Journal of Computer and System Sciences 111 (2020) 22–41

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

A multivariate analysis of the strict terminal connection problem ☆

Alexander A. Melo [a,*], Celina M.H. Figueiredo [a], Uéverton S. Souza [b]

[a] Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil
[b] Universidade Federal Fluminense, Niterói, Brazil

ARTICLE INFO

Article history:
Received 30 August 2018
Received in revised form 29 January 2020
Accepted 4 February 2020
Available online 10 February 2020

Keywords:
Terminal vertices
Connection tree
Steiner tree
Bounded degree
Parameterized complexity

ABSTRACT

A strict connection tree of a graph G for a set W is a tree subgraph of G whose leaf set equals W. The Strict terminal connection problem (S-TCP) is a network design problem whose goal is to decide whether G admits a strict connection tree T for W with at most ℓ vertices of degree 2 and r vertices of degree at least 3. We establish a Poly vs. NP-c dichotomy for S-TCP with respect to ℓ and $\Delta(G)$. We prove that S-TCP parameterized by r is W[2]-hard even if ℓ is bounded by a constant; we provide a kernelization for S-TCP parameterized by ℓ, r and $\Delta(G)$, and we prove that such a version of the problem does not admit a polynomial kernel, unless NP $\subseteq$ coNP/poly. Finally, we analyze S-TCP on split graphs and cographs.

1. Introduction

Network design problems are combinatorial questions of great practical and theoretical interest. Indeed, such problems are challenging tasks closely related to real-world applications. In this paper, we investigate the computational complexity of a network design problem called Strict terminal connection.

A *connection tree* T of a graph $G = (V, E)$ for a *terminal set* $W \subseteq V$ is a tree subgraph of G such that $W \subseteq V(T)$ and every leaf of T belongs to W *cf.* [1,2]. In a connection tree T for W, the vertices belonging to $V(T) \setminus W$ are called *non-terminal* and are classified into two types according to their respective degrees in T, namely: the non-terminal vertices with degree exactly equal to 2 in T are called *linkers* and the non-terminal vertices with degree at least 3 in T are called *routers cf.* [1,2]. Thus, there exists a partition $\mathcal{V}_T = \{W, \mathsf{L}(T), \mathsf{R}(T)\}$ of the vertex set of a connection tree T into terminal vertices, linkers and routers, where $\mathsf{L}(T)$ and $\mathsf{R}(T)$ denote the linker and router sets of T, respectively.

In some applications, the terminal vertices must be leaves. For example, in telecommunications, the message senders and receivers, which correspond to the terminal vertices, are not allowed to behave as transmitters [3], which correspond to the vertices with degree greater than 1. A connection tree T for W is said *strict* if all vertices belonging to W are leaves of T, i.e. the leaf set of T coincides with the terminal set W. Based on that and also motivated by applications in information security and network routing, Dourado et al. [2] introduced the Strict terminal connection problem (S-TCP), which has as

☆ This work was partially supported by the Brazilian agencies CNPq (Grant numbers: 132890/2015-1, 140399/2017-8, 407635/2018-1, 303726/2017-2), CAPES (Finance Code 001) and FAPERJ (Grant numbers: CNE E-26/202.793/2017, JCNE E-26/203.272/2017).

* Corresponding author.
E-mail addresses: aamelo@cos.ufrj.br (A.A. Melo), celina@cos.ufrj.br (C.M.H. Figueiredo), ueverton@ic.uff.br (U.S. Souza).

https://doi.org/10.1016/j.jcss.2020.02.001

Table 1
Contributions of this work (in bold) and known results for S-TCP.

Graph class	Parameters				
	–	ℓ	r	ℓ, r	ℓ, r, Δ
General	NP-c [2]	NP-c [2]	Poly for $r \in \{0, 1\}$ [29] but Open for $r \geq 2$, and **W[2]-h** Theorem 4	XP [2] but **W[2]-h** Theorem 4	FPT [2] (and Theorem 5) but **No-poly kernel** Theorem 6
$\Delta = 4$	**NP-c** Theorem 1	**NP-c** Theorem 1	Poly for $r \in \{0, 1\}$ [29] but Open for $r \geq 2$	FPT [2] (and Theorem 5)	FPT [2] (and Theorem 5)
$\Delta = 3$	**NP-c** Theorem 2	**XP** Theorem 3	Poly for $r \in \{0, 1\}$ [29] but Open for $r \geq 2$	FPT [2] (and Theorem 5)	FPT [2] (and Theorem 5)
Split	**NP-c** Theorem 7	**NP-c** Theorem 7	**XP** Theorem 7 but **W[2]-h** Theorem 7	XP [2] (and Theorem 7) but **W[2]-h** Theorem 7	FPT [2] (and Theorem 5)
Cographs	**Poly** Theorem 8	**Poly** Theorem 8	**Poly** Theorem 8	**Poly** Theorem 8	**Poly** Theorem 8

input a graph $G = (V, E)$, a non-empty subset $W \subseteq V$ and two non-negative integers ℓ and r, and asks for the existence of a strict connection tree T of G for W such that $|\mathsf{L}(T)| \leq \ell$ and $|\mathsf{R}(T)| \leq r$.

Besides the practical point of view, S-TCP is strongly related to classical network design problems, such as vertex-disjoint path problems and integral network flow problems. Furthermore, S-TCP can be viewed as a close variant of the unweighted version of the Steiner tree problem in graphs, in which we are given a graph $G = (V, E)$, a terminal set $W \subseteq V$ and a positive integer k, and we aim to decide whether G contains a connected subgraph T such that $W \subseteq V(T)$ and $|E(T)| \leq k$. Since every minimal solution T for a given instance of Steiner tree is necessarily a connection tree for W, the constraints on T being a tree and its leaf set being a subset of W can be omitted without loss of generality from the definition of Steiner tree. However, for our target problem, S-TCP, neither constraint can be ignored. Indeed, as a result of the number of non-terminal vertices with degree 2 being bounded, there exist instances $I = (G, W, \ell, r)$ that would be considered Yes instances of S-TCP although all connected subgraphs of G containing the vertices in W, and with at most ℓ non-terminal vertices with degree 2 and at most r non-terminal vertices with degree at least 3, have cycles or non-terminal vertices that are leaves.

Steiner tree is a classical NP-complete problem [4], and it has been extensively studied from distinct classes of algorithmic paradigms, such as structured graph classes [5–9] and parameterized complexity [10–14].

Additionally, several variants of Steiner tree have been investigated over the years. One of the most well-known variants is the so-called Full Steiner tree (or Terminal Steiner tree), in which the terminal vertices are further constrained to be leaves of the sought connection tree T, i.e. T must be strict [15]. The original motivation to study the Full Steiner tree problem was to use it as a building block to solve the Steiner tree problem itself, provided the fact that any connection tree can be decomposed into strict connection trees [16,17]. Full Steiner tree was proved to be NP-complete [3,17,18]. On the other hand, Fernau et al. [19] proved that the problem is in FPT when parameterized by k, the maximum size of the sought strict connection tree T, but that it does not admit a strict polynomial kernel unless $\mathrm{P} = \mathrm{NP}$. It is also known that, unless $\mathrm{NP} \subseteq \mathrm{coNP/poly}$, Full Steiner tree parameterized by k does not admit a polynomial kernel *cf.* [14,12]. In addition, many approximation algorithms and approximation lower bounds for the problem have been proposed [3,17,20–25] in the last years.

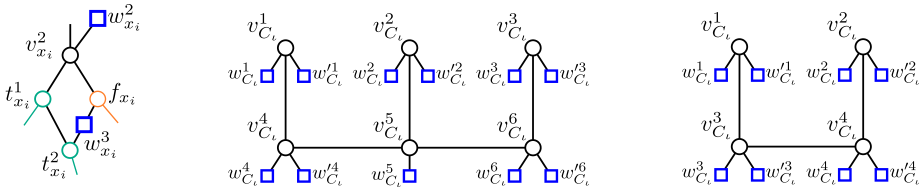
Motivated by applications in optical networks and bandwidth consumption minimization, another variant of Steiner tree that has been investigated is the one in which the number of *branching nodes*, i.e. vertices with degree at least 3 in T (not necessarily non-terminal), is bounded. In [26–28], the authors address the undirected and directed cases of this variant, for which they devise approximation and parameterized polynomial-time algorithms, apart from obtaining some intractability results.

Nevertheless, there is no variant of Steiner tree requiring simultaneously *full Steiner trees* and bounded number of *branching nodes* that has been investigated. Therefore, we emphasize that S-TCP certainly has its own merit to be studied. Thus, this paper aims to provide a multivariate analysis of S-TCP with respect to the input aspects: ℓ, number of linkers; r, number of routers; and Δ, the maximum degree of the input graph.

S-TCP was proved to be polynomial-time solvable if ℓ and r are bounded by constants [2], or if ℓ is unbounded but $r \in \{0, 1\}$ [29]; and it was proved to be in FPT when ℓ, r and Δ are parameters [2]. On the other hand, for every $\ell \geq 0$, S-TCP was proved to be NP-complete when r is unbounded, even if Δ is bounded by a constant [2].

In this paper, we extend the results described above by presenting several contributions to the complexity of S-TCP. More specifically, in Section 2, we establish a *Poly vs. NP-c* dichotomy for S-TCP with respect to ℓ and Δ; and in Section 3, we provide further complexity results for S-TCP parameterized by ℓ, r or Δ. Additionally, in Sections 4 and 5, we investigate the problem on split graphs and cographs, respectively. Finally, in Section 6, we present some directions for future work. Table 1 summarizes the contributions of this work.

Throughout this work we denote by n, m, and Δ the number of vertices, the number of edges, and the maximum degree of the input graph G, respectively.



(a) Gadget G_{x_i}. (b) Gadget G_{C_ι}, for $|C_\iota| = 3$. (c) Gadget G_{C_ι}, for $|C_\iota| = 2$.

Fig. 1. Gadgets G_{x_i} and G_{C_ι}.

2. Bounded maximum degree dichotomy

In this section, we address the analysis of the computational complexity of S-TCP when it is restricted to graphs with bounded maximum degree. More specifically, we prove that S-TCP is NP-complete even if ℓ is bounded by a constant and $\Delta = 4$, or ℓ is unbounded and $\Delta = 3$. On the other hand, we give a polynomial-time algorithm for the problem when ℓ is bounded by a constant and $\Delta = 3$. Observe that S-TCP is easily solvable if $\Delta \leq 2$. Thus, our results establish a *Poly vs. NP-c* dichotomy for S-TCP with respect to ℓ and Δ.

2.1. Hardness results

We first prove the NP-completeness of S-TCP with ℓ bounded by a constant and $\Delta = 4$. Our proof consists in a polynomial-time reduction from the NP-complete (*cf.* [30]) variant of 3-SAT, called 3-SAT(3), which has as input a set X of boolean variables and a set $\mathcal{C}$ of clauses over X such that: (1) each clause in $\mathcal{C}$ has two or three distinct literals; and (2) each variable in X appears exactly twice positive and once negative in the clauses belonging to $\mathcal{C}$; and asks for the existence of a truth assignment for the variables in X such that every clause in $\mathcal{C}$ has at least one true literal.

Theorem 1. *For every $\ell \geq 0$, S-TCP remains NP-complete even if $\Delta = 4$.*

Proof. Let $I = (X, \mathcal{C})$ be an instance of 3-SAT(3), where $X = \{x_1, x_2, \ldots, x_p\}$ and $\mathcal{C} = \{C_1, C_2, \ldots, C_q\}$. We construct from I an instance $f(I) = (G, W, r)$ of S-TCP with ℓ bounded by a constant, such that $\Delta = 4$, as follows (see Fig. 2):

- first, we create ℓ vertices $u_1, u_2, \ldots, u_\ell$ and, for each $i \in \{1, 2, \ldots, \ell - 1\}$, add the edges $u_i u_{i+1}$; moreover, we create the vertices w_I and v_I and add the edges $w_I u_1$ and $u_\ell v_I$, originating the path $P_I = \langle w_I, u_1, \ldots, u_\ell, v_I \rangle$;
- for each variable $x_i \in X$, we create the gadget G_{x_i} (see Fig. 1a) such that
  - $V(G_{x_i}) = \{v^2_{x_i}, w^2_{x_i}, w^3_{x_i}, t^1_{x_i}, t^2_{x_i}, f_{x_i}\}$ and
  - $E(G_{x_i}) = \{w^2_{x_i} v^2_{x_i}, v^2_{x_i} t^1_{x_i}, t^1_{x_i} t^2_{x_i}, t^2_{x_i} w^3_{x_i}, w^3_{x_i} f_{x_i}, f_{x_i} v^2_{x_i}\}$;

  moreover, we create the vertices $w^1_{x_i}$ and $v^1_{x_i}$, and we add the edges $w^1_{x_i} v^1_{x_i}$ and $v^1_{x_i} v^2_{x_i}$;
- we create a complete strict binary tree T_I, rooted at v_I, whose leaves are the vertices $v^1_{x_1}, v^1_{x_2}, \ldots, v^1_{x_p}$;
- for each clause $C_\iota \in \mathcal{C}$, we create the gadget G_{C_ι} such that, if $|C_\iota| = 3$, then (see Fig. 1b)
  - $V(G_{C_\iota}) = \{v^\kappa_{C_\iota}, w^\kappa_{C_\iota}, w'^\kappa_{C_\iota} \mid \kappa \in \{1, 2, 3, 4, 6\}\} \cup \{v^5_{C_\iota}, w^5_{C_\iota}\}$ and
  - $E(G_{C_\iota}) = \{v^\kappa_{C_\iota} w^\kappa_{C_\iota}, v^\kappa_{C_\iota} w'^\kappa_{C_\iota} \mid \kappa \in \{1, 2, 3, 4, 6\}\} \cup \{v^5_{C_\iota} w^5_{C_\iota}\} \cup \{v^1_{C_\iota} v^4_{C_\iota}, v^4_{C_\iota} v^5_{C_\iota}, v^5_{C_\iota} v^2_{C_\iota}, v^5_{C_\iota} v^6_{C_\iota}, v^6_{C_\iota} v^3_{C_\iota}\}$,

  and if $|C_\iota| = 2$, then (see Fig. 1c)
  - $V(G_{C_\iota}) = \{v^\kappa_{C_\iota}, w^\kappa_{C_\iota}, w'^\kappa_{C_\iota} \mid \kappa \in \{1, \ldots, 4\}\}$ and
  - $E(G_{C_\iota}) = \{v^\kappa_{C_\iota} w^\kappa_{C_\iota}, v^\kappa_{C_\iota} w'^\kappa_{C_\iota} \mid \kappa \in \{1, \ldots, 4\}\} \cup \{v^1_{C_\iota} v^3_{C_\iota}, v^3_{C_\iota} v^4_{C_\iota}, v^4_{C_\iota} v^2_{C_\iota}\}$;
- for each clause $C_\iota \in \mathcal{C}$, we add the edge $t^j_{x_i} v^\kappa_{C_\iota}$ if the κ-th literal belonging to C_ι corresponds to the j-th occurrence in I of the positive literal x_i, for $x_i \in X$, $j \in \{1, 2\}$ and $\kappa \in \{1, \ldots, |C_\iota|\}$; on the other hand, we add the edge $f_{x_i} v^\kappa_{C_\iota}$ if the κ-th literal belonging to C_ι corresponds to the (only) occurrence in I of the negative literal $\overline{x}_i$, for $x_i \in X$ and $\kappa \in \{1, \ldots, |C_\iota|\}$;
- we define $W = W_{\mathcal{C}} \cup \{w_I\} \cup \{w^1_{x_i}, w^2_{x_i}, w^3_{x_i} \mid x_i \in X\}$, where $W_{\mathcal{C}} = \bigcup_{C_\iota \in \mathcal{C}} W_{C_\iota}$ and
  - $W_{C_\iota} = \begin{cases} \{w^\kappa_{C_\iota}, w'^\kappa_{C_\iota} \mid \kappa \in \{1, 2, 3, 4, 6\}\} \cup \{w^5_{C_\iota}\} \text{ if } |C_\iota| = 3 \\ \{w^\kappa_{C_\iota}, w'^\kappa_{C_\iota} \mid \kappa \in \{1, \ldots, 4\}\} \text{ if } |C_\iota| = 2; \end{cases}$
- finally, we define $r = |V \setminus W| - \ell$.

One may verify that the maximum degree of G is 4.

Fig. 2 exemplifies the graph G and the terminal set W of $f(I)$.

Now, we prove that I is a Yes instance of 3-SAT(3) if and only if $f(I)$ is a Yes instance of S-TCP with ℓ bounded by a constant.

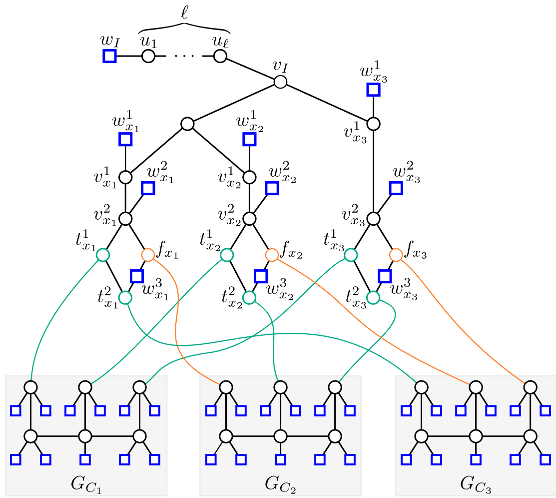


Fig. 2. Graph G and terminal set W (blue square vertices) of $f(I)$ obtained from the instance $I = (X, \mathcal{C})$ of 3-SAT(3), where $X = \{x_1, x_2, x_3\}$ and $\mathcal{C} = \{C_1 = \{x_1, x_2, x_3\}, C_2 = \{\overline{x}_1, x_2, x_3\}, C_3 = \{x_1, \overline{x}_2, \overline{x}_3\}\}$. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

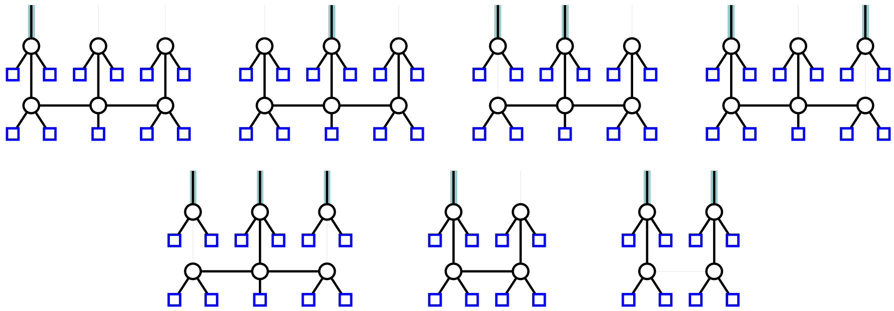


Fig. 3. Patterns used to connect the vertices of G_{C_ι} to T.

First, suppose that I is a Yes instance of 3-SAT(3). Hence, there exists a truth assignment $\alpha \colon X \to \{true, false\}$ that satisfies all clauses in $\mathcal{C}$. Based on α, we construct a strict connection tree T of G for W as follows:

- we add the path P_I to T;
- we add the complete strict binary tree T_I to T along with the vertices $w^1_{x_i}$ and the edges $w^1_{x_i} v_{x_i}$ for every $x_i \in X$;
- for each variable $x_i \in X$, we add the vertices $v^2_{x_i}$, $w^2_{x_i}$ and $w^3_{x_i}$ and the edges $v^1_{x_i} v^2_{x_i}$ and $v^2_{x_i} w^2_{x_i}$ to T; furthermore, if $\alpha(x_i) = true$, then we add the vertices $t^1_{x_i}$ and $t^2_{x_i}$ to T along with all of their neighbors and incident edges in G; on the other hand, if $\alpha(x_i) = false$, then we add the vertex f_{x_i} to T along with all of its neighbors and incident edges in G.

Since by hypothesis α satisfies all clauses in $\mathcal{C}$, for each $C_\iota \in \mathcal{C}$, there exists at least one vertex either $t^j_{x_i}$ (where $j \in \{1, 2\}$) or f_{x_i}, for some $x_i \in X$, which is adjacent to one of the vertices $v^1_{C_\iota}$, $v^2_{C_\iota}$ (and $v^3_{C_\iota}$ if $|C_\iota| = 3$) in T. Thus, we can connect all the other vertices of the gadget G_{C_ι} to T by following one of the patterns (or their symmetrical cases) depicted in Fig. 3, concluding the construction of T.

Fig. 4 exemplifies the strict connection tree T of G for W, referring to the instance $f(I)$ described in Fig. 2, obtained from a truth assignment α.

Observe that, T is indeed a strict connection tree of G for W and, besides that, $\mathsf{L}(T) = \{u_1, u_2, \ldots, u_\ell\}$ and $\mathsf{R}(T) = V(T_I) \cup \left(\bigcup_{C_\iota \in \mathcal{C}} V(G_{C_\iota}) \setminus W_C\right) \cup \{v^2_{x_i} \mid x_i \in X\} \cup \{t^1_{x_i}, t^2_{x_i} \mid \alpha(x_i) = true, x_i \in X\} \cup \{f_{x_i} \mid \alpha(x_i) = false, x_i \in X\}$. Hence, $|\mathsf{L}(T)| \le \ell$ and, obviously, $|\mathsf{R}(T)| \le |V \setminus W| - \ell$. Therefore, $f(I)$ is a Yes instance of S-TCP with ℓ bounded by a constant.

Conversely, suppose that $f(I)$ is a Yes instance of S-TCP with ℓ bounded by a constant. Hence, there exists a strict connection tree T of G for W such that $|\mathsf{L}(T)| \le \ell$ and $|\mathsf{R}(T)| \le |V \setminus W| - \ell$. Since the path P_I is necessarily contained in T and there are precisely ℓ vertices in P_I with degree 2, the vertices belonging to $V(T) \setminus (W \cup V(P_I))$ are routers of T. However, we have $d_G(t^1_{x_i}) = d_G(t^2_{x_i}) = d_G(f_{x_i}) = 3$. Thus, if $t^1_{x_i} \in V(T)$, then $N_T(t^1_{x_i}) = N_G(t^1_{x_i})$; if $t^2_{x_i} \in V(T)$, then $N_T(t^2_{x_i}) = N_G(t^2_{x_i})$; and, if $f_{x_i} \in V(T)$, then $N_T(f_{x_i}) = N_G(f_{x_i})$. Hence, if $t^1_{x_i} \in V(T)$ or $t^2_{x_i} \in V(T)$, then $f_{x_i} \notin V(T)$, otherwise T would have a cycle (and the degree of the terminal $w^3_{x_i}$ would be greater than 1 in T). Analogously, if $f_{x_i} \in V(T)$, then $t^1_{x_i}, t^2_{x_i} \notin V(T)$. Thus, we can define a truth assignment $\alpha \colon X \to \{true, false\}$ in the following way: $\alpha(x_i) = true$ if and only if $f_{x_i} \notin$

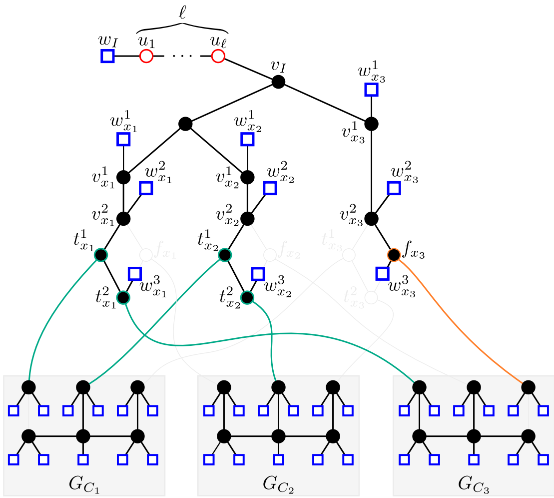


Fig. 4. Strict connection tree of G for W obtained from the truth assignment $\alpha(x_1) = true$, $\alpha(x_2) = true$ and $\alpha(x_3) = false$.

$V(T)$. Since by hypothesis $W_{\mathcal{C}} \subset W \subseteq V(T)$ and, for every clause $C_\iota \in \mathcal{C}$, the path in T between any terminal $w_{C_\iota} \in W_{C_\iota}$ and any other terminal $w \in W \setminus W_{C_\iota}$ must contain either $t^j_{x_i}$ (where $j \in \{1, 2\}$) or f_{x_i}, for some $x_i \in X$, we have that all clauses in $\mathcal{C}$ are satisfied by α. Therefore, I is a Yes instance of 3-SAT(3). $\square$

Now, we analyze S-TCP when $\Delta = 3$. First, by a polynomial-time reduction from the 1-in-3-SAT(3) problem — which has the same input of 3-SAT(3) but asks whether there exists a truth assignment such that every clause has exactly (instead of at least) one true literal — we prove that, if ℓ is unbounded, then S-TCP remains NP-complete even if $\Delta = 3$.

The next proposition shows that 1-in-3-SAT(3) is an NP-complete problem and that, unless ETH fails, it does not admit a $2^{o(|X|+|\mathcal{C}|)}$-time algorithm. To prove such a result, we first present a polynomial-time reduction from 3-SAT (where each clause has exactly three distinct literals) to 1-in-3-SAT, which has the same input of 3-SAT and the same question of 1-in-3-SAT(3); then, we present a polynomial-time reduction from 1-in-3-SAT to 1-in-3-SAT(3).

For simplicity, in the context of 1-in-3-SAT and 1-in-3-SAT(3), we say that a truth assignment α *satisfies* a given clause C if there is exactly (instead of at least) one true literal in C under α.

Proposition 1. *1-in-3-SAT(3) is NP-complete and cannot be solved in time $2^{o(|X|+|\mathcal{C}|)}$, unless ETH fails.*

Proof. Polynomial-time reduction from 3-SAT to 1-in-3-SAT.

Let $I = (X', \mathcal{C}')$ be an instance of 3-SAT. We construct from I an instance $f(I) = (X'', \mathcal{C}'')$ of 1-in-3-SAT, as follows: $X'' = X' \cup \{y^1_j, y^2_j, y^3_j, y^4_j \mid C_j \in \mathcal{C}'\}$ and $\mathcal{C}'' = \{C^1_j, C^2_j, C^3_j \mid C_j \in \mathcal{C}'\}$, where, for $C_j = \{z^1_j, z^2_j, z^3_j\}$, with $z^i_j \in \{x^i_j, \overline{x}^i_j\}$ for some variable $x^i_j \in X'$, we have $C^1_j = \{\overline{z}^1_j, y^1_j, y^2_j\}$, $C^2_j = \{z^2_j, y^2_j, y^3_j\}$ and $C^3_j = \{\overline{z}^3_j, y^3_j, y^4_j\}$. Note that, $|X''| = |X'| + 4|\mathcal{C}'|$ and $|\mathcal{C}''| = 3|\mathcal{C}'|$.

Suppose that I is a Yes instance of 3-SAT, and let $\alpha\colon X' \to \{true, false\}$ be a truth assignment that satisfies all clauses in $\mathcal{C}'$. Then, from α, we defined a truth assignment $\beta\colon X'' \to \{true, false\}$, as follows: $\beta(x_i) = \alpha(x_i)$ for all $x_i \in X'$, and

- $\beta(y^1_j) = false$, $\beta(y^2_j) = true$, $\beta(y^3_j) = false$ and $\beta(y^4_j) = false$ if $\alpha(z^1_j) = true$, $\alpha(z^2_j) = false$ and $\alpha(z^3_j) = false$;
- $\beta(y^1_j) = false$, $\beta(y^2_j) = false$, $\beta(y^3_j) = false$ and $\beta(y^4_j) = false$ if $\alpha(z^1_j) = false$, $\alpha(z^2_j) = true$ and $\alpha(z^3_j) = false$;
- $\beta(y^1_j) = false$, $\beta(y^2_j) = false$, $\beta(y^3_j) = true$ and $\beta(y^4_j) = false$ if $\alpha(z^1_j) = false$, $\alpha(z^2_j) = false$ and $\alpha(z^3_j) = true$;
- $\beta(y^1_j) = true$, $\beta(y^2_j) = false$, $\beta(y^3_j) = false$ and $\beta(y^4_j) = false$ if $\alpha(z^1_j) = true$, $\alpha(z^2_j) = true$ and $\alpha(z^3_j) = false$;
- $\beta(y^1_j) = false$, $\beta(y^2_j) = true$, $\beta(y^3_j) = false$ and $\beta(y^4_j) = true$ if $\alpha(z^1_j) = true$, $\alpha(z^2_j) = false$ and $\alpha(z^3_j) = true$;
- $\beta(y^1_j) = false$, $\beta(y^2_j) = false$, $\beta(y^3_j) = false$ and $\beta(y^4_j) = true$ if $\alpha(z^1_j) = false$, $\alpha(z^2_j) = true$ and $\alpha(z^3_j) = true$;
- $\beta(y^1_j) = true$, $\beta(y^2_j) = false$, $\beta(y^3_j) = false$ and $\beta(y^4_j) = true$ if $\alpha(z^1_j) = true$, $\alpha(z^2_j) = true$ and $\alpha(z^3_j) = true$;

for all $C_j \in \mathcal{C}'$, where $\alpha(z^i_j) = \alpha(x^i_j)$ if $z^i_j = x^i_j$, and $\alpha(z^i_j) = \overline{\alpha(x^i_j)}$ otherwise. It is easy to see that β satisfies all clauses in $\mathcal{C}''$. Therefore, $f(I)$ is a Yes instance of 1-in-3-SAT.

Conversely, suppose that $f(I)$ is a Yes instance of 1-in-3-SAT, and let $\beta\colon X'' \to \{true, false\}$ be a truth assignment that satisfies all clauses in $\mathcal{C}''$. We let $\alpha\colon X' \to \{true, false\}$ be the truth assignment such that, for each $x_i \in X'$, $\alpha(x_i) = \beta(x_i)$. For the sake of contradiction, suppose that there exists a clause $C_j = \{z^1_j, z^2_j, z^3_j\} \in \mathcal{C}'$ that is false under α, i.e. $\alpha(z^1_j) = false$, $\alpha(z^2_j) = false$ and $\alpha(z^3_j) = false$, where $z^i_j \in \{x^i_j, \overline{x}^i_j\}$ for some variable $x^i_j \in X'$. Then, $\beta(y^1_j) = false$ and $\beta(y^2_j) = false$, otherwise

C_j^1 would have more than one true literal under β. Consequently, $\beta(y_j^3) = true$, otherwise C_j^2 would have no true literal under β. However, this contradicts the hypothesis that C_j^3 has exactly one true literal under β, since in this case $\overline{z}_j^3$ and y_j^3 are both true literals under β. As a result, we obtain that α is indeed a truth assignment that satisfies all clauses in $\mathcal{C}'$. Therefore I is YES instance of 3-SAT.

Polynomial-time reduction from 1-IN-3-SAT to 1-IN-3-SAT(3).

Now, let $I = (X'', \mathcal{C}'')$ be an instance of 1-IN-3-SAT. For a given variable $x_i \in X''$, let γ_i denote the number of positive occurrences of x_i in I, and let $\overline{\gamma}_i$ denote the number of negative occurrences of x_i in I. We construct from I an instance $f(I) = (X, \mathcal{C})$ of 1-IN-3-SAT(3), as follows:

- for each $x_i \in X''$ such that $\gamma_i \geq 2$ and each $j \in \{2, \ldots, \gamma_i\}$, we create the variable μ_i^j and replace the j-th positive occurrence of x_i with the positive literal μ_i^j; moreover, if $\overline{\gamma}_i \geq 1$, then we create the variable $\mu_i^{\gamma_i+1}$;
- for each $x_i \in X''$ such that $\overline{\gamma}_i \geq 1$ and each $j \in \{1, \ldots, \overline{\gamma}_i\}$, we create the variable ν_i^j and replace the j-th negative occurrence of x_i with the positive literal ν_i^j; moreover, we create the variable $\nu_i^{\overline{\gamma}_i+1}$;
- for each $x_i \in X''$ such that $\gamma_i = 0$ and $\overline{\gamma}_i \geq 1$, we create the clauses $\{x_i, \overline{x}_i\}$, $\{x_i, \nu_i^1\}$, $\{\overline{\nu}_i^1, \nu_i^2\}$, ..., $\{\overline{\nu}_i^{\overline{\gamma}_i-1}, \nu_i^{\overline{\gamma}_i}\}$, $\{\overline{\nu}_i^{\overline{\gamma}_i}, \nu_i^{\overline{\gamma}_i+1}\}$, $\{\nu_i^{\overline{\gamma}_i+1}, \overline{\nu}_i^{\overline{\gamma}_i+1}\}$;
- for each $x_i \in X''$ such that $\gamma_i = 1$ and $\overline{\gamma}_i \geq 1$, we create the clauses $\{x_i, \nu_i^1\}$, $\{\overline{\nu}_i^1, \nu_i^2\}$, ..., $\{\overline{\nu}_i^{\overline{\gamma}_i-1}, \nu_i^{\overline{\gamma}_i}\}$, $\{\overline{\nu}_i^{\overline{\gamma}_i}, \overline{x}_i\}$;
- for each $x_i \in X''$ such that $\gamma_i \geq 2$ and $\overline{\gamma}_i = 0$, we create the clauses $\{\overline{x}_i, \mu_i^2\}$, $\{\overline{\mu}_i^2, \mu_i^3\}$, ..., $\{\overline{\mu}_i^{\gamma_i-1}, \mu_i^{\gamma_i}\}$, $\{\overline{\mu}_i^{\gamma_i}, x_i\}$;
- finally, for each $x_i \in X''$ such that $\gamma_i \geq 2$ and $\overline{\gamma}_i \geq 1$, we create the clauses $\{\overline{x}_i, \mu_i^2\}$, $\{\overline{\mu}_i^2, \mu_i^3\}$, ..., $\{\overline{\mu}_i^{\gamma_i}, \mu_i^{\gamma_i+1}\}$, $\{\overline{\mu}_i^{\gamma_i+1}, x_i\}$, $\{\mu_i^{\gamma_i+1}, \nu_i^1\}$, $\{\overline{\nu}_i^1, \nu_i^2\}$, ..., $\{\overline{\nu}_i^{\overline{\gamma}_i-1}, \nu_i^{\overline{\gamma}_i}\}$, $\{\overline{\nu}_i^{\overline{\gamma}_i}, \nu_i^{\overline{\gamma}_i+1}\}$, $\{\nu_i^{\overline{\gamma}_i+1}, \overline{\nu}_i^{\overline{\gamma}_i+1}\}$.

It is easy to see that every variable in X appears exactly twice positive and once negative in $f(I)$. Additionally, note that $|X| \leq 2|X''| + \sum_{x_i \in X''}(\gamma_i + \overline{\gamma}_i) \leq 2|X''| + 3|\mathcal{C}''|$ and $|\mathcal{C}| \leq |\mathcal{C}''| + 3|X''| + \sum_{x_i \in X''}(\gamma_i + \overline{\gamma}_i) \leq 4|\mathcal{C}''| + 3|X''|$.

Suppose that I is a YES instance of 1-IN-3-SAT, and let $\alpha\colon X'' \to \{true, false\}$ be a truth assignment that satisfies all clauses in $\mathcal{C}''$. From α, we defined a truth assignment $\beta\colon X \to \{true, false\}$, as follows: $\beta(x_i) = \alpha(x_i)$ for all $x_i \in X''$; $\beta(\mu_i^j) = \alpha(x_i)$ for all $j \in \{2, \ldots, \gamma_i\} \cup \{\gamma_i + 1 \mid \gamma_i \geq 2, \overline{\gamma}_i \geq 1\}$ and all $x_i \in X''$; and $\beta(\nu_i^j) = \overline{\alpha(x_i)}$ for all $j \in \{1, \ldots, \overline{\gamma}_i\} \cup \{\overline{\gamma}_i + 1 \mid \overline{\gamma}_i \geq 1\}$ and all $x_i \in X''$. One may verify that β satisfies all clauses in $\mathcal{C}''$. Therefore, $f(I)$ is a YES instance of 1-IN-3-SAT(3).

Conversely, suppose that $f(I)$ is a YES instance of 1-IN-3-SAT(3), and let $\beta\colon X \to \{true, false\}$ be a truth assignment that satisfies all clauses in $\mathcal{C}$. Note that, the truth assignment $\alpha\colon X'' \to \{true, false\}$, where $\alpha(x_i) = \beta(x_i)$ for all $x_i \in X''$, satisfies all clauses in $\mathcal{C}''$. Therefore I is YES instance of 1-IN-3-SAT.

To conclude this proof, note that, based on the above reductions, the existence of a $2^{o(|X|+|\mathcal{C}|)}$-time algorithm for 1-IN-3-SAT(3) implies the existence of a $2^{o\left(|X'|+|\mathcal{C}'|\right)}$-time algorithm for 3-SAT. Thus, unless ETH fails, 1-IN-3-SAT(3) cannot be solved in time $2^{o(|X|+|\mathcal{C}|)}$. □

Theorem 2. *S-TCP remains NP-complete and, unless ETH fails, cannot be solved in time $2^{o(\ell+n)}$ even if $\Delta = 3$.*

Proof. Let $I = (X, \mathcal{C})$ be an instance of 1-IN-3-SAT(3), where $X = \{x_1, x_2, \ldots, x_p\}$ and $\mathcal{C} = \{C_1, C_2, \ldots, C_q\}$. We construct from I an instance $f(I) = (G, W, \ell, r)$ of S-TCP, such that $\Delta = 3$, as follows (see Fig. 5):

- first, we create the vertices v_I and w_I and add the edge $v_I w_I$;
- for each variable $x_i \in X$, we create the gadget G_{x_i} such that
  - $V(G_{x_i}) = \{v_{x_i}^2, u_{x_i}^1, u_{x_i}^2, t_{x_i}^1, t_{x_i}^2, f_{x_i}, w_{x_i}\}$ and
  - $E(G_{x_i}) = \{v_{x_i}^2 u_{x_i}^1, u_{x_i}^1 t_{x_i}^1, t_{x_i}^1 t_{x_i}^2, t_{x_i}^2 w_{x_i}, w_{x_i} f_{x_i}, f_{x_i} u_{x_i}^2, u_{x_i}^2 v_{x_i}^2\}$;

  we also create the vertices w'_{x_i} and $v_{x_i}^1$ and add the edges $w'_{x_i} v_{x_i}^1$ and $v_{x_i}^1 v_{x_i}^2$;
- we create a complete strict binary tree T_I, rooted at v_I, whose leaves are the vertices $v_{x_1}^1, v_{x_2}^1, \ldots, v_{x_p}^1$;
- for each clause $C_\iota \in \mathcal{C}$, we create the vertex w_{C_ι}; moreover, we add the edge $t_{x_i}^j w_{C_\iota}$ if one of the literals in C_ι corresponds to the j-th occurrence in I of the positive literal x_i, for $x_i \in X$ and $j \in \{1, 2\}$; on the other hand, we add the edge $f_{x_i} w_{C_\iota}$ if one of the literals in C_ι corresponds to the (only) occurrence in I of the negative literal $\overline{x}_i$;
- we define $W = \{w_I\} \cup \{w_{x_i}, w'_{x_i} \mid x_i \in X\} \cup \{w_{C_\iota} \mid C_\iota \in \mathcal{C}\}$;
- finally, we define $\ell = 2p$ and $r = 2p + |V(T_I)| = 4p - 1$.

One may verify that the maximum degree of G is 3. Furthermore, note that, G has $n = 10p + q$ vertices.

Fig. 5 exemplifies the graph G and the terminal set W of $f(I)$.

Now, we prove that I is a YES instance of 1-IN-3-SAT(3) if and only if $f(I)$ is a YES instance of S-TCP.

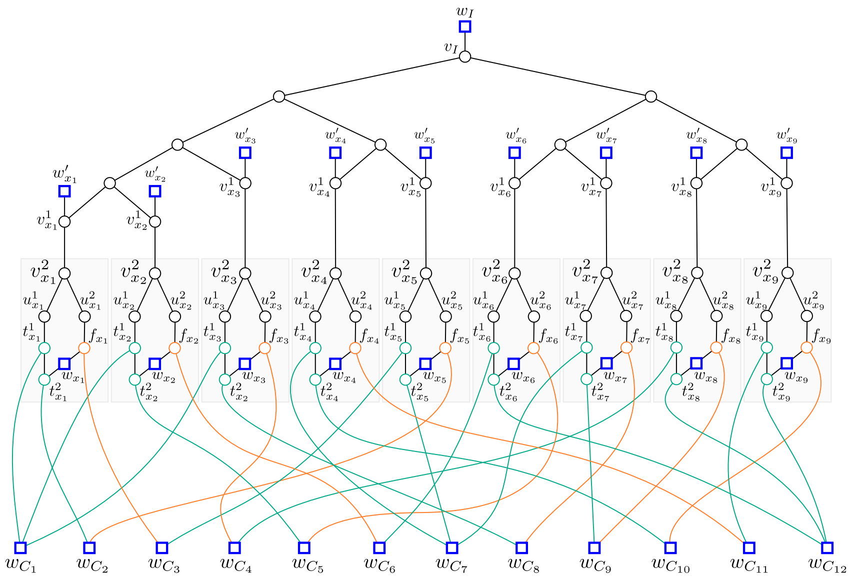


Fig. 5. Graph G and terminal set W (blue square vertices) of $f(I)$ obtained from the instance $I = (X, \mathcal{C})$ of 1-IN-3-SAT(3), where $X = \{x_1, x_2, \dots, x_9\}$ and $\mathcal{C} = \{C_1 = \{x_1, x_2, x_3\}, C_2 = \{x_1, \overline{x}_5\}, C_3 = \{\overline{x}_1, x_5\}, C_4 = \{\overline{x}_3, x_8\}, C_5 = \{x_2, \overline{x}_6\}, C_6 = \{\overline{x}_2, x_6\}, C_7 = \{x_4, x_5, x_7\}, C_8 = \{x_3, \overline{x}_7\}, C_9 = \{x_7, \overline{x}_8\}, C_{10} = \{x_4, \overline{x}_9\}, C_{11} = \{\overline{x}_4, x_9\}, C_{12} = \{x_6, x_8, x_9\}\}$.

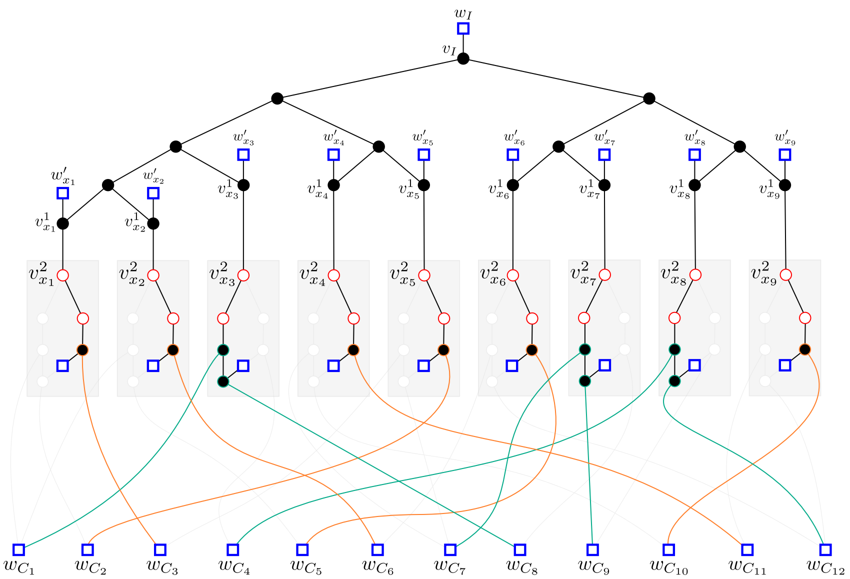


Fig. 6. Strict connection tree of G for W obtained from the truth assignment $\alpha(x_1) = \textit{false}$, $\alpha(x_2) = \textit{false}$, $\alpha(x_3) = \textit{true}$, $\alpha(x_4) = \textit{false}$, $\alpha(x_5) = \textit{false}$, $\alpha(x_6) = \textit{false}$, $\alpha(x_7) = \textit{true}$, $\alpha(x_8) = \textit{true}$ e $\alpha(x_9) = \textit{false}$. (For simplicity, some vertex labels are omitted.)

First, suppose that I is a YES instance of 1-IN-3-SAT(3). Hence, there exists a truth assignment $\alpha\colon X \to \{\textit{true}, \textit{false}\}$ that satisfies all clauses in $\mathcal{C}$. Based on α, we construct a strict connection tree T of G for W as follows:

- we add the complete strict binary tree T_I to T along with the terminal vertex w'_{x_i} and the edges $w'_{x_i} v^1_{x_i}$ for every $x_i \in X$; we also add the vertex w_I and the edge $w_I v_I$ to T;
- for each variable $x_i \in X$, we add the vertices $v^2_{x_i}$ and w_{x_i} and the edge $v^1_{x_i} v^2_{x_i}$ to T; moreover, if $\alpha(x_i) = \textit{true}$, then we add the vertices $u^1_{x_i}$, $t^1_{x_i}$ and $t^2_{x_i}$ to T along with all of their neighbors and incident edges in G; on the other hand, if $\alpha(x_i) = \textit{false}$, then we add the vertices $u^2_{x_i}$ and f_{x_i} to T along with all of their neighbors and incident edges in G.

Fig. 6 exemplifies the strict connection tree T of G for W, referring to the instance $f(I)$ described in Fig. 5, obtained from a truth assignment α.

Clearly, for every clause $C_\iota \in \mathcal{C}$, $w_{C_\iota} \in V(T)$. Furthermore, observe that these terminals are necessarily leaves of T, since by hypothesis every clause $C_\iota \in \mathcal{C}$ has exactly one true literal under the truth assignment α. Thus, it is easy to see that T is a strict connection tree of G for W. Additionally, note that $\mathsf{L}(T) = \{v^2_{x_i} \mid x_i \in X\} \cup \{u^1_{x_i} \mid \alpha(x_i) = \mathit{true}, x_i \in X\} \cup \{u^2_{x_i} \mid \alpha(x_i) = \mathit{false}, x_i \in X\}$ and $\mathsf{R}(T) = V(T_I) \cup \{t^1_{x_i}, t^2_{x_i} \mid \alpha(x_i) = \mathit{true}, x_i \in X\} \cup \{f_{x_i} \mid \alpha(x_i) = \mathit{false}, x_i \in X\}$. Consequently, $|\mathsf{L}(T)| = 2p = \ell$ and $|\mathsf{R}(T)| \le 2p + |V(T_I)| = r$. Therefore, $f(I)$ is a Yes instance of S-TCP.

Conversely, suppose that $f(I)$ is a Yes instance of S-TCP. Hence, there exists a strict connection tree T of G for W such that $|\mathsf{L}(T)| \le \ell$ and $|\mathsf{R}(T)| \le r$. Note that, for every variable $x_i \in X$, the path in T between the terminals w_{x_i} and w'_{x_i}, say P_{x_i}, necessarily consists in one of the following possibilities: either $P_{x_i} = \langle w_{x_i}, t^2_{x_i}, t^1_{x_i}, u^1_{x_i}, v^2_{x_i}, v^1_{x_i}, w'_{x_i} \rangle$ or $P_{x_i} = \langle w_{x_i}, f_{x_i}, u^2_{x_i}, v^2_{x_i}, v^1_{x_i}, w'_{x_i} \rangle$. Hence, $f_{x_i} \in V(T)$ if and only if $t^1_{x_i} \notin V(T)$. Indeed, if f_{x_i} and $t^1_{x_i}$ simultaneously belonged to $V(T)$, then $u_{x^1_i}$, $u_{x^2_i}$, and at least one of the vertices f_{x_i}, $t^1_{x_i}$ or $t^2_{x_i}$ would be linkers of T, which would imply $|\mathsf{L}(T)| > 2p = \ell$. Thus, we can define a truth assignment $\alpha\colon X \to \{\mathit{true}, \mathit{false}\}$ in the following way: $\alpha(x_i) = \mathit{true}$ if $f_{x_i} \notin V(T)$, and $\alpha(x_i) = \mathit{false}$ otherwise.

Let $W_{\mathcal{C}} = \{w_{C_\iota} \mid C_\iota \in \mathcal{C}\}$. Since $W_{\mathcal{C}} \subset W \subseteq V(T)$ and every path in T between the terminal $w_{C_\iota} \in W_{\mathcal{C}}$ and any other terminal $w \in W$ must contain one of the edges $w_{C_\iota} t^1_{x_i}$, $w_{C_\iota} t^2_{x_i}$ or $w_{C_\iota} f_{x_i}$, for some $x_i \in X$, all clauses in $\mathcal{C}$ have at least one true literal under α. Indeed, for every clause $C_\iota \in \mathcal{C}$, if $w_{C_\iota} t^1_{x_i} \in E(T)$ or $w_{C_\iota} t^2_{x_i} \in E(T)$, then $f_{x_i} \notin V(T)$, and so the assignment $\alpha(x_i) = \mathit{true}$ satisfies C_ι; on the other hand, if $w_{C_\iota} f_{x_i} \in E(T)$, then obviously $f_{x_i} \in V(T)$, and so the assignment $\alpha(x_i) = \mathit{false}$ satisfies C_ι. Furthermore, observe that, each clause in $\mathcal{C}$ has no more than one true literal under α, since by hypothesis all terminals belonging to $W \supset W_{C_\iota}$ are leaves of T and, for every vertex $\rho_{x_i} \in \{t^1_{x_i}, t^2_{x_i}, f_{x_i} \mid x_i \in X\} \cap V(T)$, we have that $N_T(\rho_{x_i}) = N_G(\rho_{x_i})$ — otherwise, $|\mathsf{L}(T)| > 2p = \ell$ or some non-terminal vertex would be a leaf of T. Therefore, I is a Yes instance of 1-in-3-SAT(3).

To conclude this proof, note that, since $n = 10p + q$ and $\ell = 2p$, the existence of a $2^{o(\ell+n)}$-time algorithm for S-TCP, even if $\Delta = 3$, implies the existence of a $2^{o(p+q)}$-time algorithm for 1-in-3-SAT(3). Therefore, based on Proposition 1, unless ETH fails, S-TCP cannot be solved in time $2^{o(\ell+n)}$ even if $\Delta = 3$. □

2.2. *Tractable case: maximum degree 3 and bounded number of linkers*

Motivated by Theorems 1 and 2, which state that S-TCP remains NP-complete when $\Delta = 4$ — even if ℓ is bounded by a constant — and when $\Delta = 3$, respectively, we now prove that, if ℓ is bounded by a constant and $\Delta = 3$, then S-TCP is polynomial-time solvable. More specifically, we show that S-TCP can be solved in time $2^{\mathcal{O}(\ell \log n)}$ when $\Delta = 3$. The following proposition and the following lemmas provide the basis of this result.

Let G be a graph, and let $W \subseteq V$ such that $|W| \ge 3$. Given a strict connection tree T of G for W, we denote by $\mathsf{L}'(T)$ the subset of $\mathsf{L}(T)$ defined as follows: $v \in \mathsf{L}'(T)$ if and only if $v \in \mathsf{L}(T)$ and v belongs to a path in T whose (two) endpoints are routers of T; and we denote by T^* the subgraph of T induced by the vertices belonging to $\mathsf{L}'(T) \cup \mathsf{R}(T)$. Observe that, since $|W| \ge 3$, $\mathsf{R}(T) \ne \varnothing$, and thus we have that T^* is well-defined, containing at least one vertex. Furthermore, observe that T^* is a tree.

Proposition 2. *Let $G = (V, E)$ be a graph, and let $W \subseteq V$ such that $|W| \ge 3$. If T is a strict connection tree of G for W, then* $\left\lceil \frac{|W|-2}{\Delta-2} \right\rceil \le |\mathsf{R}(T)| \le |W| - 2$.

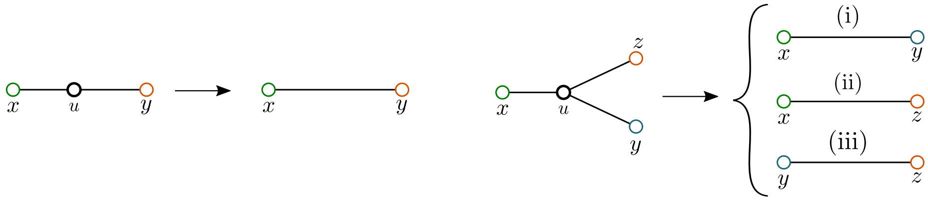
Proof. Note that, $|W| = \sum_{v \in \mathsf{R}(T)} (d_T(v) - d_{T^*}(v))$. Furthermore, we have that

$$\begin{aligned}\sum_{v \in \mathsf{R}(T)} d_{T^*}(v) &= \sum_{v \in V(T^*) \setminus \mathsf{L}'(T)} d_{T^*}(v) = \sum_{v \in V(T^*)} d_{T^*}(v) - \sum_{v \in \mathsf{L}'(T)} d_{T^*}(v) \\ &= 2|E(T^*)| - 2|\mathsf{L}'(T)| = 2\left(|\mathsf{L}'(T)| + |\mathsf{R}(T)| - 1\right) - 2|\mathsf{L}'(T)| \\ &= 2|\mathsf{R}(T)| - 2.\end{aligned}$$

Thus, $|W| = \sum_{v \in \mathsf{R}(T)} d_T(v) - 2|\mathsf{R}(T)| + 2$. Finally, observe that $3|\mathsf{R}(T)| \le \sum_{v \in \mathsf{R}(T)} d_T(v) \le \Delta|\mathsf{R}(T)|$. Therefore, $\left\lceil \frac{|W|-2}{\Delta-2} \right\rceil \le |\mathsf{R}(T)| \le |W| - 2$. □

By Proposition 2, we can assume without loss of generality that $r = |W| - 2$ whenever $\Delta = 3$. Thus, in the remainder of this section, we omit the input aspect r in the description of the instances of S-TCP. Furthermore, we also assume that G does not contain edges whose endpoints are both terminals, i.e. W is an independent set of G.

Lemma 1. *Let $G = (V, E)$ be a graph such that $\Delta = 3$. Then, G admits a strict connection tree T for W with $\mathsf{L}(T) = \varnothing$ if and only if there exists a subset $V' \subseteq V$ such that $T' = G[V']$ is a strict connection tree for W with $\mathsf{L}(T') = \varnothing$.*



(a) Case $d_G(u) = 2$. (b) Case $d_G(u) = 3$.

Fig. 7. Operations corresponding to the possible configurations of $u \in L$ being a linker in a strict connection tree of G for W.

Proof. First, suppose that G admits such a tree T. Since $\mathsf{L}(T) = \varnothing$, all non-terminal vertices of T are routers, i.e. $V(T) \setminus W = \mathsf{R}(T)$. Thus, if $\rho \in V(T) \setminus W$, then $N_T(\rho) = N_G(\rho)$, otherwise $\rho \notin \mathsf{R}(T)$ since $\Delta = 3$. Finally, since W is an independent set of G, we have $G[V'] = T$, where $V' = V(T)$.

Conversely, it is immediate that if there exists a subset V' such that $T' = G[V']$ is a strict connection tree for W with $\mathsf{L}(T') = \varnothing$, then G admits a strict connection tree T for W with $\mathsf{L}(T) = \varnothing$. Indeed, T' is itself such a tree T. □

Given a terminal vertex $w \in W$ and a non-terminal vertex $v \in N_G(w)$, we denote by $\mathcal{H}_v$ the subgraph of G induced by the vertices belonging to $V(\mathcal{H}'_v) \cup W$, where $\mathcal{H}'_v$ is the component of $G - W$ that contains v.

Lemma 2. *Let $G = (V, E)$ be a graph such that $\Delta = 3$, and let $w \in W$ be an arbitrary terminal. Then, there exists a subset $V' \subseteq V$ such that $T' = G[V']$ is a strict connection tree for W with $\mathsf{L}(T') = \varnothing$ if and only if, for some non-terminal vertex $v \in N_G(w)$, $\mathcal{H}_v$ is a strict connection tree for W with $\mathsf{L}(\mathcal{H}_v) = \varnothing$.*

Proof. First, suppose that there exists such a subset $V' \subseteq V$. Let $v \in N_{T'}(w)$ be the only neighbor of w in T', where $T' = G[V']$. Since by hypothesis all terminals in W are leaves of T', we have that $T' - W = G[V' \setminus W]$ is connected. Consequently, all non-terminal vertices $\rho \in V' \setminus W$ (including v) belong to a same component of $G - W$. Thus, $V' \setminus W \subseteq V(\mathcal{H}'_v)$. On the other hand, since $\Delta = 3$ and $|\mathsf{L}(T')| = \varnothing$, $N_{T'}(\rho) = N_G(\rho)$ for every $\rho \in V' \setminus W$. Consequently, we have that $V' \setminus W \supseteq V(\mathcal{H}'_v)$. Thus, $V' = V(\mathcal{H}'_v) \cup W$ and, therefore, $\mathcal{H}_v$ is a strict connection tree for W such that $\mathsf{L}(\mathcal{H}_v) = \varnothing$.

Conversely, suppose that, for some non-terminal vertex $v \in N_G(w)$, $\mathcal{H}_v$ is a strict connection tree for W with $\mathsf{L}(\mathcal{H}_v) = \varnothing$. Therefore, for $V' = V(\mathcal{H}_v)$, we have that $T' = G[V']$ is a strict connection tree for W such that $\mathsf{L}(T') = \varnothing$. □

Corollary 1. *S-TCP is linear-time solvable when $\ell = 0$ and $\Delta = 3$.*

Proof. Let $I = (G, W)$ be a given instance of S-TCP with $\ell = 0$ and $\Delta = 3$. Let $w \in W$ be an arbitrary terminal vertex. It is easy to see that, for every non-terminal vertex $v \in N_G(w)$, the graph $\mathcal{H}_v$ can be constructed in time linear in the size of I; for instance, we can obtain $\mathcal{H}_v$ by running the variant of the breadth-first search rooted at v on which the terminal vertices are not explored (i.e. they must be leaves in the resulting search tree). Moreover, we can also verify in linear-time whether $\mathcal{H}_v$ is a strict connection tree for W such that $\mathsf{L}(\mathcal{H}_v) = \varnothing$. Therefore, it follows from Lemmas 1 and 2 that S-TCP is linear-time solvable if $\Delta = 3$ and $\ell = 0$. □

Theorem 3. *S-TCP can be solved in time $2^{\mathcal{O}(\ell \log n)}$ when $\Delta = 3$, but assuming ETH there is no $2^{o(\ell+n)}$-time algorithm for the problem.*

Proof. We first prove that S-TCP can be solved in time $2^{\mathcal{O}(\ell \log n)}$ when $\Delta = 3$. Let $I = (G, W, \ell)$ be an instance of S-TCP such that $\Delta = 3$. For each subset $L \subseteq V \setminus W$ such that $|L| \leq \ell$, we generate all combinations of graphs G'_L obtained from G by successively applying, for each vertex $u \in L$, the operation depicted in Fig. 7a if $d_G(u) = 2$, or one of the three operations depicted in Fig. 7b if $d_G(u) = 3$. These operations simulate the possible configurations of the vertex $u \in L$ being a linker in a strict connection tree of G for W. In the end, after all vertices belonging to L have been processed as described above, we verify whether the resulting graphs G'_L admit a strict connection tree T' for W such that $\mathsf{L}(T') = \varnothing$. Then, the algorithm returns that I is a Yes instance of S-TCP if and only, for some subset $L \subseteq V \setminus W$, with $|L| \leq \ell$, there exists a graph G'_L that admits such a tree T'. Algorithm 1 presents this Turing reduction formally, where the function Get-tree-without-linkers, with input (G, W), denotes a linear-time procedure for solving S-TCP with $\ell = 0$, when $\Delta = 3$, that returns either a strict connection tree T of G for W such that $\mathsf{L}(T) = \varnothing$ or *null* if such a tree does not exist.

The correctness of the algorithm follows from the fact that all possible relevant configurations for the existence of a strict connection tree of G for W with at most ℓ linkers are analyzed. In fact, if there exists a graph G'_L that admits a strict connection tree T'_L for W such that $\mathsf{L}(T'_L) = \varnothing$, for some subset $L \subseteq V \setminus W$, with $|L| \leq \ell$, then by construction G admits a strict connection tree T for W such that $\mathsf{L}(T) \subseteq L$, and thus we have that $I = (G, W, \ell)$ is a Yes instance of S-TCP. Conversely, if G admits a strict connection tree T for W such that $|\mathsf{L}(T)| \leq \ell$, then clearly, for $L = \mathsf{L}(T)$, there exists a graph G'_L that admits a strict connection tree T'_L for W such that $\mathsf{L}(T'_L) = \varnothing$, and thus we have that $I' = (G'_L, W)$ is a Yes instance of S-TCP with $\ell = 0$.

Algorithm 1: Turing reduction from S-TCP to S-TCP with $\ell = 0$, restricted to graphs with maximum degree 3.

Input: A graph $G = (V, E)$ such that $\Delta = 3$, a terminal set $W \subseteq V$ and a non-negative integer ℓ.
Output: A strict connection tree T of G for W such that $|\mathsf{L}(T)| \leq \ell$ and $|\mathsf{R}(T)| \leq r$, or *null* if such a tree does not exist.

Let $\mathcal{L}$ be a collection of all subsets $L \subseteq V \setminus W$, with $|L| \leq \ell$, such that the sets in $\mathcal{L}$ are ordered according to their inclusion order.
foreach $L \in \mathcal{L}$ **do**
    $T :=$ Get-tree-given-linker-superset$(G, W, L, 1)$
    if $T \neq$ null **then** **return** T
return *null*

Function Get-tree-given-linker-superset(G, W, L, i)
    if $i > |L|$ **then** **return** Get-tree-without-linkers(G, W)
    $u := L[i]$ // $L[i]$ denotes the i-th element in L
    if $d_G(u) = 1$ **then** **return** *null*
    foreach $x, y \in N_G(u)$ *such that* $x \neq y$ **do**
        Let G'_L be the graph defined as follows: $V(G'_L) := V \setminus \{u\}$ and $E(G'_L) := (E \setminus \{uv \mid v \in N_G(u)\}) \cup \{xy\}$
        $T :=$ Get-tree-given-linker-superset$(G'_L, W, L, i+1)$
        if $T \neq$ null **then**
            $E(T) := (E(T) \setminus \{xy\}) \cup \{ux, uy\}$
            return T
    return *null*

Regarding the running time of the algorithm, in the worst case the number of recursive calls to Get-tree-given-linker-superset is

$$\sum_{\substack{L \subseteq V \setminus W \\ |L| \leq \ell}} \prod_{u \in L} \binom{d_G(u)}{2} \leq \sum_{\substack{L \subseteq V \setminus W \\ |L| \leq \ell}} 3^{|L|} \leq \sum_{i=0}^{\ell} \binom{n}{i} 3^i = \mathcal{O}\left(3^\ell \cdot n^\ell\right).$$

Thus, the total time spent by the algorithm is $\mathcal{O}\left(3^\ell \cdot n^{\ell+1}\right)$, since Get-tree-without-linkers runs in time linear in n and all the other operations of the algorithm can be performed in constant time. Therefore, S-TCP can be solved in time $2^{\mathcal{O}(\ell \log n)}$ when $\Delta = 3$.

The proof that S-TCP cannot be solved in time $2^{o(\ell+n)}$, even when $\Delta = 3$, unless ETH fails, follows directly from Theorem 2. □

3. Using ℓ, r and Δ as parameters

In the present section, we investigate the parameterized complexity of S-TCP when ℓ, r and Δ are parameters. We remark that, as a result of Theorem 1, S-TCP parameterized by ℓ and Δ is para-NP-complete; consequently, the problem does not even admit an XP-time algorithm, unless P = NP. On the other hand, Dourado et al. [2] showed that S-TCP parameterized by ℓ and r is in XP.

Nevertheless, we now prove that S-TCP parameterized by ℓ and r is W[2]-hard. Particularly, we show that, for every $\ell \geq 0$, S-TCP parameterized by r is W[2]-hard. Thus, unless FPT = W[2], S-TCP does not admit an algorithm with running time $g(r) \cdot n^{h(\ell)}$, for any computable functions g and h.

Theorem 4. *For every $\ell \geq 0$, S-TCP parameterized by r is W[2]-hard.*

Proof. Let $I = (U, \mathcal{F}, k)$ be an instance of Set cover, a classical W[2]-hard problem [14], where U is the universe, $\mathcal{F}$ is the collection of non-empty sets over U, and k is the parameter of the problem, a non-negative integer. We construct an instance $f(I) = (G, W, r)$ of S-TCP with ℓ bounded by a constant as follows:

- for each $i \in \{1, \ldots, \ell\}$, we create the vertices u_i and w'_i, and we add the edge $u_i w'_i$; let $L = \{u_1, u_2, \ldots, u_\ell\}$;
- for each set $F_i \in \mathcal{F}$, we create the vertex v_{F_i}; moreover, for each pair $F_i, F_j \in \mathcal{F}$ with $i \neq j$, we add the edge $v_{F_i} v_{F_j}$; let $K_{\mathcal{F}} = \{v_{F_i} \mid F_i \in \mathcal{F}\}$;
- we create the vertices w_a and w_b and, for each set $F_i \in \mathcal{F}$, we add the edges $w_a v_{F_i}$ and $w_b v_{F_i}$;
- for each pair of vertices u_i, v_{F_j}, where $u_i \in L$ and $v_{F_j} \in K_{\mathcal{F}}$, we add the edge $u_i v_{F_j}$;
- for each element $x_i \in U$, we create the vertex w_i;
- for each set $F_j \in \mathcal{F}$ and each element $x_i \in F_j$, we add the edge $v_{F_j} w_i$;
- finally, we define $W = \{w_a, w_b\} \cup \{w'_i \mid i \in \{1, \ldots, \ell\}\} \cup \{w_i \mid x_i \in U\}$ and $r = k$.

Now we prove that I is a Yes instance of Set cover if and only if $f(I)$ is a Yes instance of S-TCP with ℓ bounded by a constant.

First, suppose that I is a Yes instance of Set cover, and let $\mathcal{F}' = \{F'_1, F'_2, \dots, F'_z\}$ be a subcollection of $\mathcal{F}$ such that $\bigcup_{F' \in \mathcal{F}'} F' = U$ and $z \le k$. Assume without loss of generality that $\mathcal{F}'$ is minimal with respect to the property of covering all elements of U, i.e. for any set $F' \in \mathcal{F}'$, $\mathcal{F}' \setminus \{F'\}$ does not cover all elements of U. Based on $\mathcal{F}'$, we construct a strict connection tree T of G for W as follows:

- for each $i \in \{1, 2, \dots, \ell\}$, we add the vertices u_i and w'_i to T along with the edges $u_i w'_i$;
- for each set $F'_j \in \mathcal{F}'$, we add the vertex $v_{F'_j}$ to T;
- for each $i \in \{1, 2, \dots, z-1\}$, we add the edge $v_{F'_i} v_{F'_{i+1}}$ to T;
- moreover, for each $i \in \{1, 2, \dots, \ell\}$, we add the edge $u_i v_{F'_1}$ to T; we also add the edges $w_a v_{F'_1}$ and $w_b v_{F'_z}$ to T;
- finally, for each element $x_i \in U$, we add the vertex w_i to T along with the edge $v_{F'_j} w_i$, where $j = \min\{1, \dots, z\}$ such that $x_i \in F'_j$ and $F'_j \in \mathcal{F}'$.

It is easy to verify that T is a strict connection tree of G for W. Now, we prove that $|\mathsf{L}(T)| \le \ell$ and $|\mathsf{R}(T)| \le r$. First, note that the vertices $u_1, u_2, \dots, u_\ell$ have degree exactly 2 in T. Thus, $\mathsf{L}(T) \supseteq L$. On the other hand, it follows from the minimality of $\mathcal{F}'$ that, for every set $F'_j \in \mathcal{F}$, the vertex $v_{F'_j}$ is adjacent to at least one terminal $w_i \in W$, since F'_j covers at least one element $x_i \in U$ which is not covered by any other set in $\mathcal{F}'$. Consequently, every vertex $v_{F'_j}$ has degree at least 3 in T, for $F'_j \in \mathcal{F}'$. Thus, $\mathsf{L}(T) = L$ and $\mathsf{R}(T) = \{v_{F'_j} \mid F'_j \in \mathcal{F}'\}$, which implies $|\mathsf{L}(T)| \le \ell$ and $|\mathsf{R}(T)| = z \le k = r$. Therefore, $f(I)$ is a Yes instance of S-TCP with ℓ bounded by a constant.

Conversely, suppose that G admits a strict connection tree T for W such that $|\mathsf{L}(T)| \le \ell$ and $|\mathsf{R}(T)| \le r = k$. Note that, for every $i \in \{1, 2, \dots, \ell\}$, the path in T between the terminal w'_i and any other terminal belonging to W necessarily contains the vertex $u_i \in L$. Hence, $V(T) \supseteq L$. Furthermore, since the vertices in $W \supset \{w_i \mid x_i \in U\}$ are leaves of T, there is a vertex $v_{F_j} \in K_{\mathcal{F}}$ such that $N_T(w_i) = \{v_{F_j}\}$ for every $x_i \in U$. However, the vertices v_{F_j} are non-terminal and T contains at most $\ell + r$ non-terminal vertices. Thus, since $V(T) \supseteq L$ and $|L| = \ell$, there are at most r vertices belonging to $K_{\mathcal{F}}$ in T, i.e. $|V(T) \cap K_{\mathcal{F}}| \le r = k$. Then, $\mathcal{F}' = \{F_j \mid v_{F_j} \in V(T) \cap K_{\mathcal{F}}\}$ is a subcollection of $\mathcal{F}$ such that $|\mathcal{F}'| \le k$. Finally, it follows from the fact that $W \subseteq V(T)$ that $\bigcup_{F' \in \mathcal{F}'} F' = U$. Therefore, I is a Yes instance of Set cover. □

Based on the technique called bounded search tree, Dourado et al. [2] provided an $\mathcal{O}\left((2^{\Delta-1})^{\ell+r} \Delta n\right)$-time algorithm for S-TCP. As an immediate result, they proved that if, besides ℓ and r, the maximum degree Δ of G is also considered as a parameter, then S-TCP is in FPT. We now present an alternative, but substantially simpler, proof for the tractability of S-TCP parameterized by ℓ, r and Δ, which consists in a kernelization algorithm for the problem derived from the following reduction rules.

Reduction rule 1. For any two terminals $w_1, w_2 \in W$, if the distance between them in $G - (W \setminus \{w_1, w_2\})$ is greater than $\ell + r + 1$, then conclude that G does not admit a strict connection for W with at most ℓ linkers and at most r routers.

Reduction rule 2. Let $v \in V \setminus W$ and $w \in W$. If the distance between v and w in $G - (W \setminus \{w\})$ is greater than $\ell + r + 1$, then remove v from G.

Lemma 3. *Reduction rules 1 and 2 are safe.*

Proof. Suppose that there exist $w_1, w_2 \in W$ such that the distance between w_1 and w_2 in $G - (W \setminus \{w_1, w_2\})$ is greater than $\ell + r + 1$. Thus, every strict connection tree of G for $W \supseteq \{w_1, w_2\}$ has more than ℓ linkers or more than r routers. Therefore, we are dealing with a No instance of the problem in this case, and so Reduction rule 1 is indeed safe. Now, suppose that there exists a non-terminal vertex $v \in V \setminus W$ such that, for some terminal $w \in W$, the distance between v and w in $G - (W \setminus \{w\})$ is greater than $\ell + r + 1$. Note that, v does not belong to any strict connection tree T of G for $W \supset \{w\}$, otherwise T would have more than ℓ linkers or more than r routers. Thus, G admits a strict connection tree for W with at most ℓ linkers and at most r routers if and only if $G - v$ admits a strict connection tree for W with at most ℓ linkers and at most r routers. Therefore, we have that Reduction rule 2 is also safe. □

Theorem 5. *S-TCP admits a kernel with $\mathcal{O}\left(\Delta^{2(\ell+r+1)}\right)$ vertices.*

Proof. Based on Reduction rule 1, suppose that, for every pair of terminal vertices $w_1, w_2 \in W$, the distance between them in $G - (W \setminus \{w_1, w_2\})$ is at most $\ell + r + 1$. Moreover, while it is possible, apply Reduction rule 2 successively. Let G' denote the resulting graph. Note that, the distance between any non-terminal vertex $v \in V(G') \setminus W$ and any terminal vertex $w \in W$ in $G' - (W \setminus \{w\})$ is at most $\ell + r + 1$. Moreover, the distance between any two non-terminal vertices $u, v \in V \setminus W$ in G' is at most $2(\ell + r + 1)$. Thus, the diameter of G' is at most $2(\ell + r + 1)$ and, therefore, the number of vertices in G' is $\mathcal{O}\left(\Delta^{2(\ell+r+1)}\right)$. □

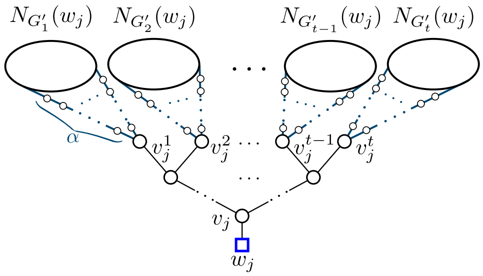


Fig. 8. Complete strict binary tree T_j and paths $P(v_j^i, x)$, where $x \in N_{G'_i}(w_j)$, for $i \in \{1, 2, \dots, t\}$ and $j \in \{1, 2, \dots, k\}$.

At this point, a natural question that arises is whether there exists a polynomial kernel for S-TCP parameterized by ℓ, r and Δ. However, we show that this does not seem to be the case, the existence of such a kernel is unlikely. By using a framework developed by Bodlaender *et al.* [31,32], called *cross-decomposition*, we prove that S-TCP parameterized by ℓ, r and Δ does not admit a polynomial kernel, unless NP $\subseteq$ coNP/poly.

By Proposition 2, if a graph G with maximum degree Δ admits a strict connection tree T for a terminal set W with at most r routers, then $\left\lceil \frac{|W|-2}{\Delta-2} \right\rceil \leq r$, which implies $|W| \leq r(\Delta-2)+2$. Thus, if r and Δ are parameters, then, without loss of generality, $|W|$ can be considered as a parameter as well.

Theorem 6. *S-TCP parameterized by ℓ, r, Δ and $|W|$ does not admit a polynomial kernel, unless NP $\subseteq$ coNP/poly.*

Proof. We prove this theorem by showing that S-TCP cross-composes into S-TCP parameterized by ℓ, r, Δ and $|W|$. We first need to present a polynomial equivalence relation. Thus, let $\mathcal{R}$ be the equivalence relation defined as follows: all bitstrings which do not encode a valid instance of S-TCP belong to a same equivalence class; and two well-formed instances (G_1, W_1, ℓ_1, r_1) and (G_2, W_2, ℓ_2, r_2) of S-TCP belong to a same equivalence class if and only if $|V(G_1)| = |V(G_2)|$, $\Delta(G_1) = \Delta(G_2)$, $|W_1| = |W_2|$, $\ell_1 = \ell_2$ and $r_1 = r_2$. One may verify that any set of well-formed instances of S-TCP on at most n vertices each can be partitioned into $\mathcal{O}(n^5)$ equivalence classes. Therefore, $\mathcal{R}$ is a polynomial equivalence relation.

Let $I'_1, I'_2, \dots, I'_t$ be $t \geq 1$ input instances which are equivalent under $\mathcal{R}$. If such instances are not well-formed, then we output a single trivial No instance of S-TCP parameterized by ℓ, r, Δ and $|W|$. Thus, assume that all of the input instances $I'_1, I'_2, \dots, I'_t$ are well-formed and encode structures $(G'_1, W'_1, \ell'_1, r'_1)$, $(G'_2, W'_2, \ell'_2, r'_2)$, ..., $(G'_t, W'_t, \ell'_t, r'_t)$, respectively. For simplicity, we also assume without loss of generality that, for every $i \in \{1, 2, \dots, t\}$, $\ell'_i = \ell' \leq |V(G'_i)|$, $r'_i = r' \leq |V(G'_i)|$, $\Delta(G'_i) = \gamma \geq 3$ and W'_i is an independent set of G'_i such that $W'_i = \{w_1, w_2, \dots, w_k\}$, where $k \geq 3$. Then, we compose $I'_1, I'_2, \dots, I'_t$ into a single instance $I = (G, W, \ell, r)$ of S-TCP parameterized by ℓ, r, Δ and $|W|$, as follows:

- we add the vertices $w_1, w_2, \dots, w_k$ to G;
- for each $j \in \{1, 2, \dots, k\}$, we create the vertices $v_j, v_j^1, v_j^2, \dots, v_j^t$ and a complete strict binary tree T_j, rooted at v_j, whose leaves are the vertices $v_j^1, v_j^2, \dots, v_j^t$; moreover, we add the edge $w_j v_j$ to G;
- for each $i \in \{1, 2, \dots, t\}$, we add all the vertices and all the edges of the graph $G'_i - W'_i$ to G;
- for each $i \in \{1, 2, \dots, t\}$, for each $j \in \{1, 2, \dots, k\}$ and for each vertex $x \in N_{G'_i}(w_j)$, we add the edge $v_j^i x$ to G and subdivide this edge into α new vertices, where $\alpha = \ell' + k\lceil \log_2 t \rceil + 1$; in other words, for each vertex $x \in N_{G'_i}(w_j)$, we create the vertices $x_1^*, x_2^*, \dots, x_\alpha^*$ and the path $P(v_j^i, x) = \langle v_j^i, x_1^*, x_2^*, \dots, x_\alpha^*, x \rangle$ (see Fig. 8);
- finally, we define $W = \{w_1, w_2, \dots, w_k\}$, $\ell = (k+1)\alpha - 1$ and $r = r'$.

Fig. 9 illustrates the overall structure of G and W.

One may easily verify that I can be constructed in time polynomial in $\sum_{i=1}^{t} |I'_i|$, and that $\Delta(G) \leq \gamma + 1$, $|W| = k$, $\ell = \mathcal{O}(k\ell' + k^2 \log_2 t)$ and $r = r'$.

We now proof that there exists $i \in \{1, 2, \dots, t\}$ such that I'_i is a Yes instance of S-TCP if and only if I is a Yes instance of S-TCP parameterized by ℓ, r, Δ and $|W|$.

First, suppose that, for some $i \in \{1, 2, \dots, t\}$, I'_i is a Yes instance of S-TCP, and let T' be a strict connection tree of G'_i for W'_i such that $|\mathsf{L}(T')| \leq \ell'$ and $|\mathsf{R}(T')| \leq r'$. Then, consider the subgraph T of G defined as follows:

$$V(T) = V(T') \cup \bigcup_{w_j \in W} V(P(w_j, v_j^i)) \cup \bigcup_{\substack{x \in N_{T'}(w_j) \\ w_j \in W}} V(P(v_j^i, x)) \text{ and}$$

$$E(T) = \big(E(T') \setminus \{w_j x \mid w_j \in W\}\big) \cup \bigcup_{w_j \in W} E(P(w_j, v_j^i)) \cup \bigcup_{\substack{x \in N_{T'}(w_j) \\ w_j \in W}} E(P(v_j^i, x)),$$

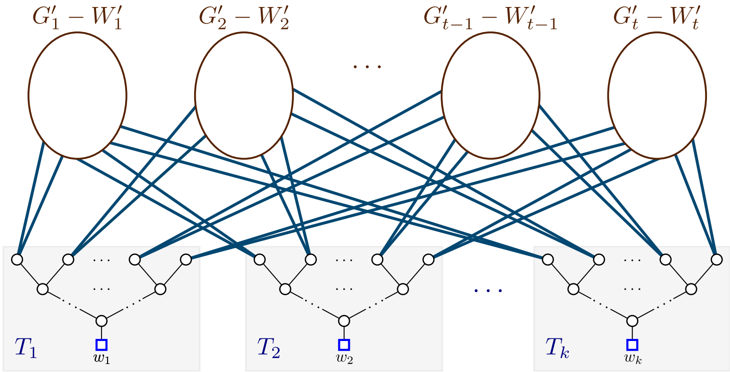


Fig. 9. Graph G and terminal set W. For simplicity, the vertices $x_1^*, x_2^*, \ldots, x_\alpha^*$ corresponding to the subdivision of the edges $v_j^i x$ are omitted, where $x \in N_{G_i'}(w_j)$.

where $P(w_j, v_j^i)$ denotes the only path in T_j between the vertices w_j and v_j^i. It is easy to see that T is a strict connection tree of G for W such that $|\mathsf{L}(T)| \le |\mathsf{L}(T')| + k\lceil \log_2 t \rceil + k\alpha \le (k+1)\alpha - 1 = \ell$ and $|\mathsf{R}(T)| \le r' = r$, since $\mathsf{L}(T) = \mathsf{L}(T') \cup \bigcup_{\substack{x \in N_{T'}(w_j) \\ w_j \in W}} \left(V(P(v_j^i, x)) \setminus \{x\} \right) \cup \bigcup_{w_j \in W} \left(V(P(w_j, v_j^i)) \setminus \{w_j\} \right)$ and $\mathsf{R}(T) = \mathsf{R}(T')$. Therefore, I is a Yes instance of S-TCP.

Conversely, suppose that I is a Yes instance of S-TCP, and let T be a strict connection tree of G for W such that $|\mathsf{L}(T)| \le \ell$ and $|\mathsf{R}(T)| \le r$. Note that, for each $w_j \in W$, all paths in G between w_j and any other terminal must contain a path $P(v_j^i, x)$ as a subpath, for some $x \in N_{G_i'}(w_j)$ and some $i \in \{1, 2, \ldots, t\}$; so, since $W \subseteq V(T)$, for each $w_j \in W$, T contains a path $P(v_j^i, x)$. Thus, $|\mathsf{L}(T)| \ge k\alpha$. Consequently, for every $j \in \{1, 2, \ldots, k\}$, the intersection between T and the complete binary tree T_j can only contain the path $P(w_j, v_j^i)$ of T_j whose endpoints are w_j and v_j^i, otherwise: T would have a leaf which is not a terminal; or, besides $P(v_j^i, x)$, T would have a further path $P(v_j^\iota, y)$, for some $y \in N_{G_\iota'}(w_j)$ and some $\iota \in \{1, 2, \ldots, t\}$ with $\iota \ne i$, which would imply $|\mathsf{L}(T)| \ge (k+1)\alpha > \ell$. By similar reasons, we have that the degree of v_j^i in T must be equal to 2. Hence, for every $j \in \{1, 2, \ldots, k\}$, the subgraph of T_j in T – i.e. the path $P(w_j, v_j^i)$ – can be viewed as a leaf of T (in the sense that its only purpose in T is connecting the terminal w_j, as a leaf of T). Consequently, there exists precisely one index $i \in \{1, 2, \ldots, t\}$ such that $V(T) \cap V(G_i' - W_i') \ne \varnothing$, otherwise T would be disconnected. Therefore, I_i' is a Yes instance of S-TCP. Indeed, the graph T', where $V(T') = \left(V(T) \cap V(G_i' - W_i')\right) \cup W_i'$ and $E(T') = \left(E(T) \cap E(G_i' - W_i')\right) \cup \{w_j x \mid v_j^i x \in E(T)\}$, is a strict connection tree for W_i' such that $|\mathsf{L}(T')| \le \ell'$ and $|\mathsf{R}(T')| \le r'$, since $\mathsf{L}(T') = \mathsf{L}(T) \cap V(G_i' - W_i')$ and $\mathsf{R}(T') = \mathsf{R}(T)$. □

4. The split graph case

A *split graph* is a graph whose vertex set can be partitioned into a clique and an independent set.

S-TCP on split graphs may have interesting applications in IoT (*Internet of Things*), where devices with high communicating/processing power (such as wireless routers) are modeled as a cluster, while devices with low communicating/processing power (such as wireless printers) are modeled as an independent set, being able to send (receive, resp.) messages just to (from, resp.) devices of the cluster. Thus, motivated by applications in IoT and by the fact that it is well-known that Steiner tree is NP-complete on split graphs [6], we analyze in this section the complexity of S-TCP restricted to split graphs.

More specifically, we prove that S-TCP restricted to split graphs can be solved in time $n^{\mathcal{O}(r)}$, implying thereby that S-TCP on split graphs is polynomial-time solvable when r is bounded by a constant. On the other hand, we extend Theorem 4 by showing that S-TCP parameterized by r remains W[2]-hard even if it is restricted to split graphs and ℓ is bounded by a constant; furthermore, we show that, for any computable function g, there is no $g(r) \cdot n^{o(r)}$-time algorithm for the problem, unless ETH fails.

Given an instance $I = (G, W, \ell, r)$ of S-TCP, where $G = (V, E)$ is a split graph. We assume throughout this section that $V = K \cup S$, where K is a maximal clique and S is a maximal independent set of G. We also assume that $r \ge 1$ and $|W| \ge 3$.

Fact 1. If $K \subseteq W$, then G does not admit a strict connection tree for W.

Fact 2. If $K \setminus W \ne \varnothing$ and $W \cap S = \varnothing$ (i.e. $W \subset K$), then G admits a strict connection tree T for W such that $|\mathsf{L}(T)| = 0$ and $|\mathsf{R}(T)| = 1$.

Lemma 4. *Suppose that $K \setminus W \neq \varnothing$ and $W \cap S \neq \varnothing$. If G admits a strict connection tree T' for W, then there exists a strict connection tree T of G for W such that $|\mathsf{L}(T)| \leq |\mathsf{L}(T')|$, $|\mathsf{R}(T)| \leq |\mathsf{R}(T')|$, and $\mathsf{R}(T) \subseteq K$.*

Proof. Since S is an independent set of G, $N_{T'}(S) \subseteq K$. Moreover, it follows from the assumptions $W \cap S \neq \varnothing$ and $|W| \geq 3$ that $\big(V(T') \setminus W\big) \cap K \neq \varnothing$. Thus, let v be an arbitrary vertex in $\big(V(T') \setminus W\big) \cap K$. Since K is a clique of G, we have that $v \in N_G(v')$ for all $v' \in N_{T'}(S) \setminus \{v\}$. Then, consider the graph T defined as follows: $V(T) = V(T') \setminus \big(\mathsf{R}(T') \cap S\big)$ and $E(T) = \big(E(T') \setminus \{\rho v' \mid v' \in N_{T'}(\rho),\ \rho \in \mathsf{R}(T') \cap S\}\big) \cup \{vv' \mid v' \in N_{T'}(\mathsf{R}(T') \cap S)\}$. One may easily verify that T is a strict connection tree of G for W such that $|\mathsf{L}(T)| \leq |\mathsf{L}(T')|$, $|\mathsf{R}(T)| \leq |\mathsf{R}(T')|$, and $\mathsf{R}(T) \subseteq K$. □

Lemma 5. *Let T' be a strict connection tree of G for W such that $\mathsf{R}(T') \subseteq K$. There exists a strict connection tree T of G for W, with $|\mathsf{L}(T)| \leq |\mathsf{L}(T')|$ and $\mathsf{R}(T) \subseteq \mathsf{R}(T')$, which holds the following properties:*

(i) $\mathsf{L}(T) \subseteq K$;
(ii) *each vertex in $L(T)$ is adjacent to exactly one vertex in $\mathsf{R}(T)$ and exactly one vertex $w \in W$, where $w \in S$ and $w \notin N_G(\mathsf{R}(T))$;*
(iii) *$T[R(T)]$ is a path.*

Proof. (i). Note that, for every vertex $u \in \mathsf{L}(T') \cap S$, if x_u and y_u are the two distinct neighbors of u in T', then $x_u, y_u \in K$. Thus, the graph obtained from T' by removing all the vertices in $\mathsf{L}(T') \cap S$ and adding all the edges in $\{x_u y_u \mid x_u, y_u \in N_{T'}(u), u \in \mathsf{L}(T') \cap S\}$ is a strict connection tree of G for W with linker set $\mathsf{L}(T') \setminus S \subseteq K$ and router set $\mathsf{R}(T')$. Thus, for simplicity, we assume hereinafter that $\mathsf{L}(T') \subseteq K$.

(ii). Since $|W| \geq 3$, for every vertex $u \in \mathsf{L}(T')$, if x_u and y_u are the two distinct neighbors of u in T', then $x_u \notin W$ or $y_u \notin W$, otherwise T' would not be a strict connection tree for W. If $x_u, y_u \notin W$, then $x_u, y_u \in K$. Hence, we can remove u from T' and add the edge $x_u y_u$. Thus, suppose that $x_u \in W$ and $y_u \notin W$. If $y_u \in \mathsf{L}(T')$, then y_u has exactly one neighbor in T' in addition to u. Let z be this second neighbor of y_u in T'. Since $|W| \geq 3$ and we are supposing that $x_u \in W$, we have $z \notin W$, which implies $z \in K$. As a result, the graph obtained from T' by removing y_u and adding the edge uz is a strict connection tree of G for W with linker set $\mathsf{L}(T') \setminus \{y_u\}$ and router set $\mathsf{R}(T')$. Suppose now that $y_u \in \mathsf{R}(T')$ but there exists a vertex $\rho \in \mathsf{R}(T')$, possibly $\rho = y_u$, such that $\rho x_u \in E(G)$. Consequently, the graph H obtained from T' by removing u and adding the edge ρx_u is a strict connection tree of G for W such that $\mathsf{L}(H) = \mathsf{L}(T') \setminus \{u\}$ and $\mathsf{R}(H) = \mathsf{R}(T')$ if $d_{T'}(y_u) > 3$ or $\rho = y_u$, and $\mathsf{L}(H) = \big(\mathsf{L}(T') \setminus \{u\}\big) \cup \{y_u\}$ and $\mathsf{R}(H) = \mathsf{R}(T') \setminus \{y_u\}$ otherwise. Therefore, one may verify that, by applying successively the steps described above, it is always possible to obtain a strict connection tree of G for W which holds property (ii).

(iii). If $|\mathsf{R}(T')| \leq 1$, then trivially T' holds property (iii). Thus, assume that $|\mathsf{R}(T')| \geq 2$. Additionally, assume that T' holds property (ii). Consequently, $H_R = T[\mathsf{R}(T')]$ is a tree. Note that, H_R contains at least two leaves. Let R^* be the set defined in the following way: $\rho^* \in R^*$ if and only if $\rho^* \in \mathsf{R}(T')$ and there is at least one terminal vertex $w \in W$ such that $\mathrm{dist}_{T'}(w, \rho^*) = \mathrm{dist}_{T'}(w, \mathsf{R}(T'))$, i.e. the path between w and ρ^* in T' does not contain any other router. Note that, every leaf of H_R necessarily belongs to R^*; more specifically, for every leaf ρ^* of H_R, there exists at least two distinct terminal vertices $w^1_{\rho^*}, w^2_{\rho^*} \in W$ such that $\mathrm{dist}_{T'}(w^i_{\rho^*}, \rho^*) = \mathrm{dist}_{T'}(w^i_{\rho^*}, \mathsf{R}(T'))$ for $i \in \{1, 2\}$, otherwise the degree of ρ^* in T' would be less than 3. Let $\langle \rho_1, \ldots, \rho_k \rangle$ be an arbitrary ordering of the vertices in R^* such that ρ_1 and ρ_k are leaves of H_R, where $k = |R^*|$. Then, consider the graph T defined as follows: $V(T) = V(T') \setminus \big(\mathsf{R}(T') \setminus R^*\big)$ and $E(T) = \big(E(T') \setminus E(H_R)\big) \cup \{\rho_i \rho_{i+1} \mid i \in \{1, \ldots, k-1\}\}$. One may verify that T is a strict connection tree of G for W such that $\mathsf{L}(T) = \mathsf{L}(T')$, $\mathsf{R}(T) = R^* \subseteq \mathsf{R}(T')$, and $T[\mathsf{R}(T)]$ is a path. □

Proposition 3. *Suppose that $K \setminus W \neq \varnothing$ and $W \cap S \neq \varnothing$. Given two non-negative integers ℓ and r, with $r \geq 1$, we can in time $n^{\mathcal{O}(r)}$ obtain a strict connection tree T of G for W such that $|\mathsf{L}(T)| \leq \ell$ and $|\mathsf{R}(T)| \leq r$, or conclude that such a tree does not exist.*

Proof. Since S-TCP can be solved in polynomial-time when $r \leq 1$ [29], for simplicity, we assume that G does not admit a strict connection tree T for W such that $|\mathsf{L}(T)| \leq \ell$ and $|\mathsf{R}(T)| \leq 1$. Based on Lemmas 4 and 5, our strategy consists in enumerating all possible subsets $R \subseteq K \setminus W$, with $2 \leq |R| = k \leq r$, and all possible unordered pairs $\{\rho_1, \rho_k\} \subseteq R$ of distinct vertices in order to try to obtain a strict connection T of G for W such that $|\mathsf{L}(T)| \leq \ell$, $\mathsf{R}(T) = R$ and $T[\mathsf{R}(T)]$ is a path with endpoints ρ_1 and ρ_k. Hence, let $R \subseteq K \setminus W$, with $2 \leq |R| = k \leq r$, and ρ_1 and ρ_k be two distinct vertices belonging to R.

Let $W_R = W \cap N_G(R)$ and $\overline{W_R} = W \setminus W_R$. Note that, if $|\overline{W_R}| > \ell$, then G does not admit a strict connection tree for W such that $|\mathsf{L}(T)| \leq \ell$ and $\mathsf{R}(T) = R$. Thus, assume $|\overline{W_R}| \leq \ell$. Let H_1 be the bipartite graph defined as follows: $V(H_1) = X_1 \cup Y_1$ and $E(H_1) = \{xy \in E(G) \mid x \in X_1, y \in Y_1\}$, where $X_1 = \overline{W_R}$ and $Y_1 = (V(G) \setminus (R \cup W)) \cap N_G(X_1)$.

Claim 1. *If $X_1 \neq \varnothing$ and G admits a strict connection tree T for W such that $|\mathsf{L}(T)| \leq \ell$ and $\mathsf{R}(T) = R$, then there exists a matching M_1 in H_1 that saturates all vertices belonging to X_1.*

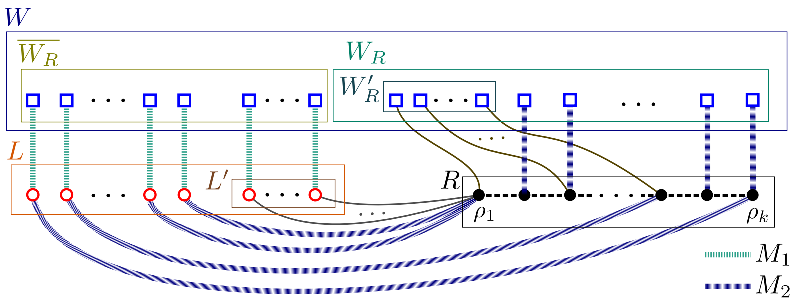


Fig. 10. Strict connection tree of G for W obtained from a matching M_2 in H_2 that saturates all vertices belonging to X_2.

Proof of claim. Assume that T holds properties (i)–(iii) described in Lemma 5. Thus, each linker $u \in \mathsf{L}(T)$ is adjacent to exactly one vertex in $\mathsf{R}(T)$ and exactly one vertex $w \in W$ such that $w \notin N_G(\mathsf{R}(T))$. As a consequence, the set of terminal vertices which are adjacent to a linker of T coincides with X_1. In addition to that, note that, since $|W| \geq 3$, each vertex belonging to $\mathsf{L}(T)$ is adjacent in T to at most one vertex belonging to X_1. Therefore, the set $M_1 = \{uw \in E(T) \mid u \in \mathsf{L}(T),\ w \in W\}$ is a matching in H_1 that saturates all vertices belonging to X_1. □

Based on Claim 1, we assume that $X_1 = \varnothing$, or that there exists a matching M_1 in H_1 that saturates all vertices belonging to X_1. From such a matching M_1 (if any), we let $L = \varnothing$ if $X_1 = \varnothing$, and $L = \{u \in Y_1 \mid uw \in M_1, w \in X_1\}$ otherwise; and we let H_2 be the bipartite graph such that $V(H_2) = X_2 \cup Y_2$ and

$$E(H_2) = \{xy \in E(G) \mid x \in R \setminus \{\rho_1, \rho_k\}, y \in Y_2\}$$
$$\cup \{\rho_i^j y \mid \rho_i y \in E(G), y \in Y_2, i \in \{1, k\}, j \in \{1, 2\}\},$$

where $X_2 = (R \setminus \{\rho_1, \rho_k\}) \cup \{\rho_1^1, \rho_1^2, \rho_k^1, \rho_k^2\}$, $Y_2 = W_R \cup L$ and $\rho_1^1, \rho_1^2, \rho_k^1, \rho_k^2$ are new auxiliary vertices, not belonging to G.

Claim 2. *G admits a strict connection tree T for W such that $|\mathsf{L}(T)| \leq \ell$, $\mathsf{R}(T) = R$ and $T[\mathsf{R}(T)]$ is path with endpoints ρ_1 and ρ_k if and only if there exists a matching M_2 in H_2 that saturates all vertices belonging to X_2.*

Proof of claim. First, suppose that such a tree T exists. Additionally, assume that T holds properties (i)–(iii) described in Lemma 5. As a result, we have $|L| = |X_1| = |\mathsf{L}(T)|$. Let $\phi\colon \mathsf{L}(T) \to L$ be an arbitrary bijection. Since all routers of T have degree at least 3, each endpoint of the path $T[\mathsf{R}(T)]$ — i.e. the vertices ρ_1 and ρ_k — must be adjacent to at least two distinct vertices in $W_R \cup \mathsf{L}(T)$; thus, for $i \in \{1, k\}$, let $v_i^1, v_i^2 \in W_R \cup \mathsf{L}(T)$ be two arbitrary distinct neighbors of ρ_i in T. Furthermore, we have that each internal vertex of $T[\mathsf{R}(T)]$ must be adjacent to at least one vertex in $W_R \cup \mathsf{L}(T)$; thus, for $i \in \{2, \ldots, k-1\}$, let $v_i \in W_R \cup \mathsf{L}(T)$ be an arbitrary neighbor of ρ_i in T. Let $Y_{M_2} = \{v_i^j \mid i \in \{1, k\},\ j \in \{1, 2\}\} \cup \{v_i \mid i \in \{2, \ldots, k-1\}\}$. We remark that $Y_{M_2} \setminus W_R \subseteq \mathsf{L}(T)$. Therefore, one may verify that

$$M_2 = \{\rho_i^j v_i^j \mid \rho_i v_i^j \in E(T),\ v_i^j \in Y_{M_2} \cap W_R,\ i \in \{1, k\},\ j \in \{1, 2\}\}$$
$$\cup \{\rho_i v_i \in E(T) \mid v_i \in Y_{M_2} \cap W_R,\ i \in \{2, \ldots, k-1\}\}$$
$$\cup \{\rho_i^j \phi(v_i^j) \mid \rho_i v_i^j \in E(T),\ v_i^j \in Y_{M_2} \setminus W_R,\ i \in \{1, k\},\ j \in \{1, 2\}\}$$
$$\cup \{\rho_i \phi(v_i) \mid \rho_i v_i \in E(T),\ v_i \in Y_{M_2} \setminus W_R,\ i \in \{2, \ldots, k-1\}\}$$

is a matching in H_2 that saturates all vertices belonging to X_2.

Conversely, suppose that there exists a matching M_2 in H_2 that saturates all vertices belonging to X_2. Let W'_R and L' be the subsets of W_R and L, respectively, composed by the vertices which are not saturated by M_2. Also, let $\varphi\colon W'_R \to R$ be a mapping such that, for each $w \in W'_R$, if $\varphi(w) = \rho$, then $w \in N_G(\rho)$. Consider the graph T defined as follows: $V(T) = W \cup L \cup R$ and

$$E(T) = M_1 \cup (M_2 \setminus \{\rho_i^j v_i \mid v_i \in Y_2,\ i \in \{1, k\},\ j \in \{1, 2\}\})$$
$$\cup \{\rho_i v_i \mid \rho_i^j v_i \in M_2,\ v_i \in Y_2, i \in \{1, k\},\ j \in \{1, 2\}\}$$
$$\cup \{\rho_1 v \mid v \in L'\} \cup \{\varphi(w)w \mid w \in W'_R\} \cup \{\rho_i \rho_{i+1} \mid i \in \{1, \ldots, k-1\}\}.$$

Fig. 10 illustrates the graph T. One may verify that T is a strict connection tree of G for W such that $\mathsf{L}(T) = L$, $\mathsf{R}(T) = R$ and $T[\mathsf{R}(T)]$ is a path with endpoints ρ_1 and ρ_k. □

To conclude the proof of this proposition, we remark that, based on Lemmas 4 and 5, there exists a strict connection tree of G for W with at most ℓ linkers and at most r routers if and only if, for some set $R \subseteq K \setminus W$, with $2 \le |R| = k \le r$, and some unordered pair $\{\rho_1, \rho_k\} \subseteq R$ of distinct vertices, there exists a strict connection tree T of G for W such that $|\mathsf{L}(T)| \le \ell$, $\mathsf{R}(T) = R$ and $T[\mathsf{R}(T)]$ is a path with endpoints ρ_1 and ρ_k.

Furthermore, based on the previous claims, we have that, for a given set $R \subseteq K \setminus W$, with $2 \le |R| = k \le r$, and a given unordered pair $\{\rho_1, \rho_k\} \subseteq R$ of distinct vertices, we can obtain a strict connection tree T of G for W such that $|\mathsf{L}(T)| \le \ell$, $\mathsf{R}(T) = R$ and $T[\mathsf{R}(T)]$ is a path with endpoints ρ_1 and ρ_k, or conclude that such a tree T does not exist, in time polynomial in n. Therefore, since all unordered pair $\{\rho_1, \rho_k\} \subseteq R$ of distinct vertices can be enumerated in time $\mathcal{O}\left(n^2\right)$ and all subsets $R \subseteq K \setminus W$, with $2 \le |R| \le r$, can be enumerated in time $n^{\mathcal{O}(r)}$, the total running time of the algorithm is $n^{\mathcal{O}(r)}$. □

Theorem 7. *S-TCP restricted to split graphs can be solved in time $n^{\mathcal{O}(r)}$ but, assuming FPT $\neq$ W[2], cannot be solved in time $g(r) \cdot n^{h(\ell)}$ and, assuming ETH, cannot be solved in time $g(r) \cdot n^{o(r)}$, for any computable functions g and h.*

Proof. It follows immediately from Facts 1 and 2 and Proposition 3 that S-TCP restricted to split graphs can be solved in time $n^{\mathcal{O}(r)}$.

On the other hand, to see that S-TCP restricted to split graphs does not admit a $g(r) \cdot n^{h(\ell)}$-time algorithm, unless FPT = W[2], note that the proof of Theorem 4 can be easily adapted so that the constructed graph G becomes a split graph. Indeed, it is enough to add to G the edge $u_i u_j$ for each $i, j \in \{1, \ldots, \ell\}$ with $i \neq j$. In this case, $\{K = L \cup K_{\mathcal{F}}, S = W\}$ is a partition of the vertex set V of G into a clique and an independent set, respectively. Therefore, S-TCP remains W[2]-hard even if it is restricted to split graphs and ℓ is bounded by a constant.

Finally, to show that S-TCP restricted to split graphs does not admit a $g(r) \cdot n^{o(r)}$-time algorithm, unless ETH fails, consider the following claim.

Claim 3. *Assuming ETH,* Set cover *cannot be solved in time $g(k) \cdot n^{o(k)}$, for any computable function g.*

Proof of claim. We present a polynomial-time reduction from Dominating set, another classical W[2]-complete problem, which under ETH was proved not to admit a $g(k) \cdot n^{o(k)}$-time algorithm, for any computable function g, where k is the parameter of the problem *cf.* [14]. Let $I = (G', k')$ be an instance of Dominating set. We construct an instance $f(I) = (U, \mathcal{F}, k)$ of Set cover, as follows: $U = V(G')$, $\mathcal{F} = \{N_{G'}[u'] \mid u' \in V(G')\}$ and $k = k'$.

It is easy to see that, if a set $S' \subseteq V(G')$ is a dominating set of G', then $\mathcal{F}' = \{N_{G'}[u'] \mid u' \in V'\} \subseteq \mathcal{F}$ is a vertex cover of G such that $|\mathcal{F}'| = |S'|$.

Conversely, if $\mathcal{F}' \subseteq \mathcal{F}$ is a vertex cover of G, then $S' = \{u' \mid N_{G'}[u'] \in \mathcal{F}'\} \subseteq V(G')$ is a dominating set of G' such that $|S'| = |\mathcal{F}'|$.

Thus, I is a Yes instance of Dominating set if and only if $f(I)$ is a Yes instance of Set cover. Consequently, the existence of a $g(k) \cdot |V(G)|^{o(k)}$-time algorithm for Set cover implies the existence of a $g(k') \cdot |V(G')|^{o(k')}$-time algorithm for Dominating set. □

Therefore, we obtain from the proof of Theorem 4 that the existence of such an algorithm for S-TCP implies the failure of ETH. □

5. The cograph case

A *cograph* is a graph that does not contain any induced path of length 3. Alternatively, cographs can be characterized by the following recursive definition given by Corneil et al. [33]: G is a cograph if and only if $G = K_1$ or there exist two other cographs G_1 and G_2 such that either $G = G_1 \cup G_2$ or $G = G_1 \wedge G_2$, where K_1 denotes the trivial graph with a single vertex, and $G_1 \cup G_2$ and $G_1 \wedge G_2$ respectively denote the *disjoint union* and the *join* of G_1 and G_2, i.e. $V(G_1 \cup G_2) = V(G_1 \wedge G_2) = V(G_1) \cup V(G_2)$, $E(G_1 \cup G_2) = E(G_1) \cup E(G_2)$ and $E(G_1 \wedge G_2) = E(G_1) \cup E(G_2) \cup \{uv \mid u \in V(G_1), v \in V(G_2)\}$.

In complex networks, *cograph communities* are defined as the connected components of a network such that the underlying graph is a cograph. According to [34], as a whole community, cograph communities reveal more intensive social roles or biological functions than those obtained by general communities. Thus, motivated by the relevance of cographs in complex networks, we analyze in this section the complexity of S-TCP restricted to cographs.

More specifically, we prove that S-TCP on cographs is polynomial-time solvable. Although this can be an expected result (for instance, it is known that Steiner tree on cographs is polynomial-time solvable [9]), since cographs have strong structural properties that are useful for the development of polynomial-time algorithms, our proof is not trivial whatsoever, consisting in providing a sophisticated dynamic programming algorithm for the problem.

Since S-TCP can be easily solved in linear-time when $r < 1$ or $|W| < 3$, we assume that $r \ge 1$ and $|W| \ge 3$. Next, we analyze all the other possible cases, and then we finally summarize in Theorem 8 the recurrence relation of our algorithm.

Fact 3. Let $G = (V, E)$ be a cograph such that $G = G_1 \cup G_2$. Then, G admits a strict connection tree for W with at most $\ell \geq 0$ linkers and at most $r \geq 1$ routers if and only if $V(G_j) \cap W = \varnothing$ and G_i admits a strict connection tree for W with at most ℓ linkers and at most r routers, where $V(G_i) \cap W \neq \varnothing$, $i, j \in \{1, 2\}$ and $i + j = 3$.

Fact 4. Let $G = (V, E)$ be a cograph such that $G = G_1 \wedge G_2$, $V(G_1) \subseteq W$ and $|V(G_2) \cap W| \leq 1$. Then, there exists a strict connection tree of G for W if and only if $W = V(G_1)$, or $V(G_2) \cap W = \{w\}$ and $N_{G_2}(w) \neq \varnothing$. In particular, if such a tree exists, then G admits a strict connection tree T for W such that $|\mathsf{L}(T)| = 0$ and $|\mathsf{R}(T)| = 1$.

Lemma 6. *Let $G = (V, E)$ be a cograph such that $G = G_1 \wedge G_2$, $V(G_1) \subset W$ and $|V(G_2) \cap W| = 2$. Let w'_1 and w'_2 be the two vertices belonging to $V(G_2) \cap W$. Then, G admits a strict connection tree for W with at most $\ell \geq 0$ linkers and at most $r \geq 1$ routers if and only if the distance in $G'_2 = G_2 - w'_1 w'_2$ between w'_1 and w'_2 is at most $\ell + \min\{r, n_1\} + 1$, where $n_1 = |V(G_1)|$.*

Proof. First, suppose that G admits a strict connection tree T for W such that $|\mathsf{L}(T)| \leq \ell$ and $|\mathsf{R}(T)| \leq r$. Let P be the path in T between w'_1 and w'_2. Observe that, the length of P is at most $|\mathsf{L}(T)| + |\mathsf{R}(T)| + 1$, otherwise T would have more than $|\mathsf{L}(T)|$ linkers or more than $|\mathsf{R}(T)|$ routers. Since by hypothesis $V(G_1) \subset W$ and $|V(G_2) \cap W| = 2$, we have that $n_1 = |V(G_1)| = |W| - 2$. Furthermore, we know that $|\mathsf{R}(T)| \leq |W| - 2$ (see Proposition 2), which implies $|\mathsf{R}(T)| \leq n_1$. Thus, $|\mathsf{R}(T)| \leq \min\{r, n_1\}$, and so the length of P is at most $|\mathsf{L}(T)| + \min\{r, n_1\} + 1 \leq \ell + \min\{r, n_1\} + 1$. Finally, since $V(G_1) \subset W$ and $|W| \geq 3$, P is contained in G'_2, otherwise T would have some terminal with degree greater than 1. Therefore, the distance in G'_2 between w'_1 and w'_2 is at most $\ell + \min\{r, n_1\} + 1$.

Conversely, suppose that the distance in G'_2 between w'_1 and w'_2 is at most $\ell + \min\{r, n_1\} + 1$. Let $P = \langle w'_1, u'_1, u'_2, \ldots, u'_z, w'_2 \rangle$ be a shortest path in G'_2 between w'_1 and w'_2, and let $\alpha = \langle w_1, w_2, \ldots, w_{n_1} \rangle$ be an arbitrary ordering of the vertices in $V(G_1)$. We define from P and α the subgraph T of G as follows: $V(T) = V(P) \cup V(G_1)$ and $E(T) = E(P) \cup \{u'_i w_i \mid i \in \{1, \ldots, \min\{r, n_1, z\}\}\} \cup \{u'_1 w_i \mid i \in \{\min\{r, n_1, z\} + 1, \ldots, n_1\}\}$. Observe that, T is a strict connection tree of G for W such that $\mathsf{L}(T) = \{u'_i \mid i \in \{\min\{r, n_1\} + 1, \ldots, z\}\}$ and $\mathsf{R}(T) = \{u'_i \mid i \in \{1, \ldots, \min\{r, n_1, z\}\}\}$. Since by hypothesis the length of P is at most $\ell + \min\{r, n_1\} + 1$, $z \leq \ell + \min\{r, n_1\}$, and consequently $|\mathsf{L}(T)| \leq \ell$. Finally, note that $|\mathsf{R}(T)| = \min\{r, n_1, z\} \leq r$. Therefore, G admits a strict connection tree for W with at most ℓ linkers and at most r routers. □

Lemma 7. *Let $G = (V, E)$ be a cograph such that $G = G_1 \wedge G_2$, $V(G_1) \subset W$ and $|V(G_2) \cap W| \geq 3$. Then, G admits a strict connection tree for W with at most $\ell \geq 0$ linkers and at most $r \geq 1$ routers if and only if G_2 admits a strict connection tree for $V(G_2) \cap W$ with at most $\ell + \lambda$ linkers and at most $r - \lambda$ routers, for some $\lambda \in \{0, 1, \ldots, \min\{r - 1, n_1\}\}$, where $n_1 = |V(G_1)|$.*

Proof. First, suppose that G admits a strict connection tree T for W such that $|\mathsf{L}(T)| \leq \ell$ and $|\mathsf{R}(T)| \leq r$. Since $|V(G_2) \cap W| \geq 3$ and all vertices in $V(G_1)$ are terminal, there exists at least one router $\rho \in \mathsf{R}(T)$ such that $\rho \in V(G_2)$.

Suppose that there is a vertex $v \in V(G_2) \setminus W$ with $N_T(v) \cap V(G_1) \neq \varnothing$, such that $|N_{G_2}(v) \cap V(T)| < 2$. Then, let H be the graph obtained from T by removing the vertex v and by adding, for each $w \in N_T(v) \cap V(G_1)$, the edge ρw. We remark that H possibly contains non-terminal vertices that became leaves. Then, let T_H be the graph obtained from H by successively removing such non-terminal vertices. One can easily verify that T_H is a strict connection tree of G for W such that $\mathsf{L}(T_H) \subseteq \mathsf{L}(T)$ and $\mathsf{R}(T_H) \subseteq \mathsf{R}(T)$. Thus, hereinafter, assume without loss of generality that $|N_{G_2}(v) \cap V(T)| \geq 2$ for every vertex $v \in V(G_2) \setminus W$ with $N_T(v) \cap V(G_1) \neq \varnothing$.

Let $T' = T - V(G_1)$. Since by hypothesis $V(G_1) \subset W$, the vertices belonging to $V(G_1)$ are leaves of T. Thus, T' is a connected graph. More specifically, it follows from the assumption described in the previous paragraph that T' is a strict connection tree of G_2 for $V(G_2) \cap W$. Let

$$R' = \{v \in V(G_2) \setminus W \mid N_T(v) \cap V(G_1) \neq \varnothing, d_{T'}(v) < 3\}.$$

Note that, the vertices belonging to R' are routers of T and linkers of T'. Thus, $|\mathsf{L}(T')| = |\mathsf{L}(T)| + |R'|$ and $|\mathsf{R}(T')| = |\mathsf{R}(T)| - |R'|$. Moreover, since the vertices belonging to $V(G_1)$ are leaves of T and $|W| \geq 3$, each vertex $w \in V(G_1)$ is adjacent in T to exactly one vertex $v \in V(G_2) \setminus W$. Consequently, $|R'| \leq n_1$. On the other hand, since by hypothesis $|V(G_2) \cap W| \geq 3$, T' has at least one router. Thus, $|R'| \leq r - 1$, which implies $|R'| \leq \min\{r - 1, n_1\}$. Therefore, we can define $\lambda = |R'|$, and so we have that G_2 admits a strict connection tree for $V(G_2) \cap W$ with at most $\ell + \lambda$ linkers and at most $r - \lambda$ routers.

For the converse, suppose now that G_2 admits a strict connection tree T' for $V(G_2) \cap W$ such that $|\mathsf{L}(T')| \leq \ell + \lambda$ and $|\mathsf{R}(T')| \leq r - \lambda$, for some $\lambda \in \{0, 1, \ldots, \min\{r - 1, n_1\}\}$. Since $|V(G_2) \cap W| \geq 3$, $\mathsf{R}(T')$ is non-empty. Then, let $\rho \in \mathsf{R}(T')$, arbitrarily chosen. Let R' be a subset of $\mathsf{L}(T')$ such that $|R'| = z$, where $z = \min\{|\mathsf{L}(T')|, \lambda\}$; and let $\langle u'_1, u'_2, \ldots, u'_z \rangle$ be an arbitrary ordering of the vertices belonging to R'. Also, let $\langle w_1, w_2, \ldots, w_{n_1} \rangle$ be an arbitrary ordering of the vertices belonging to $V(G_1)$. Then, the graph T, defined as follows: $V(T) = V(T') \cup V(G_1)$ and $E(T) = E(T') \cup \{u'_i w_i \mid i \in \{1, \ldots, z\}\} \cup \{\rho w_i \mid i \in \{z+1, \ldots, n_1\}\}$, is a strict connection tree for W such that $\mathsf{L}(T) = \mathsf{L}(T') \setminus \{u'_i \mid i \in \{1, \ldots, z\}\}$ and $\mathsf{R}(T) = \mathsf{R}(T') \cup \{u'_i \mid i \in \{1, \ldots, z\}\}$. Therefore, G admits a strict connection tree T for W such that $|\mathsf{L}(T)| = |\mathsf{L}(T')| - z = |\mathsf{L}(T')| - \min\{|\mathsf{L}(T')|, \lambda\} \leq \ell$ and $|\mathsf{R}(T)| = |\mathsf{R}(T')| + z = |\mathsf{R}(T')| + \min\{|\mathsf{L}(T')|, \lambda\} \leq r$. □

Proposition 4. *Let $G = (V, E)$ be a cograph such that $G = G_1 \wedge G_2$, $V(G_1) \not\subseteq W$, $V(G_1) \cap W \neq \varnothing$, $V(G_2) \not\subseteq W$ and $V(G_2) \cap W \neq \varnothing$. Given $\ell \geq 0$ and $r \geq 1$, we can in polynomial-time obtain a strict connection tree T for W such that $|\mathsf{L}(T)| \leq \ell$ and $|\mathsf{R}(T)| \leq r$, or conclude that such a tree does not exist.*

Proof. First, consider $r = 1$. By a Turing reduction to the problem of finding vertex-disjoint paths with minimum total cost, we can in polynomial-time obtain a strict connection tree T of G for W such that $|\mathsf{L}(T)| \leq \ell$ and $|\mathsf{R}(T)| = 1$, or conclude that such a tree does not exist [29].

Thus, hereinafter, assume that $r \geq 2$ and that G does not admit a strict connection tree T for W such that $|\mathsf{L}(T)| \leq \ell$ and $|\mathsf{R}(T)| \leq 1$. Consequently, assume also that $|W| \geq 4$. There are two cases to be analyzed.

Case 1: $|V(G_1) \cap W| = 1$.

In this case, we can additionally assume that $\ell = 0$ and $N_{G_1}(w) = \varnothing$, where w is the only vertex in $V(G_1) \cap W$. Indeed, suppose that $\ell \geq 1$. Then, for $v \in V(G_1) \setminus W$ and $v' \in V(G_2) \setminus W$, we have that the graph T, where $V(T) = W \cup \{v, v'\}$ and $E(T) = \{vv', v'w\} \cup \{vw' \mid w' \in V(G_2) \cap W\}$, is a strict connection tree of G for W such that $\mathsf{L}(T) = \{v'\}$ and $\mathsf{R}(T) = \{v\}$. Now, suppose that $N_{G_1}(w) \neq \varnothing$, and let $v \in N_{G_1}(w)$. Note that, $N_G(v) \supseteq W$. Consequently, G admits a strict connection tree for W without linkers and with at most one router. For example, the graph T, where $V(T) = W \cup \{v\}$ and $E(T) = \{vw \mid w \in W\}$, is a strict connection tree of G for W such that $\mathsf{L}(T) = \varnothing$ and $\mathsf{R}(T) = \{v\}$. Thus, we assume hereinafter that $\ell = 0$ and $N_{G_1}(w) = \varnothing$.

If $N_{G_2}(w') \setminus W \neq \varnothing$ for some terminal $w' \in V(G_2) \cap W$, then G admits a strict connection tree T for W such that $|\mathsf{L}(T)| = 0$ and $|\mathsf{R}(T)| = 2$. Indeed, for $v' \in N_{G_2}(w') \setminus W$ and $v \in V(G_1) \setminus W$, we have that the graph T, where $V(T) = W \cup \{v, v'\}$ and $E(T) = \{vv', v'w, v'w'\} \cup \{vw' \mid w' \in (V(G_2) \cap W) \setminus \{w'\}\}$, is a strict connection tree for W such that $\mathsf{L}(T) = \varnothing$ and $\mathsf{R}(T) = \{v, v'\}$. Thus, assume that, for every vertex $w' \in V(G_2) \cap W$, $N_{G_2}(w') \setminus W = \varnothing$.

Note that, if $r = 2$ or $|V(G_2) \cap W| = 3$, then G does not admit a strict connection tree for W without linkers and with at most r routers. Suppose for purposes of contradiction that it is not true, and let T be such a tree. Since $N_{G_1}(w) = \varnothing$, the only neighbor of w in T is a non-terminal vertex $v' \in V(G_2) \setminus W$. However, if $r = 2$, then $|\mathsf{R}(T)| = 2$; and if $|V(G_2) \cap W| = 3$, then $|\mathsf{R}(T)| \leq |W| - 2 = 2$ (see Proposition 2). Consequently, T has at most two non-terminal vertices, being v' one of those vertices. Thus, v' is adjacent to at most one non-terminal vertex in T, and so its degree in T is at most 2, since $N_{G_2}(v') \cap W = \varnothing$. Therefore, G does not admit a strict connection tree for W without linkers and with at most r routers. Similarly, it is easy to see that G does not admit such a tree if $|V(G_1) \setminus W| = 1$. Hence, assume that $r \geq 3$, $|V(G_2) \cap W| \geq 4$ and $|V(G_1) \setminus W| \geq 2$.

Let $v_1, v_2 \in V(G_1) \setminus W$ and $v' \in V(G_2) \setminus W$. Also, let $\langle w'_1, w'_2, \ldots, w'_{n_2} \rangle$ be an arbitrary ordering of the vertices belonging to $V(G_2) \cap W$, where $n_2 = |V(G_2) \cap W|$. Then, the graph T, where $V(T) = W \cup \{v_1, v_2, v'\}$ and $E(T) = \{v_1v', v_2v', v_1w'_1, v_2w'_2, v'w'\} \cup \{v_2w'_i \mid i \in \{3, \ldots, n_2\}\}$, is a strict connection tree for W such that $\mathsf{L}(T) = \varnothing$ and $\mathsf{R}(T) = \{v_1, v_2, v'\}$. Therefore, G admits a strict connection tree T for W such that $|\mathsf{L}(T)| = 0 = \ell$ and $|\mathsf{R}(T)| = 3 \leq r$.

Case 2: $|V(G_1) \cap W| \geq 2$ and $|V(G_2) \cap W| \geq 2$.

Let $v \in V(G_1) \setminus W$ and $v' \in V(G_2) \setminus W$. Then, the graph T, where $V(T) = W \cup \{v, v'\}$ and $E(T) = \{vv'\} \cup \{vw' \mid w' \in V(G_2) \cap W\} \cup \{v'w \mid w \in V(G_1) \cap W\}$, is a strict connection tree for W such that $\mathsf{L}(T) = \varnothing$ and $\mathsf{R}(T) = \{v, v'\}$. Therefore, in this case, G always admits a strict connection tree T for W such that $|\mathsf{L}(T)| = 0$ and $|\mathsf{R}(T)| = 2$.

To conclude the proof of this proposition, note that all operations described above can be computed in time polynomial in n. □

An important property of cographs is that every cograph G can be uniquely represented by a rooted tree $\mathcal{T}_G$, called *cotree*, such that (1) the leaves of $\mathcal{T}_G$ correspond to the vertices of G; and, (2) each internal node u of $\mathcal{T}_G$ corresponds to either the disjoint union or the join of the cographs induced by the leaves of the subtrees of T_G rooted at each child of u [33]. Throughout this section, we assume without loss of generality that a cotree is a binary tree. Thus, the cotree $\mathcal{T}_G$ of a cograph G can be viewed as the tree corresponding to the unique decomposition of G as the trivial graph K_1, or either the join or the union of two other cographs. Another important property is the fact that the recognition of a given graph G as a cograph, as well as obtaining its respective cotree (if any), can be performed in time linear in n and m [35].

Theorem 8. *S-TCP is polynomial-time solvable if it is restricted to cographs.*

Proof. Let $I = (G, W, \ell, r)$ be an instance of S-TCP, where $G = (V, E)$ is a cograph, and let $\mathcal{T}_G$ be the cotree of G. We define a dynamic programming table M such that: for each node u' of $\mathcal{T}_G$ and for each pair of non-negative integers ℓ' and r', with $\ell' + r' < |V(G_{u'}) \setminus W|$, there is an entry $M[G_{u'}, \ell', r']$ which is set *true* if and only if $V(G_{u'}) \cap W \neq \varnothing$ and $G_{u'}$ admits a strict connection tree for $V(G_{u'}) \cap W$ with at most ℓ' linkers and at most r' routes, where $G_{u'}$ denotes the cograph associated with the subtree of $\mathcal{T}_G$ rooted at u'.

Facts 3 and 4, Lemmas 6 and 7 and Proposition 4 are used to fill M and, thus, decide whether G admits a strict connection tree for W with at most ℓ linkers and at most r routers. More specifically, $M[G,\ell,r]$ is defined on the basis of the following rules:

$$M[G,\ell,r] := \begin{cases} \left[\begin{array}{l} \textbf{case 1. } |V \cap W| \le 2 \text{ or } r = 0: \\ \quad \textit{true} \;\; \text{if } |V \cap W| = 1, \\ \quad \textit{true} \;\; \text{if } V \cap W = \{w_1, w_2\} \text{ and } \text{dist}_G(w_1,w_2) \le \ell + 1, \\ \quad \textit{false} \;\; \text{otherwise;} \end{array}\right. \\ \left[\begin{array}{l} \textbf{case 2. } G = G_1 \cup G_2: \\ \quad M[G_1,\ell,r] \;\; \text{if } V(G_2) \cap W = \varnothing, \\ \quad \textit{false} \;\; \text{otherwise;} \end{array}\right. \\ \left[\begin{array}{l} \textbf{case 3. } G = G_1 \wedge G_2 \text{ and } V(G_1) = W: \\ \quad \textit{true}; \end{array}\right. \\ \left[\begin{array}{l} \textbf{case 4. } G = G_1 \wedge G_2,\, V(G_1) \subset W \text{ and } V(G_2) \cap W = \{w\}: \\ \quad \textit{true} \;\; \text{if } N_{G_2}(w) \ne \varnothing, \\ \quad \textit{false} \;\; \text{otherwise;} \end{array}\right. \\ \left[\begin{array}{l} \textbf{case 5. } G = G_1 \wedge G_2,\, V(G_1) \subset W \text{ and } V(G_2) \cap W = \{w'_1, w'_2\}: \\ \quad \textit{true} \;\; \text{if } \text{dist}_{G'_2}(w'_1,w'_2) < \ell + \min\{r,n_1\} + 1, \\ \quad \textit{false} \;\; \text{otherwise,} \\ \text{where } G'_2 = G_2 - w'_1 w'_2 \text{ and } n_1 = |V(G_1)|; \end{array}\right. \\ \left[\begin{array}{l} \textbf{case 6. } G = G_1 \wedge G_2,\, V(G_1) \subset W \text{ and } |V(G_2) \cap W| \ge 3: \\ \quad \displaystyle\bigvee_{\lambda=0}^{\min\{r-1,n_1\}} M[G_2, \ell+\lambda, r-\lambda], \\ \text{where } n_1 = |V(G_1)|; \end{array}\right. \\ \left[\begin{array}{l} \textbf{case 7. } G = G_1 \wedge G_2,\, V(G_i) \not\subset W,\, V(G_i) \cap W \ne \varnothing, \forall i \in \{1,2\}: \\ \quad \textsc{Alg}(G,W,\ell,r), \\ \text{where } \textsc{Alg} \text{ denotes the algorithm described in Proposition 4.} \end{array}\right. \end{cases}$$

Note that, the size of M is $\mathcal{O}(n^3)$. Furthermore, one may verify that each entry of M can be computed in time polynomial in n, in a bottom-up manner according to the post-order traversal of $\mathcal{T}_G$. Regarding the correctness of the dynamic programming algorithm, case 1 can be easily verified *cf.* [29]; case 2 derives from Fact 3; case 3 and 4 derive from Fact 4; case 5 derives from Lemma 6; case 6 derives from Lemma 7; and case 7 clearly derives from Proposition 4. □

6. Conclusions and open problems

We have presented several complexity results for S-TCP (see Table 1). Nonetheless, the complexity of the problem remains unknown on some particular cases. Thus, to conclude this work, three open questions are highlighted.

(i) Is S-TCP parameterized by r in XP?
(ii) Is S-TCP parameterized by ℓ in FPT when $\Delta = 3$?
(iii) Is S-TCP parameterized by $|W|$ in FPT? And if r and Δ are parameters?

Although Steiner tree parameterized by $|W|$ is in FPT [10], it is not clear that S-TCP parameterized by $|W|$, or even parameterized by r and Δ, is also in FPT.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M.C. Dourado, R.A. Oliveira, F. Protti, U.S. Souza, Design of connection networks with bounded number of non-terminal vertices, in: Proceedings of V Latin-American Workshop on Cliques in Graphs, in: Matemática Contemporânea, vol. 42, SBM, Buenos Aires, 2014, pp. 39–47.
[2] M.C. Dourado, R.A. Oliveira, F. Protti, U.S. Souza, Conexão de terminais com número restrito de roteadores e elos, in: Proccedings of XLVI Simpósio Brasileiro de Pesquisa Operacional, 2014, pp. 2965–2976.
[3] G. Lin, G. Xue, On the terminal Steiner tree problem, Inf. Process. Lett. 84 (2) (2002) 103–107.
[4] R.M. Karp, Reducibility Among Combinatorial Problems, Springer US, Boston, MA, 1972, pp. 85–103.
[5] M.R. Garey, D.S. Johnson, The rectilinear Steiner tree problem is NP-complete, SIAM J. Appl. Math. 32 (4) (1977) 826–834.
[6] K. White, M. Farber, W. Pulleyblank, Steiner trees, connected domination and strongly chordal graphs, Networks 15 (1) (1985) 109–124.
[7] H. Müller, A. Brandstädt, The NP-completeness of Steiner tree and dominating set for chordal bipartite graphs, Theor. Comput. Sci. 53 (2–3) (1987) 257–265.
[8] A. D'Atri, M. Moscarini, Distance-hereditary graphs, Steiner trees, and connected domination, SIAM J. Comput. 17 (3) (1988) 521–538.
[9] C.J. Colbourn, L.K. Stewart, Permutation graphs: connected domination and Steiner trees, Discrete Math. 86 (1–3) (1990) 179–189.
[10] S.E. Dreyfus, R.A. Wagner, The Steiner problem in graphs, Networks 1 (3) (1971) 195–207.
[11] A. Björklund, T. Husfeldt, P. Kaski, M. Koivisto, Fourier meets Möbius: fast subset convolution, in: Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing, STOC '07, Association for Computing Machinery, New York, NY, USA, 2007, pp. 67–74.
[12] M. Cygan, M. Pilipczuk, M. Pilipczuk, J.O. Wojtaszczyk, Kernelization hardness of connectivity problems in d-degenerate graphs, Discrete Appl. Math. 160 (15) (2012) 2131–2141.
[13] J. Nederlof, Fast polynomial-space algorithms using inclusion-exclusion, Algorithmica 65 (4) (2013) 868–884.
[14] M. Cygan, F.V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, S. Saurabh, Parameterized Algorithms, Springer, 2015.
[15] F.K. Hwang, D.S. Richards, P. Winter, The Steiner Tree Problem, Annals of Discrete Mathematics, vol. 53, Elsevier, 1992.
[16] F. Hwang, A linear time algorithm for full Steiner trees, Oper. Res. Lett. 4 (5) (1986) 235–237.
[17] C.L. Lu, C.Y. Tang, R.C.-T. Lee, The full Steiner tree problem, Theor. Comput. Sci. 306 (1–3) (2003) 55–67.
[18] A. Biniaz, A. Maheshwari, M. Smid, On the hardness of full Steiner tree problems, J. Discret. Algorithms 34 (2015) 118–127.
[19] H. Fernau, T. Fluschnik, D. Hermelin, A. Krebs, H. Molter, R. Niedermeier, Diminishable parameterized problems and strict polynomial kernelization, in: Sailing Routes in the World of Computation. Proceedings of 4th Conference on Computability in Europe, in: Lecture Notes in Computer Science, vol. 10936, Springer, 2018, pp. 161–171.
[20] R. Khandekar, G. Kortsarz, Z. Nutov, On some network design problems with degree constraints, J. Comput. Syst. Sci. 79 (5) (2013) 725–736.
[21] B. Fuchs, A note on the terminal Steiner tree problem, Inf. Process. Lett. 87 (4) (2003) 219–220.
[22] D.E. Drake, S. Hougardy, On approximation algorithms for the terminal Steiner tree problem, Inf. Process. Lett. 89 (1) (2004) 15–18.
[23] F.V. Martinez, J.C. de Pina, J. Soares, Algorithms for terminal Steiner trees, Theor. Comput. Sci. 389 (1) (2007) 133–142.
[24] Y.H. Chen, An improved approximation algorithm for the terminal Steiner tree problem, in: Proceedings of International Conference on Computational Science and Its Applications, Springer, Berlin, Heidelberg, 2011, pp. 141–151.
[25] A. Biniaz, A. Maheshwari, M. Smid, On full Steiner trees in unit disk graphs, Comput. Geom. 48 (6) (2015) 453–458.
[26] L. Gargano, M. Hammar, P. Hell, L. Stacho, U. Vaccaro, Spanning spiders and light-splitting switches, Discrete Math. 285 (1) (2004) 83–95.
[27] D. Watel, M.-A. Weisser, C. Bentz, D. Barth, Steiner problems with limited number of branching nodes, in: Proceedings of 20th International Colloquium on Structural Information and Communication Complexity, in: Lecture Notes in Computer Science, vol. 8179, Springer-Verlag Inc., New York, 2013, pp. 310–321.
[28] D. Watel, M.-A. Weisser, C. Bentz, D. Barth, Directed Steiner trees with diffusion costs, J. Comb. Optim. 32 (4) (2016) 1089–1106.
[29] A.A. Melo, C.M.H. Figueiredo, U.S. Souza, Connecting terminals using at most one router, in: Proceedings of VII Latin-American Workshop on Cliques in Graphs, in: Matemática Contemporânea, vol. 45, SBM, 2017, pp. 49–57.
[30] H. Müller, Hamiltonian circuits in chordal bipartite graphs, Discrete Math. 156 (1–3) (1996) 291–298.
[31] H.L. Bodlaender, B.M.P. Jansen, S. Kratsch, Cross-composition: a new technique for kernelization lower bounds, CoRR, arXiv:1011.4224 [abs].
[32] H.L. Bodlaender, B.M.P. Jansen, S. Kratsch, Kernelization lower bounds by cross-composition, SIAM J. Discrete Math. 28 (1) (2014) 277–305.
[33] D.G. Corneil, H. Lerchs, S.L. Burlingham, Complement reducible graphs, Discrete Appl. Math. 3 (3) (1981) 163–174.
[34] S. Jia, L. Gao, Y. Gao, J. Nastos, Y. Wang, X. Zhang, H. Wang, Defining and identifying cograph communities in complex networks, New J. Phys. 17 (1) (2015) 013044.
[35] D.G. Corneil, Y. Perl, L.K. Stewart, A linear recognition algorithm for cographs, SIAM J. Comput. 14 (4) (1985) 926–934.