Received: 20 July 2020 | Accepted: 20 July 2020 | Published on: 30 August 2020
DOI: 10.1002/net.21976

SPECIAL ISSUE ARTICLE

WILEY

On undirected two-commodity integral flow, disjoint paths and strict terminal connection problems

Alexsander A. de Melo[1] | **Celina M. H. de Figueiredo**[1] | **Uéverton dos Santos Souza**[2]

[1]Program Systems Engineering and Computer Science - PESC/COPPE, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil
[2]Institute of Computing, Federal Fluminense University, Niterói, Brazil

Correspondence
Alexsander A. de Melo, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil.
Email: aamelo@cos.ufrj.br

Funding information
This research was supported by the Conselho Nacional de Desenvolvimento Científico e Tecnológico, Grant/Award Numbers: 140399/2017-8, 303726/2017-2, 407635/2018-1; Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, Grant/Award Number: Finance code 001; Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro, Grant/Award Numbers: E-26/202.793/2017, E-26/203.272/2017.

Abstract

Even, Itai, and Shamir (1976) proved simple two-commodity integral flow is NP-complete both in the directed and undirected cases. In particular, the directed case was shown to be NP-complete even if one demand is unitary, which was improved by Fortune, Hopcroft and Wyllie (1980) who proved the problem is still NP-complete if both demands are unitary. The undirected case, on the other hand, was proved by Robertson and Seymour (1995) to be polynomial-time solvable if both demands are constant. Nevertheless, the complexity of the undirected case with exactly one constant demand has remained unknown. We close this 40-year complexity gap, by showing the undirected case is NP-complete even if exactly one demand is unitary. As a by product, we obtain the NP-completeness of determining whether a graph contains $1+d$ pairwise vertex-disjoint paths, such that one path is between a given pair of vertices and d paths are between a second given pair of vertices. Additionally, we investigate the complexity of another related network design problem called strict terminal connection.

KEYWORDS

connection tree, disjoint paths, multicommodity integral flow, router vertices, Steiner tree, terminal vertices, unitary demand

1 | INTRODUCTION

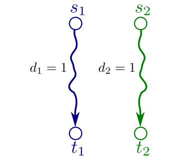
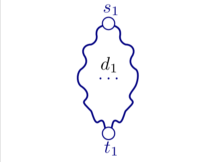
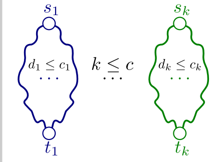
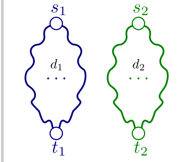
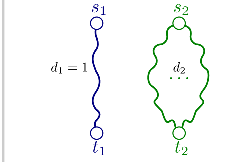
Network design constitutes one of the most important class of combinatorial problems, with a wide variety of theoretical and practical applications. Two fundamental network design problems are disjoint paths and, more generally, flow problems, which, besides their undoubted theoretical importance, have applications in VLSI design [16], transportation networks cf. [15] and routing networks [30], for instance.

The classical MAXIMUM FLOW problem, also known as SINGLE-COMMODITY FLOW, is polynomial-time solvable even when the flow function must be integral, that is, the flow is an integer-valued function [7]. A natural generalisation of SINGLE-COMMODITY FLOW is a flow problem with k *different commodities*, that is, a set of k pairs of source and sink vertices such that each pair has a distinct demand. This problem is the so-called MULTICOMMODITY FLOW. By using linear-programming, one can show that the MULTICOMMODITY FLOW problem is polynomial-time solvable, in the directed and undirected cases, when the flow of each commodity is a real-valued function cf. [3]. In contrast, Knuth proved that MULTICOMMODITY FLOW is an NP-complete problem, in both directed and undirected cases, if the flow functions must be integral cf. [9,13].

An instance of a network flow problem is called *simple* if the capacities of all edges of the input network are unitary. In 1976, Even et al. extended the result due to Knuth by proving the NP-completeness of SIMPLE TWO-COMMODITY INTEGRAL FLOW (SIMPLE 2CIF), also in the directed and undirected cases [9]. For the directed case, they proved that the problem is

†An extended abstract of this paper was published in the proceedings of IX Latin and American Algorithms, Graphs and Optimization Symposium [21].

TABLE 1 Contribution of this work (last column) and known complexity results for SIMPLE MULTICOMMODITY INTEGRAL FLOW, both in directed and undirected cases, and for the related vertex-disjoint paths problems, where k denotes the number of commodities and d_i denotes the demand of the commodity (s_i, t_i)

Directed case	Undirected case			
				
Fixed $k \geq 2$: NP-c Even if $d_i = 1\ \forall i$ (Fortune et al. [11])	$k = 1$: Poly Even if d_1 is arbitrary (Edmonds and Karp [7])	Fixed $k \geq 1$: Poly If d_i is fixed $\forall i$ (Roberson and Seymour [26])	Fixed $k \geq 2$: NP-c d_1 and d_2 are both arbitrarily large (Even et al. [9])	**Fixed $k \geq 2$: NP-c** Even if $d_i = 1\ \forall i \neq k$ and only d_k is arbitrarily large (Theorem 2 and Corollary 4)

still NP-complete if the demand of exactly one commodity is unitary. Nevertheless, for the undirected case, the hard instance constructed by them does not satisfy the condition of having a commodity with unitary demand, or even a commodity with constant demand. In other words, in their proof, both commodities must have arbitrarily large demands. Thus, in the present paper, we close this 40-year gap by establishing the NP-completeness of SIMPLE UNDIRECTED TWO-COMMODITY INTEGRAL FLOW (SIMPLE U2CIF) when the demand of exactly one commodity is unitary.

SIMPLE U2CIF is a particular case of the EDGE-DISJOINT PATHS problem, which through the line graph reduces to the VERTEX-DISJOINT PATHS problem, proved to be NP-complete cf. [13]. On the other hand, when k is fixed, Robertson and Seymour gave an $\mathcal{O}(n^3)$-time algorithm for VERTEX-DISJOINT PATHS [26], which was improved to $\mathcal{O}(n^2)$-time by Kawarabayashi et al. [14], where n denotes the number of vertices of the input graph. Consequently, EDGE-DISJOINT PATHS for fixed k and so, SIMPLE U2CIF with the two demands bounded by constants are both polynomial-time solvable. With respect to the directed case, when $k \geq 2$ is fixed, Fortune et al. proved that VERTEX-DISJOINT PATHS is NP-complete [11], and a slight change of the proof also provides the NP-completeness of the edge-disjoint version cf. [8]. We remark that, if the demands of both commodities are unitary, SIMPLE 2CIF coincides with EDGE-DISJOINT PATHS when $k = 2$. Consequently, SIMPLE DIRECTED 2CIF remains NP-complete even if both demands are unitary.

Moreover, we observe that SIMPLE U2CIF with exactly one unitary demand coincides with the decision problem of determining whether an undirected graph contains $1 + d$ pairwise edge-disjoint paths, such that one of these paths is between a given pair of vertices and d of these paths are between another given pair of vertices. As a corollary of the proposed NP-completeness of SIMPLE U2CIF with exactly one unitary demand, we also obtain the NP-completeness of $1 + d$ DISJOINT PATHS, both the edge and vertex disjoint versions.

Table 1 summarizes the computational complexity of the problems discussed in the previous paragraphs and highlights where exactly our contribution lies. For thorough references on integral flow and disjoint paths problems, we refer to [24,28]. It is worth mentioning that, not only do our results close a long-standing complexity gap of a class of fundamental problems, but they also reaffirm the fact brought up by Naves and Sebő [24] that, among the variety of possibilities, there are some interesting questions related to integral flow and disjoint paths that may not even have been realised. As a theoretical application of our results, we further analyse the complexity of the STRICT TERMINAL CONNECTION problem, which is a variant of STEINER TREE, closely related to vertex-disjoint paths problems.

Given a graph G, a terminal set $W \subseteq V(G)$ and two nonnegative integers ℓ and r, the STRICT TERMINAL CONNECTION problem (S-TCP) aims to decide whether G contains a tree subgraph whose leaf set coincides with W and that has at most ℓ vertices of degree exactly 2 and at most r vertices of degree at least 3. Dourado et al. proved that S-TCP is NP-complete, for each fixed $\ell \geq 0$, and polynomial-time solvable when ℓ and r are simultaneously fixed [5]. Regarding the complexity of S-TCP when only r is fixed, Melo et al. showed that the problem can be solved in polynomial-time when $r \in \{0, 1\}$ [19]. More specifically, they showed that, for $r \in \{0, 1\}$, S-TCP is Turing reducible to the problem of deciding whether a graph admits d vertex-disjoint paths between a single given pair of vertices, whose sum of their lengths is at most a given positive integer x, which was proved to be polynomial-time solvable by Suurballe [31]. In addition, Melo et al. studied S-TCP from the perspective of graph classes and parameterized complexity [20]. It was proved that S-TCP restricted to split graphs can be solved in time $n^{\mathcal{O}(r)}$ but that the existence of an $f(r) \cdot n^{\mathcal{O}(1)}$-time algorithm is unlikely for any computable function f, where n denotes the number of vertices of the input graph. In spite of these results, for fixed $r \geq 2$, the complexity of S-TCP on general graphs remains unsettled. It is widely open whether, under reasonable complexity assumptions, S-TCP on general graphs admits an $n^{\mathcal{O}(r)}$-time algorithm.

The nonstrict variant, called TERMINAL CONNECTION (TCP), has the same input of S-TCP but asks for tree subgraphs whose leaf sets contain the terminal set W—instead of coinciding with W—and that have at most ℓ nonterminal vertices of degree exactly 2 and at most r nonterminal vertices of degree at least 3. Similarly to S-TCP, Dourado et al. proved that TCP is NP-complete, for each fixed $\ell \geq 0$, and is polynomial-time solvable when ℓ and r are both bounded by constants [4]. On the other hand, they also proved that TCP remains NP-complete when $r \geq 0$ is fixed.

Besides being related to disjoint paths, S-TCP and TCP might be seen as variants, or even generalisations, of the classical STEINER TREE problem. STEINER TREE has as input a graph G, a terminal set $W \subseteq V(G)$ and a nonnegative integer x, and aims to decide whether G contains a tree subgraph, called *Steiner tree*, that contains all vertices belonging to W and has at most x nonterminal vertices, commonly called *Steiner vertices*. Among the several variants of STEINER TREE studied over the years, the FULL STEINER TREE problem asks for Steiner trees whose leaf sets coincide with the terminal set W cf. [12,17,18]; and another variant requires the number of *branching nodes*, that is, vertices of degree at least 3 in the Steiner tree—which not necessarily are Steiner vertices—, to be bounded cf. [34,35]. Nevertheless, there is no variant of STEINER TREE in the literature that requires *full Steiner trees* with bounded number of branching nodes. Therefore, we emphasise that S-TCP merits to be studied.

In this work, we focus on the open question regarding the complexity of S-TCP when $r \geq 2$ is fixed. We establish some relations between S-TCP with fixed r and vertex-disjoint paths problems. Furthermore, in order to advance on the understanding of such a question, we study variants of S-TCP for which additional constraints on topology and connectedness are imposed.

2 | SIMPLE UNDIRECTED TWO-COMMODITY INTEGRAL FLOW

In this section, we present one of the main contributions of this paper, which consists in the NP-completeness proof of SIMPLE UNDIRECTED TWO-COMMODITY INTEGRAL FLOW when the demand of exactly one commodity is unitary. This proof closes a 40-year complexity gap. Next, we present a formal definition for the problem.

SIMPLE UNDIRECTED TWO-COMMODITY INTEGRAL FLOW (SIMPLE U2CIF)

Input:	A graph G, two distinct unordered pairs of vertices $\{s_1, t_1\}$ and $\{s_2, t_2\}$ of G and two positive integers, called *demands*, d_1 and d_2.
Question:	Are there two flow functions $f_1, f_2 : \{\overrightarrow{uv}, \overrightarrow{vu} \mid uv \in E(G)\} \rightarrow \mathbb{Z}_0^+$ such that
	(a) for each $i \in \{1, 2\}$ and each edge $uv \in E(G)$, $f_i(\overrightarrow{uv}) = 0$ or $f_i(\overrightarrow{vu}) = 0$;
	(b) for each $i \in \{1, 2\}$ and each vertex $v \in V(G) \backslash s_i, t_i$, the flow function f_i is *conserved* at v, i.e., $\sum_{u \in N_G(v)} f_i(\overrightarrow{uv}) = \sum_{u \in N_G(v)} f_i(\overrightarrow{vu})$;
	(c) for each $i \in 1, 2$, the net flow from s_i, i.e., $\sum_{v \in N_G(s_i)} (f_i(\overrightarrow{s_i v}) - f_i(\overrightarrow{v s_i}))$, is at least d_i; and
	(d) for each edge $uv \in E(G)$, the total flow through uv, i.e., $\max\{f_1(\overrightarrow{uv}), f_1(\overrightarrow{vu})\} + \max\{f_2(\overrightarrow{uv}), f_2(\overrightarrow{vu})\}$, is at most 1?

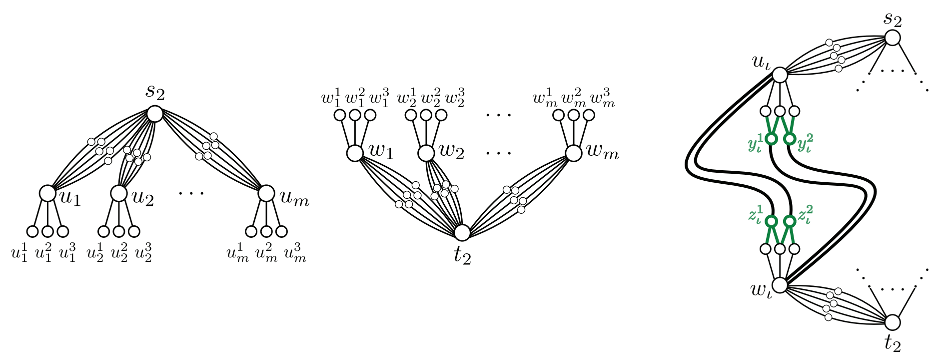
The NP-completeness proof that we provide for SIMPLE U2CIF with exactly one unitary demand is built on the polynomial-time reduction described below, from the variant of 3-SAT in which each clause has three distinct literals.

Construction 1. *Let $I = (X, C)$ be an instance of 3-SAT, with variable set $X = \{x_1, \dots, x_n\}$ and clause set $C = \{C_1, \dots, C_m\}$, such that each clause in C has exactly three distinct literals. We let $g(I) = (G, \{s_1, t_1\}, \{s_2, t_2\}, d_1, d_2)$ be the instance of* SIMPLE U2CIF *defined follows.*

- *We create the vertices s_1, t_1, s_2 and t_2.*
- *For each variable $x_i \in X$, we create the variable gadget H_i such that*
  - $V(H_i) = \{v_i^s, v_i^t\} \cup \{v_i^\ell \mid \ell \in \{1, \dots, 2p_i\}\} \cup \{\overline{v}_i^\ell \mid \ell \in \{1, \dots, 2q_i\}\}$ *and*
  - $E(H_i) = \{v_i^s v_i^1, v_i^s \overline{v}_i^1, v_i^{2p_i} v_i^t, \overline{v}_i^{2q_i} v_i^t\} \cup \{v_i^\ell v_i^{\ell+1} \mid \ell \in \{1, \dots, 2p_i - 1\}\} \cup \{\overline{v}_i^\ell \overline{v}_i^{\ell+1} \mid \ell \in \{1, \dots, 2q_i - 1\}\}$,

  where p_i and q_i denote the numbers of positive and negative occurrences of x_i in I, respectively.
- *We connect the variable gadgets in series, that is, we add the edges $v_i^t v_{i+1}^s$ for each $i \in \{1, \dots, n-1\}$. Additionally, we add the edges $s_1 v_1^s$ and $v_n^t t_1$.*
- *For each clause $C_\iota \in C$, we create the clause vertices u_ι and w_ι; for each $\kappa \in \{1, \dots, 5\}$, we create the vertices a_ι^κ and b_ι^κ and add the edges $s_2 a_\iota^\kappa$, $a_\iota^\kappa u_\iota$, $w_\iota b_\iota^\kappa$ and $b_\iota^\kappa t_2$; moreover, for each $j \in \{1,2,3\}$, we create the literal vertices u_ι^j and w_ι^j and add the edges $u_\iota u_\iota^j$ and $w_\iota^j w_\iota$ (see* Figure 1A*).*
- *In addition, for each clause $C_\iota \in C$, we create the vertices y_ι^1, y_ι^2, z_ι^1 and z_ι^2 and add the edges $u_\iota^1 y_\iota^1$, $u_\iota^2 y_\iota^1$, $y_\iota^1 w_\iota$, $u_\iota^2 y_\iota^2$, $u_\iota^3 y_\iota^2$, $y_\iota^2 w_\iota$ and the edges $z_\iota^1 w_\iota^1$, $z_\iota^1 w_\iota^2$, $u_\iota z_\iota^1$, $z_\iota^2 w_\iota^2$, $z_\iota^2 w_\iota^3$ and $u_\iota z_\iota^2$ (see* Figure 1B*).*
- *For each clause $C_\iota \in C$, we also add the edges $u_\iota^j v_i^{2\ell-1}$ and $v_i^{2\ell} w_\iota^j$ if the j-th literal in C_ι corresponds to the ℓ-th occurrence of the positive literal x_i, where $j \in \{1,2,3\}$ and $\ell \in \{1, \dots, p_i\}$; on the other hand, we add the edges $u_\iota^j \overline{v}_i^{2\ell-1}$ and $\overline{v}_i^{2\ell} w_\iota^j$ if the jth literal in C_ι corresponds to the ℓth occurrence of the negative literal $\overline{x}_i$, where $j \in \{1,2,3\}$ and $\ell \in \{1, \dots, q_i\}$.*
- *Finally, we define $d_1 = 1$ and $d_2 = 5m$.*

Figure 2 exemplifies the instance $g(I)$ of SIMPLE U2CIF, described in Construction 1.



(A) Vertices: s_2, a_ι^κ, u_ι and u_ι^j; and vertices: w_ι^j, w_ι, b_ι^κ and t_2. (B) Vertices: y_1^ι, y_2^ι, z_1^ι and z_2^ι.

FIGURE 1 Partial construction of the instance $g(I)$ of SIMPLE U2CIF: vertices and edges obtained from the clause set C of a given instance I of 3-SAT. For readability, the labels of the vertices a_ι^κ and b_ι^κ are omitted [Color figure can be viewed at wileyonlinelibrary.com]

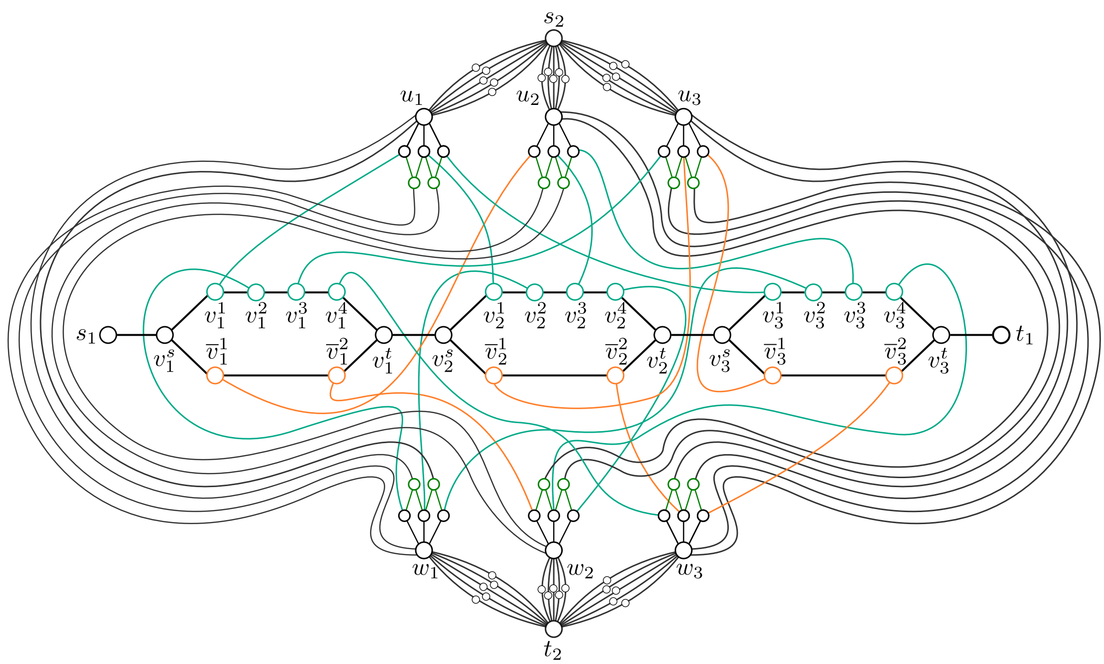


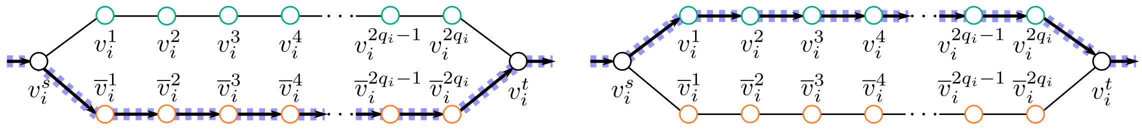
FIGURE 2 Graph G of the instance $g(I)$ of SIMPLE U2CIF, obtained from the instance $I = (X, C)$ of 3-SAT where $X = \{x_1, x_2, x_3\}$ and $C = \{C_1 = \{x_1, x_2, x_3\}, C_2 = \{\overline{x}_1, x_2, x_3\}, C_3 = \{x_1, \overline{x}_2, \overline{x}_3\}\}$ [Color figure can be viewed at wileyonlinelibrary.com]

Theorem 2. SIMPLE U2CIF *remains NP-complete even if the demand of exactly one commodity is unitary.*

Proof. Let $I = (X, C)$ be an instance of 3-SAT such that $X = \{x_1, \ldots, x_n\}$, $C = \{C_1, \ldots, C_m\}$ and each clause in C has exactly three distinct literals. Let $g(I) = (G, \{s_1, t_1\}, \{s_2, t_2\}, d_1, d_2)$ be the instance of SIMPLE U2CIF described in Construction 1.

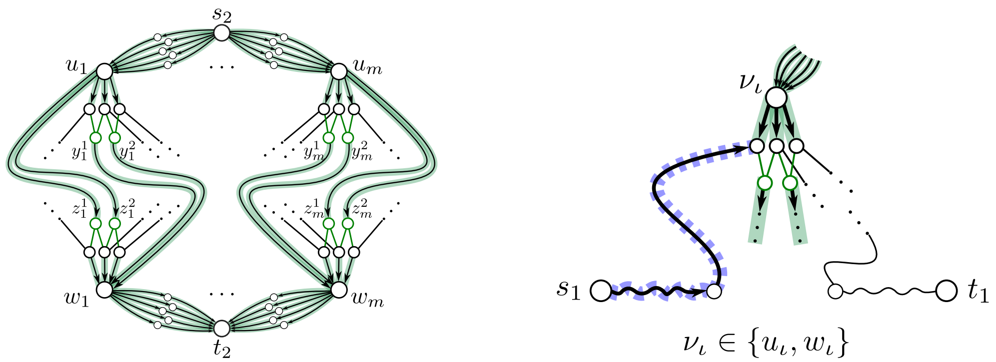
Assume that there exists a truth assignment $\alpha : X \rightarrow \{true, false\}$ that satisfies all clauses in C. Based on α, we define flow functions f_1 and f_2 for the first and the second commodities, respectively. Initially, consider $f_i(\overrightarrow{uv}) = 0$ and $f_i(\overrightarrow{vu}) = 0$ for each $uv \in E(G)$ and each $i \in \{1, 2\}$. Next, update f_1 as follows: for each $x_i \in X$, if $\alpha(x_i) = true$, then the first commodity flow passes through the lower part of H_i (see Figure 3A); otherwise, it passes through the upper part of H_i (see Figure 3B). Additionally, set $f_1(\overrightarrow{s_1 v_1^s}) = 1, f_1(\overrightarrow{v_1^t v_2^s}) = 1, \ldots, f_1(\overrightarrow{v_{n-1}^t v_n^s}) = 1$ and $f_1(\overrightarrow{v_n^t t_1}) = 1$.

For each $C_\iota \in C$, let γ_ι be a true literal under α of C_ι. Assume that γ_ι is the j-th literal in C_ι, for some $j \in \{1,2,3\}$. Then, update f_2 as follows: if γ_ι is the ℓ-th occurrence of the positive literal x_i, then the second commodity flow passes through the path $\langle s_2, a_\iota^1, u_\iota, u_\iota^j, v_i^{2\ell-1}, v_i^{2\ell}, w_\iota^j, w_\iota, b_\iota^1, t_2\rangle$; on the other hand, if γ_ι is the jth occurrence of the negative literal $\overline{x}_i$, then the second commodity flow passes through the path $\langle s_2, a_\iota^1, u_\iota, u_\iota^j, \overline{v}_i^{2\ell-1}, \overline{v}_i^{2\ell}, w_\iota^j, w_\iota, b_\iota^1, t_2\rangle$. Note that, only $|C|$ units of the second commodity flow, from s_2 into t_2, have been sent. The remaining $4|C|$ units are sent through the paths $\langle s_2, a_\iota^2, u_\iota, u_\iota^{j_1}, y_\iota^1, w_\iota, b_\iota^2, t_2\rangle$ and $\langle s_2, a_\iota^3, u_\iota, u_\iota^{j_2}, y_\iota^2, w_\iota, b_\iota^3, t_2\rangle$ and the paths $\langle s_2, a_\iota^4, u_\iota, z_\iota^1, w_\iota^{j_1}, w_\iota, b_\iota^4, t_2\rangle$ and $\langle s_2, a_\iota^5, u_\iota, z_\iota^2, w_\iota^{j_2}, w_\iota, b_\iota^5, t_2\rangle$, where $j_1, j_2 \in \{1,2,3\} \setminus \{j\}$ and $j_1 < j_2$. Clearly, f_1 and f_2 meet the demands $d_1 = 1$ and $d_2 = 5m$,



(A) $\alpha(x_i)$ = *true.* (B) $\alpha(x_i)$ = *false*

FIGURE 3 The first commodity flow passing through the lower and upper parts of the variable gadget H_i, according to the truth values $\alpha(x_i) = true$ and $\alpha(x_i) = false$, respectively [Color figure can be viewed at wileyonlinelibrary.com]



(A) Edges necessarily used by the second commodity flow. (B) Flow obstruction.

FIGURE 4 Edges that must be used by the second commodity flow; and flow obstruction as a consequence of the first commodity flow using an edge whose either endpoint is not a vertex of the variable gadgets [Color figure can be viewed at wileyonlinelibrary.com]

respectively, and satisfy the unitary capacity and flow conservation constraints. Moreover, for each $uv \in E(G)$ and each $i \in \{1, 2\}$, $f_i(\overrightarrow{uv}) = 0$ or $f_i(\overrightarrow{vu}) = 0$. Therefore, we obtain that $g(I)$ is a YES-instance of SIMPLE U2CIF.

Conversely, suppose now that there exist flow functions f_1 and f_2 that certify that $g(I)$ is a YES-instance of SIMPLE U2CIF. Since $d_G(s_2) = d_G(t_2) = 5m = d_2$, all edges incident to s_2 and all edges incident to t_2 are used by the second commodity flow, that is, $f_2(\overrightarrow{s_2u}) = 1$ for each $u \in N_G(s_2)$, and $f_2(\overrightarrow{vt_2}) = 1$ for each $v \in N_G(t_2)$. Furthermore, $d_G(u) = 2$ for each $u \in N_G(s_2)$ and $d_G(v) = 2$ for each $v \in N_G(t_2)$. Thus, all edges incident to the neighbours of s_2 and all edges incident to neighbours of t_2 are used by the second commodity flow. Moreover, $\cup_{u\in N_G(s_2)} N_G(u)\backslash\{s_2\} = \{u_\iota | C_\iota \in \mathcal{C}\}$, $\cup_{v\in N_G(t_2)} N_G(v)\backslash\{t_2\} = \{w_\iota | C_\iota \in \mathcal{C}\}$ and, for each $C_\iota \in \mathcal{C}$, $|N_G(u_\iota)\backslash N_G(s_2)| = 5$ as well as $|N_G(w_\iota)\backslash N_G(t_2)| = 5$. Consequently, all edges incident to the vertices $u_1, u_2, \ldots, u_m$ and all edges incident to the vertices $w_1, w_2, \ldots, w_m$ are also used by the second commodity flow (see Figure 4A). Then, apart from $s_1v_1^s$ and $v_n^tt_1$, the first commodity flow only uses edges whose both endpoints belong to $\cup_{x_i\in X} V(H_i)$, that is, for each $uv \in E(G)\backslash\{s_1v_1^s, v_n^tt_1\}$ such that $\{u, v\} \not\subseteq \cup_{x_i\in X} V(H_i)$, we have that $f_1(\overrightarrow{uv}) = 0$ and $f_1(\overrightarrow{vu}) = 0$, otherwise the unitary capacity and flow conservation constraints would not be simultaneously satisfied by f_1 and f_2 (see Figure 4B).

As a result, the first commodity flow passes through all the variable gadgets H_i, for $x_i \in X$. Thus, we can define a truth assignment $\alpha : X \to \{true, false\}$ as follows: if the first commodity flow passes through the lower part of H_i, then we set $\alpha(x_i) = true$; otherwise, if it passes through the upper part of H_i, then we set $\alpha(x_i) = false$. It is easy to verify that, for each clause $C_\iota \in \mathcal{C}$, the second commodity flow necessarily uses, for some $x_i \in X$, an edge $e_\iota \in E(H_i)$ such that either $e_\iota = v_i^\ell v_i^{\ell+1}$ or $e_\iota = \overline{v}_i^l\overline{v}_i^{l+1}$, where $\ell \in \{1, \ldots, 2p_i - 1\}$ and $l \in \{1, \ldots, 2q_i - 1\}$. If $e_\iota = v_i^\ell v_i^{\ell+1}$, then the first commodity flow must pass through the lower part of H_i, and so the truth value $\alpha(x_i) = true$ satisfies C_ι; on the other hand, if $e_\iota = \overline{v}_i^l\overline{v}_i^{l+1}$, then the first commodity flow must pass through the upper part of H_i, and so the truth value $\alpha(x_i) = false$ satisfies C_ι. Consequently, the truth assignment α satisfies all clauses belonging to $\mathcal{C}$. Therefore, I is a YES-instance of 3-SAT. ■

3 | 1+*d* VERTEX-DISJOINT PATHS

A problem closely related to SIMPLE U2CIF is EDGE-DISJOINT PATHS, which has as input a graph G and $k \geq 1$ distinct unordered pairs $\{s_1, t_1\}, \{s_2, t_2\}, \ldots, \{s_k, t_k\}$ of vertices of G, and asks whether G contains k pairwise edge-disjoint paths $P_1, P_2, \ldots, P_k$ such that, for each $i \in \{1, \ldots, k\}$, P_i is a path between s_i and t_i. We remark that SIMPLE U2CIF is a particular case of EDGE-DISJOINT PATHS. Indeed, let $I = (G, \{s_1, t_1\}, \{s_2, t_2\}, d_1, d_2)$ be an instance of SIMPLE U2CIF and $I' = (G', \{s_1^1, t_1^1\}, \ldots, \{s_1^{d_1}, t_1^{d_1}\}, \{s_2^1, t_2^1\}, \ldots, \{s_2^{d_2}, t_2^{d_2}\})$ be the instance of EDGE-DISJOINT PATHS, where $s_i^1 = s_i$, $t_i^1 = t_i$ and G' is the graph obtained from G by simply adding $d_i - 1$ twins $s_i^2, \ldots, s_i^{d_i}$ of the vertex s_i, and adding $d_i - 1$ twins $t_i^2, \ldots, t_i^{d_i}$ of the vertex t_i, for each $i \in \{1, 2\}$. One can easily verify that I is a YES-instance of SIMPLE U2CIF if and only if I' is a YES-instance of EDGE-DISJOINT PATHS.

A more interesting observation relating SIMPLE U2CIF with disjoint paths is the fact that SIMPLE U2CIF with exactly one unitary demand coincides with the problem $1+d$ EDGE-DISJOINT PATHS ($1+d$ EDP), which has as input a graph G, two distinct unordered pairs $\{s_1, t_1\}$, $\{s_2, t_2\}$ of vertices of G and a positive integer d, and asks whether G contains $1+d$ pairwise edge-disjoint paths, such that one path is between s_1 and t_1 and d paths are between s_2 and t_2. As a result, it immediately follows from Theorem 2 that $1+d$ EDP is an NP-complete problem. Thus, a natural question that arises is about the complexity of $1+d$ VERTEX-DISJOINT PATHS ($1+d$ VDP), the vertex-disjoint version of $1+d$ EDP.

We show that $1+d$ VDP is also an NP-complete problem. More specifically, building on the fact that edge-disjoint paths problems are polynomially reducible to their respective vertex-disjoint paths problems through the line graph cf. [14,25,28], we obtain as a corollary of Theorem 2 the NP-completeness of $1+d$ VDP.

Construction 3. *Let* $I = (G, \{s_1, t_1\}, \{s_2, t_2\}, d_1, d_2)$ *be an instance of* SIMPLE U2CIF*, such that* $d_1 = 1$*. Then, we let* $g(I) = (G', \{s_1', t_1'\}, \{s_2', t_2'\}, d)$ *be the instance of* $1+d$ VDP *defined as follows.*

- *We create the vertices* s_1', t_1', s_2' *and* t_2'.
- *We define* G' *as the graph with vertex set and edge set*
  - $V(G') = V(L(G)) \cup \{s_1', t_1', s_2', t_2'\}$ *and*
  - $E(G') = E(L(G)) \cup s_i'e \mid e \text{ is incident to } s_i, i \in \{1,2\}\Big\} \cup \{t_i'e \mid e \text{ is incident to } t_i, i \in \{1,2\}\}$, *respectively, where* $L(G)$ *denotes the line graph of* G*,* that is, *the graph with vertex set* $V(L(G)) = E(G)$ *and edge set* $E(L(H)) = \{ee' \mid e \text{ and } e' \text{ share an endpoint}\}$.
- *Finally, we define* $d = d_2$.

Corollary 4. $1+d$ VDP *is NP-complete*.

Proof. Let I be an instance of SIMPLE U2CIF with $d_1 = 1$, and let $g(I)$ be the instance of $1+d$ VDP described in Construction 3. Note that $g(I)$ can be constructed in time polynomial in the size of I. Furthermore, since SIMPLE U2CIF with exactly one unitary demand coincides with $1+d$ EDP, one can verify that I is an YES-instance of SIMPLE U2CIF if and only if $g(I)$ is an YES-instance of $1+d$ VDP. Therefore, $1+d$ VDP is NP-complete. ■

4 | CONNECTING TERMINALS WITH FEW ROUTERS

Let G be a graph and W be a nonempty subset of $V(G)$. A *connection tree* T of G for W is a tree subgraph of G such that $leaves(T) \subseteq W \subseteq V(T)$. The vertices in W are called *terminal*, and the vertices in $V(T) \backslash W$ are called *nonterminal* and are classified into two types according to their respective degrees in T, namely: the nonterminal vertices of degree exactly 2 in T are called *linkers* and the nonterminal vertices of degree at least 3 in T are called *routers*. We let $\mathrm{L}(T) = \{u \in V(T) \mid d_T(u) = 2\}$ denote the linker set of T and $\mathrm{R}(T) = \{u \in V(T) \mid d_T(u) \geq 3\}$ denote the router set of T. We remark that the vertex set of every connection tree can be partitioned into terminal vertices, linkers and routers.

Note that, the definition of connection tree does not impose any restriction on the degree of terminal vertices. However, in some applications, such as in telecommunications cf. [17], terminal vertices are not allowed to behave as linkers or routers, they must be leaves. A connection tree T for W is said to be *strict* if all vertices belonging to W are leaves of T, that is, $leaves(T) = W$. Next, we present a formal definition for the STRICT TERMINAL CONNECTION problem.

STRICT TERMINAL CONNECTION problem (S-TCP)	
Input:	A connected graph G, a terminal set $W \subseteq V(G)$ with $\lvert W\rvert \geq 2$, and two nonnegative integers ℓ and r.
Question:	Does G admit a strict connection tree T for W such that $\lvert \mathrm{L}(T)\rvert \leq \ell$ and $\lvert \mathrm{R}(T)\rvert \leq r$?

S-TCP is strongly related to vertex-disjoint paths problems. As shown in [19], S-TCP with $r = 1$ is Turing reducible to the MIN-SUM *st*-VERTEX-DISJOINT PATHS problem (MIN-SUM *st*-VDP), which has as input a graph G, a single pair $\{s, t\}$ of vertices of G and two positive integers d and x, and asks whether G contains d pairwise vertex-disjoint paths between s and t whose sum of their lengths is at most x. Additionally, one can verify without much effort that S-TCP with $r = 2$ is Turing reducible to the min-sum version of $1+d$ VDP, which is an NP-complete problem by Corollary 4.

More generally, given an instance $I = (G, W, \ell, r)$ of S-TCP and a set $R \subseteq V(G) \backslash W$ such that $2 \leq \lvert R\rvert \leq r$, a strategy to decide whether G admits a strict connection tree for W such that $\lvert \mathrm{L}(T)\rvert \leq \ell$ and $\mathrm{R}(T) = R$ may consist in solving in a combined way the following two problems.

I. Connecting the terminals to the vertices in R, through vertex-disjoint paths whose sum of their lengths is bounded, in a way that each terminal is connected to exactly one vertex in R and, for a given map $f : R \rightarrow \{0,1,2\}$, each $\rho \in R$ connects at least $f(\rho)$ terminals.

II. Connecting the vertices in R to one another, through vertex-disjoint paths whose sum of their lengths is bounded.

We remark that, even if both of these problems are separately polynomial-time solvable, not necessarily the corresponding case of S-TCP can be solved in polynomial-time. On the other hand, it is easy to see that the NP-completeness of Problem (II) implies the NP-completeness of the corresponding case of S-TCP. However, through a Turing reduction to MIN-SUM *st*-VDP, one can prove that, for $|R| \leq 3$, Problem (II) is polynomial-time solvable. For fixed $|R| \geq 4$, nevertheless, the complexity of Problem (II) is surprisingly unknown. We remark that, by considering R as the terminal set, Problem (II) might be seen as the variant of STEINER TREE that asks for Steiner trees whose nonterminal vertices (i.e., *Steiner vertices*) must have degree 2, and through polynomial-time reductions it might be seen as a generalisation of the so-called SHORTEST K-CYCLE problem, whose complexity for fixed number of terminals is a long-standing open question cf. [2,33]. With respect to Problem (I), one can also prove by a Turing reduction to MIN-SUM *st*-VDP, similar to the one described in Lemma 10, that Problem (I) is polynomial-time solvable, even if $|R|$ is not fixed.

Besides the relations described above, we have that, if r and the number of terminal vertices are both fixed, then the variant of S-TCP that does not impose any restriction on the number of linkers can be solved in polynomial-time through a Turing reduction to the VERTEX-DISJOINT PATHS problem. A question that naturally arises from this remark is about using a similar approach in order to solve the original S-TCP—which asks for strict connection trees with a bounded number linkers—when both $r \geq 2$ and the number of terminals are fixed. However, for fixed $k \geq 2$, the complexity of MIN-SUM VERTEX-DISJOINT PATHS (MIN-SUM VDP), the min-sum version of VERTEX-DISJOINT PATHS, has remained open for more than 20 years cf. [8,10,15]. Despite the recent result due to Björklund and Husfeldt [1], which proves that, for $k = 2$, MIN-SUM VDP admits a Monte Carlo polynomial-time algorithm, there is no known deterministic polynomial-time algorithm for the problem.

Most of the Turing reductions mentioned in the previous paragraphs have as a subroutine enumerating all sets $R \subseteq V(G)\backslash W$ with $|R| \leq r$ and, then, asking for the existence of a strict connection tree T for W, such that $|\mathrm{L}(T)| \leq \ell$ and $\mathrm{R}(T) \subseteq R$. Motivated by this fact, one can also consider the variant of S-TCP called CONSTRAINED ROUTER SET, which has as input a connected graph G, a terminal set $W \subseteq V(G)$ with $|W| \geq 2$, a nonnegative integer ℓ and a nonempty set $R \subseteq V(G)\backslash W$, and asks whether G admits a strict connection tree T for W such that $|\mathrm{L}(T)| \leq \ell$ and $\mathrm{R}(T) \subseteq R$.

Clearly, for fixed $r \geq 0$, S-TCP is Turing reducible to CONSTRAINED ROUTER SET. Indeed, $I = (G, W, \ell, r)$ is a YES-instance of S-TCP if and only there exists a set $R \subseteq V(G)\backslash W$, with $|R| \leq r$, such that (G, W, ℓ, R) is a YES-instance of CONSTRAINED ROUTER SET. Thus, if CONSTRAINED ROUTER SET is polynomial-time solvable when $|R| \in \{0, \dots, c\}$ for some fixed $c \geq 0$, then S-TCP is polynomial-time solvable when $r \leq c$. However, the converse not necessarily holds: it is not clear whether, for fixed $|R| \geq 0$, CONSTRAINED ROUTER SET is Turing reducible to S-TCP with fixed r.

To better understand S-TCP when $r \geq 2$ is fixed, we investigate in Section 4.1 the complexity of some variants of the problem for which further constraints on the topology and/or connectedness of the router set are imposed. Figure 5 provides a general overview on the complexity landscape of S-TCP and its variants, focusing especially on the relationship (i.e., the existence of polynomial-time or Turing reduction) between the vertex-disjoint paths problems and the variants of S-TCP discussed in the previous paragraphs, and the relationship of the former problems with the variants of S-TCP that we analyse in Section 4.1.

4.1 | Constraints on topology and connectedness

Let G be a graph, $W \subseteq V(G)$ be a terminal set and T be a strict connection tree of G for W. We say that a terminal vertex $w \in W$ is *connected in* T by a router $\rho \in \mathrm{R}(T)$ if the distance in T between w and ρ is less than the distance in T between w and any other router $\rho' \in \mathrm{R}(T)\backslash\rho$. In this case, we equivalently say that ρ *connects* w in T. We remark that every terminal vertex belonging to W is connected in T by exactly one router.

The first variant of S-TCP that we consider is called CONSTRAINED TERMINAL PARTITION. Note that, every strict connection tree T for a terminal set W induces a partition $\mathcal{W}$ of W, such that the terminal vertices belonging to a same part of $\mathcal{W}$ are connected in T by a same router. Motivated by this fact, it is interesting to analyse the variant of S-TCP which additionally gives in its input a partition of the terminal set and asks for strict connection trees whose terminal vertices belonging to a same part of the partition are enforced to be connected by a same router.

CONSTRAINED TERMINAL PARTITION

Input:	A connected graph G, a terminal set $W \subseteq V(G)$ with $|W| \geq 2$, two nonnegative integer ℓ and r, and a partition $\mathcal{W}$ into subsets of W.
Question:	Does G admit a strict connection tree T for W such that $|\mathrm{L}(T)| \leq \ell$, $|\mathrm{R}(T)| \leq r$, and, for every subset $W' \in \mathcal{W}$, the terminals belonging to W' are connected in T by a same router $\rho \in \mathrm{R}(T)$?

By a polynomial-time reduction from $1 + d$ VDP, we prove in Theorem 6 that CONSTRAINED TERMINAL PARTITION is an NP-complete problem.

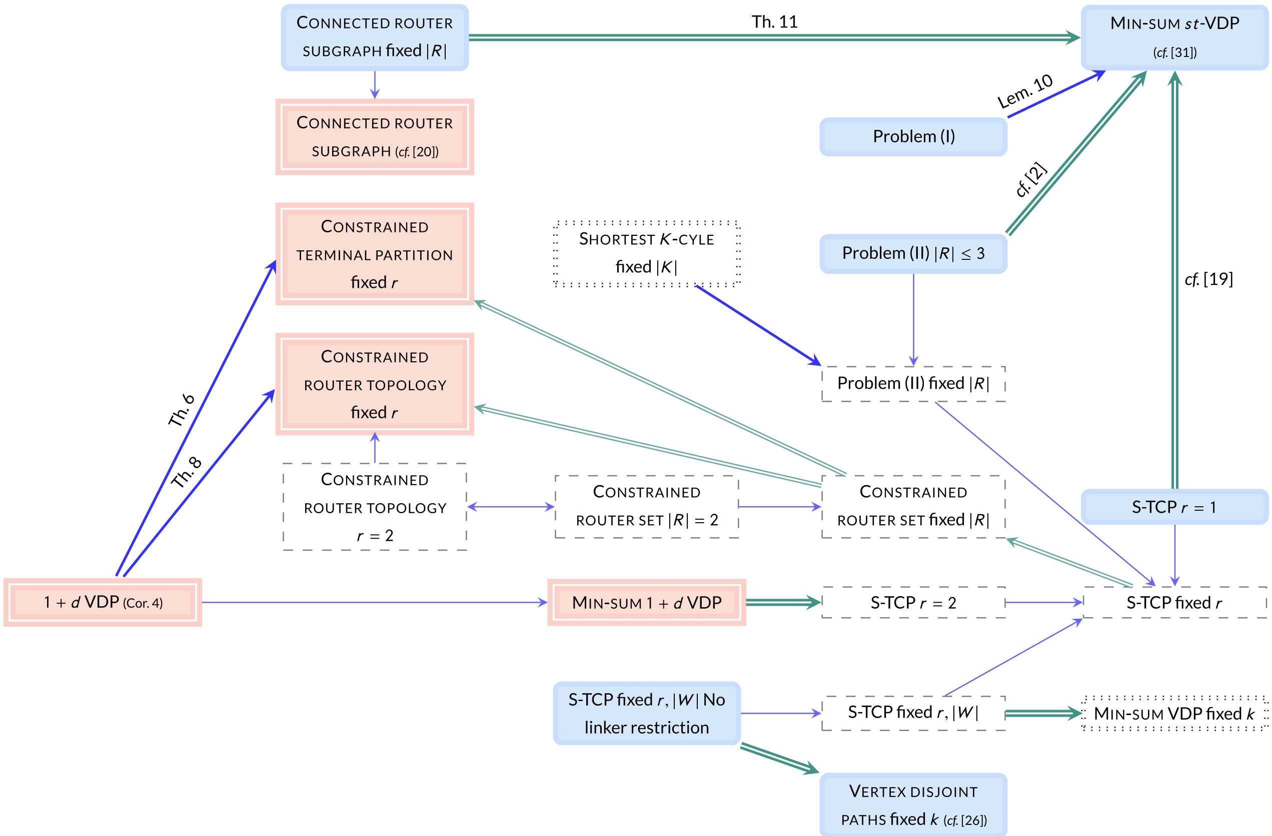


FIGURE 5 Relationship between vertex-disjoint paths problems and variants of S-TCP, where $\pi \to \pi'$ denotes that π is polynomially reducible to π', and $\pi \Rightarrow \pi'$ denotes that π is Turing reducible to π'; in particular, we use thinner lines to denote that the corresponding reduction is straightforward. Moreover, we use (blue) rounded squares to represent polynomial-time solvable problems, (pinkish-orange) double squares to represent NP-complete problems, dashed squares to represent problems whose complexity is open, and double dotted squares to represent problems whose complexity is a long-standing open question. For readability, some reductions are omitted [Color figure can be viewed at wileyonlinelibrary.com]

Construction 5. *Let $I = (G, \{s_1, t_1\}, \{s_2, t_2\}, d)$ be an instance of $1 + d$ VDP and r be an integer such that $r \geq 2$. We let $g(I, r) = (G', W, \ell, r, \mathcal{W})$ be the instance of* CONSTRAINED TERMINAL PARTITION *defined as follows.*

- *We define G' as the graph obtained from G by performing the following operations:*
  - *we let $\rho_1 = s_1$ and $\rho_2 = s_2$, create the vertices $\rho_3, \ldots, \rho_r$, and add the edges $\rho_1\rho_2, \ldots, \rho_{r-1}\rho_r$;*
  - *for each $i \in \{1, \ldots, r\}$, we create the vertices w_i^1 and w_i^2, and add the edges $w_i^1\rho_i$ and $w_i^2\rho_i$;*
  - *we replace the vertex t_2 with d twins $t_2^1, \ldots, t_2^d$ of itself.*
- *We define $W = \{w_1^1, w_1^2, \ldots, w_r^1, w_r^2\} \cup \{t_1, t_2^1, \ldots, t_2^d\}$.*
- *We define $\ell = |V(G)\backslash W| - r$.*
- *Finally, we define $\mathcal{W} = \{W_1, \ldots, W_r\}$, where $W_1 = \{w_1^1, w_1^2\} \cup \{t_1\}$, $W_2 = \{w_2^1, w_2^2\} \cup \{t_2^1, \ldots, t_2^d\}$ and, for each $i \in \{3, \ldots, r\}$, $W_i = \{w_i^1, w_i^2\}$.*

Theorem 6. *For each fixed $r \geq 2$,* CONSTRAINED TERMINAL PARTITION *is NP-complete.*

Proof. Let $I = (G, \{s_1, t_1\}, \{s_2, t_2\}, d)$ be an instance of $1 + d$ VDP and $g(I, r) = (G', W, \ell, r, \mathcal{W})$ be the instance of CONSTRAINED TERMINAL PARTITION described in Construction 4. Consider $R = \{\rho_1, \ldots, \rho_r\}$.

First, assume that there exist in G pairwise vertex-disjoint paths P_1 and $P_{2,1}, \ldots, P_{2,d}$, such that P_1 is between s_1 and t_1 and $P_{2,1}, \ldots, P_{2,d}$ are between s_2 and t_2. For each $j \in \{1, \ldots, d\}$, let $P'_{2,j}$ be the path in G' obtained from $P_{2,j}$ by replacing the vertex t_2 with its twin t_2^j. We let T be the graph with vertex set $V(T) = R \cup W \cup V(P_1) \cup \cup_{j=1}^{d} V(P'_{2,j})$ and edge set $E(T) = \{\rho_1\rho_2, \ldots, \rho_{r-1}\rho_r\} \cup \{w_1^1\rho_1, w_1^2\rho_1, \ldots, w_r^1\rho_r, w_r^2\rho_r\} \cup \cup_{i=1}^{d} E(P'_{2,i})$. One can see that T is a strict connection tree of G' for W, such that $|\mathrm{L}(T)| \leq \ell$, $|\mathrm{R}(T)| \leq r$ and, for every $i \in \{1, \ldots, r\}$, the terminals in W_i are connected in T by the router $\rho_i \in \mathrm{R}(T)$. Thus, $g(I, r)$ is a YES-instance of CONSTRAINED TERMINAL PARTITION.

Conversely, assume that G' admits a strict connection tree T for W, such that $|\mathrm{L}(T)| \leq \ell$, $|\mathrm{R}(T)| \leq r$ and, for every $i \in \{1, \ldots, r\}$, the terminals in W_i are connected in T by a same router in $\mathrm{R}(T)$. Note that, the only neighbour in G' of the terminals w_i^1 and w_i^2 is the vertex ρ_i. Thus, $\mathrm{R}(T) = R$. In addition, since $W_i \supseteq \{w_i^1, w_i^2\}$, every terminal in W_i is connected in T by ρ_i, which implies that $t_1 \in W_1$ is connected in T by ρ_1, and $t_2^1, \ldots, t_2^d \in W_2$ are connected in T by ρ_2. Let P_1' be the path in T between $s_1 = \rho_1$ and t_1. For each $j \in \{1, \ldots, d\}$, let $P_{2,j}$ be the path in G between $s_2 = \rho_2$ and t_2 obtained from $P_{2,j}'$ by simply replacing t_2^j with t_2, where $P_{2,j}'$ denotes the path in T between $s_2 = \rho_2$ and t_2^j. Since $V(P_1') \cap \mathrm{R}(T) = \{\rho_1\}$, every internal vertex of P_1' has degree 2 in T. Similarly, for each $j \in \{1, \ldots, d\}$, $V(P_{2,j}') \cap \mathrm{R}(T) = \{\rho_2\}$, which implies every internal vertex of $P_{2,j}'$ has degree 2 in T. As a result, P_1 and $P_{2,1}, \ldots, P_{2,d}$ are pairwise vertex-disjoint paths of G, and therefore I is a YES-instance of $1+d$ VDP. ■

An additional consequence obtained from the proof of Theorem 5 is that CONSTRAINED TERMINAL PARTITION remains NP-complete even if the restriction on the number of linkers in the strict connection trees is not taken into account. In fact, in Construction 4 the parameter ℓ is defined as being the largest possible.

The *labelled router topology*, or simply *router topology*, of a strict connection tree T is the unique tree $\tau(T)$ with vertex set $V(\tau(T)) = \mathrm{R}(T)$ that satisfies the following property: for each pair of distinct vertices $\rho_i, \rho_j \in \mathrm{R}(T)$, there exists the edge $\rho_i\rho_j \in E(\tau(T))$ if and only if the path P in T between ρ_i and ρ_j does not contain any other vertex belonging to $\mathrm{R}(T)$, that is, $V(P) \cap \mathrm{R}(T) = \{\rho_i, \rho_j\}$. We remark that $\tau(T)$ is a topological minor of T.

Given an instance $I = (G, W, \ell, r)$ of S-TCP, with $r \geq 2$ fixed, a reasonable strategy to decide in polynomial-time whether I is a YES-instance of the problem would consist in enumerating all possible subsets $R \subseteq V(G)\backslash W$, with $|R| \leq r$, enumerating all spanning trees H_R of $G[R]$ and, then, verifying whether G admits a strict connection tree for W with at most ℓ linkers and whose router topology is equal to H_R. Nevertheless, we prove in Theorem 7 that, unless P = NP, this strategy does not work. In order to prove such a result, we present a polynomial-time reduction from $1+d$ VDP to the variant of S-TCP called CONSTRAINED ROUTER TOPOLOGY, formally defined below.

CONSTRAINED ROUTER TOPOLOGY	
Input:	A connected graph G, a terminal set $W \subseteq V(G)$ with $\lvert W\rvert \geq 2$, a nonnegative integer ℓ, and a tree H.
Question:	Does G admit a strict connection tree T for W such that $\lvert\mathrm{L}(T)\rvert \leq \ell$ and whose router topology $\tau(T)$ is equal to H?

Construction 7. *Let $I = (G, \{s_1, t_1\}, \{s_2, t_2\}, d)$ be an instance of $1+d$ VDP and r be an integer such that $r \geq 3$. We let $g(I, r) = (G', W, \ell, H)$ be the instance of* CONSTRAINED ROUTER TOPOLOGY, *with $\lvert V(H)\rvert = r$, defined as follows.*

- *We define G' as the graph obtained from G by performing the following operations:*
  - *we let $\rho_3 = s_2$, create the vertices ρ_1, ρ_2 and $\rho_4, \ldots, \rho_r$, and add the edges $\rho_2\rho_3, \ldots, \rho_{r-1}\rho_r$;*
  - *we add the edges $\rho_1 s_1$ and $t_1\rho_2$;*
  - *for each $i \in \{1, \ldots, r\}$, we create the vertices w_i^1 and w_i^2, and add the edges $w_i^1\rho_i$ and $w_i^2\rho_i$;*
  - *we replace the vertex t_2 with d twins $t_2^1, \ldots, t_2^d$ of itself.*
- *We define $W = \{w_1^1, w_1^2, \ldots, w_r^1, w_r^2\} \cup \{t_2^1, \ldots, t_2^d\}$.*
- *We define $\ell = \lvert V(G)\backslash W\rvert - r$.*
- *Finally, we define H as the graph with vertex set $V(H) = \{\rho_1, \ldots, \rho_r\}$ and edge set $E(H) = \{\rho_1\rho_2, \ldots, \rho_{r-1}\rho_r\}$.*

Theorem 8. *For each fixed $\lvert V(H)\rvert = r \geq 3$,* CONSTRAINED ROUTER TOPOLOGY *is NP-complete.*

Proof. Let $I = (G, \{s_1, t_1\}, \{s_2, t_2\}, d)$ be an instance of $1+d$ VDP and $g(I, r) = (G', W, \ell, H)$ be the instance of CONSTRAINED ROUTER TOPOLOGY described in Construction 7. Consider $R = \{\rho_1, \ldots, \rho_r\}$.

The proof that I being a YES-instance of $1+d$ VDP implies $g(I, r)$ being a YES-instance of CONSTRAINED ROUTER TOPOLOGY is analogous to the corresponding proof, for CONSTRAINED TERMINAL PARTITION, presented in Theorem 6.

For the converse, assume that $g(I, r)$ is a YES-instance of CONSTRAINED ROUTER TOPOLOGY, and let T be a strict connection tree of G' for W, such that $\mathrm{L}(T) \leq \ell$ and whose router topology is equal to H. Since $\rho_1\rho_2 \in E(H)$, the path between ρ_1 and ρ_2 in T, say the path P_1', does not contain any other router of T. Consequently, every internal vertex of P_1' has degree 2 in T. Furthermore, note that the only nonterminal neighbour of ρ_1 (ρ_2, resp.) in G' is the vertex s_1 (t_1, resp.). Thus, since $\rho_1\rho_2 \not\in E(G)$, P_1' necessarily passes through s_1 and t_1. In addition, we obtain that neither ρ_1 nor ρ_2 connects $t_2^1, \ldots, t_2^d$ in T, otherwise there would exist an internal vertex of P_1' with degree greater than 2 in T. As a result, all the terminals $t_2^1, \ldots, t_2^d$ are necessarily connected in T by ρ_3. In other words, for each $j \in \{1, \ldots, d\}$, if $P_{2,j}'$ is the path in T between ρ_3 and t_2^j, then $V(P_2') \cap \mathrm{R}(T) = \{\rho_3\}$, which implies that every internal vertex of $P_{2,j}'$ has degree 2 in T. Therefore, I is a YES-instance of $1+d$ VDP. Indeed, let P_1 be the path in G between s_1 and t_1 obtained from P_1' by removing ρ_1 and ρ_2, and for each $j \in \{1, \ldots, d\}$, let $P_{2,j}$ be the path in G between $s_2 = \rho_3$ and t_2 obtained from $P_{2,j}'$ by simply replacing t_2^j with t_2. Then, one can readily verify that P_1 and $P_{2,1}, \ldots, P_{2,d}$ are pairwise vertex-disjoint paths of G, which certifies that I is a YES-instance of $1+d$ VDP. ■

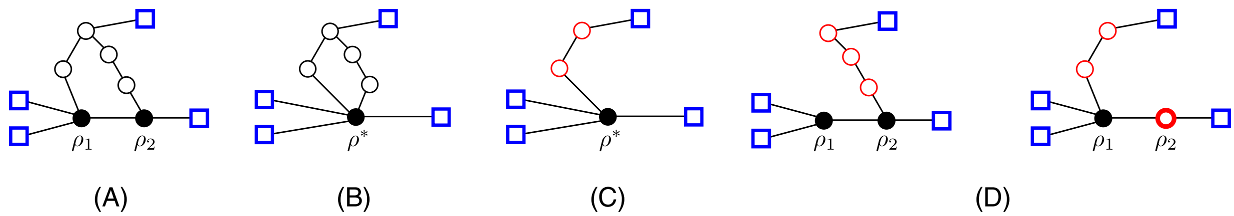


FIGURE 6 (A) Instance $I = (G, W, \ell, R')$, where $\ell = 2$ and $R' = \{\rho_1, \rho_2\}$; (B) instance $I_{R'} = (G_{R'}, W, \ell, R = \{\rho^*\})$; (C) A YES-certificate of $I_{R'}$ for the CONSTRAINED ROUTER SET problem; (D) the only two strict connection trees T of G for W, such that $\mathrm{R}(T) \subseteq R'$ [Color figure can be viewed at wileyonlinelibrary.com]

Similarly to CONSTRAINED TERMINAL PARTITION, CONSTRAINED ROUTER TOPOLOGY remains NP-complete even if the restriction on the number of linkers in the strict connection trees is not considered. Furthermore, it is worth mentioning that the proof that we presented for Theorem 8 actually does not depend on the constructed topology H itself, but it only depends on the facts that the target strict connection tree T might have three distinct routers and, for two specific nonterminal vertices $\rho_1, \rho_2 \in V(G)\backslash W$, $\rho_1\rho_2$ must be an edge of $\tau(T)$. Consequently, through a slight change of Construction 7, we also obtain as a by-product of the proof of Theorem 7 that the following simpler problem is still NP-complete: given graph G, a terminal set $W \subseteq V(G)$ with $|W| \geq 2$, a set $R \subseteq V(G)\backslash W$ with fixed $|R| \geq 3$, and two specified vertices $\rho_1, \rho_2 \in R$, decide whether G admits a strict connection tree T for W such that $\mathrm{R}(T) \subseteq R$ and $\rho_1\rho_2 \in E(\tau(T))$. In addition, we remark that the proof of Theorem 7 can be also used to show the NP-completeness of the variant of the problem that further asks for $T[\mathrm{R}(T)] + \rho_1\rho_2$ being a connected graph. Note that, requiring the connectedness of $T[\mathrm{R}(T)]$ is equivalent to requiring the equality $\tau(T) = T[\mathrm{R}(T)]$. Hence, $\rho_1\rho_2 \in E(\tau(T))$ and $T[\mathrm{R}(T)] + \rho_1\rho_2$ is a connected graph if and only if $\tau(T) = T[\mathrm{R}(T)] + \rho_1\rho_2$.

Now, we analyse the variant of S-TCP called CONNECTED ROUTER SUBGRAPH, which has the same input of CONSTRAINED ROUTER SET but asks for the existence of strict connection trees T that additionally satisfy the condition of having their *router subgraph* $T[\mathrm{R}(T)]$ connected, that is, $\tau(T) = T[\mathrm{R}(T)]$. It follows from the observations described in the previous paragraph that if, for some especial pair of vertices $\rho_1, \rho_2 \in R$, the existence of the single edge $\rho_1\rho_2 \in E(\tau(T))\backslash E(T)$ is allowed, then the corresponding problem becomes NP-complete. Contrasting with this fact, we prove that CONNECTED ROUTER SUBGRAPH is polynomial-time solvable for each fixed $|R| \geq 1$, although it can be shown to be NP-complete if $|R|$ is not fixed (e.g., see Theorem 1 in [20]). Below, we present a formal definition for the problem.

CONNECTED ROUTER SUBGRAPH	
Input:	A connected graph G, a terminal set $W \subseteq V(G)$ with $|W| \geq 2$, a nonnegative integer ℓ and a nonempty set $R \subseteq V(G)\backslash W$.
Question:	Does G admit a strict connection tree T for W such that $|\mathrm{L}(T)| \leq \ell$, $\mathrm{R}(T) \subseteq R$ and $T[\mathrm{R}(T)]$ is a connected subgraph?

Since CONSTRAINED ROUTER SET is polynomial-time solvable when $|R| = 1$ cf. [19], one could try to decide whether an arbitrary instance $I = (G, W, \ell, R)$, with $|R| \geq 1$ fixed, is a YES-instance of CONNECTED ROUTER SUBGRAPH by performing the following operations, for each nonempty subset $R' \subseteq R$:

- Construct the graph $G_{R'}$ obtained from G by contracting all vertices belonging to R' into a unique vertex ρ^*;
- Return that I is a YES-instance of S-TCP if $(G_{R'}, W, \ell, R = \{\rho^*\})$ is a YES-instance of CONSTRAINED ROUTER SET.

Despite reasonable at first glance, especially when $|R'| = 2$, the approach described above does not work. Possibly, $I_{R'}$ is a YES-instance of CONSTRAINED ROUTER SET whereas I is a NO-instance of CONNECTED ROUTER SUBGRAPH. As a matter of fact, Figure 6 exemplifies such a case.

By a Turing reduction to MIN-SUM *st*-VDP, we prove in Theorem 11 the polynomial tractability of CONNECTED ROUTER SUBGRAPH when $|R| \geq 2$ is fixed.

Construction 9. *Let $I = (G, W, \ell, R)$ be an instance of* CONNECTED ROUTER SUBGRAPH, *R' be a subset of R such that $|R'| \geq 2$ and $G[R']$ is a connected graph, and let $T_{R'}$ be a spanning tree of $G[R']$. We let $g(I, T_{R'}) = (G', \{s, t\}, d, x)$ be the instance of* MIN-SUM *st*-VDP *defined as follows. Consider $R'_1 = \{\rho_i \in R' | d_{T_{R'}}(\rho_i) = 1\}$, $R'_2 = \{\rho_i \in R' | d_{T_{R'}}(\rho_i) = 2\}$ and $q = |W| - (2|R'|_1 + |R'_2|)$. If $q \leq 0$, then clearly G does not admit a strict connection tree T for W such that $\mathrm{R}(T) = R'$ and $\tau(T) = T_{R'}$. Thus, assume without loss of generality that $q \geq 0$.*

- *We define G' as the graph obtained from G by performing the following operations:*
  - *we create the vertices s and t and remove all the edges incident to the vertices belonging to R';*
  - *for each $\rho_i \in R'_1 \cup R'_2$, we let $\rho_i^1 = \rho_i$, and we create the vertex ρ_i^2 if $\rho_i \in R'_1$; moreover, for each $j \in \{1, 3 - d_{T_{R'}}(\rho_i)\}$, we add the edges $s\rho_i^j$ and $\rho_i^j v$ for each $v \in N_G(\rho_i)\backslash R'$;*
  - *if $q \geq 1$, then, for each $j \in \{1, \ldots, q\}$, we create the vertex ρ_*^j and add the edges $s\rho_*^j$ and $\rho_*^j v$ for each $v \in N_G(R')\backslash R'$;*

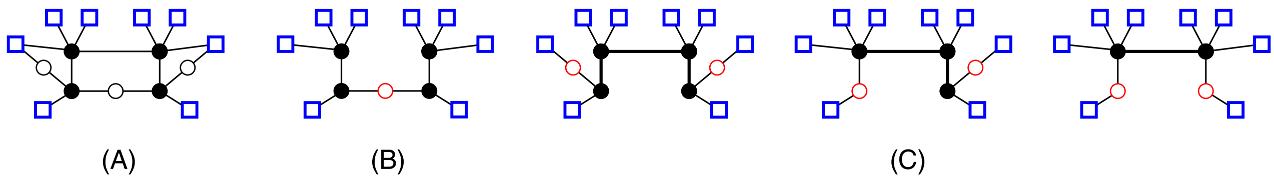


FIGURE 7 (A) Instance $I = (G, W, \ell = 1, R)$; (B) YES-certificate of I for the CONSTRAINED ROUTER SET problem; (C) the only three (nonsymmetrical) strict connection trees T of G for W, such that $\mathrm{R}(T) \subseteq R$ and $T[\mathrm{R}(T)]$ is connected [Color figure can be viewed at wileyonlinelibrary.com]

  - *for each $w \in W$, we add the edge wt.*
- *Finally, we define $d = |W|$ and $x = \ell + 3d$.*

Lemma 10. *$I = (G, W, \ell, R)$ is a* YES*-instance of* CONNECTED ROUTER SUBGRAPH *if and only if there exists a nonempty subset $R' \subseteq R$ and a spanning tree $T_{R'}$ of $G[R']$ such that $g(I, T_{R'})$ is a* YES*-instance of* MIN-SUM *st*-VDP, *where $g(I, T_{R'})$ denotes the instance described in Construction 9.*

Proof. First, assume that there exists a strict connection tree T of G for W such that $|\mathrm{L}(T)| \leq \ell$, $\mathrm{R}(T) \subseteq R$ and $T[\mathrm{R}(T)]$ is connected. Assume that $|\mathrm{R}(T)| \geq 2$. Let $R' = \mathrm{R}(T)$ and $T_{R'} = T[R']$. Consider $R'_1 = \{\rho_i \in R' | d_{T_{R'}}(\rho_i) = 1\}$ and $R'_2 = \{\rho_i \in R' | d_{T_{R'}}(\rho_i) = 2\}$. For each router $\rho_i \in R'$, let $Q_i = \emptyset$ if $\rho_i \not\in R'_1 \cup R'_2$, $Q_i = \{\rho_i^1, \rho_i^2\}$ if $\rho_i \in R'_1$ and $Q_i = \{\rho_i^1\}$ if $\rho_i \in R'_2$. Additionally, let $Q_* = \{\rho_*^1, \rho_*^2, \ldots, \rho_*^q\}$, where $q = |W| - (2|R'_1| + |R'_2|)$. By definition $|Q_1 \cup \cdots \cup Q_{|R'|} \cup Q_*| = |W|$. Moreover, note that, for $i \in \{1, 2\}$, each router belonging to R'_i connects at least $3 - i$ terminals in T. Thus, there exists a bijection $\alpha : W \to Q_1 \cup \cdots \cup Q_{|R'|} \cup Q_*$ such that, for each $w \in W$, if $\alpha(w) = \rho_i^j \in Q_i$ for some $j \in \{1, |Q_i|\}$ and some $\rho_i \in R'_1 \cup R'_2$, then the terminal w is connected in T by ρ_i. For each terminal $w \in W$, let P_w be the path in T between w and its connecting router $\rho_i \in R'$, and let P'_w be the path obtained from P_w by adding s and t at the beginning and at the end, respectively, and replacing ρ_i with $\alpha(w) \in Q_i \cup Q_*$. One can verify that, if $W = \{w_1, \ldots, w_d\}$, then $P'_{w_1}, \ldots, P'_{w_d}$ are d pairwise vertex-disjoint paths, between s and t, such that the sum of their lengths is at most $|\mathrm{L}(T)| + 3d \leq \ell + 3d$.

Conversely, assume that there exists a nonempty subset $R' \subseteq R$ and a spanning tree $T_{R'}$ of $G[R']$ such that $g(I, T_{R'}) = (G', \{s, t\}, d, x)$ is a YES-instance of MIN-SUM *st*-VDP. Let $P'_1, \ldots, P'_d$ be d pairwise vertex-disjoint paths, between s and t, such that the sum of their lengths is at most $x = \ell + 3d$. Note that, for each $h \in \{1, \ldots, d\}$, the path P'_h contains either a vertex ρ_i^j, where $\rho_i \in R'$ and $j \in \{1, 2\}$, or a vertex ρ_*^j, where $j \in \{1, \ldots, d\}$. If P'_h contains a vertex ρ_i^j, then let P_h be the path obtained from P'_h by removing s and t and replacing ρ_i^j with ρ_i. On the other hand, if P'_h contains a vertex ρ_*^j, then let P_h be the path obtained from P'_h by removing s and t and replacing ρ_*^j with an arbitrary vertex $\rho \in R' \cap N_G(v)$, where v denotes the vertex that immediately follows ρ_*^j in P'_h. Let T be the graph with vertex set $V(T) = \cup_{h=1}^{d} V(P_h)$ and edge set $E(T) = E(T_{R'}) \cup \cup_{h=1}^{d} E(P_h)$. One can verify that T is a strict connection tree of G for W such that $|\mathrm{L}(T)| \leq \ell$, $\mathrm{R}(T) = R' \subseteq R$ and $T[\mathrm{R}(T)] = T_{R'}$ is connected. ■

Theorem 11. *For each fixed $|R| \geq 2$,* CONNECTED ROUTER SUBGRAPH *can be solved in polynomial time.*

Proof. Let $I = (G, W, \ell, R)$ be an instance of CONNECTED ROUTER SUBGRAPH. For each nonempty subset $R' \subseteq R$, such that $|R'| \geq 2$ and $G[R']$ is connected, and for each spanning tree $T_{R'}$ of $G[R']$, construct the instance $g(I, T_{R'})$ of MIN-SUM *st*-VDP described in Construction 9. Then, decide whether $g(I, T_{R'})$ is a YES-instance by using a polynomial time algorithm for MIN-SUM *st*-VDP, such as Suurballe's algorithm [31]. If $g(I, T_{R'})$ is a YES-instance of MIN-SUM *st*-VDP for some R' and some $T_{R'}$, then return that I is a YES-instance of CONNECTED ROUTER SUBGRAPH. Otherwise, return that I is a NO-instance of CONNECTED ROUTER SUBGRAPH. The correctness of this procedure follows directly from Lemma 10. Regarding its running time, since $|R|$ is fixed, the numbers of subsets R' and spanning trees $T_{R'}$ to be considered are constant. Furthermore, note that, for every nonempty subset $R' \subseteq R$ and every spanning tree $T_{R'}$ of $G[R']$, the instance $g(I, T_{R'})$ can be constructed (and solved) in time polynomial in the size of I. ■

As a by-product of the proof of Theorem 11, we obtain that CONSTRAINED ROUTER SET is polynomial-time solvable when restricted to instances I in which $1 \leq |R| \leq 3$ and $G[R]$ is connected. Indeed, through simple analyses one can verify that, in this restricted case, I is a YES-instance of CONSTRAINED ROUTER SET if and only if I is a YES-instance of CONNECTED ROUTER SUBGRAPH. On the other hand, this remark is not extensible for instances of CONSTRAINED ROUTER SET in which $|R| \geq 4$ and $G[R]$ is connected. Figure 7 exemplifies such an instance $I = (G, W, \ell, R)$ that is a YES-instance of CONSTRAINED ROUTER SET but is a NO-instance of CONNECTED ROUTER SUBGRAPH.

5 | CONCLUDING REMARKS

The main result of this work consists in establishing that SIMPLE UNDIRECTED 2-COMMODITY INTEGRAL FLOW (SIMPLE U2CIF) is still NP-complete if the demand of exactly one commodity is unitary—equivalently, if the demand of exactly one commodity is arbitrary large—, closing a 40-year complexity gap. As a consequence of our proof, we obtain that, for every fixed k, SIMPLE UNDIRECTED k-COMMODITY INTEGRAL FLOW remains NP-complete if there exists exactly one commodity with unbounded demand, and all the others have unitary demands. The theoretical importance of this result lies in the fact that it completes the *P* vs *NP-hard dichotomy* on the complexity of simple integral flow problems, on general graphs, with respect to the number of commodities and their respective demands, as properly shown in Table 1. In addition, our result emphasises that, among the variety of possibilities, there are interesting questions related to integral flow and disjoint paths that may not even have been realised cf. [24]. Indeed, the comprehensive survey by Naves and Sebő presents a table outlining the complexity of 189 interesting special cases of integral multicommodity flow and disjoint paths problems, of which 21 remain unsettled, and our unitary cost distinction represents a possible extra row in the table, as a subcase of two commodities with demands encoded in unary.

As an open problem, we intend to investigate the complexity of SIMPLE U2CIF when the input graph is constrained to be planar. Despite the several results in the literature with respect to multicommodity integral flow and disjoint paths problems on planar graphs [8,15,22,23,29,32], as far as we were able to verify, none of them answer the question about the complexity of the particular case of SIMPLE U2CIF with exactly one unitary demand on planar graphs. On the other hand, it is worth mentioning that $1 + d$ VDP on planar graphs can be solved in polynomial-time [27]. For a thorough reference on this topic, we refer to Schrijver's book [28], in addition to Naves and Sebő's survey [24].

Another interesting question is about the complexity of the MIN-SUM VERTEX-DISJOINT PATHS problem when the number of required paths k is fixed. For $k = 2$, Björklund and Husfeldt proved that the problem admits a Monte Carlo polynomial-time algorithm [1]. Nevertheless, even for $k = 2$, determining whether MIN-SUM VERTEX-DISJOINT PATHS is in P still remains a challenging open problem [1,8,10,15].

Finally, with respect to the STRICT TERMINAL CONNECTION problem (S-TCP), the main open question that we leave in this work consists in determining the complexity of the problem when $r \geq 2$ is fixed. It is important mentioning that, although it is well-known that STEINER TREE can be solved in polynomial-time when $|W|$ is fixed [6], the complexity of S-TCP in this even more restricted case also remains unknown.

ACKNOWLEDGMENTS

The authors thank András Sebő for valuable discussions and suggestions on this work, especially about multicommodity integral flow problems, on the occasion of IX Latin and American Algorithms, Graphs and Optimization Symposium. The authors also thank anonymous referees for diligent reading and useful suggestions.

ORCID

Alexsander A. de Melo https://orcid.org/0000-0001-5268-6997
Celina M. H. de Figueiredo https://orcid.org/0000-0002-6393-0876
Uéverton dos Santos Souza https://orcid.org/0000-0002-5320-9209

REFERENCES

[1] A. Björklund and T. Husfeldt, *Shortest two disjoint paths in polynomial time*, *Automata, Languages, and Programming*, Lecture Notes in Computer Science, Berlin, Heidelberg: Springer; **8572** (2014), pp. 211–222.
[2] A. Björklund, T. Husfeldt, and N. Taslaman, "*Shortest cycle through specified elements*," *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, ACM, New York, (2012), pp. 1747–1753.
[3] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms*, The MIT Press, 3rd., Cambridge, MA, 2009.
[4] M.C. Dourado, R.A. Oliveira, F. Protti, and U.S. Souza, "Design of connection networks with bounded number of non-terminal vertices," *Proceedings of V Latin-American Workshop on Cliques in Graphs.* Matemática Contemporânea, vol. **42**. SBM, Buenos Aires, 2014, pp. 39–47.
[5] M.C. Dourado, R.A. Oliveira, F. Protti, and U.S. Souza, "Conexão de terminais com número restrito de roteadores e elos," *Proceedings of XLVI Simpósio brasileiro de pesquisa operacional*, Salvador, Bahia: SOBRAPO, (2014), pp. 2965–2976.
[6] S.E. Dreyfus and R.A. Wagner, *The Steiner problem in graphs*, Networks **1** (1971), 195–207.
[7] J. Edmonds and R.M. Karp, *Theoretical improvements in algorithmic efficiency for network flow problems*, J. ACM **19** (1972), 248–264.
[8] T. Eilam-Tzoreff, *The disjoint shortest paths problem*, Discr. Appl. Math. **85** (1998), 113–138.
[9] S. Even, A. Itai, and A. Shamir, *On the complexity of timetable and multicommodity flow problems*, SIAM J. Comput. **5** (1976), 691–703.
[10] T. Fenner, O. Lachish, and A. Popa, *Min-sum 2-paths problems*, Theory Comput. Syst. **58** (2016), 94–110.
[11] S. Fortune, J. Hopcroft, and J. Wyllie, *The directed subgraph homeomorphism problem*, Theor. Comput. Sci. **10** (1980), 111–121.
[12] F.K. Hwang, D.S. Richards, and P. Winter, *The Steiner Tree Problem*, Annals of Discrete Mathematics, vol. **53**, Elsevier, 1st., Amsterdam, 1992.

[13] R.M. Karp, *On the computational complexity of combinatorial problems*, Networks **5** (1975), 45–68.
[14] K.i. Kawarabayashi, Y. Kobayashi, and B. Reed, *The disjoint paths problem in quadratic time*, J. Combin. Theory Ser. B **102** (2012), 424–435.
[15] Y. Kobayashi and C. Sommer, *On shortest disjoint paths in planar graphs*, Discr. Optim. **7** (2010), 234–245.
[16] B. Korte, L. Lovász, H.J. Prömel, and A. Schrijver, "Paths, flows, and VLSI-layout," *Algorithms and Combinatorics*, Springer-Verlag, Berlin, 1990.
[17] G. Lin and G. Xue, *On the terminal Steiner tree problem*, Informat Process. Lett. **84** (2002), 103–107.
[18] C.L. Lu, C.Y. Tang, and R.C.T. Lee, *The full Steiner tree problem*, Theor. Comput. Sci. **306** (2003), 55–67.
[19] A.A. Melo, C.M.H. Figueiredo, and U.S. Souza, "Connecting terminals using at most one router," *Proceedings of VII Latin-American Workshop on Cliques in Graphs,* Matemática Contemporânea, vol. **45**. SBM, Buenos Aires, 2017, pp. 49–57.
[20] A.A. Melo, C.M.H. Figueiredo, and U.S. Souza, *A multivariate analysis of the strict terminal connection problem*, J. Comput. Syst. Sci. **111** (2020), 22–41.
[21] A.A. Melo, C.M.H. Figueiredo, and U.S. Souza, "Simple undirected two-commodity integral flow with a unitary demand," *Proceedings of IX Latin and American Algorithms, Graphs and Optimization,* Electronic Notes in Discrete Mathematics, vol. **62**. Elsevier, Amsterdam, 2017, pp. 279–284.
[22] M. Middendorf and F. Pfeiffer, *On the complexity of the disjoint paths problem*, Combinatorica **13** (1993), 97–107.
[23] G. Naves, *The hardness of routing two pairs on one face*, Math. Program. **131** (2012), 49–69.
[24] G. Naves and A. Sebő, "Multiflow feasibility: An annotated tableau," *Research Trends in Combinatorial Optimization*, Springer, Berlin, 2009, pp. 261–283. http://assert-false.net/callcc/Guyslain/Works/tableau.
[25] T. Nishizeki, J. Vygen, and X. Zhou, *The edge-disjoint paths problem is NP-complete for series–parallel graphs*, Discr. Appl. Math. **115** (2001), 177–186.
[26] N. Robertson and P.D. Seymour, *Graph minors. Xiii. The disjoint paths problem*, J. Combin. Theory Ser. B **63** (1995), 65–110.
[27] A. Schrijver, "*Homotopic routing methods*," *Paths, flows, and VLSI-Layout*, B. Korte, L. Lovász, H.J. Prömel, and A. Schrijver (eds), Springer-Verlag, Berlin, 1990, pp. 329–371.
[28] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*. Algorithms and Combinatorics, vol. **24**, Berlin, Heidelberg, Springer-Verlag, 2003.
[29] A. Sebő, *Integer plane multiflows with a fixed number of demands*, J. Combin. Theory Ser. B **59** (1993), 163–171.
[30] A. Srinivas and E. Modiano, *Finding minimum energy disjoint paths in wireless ad-hoc networks*, Wireless Netw. **11** (2005), 401–417.
[31] J.W. Suurballe, *Disjoint paths in a network*, Networks **4** (1974), 125–145.
[32] J. Vygen, *NP-completeness of some edge-disjoint paths problems*, Discr. Appl. Math. **61** (1995), 83–90.
[33] M. Wahlström, "Abusing the Tutte matrix: An algebraic instance compression for the K-set-cycle problem," *30th International Symposium on Theoretical Aspects of Computer Science,* Leibniz International Proceedings in Informatics (LIPIcs), vol. **20**. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, 2013, pp. 341–352.
[34] D. Watel, M.A. Weisser, C. Bentz, and D. Barth, "Steiner problems with limited number of branching nodes," *Proceedings of 20th International Colloquium on Structural Information and Communication Complexity,* Lecture Notes in Computer Science, vol. **8179**. Springer, Cham, 2013, pp. 310–321.
[35] D. Watel, M.A. Weisser, C. Bentz, and D. Barth, *Directed Steiner trees with diffusion costs*, J. Combin. Optim. **32** (2016), 1089–1106.

How to cite this article: de Melo AA, de Figueiredo CMH, Souza US. On undirected two-commodity integral flow, disjoint paths and strict terminal connection problems. *Networks.* 2021;77:559–571. https://doi.org/10.1002/net.21976