On Approximation Hardness of the Bandwidth Problem

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 and is known to be NP-hard, Papadimitriou [Pa 76]. There is not much known though on approximation hardness of this problem. In this paper we show, that there are no efficient polynomial time approximation schemes for the bandwidth problem under some plausible assumptions. Furthermore we show that there are no polynomial time approximation algorithms with an absolute error guarantee of $n^{1-\epsilon}$ for any $\epsilon > 0$ unless $P = NP$.

[*]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 $f : V \rightarrow \{1, ..., n\}$. 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.

The problem of finding the bandwidth of a graph is NP-hard [Pa 76], even for trees with maximum degree 3 [GGJK 78]. The general problem is not known to have any sublinear n^ϵ-approximation algorithm (cf. [KWZ 97], [Ka 97]).

Smithline [Sm 95] proved that the bandwidth of a complete k-ary tree can be computed in polynomial time. For 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. Karpinski, Wirtgen and Zelikovsky [KWZ 97] constructed a 3-approximation algorithm for δ-dense graphs.

Definition 1.1 *We call a graph dense if it has $\Omega(n^2)$ edges. 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 in computer science. In the best situation one can find approximation algorithms which work in polynomial time and approximate the optimal solutions within an arbitrary given constant. Such (meta-) algorithms are called polynomial time approximation schemes ($PTAS$s) cf. eg., [Ho 97]. Besides a problem instance, they have an additional input, an approximation ratio r. These schemes give us solutions which are r-near to the optimum in time, polynomial in the instance size. There are only few problems for which such schemes exist. For example the $KNAPSACK$ [IK 75], and $BINPACKING$ [FL 81], [KK 82], are known to have $PTAS$s. For the dense instances of problems in $MAXSNP$

[PY 88], the existence of $PTAS$ has been proven by Arora, Karger, Karpinski [AKK 95]. Arora [Ar 96] designed recently a $PTAS$ for the Euclidean traveling salesman problem.

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 the 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 $MAXCUT$ problem, which runs in $2^{(1/\epsilon)^{O(1)}}n^{O(1)}$. Frieze and Kannan [FK 96] obtained similar bounds for dense instances of some $MAXSNP$-hard problems using an algorithmic version of Szemeredi's regularity lemma. Another improvement was given by [GGR 96] and [FK 97].

In this paper we 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 sufficiently strong conditions. A similar approach was recently made by Cai and Chen [CC 93], and Cesati and Trevisan [CT 97].

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]. We relate the bandwidth to the treewidth and show similar results for the bandwidth problem, even for some special graph classes.

This paper is organized as follows. We introduce in section 2 the parameterized complexity hierarchy and follow the methods of Cesati and Trevisan [CT 97] to prove that there is probably no efficient approximation scheme for the bandwidth problem. In Section 3 we introduce some graph theoretical notions related to the bandwidth and discuss some known results of [ACP 87], [BGHK 95], [KKM 96]. In section 4 we relate the results of section 3 to the bandwidth problem in everywhere dense graphs and note in passing its NP-hardness, [KW 97]. Section 5 gives the proof, that there is no absolute approximation algorithm for the bandwidth problem unless $NP = P$.

2 There is no $EPTAS$ for the Bandwidth Problem

Many practical optimization problems are computationally intractable in the exact setting. A natural and reasonable way of dealing with this situation is to design heuristics for these problems. Such heuristics should guarantee their solutions reasonably near the optimum solution.

We say an approximation algorithm $\mathcal{A}$ for an optimization problem X has an *approximation ratio* r if $\mathcal{A}$ outputs for any instance $I \in X$ a solution with costs $\mathcal{A}(I)$, such that

$$\max\left\{\frac{\mathcal{A}(I)}{OPT(I)}, \frac{OPT(I)}{\mathcal{A}(I)}\right\} \leq r$$

with $OPT(I)$ the costs of the optimal solution. We are very interested in (meta-) algorithms that have an additional input, the approximation ratio. We call these algorithms *approximation schemes*. Formally, we define them as follows.

Definition 2.1 ($PTAS$) *Given an optimization problem X, we call a (meta-) algorithm $\mathcal{A}$ a polynomial time approximation scheme (PTAS) if for every fixed $\epsilon > 0$, $\mathcal{A}$ has approximation ratio $r = 1 + \epsilon$, and its running time is polynomial in the input size.*

Note that the running time can be exponential in $1/\epsilon$ or can be of order $n^{O(1/\epsilon)}$. We strengthen the above definition to the more efficient approximation algorithms classes.

Definition 2.2 ($EPTAS$,$FPTAS$) *We call a PTAS*

- *an efficient polynomial time approximation scheme (EPTAS) if the running time is of order $f(1/\epsilon)n^{O(1)}$. That means, the exponent in our polynomial is independent of ϵ but f can be a very fast growing function.*
- *a fully polynomial time approximation scheme (FPTAS), if the running time is a polynomial in n and $1/\epsilon$.*

In this section we note that there is probably no $EPTAS$ (and therefore no $FPTAS$) for the bandwidth problem. We follow the methods of Cesati and Trevisan [CT 97] to relate this approximation problem to the so called parameterized complexity and fixed parameter tractable problems (see cf. Downey and Fellows [DF 92]).

Many NP-hard problems become polynomial solvable if we restrict the problems to those, whose instances have constant size k. For the bandwidth problem for example there are $O(n^k)$-time algorithms [Sa 80]. Formally the parameterized problems are defined as follows.

Definition 2.3 *Let A be a subset of Σ^*. The parameterized problem of A is defined as the family $\{A_k\}_{k\in I\!N}$ where A_k is the set of instances whose witnesses have size k. A parameterized problem $\{A_k\}_{k\in I\!N}$ is in the class SP, if there is an algorithm whose running time is bounded by $n^{g(k)}$, where n is the size of the instance. $\{A_k\}_{k\in I\!N}$ is fixed parameter tractable (it is in the class FPT), if there is an algorithm with a running time bounded by $f(k)n^{O(1)}$.*

Downey and Fellows introduced a new notion of reduction for parameterized problems, the so called *parameterized reduction.* They introduced for each $k \in I\!N$, a basic parameterized problem Π_s and defined the class $W[s]$ to be the set of problems, which parameterized reduce to Π_s (see [DF 92]). This leads to the following parameterized hierarchy.

$$FPT \subseteq W[1] \subseteq W[2] ... \subseteq SP$$

It is believed that each of the inclusions is pure. We say that a problem is $W[s]$-*hard* if every problem in $W[s]$ parameterized reduces to it. If it is both in $W[s]$ an $W[s]$-hard, then it is $W[s]$-*complete*. Each problem which is hard for $W[s]$ for some $s \geq 1$ is conjectured not to have an algorithm with complexity bound $f(k)n^{O(1)}$. For most of the $W[s]$ classes are natural complete problem known [DF 92]. Under them is interestingly also the bandwidth problem.

Theorem 2.4 ([BFH 94]) *The bandwidth problem is $W[s]$-hard for all $s \geq 1$.*

This makes it improbable, that there are $f(k)n^{O(1)}$ algorithms for the k-parameterized bandwidth problem. Similar to [CT 97] we use this fact, to show that there is no $EPTAS$ for the bandwidth problem under certain conditions.

Theorem 2.5 *There is no $EPTAS$ for the bandwidth problem unless $FPT = W[k]$ (for all $k \geq 1$)*

PROOF: Suppose, that there is a $EPTAS$ $\mathcal{A}$ for the bandwidth problem. For fixed $\epsilon > 0$ and input length n $\mathcal{A}$ has a running time bounded by $T(n, \epsilon) = f(1/\epsilon)n^c$ ($c \in O(1)$). Now take any k-parameterized bandwidth instance G with $|G| = n$. We have to decide in time $g(k)n^{O(1)}$, whether $B(G) \leq k$ or $B(G) \geq k + 1$. Fix $\epsilon = \frac{1}{2k}$ and run $\mathcal{A}$ in time $f(2k)n^c$. We have two possibilities.

- $B(G) \leq k \Rightarrow \mathcal{A}(G) \leq (1 + \frac{1}{2k})k = k + 1/2$
- $B(G) \geq k + 1 \Rightarrow \mathcal{A}(G) \geq k + 1$

■

3 The Relationship of the Bandwidth and the Treewidth

We introduce now the notion of k-trees [ACP 87] and treewidth [BGHK 95] and relate them to the bandwidth. The class of *k-trees* is defined recursively as follows:

1. The complete graph on k vertices is a k-tree.
2. Let G be a k-tree on n vertices, then the graph constructed as follows is also a k-tree: add a new vertex and connect it to all vertices of a k-clique of G, and only to these vertices.

Any subgraph of a k-tree is called *partial k-tree*. *PARTIAL-k-TREE* is the problem given a graph G and an integer k, decide whether G is a partial k-tree or not. [ACP 87] shows that *PARTIAL-k-TREE* is NP-complete.

A *tree decomposition* of a graph $G = (V, E)$ is a pair $(\{X_i | i \in I\}, T = (I, F))$, where T is a tree and $\{X_i\}$ is a set of subsets of V such that

1. $\bigcup_{i \in I} X_i = V$
2. For all $\{u, v\} \in E$, there is an $i \in I$ with $u, v \in X_i$
3. For all $i, j, k \in I$, if j is on the path from i to k in T, then $X_i \cap X_k \subseteq X_j$.

The *treewidth* $tw((\{X_i\}, T), G)$ of a tree decomposition $(\{X_i\}, T)$ is defined by

$$tw((\{X_i\}, T), G) = \max_i |X_i| - 1$$

The *treewidth* $tw(G)$ of a graph G is then

$$tw(G) = \min_{(\{X_i\}, T)} tw((\{X_i\}, T), G)$$

Between the treewidth of a graph and the smallest k such that G is a partial k-tree exists the following well known connection:

Lemma 3.1 *For $k \geq 1$ the treewidth of a graph G is at most k if and only if G is a partial k-tree. Thus $tw(G)$ equals to the smallest k such that G is a partial k-tree.*

PROOF: See, for example, [Le 90]. ■

There is also a connection between the bandwidth and the treewidth of cobipartite graphs as showed in [KKM 96]. We call a graph *cobipartite* if it is the complement of a bipartite graph.

Lemma 3.2 ([KKM 96]) *Let G be a cobipartite graph. Then*

$$B(G) = tw(G)$$

Using Lemma 3.1 we get

Corollary 3.3 *Let G be a cobipartite graph. Then $B(G)$ equals to the smallest k such that G is a partial k-tree.*

In section 4 we will have a closer look at the proof of NP-hardness of this problem and prove that the instance for *PARTIAL-k-TREE* constructed there, is everywhere dense and cobipartite. Thus it is easy to show that the bandwidth problem on everywhere dense graphs is NP-hard.

4 NP-Hardness for Everywhere Dense Cobipartite Graphs

We sketch now the proof of NP-hardness of *PARTIAL-k-TREE* given in [ACP 87] to show that the constructed instance is a everywhere dense cobipartite graph. By the results stated in section 3, the NP-hardness of the bandwidth in everywhere dense graphs will follow.

Theorem 4.1 ([ACP 87]) *PARTIAL-k-TREE is NP-hard.*

PROOF: (Sketch) Let $G = (V, E)$ be a input graph of the NP-complete MINIMUM CUT LINEAR ARRANGEMENT (MCLA) problem (for the proof of NP-completeness see [GJ 79], [GT44]): given G and a positive integer k, does there exist a numbering f of V, such that

$$c(f,G) = \max_{1 \le j < n} |\{\{u,v\} \in E | f(u) \le j < f(v)\}| \le k$$

We will construct a bipartite graph $G' = (A \cup B, E')$. The vertices are defined as follows:

- Each $v \in V$ is represented by $\Delta(G)+1$ vertices in A, building the set A_v (We denote by $\Delta(G)$ the maximum vertex degree in G) and $\Delta(G) - \deg(v) + 1$ vertices in B, building the set B_v.
- For each edge $e \in E$ there are two vertices in B. They are denoted by B_e.

There are two different edge types in E':

- All vertices in A_v are connected to both vertices in B_e, if $v \in e$.
- All vertices of A_v are connected with all vertices in B_v.

Now define G'' to be G' after inserting all edges in A and B. Arnborg et al. showed the following connection: *G has a minimum linear cut value k with respect to some numbering f, if and only if the corresponding graph G'' is a partial k'-tree for* $k' = (\Delta(G)+1)(|V|+1)+k-1$. Since the construction of G'' is polynomial, it follows that $PARTIAL$-k'-$TREE$ is NP-hard. ∎

As a corollary we get the following theorem.

Theorem 4.2 *The bandwidth problem on everywhere dense graphs is NP-hard.*

Proof: Observe that the instance $G = (V, E)$ $(n = |V|)$ for $PARTIAL$-k-$TREE$ constructed in the proof of Theorem 4.1 is cobipartite. Further it is at least 1/2-dense, since the sets A and B build cliques and $|A| = |B|$:

$$\begin{aligned} |A| &= (\Delta(G)+1)|V| \\ &= \Delta(G)n + n \\ &= \Delta(G)n + n - \sum_{v\in V} \deg(v) + 2|E| \\ &= \sum_{v\in V} (\Delta(G) - \deg(v) + 1) + 2|E| \\ &= |B| \end{aligned}$$

Applying Corollary 3.3 it follows that the bandwidth on everywhere dense graphs is NP-hard, since G is cobipartite. ∎

5 There is no Approximation Algorithm with Absolute Error Guarantee of $n^{1-\epsilon}$

Bodlaender et al. [BGHK 95] shows an approximation lower bound for various parameters of graphs connected to sparse matrix factorization, including treewidth, pathwidth, and minimum elimination tree height. In particular they prove, that there is no absolute $n^{1-\epsilon}$ approximation (for any $\epsilon > 0$) for the treewidth of a graph. We use the connections of the bandwidth and the treewidth of cobipartite graphs, pointed out in Section 3 to show the same approximation lower bound for bandwidth problem.

An algorithm $\mathcal{A}$ is an absolute $f(n)$-approximation problem for a minimization problem X if for all $I \in X$

$$\mathcal{A}(I) \leq OPT(I) + f(n).$$

Only a few problems have such algorithms. For example for the edge coloring, we can use Vizing's [Vi 64] theorem to construct an absolute 1-approximation algorithm [Ho 81]. But the existence of such algorithms do not imply the existence of $PTAS$. It is known, that there are no 4/3-approximation algorithms for the edge coloring problem, unless $P = NP$ [Ho 81].

At first we apply the construction of [BGHK 95]. Given a graph $G = (V, E)$, we define for a constant k $G_k = (V_k, E_k)$ to be the graph with a $(k+1)$-clique for each vertex v in G. In G_k are all the vertices of the cliques corresponding to v and w connected, if and only if v and w are in G connected. Thus $V_k = \{v_i | 1 \leq i \leq k+1, v \in V\}$ and $E_k = \{\{v_i, w_j\} | 1 \leq i, j \leq k+1, \{v, w\} \in E\}$.

Lemma 5.1 ([BGHK 95]) $tw(G_k) = (tw(G) + 1)(k + 1) - 1$

PROOF: Let $(\{X_i | i \in I\}, T = (I, F))$ be a tree decomposition of G. It is easy to see, that $(\{Y_i | i \in I\}, T = (I, F))$ with $Y_i = \{v_j | v \in X_i, 1 \leq j \leq k+1\}$ is a tree decomposition of G_k with $tw((\{Y_i\}, T), G_k) = (tw((\{X_i\}, T), G) + 1)(k + 1) - 1$. Thus

$$tw(G_k) \leq (tw(G) + 1)(k + 1) - 1$$

On the other hand, take some tree decomposition $(\{Y_i | i \in I\}, T = (I, F))$ of G_k. Define $X_i = \{v \in V | \{v_1, ..., v_{k+1}\} \subseteq Y_i\}$

Take some edge $\{v, w\} \in E$. By the definition of G_k, $v_1, ..., v_{k+1}, w_1, ..., w_{k+1}$ form a clique. Then there exists an $i \in I$ with $\{v_1, ..., v_{k+1}, w_1, ..., w_{k+1}\} \in Y_i$ and thus $v, w \in X_i$. Let $j \in I$ be on the path in T from $i \in I$ to $k \in I$. If $v \in X_i \cap X_k$, then $\{v_1, ..., v_{k+1}\} \subseteq Y_i \cap Y_k$ and by the definition of the tree decomposition $\{v_1, ..., v_{k+1}\} \subseteq Y_j$. Thus v is in X_j. Therefore $(\{X_i\}, T)$ is a tree decomposition of G. Clearly $(k+1) \max_{i \in I} |X_i| \leq \max_{i \in I} |Y_i|$ and so

$$tw(G_k) \geq (tw(G) + 1)(k + 1) - 1$$

It follows, that $tw(G_k) = (tw(G) + 1)(k + 1) - 1$ ■

Since the construction of G_k is polynomial time, for $k \in n^{O(1)}$ we have the following theorem.

Theorem 5.2 ([BGHK 95]) *There is no polynomial time absolute $n^{1-\epsilon}$ approximation algorithm $\mathcal{A}$ for the treewidth problem for any $\epsilon > 0$.*

PROOF: At first we suppose, that k is a constant. Assume, we have a polynomial time absolute k approximation algorithm $\mathcal{A}$ for the treewidth problem. Take a NP-hard instance G of the treewidth problem. We will show that a polynomial time absolute k approximation algorithm can solve this instance exactly. Construct

G_k and run $\mathcal{A}$ on this modified instance, with $(\{Y_i\}, T)$ as a result. For the costs of this result we have the following bounds,

$$tw((\{Y_i\}, T), G_k) \leq OPT(G_k) + k$$

Now we apply the construction of Lemma 5.1, to find a tree decomposition of $(\{X_i\}, T)$ G with the costs

$$\begin{aligned} tw((\{X_i\}, T), G) &= \frac{(tw((\{Y_i\}, T), G_k) + 1)}{k+1} - 1 \\ &\leq \frac{(OPT(G_k) + k + 1)}{k+1} - 1 \\ &= \frac{OPT(G_k)}{k+1} \\ &= \frac{(OPT(G) + 1)(k+1) - 1}{k+1} \\ &= OPT(G) + 1 - 1/(k+1) \end{aligned}$$

Hence, we have a polynomial time algorithm for our instance G. Now we choose $k = n^c$ with $c = \lceil \epsilon/(\epsilon - 1) \rceil$. Since the construction is still polynomial, we meet the lower approximation bounds, stated in the theorem. ∎

To get the same lower bound for the bandwidth problem, we make the following observation.

Lemma 5.3 *If G is cobipartite, then G_k is cobipartite for all integers k.*

Now we can take the NP-hard instance of the bandwidth problem constructed in Theorem 4.2 and use the equivalence of treewidth and bandwidth for special graphs to get the following theorem.

Theorem 5.4 *There is no polynomial time absolute $n^{1-\epsilon}$ approximation algorithm $\mathcal{A}$ for the bandwidth problem for any $\epsilon > 0$.*

PROOF: Since the bandwidth of a cobipartite graph is equal to its treewidth and Lemma 5.3 holds, we can take the NP-hard instance of Theorem 4.2 in the proof of Theorem 5.2. ∎

Since cobipartite graphs are dense, and in our constructed hard instance everywhere dense, we get the following corollary.

Corollary 5.5 *There is no polynomial time absolute $n^{1-\epsilon}$ approximation algorithm $\mathcal{A}$ for the bandwidth problem on*

- *dense graphs,*
- *everywhere dense graphs,*

- *cobipartite graphs,*
- *cocomparability graphs*
- *asteroidal triple-free graphs,*

for any $\epsilon > 0$.

We have shown that it is unlikely that there are absolute approximation algorithms for the bandwidth problem in several graph classes. However it still possible, that there are $PTAS$s for some of this graph classes, since for all, but the first exist approximation algorithms, which construct solutions, which costs are within a constant multiplicative of the optimal costs [KKM 96], [KWZ 97].

6 Open Problems

An important computational problem remains open about the existence of a $PTAS$ for the bandwidth problem. At this moment we do not have even sublinear approximation algorithms for this problem. We also do not know the status of the problem for the very special case of binary trees.

Acknowledgment

We thank Haim Kaplan, Alex Zelikovsky and Uri Zwick for helpful discussions.

References

[ACP 87] Arnborg, S., Corneil, D., Proskurowski, A., *Complexity of Finding Embeddings in a k-Tree*, SIAM Journal on Algebraic Discrete Methods **8** (1987), pp. 277–284.

[Ar 96] Arora, S., *Polynomial Time Approximation Schemes for Euclidean TSP and other Geometric Problems*, Proc. 37^{th} IEEE FOCS (1996), pp. 2–11.

[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.

[BFH 94] Bodlaender, H., Fellows, M., Hallet, M., *Beyond NP-Completeness for problems of bounded width: Hardness for the W hierachy*, Proc. ACM STOC (1994), pp. 449–458.

[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.

[CC 93] Cai, L., Chen, J., *On fixed-parameter tractability an approximability of NP-hard optimization problems*, Proc. 2^{nd} Israel Symposium on Theory of Computing and Systems (1993), pp. 118–126.

[CT 97] Cesati, M., Trevisan, L., *On the Efficiency of Polynomial Time Approximation Schemes*, Information Processing Letters (1997), to appear.

[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.

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

[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.

[FL 81] Fernandez-de-la-Vega, W., Lueker, G., *Bin Packing can be solved within $1+\epsilon$ in Linear Time*, Combinatorica **1** (1981), pp. 349–355.

[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.

[GJ 79] Garey, M. R., Johnson, D. S., *Computers and Intractability*, W. H. Freeman and Company, 1979.

[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.

[Ho 81] Holyer, I., *The NP-Completeness of Edge-Coloring*, SIAM Journal on Computing **10** (1981), pp. 718–720.

[IK 75] Ibarra, O., Kim, C., *Fast Approximation Algorithms for the Knapsack and Sum of Subsets Problems*, Journal of the ACM **22** (1975), pp. 463–468.

[KK 82] Karmarkar, M., Karp, R., *An Efficient Approximation Scheme for the One-Dimensional Bin-Packing Problem*, Proc. 23^{rd} IEEE FOCS (1982), pp. 312–320.

[Ka 97] Karpinski, M., *Polynomial Time Approximation Schemes for Some Dense Instances of NP-Hard Optimization Problems*, Proc. 1^{st} Symposium on Randomization and Approximation Techniques in Computer Science, RANDOM'97, Lecture Notes in Computer Science Vol.1269, Springer Verlag (1997), pp. 1–14.

[KW 97] Karpinski, M., Wirtgen, J., *NP-Hardness of the Bandwidth Problem on Dense Graphs*, Research Report No. 85176-CS, University of Bonn, 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.

[KKM 96] Kloks, T., Kratsch, D., Müller, H., *Approximating the Bandwidth for Asteroidal Triple-Free Graphs*, Proc. 3^{rd} European Symposium on Algorithms (1996), pp. 434–447, Lecture Notes in Comput. Sci. 979, Springer Verlag.

[Le 90] Leeuwen, J. v., *Graph Algorithms*, in Handbook of Theoretical Computer Science, Volume A, Algorithms and Comlexity, pp. 525–631, Elsevier and The MIT Press, 1990.

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

[PY 88] Papadimitriou, C., Yannakakis, M., *Optimization, Approximation, and Complexity Classes*, Proc. 20^{th} ACM STOC (1988), pp. 229–234.

[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.

[Vi 64] Vizing, P., *On an Estimate of the Chromatic Class of a p-Graph*, Diskret. Analiz **3** (1964), pp. 23–30, in Russian.