NP-Hardness of the Bandwidth Problem on Dense Graphs

Marek Karpinski* Jürgen Wirtgen†

Abstract

The *bandwidth problem* is the problem of numbering 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 and varied history and is known to be NP-hard Papadimitriou [Pa 76]. Recently for dense graphs a constant ratio approximation algorithm for this problem has been constructed in Karpinski, Wirtgen and Zelikovsky [KWZ 97]. In this paper we prove that the bandwidth problem on the dense instances remains NP-hard.

*Dept. of Computer Science, University of Bonn, 53117 Bonn. 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 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.

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 algorithms. There are only few cases where we can find the optimal layout in polynomial time. Saxe [Sa 80] designed an algorithm which decides whether a given graph has bandwidth at most k in time $O(n^k)$ by dynamic programming. Bandwidth two can be checked in linear time [GGJK 78]. Kratsch [Kr 87] introduced an exact $O(n^2 \log n)$ algorithm for the bandwidth problem in interval graphs. Smithline [Sm 95] proved that the bandwidth of the complete k-ary tree $T_{k,d}$ with d levels and k^d leaves is exactly $\lceil k(k^d - 1)/(k - 1)(2d) \rceil$. Her proof is constructive and contains a polynomial time algorithm, for this task. 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. In Karpinski, Wirtgen and Zelikovsky [KWZ 97] introduced a 3-approximation algorithm for everywhere δ-dense graphs.

Definition 1 ([AKK 95]) *We call a graph G (everywhere) δ-dense, if the minimum degree $\delta(G)$ is at least δn. We call it dense in average, if the number of edges is in $\Omega(n^2)$.*

In this paper we show that the bandwidth problem on dense graphs is NP-hard, answering the question raised in [KWZ 97].

This paper is organized as follows. Section 2 gives a simple proof for the NP-hardness of the bandwidth problem for dense graphs in average. In Section 3 we observe some to the bandwidth related notations in graph theory 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 dense graphs and prove its NP-hardness.

2 NP-Hardness for Dense Graphs in Average

In this section we will prove the NP-hardness of the bandwidth problem on dense graphs in average. Although it is a weaker result as the result of 4, we include it to introduce some new techniques.

We start with the standard PARTITION problem: given a set $A = \{1, ..., n\}$ and sizes $s : A \to \mathbb{Z}^+$ - the question is, whether there is a subset $I \subseteq A$, such that $\sum_{i \in I} s(i) = \sum_{i \in A \setminus I} s(i)$? The problem is NP-complete (cf. [GJ 79] [SP12]).

We can reformulate the PARTITION problem as the following GPARTITION problem: given a graph $G = (V, E)$ partition V in $V_1 \cup V_2$, such that

- $|V_1| = |V_2|$
- There are no edges between V_1 and V_2.

We will call such a partition a *good* partition.

Lemma 2 *GPARTITION is NP-complete.*

PROOF: Clearly it is NP-easy. For the NP-hardness take an instance (A, s) of PARTITION and construct the graph $G = (V, E)$ as follows: for each $i \in A$ there are $s(i)$ copies $v_{i,1}, ..., v_{i,s(i)}$ in V, building a clique. Now suppose we have a partition (V_1, V_2) of V such that $|V_1| = |V_2|$ and that there are no edges between V_1 and V_2. Each clique is completely in one of the sets. Define $I = \{i | v_{i,1} \in V_1\}$. Since $|V_1| = |V_2|$, we have $\sum_{i \in I} s(i) = \sum_{i \in A \setminus I} s(i)$. ■

Now we show in Theorem 3 how to reduce GPARTITION to the bandwidth problem, yielding an alternative proof of its NP-hardness [Pa 76]. In Corollary 4 we show how to densify the instances constructed in the proof of Theorem 3 to get the NP-hardness of the bandwidth problem on dense graphs in average.

Theorem 3 ([Pa 76]) *The bandwidth problem is NP-hard.*

PROOF: We take an instance $G = (V = \{v_1, ..., v_n\}, E)$ of GPARTITION. Now construct the graph $G' = (V', E')$:

- $V' = V \cup \{z_1, ..., z_n, x\}$
- $E' = E \cup \{\{v_1, z_i\} | i = 1, ..., n\} \cup \{\{x, z_i\} | i = 1, ..., n\}$

Since $\deg(x) = n$ we know that $B(G') \geq n/2$. We will show, that $B(f, G') = n/2$ iff the numbering f defines a good partition. If $B(f, G') = n/2$, then the z-vertices have to be on both sides of x, building blocks Z_1 and Z_2. Thus the vertices of V have to be either on the left side of Z_1 or on the right side of Z_2, building blocks V_1 and V_2 (see figure 1).

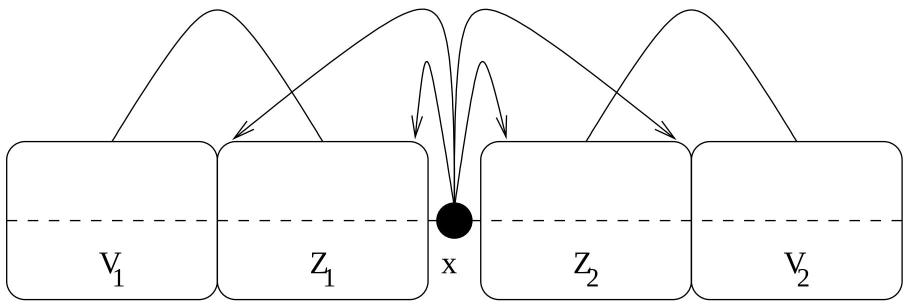

Figure 1: A layout f with $B(f, G') = n/2$.

Since $B(f, G') = n/2$ no edges cross x. The only possibility for the arrangement is that the vertices in V_i are the same as in Z_i appearing in the same order as their copies. It follows, that the partition (V_1, V_2) is a good partition for GPARTITION. Thus $B(G) = n/2$ iff there is a good partition of V. ■

Now we add a large clique $C_{n/2+1}$ to the graph. Since $|V'| = O(n)$, the resulting graph is dense in average. Using the same arguments as in the proof of theorem 3, we get

Corollary 4 *The bandwidth problem for dense graphs in average is NP-hard.*

In section 4 we will strengthen the result to everywhere dense graphs.

3 Related Notations and Known Results

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*. Arnborg et al. showed in [ACP 87] that PARTIAL-k-TREE is NP-complete. PARTIAL-k-TREE is the problem given a graph G and an integer k, decide whether G is a partial k-tree or not.

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 5 *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 6 ([KKM 96]) *Let G be a cobipartite graph. Then*

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

Using Lemma 5 we get

Corollary 7 *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 to the proof of NP-hardness of PARTIAL-k-TREE 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 dense graphs is NP-hard.

4 NP-Hardness for Everywhere Dense Graphs

First of all we sketch the proof of NP-hardness of PARTIAL-k-TREE proposed 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 bandwidth in everywhere dense graphs follows.

Theorem 8 ([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

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

We will construct a bipartite graph $G' = (A \dot{\cup} B, E')$. 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 9

The bandwidth problem on everywhere $1/2$-dense graphs is NP-hard.

PROOF: Observe that the instance for PARTIAL-k-TREE constructed in the proof of Theorem 8 is cobipartite. Further it is at least $1/2$-dense, since the sets A

and B build cliques and $|A| = |B|$:

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

Applying Corollary 7 it follows, since G is cobipartite that the bandwidth on dense graphs is NP-hard. ■

5 Open Problems

An important computational problem remains open about the approximation hardness of the bandwidth on the dense instances.

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.

[AKK 95] Arora, S., Karger, D., Karpinski, M., *Polynomial Time Approximation Schemes for Dense Instances of NP-Hard Problems*, Proc. 36^{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.

[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.

[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.

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

[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.

[Kr 87] Kratsch, D., *Finding the Minimum Bandwidth of an Interval Graph*, Information and Computing **74** (1987), pp. 140–187.

[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.

[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.