Bandwidth of graphs with few P_4s

Ton Kloks[1] and Richard Tan[2]

[1] Department of Applied Mathematics and DIMATIA
Charles University
Malostranské nám. 25, 118 00 Praha 1
Czech Republic
kloks@kam.ms.mff.cuni.cz

[2] Utrecht University
Department of Computer Science
P. O. Box 80.089, 3508 TB Utrecht
The Netherlands
rbtan@cs.uu.nl

Abstract. In this paper we give a polynomial time algorithm to compute the bandwidth of a $(q, q-4)$-graph for each constant q.

1 Introduction

The BANDWIDTH minimization problem is the following problem: given a graph G and an integer $k \geq 0$, to map the vertices of G to distinct positive integers, so that no edge of G has its endpoints mapped to integers that differ by more than k.

The problem is motivated by the bandwidth minimization problem for matrices: given an $n \times n$ matrix A and a nonnegative integer k, to find whether there is a permutation matrix P such that $P \cdot A \cdot P^T$ is a matrix with all nonzero entries on the main diagonal or on the k diagonals on either side of this main diagonal.

Computing the bandwidth of a graph is NP-complete [14], even when restricted to trees of maximum degree three [7].

There are only a few graph classes known for which the bandwidth can be computed efficiently. Such graph classes are the class of theta graphs [15], cographs [10] and chain graphs. Another one is the class of caterpillars with hairs of length one and two [1]. However, for caterpillars with hairs of length at most three, the BANDWIDTH problem remains NP-complete [13].

It can be checked in linear time whether the bandwidth of a graph is at most two [7]. For general k, there is an $O(n^k)$ algorithm to check whether the bandwidth of a graph is at most k [8]. In some sense, this is best possible, since it was shown in [4] that BANDWIDTH is $W[t]$-hard for all t in the fixed parameter hierarchy. Hence in general it is not expected that there is an $O(n^\alpha)$ algorithm for any fixed α.

There is one other non-trivial graph class for which the bandwidth can be computed efficiently. This is the class of interval graphs. It was shown in a series of papers that the bandwidth of an interval graph can be computed efficiently in a greedy manner [11, 12, 16].

In this paper we extend the result of [10]. We show that the bandwidth problem is solvable in polynomial time for graphs with few P_4s. (In this paper a P_4 is a path with four vertices.) This has become the common name for $(q, q-4)$-graphs which are graphs for which no set of at most q vertices contains more than $q-4$ distinct P_4's. Cographs are a well-known example of this parametrized class; cographs are exactly the $(4, 0)$-graphs.

We show that the bandwidth problem for $(q, q-4)$-graphs can be solved in polynomial time for every fixed constant q.

2 Preliminaries

Definition 1. A *layout* L of the graph $G = (V, E)$ is a 1-1 mapping $V \leftrightarrow \{1, ..., |V|\}$. If G has no edges then the *width* $b(G, L)$ of L is zero. Otherwise the width $b(G, L) = \max\{|L(u) - L(v)| \mid \{u, v\} \in E\}$. The *bandwidth* of G is

$$bw(G) = \min\{b(G, L) \mid L \text{ is a layout of } G\}.$$

Definition 2. A graph is a *(q, t)-graph* if no set of at most q vertices induces more than t distinct P_4's.

Notice that the class of cographs are exactly the $(4, 0)$-graphs.

It was shown in [3] that many problems can be solved efficiently for $(q, q-4)$-graphs for each constant q. In this paper we show that also the bandwidth problem can be solved in polynomial time for $(q, q-4)$-graphs for each constant q.

A result of [2] shows that $(q, q-4)$-graphs can be characterized quite effectively using the *primeval tree decomposition* introduced in [9].

We need some preliminaries.

Definition 3. A *spider* is a splitgraph consisting of a clique and an independent set of equal size at least two such that each vertex of the independent set has precisely one neighbor in the clique and each vertex of the clique has precisely one neighbor in the independent set, or it is the complement of such a graph.

Definition 4. A spider is *thin* if every vertex of the independent set has one neighbor in the clique. A spider is *thick* if every vertex of the independent set is non adjacent to one vertex of the clique.

Notice that a P_4 is both thick and thin and that this is the smallest spider.

Definition 5. A subset M of V with $1 < |M| < |V|$ is called a *homogeneous set* if each vertex outside is either adjacent to all vertices of H or to none of them.

Definition 6. The graph obtained from G by shrinking every maximal homogenous set to one single vertex is called the *characteristic graph* of G.

Recall that a splitgraph is a graph of which the vertex set can be split into two sets K and S such that K induces a clique and S induces an independent set in G.

Definition 7. A graph G is called an *expanded splitgraph* if its characteristic is a splitgraph. The maximal homogeneous sets of G corresponding with vertices of the clique of the characteristic are called the *clique modules* and the other maximal homogeneous sets are called *independent set modules*.

It was shown in [9, 2] that $(q, q-4)$-graphs are exactly the graphs that can be constructed using the following primeval decomposition. A primeval tree is a rooted binary tree, in which the internal nodes (i.e., all nodes except the leaves) are labeled with 0, 1, or 2. Each node of the tree corresponds with a graph. The leaves of the tree are either spiders or expanded splitgraphs with less than q vertices.

Each internal node of the tree corresponds with a graph constructed from its two children as follows. Let G_1 and G_2 be the two graphs corresponding with the children of the internal node.

If the label of the node is 0 the graph corresponding with this node is the *union* of $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$: i.e., the union is $G = (V_1 \cup V_2, E_1 \cup E_2)$. If the label of the node is 1 the graph is the *sum* of G_1 and G_2, i.e., every vertex of G_1 is made adjacent to every vertex of G_2. If the label of the node is 2, one of the graphs, say G_1 is either a spider or an expanded splitgraph. If G_1 is a spider, all vertices of G_2 are made adjacent exactly to all vertices of the clique of G_1. If G_1 is an expanded splitgraph, all vertices of G_2 are made adjacent exactly to all vertices of every clique module of G_1.

The *primeval tree decomposition* for $(q, q-4)$-graphs can be found in linear time [5].

If only the first two operations are considered and if all leaves of the tree are single vertices we get the extensively studied and well known class of cographs (i.e., the graphs without induced P_4 or, equivalently, the $(4, 0)$-graphs). A linear time algorithm for this special case was recently developed in [10].

In the following sections we prove the following theorem.

Theorem 8. *For every constant q there exists a polynomial time algorithm to compute the bandwidth of a $(q, q-4)$-graph.*

3 Bandwidth of spiders

Consider the primeval tree decomposition of the $(q, q-4)$-graph G. The leaves of this decomposition are either spiders or graphs of bounded size (i.e., with less than q vertices). The bandwidth of a graph with at most q (q considered as a constant) vertices can clearly be computed in constant time by considering all possible layouts. In this subsection we consider the other type of leaves, i.e. thick and thin spiders.

Lemma 9. *Let G be a thin spider with clique K and independent set S. Then $bw(G) = |K| - 1$.*

Proof. By definition $|K| > 1$. Clearly $bw(G) \geq |K| - 1$, since for any layout there is an edge between the last and first vertex of the clique. So we only have to show that there is a layout which achieves the lower bound. Take a layout that starts with all vertices of K. Then the next vertex is the vertex of S adjacent to the first vertex of K. etc. This gives a layout with width $|K| - 1$. □

Lemma 10. *Let G be a thick spider with clique K and independent set S. then $bw(G) = \left\lceil \frac{3|K|}{2} \right\rceil - 2$.*

Proof. First we show that we can obtain a layout with the width mentioned in the lemma. Start with $\left\lceil \frac{|K|}{2} \right\rceil$ vertices of S, next all vertices of K and end with the rest of the vertices of S. Now choose the first vertex of S such that it is not adjacent to the last vertex of K and choose the last vertex of S such that it is not adjacent to the first vertex of K. The width of this layout is then $\left\lceil \frac{3|K|}{2} \right\rceil - 2$.

Consider an optimal layout L. Let i be the first position of a vertex $x \in K$ and let j be the last position of a vertex in $y \in K$ in L. Now x is non adjacent to at most one of the two first vertices in the layout L and y is non adjacent to at most one of the last two vertices in L. Since i is the first position and j the last position of a vertex of K: $j - i + 1 \geq |K|$. Hence:

$$bw(G, L) \geq \max(2|K| - 1 - i, j - 2) \geq \frac{2|K| + j - i - 3}{3} \geq \frac{3|K| - 4}{2}$$

i.e., $bw(G) \geq \left\lceil \frac{3|K|}{2} \right\rceil - 2$. □

4 Bandwidth for the sum and the union

The results presented in this section appeared earlier in [10]. For completeness sake, we added them to this paper.

Lemma 11. *If G is the union of G_1 and G_2 then $bw(G) = \max(bw(G_1), bw(G_2))$.*

Proof. Obvious. □

We now consider the case where G is the sum of G_1 and G_2. Let n_1 be the number of vertices of G_1, n_2 the number of vertices of G_2 and $n = n_1 + n_2$ the number of vertices of G.

Consider an optimal layout L. If the first vertex in L is a vertex of G_1 and the last vertex in L is a vertex of G_2 then the width of L is $n - 1$. This is clearly the worst possible case. Without loss of generality we assume that in some optimal layout the first and last vertex of L are vertices of G_1.

Let i be the first position in L of a vertex of G_2 and j be the last position of a vertex of G_2. Then clearly $bw(G) \geq \max(j - 1, n - i) \geq \left\lceil \frac{n_1}{2} \right\rceil + n_2 - 1$.

Claim. There exists an optimal layout such that all vertices of G_2 occur consecutively.

Proof. Notice that the width of L is either achieved by the above mentioned lower bound, or by an edge connecting two vertices of G_1; one in a position $< i$ and one in a position $> j$. Permuting the vertices in positions from i to j does not change the width of L. Hence we may permute these vertices in such a way that all vertices of G_2 occur in a consecutive order in L. □

We consider a layout L obtained as follows. Take any optimal layout L_1 of G_1. Add the vertices of G_2 (in any order) after the first $\lceil \frac{n_1}{2} \rceil$ vertices of G_1. We claim this gives an optimal layout (under the assumption that an optimal layout can be obtained with the first and last position occupied by vertices of G_1).

Consider two cases. First consider the case where $bw(G_1) \leq \lceil \frac{n_1}{2} \rceil - 1$. Notice that a "cross-over" edge cannot have width larger than $\lceil \frac{n_1}{2} \rceil + n_2 - 1$. Hence in this case the width of L is $\lceil \frac{n_1}{2} \rceil + n_2 - 1$ (which must be optimal since this is the above mentioned lower bound).

Now consider the case $bw(G_1) > \lceil \frac{n_1}{2} \rceil - 1$. In this case the width of L is $n_2 + bw(G_1)$, and we claim that this is optimal. Consider some layout L' starting and ending with a vertex of G_1 and with all vertices of G_2 consecutive. Remove all vertices of G_2 from L' and call L^* this layout for G_1. Let p and q (with $p < q$) be the positions in L^* of vertices of G_1 which are then furthest apart. If p and q appear both before the first vertex of G_2 in L', the longest edge in L' is at least $n_2 + q - p \geq n_2 + bw(G_1)$, since there is an edge between the last vertex of G_2 and the vertex at position p. A similar argument gives the same bound if p and q appear both after the last element of G_2. If p appears before the vertices of G_2 and q appears after the vertices of G_2, we also obtain a width $q - p + n_2 \geq n_2 + bw(G_1)$.

Corollary 12. *There exists a polynomial time algorithm to compute the bandwidth of cographs.*

Remark. In [10] it is shown that this algorithm can be implemented such that it runs in linear time.

5 Bandwidth for $(q, q-4)$-graphs

In this section we consider operation 2. In this case, G is obtained from a graph $G_2 = (V_2, E_2)$ and a a graph $G_1 = (V_1, E_1)$ which is either a spider or a graph with less than q vertices.

5.1 $V_1 < q$ and the characteristic of G_1 is a splitgraph

Since V_2 is a module in G, we can assume that an optimal layout for G can be obtained by taking *any* optimal layout for G_2 and add vertices of G_1 to it.

Corollary 13. *In this case the bandwidth of G can be computed in $O(|V_2|^{q-1}(n+e))$ time (given an optimal layout for G_2).*

Proof. Add the vertices of G_1 in all possible position in the layout for G_2, and compute the resulting bandwidth. $\square$

5.2 G_1 is a thin spider.

Let the clique of G_1 be K and the independent set be S. The vertices of G_2 are adjacent to all vertices of K and no vertices of S.

Lemma 14. *If $bw(G_2) \leq \lceil \frac{n_2}{2} \rceil - 1$, then $bw(G) = |K| + \lceil \frac{n_2}{2} \rceil - 1$.*

Proof. First we show there exists a layout L with this width. Consider any optimal layout L_2 of G_2. Construct L as follows. Start with $\left\lfloor \frac{|K|}{2} \right\rfloor$ vertices of S. Then take the first $\lceil \frac{n_2}{2} \rceil$ of L_2. Next take all vertices of K, then the next $\lfloor \frac{n_2}{2} \rfloor$ vertices of G_2 and finally the last $\left\lceil \frac{|K|}{2} \right\rceil$ vertices of S. Take the ordering of the vertices of S in such a way that the first vertex of S is adjacent to the first vertex of K, the second vertex of S adjacent to the second vertex of K etc. And the last vertex of S adjacent to the last vertex of K. The width of this layout $bw(G, L) = |K| + \lceil \frac{n_2}{2} \rceil - 1$. (A cross-over edge between vertices of G_2 cannot have width larger than this because $bw(G_2) \leq \lceil \frac{n_2}{2} \rceil - 1$.)

Next we show that $|K| + \lceil \frac{n_2}{2} \rceil - 1$ is a lowerbound for the bandwidth of the graph induced by $V_2 \cup K$. By the claim proved in the previous section, we may assume that either the vertices of k are consecutive or the vertices of G_2 are consecutive in an optimal layout for $G[V_2 \cup K]$. If the vertices of G_2 were consecutive the width of a layout would be at least $|K| + n_2 - 1$ since K is a clique. Hence we may assume that the vertices of G_2 occur consecutively. Let i be the first position of a vertex of K in an optimal layout of $G[V_2 \cup K]$ and let j be the last position of a vertex of K. Hence $j - i + 1 = |K|$. Now the width of this layout is at least $\max(j - 1, n_2 + |K| - i) \geq \lceil \frac{n_2}{2} \rceil + |K| - 1$ (since the maximum of the two values is at least the average). $\square$

Lemma 15. *If $bw(G_2) > \lceil \frac{n_2}{2} \rceil - 1$ then $bw(G) = |K| + bw(G_2)$.*

Proof. First we show that the layout L constructed in the proof of Lemma 14 has the width mentioned in this lemma. Since $bw(G_2) > \lceil \frac{n_2}{2} \rceil - 1$, there must be an edge between vertices of G_2 crossing over all vertices of K. No other edges have a larger width.

Now we show that $G[V_2 \cup K]$ has bandwidth $|K| + bw(G_2)$. The claim in the previous section shows that there is an optimal layout for this subgraph such that either all vertices of K are consecutive or all vertices of G_2. If all vertices of G_2 would be consecutive, we would obtain a width of $|K| + n_2 - 1$ which is clearly at least $|K| + bw(G_2$. So we may assume there is an optimal layout with all vertices of K consecutive. Now the rest of the proof is exactly the same as in the second case of the previous section. $\square$

5.3 G_1 is a thick spider.

Let again the clique of G_1 be K and the independent set be S. Notice that every vertex of K is non adjacent to exactly one unique vertex of S and adjacent to all other vertices.

Lemma 16. $bw(G) \geq \lceil \frac{n_2+3|K|}{2} \rceil - 2$.

Proof. Consider an optimal layout L for G. Let i be the first position of a vertex of K and let j be the last position of a vertex of K. Then clearly $j - i + 1 \geq |K|$. Since each vertex of K is adjacent to all vertices except one vertex of S we have

$$bw(G, L) \geq \max(n - 1 - i, j - 2) \geq \frac{n + j - i - 3}{2} \geq \frac{n_2 + 3|K|}{2} - 2$$

Hence $bw(G) \geq \lceil \frac{n_2+3|K|}{2} \rceil - 2$. □

Lemma 17. *If* $bw(G_2) \leq \lceil \frac{n_2+|K|}{2} \rceil - 2$ *then* $bw(G) = \lceil \frac{n_2+3|K|}{2} \rceil - 2$.

Proof. We show that there is a layout achieving the lowerbound of Lemma 16. Consider the same layout as in the proof of Lemma 14. The width of this layout is

$$\max(\left\lfloor \frac{n_2}{2} \right\rfloor + \left\lceil \frac{3|K|}{2} \right\rceil - 2, \left\lceil \frac{n_2}{2} \right\rceil + \left\lfloor \frac{3|K|}{2} \right\rfloor - 2) = \left\lceil \frac{n_2 + 3|K|}{2} \right\rceil - 2$$

□

Lemma 18. *If* $bw(G_2) > \lceil \frac{n_2+|K|}{2} \rceil - 2$ *then* $bw(G) = bw(G_2) + |K|$.

Proof. First notice that the layout considered in the proof of the previous lemma has width $bw(G_2) + |K|$, hence $bw(G) \geq bw(G_2) + |K|$ in this case.

We now show that this is optimal. Again, it is sufficient to show that the subgraph induced by $V_2 \cup K$ has this width. The proof of this fact is exactly the same as in the proof of Lemma 15. □

6 Concluding remarks

In this paper we have shown that the bandwidth can be computed in polynomial time for $(q, q-4)$-graphs for each constant q. We would like to mention the following open problems. As far as we know, the complexity of the bandwidth problem for permutation graphs and for splitgraphs is still open.

References

1. S.F. Assmann, G.W. Peck, M.M. Sysło and J. Zak, The bandwidth of caterpillars with hairs of length 1 and 2, *SIAM Journal on Algebraic and Discrete Methods* **2** (1981), pp. 387–393.
2. Babel, L., On the P_4-structure of graphs, *Habilitationsschrift*, Zentrum für Mathematik, Technische Universität München, 1997.
3. Babel, L., T. Kloks, J. Kratochvíl, D. Kratsch, H. Müller, and S. Olariu, Efficient algorithms for graphs with few P_4s. Manuscript 1998.
4. H. Bodlaender, M.R. Fellows and M.T. Hallet, Beyond NP-completeness for problems of bounded width: Hardness for the W hierarchy, *Proceedings of the 26th Annual Symposium on the Theory of Computation*, 1994, pp. 449–458.
5. Baumann, S., A linear algorithm for the homogeneous decomposition of graphs, Report No. M-9615, Zentrum für Mathematik, Technische Universität München, 1996.
6. P.Z. Chinn, J. Chvátalová, A.K. Dewdney and N.E. Gibbs, The bandwidth problem for graphs and matrices—a survey, *Journal of Graph Theory* **6** (1982), pp. 223–254.
7. M.R. Garey, R.L. Graham, D.S. Johnson and D.E. Knuth, Complexity results for bandwidth minimization, *SIAM Journal on Applied Mathematics* **34** (1978), pp. 477–495.
8. E. Gurari and I.H. Sudborough, Improved dynamic programming algorithms for the bandwidth minimization and the mincut linear arrangement problem, *Journal of Algorithms* **5** (1984), pp. 531–546.
9. Jamison, B. and S. Olariu, p-components and the homogeneous decomposition of graphs, *SIAM J. Discrete Mathematics* **8**, (1995), pp. 448–463.
10. S. Jiang, The bandwidth problem and bandwidth of cographs. Manuscript, 1992.
11. D.J. Kleitman and R.V. Vohra, Computing the bandwidth of interval graphs, *SIAM Journal on Discrete Mathematics* **3** (1990), pp. 373–375.
12. R. Mahesh, C.P. Rangan and A. Srinivasan, On finding the minimum bandwidth of interval graphs, *Information and Computation* **95** (1991), pp. 218–224.
13. B. Monien, The bandwidth minimization problem for caterpillars with hair length 3 is NP-complete, *SIAM Journal on Algebraic and Discrete Methods* **7** (1986), pp. 505–512.
14. C.H. Papadimitriou, The NP-completeness of the bandwidth minimization problem, *Computing* **16** (1976), pp. 263–270.
15. G.W. Peck and A. Shastri, Bandwidth of theta graphs with short paths, *Discrete Mathematics* **103** (1992), pp. 177–187.
16. A.P. Sprague, An $O(n \log n)$ algorithm for bandwidth of interval graphs, *SIAM Journal on Discrete Mathematics* **7** (1994), pp. 213–220.

This article was processed using the LaTeX macro package with LLNCS style