Matemática Contemporânea, Vol. 45, 49–57
http://doi.org/10.21711/231766362017/rmc456

Connecting terminals using at most one router

Alexsander A. Melo **Celina M. H. de Figueiredo**
Uéverton S. Souza

Abstract

A *connection tree* T of a graph G for a *terminal* set $W \subseteq V(G)$ is a tree subgraph of G, such that $W \subseteq V(T)$ and every leaf of T belongs to W. A *non-terminal* vertex $v \in V(T) \backslash W$ is called *linker* if its degree in T is exactly 2, and it is called *router* if its degree in T is at least 3. Given a graph G, a terminal set $W \subseteq V(G)$ and two non-negative integers ℓ and r, the TERMINAL CONNECTION PROBLEM (TCP) asks whether G admits a connection tree for W with at most ℓ linkers and at most r routers. The STRICT TCP (S-TCP) further requires that every terminal is a leaf of the connection tree. In the present extended abstract, we prove that S-TCP is polynomial-time solvable if $r \in \{0, 1\}$, contrasting with the complexity of TCP, which is known to be $\mathcal{NP}$-complete for all $r \geq 0$.

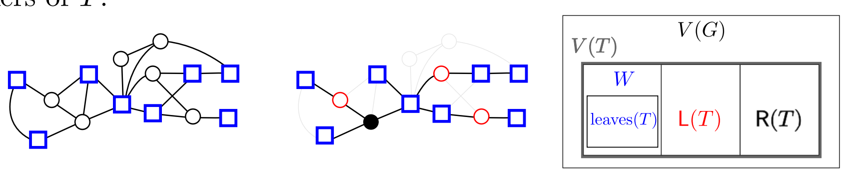
1 Introduction

Problems concerning to network design are usually challenging combinatorial questions of great practical and theoretical interest. In the

2000 AMS Subject Classification: 68R10, 68R15.
Key Words and Phrases: Terminal connection, Strict connection tree, Bounded number of routers, Steiner tree, Disjoint paths.
Partially supported by CNPq and FAPERJ.

present extended abstract, we analyse the computational complexity of a recently proposed network design problem, called the Terminal connection problem (TCP). More specifically, we consider a particular case of its strict variant, S-TCP, where the maximum number of *routers* is bounded by a constant.

Let $G = (V(G), E(G))$ be a graph, and let W be a non-empty subset of $V(G)$. We say that a subgraph T of G is a *connection tree* of G for W if the following conditions are satisfied: T is a tree, $W \subseteq V(T)$ and every leaf of T belongs to W. In a connection tree T for W, the vertices in W are called *terminals* and the vertices in $V(T)\backslash W$ are called *non-terminals*, which are classified into two types according to their respective degrees in T, namely: the vertices in $V(T)\backslash W$ with degree exactly 2 in T are called *linkers* and the vertices in $V(T)\backslash W$ with degree at least 3 in T are called *routers*. Therefore, there exists a partition $V(T) = W \cup \mathsf{L}(T) \cup \mathsf{R}(T)$ of the vertex set of T into terminals, linkers and routers, where $\mathsf{L}(T)$ denotes the linker set of T and $\mathsf{R}(T)$ denotes the router set of T. Figure 1 exemplifies a connection tree T of a graph G for a given terminal set $W \subseteq V(G)$, as well as illustrates the partition of $V(T)$ into terminals, linkers and routers, where the squared vertices (in blue) define the terminal set W, the black filled vertex is a router of T and the remaining vertices (in red) are the linkers of T.



(a) (b) (c)

Figure 1: (a) A graph G and a terminal set $W \subseteq V(G)$. (b) A connection tree T of G for W. (c) Partition of $V(T)$ into terminals, linkers and routers.

Next, we formally define the Terminal connection problem, which was proposed by Dourado et al. [1] motivated by applications in information security and network routing (additionally, there also are applications related to the problem in the implementation of protocols for the *Internet of Things*).

TERMINAL CONNECTION PROBLEM (TCP)	
Instance:	A connected graph G, a non-empty subset $W \subseteq V(G)$ and two non-negative integers ℓ and r.
Question:	Does G admit a connection tree T for W such that $\lvert \mathsf{L}(T)\rvert \leq \ell$ and $\lvert \mathsf{R}(T)\rvert \leq r$?

In [1], Dourado et al. showed that TCP is closely related to the classical network design problems MINIMUM SPANNING TREE and STEINER TREE. Furthermore, they formulated three variants of TCP by bounding the parameters ℓ or r by constants, as follows: TCP(ℓ) refers to the variant of TCP where ℓ is bounded by a constant; TCP(r) refers to the variant of TCP where r is bounded by a constant; and TCP(ℓ, r) refers to the variant of TCP where ℓ and r are both bounded by constants.

The authors [1] proved that TCP(ℓ) is $\mathcal{NP}$-complete, for all fixed $\ell \geq 0$, by a polynomial-time reduction from 3-SAT. Moreover, they proved that TCP(r) is $\mathcal{NP}$-complete, for all fixed $r \geq 0$, by a polynomial-time reduction from HAMILTONIAN PATH. Nevertheless, they showed that TCP(ℓ, r) is polynomial-time solvable, for all fixed $\ell \geq 0$ and $r \geq 0$. Their algorithm consists in an exhaustive search for a connection tree with at most ℓ linkers and at most r routers, which considers every possible linker set $L \subseteq V(G) \setminus W$ and every possible router set $R \subseteq V(G) \setminus W$ such that $|L| \leq \ell$, $|R| \leq r$ and $L \cap R = \varnothing$.

The strict version of TCP. Note that, although every leaf of a connection tree for W belongs to W (by definition), possibly there are terminals in W which are not leaves of T. In such a case, these terminals may work as either a linker or a router, which can be undesirable in some real applications [2, 4]. Given that, Dourado et al. [2] proposed in 2014 the strict variant of TCP, called the STRICT TERMINAL CONNECTION PROBLEM, where it is further required that the connection tree is *strict*, *i.e.* the leaf set of the connection tree must be equal to the terminal set W.

Strict terminal connection problem (S-TCP)	
Instance:	A connected graph G, a non-empty subset $W \subseteq V(G)$ and two non-negative integers ℓ and r.
Question:	Does G admit a strict connection tree for W such that $\lvert \mathsf{L}(T)\rvert \leq \ell$ and $\lvert \mathsf{R}(T)\rvert \leq r$?

Similarly to TCP(ℓ) and to TCP(ℓ, r), respectively, Dourado et al. [2] proved that S-TCP(ℓ) is $\mathcal{NP}$-complete, for all fixed $\ell \geq 0$, and proved that S-TCP(ℓ, r) is polynomial-time solvable, for all fixed $\ell \geq 0$ and $r \geq 0$. Furthermore, they presented a polynomial-time reduction from TCP(ℓ) to S-TCP(ℓ). However, the computational complexity of S-TCP(r) was left open, for all fixed $r \geq 0$. The main goal of this extended abstract is to prove that, for $r \in \{0, 1\}$, S-TCP(r) is polynomial-time solvable, contrasting with the complexity of TCP(r), which is $\mathcal{NP}$-complete, for all fixed $r \geq 0$.

2 Main results

In this section, we present the contributions of this work. Firstly, note that if we consider a terminal set W with just one element, then the answer for the question of S-TCP is always No, regardless of the input graph and the parameters ℓ and r. Thus, we may suppose without loss of generality that there are at least two terminals in W. The next lemma provides a necessary and sufficient condition for an instance $I = (G, W, \ell, r)$ with $|W| = 2$ of S-TCP (and of TCP) to be a Yes instance.

Lema 2.1. *Let $I = (G, W, \ell, r)$ be an instance of S-TCP such that $|W| = 2$. If $W = \{w_1, w_2\}$, then I is a* Yes *instance if and only if the distance in G between w_1 and w_2 is at most $\ell + 1$.*

Proof. Suppose that I is a Yes instance of S-TCP. Hence, G admits a strict connection tree T for W such that $\lvert \mathsf{L}(T)\rvert \leq \ell$ and $\lvert \mathsf{R}(T)\rvert \leq r$. Since

T is a tree, for every two vertices $u, v \in V(T)$, there exists exactly one path in T whose endpoints are u and v. So, let P be the path in T between w_1 and w_2. We know by hypothesis that $|\mathsf{L}(T)| \leq \ell$, therefore the length of P is at most $\ell + 1$, and so the distance in G between w_1 and w_2 is at most $\ell + 1$.

Conversely, if the distance in G between w_1 and w_2 is at most $\ell + 1$, then G contains a path P between w_1 and w_2 whose length is at most $\ell + 1$. Note that, P is a strict connection tree T for $W = \{w_1, w_2\}$ such that $|\mathsf{L}(T)| \leq \ell$ and $|\mathsf{R}(T)| = 0 \leq r$. Therefore, I is a Yes instance of S-TCP. ■

Lema 2.2. *Let T be a tree. If T has at least three leaves, then $\Delta(T) \geq 3$.*

Proof. Suppose that $\Delta(T) \leq 2$. Since T is a tree, either $|V(T)| = 1$ or T consists in a path with at least two vertices. If $|V(T)| = 1$, then clearly T has no leaf. On the other hand, if T is a path with at least two vertices, then T has exactly two leaves, which are its two endpoints. Therefore, if T has at least three leaves, then $\Delta(T) \geq 3$. ■

Corollary 2.3. *S-TCP($r = 0$) is polynomial-time solvable.*

Proof. Let $I = (G, W, \ell)$ be an instance of S-TCP($r = 0$). If $|W| = 2$, then we obtain by Lemma 2.1 that S-TCP($r = 0$) can be solved in polynomial-time simply by using a polynomial-time algorithm for the Shortest path problem. On the other hand, if $|W| \geq 3$, then it follows from Lemma 2.2 that every strict connection tree for W must contain at least one non-terminal vertex with degree greater than or equal to 3, which implies the existence of a router in T. Thus, if $|W| \geq 3$, then I is certainly a No instance of S-TCP($r = 0$). ■

We now analyse the computational complexity of S-TCP(r) with $r = 1$. Let G be a graph, $s, t \in V(G)$ and $P^1, P^2, \ldots, P^k$ be k paths in G between s and t. We say that $P^1, P^2, \ldots, P^k$ are *internally vertex-disjoints* (or simply, *vertex-disjoints*) if $V(P^i) \cap V(P^j) \setminus \{s, t\} = \varnothing$, for all

$i, j \in \{1, \ldots, k\}$ with $i \neq j$. Based on this concept of disjoint paths, we prove that S-TCP($r = 1$) is polynomial-time solvable. Our proof consists in a Turing reduction[1] to the so-called MIN-SUM st-DISJOINT PATHS (MIN-SUM st-DP) problem.

MIN-SUM st-DISJOINT PATHS (MIN-SUM st-DP)	
Instance:	A graph G, two vertices $s, t \in V(G)$ and two non-negative integers k and x.
Question:	Does there exist k vertex-disjoint paths between s and t in G, such that the sum of their lengths is at most x?

MIN-SUM st-DP is a classical disjoint path problem, proved to be polynomial-time solvable by Suurballe in 1974, as stated in the following theorem.

Theorem 2.4 (Suurballe [5]). MIN-SUM st-DP *is polynomial-time solvable.*

First, observe that, as S-TCP($r = 0$) admits a polynomial-time algorithm, we may suppose without loss of generality that an instance $I = (G, W, \ell)$ of S-TCP($r = 1$) is a NO instance of S-TCP($r = 0$), since a first natural strategy to solve I as an instance of S-TCP($r = 1$) is to verify (in polynomial time) whether G admits a strict connection tree for W which has at most ℓ linkers and has no router. Note also that, if $|W| \leq 2$ and I is a YES instance of S-TCP($r = 1$), then I is certainly a YES instance of S-TCP($r = 0$). Thus, we may also suppose that $|W| \geq 3$.

Given an instance $I = (G, W, \ell)$ of S-TCP($r = 1$) and a vertex $\rho \in V(G) \backslash W$ such that $d_G(\rho) \geq |W|$, we define an instance $f(I, \rho) = (G', s, t, k, x)$ of MIN-SUM st-DP corresponding to I as follows: $V(G') = V(G) \cup \{t\}$, where t is a new vertex, $E(G') = E(G) \cup \{wt \mid w \in W\}$, $s = \rho$, $k = |W|$ and $x = \ell + 2k$. Figure 2a illustrates an instance I of S-TCP($r = 1$) and the instance $f(I, \rho)$ of MIN-SUM st-DP corresponding to I.

[1] A *Turing reduction* from a problem Π to a problem Π' is an algorithm A that solves Π by using a hypothetical subroutine S for solving Π' such that, if S is a polynomial-time algorithm for Π', then A is a polynomial-time algorithm for Π [3].

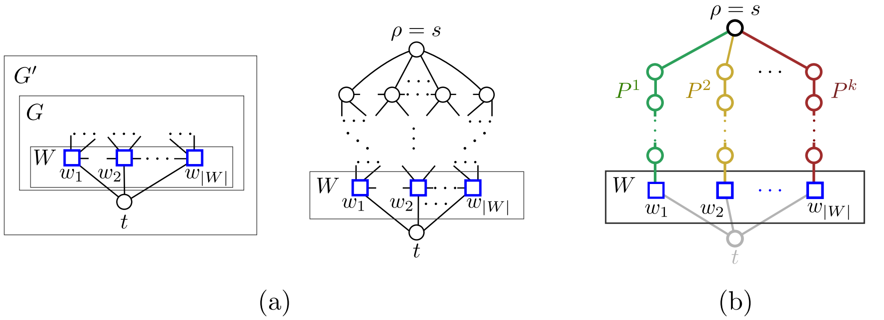


Figure 2: (a) Instance $f(I, \rho)$ of Min-sum st-DP. (b) Vertex-disjoint paths between s and t in G' and strict connection tree of G for W.

Lema 2.5. *An instance $I = (G, W, \ell)$ is a* Yes *instance of S-TCP($r = 1$) if and only if there exists a non-terminal vertex $\rho \in V(G) \setminus W$, with $d_G(\rho) \geq |W|$, such that $f(I, \rho)$ is a* Yes *instance of* Min-sum *st*-DP.

Proof. Suppose that I is a Yes instance of S-TCP($r = 1$). Let T be a strict connection tree of G for W such that $|L(T)| \leq \ell$ and $|R(T)| = 1$, and let $\rho \in \mathsf{R}(T)$ be the (unique) router of T. Since T is a tree and $W \subseteq V(T)$, we have that, for each terminal $w \in W$, there exists exactly one path in T whose endpoints are ρ and w. Furthermore, note that, those paths are vertex-disjoints, since by hypothesis every vertex in $V(T) \setminus \{\rho\}$ has degree at most 2 in T. Hence, we can construct a solution $\mathcal{P}$ (or more formally, a Yes certificate) for the instance $f(I, \rho)$ of Min-sum $st-$DP by the union of such paths between ρ and the terminals $w \in W$, along with the addition of the edges $wt \in E(G')$, *i.e.* $\mathcal{P} = \bigcup_{w \in W} (P_{\rho,w} \cup \{wt\})$, where $P_{\rho,w}$ is the path between ρ and w in T. It is easy to see that, $\mathcal{P}$ consists in $|W| = k$ vertex-disjoint paths of G' between $\rho = s$ and t such that the sum of their lengths is at most $\ell + 2|W| = \ell + 2k = x$. Therefore, $f(I, \rho) = (G', s, t, k, x)$ is a Yes instance of Min-sum st-DP.

On the other hand, suppose that there exists a non-terminal vertex $\rho \in V(G) \setminus W$, with $d_G(\rho) \geq |W|$, such that $f(I, \rho)$ is a Yes instance of Min-sum st-DP. Thus, G' contains $k = |W|$ vertex-disjoint paths $P^1, P^2, \ldots, P^k$ between s and t whose sum of their lengths is at most $x = \ell + 2k$. So, consider the subgraph T of G' induced by the edge set

$E' = \left(\bigcup_{1\leq i\leq k} E(P^i - t)\right)$. Clearly, T is a tree subgraph of G. Note also that, for $1 \leq i \leq k = |W|$, the endpoints of the paths $P^i - t$ constitute exactly the set $W \cup \{s\}$. Hence, the leaf set of T is equal to W. Moreover, since the paths $P^1 - t, P^2 - t, \ldots, P^k - t$ are internally vertex-disjoints, all vertices in $V(T) \setminus (\{s\} \cup W)$ have degree 2 in T. Thus, the unique vertex which has degree greater than or equal to 3 in T is s. Finally, note that, since $\sum_{1\leq i\leq k} \left(|E(P^i - t)| - 1\right) \leq x - 2k = \ell$, T has at most ℓ vertices with degree 2. So, T is a strict connection tree of G for W with at most ℓ linkers and exactly one router (see Figure 2b). Therefore, $I = (G, W, \ell)$ is a Yes instance of S-TCP($r = 1$). ■

Corollary 2.6. *S-TCP($r = 1$) is polynomial-time solvable.*

Proof. Let I be an instance of S-TCP($r = 1$). For each non-terminal vertex $\rho \in V(G) \setminus W$ with $d_G(\rho) \geq |W|$, we construct the instance $f(I, \rho)$ of Min-sum st-DP, as previously defined, and we verify in polynomial-time (based on Theorem 2.4) whether $f(I, \rho)$ is a Yes instance of Min-sum *st*-DP. If this is true for some $\rho \in V(G) \setminus W$, then we return that I is a Yes instance of S-TCP($r = 1$); otherwise (*i.e.* if, for all $\rho \in V(G) \setminus W$ with $d_G(\rho) \geq |W|$, $f(I, \rho)$ is No instance of Min-sum *st*-DP), we return that I is a No instance of S-TCP($r = 1$). The correctness of this algorithm follows from Lemma 2.5. ■

3 Conclusion

In the present extended abstract, we have considered S-TCP restricted to the case in which $r \in \{0, 1\}$. More precisely, we have shown that, for $r = 0$, the problem can be solved in polynomial time simply by analysing the cardinality of the terminal set W and using a polynomial-time algorithm for Shortest path; and, for $r = 1$, we have presented a Turing reduction to Min-sum *st*-DP, a classical polynomial-time solvable disjoint path problem. As future works, we intend to determine the computational complexity of S-TCP for all fixed $r \geq 2$.

References

[1] Dourado, M. C., Oliveira, R. A., Protti, F., e Souza, U. S. Design of connection networks with bounded number of non-terminal vertices. In *Proceedings of V Latin-American Workshop on Cliques in Graphs, LAWCliques 2012* (2014), vol. 42, *Matemática Contemporânea*, SBM, pp. 39–47.

[2] Dourado, M. C., Oliveira, R. A., Protti, F., e Souza, U. S. Conexão de terminais com número restrito de roteadores e elos. In *Proceedings of XLVI Simpósio Brasileiro de Pesquisa Operacional* (2014), pp. 2965–2976.

[3] Garey, M. R., e Johnson, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., 1979.

[4] Lin, G., e Xue, G. On the terminal steiner tree problem. *Information Processing Letters 84*, 2 (2002), pp. 103 – 107.

[5] Suurballe, J. W. Disjoint paths in a network. *Networks 4*, 2 (1974), pp. 125–145.

Alexsander A. Melo,
Celina M. H. de Figueiredo
UFRJ, Rio de Janeiro, Brasil.
{aamelo, celina}@cos.ufrj.br

Uéverton S. Souza
UFF, Niterói, Brasil.
ueverton@ic.uff.br