

Parameterized Complexity of the Spanning Tree Congestion Problem

Hans L. Bodlaender · Fedor V. Fomin ·
Petr A. Golovach · Yota Otachi ·
Erik Jan van Leeuwen

Received: 29 October 2010 / Accepted: 23 August 2011

© The Author(s) 2011. This article is published with open access at Springerlink.com

Abstract We study the problem of determining the *spanning tree congestion* of a graph. We present some sharp contrasts in the parameterized complexity of this problem. First, we show that on apex-minor-free graphs, a general class of graphs containing planar graphs, graphs of bounded treewidth, and graphs of bounded genus, the problem to determine whether a given graph has spanning tree congestion at most k can be solved in linear time for every fixed k . We also show that for every fixed k and d the problem is solvable in linear time for graphs of degree at most d . In contrast, if we allow only one vertex of unbounded degree, the problem immediately becomes NP-complete for any fixed $k \geq 8$. Moreover, the hardness result holds for graphs excluding the complete graph on 6 vertices as a minor. We also observe that for $k \leq 3$ the problem becomes polynomially time solvable.

Extended abstract of some results in this paper appeared in the proceedings of WG 2010 [40].
Y. Otachi belongs to JSPS Research Fellow.

H.L. Bodlaender

Department of Information and Computing Sciences, Utrecht University, P.O. Box 80.089, 3508 TB
Utrecht, The Netherlands
e-mail: hansb@cs.uu.nl

F.V. Fomin (✉) · E.J. van Leeuwen

Department of Informatics, University of Bergen, P.O. Box 7803, 5020 Bergen, Norway
e-mail: fomin@ii.uib.no

E.J. van Leeuwen

e-mail: E.J.van.Leeuwen@ii.uib.no

P.A. Golovach

School of Engineering and Computing Sciences, Durham University, South Road,
Durham, DH1 3LE, UK
e-mail: petr.golovach@durham.ac.uk

Y. Otachi

Graduate School of Information Sciences, Tohoku University, Sendai 980-8579, Japan
e-mail: otachi@dais.is.tohoku.ac.jp

Keywords Spanning tree congestion · Graph minor · Parameterized algorithms · Apex graph

1 Introduction

Spanning tree congestion is a relatively new graph parameter, which was formally defined by Ostrovskii [38] in 2004. Prior to Ostrovskii [38], Simonson [46] studied the same parameter under a different name to approximate the cutwidth of outerplanar graphs. Although several graph theoretical results have been presented [6, 27, 28, 30, 33, 39] after Ostrovskii [38], so far, no results on the complexity of the problem were known. In this paper, we present the first such results. The parameter is defined as follows. Let G be a graph and T a spanning tree of G . The *detour* for an edge $\{u, v\} \in E(G)$ is the unique u - v path in T . We define the *congestion* of $e \in E(T)$, denoted by $\text{cng}_{G,T}(e)$, as the number of detours that contain e . The *congestion of G in T* , denoted by $\text{cng}_G(T)$, is the maximum congestion over all edges in T . The *spanning tree congestion* of G , denoted by $\text{stc}(G)$, is the minimum congestion over all spanning trees of G . We denote by STC the problem of determining whether a given graph has spanning tree congestion at most some given k . If k is fixed, we denote the problem by k -STC.

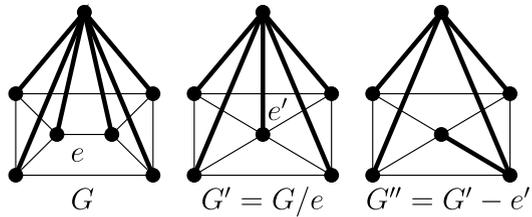
The name of the parameter comes from the following analogy [6]: Edges of G are roads, and edges of T are those roads which are cleaned from snow after snowstorms. For an edge $h \in E(T)$, it is natural to define the congestion of h as the number of detours passing through h . Clearly, the congestion of the busiest roads should be minimized.

The *tree spanner problem* [5] is a variant of the STC problem, which minimizes the dilation, that is, the length of the longest detours. The problem is NP-complete even on chordal and chordal bipartite graphs [37]. This is a well-studied problem with various applications in distributed systems and communication networks [41]. Several pairs of congestion and dilation problems are known [42]. The most famous pair is the cutwidth problem and the bandwidth problem. It is worth to mention that while the nature of this problem is different from STC, some algorithmic techniques work well for both problems. For example, the techniques developed in this work were used to obtain FPT-algorithms on graphs of bounded degree [22] and the approach to solve the tree spanner problem on planar, and more generally, apex-minor-free, graphs developed in [17] is used to solve STC in this paper.

Our contribution In this paper, we obtain the following results on the STC problem. We show that the problem is *fixed-parameter tractable* on large classes of sparse graphs. We refer to the book of Downey and Fellows [19] for an introduction to Parameterized Complexity. In particular, we prove that

- k -STC can be solved in linear time for every fixed k on apex-minor-free graphs, and thus on planar graphs and graphs of bounded genus, and on graphs of bounded degree;
- k -STC can be solved in linear time for $1 \leq k \leq 3$.

Fig. 1 $\text{stc}(G) = 4$, $\text{stc}(G') = 5$, and $\text{stc}(G'') = 6$



It is natural to ask if our results can be extended to even larger classes of sparse graphs like d -degenerate graphs or graphs excluding some non-apex graph as a minor. We show that this is not the case and thus our results are in some sense tight. We prove that

- k -STC is NP-complete for every fixed $k \geq 8$ on graphs excluding K_6 as a minor and with only one vertex of unbounded degree.

The remaining part of the paper is organized as follows. Section 2 provides definitions and some preliminary facts. In Sect. 3, we show that k -STC is expressible in Monadic Second Order Logic (MSOL), which combined with the combinatorial results obtained in Sects. 4 and 5 gives the main algorithmic results of the paper. In Sect. 4, we prove that for every apex-minor-free graph G , its treewidth is at most $C \cdot \text{stc}(G)$ for some constant C depending only on the size of the excluded apex graph. The proof of this result is based on extensions of ideas from bidimensionality theory of Demaine et al. [10] and the Structure Theorem of Robertson and Seymour [44]. However, the framework of Demaine et al. [10] for solving parameterized problems is strongly based on the assumption that problems should be minor or edge contraction closed, which is not the case for k -STC, see e.g. Fig. 1. Here, to prove the bound, we follow the approach based on topological minors used in [17] for sparse spanners. In Sect. 5, we obtain similar combinatorial bounds on the treewidth of a graph as a function of its maximum vertex degree and its congestion. Combined with results on the expressibility in MSOL, this yields that k -STC is fixed-parameter tractable on apex-minor-free graphs and graphs of bounded maximum degree.

In Sect. 7, we provide a number of complexity results. We start by showing that k -STC remains NP-complete on planar graphs when k is part of the input. In Sect. 7.2, we show that k -STC is NP-complete for edge-weighted graphs if $k \geq 8$ on a very specific class of graphs, namely, apex graphs with one vertex of unbounded degree. Using the result of Sect. 7.2, we show in Sect. 7.3 that for $k \geq 8$, k -STC is NP-complete for simple unweighted apex graphs with only one vertex of unbounded degree. In particular, this shows that the problem is NP-complete even on K_6 -free graphs for $k \geq 8$, and thus the fixed-parameter tractability of the problem on apex-minor-free graphs cannot be extended to larger classes of graphs excluding some fixed graph as a minor. In the last section, we show the approximation hardness of the spanning tree congestion problem and conclude the paper with open questions.

2 Preliminaries

We consider finite undirected graphs that have no loops or multiple edges if it is not stated otherwise explicitly. Let $G = (V, E)$ be a graph. We refer to the vertex and edge sets of G as $V(G)$ and $E(G)$ respectively. For a vertex v , we denote by $N_G(v)$ its (*open*) *neighborhood*, i.e. the set of vertices which are adjacent to v in G . By $\deg_G(v)$ we denote the *degree* of v in G . We may omit the index if the graph under consideration is clear from the context. For $U \subseteq V(G)$, we denote by $G[U]$ the subgraph induced by U . If $U \subseteq V(G)$ (resp. $u \in V(G)$ or $E \subseteq E(G)$ or $e \in E(G)$) then $G - U$ (resp. $G - u$ or $G - E$ or $G - e$) is the graph obtained from G by the removal of vertices of U (resp. of vertex u or the edges of E or of the edge e). For graphs G_1 and G_2 , $G_1 \cap G_2$ (respectively $G_1 \cup G_2$) is the graph with the vertex set $V(G_1) \cap V(G_2)$ and the edge set $E(G_1) \cap E(G_2)$ (respectively the vertex set $V(G_1) \cup V(G_2)$ and the edge set $E(G_1) \cup E(G_2)$).

We extend the notion of spanning tree congestion to edge-weighted graphs, by defining the congestion of an edge as the sum of the weights of the edges whose detours pass through the edge. We denote by $w(F)$ the sum of the weights of the edges in F for an edge set $F \subseteq E(G)$.

Let G be a connected graph. For $A, B \subseteq V(G)$, we define $E_G(A, B) = \{\{u, v\} \in E(G) \mid u \in A, v \in B\}$. For $S \subseteq V(G)$, we define the *boundary edges* of S , denoted by $\theta_G(S)$, as $\theta_G(S) = E_G(S, V(G) \setminus S)$. Using this notation, we can redefine $\text{cng}_{G,T}(e)$ as $\text{cng}_{G,T}(e) = |\theta_G(A_e)|$, where A_e is the vertex set of one of the two components of $T - e$. From this redefinition through boundary edges, we can see that *c-cut trees* defined by Fekete and Kremer [21] and spanning trees of congestion at most c are equivalent.

For an edge e in a tree T , we say that e separates A and B if $A \subseteq A_e$ and $B \subseteq B_e$, where A_e and B_e are the vertex sets of the two components of $T - e$. The following often used proposition can easily be observed.

Proposition 2.1 *Let T be a spanning tree of G and let $e \in E(T)$ separate A and B . If G is unweighted, then $\text{cng}_{G,T}(e) \geq |E(A, B)|$, and if G is weighted, then $\text{cng}_{G,T}(e) \geq w(E(A, B))$.*

From the definition of the spanning tree congestion, the following proposition holds.

Proposition 2.2 *The spanning tree congestion of G equals the maximum spanning tree congestion of its biconnected components.*

Ostrovskii [38] showed the following lower bound on the spanning tree congestion of graphs.

Lemma 2.3 (Ostrovskii [38]) *Let G be a graph, and let $u, v \in V(G)$. If G has k edge disjoint u - v paths, then $\text{stc}(G) \geq k$.*

Treewidth The concept of treewidth was introduced by Robertson and Seymour in their project of Graph Minor Theory (see for example [43]). A *tree decomposition* of a graph G is a pair (\mathcal{X}, T) , where T is a tree and $\mathcal{X} = \{X_i \mid i \in V(T)\}$ is a collection of subsets of $V(G)$ such that

- $\bigcup_{i \in V(T)} X_i = V(G)$,
- for each edge $\{u, v\} \in E(G)$, there is a *node* $i \in V(T)$ such that $u, v \in X_i$, and
- for each $v \in V(G)$, the set of nodes $\{i \mid v \in X_i\}$ forms a subtree of T .

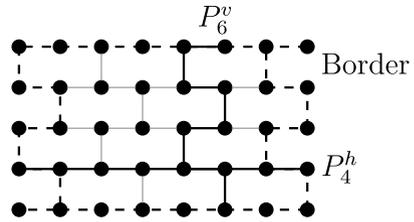
The elements in \mathcal{X} are called *bags*. The *width* of a tree decomposition (\mathcal{X}, T) equals $\max_{i \in V(T)} |X_i| - 1$. The *treewidth* of G , denoted by $\text{tw}(G)$, is the minimum width over all tree decompositions of G . A tree decomposition where T is a path is called a *path decomposition* and the *pathwidth* of G is the minimum width over all path decompositions of G .

Embeddings in Surfaces and Euler Genus A *surface* Σ is a compact 2-manifold without boundary (we always consider connected surfaces). A *line* in Σ is a subset homeomorphic to $[0, 1]$ and a (closed) *disk* (resp. *open disk*) $\Delta \subseteq \Sigma$ is a subset homeomorphic to $\{(x, y) : x^2 + y^2 \leq 1\}$ (resp. $\{(x, y) : x^2 + y^2 < 1\}$) in \mathbb{R}^2 . Whenever we refer to a Σ -*embedded graph* G , we consider a 2-cell embedding of G in Σ . To simplify notation, we do not distinguish between a vertex of G and the point of Σ used in the drawing to represent the vertex, or between an edge and the line representing it. We also consider a graph G embedded in Σ as the union of the points corresponding to its vertices and edges. This way, a subgraph H of G can be seen as a graph H , where $H \subseteq G$.

The Euler genus $\text{eg}(\Sigma)$ of a non-orientable surface Σ is equal to the non-orientable genus $\tilde{g}(\Sigma)$ (or the crosscap number). The Euler genus $\text{eg}(\Sigma)$ of an orientable surface Σ is $2g(\Sigma)$, where $g(\Sigma)$ is the orientable genus of Σ . The *Euler genus* of a graph G , $\text{eg}(G)$, is the minimum Euler genus of a surface Σ such that G can be Σ -embedded. For additional information about graphs on surfaces we refer to the book by Mohar and Thomassen [36].

A graph is *planar* if it is embeddable in a sphere or a plane. Let G be a graph embedded in a plane Σ . Then the set $\Sigma \setminus G$ is open, and its regions are called the *faces* of G . Let $\mathcal{F}(G)$ be the set of faces of the embedding of G . It is said that an edge $e \in E(G)$ is *incident* to a face $f \in \mathcal{F}(G)$ if e is in the border of f . Recall that each edge is incident either to the unique outer face if it is a bridge or to exactly two faces otherwise. A *dual graph* G^* of G is a multigraph (i.e. G^* can have loops and multiple edges) with the vertex set $\mathcal{F}(G)$ in which for any edge $e \in E(G)$ incident to a single face f , there is a separate loop $\{f, f\}$ in G^* (i.e. if we have several edges incident only to f then there is the same number of loops), and for any edge $e \in E(G)$ incident to two distinct faces f, f' , there is a separate edge $\{f, f'\}$. Hence, for each edge $e \in E(G)$, there is the *dual edge* e^* in G^* . For G^* , we always assume that this multigraph is embedded in a plane, and that this embedding is induced by the embedding of G in such a way that each vertex $f \in V(G^*)$ is a point inside the face f , and for any $e \in E(G)$, the line e intersect only e^* in G^* in exactly one point. It is well known that if G is connected then $G = (G^*)^*$, and that for an edge set $X \subseteq E(G)$, X is the set of edges of a cycle in G if and only if the set $X^* = \{e^* \mid e \in X\}$ is a minimal edge-cut in G^* .

Fig. 2 A (5, 8)-wall with the border and paths P_4^h, P_6^v



Minors and Topological Minors Let G be a graph. We say that a graph H is obtained from G by an *edge subdivision* if $V(H) = V(G) \cup \{w\}$ and $E(H) = E(G) \setminus \{u, v\} \cup \{u, w\}, \{w, v\}$ for some edge $\{u, v\} \in E(G)$ and a new vertex w . We say that H is a *subdivision* of G if H can be obtained from G by a finite sequence of edge subdivisions. If a subdivision of H is a subgraph of G , then H is a *topological minor* of G .

Given an edge $e = \{x, y\}$ of a graph G , the graph G/e is obtained from G by contracting the edge e ; that is, to get G/e we identify the vertices x and y , remove all loops, and replace all multiple edges by simple edges. A graph H obtained by a sequence of edge-contractions is said to be a *contraction* of G . Graph H is a *minor* of G if H is a subgraph of a contraction of G .

We say that a graph G is *H-minor-free* when it does not contain H as a minor. We also say that a graph class \mathcal{G} is *H-minor-free* (or, excludes H as a minor) when all its members are H -minor-free. For example, the class of planar graphs is a K_5 -minor-free graph class.

An *apex graph* is a graph obtained from a planar graph G by adding a vertex and making it adjacent to some vertices of G . A graph class \mathcal{G} is *apex-minor-free* if \mathcal{G} excludes a fixed apex graph H as a minor. Many classes of graphs are apex-minor-free, including the classes of planar graphs, graphs of bounded treewidth, and graphs of bounded genus. This class was studied intensively from combinatorial and algorithmic perspectives [11, 15, 17, 18, 20, 24].

Grids and Walls The (r, s) -grid is the Cartesian product of two paths of lengths $r - 1$ and $s - 1$. The (r, s) -wall is a graph $W_{r,s}$ with the vertex set

$$\{(i, j) : 1 \leq i \leq r, 1 \leq j \leq s\}$$

such that two vertices (i, j) and (i', j') are adjacent if and only if either $i = i'$ and $j' \in \{j - 1, j + 1\}$, or $j = j'$ and $i' = i + (-1)^{i+j}$.

Let $W_{r,s}$ be a wall. By P_i^h we denote the shortest path connecting vertices $(i, 1)$ and (i, s) , and by P_j^v is denoted the shortest path connecting vertices $(1, j)$ and (r, j) with the assumption that, for $j > 1$, P_j^v contains only vertices (x, y) with $x = j - 1, j$. See Fig. 2 for an illustration of these notions. If W is obtained by subdividing edges of $W_{r,s}$, with slightly abusing the notation, we also will be using these terms for the paths obtained by subdivisions from the corresponding paths of $W_{r,s}$. Vertices of paths P_1^h, P_r^h, P_1^v and P_s^v are called *border vertices* of W . The paths P_1^h, P_r^h, P_1^v and P_s^v compose the *border* of W . We say that vertices of $W_{r,s}$ are the *wall vertices* of W .

Notice that if a graph G contains the (r, r) -grid as a minor, then it contains W_{rr} as a topological minor: since W_{rr} is a subgraph of the (r, r) -grid, we have that when G

contains the (r, r) -grid as a minor, it also contains W_{rr} as a minor, and every minor with vertices of degree at most 3 is also a topological minor (see e.g. [16]).

Monadic Second Order Logic The syntax of Monadic Second Order Logic (MSOL) of graphs includes the logical connectives $\vee, \wedge, \neg, \Leftrightarrow, \Rightarrow$, variables for vertices, edges, sets of vertices, and sets of edges, the quantifiers \forall, \exists that can be applied to these variables, and the following five binary relations:

1. $u \in U$ where u is a vertex variable and U is a vertex set variable.
2. $d \in D$ where d is an edge variable and D is an edge set variable.
3. $\text{inc}(d, u)$, where d is an edge variable, u is a vertex variable, and the interpretation is that the edge d is incident to the vertex u .
4. Equality, $=$, of variables representing vertices, edges, sets of vertices and sets of edges.

3 Expressibility in MSOL

All algorithmic results obtained in this paper have the following “ingredients” in common. We bound the treewidth of a graph by some function of its spanning tree congestion. After we obtain the bound on treewidth, we use Courcelle’s Theorem [8] to solve k -STC in linear time on graphs of bounded treewidth. In order to be able to apply Courcelle’s Theorem, we need the following lemma. For a different proof of Courcelle’s Theorem and more information on how problems can be expressed in MSOL, see [2].

Lemma 3.1 *The k -STC problem is expressible in Monadic Second Order Logic (MSOL).*

Proof Let $G = (V, E)$ and $|G|_2 := \langle V, E, \text{inc} \rangle$. For a vertex $v \in V$ and an edge $e \in E$, $\text{inc}(v, e)$ if and only if e has v as an endpoint. For $F \subseteq E(G)$, we denote by $G\langle F \rangle$ the subgraph induced by F , that is, $E(G\langle F \rangle) = F$ and $V(G\langle F \rangle) = \bigcup_{\{u,v\} \in F} \{u, v\}$.

We first define the following basic expressions:

$$\begin{aligned} \text{Deg1}(v_1, E_1) &:= (\exists e_1 \in E_1)(\forall e_2 \in E_1)(e_1 = e_2 \iff \text{inc}(v_1, e_2)), \\ \text{Part}(V_1, V_2, V_3) &:= V_2 \neq \emptyset \wedge V_3 \neq \emptyset \wedge (V_2 \cup V_3 = V_1) \wedge (V_2 \cap V_3 = \emptyset), \\ \text{Adj}(v_1, v_2, E_1) &:= v_1 \neq v_2 \wedge (\exists e_1 \in E_1)(\text{inc}(v_1, e_1) \wedge \text{inc}(v_2, e_1)), \\ E_1 = \text{Ind}(V_1) &:= (\forall e_1)(e_1 \in E_1 \iff (\exists v_1, v_2 \in V_1) \\ &\quad (v_1 \neq v_2 \wedge \text{inc}(v_1, e_1) \wedge \text{inc}(v_2, e_1))), \\ E_1 = \text{Inc}_E(v_1) &:= (\forall e_1)(e_1 \in E_1 \iff \text{inc}(v_1, e_1)), \\ V_1 = \text{Inc}_V(E_1) &:= (\forall v_1)(v_1 \in V_1 \iff (\exists e_1 \in E_1)(\text{inc}(v_1, e_1))). \end{aligned}$$

It is easy to see that $\text{Deg1}(v_1, E_1)$ if and only if v_1 has only one neighbor in $G\langle E_1 \rangle$, $\text{Part}(V_1, V_2, V_3)$ if and only if (V_2, V_3) forms a partition of V_1 , $\text{Adj}(v_1, v_2, E_1)$ if

and only if an edge $\{v_1, v_2\}$ is in E_1 , $E_1 = \text{Ind}(V_1)$ if and only if E_1 is the edge set of $G[V_1]$, $E_1 = \text{Inc}_E(v_1)$ if and only if E_1 is the set of edges between v_1 and its neighbors, and $V_1 = \text{Inc}_V(E_1)$ if and only if V_1 is the vertex set of $G\langle E_1 \rangle$.

Using the above basic expressions, we define some expressions related to connectivity of graphs.

$$\begin{aligned} \text{Conn}(E_1) &:= (\forall V_2, V_3)(\text{Part}(\text{Inc}_V(E_1), V_2, V_3) \\ &\implies (\exists v_2 \in V_2, v_3 \in V_3)(\text{Adj}(v_2, v_3, E_1))), \\ \text{BiConn}(E_1) &:= (\exists v_1, v_2, v_3 \in \text{Inc}_V(E_1)) \left(\bigwedge_{1 \leq i < j \leq 3} (v_i \neq v_j) \right) \\ &\wedge (\forall v_4)(\text{Conn}(E_1 \setminus \text{Inc}_E(v_4))). \end{aligned}$$

Clearly, $\text{Conn}(E_1)$ if and only if $G\langle E_1 \rangle$ is connected, and $\text{BiConn}(E_1)$ if and only if $G\langle E_1 \rangle$ is biconnected. Using these expressions, we can define the following expressions.

$$\begin{aligned} \text{Forest}(E_1) &:= (\forall V_1 \subseteq \text{Inc}_V(E_1))(\neg \text{BiConn}(\text{Ind}(V_1) \cap E_1)), \\ \text{Tree}(E_1) &:= \text{Forest}(E_1) \wedge \text{Conn}(E_1), \end{aligned}$$

$$\begin{aligned} \text{Path}(v_1, v_2, E_1) &:= \text{Tree}(E_1) \\ &\wedge (\forall v_3 \in \text{Inc}_V(E_1))(\text{Deg}1(v_3, E_1) \iff v_3 = v_1 \vee v_3 = v_2). \end{aligned}$$

The meanings are clear: $\text{Forest}(E_1)$ if and only if $G\langle E_1 \rangle$ is a forest, $\text{Tree}(E_1)$ if and only if $G\langle E_1 \rangle$ is a tree, and $\text{Path}(v_1, v_2, E_1)$ if and only if $G\langle E_1 \rangle$ is a v_1 - v_2 path. Then, defining the expression $\text{SpnTree}(E_1)$ that means $G\langle E_1 \rangle$ is a spanning tree of G is an easy task.

$$\text{SpnTree}(E_1) := \text{Tree}(E_1) \wedge (\forall v)(v \in \text{Inc}_V(E_1)).$$

It is also easy to define the expression $\text{Detour}(e_1, E_1)$ such that $\text{Detour}(e_1, E_1)$ if and only if $G\langle E_1 \rangle$ forms a detour for e_1 :

$$\text{Detour}(e_1, E_1) := (\exists v_1, v_2)(v_1 \neq v_2 \wedge \text{inc}(v_1, e_1) \wedge \text{inc}(v_2, e_1) \wedge \text{Path}(v_1, v_2, E_1)).$$

Let us remark that every edge of a spanning tree is a detour of itself. In particular, e_0 itself is a detour containing e_0 . The following expression $\text{Cong}_k(e_0, E_0)$ means that e_0 is contained in at most k detours in $G\langle E_0 \rangle$.

$$\begin{aligned} \text{Cong}_k(e_0, E_0) &:= \neg(\exists e_1, \dots, e_k) \left(\bigwedge_{1 \leq i \leq k} e_i \notin E_0 \wedge \bigwedge_{1 \leq i < j \leq k} e_i \neq e_j \right. \\ &\left. \wedge \bigwedge_{1 \leq i \leq k} (\exists E_i)(e_0 \in E_i \wedge E_i \subseteq E_0 \wedge \text{Detour}(e_i, E_i)) \right). \end{aligned}$$

Obviously, $\text{stc}(G) \leq k$ if and only if $G \models (\exists E_0)(\text{SpnTree}(E_0) \wedge (\forall e_0 \in E_0)(\text{Cong}_k(e_0, E_0)))$. □

Using Lemma 3.1, the following lemma follows almost directly from deep results of Bodlaender [1] and Courcelle [8].

Lemma 3.2 *Let \mathcal{G} be a class of graphs such that, for every $G \in \mathcal{G}$, $\text{stc}(G)$ is at least $c_{\mathcal{G}} \cdot \text{tw}(G)$, where $c_{\mathcal{G}}$ is a constant which depends only on \mathcal{G} . Then k -STC is solvable in linear time on \mathcal{G} for every fixed k .*

Proof Let $G \in \mathcal{G}$ be a graph on n vertices and m edges. For a given integer k , we use Bodlaender’s Algorithm [1] to decide in time $O(n + m)$ if $\text{tw}(G) \leq k/c_{\mathcal{G}}$ (the hidden constants in the big-O depend only on k and $c_{\mathcal{G}}$). If Bodlaender’s Algorithm reports that $\text{tw}(G) > k/c_{\mathcal{G}}$, then we conclude that $\text{stc}(G) > k$. Otherwise (when $\text{tw}(G) \leq k/c_{\mathcal{G}}$), Bodlaender’s Algorithm computes a tree decomposition of G of width at most $k/c_{\mathcal{G}}$. Now we apply Courcelle’s Theorem [8], namely that every problem expressible in MSOL can be solved in linear time on graphs of constant treewidth. \square

4 Spanning Tree Congestion of Apex-Minor-Free Graphs

In this section, we prove that apex-minor-free graphs of large treewidth have a large spanning tree congestion. First, we need the following technical lemma.

Lemma 4.1 *Let graph G be the union of two graphs G_P and G^+ (the set $V(G^+)$ can be empty), such that G_P is planar and for some planar embedding of G_P the only common vertices of G_P and G^+ are the vertices from the border of the external face of G_P . If G_P contains an (r, r) -wall as a topological minor, then $\text{stc}(G) \geq \frac{r}{4} - 2$.*

Proof Let $r \geq 12$, otherwise the inequality is trivial. Assume that G_P is embedded in a plane in such a way that the only common vertices of G_P and G^+ are the vertices from the border of the external face of G_P . Since G_P contains an (r, r) -wall as a topological minor, it also contains a subdivided (r, r) -wall W as a subgraph. Let $\{(i, j) : 1 \leq i \leq r, 1 \leq j \leq r\}$ be the set of wall vertices of W . Let T be a spanning tree of G such that $\text{cng}_G(T) = \text{stc}(G)$. Denote by u the wall vertex $(\lfloor \frac{r}{2} \rfloor, \lfloor \frac{r}{2} \rfloor)$ of G_P . Let P be the shortest path in the spanning tree T connecting u with one of the border vertices of W . Observe that this path is a path in G_P and, moreover, this path is inside the disk Δ in the plane bordered by the border of W . Let v be the first vertex of P on the way from u which is on one of the paths $P_{\lfloor \frac{r}{4} \rfloor + 1}^h, P_{r - \lfloor \frac{r}{4} \rfloor}^h, P_{\lfloor \frac{r}{4} \rfloor + 1}^v$ or $P_{r - \lfloor \frac{r}{4} \rfloor}^v$ and let e be the first edge of P incident to v . This construction is shown in Fig. 3. We denote by T_1, T_2 the subtrees of T and by P_1, P_2 the subpaths of P obtained by the removal of e . We assume that $u \in V(P_1) \subseteq V(T_1)$ and $v \in V(P_2) \subseteq V(T_2)$. We consider the sets $U_1 = V(T_1) \cap V(G_P)$ and $U_2 = V(T_2) \cap V(G_P)$. Clearly, U_1, U_2 is a partition of $V(G_P)$. The subgraph $G_P[U_1]$ can have several components. We choose the component which contains P_1 and denote by U'_1 the set of vertices of this component. Let U'_2 be the set of vertices of the component of $G_P[U_2]$ which contains P_2 respectively. The set of edges $Z = \{\{x, y\} \in E(G_P) | x \in U_1, y \in U_2\}$ separates U_1, U_2 and, therefore, U'_1, U'_2 . Consider a minimal edge-cut $X \subseteq Z$ in G_P that separates U'_1 and U'_2 . Since $G_P[U'_1]$ and $G_P[U'_2]$ are connected, X is a minimal

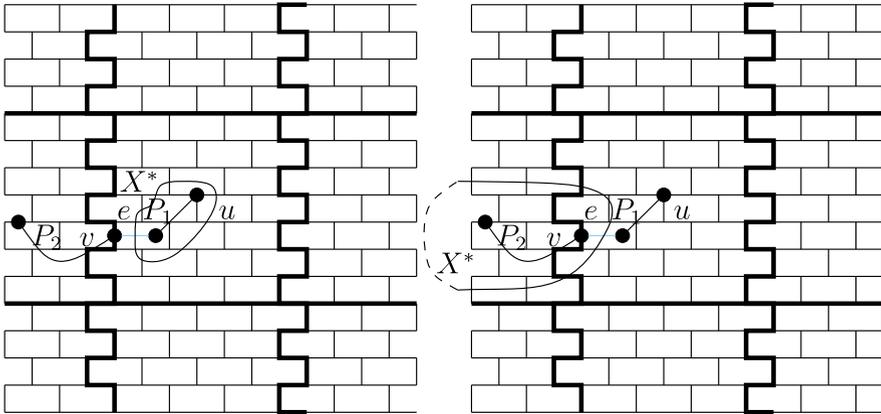


Fig. 3 Separation of the paths P_1 and P_2 in the wall

edge-cut in G_P . Hence, the dual graph G_P^* contains a cycle with the set of edges X^* which corresponds to the cut X . This cycle has the following properties:

- Edges of X^* form the border of a disk in the plane. Moreover, one of the paths P_1 and P_2 is inside this disk and the other is outside.
- The dual edge e^* corresponding to e is in X^* .

The edge e^* crosses e in the embeddings of G_P and G_P^* . Observe also that e separates two different faces f_1 and f_2 of G_P . Let Q be the f_1 - f_2 path with the edge set $X^* \setminus \{e^*\}$. Now we estimate the number of edges of G_P in W that are crossed by X^* . We consider two cases.

Case 1. The cycle with the edge set X^* is inside of Δ (see the left half of Fig. 3). We have two subcases.

- v is a vertex of $P_{\lfloor \frac{r}{4} \rfloor + 1}^h$ or $P_{r - \lfloor \frac{r}{4} \rfloor}^h$. Because of symmetry, we assume that $v \in V(P_{\lfloor \frac{r}{4} \rfloor + 1}^h)$. The path Q should at least twice intersect each path P_i^h for $i \in \{\lfloor \frac{r}{4} \rfloor + 2, \dots, \lfloor \frac{r}{2} \rfloor\}$. It follows that it intersects at least $2(\lfloor \frac{r}{2} \rfloor - \lfloor \frac{r}{4} \rfloor - 1) \geq \frac{r}{4} - 3$ edges of W .
- v is a vertex of $P_{\lfloor \frac{r}{4} \rfloor + 1}^v$ or $P_{r - \lfloor \frac{r}{4} \rfloor}^v$. Using the symmetry, we assume that $v \in V(P_{\lfloor \frac{r}{4} \rfloor + 1}^v)$. Now the path Q should at least twice intersect each path $P_{\lfloor \frac{r}{4} \rfloor + 2i + 1}^v$ for $1 \leq i \leq \frac{1}{2}(\lfloor \frac{r}{2} \rfloor - \lfloor \frac{r}{4} \rfloor - 1)$. It follows that Q intersects at least $2\lfloor \frac{1}{2}(\lfloor \frac{r}{2} \rfloor - \lfloor \frac{r}{4} \rfloor - 1) \rfloor \geq \frac{r}{4} - 3$ edges of W .

Case 2. Otherwise, the cycle with the edge set X^* contains a vertex outside Δ (see the right half of Fig. 3). Let Q_1, Q_2 be the shortest f_1 - f'_1 and f_2 - f'_2 subpaths of Q respectively such that f'_1, f'_2 are embedded outside Δ . We estimate the number of edges of G_P in W that are crossed by Q_1, Q_2 . We do it for Q_1 , since the bound for Q_2 is the same, and consider two cases.

- The edge of Q_1 incident to f'_1 crosses P_1^h or P_r^h . Because of symmetry, we assume that this edge intersects P_1^h . The path Q_1 should then intersect each path P_i^h for $i \in \{1, \dots, \lfloor \frac{r}{4} \rfloor + 1\}$. Therefore, it intersects at least $\lfloor \frac{r}{4} \rfloor + 1 \geq \frac{r}{4} - 3$ edges of W .
- The edge of Q_1 incident to f'_1 crosses P_1^v or P_r^v . Using the symmetry, we assume that this edge intersects $P_{\lfloor \frac{r}{4} \rfloor + 1}^v$. Now the path Q_1 should intersect each path P_{2i-1}^v for $1 \leq i \leq \frac{1}{2}(\lfloor \frac{r}{4} \rfloor + 2)$. It follows that Q intersects at least $\lfloor \frac{r}{8} \rfloor + 1$ edges of W .

Since $2(\lfloor \frac{r}{8} \rfloor + 1) \geq \frac{r}{4} - 3$, the paths Q_1, Q_2 (and, therefore, Q) intersect at least $\frac{r}{4} - 3$ edges of W .

Taking into account the edge e^* , we conclude that the edges of X^* intersect at least $\frac{r}{4} - 2$ edges of G_P in W . It follows that $|X^*| \geq \frac{r}{4} - 2$. Therefore,

$$\text{stc}(G) = \text{cng}_G(T) \geq \text{cng}_{G,T}(e) \geq |Z| \geq |X| = |X^*| \geq \frac{r}{4} - 2. \quad \square$$

4.1 Planar Graphs and Graphs of Bounded Genus

Next, we establish lower bounds of the spanning tree congestion for planar graphs. We will need the following result, which is due to Robertson, Seymour, and Thomas [45].

Proposition 4.2 (Robertson et al. [45]) *Every planar graph with no (r, r) -grid as a minor has treewidth $\leq 6r - 5$.*

Now we can prove the following lemma.

Lemma 4.3 *For a planar graph G , $\frac{1}{24} \text{tw}(G) - \frac{49}{24} \leq \text{stc}(G)$.*

Proof Let $r = \lfloor \frac{\text{tw}(G)+4}{6} \rfloor$. By Proposition 4.2, G contains an (r, r) -grid as a minor and therefore it contains an (r, r) -wall as a topological minor. Now by Lemma 4.1 for $G_P = G$ and $V(G^+) = \emptyset$, $\text{stc}(G) \geq \frac{r}{4} - 2$, and the proof of the lemma follows. \square

Now we extend these bounds to graphs of bounded genus. The following extension of Proposition 4.2 on graphs of bounded genus is due to Demaine et al. [10].

Proposition 4.4 (Demaine et al. [10]) *Let G be a graph embeddable on a surface with Euler genus $\text{eg}(G)$, and having treewidth more than $6r(\text{eg}(G) + 1)$. Then G contains an (r, r) -grid as a minor.*

We also need a result that for any embedding of a sufficiently large wall in a surface Σ of small genus, a large part of the wall would have a “planar” embedding. We use the following variant of this result, which is a direct corollary of the results from Geelen et al. [23]; see also the works of Mohar and Thomassen [35, 47].

Proposition 4.5 (Dragan et al. [17]) *Let g, l, r be positive integers such that $r \geq 2g(l + 1)$ and let W be an (r, r) -wall. If W is embedded on a surface Σ of Euler*

genus at most $g^2 - 1$, then some (l, l) -subwall of G is embedded in a closed disk Δ in Σ , such that the border of the (l, l) -wall composes the boundary of the disk.

Combining this result and Proposition 4.4, and using the same arguments as in the planar case, we have the following lemma.

Lemma 4.6 For a graph G of Euler genus g , $\text{stc}(G) = \Omega(\frac{\text{tw}(G)}{g^{3/2}})$.

Proof Let G be a Σ -embedded graph and let $k = \text{tw}(G)$. We put $g = \text{eg}(\Sigma)$ and $r = \lfloor \frac{k-1}{6(g+1)} \rfloor$. By Proposition 4.4, G contains an (r, r) -grid as a minor, and thus, an (r, r) -wall as a topological minor. By Proposition 4.5, there is a subgraph $W \subseteq G$, which is isomorphic to a subdivision of an $(\lfloor \frac{r}{2\sqrt{g+1}} \rfloor - 1, \lfloor \frac{r}{2\sqrt{g+1}} \rfloor - 1)$ -wall, such that the border of this wall borders some disk Δ containing W (in the Σ -embedded graph G). We assume that G_P is the subgraph induced by vertices of G which are embedded in Δ and that G^+ is the subgraph of G induced by vertices which are embedded outside Δ and by vertices on the border of the wall. By Lemma 4.1, we have that $\text{stc}(G) = \Omega(\lfloor \frac{r}{2\sqrt{g+1}} \rfloor - 1)$. Thus $\text{stc}(G) = \Omega(\frac{k}{g^{3/2}})$. \square

4.2 Apex-Minor-Free Graphs

Finally, we extend our bounds to apex-minor-free graphs. This extension is based on a structural theorem of Robertson and Seymour [44]. Before describing this theorem, we need some definitions.

Definition 1 (Clique sums) Let G_1 and G_2 be two disjoint graphs, and $h \geq 0$ an integer. For $i = 1, 2$, let $W_i \subseteq V(G_i)$ form a clique of size h , and let G'_i be the graph obtained from G_i by removing a set of edges (possibly empty) from the clique $G_i[W_i]$. Let $F : W_1 \rightarrow W_2$ be a bijection between W_1 and W_2 . We define the h -clique-sum of G_1 and G_2 , denoted by $G_1 \oplus_{h,F} G_2$, or simply by $G_1 \oplus G_2$ if there is no confusion, as the graph obtained by taking the union of G'_1 and G'_2 , identifying each $w \in W_1$ with $F(w) \in W_2$, and removing all the multiple edges. The image of the vertices of W_1 and W_2 in $G_1 \oplus G_2$ is called the *join* of the sum.

Note that some edges of G_1 and G_2 are not edges of $G = G_1 \oplus G_2$, since it is possible that they had edges which were removed by the clique-sum operation. Such edges are called *virtual edges* of G . We remark that \oplus is not well defined; different choices of G'_i and the bijection F could give different clique-sums. A sequence of h -clique-sums, not necessarily unique, which result in a graph G , is called a *clique-sum decomposition* of G .

Definition 2 (h -nearly embeddable graphs) Let Σ be a surface and $h > 0$ be an integer. A graph G is h -nearly embeddable in Σ if there is a set of vertices $X \subseteq V(G)$ (called *apices*) of size at most h , such that graph $G - X$ is the union of subgraphs G_0, \dots, G_h with the following properties:

- (i) There is a set of cycles C_1, \dots, C_h in Σ such that the cycles C_i are the borders of open pairwise disjoint disks Δ_i in Σ ;

- (ii) G_0 has an embedding in Σ in such a way that $G_0 \cap \bigcup_{i=1, \dots, h} \Delta_i = \emptyset$;
- (iii) Graphs G_1, \dots, G_h (called *vortices*) are pairwise disjoint and for $1 \leq i \leq h$, $V(G_0) \cap V(G_i) \subset C_i$;
- (iv) For $1 \leq i \leq h$, let $U_i := \{u_1^i, \dots, u_{m_i}^i\}$ be the vertices of $V(G_0) \cap V(G_i) \subset C_i$ appearing in an order obtained by a clockwise traversal of C_i . We call vertices of U_i *bases* of G_i . Then G_i has a path decomposition $\mathcal{B}_i = (B_j^i)_{1 \leq j \leq m_i}$ of width at most h , such that $u_j^i \in B_j^i$ for $1 \leq j \leq m_i$.

The following proposition is known as the Excluded Minor Theorem [44] and is the cornerstone of Robertson and Seymour’s Graph Minors theory. We need a stronger version of this theorem, which follows from its proof in [44] (see e.g. [14]).

Proposition 4.7 (Robertson and Seymour [44]) *For every non-planar graph H , there exists an integer h , depending only on H , such that every graph excluding H as a minor can be obtained by h -clique-sums on graphs that can be h -nearly embedded in a surface Σ in which H cannot be embedded. Moreover, while applying each of the clique sums, at most three vertices from each summand other than apices and vertices in vortices are identified.*

Let us remark that by the result of Demaine et al. [14] such a clique-sum decomposition can be obtained in $n^{O(1)}$ time (the exponent in the running time depends only on H). However, we use Robertson and Seymour’s theorem only for the proof of the combinatorial bound, so we do not need to construct such a decomposition.

We need the following two well-known results about treewidth.

Lemma 4.8 *If G_1 and G_2 are graphs, then $\text{tw}(G_1 \oplus G_2) \leq \max\{\text{tw}(G_1), \text{tw}(G_2)\}$.*

Lemma 4.9 *If G is a graph and $X \subseteq V(G)$, then $\text{tw}(G - X) \geq \text{tw}(G) - |X|$.*

The following lemma is implicit in the proofs from [10, 13]. Here we give it as it is stated in [12].

Lemma 4.10 (Demaine and Hajiaghayi [12, Lemma 4.3]) *Let $G = G_0 \cup G_1 \cup \dots \cup G_h$ be an h -nearly embeddable graph without apices (i.e. where $X = \emptyset$). Then $\text{tw}(G) \leq \frac{3}{2}(h + 1)^2(\text{tw}(G_0) + 2h + 1)$.*

We also use the following auxiliary claims. Suppose that a graph G is presented as an h -clique-sum $G = F \oplus F^{(1)} \oplus \dots \oplus F^{(m)}$, in such a way that $F^{(1)}, \dots, F^{(m)}$ are attached to F by these operations. Notice that we do not demand here that $F, F^{(1)}, \dots, F^{(m)}$ are h -nearly embeddable in any surface.

Lemma 4.11 $\text{stc}(F) \leq \frac{h(h-1)}{2} \cdot \text{stc}(G)$.

Proof Suppose that Q_1, \dots, Q_m are the cliques in F used to attach the summands $F^{(1)}, \dots, F^{(m)}$. Assume without loss of generality that for each $i \in \{1, \dots, m\}$,

$F_i - Q_i$ is a connected graph, since otherwise it is possible to split this summand. Let T be a spanning tree of G such that $\text{cng}_G(T) = \text{stc}(G)$.

We construct a spanning tree T' of F by the following operations for each $i \in \{1, \dots, m\}$. Consider the forest $R = T[V(F^{(i)})]$. We contract all edges of $R - Q_i$ and reduce $R - Q_i$ to a set of independent vertices I adjacent to vertices of Q_i in T . Then for each vertex $u \in I$ adjacent to vertices $v_1, \dots, v_r \in Q_i$, we also contract edge $\{u, v_1\}$. Observe that edges $\{u, v_2\}, \dots, \{u, v_r\}$ are contracted to the edges $\{v_1, v_2\}, \dots, \{v_1, v_r\}$ of Q_i . For each edge $e = \{v_1, v_j\}$, we set $a(e) = \{u, v_j\} \in E(T)$. Thus we contracted all edges of $R - Q_i$ to edges of Q_i . Now we replace edges of $R - Q_i$ by these edges of Q_i . Then the constructed T' is a connected spanning subgraph of F , and T' has a cycle only if T does. Hence T' is a spanning tree of F . Finally, we set $a(e) = e$ for each $e \in E(T) \cap E(T')$.

We claim that $\text{cng}_F(T') \leq \frac{h(h-1)}{2} \cdot \text{cng}_G(T)$. To prove it, we consider an edge $e = \{u, v\} \in E(T')$. Let T'_1, T'_2 be the connected components of $T' - e$. By our construction of T' , sets $X_1 = V(T'_1)$ and $X_2 = V(T'_2)$ are subsets of the vertex sets of the components T_1 and T_2 of $T - a(e)$. Let $\{x, y\}$ be an edge of F such that $x \in X_1$ and $y \in X_2$. If $\{x, y\} \in E(G)$, then we set $l(\{x, y\}) = \{x, y\}$. Suppose that $\{x, y\} \notin E(G)$. Then x and y belong to some Q_i . Since $F_i - Q_i$ is a connected graph, there is an x - y -path in G with all internal vertices in $F_i - Q_i$. This path must contain an edge s with endpoints in T_1 and T_2 . We set $l(\{x, y\}) = s$. Now we assign the detour (as induced by T' in F) of the edge $\{x, y\}$ to the detour (as induced by T in G) of the edge $l(\{x, y\})$. Notice that detours for several edges of F can be assigned to one detour in G , but the number of such detours for each $l(\{x, y\})$ is at most the number of edges in Q_i . Therefore, $\text{cng}_{F,T'}(e) \leq \frac{h(h-1)}{2} \cdot \text{cng}_{G,T}(a(e))$, and $\text{cng}_F(T') \leq \frac{h(h-1)}{2} \cdot \text{cng}_G(T)$. □

Assume now additionally that F is h -nearly embedded in a surface Σ . Let X be the set of apices and suppose that $F - X = F_0 \cup F_1 \cup \dots \cup F_h$, where F_0 is embedded in Σ and F_1, \dots, F_h are the vortices. Suppose also that $\Delta_1, \dots, \Delta_h$ are the corresponding disks in Σ whose boundaries are used to attach the vortices. The proof of the following lemma is implicit in the proof of Theorem 3 in [17].

Lemma 4.12 (Dragan et al. [17]) *Let H be an apex graph, and suppose that H is not a minor of G . Then there is a positive constant $c_{H,h,\Sigma}$ which depends only on H, h , and Σ such that if F_0 contains a $(c_{H,h,\Sigma} \cdot r, c_{H,h,\Sigma} \cdot r)$ -wall as a topological minor, then there is a disk Δ in Σ such that*

- $\Delta \cap \Delta_i = \emptyset$ for $i \in \{1, \dots, h\}$,
- vertices of F embedded in Δ are not adjacent to apices,
- F contains a subdivided (r, r) -wall as a subgraph completely embedded in Δ .

We are ready to prove the main combinatorial result of this section.

Theorem 4.13 *Let H be an apex graph. For any H -minor-free graph G , $\text{stc}(G) = \Omega(\text{tw}(G))$.*

Proof Let G be an H -minor-free graph. By Proposition 4.7, there is an integer h , depending only on H , such that G can be obtained by h -clique-sums from graphs that can be h -nearly embedded in a surface Σ in which H cannot be embedded. Assume that $G = G^{(1)} \oplus \dots \oplus G^{(s)}$ is a representation of G . Let $F = G^{(i)}$ such that $\text{tw}(G^{(i)}) = \max\{\text{tw}(G^{(1)}), \dots, \text{tw}(G^{(s)})\}$ and suppose that $G = F \oplus F^{(1)} \oplus \dots \oplus F^{(m)}$, where $F^{(1)}, \dots, F^{(m)}$ are attached to F by h -clique-sum operations. By Lemma 4.8,

$$\text{tw}(G) \leq \text{tw}(F). \tag{1}$$

Now we fix an h -nearly-embedding of F in Σ . Let X be the set of apices and suppose that $F - X = F_0 \cup F_1 \cup \dots \cup F_h$, where F_0 is embedded in Σ and F_1, \dots, F_h are the vortices. Suppose also that $\Delta_1, \dots, \Delta_h$ are the corresponding disks in Σ , the boundaries of which are used to attach the vortices. By Lemmata 4.9 and 4.10, there is a positive constant c_1 such that

$$\text{tw}(F) \leq c_1 \cdot \text{tw}(F_0). \tag{2}$$

Since F_0 is embedded in Σ , by Proposition 4.4, there is a constant c_2 , such that if $\text{tw}(F_0) \geq c_2 \cdot r$, then F_0 contains an (r, r) -grid as a minor and hence an (r, r) -wall as a topological minor. We use Lemma 4.12 and conclude that there is a constant c_3 , such that if $\text{tw}(F_0) \geq c_3 \cdot r$, then there is a disk Δ in Σ such that F contains a subdivided (r, r) -wall as a subgraph completely embedded in Δ , vertices of F_0 embedded in Δ are not adjacent to apices, and Δ does not intersect disks in Σ to which the vortices are attached. We apply Lemma 4.1 to the graphs F_P, F^+ , where F_P is the subgraph of F induced by the vertices of F_0 embedded in Δ and F^+ is the subgraph induced by all other vertices of F and by the vertices of F_P lying on the boundary of the external face of this graph. By the lemma, there is a constant c_4 such that

$$\text{tw}(F_0) \leq c_4 \cdot \text{stc}(F). \tag{3}$$

According to Lemma 4.11, there is a constant c_5 for which

$$\text{stc}(F) \leq c_5 \cdot \text{stc}(G). \tag{4}$$

Finally, by putting together inequalities (1)–(4), we conclude that there is a constant C_H which depends only on H such that

$$\text{tw}(F) \leq C_H \cdot \text{stc}(G). \tag{5}$$

□

Combined with Lemma 3.2, Theorem 4.13 yields the main algorithmic result of this section.

Theorem 4.14 *Let H be a fixed apex graph. For every fixed k , k -STC is solvable in linear time on H -minor-free graphs (and hence on planar graphs and graphs of bounded genus).*

In conclusion of this section, we observe that our combinatorial bounds cannot be extended to more general classes of H -minor-free graphs, and thus the requirement of Theorem 4.13 that H is an apex graph is crucial. Indeed, let G be a graph obtained from an (r, r) -grid by adding a vertex u adjacent to all vertices of the grid. Consider the spanning tree T of G which contains all edges incident to u . It is easy to see that $\text{stc}(G) \leq \text{cng}_G(T) \leq 5$, but $\text{tw}(G) = r + 1$. In Sect. 7.3, we show that the algorithmic results cannot be extended as well, by showing that the problem becomes NP-complete for $k \geq 8$ on K_6 -minor-free graphs.

5 Graphs of Bounded Degree

In this section, we show that the treewidth of a graph of bounded degree is linear in its spanning tree congestion. This upper bound improves on an earlier bound by Kozawa, Otachi, and Yamazaki [28].

Theorem 5.1 *For any connected graph G , $\text{tw}(G) \leq \max\{\text{stc}(G), \Delta(G)(\text{stc}(G) - 1)/2\}$.*

Proof Let $k = \text{stc}(G)$ and $d = \Delta(G)$. Let T be a spanning tree of G such that $\text{cng}_G(T) = k$.

Let T' be obtained from T by subdividing each edge. We use a tree decomposition with T' as tree. To each node of T' , we associate the following bag. If the node is a vertex $v \in V(G)$, then put v in the bag. If the node is an edge $\{v, w\} \in E(T)$ (i.e., the node was obtained by subdividing the edge $\{v, w\}$), put v and w in the bag. Then, for every edge $\{v, w\} \notin E(T)$, select (arbitrarily) one endpoint, say v , and add v to all bags on the path from the bag of v to the bag of w , except the bag of w . This is easily seen to be a tree decomposition.

Now, the size of a bag that corresponds to a subdivided edge $\{v, w\}$ of T is at most $k + 1$: two for v and w , and one vertex for each of the at most $k - 1$ edges for which the detour goes through $\{v, w\}$. Consider now a vertex v of T . Each edge not on T whose detour uses v as intermediate vertex counts for the congestion of two of the edges incident to v in the spanning tree. For each incident edge of v , there are at most $k - 1$ edges not on the spanning tree that count for its congestion. Hence there are at most $d(k - 1)/2$ such edges. Thus the size of a bag that corresponds to a vertex is at most $d(k - 1)/2 + 1$; one vertex for each edge, and then one for v itself. \square

By putting together Lemma 3.2 and Theorem 5.1, we obtain the following.

Theorem 5.2 *For every fixed k and Δ , k -STC is solvable in linear time on graphs with vertex degrees at most Δ .*

We remark that the bound in Theorem 5.1 is tight. It is tight on cycles, which have degree, spanning tree congestion, and treewidth all equal to two. Furthermore, any upper bound must depend at least linearly on the spanning tree congestion. It is known that $n \times n$ grids have bounded maximum degree, treewidth n , and spanning

tree congestion n [6, 27]. Finally, any upper bound must also depend at least linearly on the maximum degree. Grohe and Marx [25] show that a graph family based on expanders exists in which each member has degree at most three and treewidth linear in the number of vertices of the graph.

Proposition 5.3 *Let G be a graph and let G' be obtained from G by adding a vertex v adjacent to each vertex of G . Then $\text{tw}(G) \leq \text{tw}(G') \leq \text{tw}(G) + 1$ and $\text{stc}(G') \leq \Delta(G) + 1$.*

Proof By adding v to each bag of a tree decomposition, $\text{tw}(G') \leq \text{tw}(G) + 1$. As G is a minor of G' , $\text{tw}(G) \leq \text{tw}(G')$. A spanning tree isomorphic to $K_{1,|V(G)|}$ with v at its center has congestion $\Delta(G) + 1$. □

Using the above proposition and the family of Grohe and Marx, we obtain a family of graphs of treewidth and maximum degree linear in the number of vertices of the graph and spanning tree congestion at most four. These facts give strong evidence for the tightness of our bound.

6 Linear Time Solvability of k -STC for $1 \leq k \leq 3$

In this section, we show that k -STC can be solved in linear time for $1 \leq k \leq 3$. First, we give characterizations for graphs of spanning tree congestion one and two.

Theorem 6.1 *For a connected graph G , $\text{stc}(G) = 1$ if and only if G is a tree.*

Proof If G is a tree, then clearly $\text{stc}(G) = 1$. Assume G has a cycle C . Then, for any two vertices in C , G has two edge disjoint paths between them. Thus, by Lemma 2.3, G cannot have any cycle. □

A graph G is a *cactus graph* if no two cycles in G have a common edge.

Theorem 6.2 *For a connected graph G , $\text{stc}(G) = 2$ if and only if G is not a tree but a cactus graph.*

Proof Clearly, every biconnected component of a cactus graph G is either a cycle or a single edge, and thus G has spanning tree congestion at most two. It is easy to verify that a biconnected graph G has no vertex pair u, v such that G contains three edge disjoint $u-v$ paths if and only if G is either a cycle or a single edge. Thus, from Proposition 2.2 and Lemma 2.3, the theorem holds. □

Obviously, recognizing trees and cactus graphs can be done in linear time, by using depth-first search (see e.g. [7]). For $k = 3$, we need the following lemma.

Lemma 6.3 *For a graph G , if $\text{stc}(G) \leq 3$, then G is planar.*

Proof Suppose that T is a spanning tree of G such that $\text{cng}_G(T) \leq 3$. Recall that for every edge $e \in E(G) \setminus E(T)$, there is a unique cycle C_e in the graph obtained from T by adding e , and these cycles C_e are called the *fundamental cycles* of G with respect to T . Denote by \mathcal{F} the set of fundamental cycles. The set \mathcal{F} is a *base* of the *cycle space* of G (see e.g. [16]). Since $\text{cng}_G(T) \leq 3$, each edge $e \in E(G)$ is contained in at most two cycles from \mathcal{F} . That is, \mathcal{F} is *simple*. By MacLane’s Theorem [34], a graph is planar if and only if its cycle space has a simple basis. Thus G is planar. \square

From Lemmata 4.3 and 6.3, if $\text{stc}(G) \leq 3$, then $\text{tw}(G) \leq 72$, and by Lemma 3.2 we have the following theorem.

Theorem 6.4 *For $1 \leq k \leq 3$, k -STC can be solved in linear time.*

Observe that we cannot hope to extend these results to $k \geq 4$ in a similar way, since the treewidth of graphs of spanning tree congestion at most 4 is not bounded. Consider, for example, the graphs obtained from a wall by adding a vertex adjacent to all vertices of the wall. All such graphs have spanning tree congestion at most 4, but their treewidth can be arbitrary large.

7 Hardness

In this section, we provide NP-hardness proofs, complementing our algorithmic results.

7.1 Spanning Tree Congestion of Planar Graphs

We proved that k -STC can be solved on planar, and more generally, on apex-minor-free graphs for every fixed k in linear time. In this section, we prove that the problem cannot be solved in polynomial time on planar graphs if k is a part of the input, unless $P = NP$. Our result follows from some known results for the tree spanner problem.

Let G be a graph and T a spanning tree of G . If $\text{dist}_T(u, v) \leq k$ for any $\{u, v\} \in E(G)$, then T is a *tree k -spanner* [5]. We denote by $\text{tsp}(G)$ the minimum number k such that G has a tree k -spanner. For planar graphs, the following results are known.

Proposition 7.1 (Fekete and Kremer [21]) *It is NP-complete to decide whether $\text{tsp}(G) \leq k$ for planar graphs G and integers k .*

Since a cut in G corresponds to a cycle in G^* , the following relation holds.

Proposition 7.2 (Fekete and Kremer [21]) *For any planar graph G , $\text{stc}(G) = \text{tsp}(G^*) + 1$.*

A planar embedding of a planar graph can be constructed in linear time by an algorithm proposed by Hopcroft and Tarjan [26]. From a planar embedding of a planar graph G , we can easily construct geometrically a dual graph G^* (see e.g. [31]).

Thus, from Propositions 7.1 and 7.2, we can conclude the following.

Theorem 7.3 *It is NP-complete to decide whether $\text{stc}(G) \leq k$ for planar graphs G and integers k .*

7.2 Weighted k -STC is NP-Complete for $k \geq 8$

In this section, we prove the following hardness result for the weighted variant of the k -STC problem with positive integer weights.

Theorem 7.4 *For any fixed $k \geq 8$, k -STC is NP-complete for edge-weighted apex graphs with at most one vertex of degree greater than 4 and for which the weights of the edges are at most k .*

Clearly, the problem belongs to NP. To show NP-completeness, we present a reduction from a variant of the Planar 3-Satisfiability problem (PLANAR 3-SAT), which is a well-known NP-complete problem [32]. We consider the version of this problem introduced by Dalhouse et al. [9], see also [29]. An instance (U, C) of this problem consists of a set U of n distinct Boolean variables and a collection C of m clauses such that each clause has two or three literals, each variable occurs exactly once in positive and exactly twice in negation, and the graph with the vertex set $U \cup C$ such that a variable u_i and a clause c_j are adjacent if and only if u_i occurs in C_j , is planar. In what follows, we assume that instances of PLANAR 3-SAT satisfy these conditions, but we also need an additional condition. For a set of variables $X \subseteq U$, we denote by $C(X) \subseteq C$ the set of clauses containing literals on variables from X .

Lemma 7.5 *The PLANAR 3-SAT problem is NP-complete for instances (U, C) such that $|C(X)| \geq 4$ for every $X \subseteq U$ with $2 \leq |X| \leq 3$.*

Proof For the proof of the lemma, it is sufficient to observe that if $|C(X)| \leq 3$ then it is always possible to assign values to the variables in X in such a way that all clauses in $C(X)$ are satisfied. Hence we can exclude these variables and clauses from the instance. □

The constructions in our proof are inspired by the proof of Cai and Corneil [5] for the NP-completeness of the Weighted Tree Spanner problem. Let $k \geq 8$ be a fixed integer. For an arbitrary instance (U, C) of PLANAR 3-SAT such that $|C(X)| \geq 4$ for every $X \subseteq U$ with $2 \leq |X| \leq 3$, we construct an edge-weighted graph G_C such that C is satisfiable if and only if $\text{stc}(G_C) \leq k$. Let $a = \lceil k/2 \rceil + 1$, $b_1 = \lfloor k/2 \rfloor - 2$, $b_2 = \lfloor k/2 \rfloor - 3$ and $c = k - 1$. Observe that for $k \geq 8$, a, b_1, b_2 and c are positive integers. Each edge in G_C has a weight which will be either a, b_1, b_2, c , or 1.

From an instance (U, C) of PLANAR 3-SAT, the graph G_C is constructed as follows (see Fig. 4):

1. Take a vertex x , literal vertices u_i and \bar{u}_i for each variable $u_i \in U$, and clause vertices c_i for each clause $c_i \in C$.
2. Connect x to all literal vertices u_i by positive literal edges of weight b_1 and connect x to all literal vertices \bar{u}_i by negative literal edges of weight b_2 .

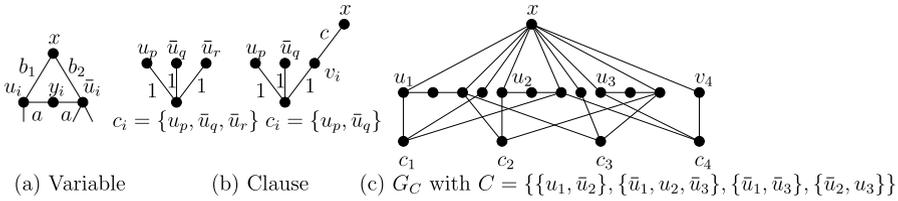


Fig. 4 Gadgets, and a constructed graph

3. For each variable $u_i \in U$, create a path of length two between u_i and \bar{u}_i such that edges in the path, which are called *bridge edges*, have weight a , and the center vertex of the path is a new vertex y_i .
4. For each clause $c_i = \{l_p, l_q, l_r\} \in C$ with 3 literals, connect the clause vertex c_i to the literal vertices l_p, l_q , and l_r by *clause edges* of unit weight.
5. For each clause $c_i = \{l_p, l_q\} \in C$ with 2 literals, add a *complement vertex* v_i , join it with x by a *complement edge* of weight c , and then connect the clause vertex c_i to the literal vertices l_p, l_q by *clause edges* of unit weight and to the complement vertex v_i by a *complement clause edge* of weight one.

Clearly, the above construction can be done in polynomial time. Observe that $G_C - x$ is planar and that for any vertex $w \in V(G_C)$, $\deg(w) \leq 4$ if $w \neq x$.

Now, we show the following useful properties of a spanning tree of G_C with small congestion.

Lemma 7.6 *Let T be a spanning tree of G_C . If $\text{cng}_{G_C}(T) \leq k$, then*

1. *All bridge edges are contained in T ;*
2. *All complement edges are contained in T ;*
3. *All complement clause edges are not included in T ;*
4. *Each clause vertex is a leaf of T ;*
5. *For each variable, exactly one of its two literal edges is contained in T .*

Proof of Property 1 Since y_i has degree two, at least one of $\{u_i, y_i\}$ and $\{\bar{u}_i, y_i\}$ must be in T . If $\{\bar{u}_i, y_i\}$ is not in T , then $\text{cng}_{G_C, T}(\{u_i, y_i\}) = w(\theta(\{y_i\})) = 2a > k$. The other case is almost the same. □

Proof of Property 2 Assume T has the first property. Suppose that a complement edge $\{x, v_i\} \notin E(T)$. Consider an $x-v_i$ -path in T . Let e be the edge of the path incident to x . If e is a complement edge, then $\text{cng}_{G_C, T}(e) \geq 2c = 2k - 2$. If e is a literal edge $\{x, u_j\}$ or $\{x, \bar{u}_j\}$, then, since the bridge edges $\{u_j, y_j\}, \{y_j, \bar{u}_j\}$ are in $E(T)$, $\text{cng}_{G_C, T}(e) \geq c + b_1 + b_2 > k$. We get a contradiction in both cases and hence the property holds. □

Proof of Property 3 Assume T has the first and the second property. Suppose that, contrarily to our claim, there is a complement clause edge $\{c_j, v_i\} \in E(T)$. Let $e = \{x, v_i\}$. If c_j is a leaf of T , then $\text{cng}_{G_C, T}(e) \geq c + 2 > k$. Suppose that $\deg_T(c_j) \geq 2$

and assume that a literal vertex l_r is adjacent to c_j in T . Since the bridge edges $\{u_r, y_r\}, \{y_r, \bar{u}_r\} \in E(T)$, $\text{cng}_{G,T}(e) \geq c + b_1 + b_2 > k$. \square

Proof of Property 4 Assume T has the first three properties. By way of contradiction, suppose some clause vertices have degree larger than one in T . We divide the proof into several cases. Recall that all bridge edges are in T from the first property.

Case 1: There is a clause vertex c_i with $\text{deg}_T(c_i) = 3$. Since complement clause edges are not included in T , the three neighbors of c_i in T are some literal vertices l_p, l_q , and l_r . Since $|C(u_p, u_q, u_r)| \geq 4$, there is a clause vertex c_j for which $j \neq i$, $c_j \in C(u_p, u_q, u_r)$, and $N(c_j) \setminus \{u_p, \bar{u}_p, u_q, \bar{u}_q, u_r, \bar{u}_r\} \neq \emptyset$. Let e be the unique literal edge in the unique c_i - x path in T . Suppose that c_j is not adjacent to vertices $u_p, \bar{u}_p, u_q, \bar{u}_q, u_r, \bar{u}_r$ in T . Then, e separates $\{x, c_j\}$ and $\{u_p, \bar{u}_p, u_q, \bar{u}_q, u_r, \bar{u}_r\}$. Thus, $\text{cng}_{G_C,T}(e) \geq w(E(\{x, c_j\}, \{u_p, \bar{u}_p, u_q, \bar{u}_q, u_r, \bar{u}_r\})) = 3(b_1 + b_2) + 1 > k$. Assume now that c_j is adjacent to one vertex from the set $\{u_p, \bar{u}_p, u_q, \bar{u}_q, u_r, \bar{u}_r\}$ in T . If c_j is also adjacent to some literal vertex u_s or \bar{u}_s for $s \neq p, q, r$, then we conclude that $\text{cng}_{G_C,T}(e) \geq 4(b_1 + b_2) > k$. Otherwise, $\text{cng}_{G_C,T}(e) \geq 3(b_1 + b_2) + 1 > k$. From now on, we assume that no clause vertex has degree three in T .

Case 2: There are two clause vertices c_i, c_j with $\text{deg}_T(c_i) = 2, \text{deg}_T(c_j) = 2$, and c_i is adjacent to some literal vertices l_p, l_q and c_j is adjacent to l_q, l_r . The arguments in this case are almost the same as in Case 1. Since $|C(u_p, u_q, u_r)| \geq 4$, there is a clause vertex c_s such that $s \neq i, j$, $c_s \in C(u_p, u_q, u_r)$ and $N(c_s) \setminus \{u_p, \bar{u}_p, u_q, \bar{u}_q, u_r, \bar{u}_r\} \neq \emptyset$. Let e be the literal edge in the c_i - x path in T . Suppose that c_s is not adjacent to vertices $u_p, \bar{u}_p, u_q, \bar{u}_q, u_r, \bar{u}_r$ in T . Then, $\text{cng}_{G_C,T}(e) \geq w(E(\{x, c_s\}, \{u_p, \bar{u}_p, u_q, \bar{u}_q, u_r, \bar{u}_r\})) = 3(b_1 + b_2) + 1 > k$. Assume now that c_s is adjacent to one vertex from the set $\{u_p, \bar{u}_p, u_q, \bar{u}_q, u_r, \bar{u}_r\}$ in T . If c_s is adjacent to some literal vertex u_t or \bar{u}_t , then we conclude that $\text{cng}_{G_C,T}(e) \geq 4(b_1 + b_2) > k$. Otherwise, $\text{cng}_{G_C,T}(e) \geq 3(b_1 + b_2) + 1 > k$. From now, we assume that there are no such clause vertices c_i, c_j .

Case 3: There is a clause vertex c_i with $\text{deg}_T(c_i) = 2$. Assume that the two neighbors of c_i in T are l_p and l_q . Let e be the literal edge in the c_i - x path in T . Since $|C(u_p, u_q)| \geq 4$, there are clauses $c_j, c_s, c_t \in C(u_p, u_q, u_r)$, $i \neq j, s, t$. Each vertex $c \in \{c_i, c_j, c_s, c_t\}$ is either not adjacent in T to vertices $u_p, \bar{u}_p, u_q, \bar{u}_q$, or adjacent in G_C to a vertex w such that $w \notin \{u_p, \bar{u}_p, u_q, \bar{u}_q\}$ and $\{c, w\} \notin E(T)$. In both cases, $\text{cng}_{G_C,T}(e) \geq 2(b_1 + b_2) + 4 > k$. \square

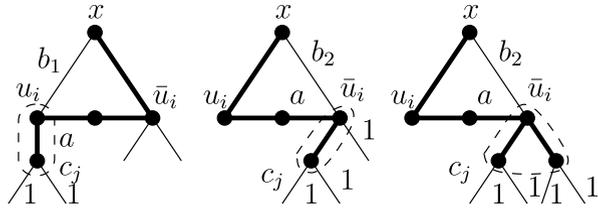
Proof of Property 5 Assume T has the first four properties. Since T is a tree and contains all bridge edges, at most one of $\{x, u_i\}$ and $\{x, \bar{u}_i\}$ can be in T for each $u_i \in U$. Suppose T contains none of them. Since any clause vertex is a leaf of T , there is no path between u_i and x . \square

The next two lemmata show that C is satisfiable if and only if $\text{stc}(G_C) \leq k$, thus proving Theorem 7.4.

Lemma 7.7 *If $\text{stc}(G_C) \leq k$, then C is satisfiable.*

Proof Let T be a spanning tree of G_C such that $\text{cng}_{G_C}(T) \leq k$. From Lemma 7.6,

Fig. 5 Unsatisfied clauses



1. T contains all bridge edges,
2. T contains exactly one literal edge for each variable, and
3. every clause vertex is a leaf of T .

From the second property, we can define a truth assignment ξ_T by setting $\xi_T(u_i) = \mathbf{true}$ if $\{x, u_i\} \in E(T)$ and $\xi_T(u_i) = \mathbf{false}$ if $\{x, \bar{u}_i\} \in E(T)$. We show that ξ_T satisfies C . It suffices to show that for every $c_j \in C$, the unique neighbor l_i in T of c_j is adjacent to x . If l_i is not adjacent to x , then either $\text{cng}_{G_C, T}(\{l_i, y_i\}) \geq a + b_1 + 2 > k$ or $\text{cng}_{G_C, T}(\{l_i, y_i\}) \geq a + b_2 + 3 > k$ (see Fig. 5). This contradicts $\text{cng}_{G_C}(T) \leq k$. \square

Lemma 7.8 *If C is satisfiable, then $\text{stc}(G_C) \leq k$.*

Proof Let ξ be a satisfying truth assignment for C . We say that a literal vertex l_i is a **true vertex** if l_i becomes **true** by the assignment ξ . We construct a spanning tree T of G_C as follows:

1. Take all complement edges.
2. Take all bridge edges.
3. Take all literal edges incident to true vertices.
4. For each clause, take an arbitrary clause edge incident to a true vertex.

Clearly, T is a spanning tree of G_C . We show that $\text{cng}_{G_C}(T) \leq k$.

Suppose that v_i is a complement vertex and $e = \{x, v_i\}$. It is easy to see that $\text{cng}_{G_C, T}(e) = c + 1 = k$. Let $u_i \in U$.

Suppose that $\{x, u_i\} \in E(T)$. Then T contains edges $\{x, u_i\}$ and $\{u_i, y_i\}, \{\bar{u}_i, y_i\}$. From the construction of T , it follows that T may contain the clause edge incident to u_i , but cannot contain any clause edge incident to \bar{u}_i . See Fig. 6. The edge $\{u_i, y_i\}$ and $\{\bar{u}_i, y_i\}$ have the same congestion, and $\text{cng}_{G_C, T}(\{\bar{u}_i, y_i\}) = w(\theta(\{\bar{u}_i\})) = a + b_2 + 2 = k$. If a clause edge incident to u_i is contained in T , then the edge has congestion $3 \leq k$. For the literal edge $\{x, u_i\}$, $\text{cng}_{G_C, T}(\{x, u_i\}) \leq b_1 + b_2 + 4 \leq k$ (see Fig. 6).

Assume now that $\{x, \bar{u}_i\} \in E(T)$. Then T contains edges $\{x, u_i\}$ and $\{u_i, y_i\}, \{\bar{u}_i, y_i\}$. From the construction of T , we have that T cannot contain the clause edge incident to u_i , but may contain clause edges incident to \bar{u}_i . See Fig. 6. The edge $\{u_i, y_i\}$ and $\{\bar{u}_i, y_i\}$ have the same congestion, and $\text{cng}_{G_C, T}(\{\bar{u}_i, y_i\}) = w(\theta(\{\bar{u}_i\})) = a + b_1 + 1 = k$. If a clause edge incident to \bar{u}_i is contained in T , then the edge has congestion $3 \leq k$. For the literal edge $\{x, \bar{u}_i\}$, $\text{cng}_{G_C, T}(\{x, \bar{u}_i\}) \leq b_1 + b_2 + 5 \leq k$. \square

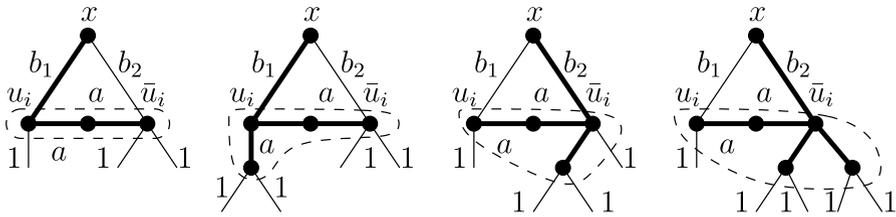


Fig. 6 A spanning tree of congestion at most k

7.3 Unweighted k -STC is NP-Complete for $k \geq 8$

Extending the result in the previous section, we prove the main hardness result of the paper, that is, the NP-completeness of k -STC on unweighted H -minor free graphs. We need the following two lemmata.

Lemma 7.9 *An edge e of weight $w \in \mathbf{Z}^+$ can be replaced by w parallel edges of unit weight without changing the spanning tree congestion.*

Proof Let G be an edge-weighted graph, and let $e = \{u, v\} \in E(G)$ be an edge of integral weight $w \geq 2$. We denote by G' the graph obtained from G by the deletion of e and the addition of w parallel edges e_1, \dots, e_w of unit weight between u and v . Clearly, any spanning tree of G' contains at most one of e_1, \dots, e_w . Without loss of generality, we assume that any spanning tree T' of G' contains only e_1 from $\{e_1, \dots, e_w\}$. By this assumption, we have a bijective correspondence between the spanning trees of G and the spanning trees of G' ; we simply identify e and e_1 .

Let T be a spanning tree of G , and let T' be the corresponding spanning tree of G' . Let $P_T = \{(A_f, B_f) \mid f \in E(T)\}$ and $P_{T'} = \{(A_f, B_f) \mid f \in E(T')\}$ denote the set of the partitions of $V(G)$ defined by edges in T and T' , respectively. It is not difficult to see that $P_T = P_{T'}$. By definition, $\text{cng}_G(T) = \max_{(A,B) \in P_T} w(E_G(A, B))$ and $\text{cng}_{G'}(T') = \max_{(A,B) \in P_{T'}} w(E_{G'}(A, B))$. If e is not between A and B , then $w(E_G(A, B)) = w(E_{G'}(A, B))$. Otherwise, $E_G(A, B) \setminus \{e\} = E_{G'}(A, B) \setminus \{e_1, \dots, e_w\}$, and thus,

$$\begin{aligned} w(E_G(A, B)) &= w(E_G(A, B) \setminus \{e\}) + w(e) = w(E_G(A, B) \setminus \{e\}) + w \\ &= w(E_{G'}(A, B) \setminus \{e_1, \dots, e_w\}) + |\{e_1, \dots, e_w\}| = w(E_{G'}(A, B)). \end{aligned}$$

Therefore, $\text{cng}_G(T) = \text{cng}_{G'}(T')$, and hence, $\text{stc}(G) = \text{stc}(G')$. □

Lemma 7.10 *Edge subdivisions do not change the spanning tree congestion of unweighted graphs.*

Proof Let G be a graph without edge weights, and let $e = \{u, v\} \in E(G)$. We denote by G' the graph obtained from G by the deletion of e , and the addition of a vertex w and two edges $e_1 = \{u, w\}$ and $e_2 = \{w, v\}$. Clearly, any spanning tree of G' contains at least one of e_1 and e_2 . Without loss of generality, we assume for any spanning

tree T' of G' , $e_2 \in E(T')$. By this assumption, we have a bijective correspondence between the spanning trees of G and the spanning trees of G' ; we identify e and e_1 , and ignore e_2 .

If $\text{stc}(G) = 1$, then G is a tree. Clearly, G' is also a tree. This implies $\text{stc}(G) = \text{stc}(G') = 1$. Now assume that $\text{stc}(G) \geq 2$. Let T be a spanning tree of G , and T' the corresponding spanning tree of G' . Clearly, if $e_1 \in E(T')$, then $\text{cng}_{G',T'}(e_1) = \text{cng}_{G',T'}(e_2)$; otherwise $\text{cng}_{G',T'}(e_2) = |\theta(\{w\})| = 2 \leq \text{stc}(G) \leq \text{cng}_G(T)$. It is easy to see that $\text{cng}_{G,T}(e) = \text{cng}_{G',T'}(e_1)$ if $e \in E(T)$, and $\text{cng}_{G,T}(f) = \text{cng}_{G',T'}(f)$ for any $f \in E(T) \setminus \{e\} = E(T') \setminus \{e_1, e_2\}$. Therefore, $\text{cng}_G(T) = \text{cng}_{G'}(T')$, and hence, $\text{stc}(G) = \text{stc}(G')$. \square

Combining the above two lemmata, we can conclude that an edge $\{u, v\}$ of weight w can be replaced by w internally disjoint $u-v$ paths of length two that consist of unweighted edges, without changing the spanning tree congestion. It is easy to see that this replacement can be done in $O(w)$ time. Thus, we have the following corollary.

Corollary 7.11 *Let G be an edge-weighted graph such that the weight of every edge of G is a positive integer, and the maximum weight of the edges is w . Then, in $O(w \cdot |E(G)|)$ time, G can be transformed into an unweighted simple graph G' for which $\text{stc}(G') = \text{stc}(G)$.*

Now, we prove the main theorem of this section.

Theorem 7.12 *For any fixed $k \geq 8$, k -STC is NP-complete on simple unweighted apex graphs that have at most one vertex of unbounded degree.*

Proof By Theorem 7.4, for any fixed $k \geq 8$, k -STC is NP-complete on edge-weighted apex graphs with at most one vertex of degree greater than 4, such that the positive integer weights of their edges are at most k . Let G be a weighted apex graph with at most one vertex of degree greater than 4, such that the weights of its edges are at most k . From Corollary 7.11, we can construct a simple unweighted graph G'_C in polynomial time for which $\text{stc}(G'_C) = \text{stc}(G_C)$. Clearly, G'_C is also an apex graph and it has at most one vertex of degree greater than $4k$. \square

We remark that no apex graph contains a clique K_6 on six vertices, and thus by Theorem 7.12, the problem is NP-complete for $k \geq 8$ on K_6 -minor-free graphs with at most one vertex of unbounded degree.

8 Concluding Remarks

We have proved that for fixed k , the problem of determining whether the spanning tree congestion of a given graph is at most k is solvable in linear time on apex-minor-free graphs and on graphs of bounded degree. We also showed that the problem can be solved in linear time on any graph if $1 \leq k \leq 3$. On the other hand, we showed that

if the input graph has one vertex of unbounded degree, then the problem becomes NP-complete for $k \geq 8$. The complexity of k -STC remains open for $4 \leq k \leq 7$.

Let us remark that our parameterized algorithms can be easily extended to weighted graphs with positive integer edge weights due to the fact that all edge weights in any “YES” instance are at most k .

Graphs of bounded degree and apex-minor-free graphs are graphs of bounded local treewidth. An interesting open problem is whether k -STC is fixed-parameter tractable on graphs of bounded local treewidth. Since the tree spanner problem is NP-hard on chordal graphs [3] and on chordal bipartite graphs [4], it would be interesting to determine the complexity of STC or k -STC for these graph classes.

Finally, we remark that our NP-hardness proof of 8-STC yields the following constant lower bound on the approximability of STC. We say that a polynomial-time algorithm for spanning tree congestion is a c_1 -approximation algorithm for a positive number c_1 if there is a positive integer c_2 such that for any input graph G , the output k of the algorithm satisfies $k \leq c_1 \cdot \text{stc}(G) + c_2$.

Theorem 8.1 *There is no polynomial-time c_1 -approximation algorithm for the spanning tree congestion of simple unweighted graphs such that $c_1 < 9/8$, unless $P = NP$.*

Proof Suppose there is a polynomial-time c_1 -approximation algorithm A for the spanning tree congestion of simple unweighted graphs with $c_1 < 9/8$. Let c_2 be the constant additive of A , that is, the output $A(G)$ of A for any graph G satisfies $A(G) \leq c_1 \cdot \text{stc}(G) + c_2$. Let t be the smallest positive integer that satisfies $(9/8 - c_1) \cdot t > c_2$.

Let G be a weighted graph for which the weight of its edges is at most $k = 8$. We denote by G' the graph obtained from G by setting the edge weights to $w_{G'}(e) = t \cdot w_G(e)$. Clearly, $\text{stc}(G') = t \cdot \text{stc}(G)$. Let G'' be the simple unweighted graph obtained from G' by Corollary 7.11. Then $\text{stc}(G'') = \text{stc}(G') = t \cdot \text{stc}(G_C)$.

Claim 8.2 $A(G'') < 9t$ if and only if $\text{stc}(G_C) \leq 8$.

Proof First, assume that $A(G'') < 9t$. Then $t \cdot \text{stc}(G) = \text{stc}(G'') \leq A(G'') < 9t$. Thus, we have $\text{stc}(G) < 9$, which implies that $\text{stc}(G) \leq 8$. Next, assume that $\text{stc}(G) \leq 8$. Then

$$\begin{aligned} A(G'') &\leq c_1 \cdot \text{stc}(G'') + c_2 = c_1 \cdot t \cdot \text{stc}(G) + c_2 \\ &= 9/8 \cdot t \cdot \text{stc}(G) - (9/8 - c_1) \cdot t \cdot \text{stc}(G) + c_2. \end{aligned}$$

Since $\text{stc}(G) \leq 8$ and $(9/8 - c_1) \cdot t > c_2$, we have

$$A(G'') < 9t - c_2(\text{stc}(G) - 1) \leq 9t. \quad \square$$

From the above claim, we can use A as a polynomial-time algorithm for k -STC on weighted graphs with positive integer weights of edges for $k = 8$. But this problem is NP-hard. Hence such an algorithm cannot exist, unless $P = NP$. □

Acknowledgements We are grateful to Agelos Georgakopoulos for pointing out the short proof of Lemma 6.3. Research of Fedor Fomin was supported by the European Research Council (ERC) grant “Rigorous Theory of Preprocessing”, reference 267959. Petr Golovach was supported by EPSRC under project EP/G043434/1.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

1. Bodlaender, H.L.: A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.* **25**, 1305–1317 (1996)
2. Borie, R.B., Parker, R.G., Tovey, C.A.: Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families. *Algorithmica* **7**, 555–581 (1992)
3. Brandstädt, A., Dragan, F.F., Le, H.-O., Le, V.B.: Tree spanners on chordal graphs: complexity and algorithms. *Theor. Comput. Sci.* **310**, 329–354 (2004)
4. Brandstädt, A., Dragan, F.F., Le, H.-O., Le, V.B., Uehara, R.: Tree spanners for bipartite graphs and probe interval graphs. *Algorithmica* **47**, 27–51 (2007)
5. Cai, L., Corneil, D.G.: Tree spanners. *SIAM J. Discrete Math.* **8**, 359–387 (1995)
6. Castejón, A., Ostrovskii, M.I.: Minimum congestion spanning trees of grids and discrete toruses. *Discuss. Math., Graph Theory* **29**, 511–519 (2009)
7. Cormen, T., Leiserson, C.E., Rivest, R.L., Stein, C.: *Introduction to Algorithms*, 3rd edn. MIT Press, Cambridge (2009)
8. Courcelle, B.: The monadic second-order logic of graphs III: Tree-decompositions, minor and complexity issues. *Theor. Inform. Appl.* **26**, 257–286 (1992)
9. Dahlhaus, E., Johnson, D.S., Papadimitriou, C.H., Seymour, P.D., Yannakakis, M.: The complexity of multiterminal cuts. *SIAM J. Comput.* **23**, 864–894 (1994)
10. Demaine, E.D., Fomin, F.V., Hajiaghayi, M., Thilikos, D.M.: Subexponential parameterized algorithms on graphs of bounded genus and H -minor-free graphs. *J. ACM* **52**, 866–893 (2005)
11. Demaine, E.D., Fomin, F.V., Hajiaghayi, M., Thilikos, D.M.: Bidimensional parameters and local treewidth. *SIAM J. Discrete Math.* **18**(3), 501–511 (2004)
12. Demaine, E.D., Hajiaghayi, M.: Graphs excluding a fixed minor have grids as large as treewidth, with combinatorial and algorithmic applications through bidimensionality. In: *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2005)*, pp. 682–689. SIAM, Philadelphia (2005)
13. Demaine, E.D., Hajiaghayi, M.: Linearity of grid minors in treewidth with applications through bidimensionality. *Combinatorica* **28**, 19–36 (2008)
14. Demaine, E.D., Hajiaghayi, M.T., Kawarabayashi, K.: Algorithmic graph minor theory: Decomposition, approximation, and coloring. In: *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005)*, pp. 637–646. IEEE Computer Society, Los Alamitos (2005)
15. Demaine, E.D., Hajiaghayi, M.T., Kawarabayashi, K.: Approximation algorithms via structural results for apex-minor-free graphs. In: *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP 2009)*. Lecture Notes in Computer Science, vol. 5555, pp. 316–327. Springer, Berlin (2009)
16. Diestel, R.: *Graph Theory*, 3rd edn. Springer, Berlin (2005)
17. Dragan, F.F., Fomin, F.V., Golovach, P.A.: Spanners in sparse graphs, *Journal of Computer and System Sciences*, doi:[10.1016/j.jcss.2010.10.002](https://doi.org/10.1016/j.jcss.2010.10.002)
18. Dragan, F.F., Fomin, F.V., Golovach, P.A.: Approximation of minimum weight spanners for sparse graphs. *Theor. Comput. Sci.* **412**, 846–852 (2011)
19. Downey, R.G., Fellows, M.R.: *Parameterized Complexity*. Springer, New York (1999)
20. Eppstein, D.: Diameter and treewidth in minor-closed graph families. *Algorithmica* **27**, 275–291 (2000)
21. Fekete, S.P., Kremer, J.: Tree spanners in planar graphs. *Discrete Appl. Math.* **108**, 85–103 (2001)

22. Fomin, F.V., Golovach, P.A., van Leeuwen, E.J.: Spanners of bounded degree graphs. *Inf. Process. Lett.* **111**, 142–144 (2011)
23. Geelen, J.F., Richter, R.B., Salazar, G.: Embedding grids in surfaces. *Eur. J. Comb.* **25**, 785–792 (2004)
24. Grohe, M.: Local tree-width, excluded minors, and approximation algorithms. *Combinatorica* **23**, 613–632 (2003)
25. Grohe, M., Marx, D.: On tree width, bramble size, and expansion. *J. Comb. Theory, Ser. B* **99**, 218–228 (2009)
26. Hopcroft, J., Tarjan, R.: Efficient planarity testing. *J. ACM* **21**, 549–568 (1974)
27. Hruska, S.W.: On tree congestion of graphs. *Discrete Math.* **308**, 1801–1809 (2008)
28. Kozawa, K., Otachi, Y., Yamazaki, K.: On spanning tree congestion of graphs. *Discrete Math.* **309**, 4215–4224 (2009)
29. Kratochvíl, J.: A special planar satisfiability problem and a consequence of its NP-completeness. *Discrete Appl. Math.* **52**, 233–252 (1994)
30. Law, H.-F.: Spanning tree congestion of the hypercube. *Discrete Math.* **309**, 6644–6648 (2009)
31. Lawler, E.: *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart and Winston, New York (1976)
32. Lichtenstein, D.: Planar formulae and their uses. *SIAM J. Comput.* **11**, 329–343 (1982)
33. Löwenstein, C., Rautenbach, D., Regen, F.: On spanning tree congestion. *Discrete Math.* **309**, 4653–4655 (2009)
34. MacLane, S.: A combinatorial condition for planar graphs. *Fundam. Math.* **28**, 22–32 (1937)
35. Mohar, B.: Combinatorial local planarity and the width of graph embeddings. *Can. J. Math.* **44**, 1272–1288 (1992)
36. Mohar, B., Thomassen, C.: *Graphs on Surfaces*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore (2001)
37. Okamoto, Y., Otachi, Y., Uehara, R., Uno, T.: Hardness results and an exact exponential algorithm for the spanning tree congestion problem. In: *Proceedings of the 8th Annual Conference on Theory and Applications of Models of Computation (TAMC 2011)*. Lecture Notes in Comput. Sci., vol. 6648, pp. 452–462 (2011)
38. Ostrovskii, M.I.: Minimal congestion trees. *Discrete Math.* **285**, 219–226 (2004)
39. Ostrovskii, M.I.: Minimum congestion spanning trees in planar graphs. *Discrete Math.* **310**, 1204–1209 (2010)
40. Otachi, Y., Bodlaender, H.L., van Leeuwen, E.J.: Complexity results for the spanning tree congestion problem. In: *Proceedings of the 6th International Workshop on Graph Theoretic Concepts in Computer Science (WG 2010)*. Lecture Notes in Comput. Sci., vol. 6410, pp. 3–14 (2010)
41. Peleg, D.: Low stretch spanning trees. In: *Proceedings of the 27th International Symposium on Mathematical Foundations of Computer Science (MFCS 2002)*. Lecture Notes in Comput. Sci., vol. 2420, pp. 68–80 (2002)
42. Raspaud, A., Sýkora, O., Vrťo, I.: Congestion and dilation, similarities and differences: a survey. In: *Proceedings of the 7th International Colloquium on Structural Information and Communication Complexity (SIROCCO 2000)*, pp. 269–280. Carleton Scientific, Oxford (2000)
43. Robertson, N., Seymour, P.D.: Graph minors. X. Obstructions to tree-decomposition. *J. Comb. Theory, Ser. B* **52**, 153–190 (1991)
44. Robertson, N., Seymour, P.D.: Graph minors. XVI. Excluding a non-planar graph. *J. Comb. Theory, Ser. B* **89**, 43–76 (2003)
45. Robertson, N., Seymour, P.D., Thomas, R.: Quickly excluding a planar graph. *J. Comb. Theory, Ser. B* **62**, 323–348 (1994)
46. Simonson, S.: A variation on the min cut linear arrangement problem. *Math. Syst. Theory* **20**, 235–252 (1987)
47. Thomassen, C.: A simpler proof of the excluded minor theorem for higher surfaces. *J. Comb. Theory, Ser. B* **70**, 306–311 (1997)