Complexity Results for the Spanning Tree Congestion Problem

Yota Otachi[1], Hans L. Bodlaender[2], and Erik Jan van Leeuwen[3]

[1] Graduate School of Information Sciences, Tohoku University, Sendai 980-8579, Japan, JSPS Research Fellow
otachi@dais.is.tohoku.ac.jp
[2] Institute of Information and Computing Sciences, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands
hansb@cs.uu.nl
[3] Department of Informatics, University of Bergen, P.O. Box 7803, 5020 Bergen, Norway
E.J.van.Leeuwen@ii.uib.no

Abstract. We study the problem of determining the *spanning tree congestion* of a graph. We present some sharp contrasts in the complexity of this problem. First, we show that for every fixed k and d the problem to determine whether a given graph has spanning tree congestion at most k can be solved 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 $k \geq 10$. For very small values of k however, the problem becomes polynomially solvable. We also show that it is NP-hard to approximate the spanning tree congestion within a factor better than $11/10$. On planar graphs, we prove the problem is NP-hard in general, but solvable in linear time for fixed k.

1 Introduction

Spanning tree congestion is a relatively new graph parameter, which was formally defined by Ostrovskii [21] in 2004. Prior to Ostrovskii [21], Simonson [25] studied the same parameter under a different name to approximate the cutwidth of outerplanar graphs. Although several graph theoretical results have been presented [7, 16–18, 20, 22] after Ostrovskii [21], 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 $cng_{G,T}(e)$, as the number of detours that contain e. The *congestion of G in T*, denoted by $cng_G(T)$, is the maximum congestion over all edges in T. The *spanning tree congestion* of G, denoted by $stc(G)$, 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 given k. If k is fixed, we denote the problem by k-STC.

The name of the parameter comes from the following analogy [7]: 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

D.M. Thilikos (Ed.): WG 2010, LNCS 6410, pp. 3–14, 2010.

tree spanner problem [6] is a variant of the problem, which minimize the dilation, that is, the length of the longest detours. Several pairs of congestion and dilation problems are known [23]. The most famous pair is the cutwidth problem and the bandwidth problem.

The rest of the paper is organized as follows. Section 2 provides some definitions and basic facts. In Section 3, we study the problem for planar graphs, and show that STC for planar graphs is NP-complete, and k-STC for planar graphs is solvable in linear time. In Section 4, we show that k-STC can be solved in linear time for $1 \leq k \leq 3$. In Section 5, we show that k-STC can be solved in linear time also for graphs of bounded degree. In Section 6, we show that k-STC is NP-complete for edge weighted graphs if $k \geq 10$. Using the result of Section 6, we show in Section 7 that for $k \geq 10$, k-STC is NP-complete for simple unweighted graphs with only one vertex of unbounded degree. In the last section, we conclude the paper and show the approximation hardness of the spanning tree congestion. Due to space limitation, some proofs are omitted.

2 Preliminaries

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 edges whose detours pass through the edge. We denote by $w(F)$ the sum of weights of edges in F for an edge set $F \subseteq E(G)$.

Let G be a connected graph. For $S \subseteq V(G)$, we denote by $G[S]$ the subgraph induced by S. For an edge $e \in E(G)$, we denote by $G - e$ the graph obtained by the deletion of e from G. 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 $cng_{G,T}(e)$ as $cng_{G,T}(e) = |\theta_G(A_e)|$, 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 [12] 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 $T - e$. Clearly, if T is a spanning tree of G and $e \in E(T)$ separates A and B, then $cng_{G,T}(e) \geq |E(A, B)|$ (if G is weighted, $cng_{G,T}(e) \geq w(E(A, B))$). If e separates A and B, we also say that *e divides $A \cup B$ into A and B*.

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

Proposition 2.1. *The spanning tree congestion of G equals the maximum spanning tree congestion of its biconnected components.*

Ostrovskii [21] showed the following lower bound on the spanning tree congestion of graphs.

Lemma 2.2 ([21]). *Let G be a graph, $u, v \in V(G)$. If G has k edge disjoint u–v paths, then $stc(G) \geq k$.*

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 H is a subdivision of a subgraph of G, then H is a *topological minor* of G.

The concept of treewidth was introduced by Robertson and Seymour in their project of Graph Minor Theory (see [24] for example). 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 $tw(G)$, is the minimum width over all tree decompositions of G.

3 Spanning Tree Congestion of Planar Graphs

Ostrovskii [22] has asked whether STC can be solved in polynomial time for planar graphs. By combining a number of known results, we answer this question negatively (assuming $P \neq NP$), and show that k-STC can be solved in linear time for planar graphs. Our results follow easily from some known results for the tree spanner problem. Let G be a graph and T a spanning tree of G. If $dist_T(u, v) \leq k$ for any $\{u, v\} \in E(G)$, then T is a *tree k-spanner* [6]. We denote by $tsp(G)$ the minimum number k such that G has a tree k-spanner. For planar graphs, the following results are known.

Lemma 3.1 ([12]). *It is NP-complete to decide $tsp(G) \leq k$ for planar graphs G and integers k.*

Lemma 3.2 ([11]). *For every fixed k, $tsp(G) \leq k$ can be decided in linear time for planar graphs G.*

A *dual graph* G^* of a planar graph G is a graph that has the vertex set $\mathcal{F}(G)$, the faces of a certain embedding of G, and in which two vertices $f, f' \in \mathcal{F}(G)$ are adjacent in G^* if and only if the two faces f and f' have a common edge in G. It is known that a graph G is planar if and only if G is a dual graph of a planar graph (see e.g. [10]). Since a cut in G corresponds to a cycle in G^*, the following relation holds.

Lemma 3.3 ([12]). *For any planar graph G, $stc(G) = tsp(G^*) + 1$.*

A planar embedding of a planar graph can be constructed in linear time by an algorithm proposed by Hopcroft and Tarjan [15]. From a planar embedding of a planar graph G, we can easily construct geometrically a dual graph G^* (see e.g. [19]). Note that $G = (G^*)^*$. Thus, from Lemma 3.3, we can have the conclusions of this section.

Theorem 3.4. *It is NP-complete to decide $stc(G) \leq k$ for planar graphs G and integers k.*

Theorem 3.5. *For every fixed k, $stc(G) \leq k$ can be decided in linear time for planar graphs G.*

4 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 4.1. *For a connected graph G, $stc(G) = 1$ if and only if G is a tree.*

Proof. If G is a tree, then clearly $stc(G) = 1$. Assume G has a cycle C. Then, for any two vertices in C, G has two edge disjoint paths between them. Thus, by Lemma 2.2, G cannot have any cycle. □

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

Theorem 4.2. *For a connected graph G, $stc(G) = 2$ 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 2.1 and Lemma 2.2, the theorem holds. □

Obviously, the recognition of trees and cactus graphs can be done in linear time, by using standard depth first search techniques (see e.g. [8]). For $k = 3$, we need the following lemma.

Lemma 4.3. *For a graph G, if $stc(G) \leq 3$, then G is planar.*

Proof. Suppose $stc(G) \leq 3$ and G is not planar. From Kuratowski's Theorem (see e.g. [10]), G has either K_5 or $K_{3,3}$ as a topological minor. If G has K_5 as a topological minor, then clearly G contains two vertices such that G has at least four edge disjoint paths between them. From Lemma 2.2, we have $stc(G) \geq 4$, which is a contradiction. Thus, G contains $K_{3,3}$ as a topological minor. Let G' be this topological minor, and $X = \{x_1, x_2, x_3\}, Y = \{y_1, y_2, y_3\} \subset V(G')$ be the two sets corresponding to the two color classes of $K_{3,3}$. By Lemma 7.2 edge subdivisions do not change the spanning tree congestion. Thus, $stc(G') = stc(K_{3,3})$. Moreover, by Hruska's result that shows $stc(K_{m,n}) = m + n - 2$ [16], we can conclude $stc(G') = 4$. Now we need the following two propositions.

Proposition 4.4. *Let H be a connected graph and H' a connected subgraph of H. If a spanning tree S of H has a spanning tree S' of H' as a subgraph, then $cng_H(S) \geq cng_{H'}(S')$.*

Proof. Let $e \in E(S') \subseteq E(S)$. Assume e divides $V(H)$ into A and B, and $V(H')$ into A' and B'. Clearly, $A' \subseteq A$ and $B' \subseteq B$. Thus, $cng_{H,S}(e) = |E(A, B)| \geq |E(A', B')| = cng_{H',S'}(e)$. □

Proposition 4.5. *Let H be a connected graph, S a spanning tree of H, and $A, B \subset V(H)$. If H has p edge disjoint paths $P_1, \ldots, P_p$ between A and B, and $e \in E(S)$ separates A and B, then $cng_{H,S}(e) \geq p$. Moreover, if e does not belong to any P_i, then $cng_{H,S}(e) \geq p + 1$.*

Proof. For each P_i, there exists at least one edge e_i such that the detour of e_i in S passes through the edge e. Since the paths $P_1, \ldots, P_p$ are edge disjoint, $cng_{H,S}(e) \geq p$. Since e itself is the detour for e, $cng_{H,S}(e) \geq p + 1$ if $e \notin \{e_i \mid 1 \leq i \leq p\}$. □

We will show that $cng_{G,T}(e) > 3$ for any spanning tree T of G. If T has a spanning tree T' of G' as a subgraph, then by Proposition 4.4, $cng_G(T) \geq cng_{G'}(T') \geq 4$. Thus T contains no such a subgraph T'. This implies that T contains an edge $e \notin E(G')$ that divides $X \cup Y$ into two nonempty sets, say A and B. Since there are nine edge disjoint paths between X and Y in G', there exist at least three edge disjoint paths between $A \subset X \cup Y$ and $B = (X \cup Y) \setminus A$ in G'. Proposition 4.5 implies $cng_{G,T}(e) \geq 4$ since $e \notin E(G')$. □

From Theorem 3.5 and Lemma 4.3, 3-STC can be solved in linear time, with the linear time algorithm for recognizing planar graphs [15]. This proves the following theorem.

Theorem 4.6. *For* $1 \leq k \leq 3$, *k-STC can be solved in linear time.*

5 Linear Time Solvability of *k*-STC for Graphs of Bounded Degree

In this section, we show that *k*-STC can be solved in linear time for graphs of bounded degree. To this end, we use Courcelle's theorem and a connection between the spanning tree congestion and the treewidth. Courcelle [9] showed that every problem expressible in MS_2 can be solved in linear time for graphs of bounded treewidth, where MS_2 is a graph logic in the monadic second-order logic (see also [14]). In MS_2, we are allowed to use the incident relation inc, the membership relation $\in$, and variables over vertices, edges, vertex sets, and edge sets.

Theorem 5.1. *For graphs of bounded treewidth, k-STC can be solved in linear time.*

Proof. We show that *k*-STC is expressible in MS_2. The proof is omitted. □

We can show that the treewidth of a graph of bounded degree is linear in its spanning tree congestion. (The proof is omitted.)

Lemma 5.2. *For any connected graph G,* $tw(G) \leq \max\{stc(G), \Delta(G)(stc(G) - 1)/2\}$. *Moreover, this bound is tight.*

Lemma 5.2 can be proved similarly to results on the edge remember number, reported in [3]. The upper bound improves on an earlier bound by Kozawa, Otachi, and Yamazaki [17]. A tight example for the bound of Lemma 5.2 is a cycle. By using expanders, we can even show that any upper bound must depend linearly on both the maximum degree and the spanning tree congestion of the graph (we omit the proof). This gives strong evidence that our bound cannot be improved upon. Combining the above facts, we can obtain the main result of this section.

Theorem 5.3. *For graphs of bounded degree, k-STC can be solved in linear time.*

Proof. Let G be a graph of bounded degree and $\Delta(G) = d$. Since k and d are constants, we can check whether $tw(G) \leq \max\{k, d(k-1)/2\}$ in linear time by Bodlaender's algorithm [2]. If the output of the algorithm is "no," then $stc(G) > k$ from Lemma 5.2. Otherwise, G has bounded treewidth. Hence, from Theorem 5.1, we can determine whether $stc(G) \leq k$ in linear time. □

6 Weighted k-STC is NP-Complete for $k \geq 10$

In this section, we prove the following hardness result.

Theorem 6.1. *For any fixed $k \geq 10$, k-STC is NP-complete for edge weighted graphs.*

Clearly, the problem belongs to NP. To show NP-completeness, we present a reduction from (3, B2)-SAT. The problem (3, B2)-SAT is a restricted version of the 3-SAT problem, which is a well-known NP-complete problem [13]. An instance (U, C) of (3, B2)-SAT consists of a set U of n distinct Boolean variables and a collection C of m clauses such that each clause has exactly three literals, and each literal occurs exactly twice. Berman, Karpinski, and Scott [1] showed the NP-completeness of (3, B2)-SAT. In their construction of a hard instance of (3, B2)-SAT, every clause has exactly three variables, that is, there is no clause like $(u, u, *)$, $(\bar{u}, \bar{u}, *)$, or $(u, \bar{u}, *)$. Thus, in what follows, we assume that instances of (3, B2)-SAT satisfy this condition as well.

The constructions in our proof are inspired by the proof of Cai and Corneil [6] for the NP-completeness of the Weighted Tree Spanners problem. Let $k \geq 10$ be a fixed integer. For an arbitrary instance (U, C) of (3, B2)-SAT, we construct an edge weighted graph G_C such that C is satisfiable if and only if $stc(G_C) \leq k$. Let $a = \lceil k/2 \rceil + 1$ and $b = \lfloor k/2 \rfloor - 3$. Each edge in G_C has a weight which will be either a, b, or 1. For example, if $k = 10$, then the weight of an edge is six, two, or one. Clearly, the following proposition holds.

Proposition 6.2. *For $k \geq 10$, $a + b + 2 = k$, $2b + 6 \leq k$, $2a > k$, $6b > k$, and $4b + 4 > k$.*

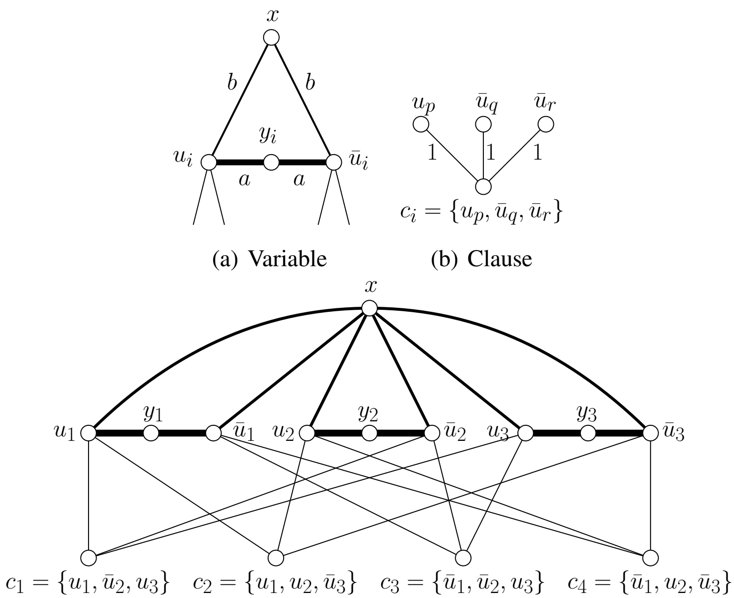
From an instance (U, C) of (3, B2)-SAT, the graph G_C is constructed as follows (see Fig. 1):

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 by *literal edges* of weight b.
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 $c_i = \{l_p, l_q, l_r\} \in C$, connect the clause vertex c_i to the literal vertices l_p, l_q, and l_r by *clause edges* of unit weight.

Clearly, the above construction can be done in polynomial time.

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

Lemma 6.3. *Let T be a spanning tree of G_C. If $cng_{G_C}(T) \leq k$, then*



(a) Variable (b) Clause

(c) G_C with $C = \{\{u_1, \bar{u}_2, u_3\}, \{u_1, u_2, \bar{u}_3\}, \{\bar{u}_1, \bar{u}_2, u_3\}, \{\bar{u}_1, u_2, \bar{u}_3\}\}$.

Fig. 1. Gadgets, and a constructed graph

1. *All bridge edges are contained in T;*
2. *Each clause vertex is a leaf of T;*
3. *For each variable, exactly one of its two literal edges is contained in T.*

Proof (of the first property). 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 $cng_{G_C,T}(\{u_i, y_i\}) = w(\theta(\{y_i\})) = 2a > k$. The other case is almost the same. □

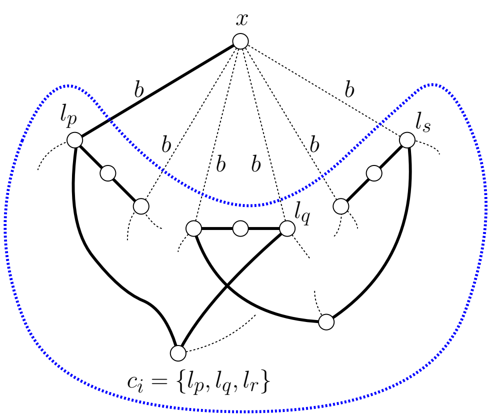
Proof (of the second property). Assume T has the first property. By way of contradiction, suppose some clause vertex $c_i = \{l_p, l_q, l_r\}$ has degree larger than one in T. Let u_p, u_q, u_r be the variables corresponding to the literals l_p, l_q, l_r, respectively. We divide the proof into two cases depending on the degree of c_i in T. Recall that all bridge edges are in T from the first property.

Case 1: $deg_T(c_i) = 3$. The three neighbors of c_i in T are l_p, l_q, and l_r. Let e be the unique literal edge in the unique c_i–x path in T. Then, e separates $\{x\}$ and $\{u_p, \bar{u}_p, u_q, \bar{u}_q, u_r, \bar{u}_r\}$. Thus, $cng_{G_C,T}(e) \geq w(E(\{x\}, \{u_p, \bar{u}_p, u_q, \bar{u}_q, u_r, \bar{u}_r\})) = 6b > k$.

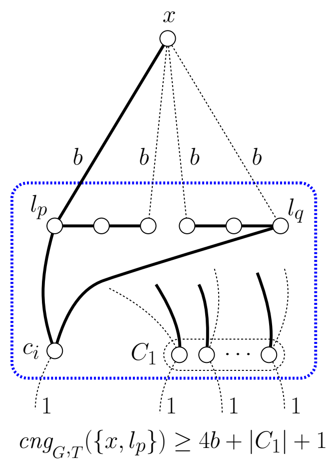
Case 2: $deg_T(c_i) = 2$. Without loss of generality, we assume that the two neighbors of c_i in T are l_p and l_q. Then, at most one of the literal edges of u_p and u_q can be in T. From the above case, we can assume that no clause vertex has degree three in T.

First, assume that none of the literal edges of u_p and u_q are in T. Let $e = \{x, l_s\}$ be the unique literal edge in the unique c_i–x path in T. Then, $l_s \notin \{u_p, \bar{u}_p, u_q, \bar{u}_q\}$, and e separates $\{x\}$ and $\{u_p, \bar{u}_p, u_q, \bar{u}_q, u_s, \bar{u}_s\}$. Thus, $cng_{G_C,T}(e) \geq 6b > k$.

Next, assume that one of the literal edges of u_p and u_q, say e, is in T (see Fig. 2). Let us consider the clause vertices adjacent to at least one of the literal vertices u_p, $\bar{u}_p$, u_q, and $\bar{u}_q$ in G_C. If a clause vertex c_z ($\neq c_i$) is adjacent to two vertices in $\{u_p, \bar{u}_p, u_q, \bar{u}_q\}$ in T, then T has a cycle. Hence, if $c_z \neq c_i$ has degree two in T, and one of the two neighbors of c_z is in $\{u_p, \bar{u}_p, u_q, \bar{u}_q\}$, then another neighbor, say l_s, is not in $\{u_p, \bar{u}_p, u_q, \bar{u}_q\}$. In such a case, e separates $\{x\}$ and $\{u_p, \bar{u}_p, u_q, \bar{u}_q, u_s, \bar{u}_s\}$, and thus, $cng_{G_C,T}(e) \geq 6b > k$ (see Fig. 2(a)). Therefore, every clause vertex (except for c_i) that has at least one of $\{u_p, \bar{u}_p, u_q, \bar{u}_q\}$ as a neighbor in T is a leaf of T. Let C_1 be the set of such leaf clauses. Since every clause has exactly three variables, each $c \in C_1$ has at most two neighbors in $\{u_p, \bar{u}_p, u_q, \bar{u}_q\}$ in G_C. Hence, $cng_{G_C,T}(e) = w(\theta(\{u_p, \bar{u}_p, u_q, \bar{u}_q\} \cup \{c_i\} \cup C_1)) \geq 4b + |C_1| + 1$ (see Fig. 2(b)). Since $cng_{G_C}(T) \leq k < 4b + 4$, we can conclude that $|C_1| \leq 2$. It is easy to see that $cng_{G_C,T}(e) \geq 4b + 5 > k$ if $|C_1| \leq 2$ (see Fig. 3). □



(a) Another clause vertex of degree two.



(b) No other clause vertex of degree two.

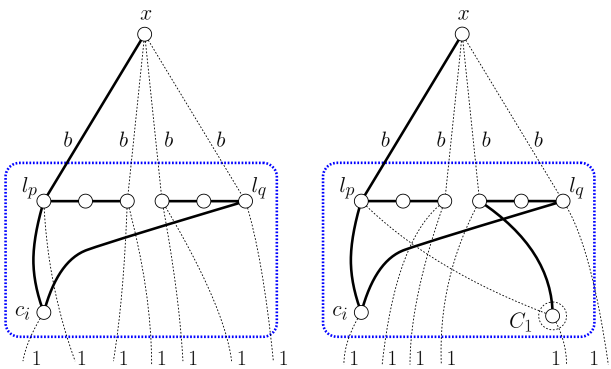
Fig. 2. A clause vertex c_i of degree two

Proof (of the third property). Assume T has the first and the second 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. □

The next two lemmas show that C is satisfiable if and only if $stc(G_C) \leq k$, thus proving Theorem 6.1.

Lemma 6.4. *If $stc(G_C) \leq k$ then C is satisfiable.*

Proof. Let T be a spanning tree of G_C such that $cng_{G_C}(T) \leq k$. From Lemma 6.3, (1) T contains all bridge edges, (2) every clause vertex is a leaf of T, and (3) T contains exactly one literal edge for each variable. From the third property, we can define a truth assignment ξ_T by setting $\xi_T(u_i) = $ **true** if $\{x, u_i\} \in E(T)$ and $\xi_T(u_i) = $ **false** if



(a) $|C_1| = 0$. (b) $|C_1| = 1$. (c) $|C_1| = 2$.

Fig. 3. The cases of $|C_1| \le 2$

$\{x, \bar{u}_i\} \in E(T)$. We show that ξ_T satisfies C. It suffices to show that for every $c_j \in C$, the unique neighbor l_i of c_j is adjacent to x. If l_i is not adjacent to x, then $cng_{G_C,T}(\{l_i, y_i\}) \ge a + b + 3 > k$ (see Fig. 4). This contradicts $cng_{G_C}(T) \le k$. □

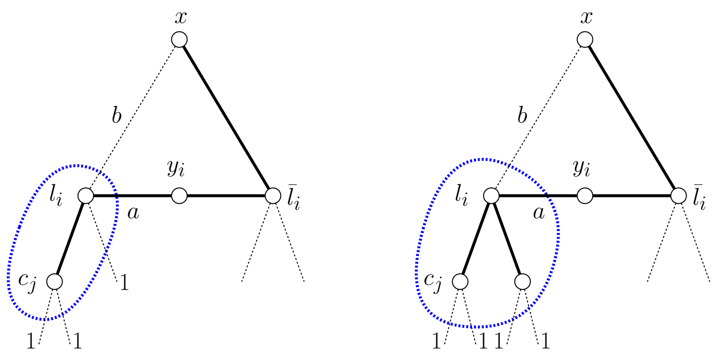


Fig. 4. Unsatisfied clauses

Lemma 6.5. *If C is satisfiable then $stc(G_C) \le k$.*

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

Clearly, T is a spanning tree of G_C. We show that $cng_{G_C}(T) \le k$.

Let $u_i \in U$. Without loss of generality, we assume 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, T may contain any clause edge incident with u_i, but cannot contain any clause edge incident with $\bar{u}_i$. See Fig. 5. Clearly, the edge $\{u_i, y_i\}$ and $\{\bar{u}_i, y_i\}$ have the same congestion, and $cng_{G_C,T}(\{\bar{u}_i, y_i\}) = w(\theta(\{\bar{u}_i\})) = a + b + 2 = k$. If a clause edge incident with u_i is contained in T, then the edge has congestion $3 \leq k$. Obviously, $cng_{G_C,T}(\{x, u_i\}) = w(\theta(\{u_i, \bar{u}_i\} \cup N_T(u_i) \setminus \{x\})) \leq 2b + 6 \leq k$ (see Fig. 5). □

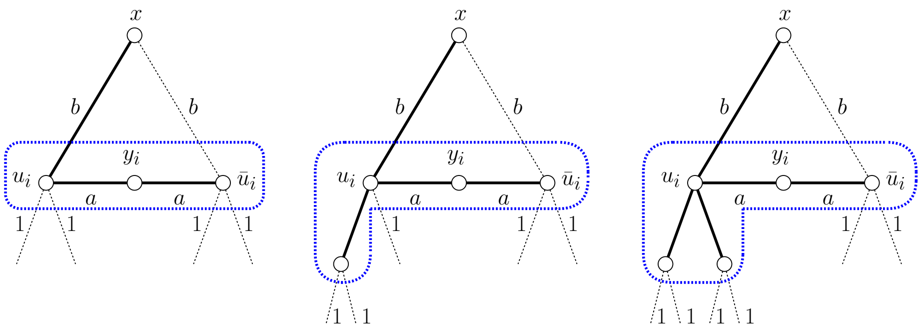


Fig. 5. A spanning tree of congestion at most k

7 Unweighted k-STC is NP-Complete for $k \geq 10$

Extending the result in the previous section, we prove the main theorem of the paper, that is, NP-completeness of k-STC for unweighted graphs. We need the following two lemmas. The proofs are omitted.

Lemma 7.1. *An edge e of weight $w \in \mathbb{Z}^+$ can be replaced by w parallel edges of unit weight without changing the spanning tree congestion.*

Lemma 7.2. *Edge subdivisions do not change the spanning tree congestion of unweighted graphs.*

Combining the above two lemmas, 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 7.3. *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 G can be transformed into unweighted simple graph G' in $O(w \cdot |E(G)|)$ time, such that $stc(G) = stc(G')$.*

Now, we prove the main theorem of the paper.

Theorem 7.4. *For any fixed $k \geq 10$, k-STC is NP-complete for simple unweighted graphs that have only one vertex of unbounded degree.*

Proof. Let (U, C) be an instance of (3, B2)-SAT, and G_C the corresponding graph constructed in the previous section. From Corollary 7.3, we can construct a simple unweighted graph G'_C in polynomial time such that $stc(G'_C) = stc(G_C)$. Clearly, $stc(G'_C) \leq k$ if and only if C is satisfiable.

We show that the vertices other than x have bounded degree. The new vertices added by subdivisions have degree two. Clause vertices have degree three in G_C. Since clause vertices are only incident to unit weight edges, they have degree three in G'_C. Since every y_i is incident to two bridge edges of weight $a = \lceil k/2 \rceil + 1$, y_i has degree $2a \leq k+3$ in G'_C. Literal vertex l_i is incident to two clause edges, one bridge edge, and one literal edge that have weight one, a, and $b = \lfloor k/2 \rfloor - 3$, respectively. Thus, $deg_{G'_C}(l_i) = a + b + 2 = k$. Hence, the maximum degree of G'_C is bounded by $k + 3$, which is a constant. □

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 for planar graphs, graphs of bounded treewidth, and graphs of bounded degree. We also show that the problem can be solved in linear time for any graph if $1 \leq k \leq 3$. On the other hand, we show that if the input graph has one vertex of unbounded degree, then the problem becomes NP-complete for $k \geq 10$. The complexity of k-STC remains open for $4 \leq k \leq 9$.

Since the problem is hard in general, an approximation algorithm with good approximation ratio is required. We say that a polynomial time algorithm for spanning tree congestion is a c_1-approximation algorithm for 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 $k \leq c_1 \cdot stc(G) + c_2$. Using NP-hardness of 10-STC, the following constant lower bound on the approximation ratio can be shown (the proof is omitted).

Theorem 8.1. *There is no polynomial time c_1-approximation algorithm for the spanning tree congestion of simple unweighted graphs such that $c_1 < 11/10$, unless P = NP.*

We also considered the complexity of STC or k-STC on some restricted graph classes. It is known that the tree spanner problem is NP-hard for chordal graphs [4] and chordal bipartite graphs [5]. It would be interesting to determine the complexity of STC or k-STC for these graph classes.

References

1. Berman, P., Karpinski, M., Scott, A.D.: Approximation hardness of short symmetric instances of MAX-3SAT, ECCC TR03-049 (2003)
2. Bodlaender, H.L.: A linear-time algorithm for finding tree-decompositions of small treewidth. SIAM J. Comput. 25, 1305–1317 (1996)
3. Bodlaender, H.L.: A partial k-arboretum of graphs with bounded treewidth. Theoret. Comput. Sci. 209, 1–45 (1998)
4. Brandstädt, A., Dragan, F.F., Le, H.-O., Le, V.B.: Tree spanners on chordal graphs: complexity and algorithms. Theoret. Comput. Sci. 310, 329–354 (2004)
5. 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)

6. Cai, L., Corneil, D.G.: Tree spanners. SIAM J. Discrete Math. 8, 359–387 (1995)
7. Castejón, A., Ostrovskii, M.I.: Minimum congestion spanning trees of grids and discrete toruses. Discuss. Math. Graph Theory 29, 511–519 (2009)
8. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 3rd edn. MIT Press, Cambridge (2009)
9. Courcelle, B.: The monadic second-order logic of graphs III: Tree-decompositions, minor and complexity issues. Theor. Inform. Appl. 26, 257–286 (1992)
10. Diestel, R.: Graph Theory, 3rd edn. Springer, Heidelberg (2005)
11. Dragan, F.F., Fomin, F.V., Golovach, P.A.: Spanners in sparse graphs. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part I. LNCS, vol. 5125, pp. 597–608. Springer, Heidelberg (2008)
12. Fekete, S.P., Kremer, J.: Tree spanners in planar graphs. Discrete Appl. Math. 108, 85–103 (2001)
13. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, New York (1979)
14. Hliněný, P., Oum, S., Seese, D., Gottlob, G.: Width parameters beyond tree-width and their applications. J. Comput. 51, 326–362 (2008)
15. Hopcroft, J., Tarjan, R.: Efficient planarity testing. J. ACM 21, 549–568 (1974)
16. Hruska, S.W.: On tree congestion of graphs. Discrete Math. 308, 1801–1809 (2008)
17. Kozawa, K., Otachi, Y., Yamazaki, K.: On spanning tree congestion of graphs. Discrete Math. 309, 4215–4224 (2009)
18. Law, H.-F.: Spanning tree congestion of the hypercube. Discrete Math. 309, 6644–6648 (2009)
19. Lawler, E.: Combinatorial Optimization: Networks and Matroids. Holt, Rinehart and Winston (1976)
20. Löwenstein, C., Rautenbach, D., Regen, F.: On spanning tree congestion. Discrete Math. 309, 4653–4655 (2009)
21. Ostrovskii, M.I.: Minimal congestion trees. Discrete Math. 285, 219–226 (2004)
22. Ostrovskii, M.I.: Minimum congestion spanning trees in planar graphs. Discrete Math. 310, 1204–1209 (2010)
23. Raspaud, A., Sýkora, O., Vrťo, I.: Congestion and dilation, similarities and differences: A survey. In: SIROCCO 2000, pp. 269–280. Carleton Scientific (2000)
24. Robertson, N., Seymour, P.D.: Graph minors. X. Obstructions to tree-decomposition. J. Combin. Theory Ser. B 52, 153–190 (1991)
25. Simonson, S.: A variation on the min cut linear arrangement problem. Math. Syst. Theory 20, 235–252 (1987)