

A Unifying Model for Locally Constrained Spanning Tree Problems

Luiz Alberto Do Carmo Viana, Manoel Campêlo, Ignasi Sau Valls, Ana Silva

► **To cite this version:**

Luiz Alberto Do Carmo Viana, Manoel Campêlo, Ignasi Sau Valls, Ana Silva. A Unifying Model for Locally Constrained Spanning Tree Problems. *Journal of Combinatorial Optimization*, Springer Verlag, 2020, 42 (1), pp.125-150. 10.1007/s10878-021-00740-2 . lirmm-03374555

HAL Id: lirmm-03374555

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-03374555>

Submitted on 12 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Unifying Model for Locally Constrained Spanning Tree Problems

Luiz Alberto do Carmo Viana¹, Manoel Campêlo², Ignasi Sau³, and Ana Silva⁴

¹Campus de Crateús, Universidade Federal do Ceará, Crateús, Brazil

²Dep. de Estatística e Matemática Aplicada, Universidade Federal do Ceará, Fortaleza, Brazil

³LIRMM, Université de Montpellier, CNRS, Montpellier, France

⁴Departamento de Matemática, Universidade Federal do Ceará, Fortaleza, Brazil

luizalberto@crateus.ufc.br, mcampelo@lia.ufc.br, ignasi.sau@lirmm.fr,
anasilva@mat.ufc.br

May 22, 2020

Abstract

Given a graph G and a digraph D whose vertices are the edges of G , we investigate the problem of finding a spanning tree of G that satisfies the constraints imposed by D . The restrictions to add an edge in the tree depend on its neighborhood in D . Here, we generalize previously investigated problems by also considering as input functions ℓ and u on $E(G)$ that give a lower and an upper bound, respectively, on the number of constraints that must be satisfied by each edge. The produced feasibility problem is denoted by **G-DCST**, while the optimization problem is denoted by **G-DCMST**. We show that **G-DCST** is NP-complete even under strong assumptions on the structures of G and D , as well as on functions ℓ and u . On the positive side, we prove two polynomial results, one for **G-DCST** and another for **G-DCMST**, and also give a simple exponential-time algorithm along with a proof that it is asymptotically optimal under the ETH. Finally, we prove that other previously studied constrained spanning tree (CST) problems can be modeled within our framework, namely, the **CONFLICT CST**, the **FORCING CST**, the **AT LEAST ONE/ALL DEPENDENCY CST**, the **MAXIMUM DEGREE CST**, the **MINIMUM DEGREE CST**, and the **FIXED-LEAVES MINIMUM DEGREE CST**.

1 Introduction

Let G be a graph and D be a (directed or undirected) graph whose vertices are the edges of G . In other terms, D defines a relation on the edge set of G . The *dependencies* of an edge $e \in E(G)$ are given by its (in-)neighborhood in D , i.e., by the set $\text{dep}_D(e) = \{e' \in$

$E(G) \mid (e', e) \in E(D)\}$. We omit the subscript D in $\text{dep}_D(e)$ whenever the dependency graph is clear from the context.

Many problems have been investigated under the light of dependencies between pairs of objects, such as the knapsack problem [26], bin-packing [24], maximum flow [40], scheduling problems [9], maximum matchings [19], shortest paths [19], or maximum acyclic subgraphs [36]. Generally, the dependency problems defined on graphs can be described as the problem of finding a subgraph H of G satisfying the dependency constraints imposed by D .

However, the notion of dependency itself may vary. For example, every (e, e') in D could mean that, whenever e' is chosen (not chosen), we get that e cannot (must) be chosen, thus expressing a *conflict constraint* (*forcing constraint*), with D being the *conflict graph* (*forcing graph*). In this paper, we introduce a generalization of dependency constrained problems, and investigate this generalization for spanning trees. In particular, our model generalizes many of the constrained spanning tree problems that have been investigated in the literature.

Our contribution. For the generalized version of dependency constrained problems, together with graph G and (di)graph D , we also consider functions ℓ and u that assign, to each $e \in E(G)$, a lower and an upper bound on the number of dependencies that must be ensured for e . This means that a subgraph $H \subseteq G$ satisfies the imposed constraints if and only if the number of edges in $E(H) \cap \text{dep}(e)$ is at least $\ell(e)$ and at most $u(e)$, for every $e \in E(G)$ ¹; we then say that H (ℓ, u) -satisfies D . When H is asked to be a spanning tree, we call the problem the GENERALIZED DEPENDENCY CONSTRAINED SPANNING TREE problem, and denote it by **G-DCST**. Also, sometimes we deal with the related optimization problem by considering weights on the edges of G ; this is called the GENERALIZED DEPENDENCY CONSTRAINED MINIMUM SPANNING TREE problem, and is denoted by **G-DCMST**. Let us observe that when $\ell(e) = 0$ and $u(e) \geq |\text{dep}(e)|$, for all $e \in E(G)$, then **G-DCST** is equivalent to deciding whether G is connected, and **G-DCMST** corresponds to the classical MINIMUM SPANNING TREE problem.

Clearly, the feasibility problem **G-DCST** is a particular case of the optimization problem **G-DCMST**, where the weight of each edge is equal to one. This is why, whenever possible, we give preference to prove NP-completeness results for the feasibility problem, and get polynomial results for the optimization problem. We use reductions from $(3, 2, 2)$ -SAT to prove our main NP-completeness results, whereas our polynomial results arise as consequences of the Matroid Intersection Theorem [23] (cf. Section 2).

Considering the generalized version of the spanning tree problem, given a graph G , a digraph $D = (E(G), A)$, and functions ℓ, u , we prove that deciding whether G has a spanning tree that (ℓ, u) -satisfies D is NP-complete in the following cases:

¹We can always assume that $0 \leq \ell(e) \leq u(e) \leq |\text{dep}(e)|$, and so $\ell(e) = u(e) = 0$ if $\text{dep}(e) = \emptyset$.

- (i) $\ell(e) = u(e) = |\text{dep}(e)|$ for every $e \in E(G)$, D is a forest of oriented paths of length at most two where all components are directed paths, out-stars, or in-stars, and G is an outerplanar chordal graph with diameter at most two. Furthermore, this problem cannot be solved in time $2^{o(n+m)}$ unless the ETH fails, where $n = |V(G)|$ and $m = |E(G)|$;
- (ii) When ℓ, u are constant functions, for every pair of constant values such that $\ell \leq u$;
- (iii) When $\ell(e) = 0$, and $u(e) = |\text{dep}(e)| - 1$, for every $e \in E(G)$.

On the positive side, we prove the following:

- (a) **G-DCST** can be solved in polynomial time when D is an oriented matching, and $\ell(e) = u(e) = |\text{dep}(e)|$ for every $e \in E(G)$;
- (b) **G-DCMST** can be solved in polynomial time when $\ell = 0$, D is a collection of symmetric complete digraphs D_1, \dots, D_k , and $u(e) = u(e')$ whenever e, e' are within the same component of D ;
- (c) **G-DCST** can be solved in time $O(2^m \cdot (n + m))$, where $n = |V(G)|$ and $m = |E(G)|$.

It is worth observing that (a) and (i) define a dichotomy between polynomial and hard cases for **G-DCST** when regarding D as a family of oriented paths. It can be solved in polynomial time if length of the longest path in the underlying graph of D is at most one, and it is NP-Complete otherwise.

We also prove that many of the constrained spanning tree (CST) problems that have been investigated in the literature can be modeled with our general problem, namely the CONFLICT CST [18], the FORCING CST [19], the AT LEAST ONE/ALL DEPENDENCY CST [45], the MAXIMUM DEGREE CST [20], the MINIMUM DEGREE CST [2], and the FIXED-LEAVES MINIMUM DEGREE CST [22]. All our reductions preserve the value of the solutions, which means that also the optimization version of these problems can be modeled within our framework.

Notice that the previously mentioned CST problems impose (vertex-wise or edge-wise) local constraints to describe their set of feasible spanning trees. This contrasts with MAXIMUM DIAMETER CST [10,11], MINIMUM DIAMETER CST [27,28] (with variations [7,32]) and MAXIMUM LEAVES CST [25,35], examples of NP-hard problems that impose constraints on global tree parameters. In [21], the authors propose an approach that includes also these global constraints, but from a practical point of view.

Related work. In what follows, we talk sometimes about the feasibility version of the problems, and sometimes about the optimization version, where also a weight function on the edges of the input graph is given. Also, when ℓ and/or u are constant functions, we write directly the constant value inside the parenthesis when saying whether a spanning tree (ℓ, u) -satisfies D .

Conflict constraints: Recall that, in the CONFLICT CONSTRAINED (MINIMUM) SPANNING TREE problem, we are given a pair of graphs G and D such that $V(D) = E(G)$, and we want to know whether there exists a spanning tree (find a minimum spanning tree) T of G such that $E(T) \cap \text{dep}(e) = \emptyset$ for every $e \in E(T)$. We denote the feasibility problem by CCST and the optimization problem by CCMST. Note that, if we consider D' as an arbitrary orientation of D (i.e., each edge e_1e_2 in D gives rise to either (e_1, e_2) or (e_2, e_1) in D'), then we get that such a tree exists if and only if there exists a spanning tree T that $(0, 0)$ -satisfies D' . This means that our problem generalizes this one and therefore inherits the NP-complete results, as well as might help with some polynomial cases. Also, observe that the problems related to results (ii) when $\ell = 0$, (iii) and (b) can be seen as generalizations of the conflict constrained problems in the sense that $\ell = 0$ (i.e., no lower bound constraint is imposed), but $u \neq 0$.

Problems CCST and CCMST have been introduced in [19], where CCMST is proved to be polynomial-time solvable if the conflict graph is a matching, and CCST is proved to be NP-complete if the conflict graph is a forest of paths of length at most two. From what is said previously, we then get that $\text{G-DCMST}(G, D, 0, 0, w)$ is polynomial when D is an oriented matching, and $\text{G-DCST}(G, D, 0, 0)$ is NP-complete when D is an orientation of a forest of paths of length at most two. When D is one of these digraphs, observe that $\Delta^-(D) \leq 2$. If $\Delta^-(D) < 2$, the other possibilities for constant values² of ℓ, u are $\ell = 0$ and $u = 1$, which is trivially polynomial, and $\ell(e) = u(e) = |\text{dep}(e)|$ for every $e \in E(G)$. For the latter case, we have results (i) and (a), which leaves open only the complexity of the optimization problem when D is an oriented matching. On the other hand, for $\Delta^-(D) = 2$, i.e., D contains a forest of in-stars with at most two leaves, result (i) shows NP-completeness when $\ell(e) = u(e) = |\text{dep}(e)|$ for every $e \in E(G)$, the other values of ℓ and u remaining open.

The CCST and CCMST problems have also been investigated in [47], where the authors prove that, if the input graph G is a cactus, then CCST is polynomial, while CCMST is still NP-hard. They further show that the optimization problem is polynomial if the conflict graph D can be turned into a collection of cliques by the removal of a constant number of vertices, i.e., there exists a subset $E' \subseteq E(G) = V(D)$ such that $D - E'$ is a collection of cliques, and $|E'|$ is bounded by a constant. We prove something similar here for the generalized problem (result (b)).

In [31], the authors investigate a conflict constrained problem where the conflict graph is only allowed to contain an edge ee' if e and e' share an endpoint in G (they called these *forbidden transitions*). Among other results, they prove that the feasibility problem is NP-complete even if the input graph G is a complete graph. Practical approaches to the conflict constrained problem have been presented in [13, 42, 47].

Another interesting, recently defined, problem that can be modeled as a conflict con-

²This means that $\ell(e) = k$ and $u(e) = k' \geq k$, for all $e \in E(G)$ such that $\text{dep}(e) \neq \emptyset$, and $\ell(e) = u(e) = 0$ if $\text{dep}(e) = \emptyset$.

strained spanning tree problem (and therefore, as a special case of **G-DCST**) is the so-called **ANGULAR CONSTRAINED SPANNING TREE** problem [6]. In this problem, we are given a set V of points on the plane, a graph $G = (V, E)$, and an angle α . A spanning tree T is called an α -spanning tree if, for every point $v \in V$, there is an angle on v of size smaller than α containing all the edges (line segments) of T incident to v . Observe that, if we let D contain an arc (vu, vw) whenever the smaller angle formed by vu and vw is bigger than α , then an α -spanning tree T also $(0,0)$ -satisfies D , and vice-versa. Besides, the conflicts in this case are forbidden transitions. In [6], one can find references on the decision version of the problem, while the optimization version is investigated in [17].

Forcing constraints: Recall that, in the **FORCING CONSTRAINED (MINIMUM) SPANNING TREE** problem, we are given a pair of graphs G and D such that $V(D) = E(G)$, and we want to know whether there exists a spanning tree (find a minimum spanning tree) T of G such that $E(T) \cap \{u, v\} \neq \emptyset$ for every $uv \in E(D)$. We denote the feasibility problem by **FCST** and the optimization problem by **FCMST**.

This problem was introduced in [19], where the authors prove that **FCST** is **NP-complete** even if the conflict graph is a forest of paths of length at most two. To the best of our knowledge, this is the only existing paper that investigates this problem. Here, we show a reduction from **FCST**(G, D) to **G-DCST**(G', D', ℓ, u), where $\ell(e) \in \{0, 1\}$ and $u(e) = |\text{dep}(e)|$ for every $e \in E(G')$, and the maximum in-degree of D' is 2. If weights are being considered, such a reduction can be made to preserve the value of the solutions, and therefore it also applies to the optimization problem.

At least one/all dependency constraints: The following two dependency constrained problems are introduced in [45]. Given a graph G and a digraph D such that $V(D) = E(G)$, one wants to know whether there exists a spanning tree T of G such that: $E(T) \cap \text{dep}(e) \neq \emptyset$ for every $e \in E(T)$ with $\text{dep}(e) \neq \emptyset$, called the **AT LEAST ONE DEPENDENCY CONSTRAINED SPANNING TREE** problem; or $\text{dep}(e) \subseteq E(T)$ for every $e \in E(T)$, called the **ALL DEPENDENCY CONSTRAINED SPANNING TREE** problem. We denote these problems by **L-DCST** and **A-DCST**, and the related optimization problems by **L-DCMST** and **A-DCMST**, respectively. Note that these are special cases of our problem.

In [45], it is proved that both **L-DCST** and **A-DCST** are **NP-complete**, even if G is a planar chordal graph with diameter two or maximum degree three, and D is the disjoint union of arborescences of height two. Here, we strengthen the constraints on D , while also getting a lower bound on the running time of exponential algorithms for these problems (result (i)). Observe that this result comprises cases where the maximum in-degree of D is one, and so the generalized problem coincides with both **L-DCST** and **A-DCST**.

Still in [45], the authors prove that **L-DCMST** and **A-DCMST** are **W[2]-hard** when parameterized by the weight of a solution, and that, unless $\text{P} = \text{NP}$, they cannot be approximated with a ratio of $\ln |V(G)|$ even if: G is bipartite; the dependency relations occur only between

adjacent edges of G ; and each weak component of D has diameter one. One can notice that the weight of a solution in their $W[2]$ -hardness reduction is $O(n)$, where $n = |V(G)|$. This means that there is no FPT algorithm for L-DCMST and A-DCMST parameterized by n , unless $FPT = W[1]$. This contrasts with the decision problem, which can be solved in time $O(2^m \cdot (n + m)) = O(2^{n^2} \cdot n^2)$, where $m = |E(G)|$ (result (c)).

Maximum degree constraints: Given a graph $G = (V, E)$, and a positive integer k , the MAXIMUM DEGREE CONSTRAINED SPANNING TREE problem consists in deciding whether G has a spanning tree T such that $d_T(v) \leq k$ for every $v \in V(G)$, where $d_T(v)$ is the degree of v in T . This problem was introduced in [20]. Observe that it is NP-complete, even for $k = 2$, since this case generalizes the Hamiltonian path problem [25]. In [39], it is proved to be NP-complete even for grid graphs of maximum degree three. Also, [39] tackles the Euclidean optimization version of the problem (i.e., vertices are points on the plane, and edges are weighted according to the Euclidean distance). The Euclidean optimization version remains NP-hard when $k \leq 3$, and is polynomial-time solvable when $k \geq 5$, remaining open for $k = 4$. Several heuristic, approximation, and exact approaches have been proposed for the problem (see [8, 33, 43] and references therein).

Here, we denote the feasibility version of this problem by MDST, and the optimization version by MDMST. We present a reduction from MDST(G, k) to G-DCST($G', D, 0, u$) where $u(e) \in \{0, k\}$ for every $e \in E(G')$. The reduction also applies to the optimization problem since it preserves the value of the solutions.

Minimum degree constraints: Given a graph $G = (V, E)$, and a positive integer k , the MINIMUM DEGREE CONSTRAINED SPANNING TREE problem consists in deciding whether G has a spanning tree T such that $d_T(v) \geq k$ for every non-leaf vertex v of T . Here, we denote the feasibility version of this problem by mDST, and the optimization version by mDMST. This problem was introduced in [2], where it is shown to be NP-hard for every $k \in \{4, \dots, \frac{|V(G)|}{2}\}$. On the other hand, [2] proves that the problem can be solved (by inspection) for degree bounds between $\frac{|V(G)|}{2} + 1$ and $|V(G)| - 1$. In [3], the problem was shown to be NP-hard for $k = 3$. The case $k \leq 2$ is equivalent to the classical spanning tree problem. Integer linear programs and solution methods were proposed in [1, 2, 37].

An interesting variation of mDST is obtained when the set of leaves is fixed in the input. More formally, given a graph G , a subset $C \subseteq V$, and a positive integer k , it consists in finding a spanning tree T of G such that $d_T(v) \geq k$, for every $v \in C$, and $d_T(v) = 1$, for every $v \in V \setminus C$. We denote the feasibility version of this problem by FmDST, and the optimization version by FmDMST. This problem was introduced in [22], where the authors prove that FmDST is NP-complete for $k \geq 2$, and FmDMST is NP-hard even for complete graphs. Also, some necessary and sufficient conditions are given for feasibility.

Here, we present a reduction from both mDST(G, k) and FmDST(G, C, k) to G-DCST(G', D, ℓ, u) where $\ell(e) \in \{0, 1, k\}$ and $u(e) \in \{1, |\text{dep}(e)|\}$ for every $e \in E(G)$.

Again, our reduction preserves the values of the solutions and therefore works for the optimization version as well.

Applications. As G-DCST generalizes all these problems, it inherits their applications, such as design of wind farm networks [12], VLSI global routing [41], or low-traffic communication networks [37]. In particular, dependency relations can model communication systems with protocol conversion restrictions [44]. Besides, we can get unified results for all of them by considering G-DCST.

Organization. In Section 2, we present the formal definitions and notation used throughout the paper; in Section 3 we present our NP-complete results; in Section 4, our positive results; in Section 5, we show how to model the many constrained spanning tree problems as special cases of our problem; and in Section 6, we discuss our results and pose some open questions.

2 Definitions and notation

Graphs. For missing basic definitions on graph theory, we refer the reader to [46]. Let G be a simple graph (henceforth called simply a graph), and D be a digraph. We denote by $E(G), E(D)$ the edge set of G and arc set of D , respectively. Also, we denote an edge $\{u, v\}$ of G by uv , and arc with head v and tail u of D by (u, v) . We say that D is *symmetric* if $(v, u) \in E(D)$ whenever $(u, v) \in E(D)$. A (di)graph G (D) is *complete* if $uv \in E(G)$ ($\{(u, v), (v, u)\} \subseteq E(D)$) for every pair of vertices u and v in G (D). It is *empty* if has no edges (no arcs).

If $C \subseteq V(G)$ is such that $uv \in E(G)$ for every $u, v \in V(G)$, $u \neq v$, then we call C a *clique*. And if there are no edges between vertices in C , we say that C is an *independent set*. A vertex $v \in V(G)$ is called *universal* if $N(v) = V(G) \setminus \{v\}$, where $N(v)$ stands for the set of neighbors of v . A tree T is called a *star* if it has a universal vertex v , called *center*. Similarly, an *out-star* (*in-star*) is a directed graph D with a vertex v such that any other vertex is an out-neighbor (in-neighbor) of v and $V(D) \setminus \{v\}$ is an independent set.

Definition of the problems. Let $G = (V, E)$ be a graph and $D = (E, A)$ be a digraph whose vertices are the edges of G . We say that $e_1 \in E$ is a *D-dependency* of $e_2 \in E$ if $(e_1, e_2) \in A$. For each $e \in E$, we define its *D-dependency set* as $\text{dep}_D(e) = \{e' \in E : (e', e) \in A\}$, and for $E' \subseteq E$, let $\text{dep}_D(E') = \cup_{e \in E'} \text{dep}_D(e)$; from now on we omit D from the subscript whenever it is clear from the context. Also, let $\ell, u : E \rightarrow \mathbb{N}$ be functions that assign a non-negative integer to each edge of G . We say that a subgraph H of G (ℓ, u) -satisfies D if $\ell(e) \leq |\text{dep}(e) \cap E(H)| \leq u(e)$, for every $e \in E(H)$.

We introduce the GENERALIZED DEPENDENCY CONSTRAINED SPANNING TREE problem as, given a graph G , a digraph $D = (E(G), A)$, and functions $\ell, u : E(G) \rightarrow \mathbb{N}$, deciding

whether there exists a spanning tree T of G such that T (ℓ, u) -satisfies D . We abbreviate this with $\text{G-DCST}(G, D, \ell, u)$. Observe that it corresponds to the feasibility problem. If we are also given a weight function $w : E \rightarrow \mathbb{R}$, then we define the **GENERALIZED DEPENDENCY CONSTRAINED MINIMUM SPANNING TREE** problem as the problem of finding a spanning tree T^* of G that minimizes the weight sum and that (ℓ, u) -satisfies D ; this problem is denoted by G-DCMST .

Polynomial reductions and Exponential Time Hypothesis. Given problems Π and Π' , we write $\Pi \preceq_P \Pi'$ if there exists a polynomial reduction from Π to Π' . This means that problem Π' is at least as hard as problem Π . The *Exponential Time Hypothesis* (denoted by **ETH**) of Impagliazzo et al. [29, 30] states that the **3-SAT** problem cannot be solved in time $2^{o(n+m)}$, where n is the number of variables and m the number of clauses of the input formula. In particular, if it is possible to reduce **3-SAT** to problem Π and the produced instance has size linear in the size of the input formula, then the **ETH** implies that problem Π cannot be solved in time $2^{|x|}$ either, where $|x|$ denotes the size of the input of Π . We refer the reader to [5] for basic background on computational complexity.

Parameterized complexity. We refer to [16] for a recent monograph on parameterized complexity. Here, we recall only some basic definitions. A *parameterized problem* is a decision problem whose instances are pairs $(x, k) \in \Sigma^* \times \mathbb{N}$, where k is called the *parameter*. A parameterized problem L is *fixed-parameter tractable* (**FPT**) if there exists an algorithm \mathcal{A} , a computable function f , and a constant c such that, given an instance $I = (x, k)$ of L , we get that \mathcal{A} (called an *FPT algorithm*) correctly decides whether $I \in L$ in time bounded by $f(k) \cdot |I|^c$. For instance, the **VERTEX COVER** problem parameterized by the size of the solution is **FPT**.

Within parameterized problems, the class $\text{W}[1]$ may be seen as the parameterized equivalent to the class **NP** of classical optimization problems. Without entering into details (see [16] for the formal definitions), a parameterized problem being $\text{W}[1]$ -hard can be seen as a strong evidence that this problem is *not* **FPT**. The canonical example of $\text{W}[1]$ -hard problem is **INDEPENDENT SET** parameterized by the size of the solution. The class $\text{W}[2]$ of parameterized problems is a class that contains $\text{W}[1]$, and such that the problems that are $\text{W}[2]$ -hard are even more unlikely to be **FPT** than those that are $\text{W}[1]$ -hard (again, see [16] for the formal definitions). The canonical example of $\text{W}[2]$ -hard problem is **DOMINATING SET** parameterized by the size of the solution.

Matroids. We state here some basic tools about matroids that we will use in the algorithms of Section 4, and we refer to [34, 38] for more background. A (finite) *matroid* M is a pair (E, \mathcal{I}) , where E is a finite set, called the *ground set*, and \mathcal{I} is a family of subsets of E , called the *independent sets*, satisfying the following properties:

1. The empty set is independent, that is, $\emptyset \in \mathcal{I}$.

2. Every subset of an independent set is independent, that is, for each $A' \subseteq A \subseteq E$, if $A \in \mathcal{I}$ then $A' \in \mathcal{I}$. This is called the *hereditary property*.
3. If $A, B \in \mathcal{I}$ with $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$. This is called the *augmentation property*.

Every graph or multigraph $G = (V, E)$ gives rise to a so-called *graphic matroid* having E as ground set, and a set $F \subseteq E$ is independent if and only if $G[F]$ is acyclic.

Given a collection $\mathcal{E} = \{E_1, E_2, \dots, E_k\}$ of pairwise disjoint sets, and integers $\{d_1, \dots, d_k\}$ such that $0 \leq d_i \leq |E_i|$ for every $i \in [k] = \{1, \dots, k\}$, the *partition matroid* with ground set $E = \bigcup_{i=1}^k E_i$ has $S \subseteq E$ as an independent set if and only if $|S \cap E_i| \leq d_i$ for every $i \in [k]$.

The Matroid Intersection Theorem, proved by Edmonds [23], states that the problem of finding a largest common independent set of two matroids over the same ground set can be solved in polynomial time.

3 NP-completeness results

In this section, we present our NP-complete results. First, we impose in Section 3.1 constraints on the structure of D (also getting constraints on G as a byproduct), and then we focus in Section 3.2 on hardness results imposing constraints on functions ℓ and u .

3.1 Constraints on D

We prove result (i). First, we consider the case where D is a forest of out-stars with at most three vertices. Later we show that the orientation of D in the reduction can be changed to get a forest of directed paths of length at most two. In addition, in Theorem 3, we modify the reduction to obtain D as a forest of in-stars with at most three vertices, thus closing all the possible orientations of forest of paths of length at most two.

Theorem 1. *G -DCST(G, D, ℓ, u) is NP-complete, even when $\ell(e) = u(e) = |\text{dep}(e)|$ for every $e \in E(G)$, D is a forest of out-stars of maximum degree two, and G is an outerplanar chordal graph with diameter at most two. Furthermore, this problem cannot be solved in time $2^{o(n+m)}$ unless the ETH fails, where $n = |V(G)|$ and $m = |E(G)|$.*

Proof. Notice that, given a spanning tree T , one can check whether T (ℓ, u)-satisfies D in polynomial time; hence, G -DCST(G, D, ℓ, u) is in NP. To prove NP-completeness, we present a reduction from (3, 2, 2)-SAT to G -DCST. In the (3, 2, 2)-SAT problem, we are given a CNF formula ϕ where each clause has at most three literals, and each variable appears at most twice positively and at most twice negatively. This problem is well-known to be NP-complete [14, 25]. So consider a CNF formula ϕ on n variables and m clauses; we build an instance (G, D, ℓ, u) of G -DCST as follows (follow the construction in Figure 1):

- Add to G vertex v , and vertices v_x , $v_{\bar{x}}$, and w_x related to each variable x , and at most three vertices $\{v_c^1, v_c^2, v_c^3\}$ related to each clause c (these vertices represent the literals in c). Then, make v adjacent to every other vertex; v_x adjacent to $v_{\bar{x}}$ for every variable x ; and add for each clause c a path (v_c^1, v_c^2) or (v_c^1, v_c^2, v_c^3) , depending on how many literals c has (observe that we can suppose that c has at least two literals). Edge vv_x will be interpreted as the true assignment of x , while edge $vv_{\bar{x}}$ as the false one.
- For each variable x , add arc $(v_x v_{\bar{x}}, vv_x)$ to D . For every variable x and each occurrence of x in a clause c , say as the i -th literal in c , add to D arc (vv_x, vv_c^i) if x appears positively in c , or arc $(vv_{\bar{x}}, vv_c^i)$ if x appears negatively in c .
- Finally, let $\ell(e) = u(e) = |\text{dep}(e)|$ for every $e \in E(G)$.

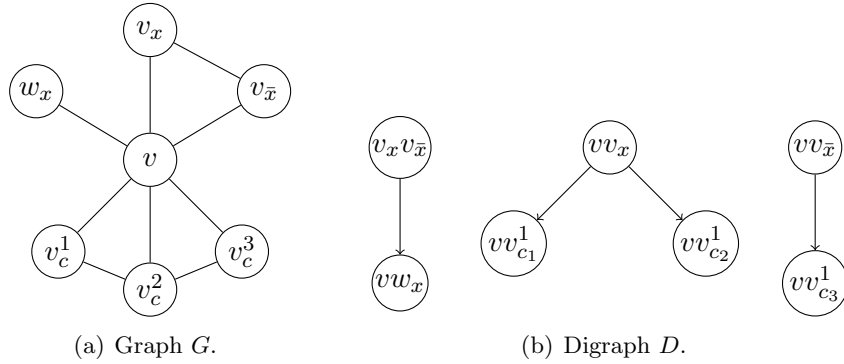


Figure 1: Illustration of the reduction from (3, 2, 2)-SAT in Theorem 1. In Figure 1(a), we represent a variable gadget together with a gadget of a clause containing three literals. In Figure 1(b), for a variable x , we represent the dependency between $v_x v_{\bar{x}}$ and vv_x , and also the arcs leaving vv_x and $vv_{\bar{x}}$ when x appears positively in c_1 and c_2 , and negatively in c_3 , being related to the first literal in each of these clauses.

One can see that G is an outerplanar chordal graph, and that each component (different from the one containing $v_x v_{\bar{x}}$ and vv_x) of D is an out-star from vv_x or $vv_{\bar{x}}$, for some variable x ; we get $\Delta^+(D) \leq 2$ by the constraint in the number of appearances of a literal.

To show the correctness of the reduction, consider first a satisfying assignment of ϕ . We build a spanning tree T of G with the following edges: for each variable x , add to T $v_x v_{\bar{x}}$, vv_x , and either vv_x if x is true, or $vv_{\bar{x}}$, if x is false; for each clause $c = (\ell_1 \vee \ell_2 \vee \ell_3)$, add path (v_c^1, v_c^2, v_c^3) and an edge vv_c^i for some $i \in \{1, 2, 3\}$ such that ℓ_i is a true literal (analogously when c has only two literals). Because, for every variable x , vv_x , $v_x v_{\bar{x}}$ and exactly one edge among $\{vv_x, vv_{\bar{x}}\}$ are chosen, and for every clause c exactly one edge among $\{vv_c^1, vv_c^2, vv_c^3\}$ is chosen, apart from the path (v_c^1, v_c^2, v_c^3) , one can see that T is

indeed a spanning tree of G . The dependencies can also be seen to be satisfied since we only choose an edge vv_c^i if the corresponding literal is true (hence the dependency is chosen too).

Conversely, let T be a solution for $\mathbf{G}\text{-DCST}(G, D, \ell, u)$. For each variable x , because vw_x is a cut edge and $(v_x v_{\bar{x}}, vw_x) \in E(D)$, we get that $\{vw_x, v_x v_{\bar{x}}\} \subseteq E(T)$. Besides, for each variable x , since vv_x and $vv_{\bar{x}}$ form a cut and also form a cycle with $v_x v_{\bar{x}}$, we get that exactly one between vv_x and $vv_{\bar{x}}$ is in T . We then assign x to true if $vv_x \in E(T)$, and to false otherwise. Now, consider a clause $c = (\ell_1 \vee \ell_2 \vee \ell_3)$; since the edges $\{vv_c^1, vv_c^2, vv_c^3\}$ form a cut, at least one of them is in T , say $vv_c^1 \in E(T)$ and say that x is the variable related to ℓ_1 . If $\ell_1 = x$, then $(vv_x, vv_c^1) \in E(D)$, which implies that $vv_x \in E(T)$ and that x is true. And if $\ell_1 = \bar{x}$, then $(vv_{\bar{x}}, vv_c^1) \in E(D)$, which implies that $vv_{\bar{x}} \in E(T)$ and that x is false (therefore ℓ_1 is true). In any case, c is satisfied. The case where c has only two literals is analogous.

Finally, for the lower bound $2^{o(n+m)}$, just observe that the constructed instance has size linear in the size of the given formula. \square

Observe that when either $\Delta^-(D) = 0$ or $\Delta^+(D) = 0$, we get that D is the empty graph, and that $\mathbf{G}\text{-DCST}(G, D, \ell, u)$ reduces to deciding whether G is connected. Also, by the previous theorem, we get that the problem is \mathbf{NP} -complete if $\Delta^-(D) = 1$, thus giving us a dichotomy with regard to the value of $\Delta^-(D)$. Concerning $\Delta^+(D)$, the previous theorem tells us that the problem becomes \mathbf{NP} -complete for $\Delta^+(D) = 2$. With a small modification on the previous reduction, we can also get a dichotomy with regard to $\Delta^+(D)$.

Theorem 2. *$\mathbf{G}\text{-DCST}(G, D, \ell, u)$ is \mathbf{NP} -complete, even when $\ell(e) = u(e) = |\text{dep}(e)|$ for every $e \in E(G)$, D is a union of directed paths with length at most two, and G is a chordal outerplanar graph with diameter two. Furthermore, this problem cannot be solved in time $2^{o(n+m)}$ unless the ETH fails, where $n = |V(G)|$ and $m = |E(G)|$.*

Proof. Consider the same construction from the Theorem 1, except that each out-star with two leaves is turned into a directed path of length two. Observe that if $(vv_c^i, uv_x, vv_{c'}^j)$ is a path in D , then the previous arguments might not work simply because we might be forced to pick edge vv_c^i when variable x is set to true (i.e., edge uv_x is chosen). However, in this case we can remove some of the edges of the path (v_c^1, v_c^2, v_c^3) in order to avoid cycles. A similar argument is made for out-stars containing a vertex of type vv_x . \square

Recall that $\mathbf{L}\text{-DCST}(G, D)$ and $\mathbf{A}\text{-DCST}(G, D)$ denote the dependency constrained spanning tree problem ($\mathbf{G}\text{-DCST}$) where at least one dependency (if any exists) or all dependencies are satisfied, respectively. Also, note that, if $\Delta^-(D) \leq 1$, then we get that $\mathbf{L}\text{-DCST}(G, D)$ and $\mathbf{A}\text{-DCST}(G, D)$ coincide with $\mathbf{G}\text{-DCST}(G, D, \ell, u)$ by assigning $\ell(e) = u(e) = |\text{dep}(e)|$ for every $e \in E(G)$. Thus, the following corollary, which strengthens the results in [45], is a direct consequence of the previous two theorems.

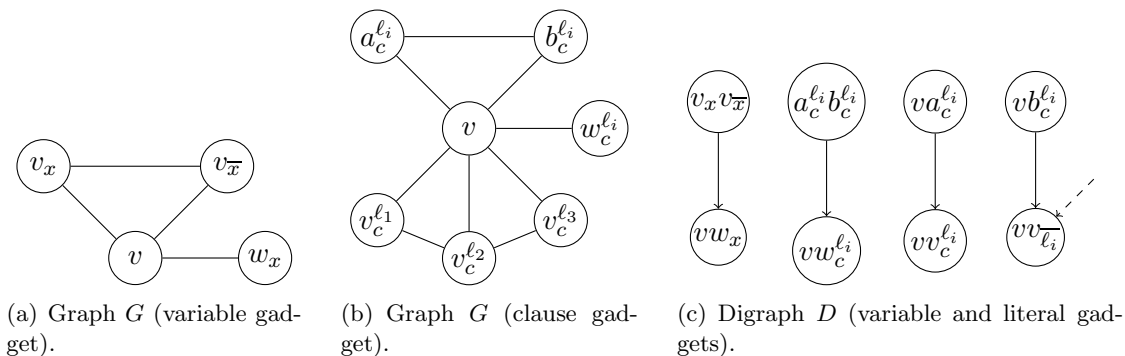


Figure 2: Illustration of the reduction from $(3, 2, 2)$ -SAT used in Theorem 3.

Corollary 1. L -DCST(G, D) and A -DCST(G, D) are NP-complete, even if G is an outerplanar chordal graph with diameter two, and D is the union of out-stars with $\Delta^+(D) = 2$, or the union of paths of length at most two. Furthermore, these problems cannot be solved in time $2^{o(n+m)}$ unless the ETH fails, where $n = |V(G)|$ and $m = |E(G)|$.

In [19], it is shown that CCMST is NP-complete if the conflict graph is a forest of paths of length at most two. An orientation of such a forest may lead to directed paths, out-stars, or in-stars. The NP-completeness of G -DCST in the first two cases is proved in Theorems 1 and 2. The case of in-stars is approached next.

Theorem 3. G -DCST(G, D, ℓ, u) is NP-complete, even when $\ell(e) = u(e) = |\text{dep}(e)|$ for every $e \in E(G)$, D is a forest of in-stars of maximum in-degree two, and G is an outerplanar chordal graph with diameter at most two. Furthermore, this problem cannot be solved in time $2^{o(n+m)}$ unless the ETH fails, where $n = |V(G)|$ and $m = |E(G)|$.

Proof. Notice that a spanning tree T of G can be checked to (ℓ, u) -satisfy D in polynomial time, thus G -DCST(G, D, ℓ, u) is in NP. To prove NP-completeness, we again make a reduction from $(3, 2, 2)$ -SAT to G -DCST. This way, consider a CNF formula ϕ on n variables and m clauses. We build an instance (G, D, ℓ, u) of G -DCST as follows (see Figure 2):

- Start by adding a vertex v , which will be universal. For each variable x , add to G vertices $v_x, v_{\bar{x}}$, and w_x , making them adjacent to v , and add edge $v_x v_{\bar{x}}$; see Figure 2(a). Selecting edge $v v_x$ will correspond to the true assignment for x , and $v v_{\bar{x}}$ to the false one.
- For each clause $c = (\ell_1 \vee \ell_2 \vee \ell_3)$ with three literals, add to G vertices $\{v_c^{l_i}, a_c^{l_i}, b_c^{l_i}, w_c^{l_i} \mid i \in [3]\}$, and make them adjacent to v . Then, add path $(v_c^{l_1}, v_c^{l_2}, v_c^{l_3})$, and edges $\{a_c^{l_i} b_c^{l_i} \mid i \in [3]\}$; see Figure 2(b). Proceed analogously if c has two literals. For each literal ℓ_i , selecting edge $v a_c^{l_i}$ will indicate that c is satisfied by ℓ_i , and selecting edge $v b_c^{l_i}$ will indicate that c must be satisfied by some of its other literals.

- For each variable x , add arc $(v_x v_{\bar{x}}, v w_x)$ to D . For each clause c and each literal ℓ_i of c , add arcs $(a_c^{\ell_i} b_c^{\ell_i}, v w_c^{\ell_i})$, $(v a_c^{\ell_i}, v v_c^{\ell_i})$, and $(v b_c^{\ell_i}, v v_{\bar{\ell}_i})$ to D ; see Figure 2(c).
- Finally, let $\ell(e) = u(e) = |\text{dep}(e)|$ for every $e \in E(G)$.

Observe that G is an outerplanar chordal graph, and that each component of D is an in-star. We get $\Delta^-(D) \leq 2$ by the constraint in the number of appearances of a literal.

To show the correctness of the reduction, consider first a satisfying assignment of ϕ . We build a spanning tree T of G as follows. For each variable x , choose edges $v_x v_{\bar{x}}$ and $v w_x$; then choose edge $v v_x$ if x is true, and edge $v v_{\bar{x}}$ otherwise. For each clause c with three literals, add edges $\{a_c^{\ell_i} b_c^{\ell_i}, v w_c^{\ell_i} \mid i \in [3]\}$. Also, for each $i \in [3]$, add $\{v v_c^{\ell_i}, v a_c^{\ell_i}\}$ if ℓ_i is true; otherwise, add $v b_c^{\ell_i}$. Finally, use path $(v_c^{\ell_1}, v_c^{\ell_2}, v_c^{\ell_3})$ to connect any possibly disconnected vertex. Proceed analogously if c has two literals. Denote by X the set of variables of ϕ , and by C the set of clauses; also, write $\ell_i \in c$ to denote the fact that literal ℓ_i appears in c . We first show that T is a spanning tree of G . It is easy to see that T spans $\{v_x, v_{\bar{x}}, v, w_x \mid x \in X\}$, and also every vertex of degree 1 in G . Now, given a clause c , because each literal ℓ_i in c is either true or false, we get that either $v a_c^{\ell_i}$ or $v b_c^{\ell_i}$ is in T , and since $a_c^{\ell_i} b_c^{\ell_i} \in E(T)$ we get that T also spans $\{a_c^{\ell_i}, b_c^{\ell_i} \mid c \in C, \ell_i \in c\}$. Finally, for each clause c , we know that at least one of its literals is true, which means that at least one of the edges linking the path $(v_c^{\ell_1}, v_c^{\ell_2}, v_c^{\ell_3})$ to v is chosen, and since it is always possible to choose edges from this path to connect any possible remaining disconnected vertex, we are done. Now, we prove that dependencies are satisfied. Dependencies in $\{(v_x v_{\bar{x}}, v w_x) \mid x \in X\}$, and in $\{(a_c^{\ell_i} b_c^{\ell_i}, v w_c^{\ell_i}) \mid c \in C, \ell_i \in c\}$ are all satisfied since all the involved edges are contained in T . Dependencies in $\{(v a_c^{\ell_i}, v v_c^{\ell_i}) \mid c \in C, \ell_i \in c\}$ are also valid because we only add these edges together. Finally, given a variable x , if x is true, then we choose edge $v v_x$, and $v b_c^{\ell_i}$ for each clause c such that \bar{x} is the i -th literal of c ; and if x is false then we choose edge $v v_{\bar{x}}$, and $v b_c^{\ell_i}$ for each clause c such that x is the i -th literal of c . This settles the last type of dependencies.

Conversely, let T be a solution for $\text{G-DCST}(G, D, \ell, u)$. For each variable x , because $v w_x$ is a cut edge and $(v_x v_{\bar{x}}, v w_x) \in E(D)$, we get that $\{v w_x, v_x v_{\bar{x}}\} \subseteq E(T)$. Besides, for each variable x , since $v v_x$ and $v v_{\bar{x}}$ form a cut and also form a cycle with $v_x v_{\bar{x}}$, we get that exactly one between $v v_x$ and $v v_{\bar{x}}$ is in T . We then assign x to true if $v v_x \in E(T)$, and to false otherwise. Now, consider a clause $c = (\ell_1 \vee \ell_2 \vee \ell_3)$; since the edges $\{v v_c^{\ell_1}, v v_c^{\ell_2}, v v_c^{\ell_3}\}$ form a cut, at least one of them is in T , say $v v_c^{\ell_1} \in E(T)$. Hence, because $(v a_c^{\ell_1}, v v_c^{\ell_1}) \in A(D)$, we get that $v a_c^{\ell_1} \in E(T)$. But then, since $a_c^{\ell_1} b_c^{\ell_1} \in E(T)$, we have that $v b_c^{\ell_1} \notin E(T)$, which in turn implies that $v v_{\bar{\ell}_1} \notin E(T)$ because of the dependency $(v b_c^{\ell_1}, v v_{\bar{\ell}_1})$. We then conclude that $v v_{\ell_1}$ must be chosen, henceforth ℓ_1 is a true literal in c . The case where c has only two literals is analogous.

Finally, since the constructed instance has size linear in the size of ϕ , we obtain the claimed lower bound $2^{o(n+m)}$ under the ETH. \square

Let us observe that, similarly to Corollary 1, the results in Theorem 3 can be stated to

$\text{A-DCST}(G, D)$, since this problem is equivalent to $\text{G-DCST}(G, D, \text{dep}, \text{dep})$. This also extends the achievements from [45].

3.2 Constraints on functions ℓ and u

In this subsection, we examine the complexity of $\text{G-DCST}(G, D, \ell, u)$ by focusing on functions ℓ and u . Recall that, given constants c, c' , $\text{G-DCST}(G, D, c, c')$ denotes the problem restricted to instances where $\ell(e) = c$ and $u(e) = c'$ for every $e \in E(G)$. Given a function $f : E \rightarrow \mathbb{N}$ and a positive integer c , we denote by $f + c$ the function obtained from f by adding c to $f(e)$ for every $e \in E$. The following lemma will be useful.

Lemma 1. *Let c, ℓ, u be positive integers with $u \geq \ell$. Then, instances (G, D, ℓ, u) and $(G', D', \ell + c, u + c)$ of G-DCST are equivalent.*

Proof. Choose any $v \in V(G)$, and let G' be obtained from G by adding $\ell + c + 1$ vertices of degree one pending in v ; denote the new edges by $e_1, \dots, e_{\ell+c+1}$. Now, add the symmetric clique on vertices $\{e_1, \dots, e_{\ell+c+1}\}$ to D , and add $e_i e$ to D , for every $i \in [c]$ and every $e \in E(G)$. Let G' and D' be the obtained graph and digraph, respectively. Finally, let $\ell'(e) = \ell + c$ and $u'(e) = u + c$, for every $e \in E(G')$. We prove that (G, D, ℓ, u) is a yes-instance of G-DCST if and only if (G', D', ℓ', u') is a yes-instance of G-DCST .

First, let T be a spanning tree of G that (ℓ, u) -satisfies D . Let T' be obtained from T by adding $e_1, \dots, e_{\ell+c+1}$. Clearly T' is a spanning tree of G' ; we prove that T' (ℓ', u') -satisfies D' . Let $e \in E(G')$. If $e \in E(G)$, because at least ℓ and at most u dependencies of e are in T , and because $E(T') \cap \text{dep}_{D'}(e) = (E(T) \cap \text{dep}_D(e)) \cup \{e_1, \dots, e_c\}$, we get that at least $\ell + c$ and at most $u + c$ dependencies of e are in T' . And if $e \in \{e_1, \dots, e_{\ell+c+1}\}$, we get that $E(T') \cap \text{dep}_{D'}(e) = \{e_1, \dots, e_{\ell+c+1}\} \setminus \{e\}$ and again the constraints hold.

On the other hand, let T' be a spanning tree of G' ; we know that $\{e_1, \dots, e_{\ell+c+1}\} \subseteq E(T')$. Let T be obtained from T' by removing these edges, and consider $e \in E(G)$. It follows that $|\text{dep}_D(e) \cap E(T)| = |(\text{dep}_{D'}(e) \cap E(T')) \setminus \{e_1, \dots, e_c\}| = |\text{dep}_{D'}(e) \cap E(T')| - c$, and since $\ell + c \leq |\text{dep}_{D'}(e) \cap E(T')| \leq u + c$ we get that T (ℓ, u) -satisfies D . \square

First, we analyze the cases where $\ell = 0$. Recall that in CCST , whenever an edge e is chosen, no dependencies of e can be chosen; this translates to having $\ell(e) = u(e) = 0$ for every $e \in E(G)$. So in a sense, when one considers instances (G, D, ℓ, u) of G-DCST where $\ell(e) = 0$ for each $e \in E(G)$, one can think of the problem as a “weak” version of CCST . It thus makes sense to ask whether this version turns out to be polynomial. Indeed, we notice that $\text{G-DCST}(G, D, \ell, u)$ with $\ell = 0$ and $u(e) \geq |\text{dep}_D(e)|$, for each $e \in E(G)$, is an easily solvable problem since every spanning tree of G trivially (ℓ, u) -satisfies D . In the following theorem, we see that this is not the case when u is a constant function.

Theorem 4. *Let c be a positive integer. Then $\text{G-DCST}(G, D, 0, c)$ is NP-complete.*

Proof. Recall that $\text{CCST}(G, D)$ is NP-complete [19] and equivalent to $\text{G-DCST}(G, D', 0, 0)$ when D' is an arbitrary orientation of D . Given an instance $(G = (V, E), D)$ of CCST , where D is an undirected graph with $V(D) = E$, we construct an equivalent instance $(G', D', 0, c)$ of G-DCST as follows (cf. Figure 3)

- Let G' be obtained from G by adding a new vertex p and, for each $i \in [c]$ and each edge $e \in E(G)$, adding a new vertex p_e^i . Then, make p adjacent to every p_e^i , and to an arbitrary vertex $q \in V(G)$. More formally, $G' = (V \cup V', E \cup E')$, where

$$V' = \{p\} \cup \{p_e^i : e \in E, i \in [c]\} \text{ and}$$

$$E' = \{pq\} \cup \{pp_e^i : e \in E, i \in [c]\}.$$

- Let D' be obtained from an arbitrary orientation of D by adding an arc (pp_e^i, e) for every $e \in E(G)$ and every $i \in [c]$.

The constructed instance has size clearly polynomial on the size of (G, D) (recall that c is a constant). Note that $G'[V' \cup \{q\}]$ is a star. Now, we show that (G, D) is a yes-instance of CCST if and only if $(G', D', 0, c)$ is a yes-instance of G-DCST .

First, let T be a solution for $\text{CCST}(G, D)$, and let T' be obtained from T by adding E' . Clearly T' is a spanning tree of G' ; hence it remains to show that T' $(0, c)$ -satisfies D' . For this, consider $e \in E(T')$. If $e \in E(T)$, then because T $(0, 0)$ -satisfies D , we have that $E(T') \cap \text{dep}_{D'}(e) = \{pp_e^1, \dots, pp_e^c\}$ and therefore $0 \leq |E(T') \cap \text{dep}_{D'}(e)| = c$. And if $e \in E'$, we have that $\text{dep}_{D'}(e) = \emptyset$ and trivially $0 = |\text{dep}_{D'}(e) \cap E(T')| \leq c$.

Conversely, let T' be a spanning tree of G' that $(0, c)$ -satisfies D' , and let $T = T'[V]$. Because p separates $V' \setminus \{p\}$ from V , we know that T is connected and, therefore, it is a spanning tree of G ; so it remains to show that T $(0, 0)$ -satisfies D . For this consider $e \in E(T)$. Since each edge in E' is a cut edge in G' , we get that $E' \subseteq E(T')$. Therefore, since $|E(T') \cap \text{dep}_{D'}(e)| \leq c$ and $\{pp_e^1, \dots, pp_e^c\} \subseteq E(T') \cap \text{dep}_{D'}(e)$, we get that $E(T') \cap (E(G) \setminus E') = \emptyset$, i.e., $|E(T) \cap \text{dep}_D(e)| = 0$, as we wanted to show. \square

Combining Lemma 1 and Theorem 4, we get result (ii), that is, $\text{G-DCST}(G, D, \ell, u)$ is NP-complete for every combination of constant values ℓ and u .

Corollary 2. *For every pair of positive integers ℓ, u with $\ell \leq u$, we have that $\text{G-DCST}(G, D, \ell, u)$ is NP-complete.*

Proof. Let $c = u - \ell$. By Theorem 4, we have that $\text{G-DCST}(G, D, 0, c)$ is NP-complete, and by Lemma 1, we have that $\text{G-DCST}(G, D, \ell, c + \ell = u)$ also is. \square

As we have already mentioned, if $u(e) = |\text{dep}(e)|$ for every $e \in E(G)$, then $\text{G-DCST}(G, D, 0, u)$ is easy since any spanning tree $(0, u)$ -satisfies D . Hence, it is natural to ask whether the problem continues to be easy when $u(e)$ is just slightly smaller than $|\text{dep}(e)|$. The following corollary is trivially obtained from previous results. It answers the

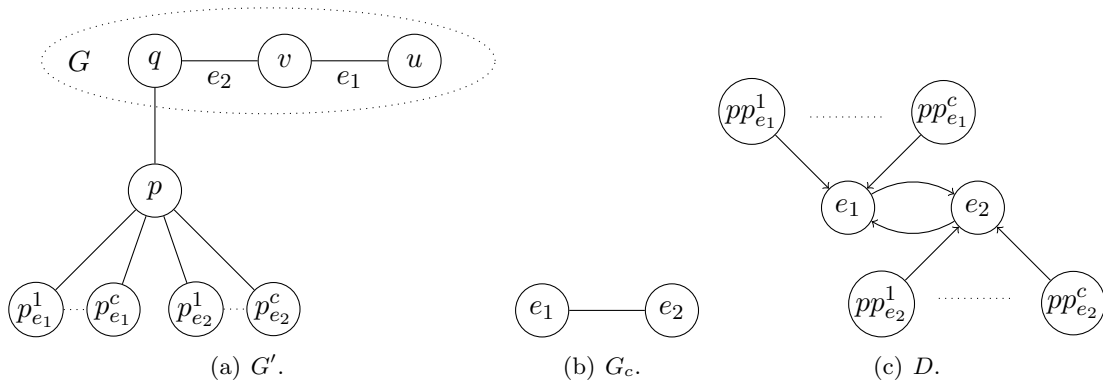


Figure 3: Illustration of the reduction for CCMST.

aforementioned question negatively and yields result (iii). Given a positive integer c , we denote by $\text{dep} - c$ the function $u : E(G) \rightarrow \mathbb{N}$ defined by $u(e) = \max\{\text{dep}(e) - c, 0\}$.

Corollary 3. $G\text{-DCST}(G, D, 0, \text{dep} - 1)$ is NP-complete, even when D is a collection of directed paths of length at most two.

Proof. Assume that D is a collection of directed paths of length at most two. Then, $\Delta^-(D) = 1$ and therefore $\text{dep} - 1$ is equal to the constant function zero. It means that we are considering $G\text{-DCST}(G, D, 0, 0)$. This problem is equivalent to $\text{CCST}(G, D')$, where D' is the (undirected) underlying graph of D . Then D' is a collection of paths of length at most two, and it is known that $\text{CCST}(G, D')$ is NP-complete [19]. \square

4 Positive results

In this section, we present some polynomial cases of the $G\text{-DCST}$ problem. Whenever possible, we deal with the optimization version instead, i.e., we consider that we are also given a weight function w and that the objective is to find a spanning tree that (ℓ, u) -satisfies D having minimum weighted sum-weight. This is denoted by $G\text{-DCMST}(G, D, \ell, u, w)$.

Similarly to the previous section, we first focus in Section 4.1 on constraints on the dependency graph D , and then in Section 4.2 on constraints on the functions ℓ and u . Finally, we present in Section 4.3 a simple exponential-time algorithm to solve the problem.

4.1 Constraints on D

We start by investigating the case where D is a collection of directed paths of length at most one (note that there might be some edges of G that are isolated vertices in D). So, given a graph $G = (V, E)$ and a digraph $D = (E, A)$, write E as $\{e_1, \dots, e_m\}$ and assume that $A = \{(e_i, e_{i+t}) : i \in \{1, \dots, t\}\}$, for some $t \leq \lfloor \frac{m}{2} \rfloor$. Let $S = \{e_i : i \in \{t+1, \dots, 2t\}\}$,

i.e., S is the set of the edges $e \in E(G)$ with $\text{dep}(e) \neq \emptyset$. The idea is to find a subset of edges $S' \subseteq S$ that connects the components of $G - S$ and such that the dependencies of S' do not form a cycle in G . Let $\mathcal{C} = \{C_1, \dots, C_k\}$ be the set of connected components of $G - S$, and let H be the multigraph obtained from G by contracting each C_i to a single vertex (loops are removed). Observe that there is an injection from the multiedges of H to the edges in S . Hence, given $S' \subseteq S$, we can pick $H(S')$ to be the subgraph of H induced by S' , i.e., $H(S')$ has \mathcal{C} as vertex set, and there is an edge between C_i, C_j , $i \neq j$, for each edge $e \in S'$ with one endpoint in C_i and the other one in C_j . Observe that some of the edges of S might appear inside a component. However, as we will see, these edges can actually be ignored, and this is why we do not need to add the loops in H .

Lemma 2. *Let $G = (V, E)$ be a graph, $D = (E, A)$ be a digraph, before, and ℓ, u be such that $\ell(e) = u(e) = |\text{dep}(e)| \in \{0, 1\}$ for every $e \in E(G)$. If there exists $S' \subseteq S$ such that $H(S')$ is a spanning tree of H and $(V, \text{dep}(S'))$ is acyclic, then $\mathcal{G}\text{-DCST}(G, D, \ell, u)$ is feasible. Conversely, any feasible solution of $\mathcal{G}\text{-DCST}(G, D, \ell, u)$ contains such a subset S' .*

Proof. First, consider $S' \subseteq S$ such that $H(S')$ is a spanning tree of H and $(V, \text{dep}(S'))$ is acyclic. Observe that, because $(V, \text{dep}(S'))$ is acyclic, $S' \subseteq S$, and $S \cap \text{dep}(S') = \emptyset$, we get that each C_i is a connected component of $G - S$. Thus, we can add edges of $G - S$ to $(V, \text{dep}(S'))$ so as to obtain a spanning forest F of G having connected components with the same vertex sets as C_1, \dots, C_k . After this, just add edges of S' to F ; because $H(S')$ is a spanning tree of H , we get that the obtained graph T connects all components of F without forming a cycle (i.e., T is a spanning tree of G). Finally, since $\text{dep}(S') \subseteq E(T)$ and $S \cap E(T) = S'$, we get that T satisfies D .

Conversely, let $T = (V, E_T)$ be a feasible solution of $\text{DCST}(G, D)$. As T is a spanning tree of G , we get that the edges in $E_T \cap S$ must connect the components of $G - S$, i.e., $H(E_T \cap S)$ is connected. Thus, choose S' as the edge set of any spanning tree of $H(E_T \cap S)$. We get that S' also forms a spanning tree of H , and since $S' \subseteq E_T$ and T satisfies D , we get that $\text{dep}(S') \subseteq E_T$ and therefore $(V, \text{dep}(S'))$ cannot contain a cycle. \square

In the following theorem we use the Matroid Intersection Theorem [23] to get result (a).

Theorem 5. *Let $G = (V, E)$ be a graph, $D = (E, A)$ be a digraph such that each component is a directed path of length at most 1, and ℓ, u be such that $\ell(e) = u(e) = |\text{dep}(e)| \in \{0, 1\}$ for every $e \in E(G)$. Then $\text{DCST}(G, D)$ can be solved in polynomial time.*

Proof. Given $E' \subseteq E(G)$, denote by $G(E')$ the subgraph $(V(G), E')$. Let H be obtained as before. Also, let $\mathcal{I}_1 = \{S' \subseteq S \mid G(\text{dep}(S')) \text{ is acyclic}\}$ and $\mathcal{I}_2 = \{S' \subseteq S \mid H(S') \text{ is acyclic}\}$. We have that (S, \mathcal{I}_1) and (S, \mathcal{I}_2) are matroids (on a common ground set S), since they are equivalent to the graphic matroids of the graph $(V, \text{dep}(S))$ and of the multigraph H , respectively. According to Lemma 2, $\text{DCST}(G, D)$ is feasible if and only if there is $S' \in \mathcal{I}_1 \cap \mathcal{I}_2$ such that $|S'| = k - 1$, where k is the number of components of $G' = (V, E \setminus S)$. The existence of such S' can be checked in polynomial time using the Matroid Intersection Theorem [23]. \square

4.2 Constraints on ℓ and u

Recall that $\text{CCMST}(G, G_c, w)$ is equivalent to $\text{G-DCMST}(G, D, 0, 0, w)$ when D is a symmetric digraph. From a result in [47] for the CCMST problem, we get that $\text{G-DCMST}(G, D, 0, 0, w)$ is solvable in polynomial time when D is the union of complete digraphs. We generalize this result for upper bound functions that have the same value on each clique (result (b)).

Theorem 6. *Let $(G, D, 0, u, w)$ be an instance of G-DCMST such that $D = D_1 \cup D_2 \cup \dots \cup D_k$ is the union of k disjoint complete digraphs and, for each $i \in [k]$, there exists $u_i \in [|V(D_i)|]$ such that $u(e) = u_i - 1$ for every $e \in V(D_i)$. Then, $\text{G-DCMST}(G, D, 0, u, w)$ can be solved in polynomial time.*

Proof. Note that, in this case, a solution for $\text{G-DCMST}(G, D, 0, u, w)$ can have at most u_i edges from D_i , for each $i \in [k]$. We show that solving such an instance can be formulated as a matroid intersection problem.

Observe that a subgraph T of G $(0, u)$ -satisfies D if and only if $|E(T) \cap D_i| \leq u_i$ for every $i \in [k]$. Therefore, if M is the partition matroid on $E(G)$ defined by $\{V(D_1), \dots, V(D_k)\}$ and $\{u_1, \dots, u_k\}$, we get that $T \subseteq G$ is a solution for our problem if and only if T is a spanning tree of G and $E(T)$ is an independent set of M . Hence, if M' is the graphic matroid associated with G (where the independent sets are the sets of edges inducing a spanning tree of G), we can solve our problem in polynomial time by applying the Matroid Intersection Theorem [23] to the matroids M and M' . \square

Observe that the same approach used in the previous theorem does not work when the values of u can differ inside the same clique. For instance, suppose that D_1 is the complete digraph on edges $\{e_1, \dots, e_4\}$ and that $u(e_1) = 1$ and $u(e_i) = 2$ for every $i \in \{2, 3, 4\}$. Then $S_1 = \{e_1, e_2\}$ and $S_2 = \{e_2, e_3, e_4\}$ are acceptable subsets within a solution, however because of e_1 , there does not exist $e \in S_2 \setminus S_1$ such that $S_1 \cup \{e\}$ is an acceptable subset. This means that the *augmentation property* (cf. Section 2) is not satisfied and the subsets that define feasible solutions do *not* form a matroid.

4.3 Exponential exact algorithm

In this section, we present an exponential exact algorithm for $\text{G-DCST}(G, D, \ell, u)$, as states result (c). Recall that, as a consequence of Theorem 3 in [45] for the optimization problems L-DCMST and A-DCMST , we get that the *optimization* problem $\text{G-DCMST}(G, D, \ell, u, w)$ is $\text{W}[2]$ -hard when parameterized by $n = |V(G)|$. The importance of the algorithm below, despite its simplicity, is that it separates the complexity of the feasibility and the optimization problems.

Theorem 7. *$\text{G-DCST}(G, D, \ell, u)$ can be solved in time $O(2^m \cdot (n + m))$, where $n = |V(G)|$ and $m = |E(G)|$.*

Proof. It suffices to observe that, given a subset $S \subseteq E(G)$, one can test in time $O(n + m)$ whether S forms a spanning tree of G , and whether the constraints imposed by D, ℓ, u are satisfied by S . Because there are 2^m possible subsets to be tested, the theorem follows. \square

5 Modeling other constrained problems

As discussed in the introduction, the spanning tree problem has been investigated under the most various constraints. In this section, we show that some of them can be modeled as special cases of our problem. We have already remarked that this is the case for the CONFLICT CONSTRAINED SPANNING TREE problem. One should also notice that, in all the reductions presented below, when the feasibility version reduces to our problem, so does the optimization version. It is a matter of observing that the graph in the source instance is a subgraph of the graph in the G -DCST instance, and that the reduction preserves the solution value when keeping the weights of the initial edges and setting to zero the weights of the new edges.

In Subsection 5.1, we present a reduction from FORCING CST (denoted by FCST), in Subsection 5.2, a reduction from MAXIMUM DEGREE CST (denoted by MDST), and in Section 5.3, from MINIMUM DEGREE CST and FIXED-LEAVES MINIMUM DEGREE CST (denoted by mDST and FmDST, respectively).

5.1 Forcing constrained spanning trees

Recall that, given graphs G and D such that $V(D) = E(G)$, FCST consists in deciding whether G has a spanning tree T such that $|E(T) \cap \{e, f\}| \geq 1$ for every $ef \in E(D)$.

Theorem 8. *Denote by G -DCST* the problem G -DCST restricted to instances $(G', D', \ell, 2)$ such that $\ell(e) \in \{0, 1\}$ for every $e \in E(G')$ and $\Delta^-(D') = 2$. Then, $FCST \preceq_P G$ -DCST*.*

Proof. Let (G, D) be an instance of FCST, and construct G', D' as follows (cf. Figure 4). Choose any $v \in V(G)$, and let G' be obtained from G by adding, for each $ee' \in E(D)$, a pendant degree one vertex in v ; denote the new edge by $p_{ee'}$. Then, let D' be the digraph with vertex set $E(G')$ and arcs $(e, p_{ee'})$ and $(e', p_{ee'})$ for every $ee' \in E(D)$. Finally, let $\ell(e) = 0$ if $e \in E(G)$, and $\ell(e) = 1$ otherwise. We prove that (G, D) is a yes-instance of FCST if and only if $(G', D', \ell, 2)$ is a yes-instance of G -DCST.

First, let T be a solution for FCST, and let T' be obtained from T by adding $p_{ee'}$ for every $ee' \in E(D)$. Clearly T' is a spanning tree of G' , and since $|E(T) \cap \{e, e'\}| \geq 1$ for every $ee' \in E(D)$, we get that $|E(T') \cap \text{dep}_{D'}(p_{ee'})| \geq 1$. The reverse implication can be proved similarly. \square

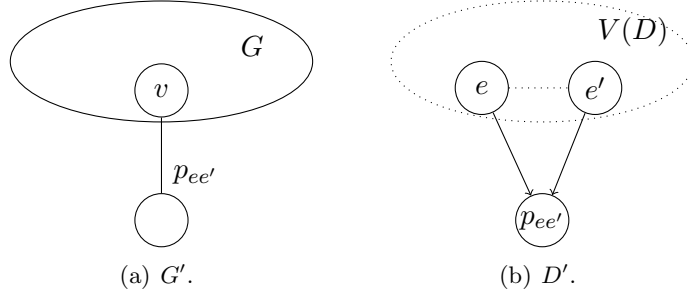


Figure 4: Illustration of the reduction for FCST.

5.2 Max-degree constrained spanning trees

Given a graph $G = (V, E)$ and a positive integer k , recall that in the $\text{MDST}(G, k)$ problem we want to find a spanning tree T such that $d_T(v) \leq k$, for every $v \in V(G)$. We prove that a generalized version of this problem reduces to ours. In $\text{MDST}(G, d^*)$, instead of being given an integer k , we are given a function $d^* : V(G) \rightarrow \mathbb{N}$ that separately sets upper bounds to the degrees of the vertices.

Theorem 9. *Denote by $G\text{-DCST}^{**}$ the problem $G\text{-DCST}$ restricted to instances $(G, D, 0, u)$. Then, $\text{MDST} \leq_P G\text{-DCST}^{**}$.*

Proof. Let (G, d^*) be an instance of MDST . We build an equivalent instance $(G', D, 0, u)$ of $G\text{-DCST}$ as follows (cf. Figure 5).

- $G' = (V \cup V', E \cup E')$, where $V' = \{v' \mid v \in V\}$ has a copy of each vertex, and $E' = \{vv' \mid v \in V\}$ connects each vertex $v \in V$ to its copy $v' \in V'$.
- $D = (E \cup E', A)$, where $A = \{(uv, uu'), (uv, vv') \mid uv \in E\}$.
- $u(e) = 0$, for each $e \in E$, while $u(vv') = d^*(v)$, for each $v \in V$.

Observe that $\text{dep}(vv')$ is the set of edges incident to v in G for every $v \in V$, and that $\text{dep}(e) = \emptyset$ for every $e \in E$. Also, note that because $\ell(e) = u(e) = 0 = |\text{dep}(e)|$ for every $e \in E$, and $\ell(e) = 0$ for every $e \in E'$, the only real constraints being imposed by D are upper bounds on the chosen dependencies for edges in E' . More specifically, we get that a spanning tree T' of G' (ℓ, u) -satisfies D if and only if $|E(T') \cap \text{dep}(vv')| \leq d^*(v)$ for every $vv' \in E' \cap E(T')$. Note that each v' has degree one in G' , which implies that every edge in E' must be part of every spanning tree of G' . Thus, we get that $(V, E_T) \subseteq G$ is a tree that satisfies the maximum degree constraints if and only if $(V \cup V', E_T \cup E')$ (ℓ, u) -satisfies D . \square

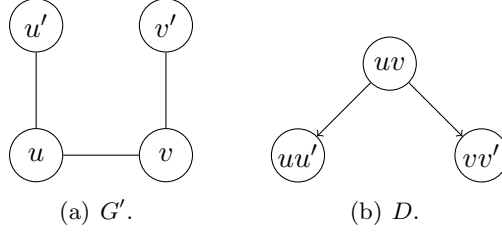


Figure 5: Illustration of the reduction for MDST.

5.3 Minimum degree constrained spanning trees

Given a graph G and a positive integer k , recall that the $\text{mDST}(G, k)$ problem consists in finding a spanning tree T of G such that $d_T(v) \geq k$ for every nonleaf vertex $v \in V(T)$. We introduce a generalized version of this problem, denoted by $\text{G-mDST}(G, \ell, u)$, where we replace the integer k by functions $\ell, u : V \rightarrow \mathbb{N}$ and require that each nonleaf vertex v of T satisfies $\ell(v) \leq d_T(v) \leq u(v)$. Clearly, $\text{mDST}(G, k)$ is a special case of $\text{G-mDST}(G, \ell, u)$ where $\ell(v) = k$ and $u(v) = d(v)$ for every $v \in V(G)$.

Theorem 10. $\text{G-mDST} \leq_P \text{G-DCST}$.

Proof. Given an instance $(G = (V, E), \ell, u)$ of G-mDST , we build an instance (G', D, ℓ', u') of G-DCST as follows (cf. Figure 6):

- $G' = (V \cup V', E \cup E')$, where $V' = \{v_1, v_2, v_3 \mid v \in V\}$ and $E' = \{vv_1, vv_2, v_1v_3, v_2v_3 \mid v \in V\}$;
- $D = (E \cup E', A_1 \cup A_2)$, where $A_1 = \{(uv, vv_1), (uv, vv_2) \mid uv \in E\}$ and $A_2 = \{(v_2v_3, v_1v_3), (v_1v_3, v_2v_3) \mid v \in V\}$;
- For each $e \in E$, let $\ell'(e) = u'(e) = 0$; and for each $v \in V$, let $\ell'(vv_1) = \ell(v)$, $u'(vv_1) = u(v)$, and $\ell'(e) = u'(e) = 1$, for each $e \in \{vv_2, v_1v_3, v_2v_3\}$.

Observe that, for each $v \in V$ and $i \in \{1, 2\}$, we have that $\text{dep}(vv_i)$ is the set of edges incident to v in G . We show that (G, ℓ, u) is a yes-instance of G-mDST if and only if (G', D, ℓ', u') is a yes-instance of G-DCST .

First, let $T = (V, E_T)$ be a solution for $\text{G-mDST}(G, \ell, u)$. We expand T into $T' = (V \cup V', E_{T'}) \subseteq G'$, where $E_{T'}$ is equal to E_T together with the following edges. For each $v \in V$, add v_1v_3 and v_2v_3 , and if v is a leaf in T then add vv_2 , otherwise add vv_1 . Observe that T' is a spanning tree of G' such that $T = T'[V]$. It remains to show that the D -dependencies are satisfied. Every edge $e \in E_T$ has $\text{dep}(e) = \emptyset$ and $\ell'(e) = u'(e) = 0$, so $\ell'(e) \leq |\text{dep}(e) \cap E(T')| \leq u'(e)$ trivially follows. The remaining types of edges in $E_{T'} \setminus E_T$ are analyzed below:

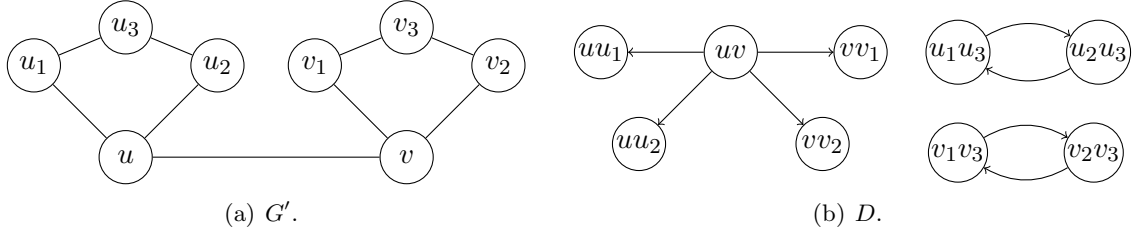


Figure 6: Illustration of the reduction for $\mathbf{G}\text{-mDST}$.

- (i) $v_i v_3$, for $v \in V$ and $i \in \{1, 2\}$: recall that $v_1 v_3$ is the unique dependency of $v_2 v_3$, and vice-versa, and they are both in T' . Hence $1 = \ell'(v_i v) \leq |\text{dep}(v_i v) \cap E(T')| \leq u'(v_i v) = 1$.
- (ii) vv_2 : by construction of T' , we get that v is necessarily a leaf in T . Since exactly one edge of $\text{dep}(vv_2)$ is in $E_{T'}$, namely the edge incident to v in T , we have that $1 = \ell'(vv_2) \leq |\text{dep}(vv_2) \cap E(T')| \leq u'(vv_2) = 1$.
- (iii) vv_1 : by construction of T' , we get that v in T is not a leaf in T . From the feasibility of T , we know that $\ell(v) \leq d_T(v) \leq u(v)$, i.e., at least $\ell(v)$ and at most $u(v)$ edges of $\text{dep}(e_1^v)$ are in $E_T \subseteq E_{T'}$, which implies that $\ell(v) = \ell'(vv_1) \leq |\text{dep}(vv_1) \cap E(T')| \leq u'(vv_1) = u(v)$.

Conversely, suppose that $T' = (V \cup V', E_{T'})$ is a solution for $\mathbf{G}\text{-DCST}(G', D, \ell', u')$. We show that $T = T'[V]$ is a solution for $\mathbf{G}\text{-mDST}(G, \ell, u)$. Due to the dependency constraints, we get that $v_1 v_3$ and $v_2 v_3$ are in T' , for each $v \in V$. From this, and since vv_1 and vv_2 are a cut in G' , exactly one of vv_1 and vv_2 is in T' , for each $v \in V$. Take $v \in V$. If vv_1 is in T' , then there are at least $\ell(v)$ and at most $u(v)$ edges incident to v in T' . And if vv_2 is in T' , then there is exactly one edge $uv \in E$ in $E_{T'}$. Therefore, either $\ell(v) \leq d_T(v) \leq u(v)$ or $d_T(v) = 1$. Since T' is a spanning tree of G' , T is a spanning tree of G , and thus T is a solution of $\mathbf{GD}\text{-MST}(G, \ell, u, w)$. \square

Finally, given a graph G , a subset $C \subseteq V$, and a function $\ell : C \rightarrow \mathbb{Z}^+$, recall that $\mathbf{FmDST}(G, C, \ell)$ denotes the problem of finding a spanning tree T of G such that $d_T(u) \geq \ell(u)$ for every $u \in C$, and $d_T(v) = 1$ for every $v \in V \setminus C$ (i.e., the set of leaves is prefixed). Observe that the same reduction of Theorem 10 can be applied to this problem by removing edge e_2^v for each $v \in C$, and edge e_1^v for each $v \in V \setminus C$. We then get the following:

Theorem 11. $\mathbf{FmDST} \preceq_{\mathbf{p}} \mathbf{G}\text{-DCST}$.

6 Conclusion

In this paper, we investigated a dependency constrained spanning tree problem that generalizes many previously studied spanning tree problems with local constraints, as for instance

degree constraints. We then inherit all of the NP-completeness results for these problems, as well as polynomial results and practical approaches to our problem will therefore hold for these other problems. Interestingly, there are other spanning tree problems that impose global constraints on the tree, as for instance, a bound on the diameter of the produced tree [10, 11], or on the number of leaves [25, 35]. A good question therefore is whether problems with this kind of constraints can be modeled within our framework.

Question 1. *Can instances of CST problems with global constraints be modeled as G-DCST instances?*

Concerning NP-completeness results, we have investigated the problem under restrictions on the dependency digraph D , and on the lower and upper bound functions ℓ and u . Among other restrictions, we have proved that G-DCST is NP-complete when D is either a forest of directed paths, a forest of out-stars, or a forest of in-stars, and each component has at most three vertices. In the first two cases, we have considered all possible values for constant functions ℓ, u . The following cases for in-stars remain open.

Question 2. *What is the complexity of G-DCST(G, D, ℓ, u) when D is a forest of in-stars with at most three vertices and $(\ell, u) = (0, 1)$ or, for every $e \in E(G)$ such that $\text{dep}(e) \neq \emptyset$, $\ell(e) = 1$ and $u(e) \in \{1, 2\}$?*

Concerning positive results, we have proved that G-DCST($G, D, \text{dep}, \text{dep}$) can be solved in polynomial time when D is an oriented matching. We ask whether this also holds for the optimization problem (recall that G-DCMST($G, D, 0, 0$) is polynomial in this case [18]).

Question 3. *What is the complexity of G-DCMST($G, D, \text{dep}, \text{dep}$) when D is an oriented matching?*

Finally, we have proved that G-DCMST can be solved in polynomial time when $\ell = 0$, D is a collection of symmetric graphs D_1, \dots, D_k , and $u(e) = u(e')$ whenever e, e' are within the same component, and that G-DCST(G, D, ℓ, u) can be solved in time $O(2^m \cdot (n + m))$ by a naive brute-force algorithm, where $n = |V(G)|$ and $m = |E(G)|$. The latter result is important in the face of the fact that, as a byproduct of a result in [45], we get that no algorithm running in time $2^{O(n)}$ exists for the optimization problem, unless ETH fails. Also, the results presented in Section 3.1 imply that no algorithm that runs in time $2^{o(n+m)}$ exists for the feasibility problem under the ETH, which means that the algorithm presented in Section 4.3 is asymptotically optimal. We mention that our algorithm can also be seen as an FPT algorithm parameterized by m . We ask whether the problem is FPT under other parameters.

Question 4. *Under which parameters is G-DCST or G-DCMST FPT?*

In order to identify parameters for the above question, note that the maximum degree of the input graph G is *not* enough, since a particular case of the problem is already NP-complete restricted to graphs with maximum degree at most three [45]. Similarly, the

maximum degree of the dependency graph D is *not* enough either, as another particular case of the problem is NP-complete even if D is a forest of paths of length at most two (see e.g. [19], as well as our results presented in Section 3.1).

A promising candidate parameter for obtaining FPT algorithms is the *treewidth* of the input graph G (see [16] for the definition); note that the treewidth of the underlying graph of D is *not* enough by the last sentence of the above paragraph, since forests have treewidth one. Suppose that, in order to use Courcelle’s Theorem [15] or any of its optimization variants [4], one tries to express the G-DCST problem in *monadic second-order* (MSO) logic. In order to guarantee that the dependencies of D are satisfied for every edge e of the desired spanning tree of the input graph G , one would probably need to *evaluate* the functions $\ell(e)$ and $u(e)$ *inside* the eventual MSO formula, and this seems to be a fundamental hurdle since these values are a priori unrelated to the treewidth of G . Nevertheless, for the particular case of G-DCST (or G-DCMST) where both functions ℓ and u are *constants* (or even equal to some constant value that depends on the treewidth of G), it is indeed possible, using standard techniques, to write such an MSO formula expressing the problem, and therefore it is FPT parameterized by the treewidth of the input graph. Note that this restriction of the G-DCST problem is NP-complete by Corollary 2, for every pair of positive integers ℓ, u with $\ell \leq u$. The next natural step would be to consider as parameters *both* the treewidth of G and the maximum degree of D .

References

- [1] Ibrahim Akgün and Barbaros Tansel. Min-degree constrained minimum spanning tree problem: New formulation via Miller–Tucker–Zemlin constraints. *Computers & Operations Research*, 37(1):72–82, 2010.
- [2] Ana Maria Almeida, Pedro Martins, and Maurício C de Souza. Min-degree constrained minimum spanning tree problem: complexity, properties, and formulations. *International Transactions in Operational Research*, 19(3):323–352, 2012.
- [3] Ana Maria Almeida, Pedro Martins, and Maurício C Souza. md-MST is NP-hard for $d \geq 3$. *Electronic Notes in Discrete Mathematics*, 36:9–15, 2010.
- [4] Stefan Arnborg, Jens Lagergren, and Detlef Seese. Easy problems for tree-decomposable graphs. *Journal of Algorithms*, 12(2):308–340, 1991.
- [5] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.
- [6] Rom Aschner and Matthew J. Katz. Bounded-angle spanning tree: Modeling networks with angular constraints. *Algorithmica*, 77(2):349–373, 2017.

- [7] Julien Baste, Didem Gözüpek, Christophe Paul, Ignasi Sau, Mordechai Shalom, and Dimitrios M. Thilikos. Parameterized complexity of finding a spanning tree with minimum reload cost diameter. *Networks*, 75(3):259–277, 2020.
- [8] Luis Henrique Bicalho, Alexandre Salles da Cunha, and Abilio Lucena. Branch-and-cut-and-price algorithms for the degree constrained minimum spanning tree problem. *Computational Optimization and Applications*, 63(3):755–792, 2016.
- [9] Hans L. Bodlaender and Klaus Jansen. On the complexity of scheduling incompatible jobs with unit-times. In A.M. Borzyszkowski and S. Sokółowski, editors, *MFCSS - International Symposium on Mathematical Foundations of Computer Science*, pages 191–300. Springer, Berlin, Heidelberg, 1993.
- [10] Paolo M. Camerini, Giulia Galbiati, and Francesco Maffioli. Complexity of spanning tree problems: Part I. *European Journal of Operational Research*, 5(5):346–352, 1980.
- [11] Paolo M. Camerini, Giulia Galbiati, and Francesco Maffioli. On the complexity of finding multi-constrained spanning trees. *Discrete Applied Mathematics*, 5(1):39–50, 1983.
- [12] Francesco Carrabs, Carmine Cerrone, and Rosa Pentangelo. A multiethnic genetic approach for the minimum conflict weighted spanning tree problem. *Networks*, 74(2):134–147, 2019.
- [13] Francesco Carrabs, Raffaele Cerulli, Rosa Pentangelo, and Andrea Raiconi. Minimum spanning tree with conflicting edge pairs: a branch-and-cut approach. *Annals of Operations Research*, 2018.
- [14] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM Symposium on Theory of computing*, pages 151–158, 1971.
- [15] Bruno Courcelle. The Monadic Second-Order Logic of Graphs. I. Recognizable Sets of Finite Graphs. *Information and Computation*, 85(1):12–75, 1990.
- [16] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
- [17] Alexandre Salles da Cunha and Abilio Lucena. Modeling and solving the angular constrained minimum spanning tree problem. *Comput. Oper. Res.*, 112, 2019.
- [18] Andreas Darmann, Ulrich Pferschy, and Joachim Schauer. Minimum spanning trees with conflict graphs. *Optimization online*, 2009.

- [19] Andreas Darmann, Ulrich Pferschy, Joachim Schauer, and Gerhard J. Woeginger. Paths, trees and matchings under disjunctive constraints. *Discrete Applied Mathematics*, 159(16):1726–1735, 2011.
- [20] N. Deo and S. Louis Hakimi. The shortest generalized Hamiltonian tree. In *Proceedings of the 6th Annual Allerton Conference, vol. 17(3)*, pages 409–423, 1968.
- [21] Narsingh Deo and Nishit Kumar. Computation of constrained spanning trees: A unified approach. In *Network Optimization. Lecture Notes in Economics and Mathematical Systems, vol. 450*, pages 196–220. Springer, Berlin, Heidelberg, 1997.
- [22] Fabio C. S. Dias, Manoel Campêlo, Críston Souza, and Rafael Andrade. Min-degree constrained minimum spanning tree problem with fixed centrals and terminals: Complexity, properties and formulations. *Computers & Operations Research*, 84:46–61, 2017.
- [23] Jack Edmonds. Matroid intersection. *Annals of discrete Mathematics*, 4:39–49, 1979.
- [24] Leah Epstein, Lene M. Favrholdt, and Asaf Levin. Online variable-sized bin packing with conflicts. *Discrete Optimization*, 8(2):333–343, 2011.
- [25] Michael R. Garey and David S. Johnson. *Computers and intractability*. Freeman, San Francisco, 1979.
- [26] Frank Gurski and Carolin Rehs. Solutions for the knapsack problem with conflict and forcing graphs of bounded clique-width. *Mathematical Methods of Operations Research*, 89:411–432, 2019.
- [27] Refael Hassin and Arie Tamir. On the minimum diameter spanning tree problem. *Information processing letters*, 53(2):109–111, 1995.
- [28] Jan-Ming Ho, DT Lee, Chia-Hsiang Chang, and CK Wong. Minimum diameter spanning trees and related problems. *SIAM Journal on Computing*, 20(5):987–997, 1991.
- [29] Russell Impagliazzo and Ramamohan Paturi. On the Complexity of k -SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.
- [30] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which Problems Have Strongly Exponential Complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.
- [31] Mamadou Moustapha Kanté, Christian Laforest, and Benjamin Momège. Trees in graphs with conflict edges or forbidden transitions. In *Proc. of the 10th International Conference on Theory and Applications of Models of Computation (TAMC)*, volume 7876 of *LNCS*, pages 343–354, 2013.

- [32] Jochen Könemann, Asaf Levin, and Amitabh Sinha. Approximating the degree-bounded minimum diameter spanning tree problem. *Algorithmica*, 41(2):117–129, 2005.
- [33] Mohan Krishnamoorthy, Andreas T. Ernst, and Yazid M. Sharaiha. Comparison of algorithms for the degree constrained minimum spanning tree. *Journal of Heuristics*, 7(6):587–611, 2001.
- [34] Eugene L. Lawler. Combinatorial optimization: Networks and matroids. *New York*, 1976.
- [35] Hsueh-I Lu and R Ravi. The power of local optimization: Approximation algorithms for maximum-leaf spanning tree. In *Proceedings of the Annual Allerton Conference on Communication Control and Computing, vol. 30*, pages 533–533, 1992.
- [36] Slvia Maria Santana Mapa and Sebastin Urrutia. On the maximum acyclic subgraph problem under disjunctive constraints. *Information Processing Letters*, 115(2):119–124, 2015.
- [37] Leonardo Conegundes Martinez and Alexandre Salles da Cunha. The min-degree constrained minimum spanning tree problem: Formulations and branch-and-cut algorithm. *Discrete Applied Mathematics*, 164:210–224, 2014.
- [38] James G. Oxley. *Matroid theory*. Oxford University Press, 1992.
- [39] Christos H. Papadimitriou and Umesh V Vazirani. On two geometric problems related to the travelling salesman problem. *Journal of Algorithms*, 5(2):231–246, 1984.
- [40] Ulrich Pferschy and Joachim Schauer. The maximum flow problem with disjunctive constraints. *J. Combinatorial Optimization*, 26:109–119, 2013.
- [41] Gabriel Robins and Jeffrey S. Salowe. Low-degree minimum spanning trees. *Discrete & Computational Geometry*, 14(2):151–165, 1995.
- [42] Phillippe Samer and Sebastián Urrutia. A branch and cut algorithm for minimum spanning trees under conflict constraints. *Optimization Letters*, 9(1):41–55, 2015.
- [43] Mohit Singh and Lap Chi Lau. Approximating minimum bounded degree spanning trees to within one of optimal. *J. ACM*, 62(1):1:1–1:19, 2015.
- [44] Luiz Alberto Viana. *Árvore geradora com dependências mínima*. Master’s thesis, Federal University of Ceará, 2016.
- [45] Luiz Alberto Viana and Manoel Campêlo. Two dependency constrained spanning tree problems. *International Transactions in Operational Research*, 27(2):867–898, 2020.

- [46] Douglas B. West. *Introduction to graph theory*. Prentice Hall Upper Saddle River, NJ, 1996.
- [47] Ruonan Zhang, Santosh N. Kabadi, and Abraham P. Punnen. The minimum spanning tree problem with conflict constraints and its variations. *Discrete Optimization*, 8(2):191–205, 2011.