, so that the star can be placed into the pocket for clause c_j in S.

   There are two possibilities to stretch a line: either it is stretched less than d positions, then the star for the bucket will be placed in the barrier and increase the bandwidth by one third of its size:

$$B(T) \geq b + \left\lceil \frac{s}{3} \right\rceil \tag{5}$$

Or the line is stretched so much that the star i is placed in the next pocket/bucket $i+1$. The previous star $i-1$ is still in the bucket $i-1$, and so the line with $2d$ nodes has to be stretched over $3d$ positions, and therefore

$$B(T) \geq \frac{3}{2}b \tag{6}$$

If a line is squeezed, one star for a bucket is placed in an already occupied area and so increases the bandwidth by a third of its size.

So in any case, the bandwidth of the layout of a tree made from an unsatisfied formula is either $\frac{3}{2}b$ or $b + \lceil \frac{s}{3} \rceil$. We will now see how to exploit this to prove the approximation hardness of the bandwidth problem on trees.

To show the approximation hardness, we have to assign s and o to suitable values such that the gap between the bandwidth for satisfied and unsatisfied formulas to be a constant multiple of the bandwidth.

$$\begin{aligned} \phi \in 3SAT &\Rightarrow B(T) \leq b \\ \phi \notin 3SAT &\Rightarrow B(T) \geq cb \end{aligned}$$

For the unsatisfied case, we have

$$B(T) \geq \min\{2p+2, \frac{3}{2}b, b + \left\lceil \frac{s}{3} \right\rceil\} \tag{7}$$

Choose $o = 3(n+2m+1)$, $s = 3l(n+2m+1)$, $l \geq 1$.

$$\begin{aligned} B(T) \geq\ & \min\{6(l+1)(n+2m+1)+2, \\ & \frac{3}{2}(n+2m+3(l+1)(n+2m+1)+1), \\ & n+2m+3(l+1)(n+2m+1)+1+l(n+2m+1)\} \\ \geq\ & \min\{6(l+1)(n+2m+1)+2, \frac{3}{2}(3l+2)(n+2m+1), \\ & 4(l+1)(n+2m+1)\} \\ \geq\ & 4(l+1)(n+2m+1) \end{aligned} \tag{8}$$

So we have for our constant c :

$$\begin{aligned} c \cdot b &= 4(l+1)(n+2m+1) \\ c \cdot (n+2m+o+s+1) &= 4(l+1)(n+2m+1) \\ c \cdot (n+2m+3(l+1)(n+2m+1)+1) &= 4(l+1)(n+2m+1) \\ c \cdot (3l+4)(n+2m+1) &= 4(l+1)(n+2m+1) \\ c &= \frac{4l+4}{3l+4} \\ c &> \frac{4}{3} - \epsilon \end{aligned} \tag{9}$$

Thereby we have proved that approximating the bandwidth problem on trees is NP-hard for a factor of $\frac{4}{3} - \epsilon$.

Theorem 2.2 *There is no PTAS for the bandwidth problem on trees, unless $P = NP$. In particular for any $\epsilon > 0$, there is no polynomial time approximation algorithm with a approximation ratio $4/3 - \epsilon \approx 1.332 - \epsilon$, unless $P = NP$.*

2.1 Better Lower Bounds for the General Problem

The critical point in the above construction, that determines the gap of the bandwidth, is the estimation of the space needed for a star in an already occupied pocket. We can increase this space by replacing the stars on the lines by cliques connected to the lines by new edges and thereby loosing the tree property. The nodes of a clique can themselves be not more then b positions apart and are spread over only two sections:

$$B(G) \geq b + \left\lceil \frac{s}{2} \right\rceil$$

Be $o = 2(n + 2m + 1)$, $s = 2l(n + 2m + 1)$, $l \geq 2$. Similar transformations lead to the following result for unsatisfied formulas:

$$B(G) \geq 3(l + 1)(n + 2m + 1) \tag{10}$$

And so we have

$$c > \frac{3}{2} - \epsilon \tag{11}$$

We can now conclude, that it is NP-hard to approximate the bandwidth of general graphs with an approximation ratio of $\frac{3}{2} - \epsilon$ or less.

Theorem 2.3 *For any $\epsilon > 0$, there is no polynomial time approximation algorithm with a approximation ratio $1.5 - \epsilon$, unless $P = NP$.*

This construction can be generalized to work for any SAT-formula, even with different numbers of literals in different clauses and multiple occurrences of the same literal in one clause. It then becomes somewhat more complicated to describe how many stars and nodes are at the different positions.

3 Open problems

An important computational problem still remains open about the existence of a $PTAS$ for the bandwidth problem on dense graphs (cf. [KWZ 97]).

Another important question is to improve both upper and lower approximation bounds on the general bandwidth problem, closing a large gap between $O(1)$ and $\log^{O(1)} n$ (cf. [Fe 97]).

References

[AKK 95] Arora, S., Karger, D., Karpinski, M., *Polynomial Time Approximation Schemes for Dense Instances of NP-Hard Problems*, Proc. 27^{th} ACM STOC (1995), pp. 284–293.

[BGHK 95] Bodlaender, H., Gilbert, J., Hafsteinsson, H., Kloks, T., *Approximating Treewidth, Pathwidth, Frontsize and Shortest Elimination Tree*, Journal of Algorithms **18** (1995), pp. 238–255.

[BJ 92] Bui, T., Jones, C., *Finding good approxminate vertex and edge partitions is NP-hard*, Information Processing Letters **42** (1992), pp. 153–159.

[CCDG 82] Chinn, P., Chvatalova, J., Dewdney, A., Gibbs, N., *The Bandwidth Problem for Graphs and Matrices - A Survey*, Journal of Graph Theory (1982), pp. 223–254.

[Ch 88] Chung, F., *Labelings of Graphs*, Beineke, L., Wilson, R., ed, Selected Topics in Graph Theory, pp. 151–168, Acadamic Press, 1988.

[DF 92] Downey, R., Fellows, M., *Fixed Parameter Intractability*, Proc. 7^{th} IEEE Conf. Structure in Complexity Theory (1992), pp. 36–49.

[Fe 97] Feige, U., *Approximating the bandwidth via volume respecting embeddings*, manuscript, 1997.

[Fe 96] Fernandez-de-la-Vega, W., *$MAX-CUT$ has a Randomized Approximation Scheme in Dense Graphs*, Random Structures and Algorithms **8** (1996), pp. 187–199.

[FK 96] Frieze, A., Kannan, R., *The Regularity Lemma and Approximation Schemes for Dense Problems*, Proc. 37^{th} IEEE FOCS (1996), pp. 12–20.

[FK 97] Frieze, A., Kannan, R., *Quick Approximation to Matrices and Applications*, Manuscript, 1997.

[GGJK 78] Garey, M., Graham, R., Johnson, D., Knuth, D., *Complexity Results For Bandwidth Minimization*, SIAM J. Appl. Math. **34** (1978), pp. 477–495.

[GGR 96] Goldreich, O., Goldwasser, S., Ron, D., *Property Testing and its Connection to Learning and Approximation*, Proc. 37^{th} IEEE FOCS (1996), pp. 339–348.

[HMM 91] Haralamides, J., Makedon, F., Monien, B., *Bandwidth minimization: an approximation algorithm for caterpillars*, Math. Systems Theory **24** (1991), pp. 169–177.

[Ho 97] Hochbaum, D., ed, *Approximation Algorithms for NP-hard Problems*, PWS Publ. Co., 1997.

[KW 97] Karpinski, M., Wirtgen, J., *On Approximation Hardness of the Bandwidth Problem*, Technical Report TR-97-041, ECCC, 1997.

[KWZ 97] Karpinski, M., Wirtgen, J., Zelikovsky, A., *An Approximation Algorithm for the Bandwidth Problem on Dense Graphs*, Technical Report TR-97-017, ECCC, 1997.

[LLR 95] Linial, N., London, E., Rabinovich, Y., *The Geometry of Graphs and some of its Algorithmic Applications*, Combinatorica **15**(2) (1995), pp. 215–245.

[Mo 86] Monien, B., *The Bandwidth Minimization Problem for Caterpillars with Hairlength 3 is NP-complete*, SIAM Journal on Algebraic Discrete Methods **7** (1986), pp. 505–512.

[Pa 76] Papadimitriou, C., *The NP-Completness of the Bandwidth Minimization Problem*, Computing **16** (1976), pp. 263–270.

[PY 91] Papadimitriou, C., Yannakakis, M., *Optimization, Approximation, and Complexity Classes*, Journal of Computing Systems Science **43** (1991), pp. 425–440.

[Sa 80] Saxe, J., *Dynamic programming algorithms for recognizing small-bandwidth graphs*, SIAM Journal on Algebraic Methods **1** (1980), pp. 363–369.

[Sm 95] Smithline, L., *Bandwidth of the complete k-ary tree*, Discrete Mathematics **142** (1995), pp. 203–212.