Discrete Mathematics and Theoretical Computer Science **3**, 1999, 141–150

A note on domino treewidth†

Hans L. Bodlaender

Department of Computer Science, University of Utrecht, P.O.Box 80.089, 3508 TB Utrecht, the Netherlands. Email: hansb@cs.uu.nl.

received August 11, 1998, *revised May 14, 1999*, *accepted May 27, 1999*.

In [DO95], Ding and Oporowski proved that for every k, and d, there exists a constant $c_{k,d}$, such that every graph with treewidth at most k and maximum degree at most d has domino treewidth at most $c_{k,d}$. This note gives a new simple proof of this fact, with a better bound for $c_{k,d}$, namely $(9k+7)d(d+1)-1$.

It is also shown that a lower bound of $\Omega(kd)$ holds: there are graphs with domino treewidth at least $\frac{1}{12}kd-1$, treewidth at most k, and maximum degree at most d, for many values k and d. The domino treewidth of a tree is at most its maximum degree.

Keywords: treewidth, domino treewidth, graph algorithms, tree decompositions

1 Introduction

In [DO95], Ding and Oporowski proved that for every k and d, every graph $G = (V,E)$ with treewidth at most k and with maximum degree at most d has a tree decomposition of width at most $\max(600k^2d^3, 5400d^3)$, such that every vertex $v \in V$ belongs to at most two of the sets associated to the nodes in the tree decomposition. Such a tree decomposition was called a *domino tree decomposition* by Bodlaender and Engelfriet in [BE97], where they independently gave a similar result, but with a more complicated proof and with a much higher constant, which was exponential, both in k and in d.

In this note, a new and easy to understand proof for the result is given. Additionally, the constant factor arising from the proof given here is smaller: it is shown that graphs with treewidth at most k and maximum degree at most d have domino treewidth at most $(9k+7)d(d+1)-1$.

The proof uses amongst others a technique from [BGHK95] (inspired by a technique from [RS95]), and some other ideas. The proof is given in Section 3.

In Section 4, it is shown that a lower bound of $\Omega(kd)$ holds: there are graphs with domino treewidth at least $\frac{1}{12}kd-1$, treewidth at most k, and maximum degree at most d, for many values k and d.

Some final remarks are made in Section 5, and it is shown that the domino treewidth of a tree is at most its maximum degree.

†This research was partially supported by ESPRIT Long Term Research Project 20244 (project ALCOM IT: *Algorithms and Complexity in Information Technology*).

2 Definitions and Preliminary Results

Definition. A tree decomposition of a graph $G = (V, E)$ is a pair $(\{X_i \mid i \in I\}, T = (I, F))$ with $\{X_i \mid i \in I\}$ a collection of subsets of V, and $T = (I, F)$ a tree, such that

- $\bigcup_{i \in I} X_i = V$
- for all edges $(v, w) \in E$ there is an $i \in I$ with $v, w \in X_i$
- 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 width of a tree decomposition $(\{X_i \mid i \in I\}, T = (I, F))$ is $\max_{i \in I} |X_i| - 1$. The treewidth of a graph $G = (V, E)$ is the minimum width over all tree decompositions of G.

In some cases, T will be considered a *rooted* tree; a specific node of T is considered to be the root. A tree decomposition $(\mathcal{X}, T)$ with T a rooted tree is called a *rooted tree decomposition*. For a node $i \in I$, we call the set X_i the *bag* of i.

Definition. A tree decomposition $(\{X_i \mid i \in I\}, T = (I, F))$ of $G = (V, E)$ is a *domino tree decomposition*, if for each vertex $v \in V$, there are at most two nodes $i \in I$ with $v \in X_i$. The *domino treewidth* of a graph $G = (V, E)$ is the minimum width over all domino tree decompositions of G.

The open neighbourhood of a set of vertices $W \subseteq V$ in a graph $G = (V, E)$ is

$$N(W) = \{v \in V - W \mid \exists w \in W : \{v, w\} \in E\}$$

For a graph $G = (V, E)$, and $W \subseteq V$, the subgraph of G, induced by W is denoted as

$$G[W] = (W, \{\{v, w\} \in E \mid v, w \in W\})$$

Lemma 2.1 *Let $T = (I, F)$ be a tree. Let $J_1 \subseteq I$. Then there exists a set $J_2 \subseteq I$ with*

- $|J_2| \leq 2 \cdot |J_1| - 1$.
- $J_1 \subseteq J_2$.
- *Every subtree of $T[I - J_2]$ is adjacent to at most two nodes in J_2.*

Proof: Choose an arbitrary root r in T.

Let $J_2 = J_1 \cup \{j \in I \mid j$ is the lowest common ancestor of two nodes in $J_1\}$. We claim that this set J_2 fulfils the conditions. Clearly, $J_1 \subseteq J_2$.

Let T' be a subtree of $T[I - J_2]$. If $i_0 \in J_2$ is adjacent to a node in T', then there are two cases:

- i_0 is an ancestor of a node in T'. Then i_0 is the unique parent of the root of T'.
- i_0 is a child of a node in T'. We claim that there can be only one node fulfilling this case (for this tree T'): suppose $i_0 \in J_2$ and $i_1 \in J_2$ are children of nodes in T'. Then, the lowest common ancestor i_2 of i_0 and i_1 belongs to T'. However, i_0 and i_1 belong to J_1 or are ancestor of a node in J_1. So, i_2 is the lowest common ancestor of two nodes in J_1, which is a contraction with the observation that it belongs to T'.

As for T', each case can appear only once, T' is adjacent to at most two nodes in J_2.

To show that $|J_2| \leq 2 \cdot |J_1| - 1$, build a tree $\mathcal{T}$ in the following way: J_2 is the set of nodes in $\mathcal{T}$. If $j \in J_2$ has an ancestor that also belongs to J_2, then take an edge from j to the closest ancestor that also belongs to J_2. One can observe that $\mathcal{T}$ is indeed a tree. Every node $j \in J_2 - J_1$ must have at least two children. So $\mathcal{T}$ is a tree with at most $|J_1|$ leaves, and without nodes with one child. A well know fact about trees tells us that $\mathcal{T}$ has at most $2|J_1| - 1$ nodes, hence $|J_2| \leq 2 \cdot |J_1| - 1$. □

Lemma 2.2 *Let $(\{X_i \mid i \in I\}, T = (I, F))$ be a tree decomposition of $G = (V, E)$. Let $W \subseteq V$, $|W| = r$. Let $s < r$.*

1. *There exists a set of $\lfloor r/(s+1) \rfloor$ nodes $J_1 \subseteq I$, such that each connected component of $G[V - \bigcup_{j \in J_1} X_i]$ contains at most s vertices from W.*

2. *There exists a set of $2\lfloor r/(s+1) \rfloor - 1$ nodes $J_2 \subseteq I$, such that each connected component Z of $G[V - \bigcup_{j \in J_2} X_i]$ contains at most s vertices from W, and for each connected component of $G[V - \bigcup_{j \in J_2} X_i]$ there are nodes $i_1, i_2 \in J_2$, such that every vertex v that is adjacent to a vertex in Z belongs to $Z \cup X_{i_1} \cup X_{i_2}$.*

Proof: 1. First, observe that for any $J \subseteq I$, $G[V - \bigcup_{j \in J} X_i]$ consists of a number of connected components, such that for any connected component Z of $G[V - \bigcup_{j \in J} X_i]$, we have a subtree J_Z of the forest $T[I - J]$ with $Z \subseteq \bigcup_{j \in J_Z} X_j$, i.e., removing J from I splits T in a number of disjoint trees, and each connected component has its vertices in the bags of the nodes in only one of these subtrees.

Choose an arbitrary root $r \in I$, and view T as a rooted tree. We will process T in a bottom-up order: a node is processed after all its children are processed. While processing vertices, we maintain a set $J_1 \subseteq I$, which is initially empty, and a set $W' \subseteq W$, for which initially $W' = W$. The idea is that nodes are added to J_1 until finally the requested set is found, and that W' gives those vertices in W that still can belong to a connected component with too many vertices in W in it.

For a node $i \in I$, let $V_i = \bigcup_{j \in I_i} X_i$, with I_i the set containing i and all its descendants in T.

While processing a node i, compute $Y_i = V_i \cap W'$. If $|Y_i| > s$, put i in J_1, and set $W' = W - Y_i$. Otherwise, nothing is done when processing node i.

We now claim that the set J_1 which is obtained after processing root node r fulfils the requirements of the lemma. Consider a connected component Z of $G[V - \bigcup_{j \in J_1} X_i]$. Let i_Z be the highest node in T whose bag contains a vertex in Z. Clearly, $i_Z \notin J_1$, as $Z \subseteq V - \bigcup_{j \in J_1} X_i$. Hence, when i_Z was processed, $|Y_{i_Z}| \leq s$. Now we note that $Z \cap W \subseteq Y_{i_Z}$: suppose $v \in Z \cap W$. By choice of i_Z, $v \in V_{i_Z}$. If $v \in W - W'$, then v belongs to a bag that is below a node in J_1 or in J_1, and hence either v belongs to $\bigcup_{j \in J_1} X_i$ or is separated from Z by $\bigcup_{j \in J_1} X_i$. This contradicts that $v \in Z$, hence $Z \cap W \subseteq Z \cap W' \subseteq Y_{i_Z}$, and we have $|Z \cap W| \leq |Y_{i_Z}| \leq s$.

To each node $i \in J_1$, we can associate the $s+1$ or more vertices that are removed from W' when i was added to J_1. As each vertex in W is associated with at most one $i \in J_1$, we have $|J_1| \leq \lfloor r/(s+1) \rfloor$.

2. First, obtain a set J_1 as above. As in Lemma 2.1 on the preceding page, create set J_2, such that

- $|J_2| \leq 2\lfloor r/(s+1) \rfloor - 1/$
- $J_1 \subseteq J_1$.
- Every subtree of $T[I - J_2]$ is adjacent to at most two nodes in J_2.

Now, let Z be a connected component of $G[V - \bigcup_{j \in J_2} X_j]$. By the properties of tree decompositions, it follows that there is a subtree T_Z of $T[I - J_2]$, such that all vertices of Z only belong to bags of nodes in T_Z. Thus, the vertices that neighbour a vertex in Z but do not belong to Z must belong to a bag X_i, with $i \in J_2$ one of the at most two nodes in J_2 that are adjacent to T_Z. □

Corollary 2.3 *Let $G = (V, E)$ have treewidth at most k and maximum degree at most d. Let $W \subseteq V$, $|W| \leq r$. Let $s < r$. There exists a set $S \subseteq V$ of at most $(k+1) \cdot (2\lfloor r/(s+1) \rfloor - 1)$ vertices, such every connected component Z in $G[V - S]$ contains at most s vertices from W and at most $(2k+2)d$ vertices that are adjacent to a vertex in S. If k is a constant, such a set S can be found in linear time.*

Proof: The non-algorithmic result follows directly from the previous lemma. (Note that for such a component W, there are at most $2k + 2$ vertices in S adjacent to vertices in W (namely the vertices in at most two bags of the tree decomposition), hence at most $(2k + 2)d$ vertices in W that are adjacent to a vertex in S.) To effectively obtain the set S, first apply the algorithm in [Bod96] to obtain an arbitrary tree decomposition of width at most k. It is not hard to see that the proofs given above then can be carried out in linear time. □

3 The domino treewidth theorem

In this section, we prove the main result of this section. The technique is inspired by a technique from [BGHK95], which was again inspired by a technique from [RS95].

Theorem 3.1 *Let $G = (V, E)$ be a graph with treewidth at most k and maximum degree at most d. Then the domino treewidth of G is at most $(9k + 7)d(d + 1) - 1$.*

Proof:

We first give a recursive procedure, called MAKEDEC, called with two arguments: a graph $H = (V_H, E_H)$ (which is always an induced subgraph of G, and is assumed to have treewidth at most k, and maximum degree at most d), and a set of vertices $W \subseteq V_H$. The procedure outputs a rooted domino tree decomposition of H, $(\{X'_i \mid i \in I'\}, T' = (I', F'))$ of width at most $(9k+7)d(d+1)-1$, such that the vertices in W only belong to the bag of the root node of the domino tree decomposition.

Procedure MAKEDEC (graph $H = (V_H, E_H)$, vertex set W) has the following steps:

1. Obtain a set $S \subseteq V_H$, such that every connected component of $H[V_H - S]$ contains at most $4k + 2$ vertices from W and at most $(2k + 2)d$ vertices that are adjacent to a vertex in S, (as in Corollary 2.3.)

2. Set $R = N(S \cup W)$.

3. Compute the connected components $H_1 = (V_1, E_1), \ldots, H_s = (V_s, E_s)$ of $H[V_H - S - W]$.

4. For each i, $1 \leq i \leq s$, call MAKEDEC(H_i, $V_i \cap R$).

5. Combine the tree decompositions obtained in the previous step in the following way: Take a new node r with $X_r = R \cup S \cup W$. This is the root of the new tree decomposition. Make r adjacent to the roots of each of the tree decompositions, obtained in the previous step. The result is the output of the procedure.

Assume that the set S found in Step 1 is at most of the size, guaranteed to exist by Corollary 2.3 on the preceding page, i.e., we have:

$$|S| \leq (k+1) \cdot (2\lfloor \frac{r}{s+1} \rfloor - 1)$$

Claim 3.1.1 *Let $H = (V_H, E_H)$ be a connected graph, and $W \subseteq V_H$, $W \neq \emptyset$. When* MAKEDEC*(H, W) is called, the procedure outputs a rooted domino tree decomposition of H, such that vertices in W only belong to the root bag of the domino tree decomposition.*

Proof: First, observe that the first parameter of a recursive call to MAKEDEC always is a connected graph, and the second parameter of every recursive call to MAKEDEC is always a non-empty set: every connected component of $H[V_H - S - W]$ must contain vertices adjacent to $S \cup W$. Thus, the recursive calls done to MAKEDEC involve graphs with fewer vertices, hence the procedure terminates.

Let $\{u, v\} \in E_H$. If $\{u, v\} \cap (S \cup W) \neq \emptyset$, then u and v belong both to the root bag X_r. Otherwise, x and y belong to the same connected component H_i of $H[V_H - S - W]$, and by induction, there will be a bag containing both x and y. In both cases, there is a bag in the resulting decomposition that contains both x and y.

Let $v \in V_H$. There are three cases.

If $v \in S \cup W$, then v does not belong to any connected component of $H[V_H - S - W]$, hence v only belongs to bag X_r, and no other bag of the decomposition.

If $v \in R$, then v belongs to X_r. In addition, v belongs to exactly one connected component H_i of $H[V_H - S - W]$. By induction, v belongs to the root bag of the domino tree decomposition yielded by the call of MAKEDEC(H_i,$V_i \cap R$) and no other bag. Thus, v belongs to exactly two bags that are adjacent.

If $v \notin R \cup S \cup W$, then v belongs to exactly one connected component H_i of $H[V_H - S - W]$, and by induction to one or two adjacent bags in the decomposition made by the recursive call to MAKEDEC(H_i,$V_i \cap R$). v does not belong to any other bag.

Hence, the claim follows. □

Claim 3.1.2 *If* MAKEDEC*(H, W) is called with $H = (V_H, E_H)$ a connected graph of maximum degree d and treewidth at most k, and $W \subseteq V_H$ a set of vertices of size at most $(6k+4)d$, then the resulting domino tree decomposition has width at most $(9k+7)d(d+1) - 1$.*

Proof: First, we estimate the size of the root bag of the resulting domino tree decomposition. We have $|W| \leq (6k+4)d$. By Corollary 2.3 on the page before, we can take:

$$|S| \leq (k+1)(2\lfloor (6k+4)d/(4k+3) \rfloor - 1) < 3(k+1)d$$

Now

$$\begin{aligned} |R| &\leq d \cdot |S \cup W| \\ &\leq d \cdot ((6k+4)d + 3(k+1)d) \\ &= (9k+7)d(d+1) \end{aligned}$$

So,

$$|R \cup S \cup W| \leq (9k+7)d(d+1)$$

Secondly, we estimate the size of a set $V_i \cap R$ in a recursive call MAKEDEC(H_i,$V_i \cap R$). Write

$$V_i \cap R = (V_i \cap N(S)) \cup (V_i \cap N(W))$$

Each connected component H_i of $H[V_H - S - W]$ is contained in a connected component H_i' of $H[V_H - S]$. H_i' contains at most $4k+2$ vertices from W, hence at most $(4k+2)d$ vertices of $N(W)$. Also, by construction of S, H_i' contains at most $(2k+2)d$ vertices in $N(S)$. As a consequence,

$$|V_i \cap R| \leq (4k+2)d + (2k+2)d = (6k+4)d$$

Now, we can use induction: each recursive call of MAKEDEC is called with as second parameter a set of size at most $(6k+4)d$, hence the recursive calls give tree decompositions of width at most $(9k+7)d(d+1)-1$, which proves the claim. □

So, from these two claims it follows, that when we call MAKEDEC(G,W) for a connected graph G of treewidth at most k, and maximum degree at most d, and any non-empty vertex subset W which has size at most $(6k+4)d$, we obtain a domino tree decomposition of G of width at most $(9k+7)d(d+1)-1$.

If G is not connected, then make separate domino tree decompositions for each connected component, and connect these to a tree in an arbitrary way. □

The new idea in the proof can be found in step 2 of the procedure MAKEDEC: by adding the neighbours of the vertices in set $S \cup W$ to the root bag of the tree decomposition to make, we do not have to use these vertices at lower levels of the tree decomposition anymore. Apart from this idea, the structure of the algorithm is similar to algorithms found in [RS95, BGHK95].

Corollary 3.2 *Let k be a constant. Given a graph with treewidth at most k and maximum degree at most d, a domino tree decomposition of G of width at most $(9k+7)d(d+1)-1$ can be built in $O(n^2)$ time.*

Proof: Use the procedure, given in the proof above. Excluding the time spent in recursive calls of MAKEDEC, one call of MAKEDEC uses $O(n)$ time. There are $O(n)$ such calls (e.g., every vertex belongs to at most two bags, hence a tree decomposition with $O(n)$ nodes is obtained, and the number of recursive calls of MAKEDEC equals the number of nodes of the resulting tree decomposition), so the total time is bounded by $O(n^2)$. □

4 A lower bound

In this section, we show that a general bound like obtained in the previous section must always be of order $\Omega(kd)$.

We first start with the following lemma, which is also interesting on its own. For a graph $G = (V, E)$, let

$$G^2 = (V, \{\{v,w\} \mid \{v,w\} \in E \vee \exists x \in V : \{v,x\} \in E \wedge \{x,w\} \in E\})$$

Lemma 4.1 *Let $G = (V, E)$ be a graph with domino treewidth at most k. The treewidth of G^2 is at most $2k$.*

Proof: W.l.o.g., suppose G is connected. Let $(\{X_i \mid i \in I\}, T = (I, F))$ be a domino tree decomposition of G of width at most k. Note that (by the properties of tree decompositions and the assumption of connectedness of G) each two adjacent bags intersect. Choose an arbitrary root r. If we add to each bag

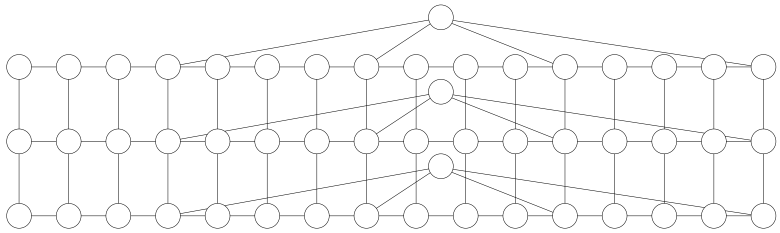

Fig. 1: Grid with added vertices z_i

X_i the bag of the parent of i (unless $i = r$), then we obtain a tree decomposition of G of width at most $2k$. (The union of two bags with a non-empty intersection and with size at most $k+1$ each is taken.)

For every edge $\{v, w\}$ in G^2, we have a bag containing both v and w: this is trivially true if $\{v, w\} \in E$. If v and w have a common neighbour x in G, then either there is a bag X_i containing both v, w, x, or there are two adjacent bags, one containing v and x, and one containing w and x. One must be a child in T (with root r) of the other. Thus, v, w, and x all three belong to a common bag in the constructed tree decomposition. □

A $1/3 - 2/3$-separator of a set W in a graph $G = (V, E)$, is a set of vertices S, such that W can be partitioned into sets W_1, W_2, and W_3, with $W_3 = S \cap W$, $|W_1| \leq 2/3|W|$, $|W_2| \leq 2/3|W|$, and every path from a vertex in W_1 to a vertex in W_2 uses a vertex in S.

The following lemma is well known. See e.g. [BGHK95, GRE84, Liu90, RS86].

Lemma 4.2 *Let $G = (V, E)$ be a graph of treewidth at most k. Let $W \subseteq V$. Then G contains a $1/3 - 2/3$-separator of W of size at most $k+1$.*

Lemma 4.3 *For all $d \geq 5$, $k \geq 2$, k even, there exists a graph G with treewidth at most k, maximum degree at most d, and domino treewidth at least $\frac{1}{12}kd - 2$.*

Proof: Consider the following graph.

First, we take a grid of size $k/2$ by d^2k. I.e., we have vertices of the form $v_{i,j}$, $1 \leq i \leq k/2$, $1 \leq j \leq d^2k$, and $v_{i,j}$ is adjacent to $v_{i',j'}$, iff $|i - i'| + |j - j'| = 1$. To this grid, we add $k/2$ additional vertices $z_1, \ldots, z_{k/2}$, with, for each i, $1 \leq i \leq k/2$, z_i adjacent to each vertex $v_{i,j \cdot dk}$, $1 \leq j \leq d$. Let $G = (V, E)$ be the resulting graph.

See Figure 1 for an illustration of the construction. (In order to make the figure not too large, the distance between successive neighbours of the vertices z_i is 4 in the figure, instead of dk.)

The maximum degree of G is $\max(5, d)$: vertices of the form $v_{i,j}$ have degree at most five, while vertices of the form z_i have degree d. It is also not hard to see that the treewidth of G is at most k. The $k/2$ by d^2k grid graph has treewidth exactly $k/2$ (see e.g. [Bod98].) As G contains $k/2$ vertices such that when these are deleted from G, G becomes a graph of treewidth $k/2$, the treewidth of G is at most k. (See e.g. [Bod98], Lemma 72.)

Call the ith row the set of all vertices of the form $v_{i,j}$, $1 \leq j \leq d^2k$. Similar, the set of all vertices of the form $v_{i,j}$, $1 \leq i \leq k/2$ is called the jth column.

Now, we claim that G^2 has treewidth at least $\frac{1}{6}dk - 1$. Note that all vertices in the ith row that were adjacent to z_i form a clique in G^2. Call the set of these vertices the ith row-clique. Let

$$W = \{v_{i,j} \mid 1 \leq i \leq k/2,\ 1 \leq j \leq kd^2\}$$

I.e., W is the set of the grid vertices in G.

Suppose S is a $1/3 - 2/3$-separator of W of minimum size in G^2, partitioning W into W_1, W_2, $W_3 = S \cap W$.

We will now show that $|S| \geq \frac{1}{6}dk$. Assume $|S| < \frac{1}{6}dk$.

Note that $|W_1| \leq \frac{2}{3} \cdot d^2k^2/2$, hence $|W_2 \cap W_3| \geq \frac{1}{6}d^2k^2$, and likewise $|W_1 \cap W_3| \geq \frac{1}{6}d^2k^2$.

Every column that contains both a vertex in W_1 and a vertex in W_2 must also contain a vertex in S. Thus, we may assume there are fewer than $\frac{1}{6}dk$ such columns. So, fewer than $\frac{1}{12}dk^2$ vertices in W can belong to such a column. It follows that there are at least $(\frac{1}{6}d^2k^2 - \frac{1}{12}dk^2)/(k/2) = \frac{1}{3}d^2k - \frac{1}{6}dk$ columns that only contain vertices in W_1, and thus, every row contains $\frac{1}{3}d^2k - \frac{1}{6}dk$ vertices in W_1. Likewise, every row contains $\frac{1}{3}d^2k - \frac{1}{6}dk$ in W_2.

We now will show that every row contains at least $\frac{1}{3}dk$ vertices in S.

Consider the ith row. Note that either all vertices in the ith row-clique belong to $W_1 \cup S$ or all vertices in the ith row-clique belong to $W_2 \cup S$. Without loss of generality, we suppose the former; the other case is identical.

We partition the vertices in the ith row in d *intervals*, where the mth interval contains vertices $v_{i,(m-1)dk+1}, v_{i,(m-1)dk+2}, \ldots, v_{i,mdk}$. At least $\lceil(\frac{1}{3}d^2k - \frac{1}{6}dk)/(dk)\rceil \geq \frac{1}{3}d$ of these intervals must contain vertices in W_2. However, each interval also contains a vertex in the ith row-clique, hence it contains a vertex in $S \cup W_1$. So, each interval that contains a vertex in W_2 must contain a vertex in S, hence the ith row contains at least $\frac{1}{3}d$ vertices in S.

As we have $k/2$ rows, it follows that $|S| \geq \frac{1}{6}dk$. By Lemma 4.2 on the page before, we have that the treewidth of G^2 is at least $\frac{1}{6}dk - 1$, hence by Lemma 4.1 on page 146, the domino treewidth of G is at least $\frac{1}{12}dk - 2$. □

5 Final remarks

It is possible to give a modified version of the procedure of Corollary 3.2 on page 146, that yields domino tree decompositions of somewhat larger width (but still of $O(kd^2)$, but that uses $O(n \log n)$ time instead of $O(n^2)$ time. However, the proof in [BE97] can be turned into an algorithm that uses linear time. It is not known how much time a procedure based upon the proof by Ding and Oporowski [DO95] would take.

The proof given in this paper seems unable to yield linear time algorithms - the approach typically leads to algorithmic results of $\Omega(n \log n)$ time. It is open whether domino tree decompositions of $O(kd^2)$ width can be obtained with a linear time algorithm.

Another interesting open problem is whether a bound of $O(k^2d)$ can essentially be improved. It would be interesting to see if better bounds, e.g., a bound of $O(kd)$ can be proved, and whether better lower bounds are possible.

In some special cases, better bounds can be obtained. For instance, for trees we have the following easy result.

Theorem 5.1 *The domino treewidth of a tree is at most its maximum degree.*

Proof: Let T be a tree with maximum degree d. Choose an arbitrary root r, and view T as a rooted tree. Let $T' = (V', E')$ be the tree, obtained by removing all leaves from T. Consider the following tree decomposition of T: $(\{X_v \mid v \in V'\}, T')$, where each set X_v consists of v and all children of v in T. One easily verifies that this is a domino tree decomposition of T with width at most d. $\Box$

So for trees (and similarly for forests), the domino treewidth is linear in its degree. (Note also that the domino treewidth of a graph with maximum degree $d \geq 1$ is at least $\lceil (d+1)/2 \rceil - 1$: at most two bags can contain a vertex of degree d and all its neighbours.) It seems interesting to see if it is also possible to obtain similar bounds for other restricted classes of graphs of bounded treewidth, e.g., series parallel graphs, Halin graphs, or arbitrary graphs of treewidth two.

References

[BE97] Hans L. Bodlaender and Joost Engelfriet. Domino treewidth. *J. Algorithms*, 24:94–123, 1997.

[BGHK95] Hans L. Bodlaender, John R. Gilbert, H. Hafsteinsson, and Ton Kloks. Approximating treewidth, pathwidth, and minimum elimination tree height. *J. Algorithms*, 18:238–255, 1995.

[Bod96] Hans L. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25:1305–1317, 1996.

[Bod98] Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theor. Comp. Sc.*, 209:1–45, 1998.

[DO95] Guoli Ding and Bogdan Oporowski. Some results on tree decomposition of graphs. *J. Graph Theory*, 20:481–499, 1995.

[GRE84] John R. Gilbert, Donald J. Rose, and Anders Edenbrandt. A separator theorem for chordal graphs. *SIAM J. Alg. Disc. Meth.*, 5:306–313, 1984.

[Liu90] J. W. H. Liu. The role of elimination trees in sparse factorization. *SIAM J. Matrix Analysis and Applications*, 11:134–172, 1990.

[RS86] Neil Robertson and Paul D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *J. Algorithms*, 7:309–322, 1986.

[RS95] Neil Robertson and Paul D. Seymour. Graph minors. XIII. The disjoint paths problem. *J. Comb. Theory Series B*, 63:65–110, 1995.