Bounding connected tree-width

Matthias Hamann       Daniel Weißauer

Department of Mathematics
University of Hamburg

Abstract

Diestel and Müller showed that the connected tree-width of a graph $G$, i.e., the minimum width of any tree-decomposition with connected parts, can be bounded in terms of the tree-width of $G$ and the largest length of a geodesic cycle in $G$. We improve their bound to one that is of correct order of magnitude.

1 Introduction

Intuitively, a tree-decomposition $(T, (V_t)_{t \in T})$ of a graph $G$ can be regarded as giving a bird’s-eye view on the global structure of the graph, represented by $T$, while each part represents local information about the graph. But this interpretation can be misleading: the tree-decomposition may have disconnceted parts, containing vertices which lie at great distance in $G$, and so this intuitively appealing distinction between local and global structure can not be maintained.

This can be remedied if we require every part to be connected. We call such a tree-decomposition connected. Jegou and Terrioux [5, 6] pointed out that the efficiency of algorithmic methods based on tree-decompositions for solving constraint satisfaction problems can be improved when using connected tree-decompositions.

The connected tree-width $\text{ctw}(G)$ is defined accordingly as the minimum width of a connected tree-decomposition of the graph $G$. Trivially, the connected tree-width of a graph is at least as large as its tree-width and, as Jegou and Terrioux [6] observed, long cycles are examples of graphs of small
Theorem 1 ([2, Theorem 1.1]). There is a function \( f : \mathbb{N}^2 \to \mathbb{N} \) such that the connected tree-width of any graph of tree-width \( k \) and without geodesic cycles of length greater than \( \ell \) is at most \( f(k, \ell) \).

They also showed that \( f(k, \ell) = O(k^3 \ell) \). In fact, their proof does not only work with geodesic cycles, but with any collection of cycles that generate the cycle space of the graph \( G \). Given a graph \( G \), we define \( \ell(G) \) to be the smallest natural number \( \ell \) such that the cycles of length at most \( \ell \) generate the cycle space of \( G \). Our main result improves the bound of Diestel and Müller significantly:

**Theorem 2.** Let \( G \) be a graph containing a cycle. Then the connected tree-width of \( G \) is at most \( \text{tw}(G)(\ell(G) - 2) \).

(Observe that every forest satisfies \( \text{ctw}(G) = \text{tw}(G) \leq 1 \).) Theorem 2 will be proved in Sections 2–4. In Section 6 we discuss an example that demonstrates that this is best possible up to a constant factor.

The tree-width duality theorem of Seymour and Thomas [7] asserts that a graph has tree-width less than \( k \) if and only if it has no bramble of order at least \( k \). Diestel and Müller [2] conjectured that a similar duality holds for connected tree-width and showed [2, Theorem 1.2] that the connected tree-width of a graph can be bounded by a function of the maximum connected order of its brambles. Here, the connected order of a bramble is the minimum size of a connected vertex set meeting every element of the bramble. In Section 5, we employ techniques and results from the previous sections to prove the following upper bound on the connected order of brambles:

**Theorem 3.** Let \( G \) be a graph containing a cycle. Then the connected order of any bramble of \( G \) is at most \( \text{tw}(G)\left\lfloor \frac{\ell(G)}{2} \right\rfloor + 1 \).

Note that if the duality conjecture for connected tree-width is true, the bound in Theorem 3 immediately improves that of Theorem 2.
2 Definitions and notation

For a tree $T$ with root $r$, we call $s$ a descendant of $t$ and $t$ an ancestor of $s$ if $t$ lies on the unique path from $r$ to $s$. If additionally $st \in E(T)$, we call $s$ a child of $t$ and $t$ the parent of $s$. We write $T_t$ for the subtree of descendants of $t$. Recall that a tree-decomposition of $G$ is a pair $(T, \mathcal{V})$ of a tree $T$ and a family $\mathcal{V} = (V_t)_{t \in T}$ of vertex sets $V_t \subseteq V(G)$, one for every node of $T$, such that:

(T1) $V(G) = \bigcup_{t \in T} V_t$,

(T2) for every edge $e$ of $G$ there exists a $t \in T$ with $e \subseteq V_t$,

(T3) $V_{t_1} \cap V_{t_3} \subseteq V_{t_2}$ whenever $t_2$ lies on the $t_1$–$t_3$ path in $T$.

The sets $V_t$ in such a tree-decomposition are its parts. For $A \subseteq V(T)$ we write $V_A := \bigcup_{t \in A} V_t$. The width of $(T, \mathcal{V})$ is $\max_{t \in T} (|V_t| - 1)$ and the tree-width $\text{tw}(G)$ of $G$ is the minimum width of any of its tree-decompositions.

In our proof of Theorem 2 we will make use of an explicit procedure that transforms a given tree-decomposition into a connected tree-decomposition by iteratively adding paths to a disconnected part of the decomposition. For this to work efficiently, we will restrict ourselves to paths of a particular kind.

Let $(T, \mathcal{V})$ be a rooted tree-decomposition of $G$, i.e. $T$ is rooted, and $t \in T$. A path $P$ in $G$ is $t$-admissible if it lies entirely in $V_t$, joins different components of $V_t$ and is shortest possible with these properties. Note that $t$-admissible paths have precisely two vertices in $V_t$:

**Lemma 4.** Let $(T, \mathcal{V})$ be a rooted tree-decomposition of a graph $G$, $t \in T$ and $P$ a $t$-admissible path. Then there is a unique child $s$ of $t$ such that all internal vertices of $P$ lie in $V_{Ts} \setminus V_t$. 

In general, $t$-admissible paths need not exist. However, as we shall see, we can easily confine ourselves to tree-decompositions that always have $t$-admissible paths.

We call a tree-decomposition $(T, \mathcal{V})$ stable if for every edge $t_1t_2 \in E(T)$ of $T$, both $V_{T_1}$ and $V_{T_2}$ are connected in $G$, where $T_i$, for $i = 1, 2$, is the component of $T - t_1t_2$ containing $t_i$. (Later, we will use this naming convention without further mention.)

**Lemma 5.** Let $(T, \mathcal{V})$ be a rooted stable tree-decomposition of a connected graph $G$. Then every $t \in T$ with disconnected $V_t$ has a $t$-admissible path.


Stable tree-decompositions were also studied in [3], where they are called connected tree-decompositions. In that article, an explicit algorithm is presented that turns a tree-decomposition of a connected graph into a stable tree-decomposition without increasing its width. For our purposes it suffices to know that every connected graph has a stable tree-decomposition of minimum width. This can also be deduced from [2, Corollary 3.5].

**Proposition 6.** Every connected graph $G$ has a stable tree-decomposition of width $tw(G)$. }

If we add a $t$-admissible path $P$ to a part $V_t$ in order to join two of its components, we might not obtain a tree-decomposition. The following lemma shows how it can be patched.

**Lemma 7.** Let $(T, V)$ be a rooted tree-decomposition of a graph $G$, $t \in T$ and $P$ a $t$-admissible path. For $u \in T$ let

$$W_u := \begin{cases} V_u \cup (V(P) \cap V_{T_u}), & \text{if } u \in T_t, \\ V_u, & \text{if } u \notin T_t. \end{cases} \quad (\ast)$$

Then $(T, W)$ is a tree-decomposition of $G$. For all $u \in T$, every component of $W_u$ contains a vertex of $V_u$. If $(T, V)$ is stable, so is $(T, W)$.

**Proof.** Since $V_u \subseteq W_u$ for all $u \in T$, every vertex and every edge of $G$ is contained in some part $W_u$.

Let $I$ be the set of internal vertices of $P$. By Lemma 4 there is a unique child $s$ of $t$ such that $I \subseteq V_{T_s} \setminus V_t$. For $x \notin I$, the set of parts containing $x$ has not changed. For $x \in I$, the set $A_x := \{u \in T : x \in V_u\}$ induces a subtree of $T_s$ and $x \in W_u$ if and only if $u \in A_x$ or $u$ lies on the path joining $t$ to $A_x$. So \{ $u : x \in W_u$ \} is also a subtree of $T$.

Note that every component of $P \cap V_{T_u}$ is a path with ends in $V_u$. Therefore every $x \in W_u \setminus V_u$ is joined to two vertices in $V_u$ and thus every component of $W_u$ contains vertices from $V_u$.

Suppose now $(T, V)$ is stable, let $t_1t_2 \in E(T)$ and $i \in \{1, 2\}$. Then $V_{T_i}$ is connected. For $x \in W_{T_1} \setminus V_{T_i}$ there is a $u \in T_i \cap T_t$ with $x \in W_u \setminus V_u$. But then, by the above, $W_u$ contains a path joining $x$ to $V_{T_i}$. As $V_{T_i} \subseteq W_{T_i}$, also $W_{T_i}$ is connected. 

\[ \square \]
3 The construction

We now describe a construction that turns a stable tree-decomposition \((T, V)\) of a connected graph into a connected tree-decomposition. First, choose a root \(r\) for \(T\) and keep it fixed. It will be crucial to our analysis that the nodes of \(T\) are processed in the induced order of the tree, i.e. we enumerate the nodes \(t_1, t_2, \ldots\) so that each node precedes its descendants and we process the nodes in this order.

Initially we set \(W_t = V_t\) for all \(t \in T\). Throughout the construction, we maintain the invariant that \((T, W)\) is a stable tree-decomposition extending \((T, V)\), by which we mean that they are tree-decompositions over the same rooted tree, satisfying \(V_t \subseteq W_t\) for all \(t \in T\).

When processing a node \(t \in T\) with disconnected part \(W_t\), we use the stability of \((T, W)\) to find a \(t\)-admissible path by Lemma 5 and update \(W\) as in \((\ast)\). By Lemma 7, this does not violate stability and it clearly reduces the number of components of \(W_t\) by one. We iterate this until \(W_t\) is connected. Once that is achieved, we continue with the next node in our enumeration.

Observe that each ‘update’ only affects descendants of the current node. Once a node \(t \in T\) has been processed, so have all of its ancestors. Hence, no further changes are made to \(W_t\) afterwards. In particular, \(W_t\) remains connected. It thus follows that, when every node has been processed, the resulting tree-decomposition is indeed connected.

In order to control the size of each part \(W_u\), we will use a bookkeeping graph \(Q_u\) to keep track of what we have added. Initially, \(Q_u\) is the empty graph on \(V_u\), and in each step \(Q_u\) is a graph on the vertices of \(W_u\). Whenever something is added to \(W_u\), we are considering a \(t\)-admissible path \(P\) for some ancestor \(t\) of \(u\) and \(P\) contains vertices of \(W_{T_u}\). Every component of \(P \cap W_{T_u}\) is a path with ends (and possibly also some internal vertices) in \(W_u\). We then add \(P \cap W_{T_u}\) to \(Q_u\), that is, we add all the vertices not contained in \(W_u\) and all the edges of \(P \cap W_{T_u}\).

**Lemma 8.** During every step of the procedure, \(Q_u\) is acyclic.

**Proof.** This is certainly true initially. Suppose now that at some step a cycle is formed in \(Q_u\). By definition, it must be that an ancestor \(t\) of \(u\) is being processed and a \(t\)-admissible path \(P\) is added such that two vertices \(a, b \in W_u\) which were already connected in \(Q_u\) lie in the same component of \(P \cap W_{T_u}\).

The vertices \(a, b\) being connected in \(Q_u\) by a path \(a = a_0a_1 \ldots a_n = b\) means that there have been, for every \(0 \leq j \leq n - 1\), ancestors \(t_j\) of \(u\) that
added paths $P_j$ such that $a_j, a_{j+1}$ were consecutive vertices on a segment $S_j$ of $P_j \cap W_{T_u}$. By the order in which the nodes are processed and by $(\ast)$, these $t_j$ are also ancestors of $t$. Therefore when $P_j$ was added to $W_{t_j}$, the segment $S_j$ was contained in a segment of $P_j \cap W_{T_i}$, since $W_{T_i} \supseteq W_{T_u}$. Therefore, at the time $P$ is added to $W_t$, all these segments are contained in $W_t$ and, in particular, $a, b \in W_t$. By Lemma 4, $P$ does not have internal vertices in $W_t$ so that $a$ and $b$ must in fact be the ends of $P$. But $W_t$ already contains a walk from $a$ to $b$, consisting of the segments $S_j$, so that the two do not lie in different components of $W_t$, contradicting the $t$-admissibility of $P$. \qed

We now show how the sparse structure of $Q_u$ reflects the efficiency of our procedure.

**Lemma 9.** The number of components of $Q_u$ never increases. Whenever something is added to $W_u$, the number of components of $Q_u$ decreases.

**Proof.** Suppose that in an iteration a change is made to $Q_u$. Then an ancestor $t$ of $u$ is being processed and the chosen path $P$ meets $W_{T_u}$. Every component of $P \cap W_{T_u}$ is a path with both ends in $W_u$. Therefore, every newly introduced vertex is joined to a vertex in $Q_u$ and no new components are created.

If a vertex from $P \cap (W_{T_u} \setminus W_u)$ is added to $W_u$, the segment containing it has length at least two and has two ends $a, b \in Q_u$. By Lemma 8, $Q_u$ must remain acyclic, so that $a$ and $b$ in fact lie in different components of $Q_u$, which are now joined. \qed

The previous lemma allows us to control the number of iterations that affect a fixed node $t \in T$. The second key ingredient for the proof of Theorem 2 will be to bound the length of each of the paths used, see Section 4.

**Proposition 10.** Let $G$ be a connected graph, $(T, \mathcal{V})$ a rooted stable tree-decomposition of $G$. For $t \in T$ let $m_t \geq 1$ be such that for every stable tree-decomposition $(T, \mathcal{W})$ extending $(T, \mathcal{V})$ and every ancestor $t'$ of $t$, the length of a $t'$-admissible path in $(T, \mathcal{W})$ does not exceed $m_t$. Then the construction produces a connected tree-decomposition $(T, \mathcal{U})$ in which for all $t \in T$

$$|U_t| \leq m_t(|V_t| - 1) + 1.$$ 

**Proof.** We have already shown that $(T, \mathcal{U})$ is connected. By Lemma 9, every time something was added to $W_t$, the number of components of $Q_t$ decreased and it never increased. Since initially $Q_t$ had precisely $|V_t|$ components, this
can only have happened at most \(|V_t| - 1\) times. In each such iteration we added some internal vertices of a \(t\)'-admissible path in a stable tree-decomposition extending \((T, V)\) for some ancestor \(t\)' of \(t\), thus at most \(m_t - 1\) vertices. In total, we have
\[
|U_t| \leq |V_t| + (m_t - 1)(|V_t| - 1) = m_t(|V_t| - 1) + 1. \tag*{\blacksquare}
\]

4 Bounding the length of admissible paths

We will now use ideas from [2] to bound the length of \(t\)-admissible paths in stable tree-decompositions. Together with Proposition 10, this will imply our main result.

Lemma 11. Let \(G\) be a graph and \(\Gamma\) a set of cycles that generates its cycle space. Let \((T, V)\) be a stable tree-decomposition of \(G\) and \(t_1t_2 \in E(T)\). Suppose that \(V_{t_1} \cap V_{t_2}\) meets two distinct components of \(V_{t_1}\). Then there is a cycle \(C \in \Gamma\) such that some component of \(C \cap V_{t_2}\) meets \(V_{t_1}\) in two distinct components.

Proof. As \(V_{t_2}\) is connected, we can choose a shortest path \(P\) in \(V_{t_2}\) joining two components of \(V_{t_1}\). Let \(x, y \in V_{t_1}\) be its ends and note that all internal vertices of \(P\) lie in \(V_{t_2} \setminus V_{t_1}\). As \(V_{t_1}\) is connected as well, we also find a path \(Q \subseteq V_{t_1}\) joining \(x\) and \(y\), which is internally disjoint from \(P\). By assumption, there is a subset \(C\) of \(\Gamma\) such that \(P + Q = \bigoplus C\). We subdivide \(C\) as follows: \(C_1\) comprises all those cycles which are entirely contained in \(V_{t_1} \setminus V_{t_2}\), \(C_2\) those in \(V_{t_2} \setminus V_{t_1}\) and \(C_X\) those that meet \(X := V_{t_1} \cap V_{t_2}\).

Assume now for a contradiction that for every \(C \in C_X\) and every component \(S\) of \(C \cap V_{t_2}\) there is a unique component \(D_S\) of \(V_{t_1}\) met by \(S\). Note that \(S\) is a cycle if \(C \subseteq V_{t_2}\) and a path with ends in \(X\) otherwise. Either way, the number of edges of \(S\) between \(X\) and \(V_{t_2} \setminus X\), denoted by \(|E_S(X, V_{t_2} \setminus X)|\), is always even. It thus follows that for any component \(D\) of \(V_{t_1}\)
\[
|E_C(D, V_{t_2} \setminus X)| = \sum_{S \subseteq C \cap V_{t_2}} |E_S(D, V_{t_2} \setminus X)| = \sum_{S : D_S = D} |E_S(X, V_{t_2} \setminus X)|
\]
is even. But then also the number of edges in \(\bigoplus C_X\) between \(D\) and \(V_{t_2} \setminus X\) is even. Since the edges of \(\bigoplus C_1\) and \(\bigoplus C_2\) do not contain vertices from \(X\), we have
\[
E_{\bigoplus C_X}(X, V_{t_2} \setminus X) = E_{P+Q}(X, V_{t_2} \setminus X) = \{xx', yy'\},
\]
where \( x' \) and \( y' \) are the neighbours of \( x \) and \( y \) on \( P \), respectively. Due to parity, \( x \) and \( y \) need to lie in the same component of \( V_{t_1} \), contrary to definition.

**Proof of Theorem 2.** Since both parameters appearing in the bound do not increase when passing to a component of \( G \) and as we can combine connected tree-decompositions of the components to obtain a connected tree-decomposition of \( G \), it suffices to consider the case that \( G \) is connected.

We use Lemma 11 to bound the length of \( t \)-admissible paths in any stable tree-decomposition of \( G \). Let \( \ell = \ell(G) \) and \( \Gamma \) be the set of all cycles of length \( \ell \), which by definition generates the cycle space of \( G \). Let \((T, W)\) be a rooted stable tree-decomposition, \( t \in T \) and \( P \) a \( t \)-admissible path. By Lemma 4 there is a child \( s \) of \( t \) such that all internal vertices of \( P \) lie in \( V_T \setminus V_t \). By Lemma 11 we find a cycle \( C \in \Gamma \) and a path \( S \subseteq C \cap V_T \) joining distinct components of \( V_{t_1} \). Since \( S \subseteq V_T \) and \( P \) was chosen to be a shortest such path, we have \( |P| \leq |S| \). The ends of \( S \) lie in distinct components of \( V_{t_1} \) and are therefore, in particular, not adjacent, so that overall

\[
|V(P)| \leq |V(S)| \leq |V(C)| - 1 \leq \ell - 1.
\]

By Proposition 6, \( G \) has a stable tree-decomposition \((T, V)\) of width \( tw(G) \). Proposition 10 then guarantees that we find a connected tree-decomposition of width at most \((\ell - 2)tw(G)\).

**5 Brambles**

Recall that a *bramble* is a collection of connected vertex sets of a given graph such that the union of any two of them is again connected. A *cover* of a bramble is a set of vertices that meets every element of the bramble. The aim of this section is to derive a strengthened upper bound on the *connected order* of a bramble, the minimum size of a connected cover.

**Lemma 12.** Suppose \((T, V)\) is a tree-decomposition of a graph \( G \) and \( k \in \mathbb{N} \) an integer such that for every \( t \in T \) there is a connected set of size at most \( k \) containing \( V_t \). Then \( G \) has no bramble of connected order greater than \( k \).

**Proof.** Let \( \mathcal{B} \) be a bramble of \( G \). By a standard argument, see e.g. the proof of [1, Theorem 12.3.9], one of the parts \( V_t \) of \((T, V)\) covers \( \mathcal{B} \) and thus so does any connected set containing \( V_t \). \( \square \)
Note that the assumption of Lemma 12 is at least a priori weaker than assuming that $G$ has connected tree-width less than $k$: The connected sets containing the $V_i$ do not necessarily induce a tree-decomposition of $G$. If the conjectured duality between connected tree-width and connected order of a bramble does hold, however, Lemma 12 would imply that the two assumptions are in fact equivalent.

**Proof of Theorem 3.** It suffices to consider the case where $G$ is connected, since every bramble can only contain vertex sets of one component. Let $\ell = \ell(G)$ and $\Gamma$ be the set of all cycles of $G$ of length at most $\ell$, which by definition generates the cycle space of $G$. By Proposition 6, $G$ has a stable tree-decomposition $(T, V)$ of width $\text{tw}(G)$. We now show that every part $V_i$ of $(T, V)$ is contained in a connected set of size at most $(|V_i| - 1)\left\lfloor \frac{\ell}{2} \right\rfloor + 1$. The statement then follows from Lemma 12.

Let now $t \in T$ be fixed. Root $T$ at $t$ and apply the construction from Section 3. As $t$ does not have any ancestors other than itself, the statement follows from Proposition 10 once we have verified that all $t$-admissible paths in a stable tree-decomposition $(T, W)$ extending $(T, V)$ have length at most $\ell/2$. So let $(T, W)$ be a stable tree-decomposition of $G$ extending $(T, V)$ and let $P$ be a $t$-admissible path. By Lemma 4, all its internal vertices lie in $W_{T_s} \setminus W_t$ for some child $s$ of $t$. By Lemma 11 we find a cycle $C \in \Gamma$ that meets $W_t$ in two vertices $x, y$ from distinct components of $W_t$. Either segment of $C$ between $x$ and $y$ lies in $W_{T_t}$ and joins two components of $W_t$, so by minimality $P$ has length at most $\left\lfloor \ell/2 \right\rfloor$. \hfill $\square$

Diestel and Müller [2] showed that if a graph $G$ contains a geodesic cycle of length $k$, then $G$ has a bramble of connected order $\lceil k/2 \rceil + 1$. Combined with the upper bound of Theorem 3, this implies that the cycle space of $G$ can not be generated by cycles of length less than $k/\text{tw}(G)$.

### 6 Example

In this section we discuss an example that shows that our bound is tight up to a constant factor. Observe first that if $H$ is a subdivision of $G$, then $\text{tw}(H) = \text{tw}(G)$. Given $n, k \in \mathbb{N}$, $n \geq 3$, let $G$ be obtained from the complete graph on $n$ vertices by subdividing every edge with $k$ newly introduced vertices. By the above, $G$ has tree-width $n - 1$. The cycle space of $G$ is generated by the set of all subdivisions of triangles of the underlying complete graph, so
\( \ell(G) = 3(k + 1) \). We will now show that the connected tree-width of \( G \) is precisely \( (n - 1)(k + 1) - \lceil (k + 1)/2 \rceil \).

Let \( A \subseteq V(G) \) denote the set of vertices of degree \( n - 1 \). The graph \( G \) thus consists of \( A \) and, for every pair \( a, b \in A \), a path \( P_{ab} \) of length \( k + 1 \) between them. We begin by proving the upper bound. Let \( (T, \mathcal{V}) \) be a connected tree-decomposition of \( G \). For \( t \in T \), let \( A_t := V_t \cap A \). Let \( u \in T \) be a node with maximal set \( A_u \). If \( |A_u| = n \), then \( V_u \), being connected, must contain at least \( n + (n - 1)k \) vertices, even more than we claimed. So now assume this is not the case and let \( a \in A \setminus A_u \). Choose \( s, t \in T \) at minimal distance such that \( A_u \subseteq V_t \) and \( a \in V_s \). Note that \( A_t = A_u \) and by maximality there is some \( b \in A_t \setminus A_u \). Assume for a contradiction that \( s, t \) were not neighbours in \( T \) and let \( z \in sTt \) be an intermediate node. By choice of \( s, t \) there is some \( c \in A_t \setminus A_u \) and \( a \notin A_u \). Since \( V_z \) separates \( V_s, V_t \) it must contain a vertex from \( P_{ac} \). Since \( V_z \) is connected, it must consist of internal vertices of \( P_{ac} \). But then \( G - V_z \) still contains the \( a-c \) path \( P_{ab} \cup P_{bc} \), which is a contradiction.

Since \( V_s \cap V_t \) contains \( a \) and \( b \), for every \( c \in A^- := A \setminus \{a, b\} \), the separator \( V_s \cap V_t \) contains a vertex from \( P_{ac} \cup P_{cb} \) and it follows that \( A \subseteq V_s \cup V_t \). For every \( c \in A^- \setminus A_t \) we must have \( P_{bc} \subseteq V_s \cup V_t \) and at least one of its internal vertices must lie in both. Furthermore \( V_t \) is connected and thus contains a tree covering \( A_t \) and avoiding all internal vertices of paths \( P_{bc} \) with \( c \in A^- \setminus A_t \). The same is true when we exchange the roles of \( s \) and \( t \). Since \( P_{ab} \subseteq V_s \cup V_t \), simple counting yields

\[
|V_s| + |V_t| \geq \begin{aligned}
(k + 1)|A_u| - k + (k + 1)|A^- \setminus A_u| \\
+ (k + 1)|A_t| - k + (k + 1)|A^- \setminus A_t| \\
+ k + 1 \end{aligned} = 2(n - 1)(k + 1) - k + 1.
\]

We now describe a connected tree-decomposition realising the width. Fix two \( a, b \in A \) and let \( A^- := A \setminus \{a, b\} \). Let \( T \) be a star with root \( s \) and leaves \( t, u_1, \ldots, u_m \) with \( m = \binom{n - 2}{2} \). Each \( V_{u_i} \) consists of a different path \( P_{cd} \) with \( c, d \in A^- \). Let \( V_s \) consist of the union of all \( P_{bc} \) with \( c \in A^- \) and the first \( \lceil (k + 1)/2 \rceil \) vertices from \( P_{ba} \). Define \( V_t \) similarly. This tree-decomposition has the desired width.
References

[1] R. Diestel. *Graph Theory*. Springer, 4th edition, 2010.

[2] R. Diestel and M. Müller. Connected tree-width. arXiv:1211.7353, 2014.

[3] P. Fraigniaud and N. Nisse. Connected treewidth and connected graph searching. In *Proceedings of the 7th Latin American Conference on Theoretical Informatics*, LATIN’06, pages 479–490. Springer-Verlag, 2006.

[4] A. Georgakopoulos and P. Sprüssel. Geodetic topological cycles in locally finite graphs. *Electronic J. Comb.*, 16:#144, 2009.

[5] P. Jegou and C. Terrioux. Bag-connected tree-width: a new parameter for graph decomposition. http://www.cs.uic.edu/pub/Isaim2014/WebPreferences/ISAIM2014_Jegou_Terrioux.pdf, 2014.

[6] P. Jegou and C. Terrioux. Tree-decompositions with connected clusters for solving constraint networks. In B. O’Sullivan, editor, *CP 2014*, volume 8656, pages 407–403. Springer Lecture Notes in Computer Science, 2014.

[7] P. Seymour and R. Thomas. Graph searching and a min-max theorem for tree-width. *J. Combin. Theory (Series B)*, 58(1):22–33, 1993.