Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 346 (2019) 711–723

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

The Generalized Dependency Constrained Spanning Tree Problem

Luiz Alberto do Carmo Viana [1,3]

Campus de Crateús
Universidade Federal do Ceará
Crateús, Brazil

Manoel Campêlo [1,2,4]

Departamento de Estatística e Matemática Aplicada
Universidade Federal do Ceará
Fortaleza, Brazil

Abstract

We introduce the Generalized Dependency Constrained Spanning Tree Problem (`G-DCST`), where an edge can be chosen only if the number of edges chosen from its dependency set lies in a certain interval. The dependency relations between the edges of the input graph G are described by the input digraph D, whose vertices are the edges of G. The in-neighbors of a vertex of D form its dependency set. We show that `G-DCST` unifies and generalizes some known spanning tree problems that apply conflict constraints over edges or lower and upper bounds on vertex degrees. We show that the feasibility version of `G-DCST` is NP-complete even under strong restrictions on the structures of G and D as well as on the functions that define the minimum or maximum number of dependencies to be satisfied. We also show that this problem keeps an $\ln n$ inapproximability threshold under tight assumptions over G and D. On the other hand, we spot a polynomial case via a matroid intersection argument.

Keywords: Dependency Constrained Spanning Tree Problem; NP-hardness; Innaproximability.

1 Introduction

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 if $(e_1, e_2) \in A$. For each $e \in E$, we define its D-dependency set as $dep_D(e) = \{e' \in E : (e', e) \in A\}$. Also, let $l, u : E \to \mathbb{N}$ be functions from the edge set of G to the naturals (we consider that

[1] Partially supported by CNPq-Brazil (Proc. 443747/2014-8) and FUNCAP (PNE-0112-00061.01.00/16)

[2] Partially supported by CNPq-Brazil (Proc. 305264/2016-8)

[3] Email: `luizalberto@crateus.ufc.br`

[4] Email: `mcampelo@lia.ufc.br`

https://doi.org/10.1016/j.entcs.2019.08.062
1571-0661/

$0 \in \mathbb{N}$). We say that a subgraph $H \subseteq G$ of G (l, u)-satisfies D if, for each $e \in E(H)$, $l(e) \leq |dep_D(e) \cap E(H)| \leq u(e)$.

Let $\mathcal{T}_G$ denote the set of all spanning trees of G. We define the Generalized Dependency Constrained Spanning Tree Problem, abbreviated as `G-DCST`(G, D, l, u), as the problem of deciding if there is a $T \in \mathcal{T}_G$ such that T (l, u)-satisfies D. Let $w : E \to \mathbb{R}^+$ be a weighting function over the edge set of G. For each subgraph $H \subseteq G$, we define $w(H) = \sum_{e \in E(H)} w_e$. We can define the Generalized Dependency Constrained Minimum Spanning Tree Problem, which we abbreviate as `G-DCMST`(G, D, l, u, w), as the problem of finding, among the $T \in \mathcal{T}_G$ that (l, u)-satisfy D, a T^* with $w(T^*)$ minimum.

The next sections develop results concerning `G-DCST` and `G-DCMST`. Section 2 establishes relations between the present problems and generalized versions of spanning tree problems found in the literature. Section 3 exposes complexity results for `G-DCMST`(G, D, l, u, w), parameterized either by G and D or mainly by l and u. Finally, Section 4 ends the paper with comments and directions of future work.

2 Related spanning tree problems

This section exposes relations between `G-DCST` and other spanning tree problems. We show that it unifies and generalizes a number of known problems. As of particular interest, we establish a relation between `G-DCST` and the Conflict Constrained Spanning Tree Problem, which could be seen as its counterpart where the digraph D is replaced by an undirected graph.

2.1 Previous dependency constrained problems

In a previous work [15], we have introduced and tackled problems similar to `G-DCST`. We have defined the Least-Dependency Constrained Spanning Tree Problem, abbreviated as `L-DCST`(G, D), which consists of deciding whether there is a $T \in \mathcal{T}_G$ such that, for each $e \in E(T)$ with $dep_D(e) \neq \emptyset$, *at least one* D-dependency of E is also in T (i.e. $dep_D(e) \cap E(T) \neq \emptyset$ if $dep_D(e) \neq \emptyset$). Also, we have introduced the All-Dependency Constrained Spanning Tree Problem, abbreviated as `A-DCST`(G, D). It consists in deciding whether there is a $T \in \mathcal{T}_G$ such that, for each $e \in E(T)$, *all* D-dependencies of e are also in T (i.e. $dep_D(e) \cap E(T) = dep_D(e)$).

Theorem 2.1 *`L-DCST`(G, D) and `A-DCST`(G, D) are particular cases of `G-DCST`(G, D, l, u).*

Proof. An instance (G, D) of `L-DCST` is equivalent to an instance (G, D, l, u) of `G-DCST`, where $l(e) = \min\{|dep_D(e)|, 1\}$ (if $dep_D(e) = \emptyset$, e can freely take part in a solution) and $u(e) = |dep_D(e)|$, for all $e \in E(G)$. Similarly, an instance (G, D) of `A-DCST` is equivalent to an instance (G, D, l, u) of `G-DCST`, where $l(e) = u(e) = |dep_D(e)|$, for all $e \in E(G)$. $\square$

As `G-DCST` generalizes both problems, it inherits their applications. For instance, dependency relations can model communication systems with protocol conversion restrictions [14]. Besides, in Section 3, we profit from Theorem 2.1 and the computational complexity study presented in [15] to get hardness results for G-DCST.

The optimization versions of L-DCST(G, D) and A-DCST(G, D) were denoted as L-DCMST(G, D, w) and A-DCMST(G, D, w), respectively, where $w : E(G) \to \mathbb{R}^+$ is a weighting function over the edge set of G. Those problems consist of finding a solution with minimum weight, according to w, among the solutions of their decision counterparts. As a consequence of Theorem 2.1, they are particular cases of G-DCMST(G, D, l, u, w). In [15], we have evaluated the computational performance of a branch-and-bound algorithm for L-DCMST(G, D, w) and A-DCMST(G, D, w), which is based on node selection and branching strategies.

2.2 Conflict Constrained Minimum Spanning Tree Problem

Let $G = (V, E)$ be an undirected graph. Also, let $G_c = (E, E_c)$ be a graph whose vertices are the edges of G. We say that $e_1, e_2 \in E$ are (G_c)-conflicting if $\{e_1, e_2\} \in E_c$. We call $H \subseteq G$ a (G_c)-conflicting free subgraph if no $e_1, e_2 \in E(H)$ are (G_c)-conflicting.

Given $w : E \to \mathbb{R}^+$, the Conflict Constrained Minimum Spanning Tree Problem, abbreviated here as CCMST(G, G_c, w), consists of finding a (G_c)-conflicting free $T^* \in \mathcal{T}_G$ with $w(T^*)$ minimum. This problem was introduced in [4]. It was shown to be strongly NP-hard, even if G_c is the disjoint union of paths of lenght 2 [4], and it has a polynomial case when G_c is the disjoint union of cliques [16]. Besides, the problem is NP-hard even if G is a cactus [16]. Experiments with heuristics and a branch-and-cut algorithm were shown in [16] and [12], respectively. G-DCMST also generalizes this problem.

Theorem 2.2 *CCMST(G, G_c, w) is a particular case of G-DCMST(G, D, l, u, w).*

Proof. Given an instance (G, G_c, w) of CCMST, we build an instance (G', D, l, u, w) of G-DCMST in the following way: first, we make $G' = G$ and $D = (E(G'), A)$, where $A = \{(e_1, e_2), (e_2, e_1) : \{e_1, e_2\} \in E(G_c)\}$; for each $e \in E(G')$, $l(e) = u(e) = 0$. The construction of D is illustrated in Figure 1.

Now, we prove that T is a solution for CCMST(G, G_c, w) if, and only if, T is a solution for G-DCMST(G', D, l, u, w).

Suppose that T is a solution for CCMST(G, G_c, w). This means that T is a spanning tree of G and no $e_1, e_2 \in E(T)$ are (G_c)-conflicting. Considering D, this implies that, for each $e_1, e_2 \in E(T)$, $e_1 \notin dep_D(e_2)$ and $e_2 \notin dep_D(e_1)$. Then, for each $e \in E(T)$, we have $dep_D(e) \cap E(T) = \emptyset$, leading to $l(e) \leq |dep_D(e) \cap E(T)| \leq u(e)$. From this, we see that T (l, u)-satisfies D, so T is a solution for G-DCMST(G', D, l, u, w).

Conversely, take T as a solution for G-DCMST(G', D, l, u, w). This means that T is a spanning tree of G' and T (l, u)-satisfies D. By definition, for each $e \in E(T)$, we have $dep_D(e) \cap E(T) = \emptyset$. Considering the construction of D, for each $e_1, e_2 \in E(T)$, e_1 and e_2 are not (G_c)-conflicting, thus T is (G_c)-conflicting free. From this, T is a solution for CCMST(G, G_c, w).

This concludes that CCMST(G, G_c, w) and G-DCMST(G', D, l, u, w) have the same solution set. Since both instances are weighted by w, they have the same optimal value. □

(a) G_c. (b) D.

Fig. 1. Illustration of the CCMST reduction.

2.3 Degree Constrained Spanning Tree Problems

Let $G = (V, E)$ be a graph. We denote by $d_G(v), v \in V$, the degree of vertex v in G, that is, the number of vertices that are adjacent to v in G. Also, we call $v \in V$ a leaf in G if $d_G(v) = 1$. For $V' \subseteq V$, let $G[V']$ be the subgraph of G induced by V'.

We present relations between G-DCMST and spanning tree problems characterized by degree constraints over their spanning tree solutions. Such constaints are expressed in terms of lower or upper bounds for the vertex degrees in the tree.

2.3.1 Max-Degree Constrained Minimum Spanning Tree

Let $G = (V, E)$ be a graph. Given $k : V \to \mathbb{N}$ and $w : E \to \mathbb{R}^+$, we present the classical Max-Degree Constrained Minimum Spanning Tree Problem, abbreviated here as MaxDeg-MST(G, k, w). It consists of finding, among the $T \in \mathcal{T}_G$ such that $d_T(v) \leq k(v)$, for each $v \in V$, a T^* with minimum $w(T^*)$. This NP-hard problem was first introduced in [11]. Since then, it has been extensively studied. Several heuristic, approximation and exact approaches have been proposed for the problem (see for example [8,13,3] and references therein).

Theorem 2.3 *MaxDeg-MST(G, k, w) is a particular case of G-DCMST(G, D, l, u, w).*

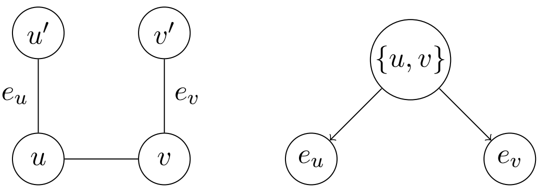
Proof. Let (G, k, w) be an instance of MaxDeg-MST. We build an instance (G', D, l, u, w') of G-DCMST as follows: $G' = (V \cup V', E \cup E')$, where $V' = \{v' : v \in V\}$ is a set of artificial vertices, one for each vertex in V, and $E' = \{e_v = \{v, v'\} : v \in V\}$ is a set of artificial cut edges, each one linking an original vertex $v \in V$ and its corresponding artificial vertex $v' \in V'$; $D = (E \cup E', A)$, where $A = \{(\{u, v\}, e_u), (\{u, v\}, e_v) : \{u, v\} \in E\}$; $l(e) = u(e) = 0$, for each $e \in E$, while $l(e_v) = 0$ and $u(e_v) = k(v)$, for each $v \in V$; at last, $w'_e = w_e$, for each $e \in E$, and $w'_e = 0$, for each $e \in E'$. This construction is illustrated in Figure 2. In particular, note that $dep_D(e_v)$ is the set of edges incident to v in G, for all $v \in V$.

We show that T is a solution for MaxDeg-MST(G, k, w) if, and only if, there is a solution T' for G-DCMST(G', D, l, u, w') with $T = T'[V]$.

First, let $T = (V, E_T)$ be a solution for MaxDeg-MST(G, k, w). We expand T into $T' = (V \cup V', E_{T'}) \subseteq G'$, where $E_{T'} = E_T \cup E'$. We need to show that T' is a solution for G-DCMST(G', D, l, u, w'). Since T is a spanning tree of G, it is clear that T' is a spanning tree of G'. Each edge $e \in E_T$ has $dep_D(e) = \emptyset$, so $l(e) = u(e) = 0$ implies that $l(e) \leq |dep_D(e) \cap E(T')| \leq u(e)$. Now, let $e \in E_{T'} \setminus E_T = E'$. Then, $e = e_v$ for some $v \in V$ and, by the feasibility of T, $d_T(v) \leq k(v)$. This implies that at most $k(v)$ edges of $dep_D(e_v)$ are in $E_{T'}$. From this, it follows that $l(e_v) \leq |dep_D(e_v) \cap E(T')| \leq u(e_v)$. Therefore, T' (l, u)-satisfies D and we have that T' is a solution for $\texttt{G-DCMST}(G', D, l, u, w')$.

Conversely, take $T' = (V \cup V', E_{T'})$ as a solution for $\texttt{G-DCMST}(G', D, l, u, w')$. We show that $T = T'[V]$ is a solution for $\texttt{MaxDeg-MST}(G, k, w)$. Let $v \in V$. As e_v is a cut edge in G', then $e_v \in E_{T'}$. By the construction of D and function u, this implies that there are at most $k(v)$ edges $\{u, v\} \in E$ in $E_{T'}$. Since T is the subtree of T' induced by V, it follows that $d_T(v) \leq k(v)$. Moreover, since T' is a spanning tree of G', T is a spanning tree of G. Thus T is a solution for $\texttt{MaxDeg-MST}(G, k, w)$.

To finish the proof, notice that, since the edges in E' have zero weight, corresponding solutions of $\texttt{MaxDeg-MST}(G, k, w)$ and $\texttt{G-DCMST}(G', D, l, u, w')$ have the same weight. This concludes that $\texttt{MaxDeg-MST}(G, k, w)$, and $\texttt{G-DCMST}(G', D, l, u, w')$ have the same optimum value. □



(a) G'.

(b) D.

Fig. 2. Illustration of MaxDeg-MST reduction.

2.3.2 Generalized Degree Constrained Minimum Spanning Tree

Let $G = (V, E)$ be a graph. Given $\kappa \in \mathbb{N}$ and $w : E \to \mathbb{R}^+$, the Min-Degree Constrained Minimum Spanning Tree Problem, abbreviated here as $\texttt{MD-MST}(G, \kappa, w)$, consists of finding, among the $T \in \mathcal{T}_G$ such that each nonleaf v of T has $d_T(v) \geq \kappa$, a T^* with minimum $w(T^*)$. This problem was introduced in [2], where it was shown to be NP-hard. Integer linear programs and solution methods were proposed in [1,2,10].

Let us introduce a generalized version of this problem, to be denoted $\texttt{GD-MST}(G, k', k, w)$, where we replace the scalar κ by functions $k, k' : V \to \mathbb{N}$ and require each nonleaf v of T to satisfy $k'(v) \leq d_T(v) \leq k(v)$.

We are about to expose a relation between GD-MST and G-DCMST. Before proceeding, we denote by $N_G(v) = \{u \in V(G) : \{u, v\} \in E(G)\}$ the set of vertices that are adjacent to v in G.

Theorem 2.4 *$\texttt{GD-MST}(G, k', k, w)$ is a particular case of $\texttt{G-DCMST}(G, D, l, u, w)$.*

Proof. Given an instance $(G = (V, E), k', k, w)$ of GD-MST, we build an instance (G', D, l, u, w') of G-DCMST in the following way: first, $G' = (V \cup V', E \cup E')$, where $V' = \{v_1, v_2, v_3 : v \in V\}$ and $E' = \{e_1^v = \{v, v_1\}, e_2^v = \{v, v_2\}, e_3^v = \{v_1, v_3\}, e_4^v = \{v_2, v_3\} : v \in V\}$; we make $D = (E \cup E', A)$, where $A = A_1 \cup A_2$, $A_1 = \{(\{u, v\}, e_1^v), (\{u, v\}, e_2^v) : v \in V, u \in N_G(v)\}$ and $A_2 = \{(e_3^v, e_4^v), (e_4^v, e_3^v) : v \in V\}$; for each $e \in E$, $l(e) = u(e) = 0$; for each $v \in V$, $l(e_1^v) = k'(v)$, $u(e_1^v) = k(v)$,

$l(e_i^v) = u(e_i^v) = 1$, $2 \leq i \leq 4$; at last, $w'_e = w_e$, if $e \in E$, and $w'_e = 0$, otherwise. This construction is illustrated in Figure 3. Observe that $dep_D(e_1^v) = dep_D(e_2^v)$ is the set of edges incident to v, for every $v \in V$.

We show that T is a solution for $\texttt{GD-MST}(G, k', k, w)$ if, and only if, there is a solution T' for $\texttt{G-DCMST}(G', D, l, u, w')$ with $T = T'[V]$.

First, let $T = (V, E_T)$ be a solution for $\texttt{GD-MST}(G, k', k, w)$. 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 $v \in V$, e_3^v and e_4^v; for each $v \in V$, either e_2^v or e_1^v, depending whether v is a leaf in T or not, respectively. Clearly, $T = T'[V]$. It remains to show that T' is a solution for $\texttt{G-DCMST}(G', D, l, u, w')$. Since T is a spanning tree of G and, for each $v \in V$, exactly three of $e_i^v, i \in [4]$, are in $E_{T'}$, T' is a spanning tree of G'. Now, we show that the D-dependencies are satisfied. Every edge $e \in E_T$ has $dep_D(e) = \emptyset$ and $l(e) = u(e) = 0$, so $l(e) \leq |dep_D(e) \cap E(T')| \leq u(e)$ trivially follows. The remaining edges in $E_{T'} \setminus E_T$ can be grouped as follows:

(i) e_3^v and e_4^v, for each $v \in V$: $1 = l(e_i^v) \leq |dep_D(e_i^v) \cap E(T')| \leq u(e_i^v) = 1$, $3 \leq i \leq 4$, is immediate because e_3^v is the unique dependency of e_4^v, and vice-versa;

(ii) e_2^v, for each leaf v in T: since exactly one edge of $dep_D(e_2^v)$ is in $E_{T'}$, we have that $1 = l(e_2^v) \leq |dep_D(e_2^v) \cap E(T')| \leq u(e_2^v) = 1$;

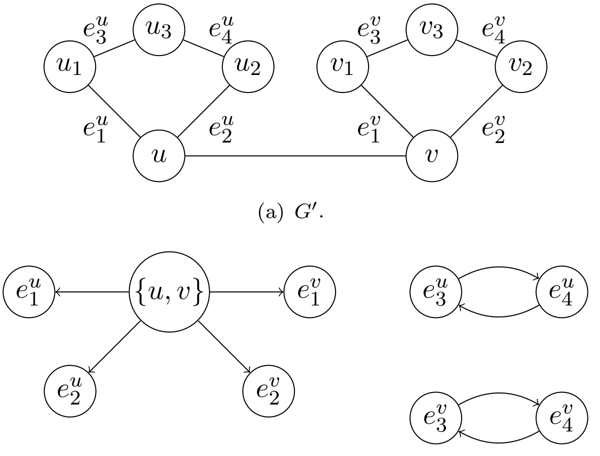
(iii) e_1^v, for each nonleaf v in T: from the feasibility of T, $k'(v) \leq d_T(v) \leq k(v)$, i.e. at least $k'(v)$ and at most $k(v)$ edges of $dep_D(e_1^v)$ are in $E_T \subseteq E_{T'}$, which implies that $k'(v) = l(e_1^v) \leq |dep_D(e_1^v) \cap E(T')| \leq u(e_1^v) = k(v)$.

Therefore, T' (l, u)-satisfies D, and we have that T' is a solution for $\texttt{G-DCMST}(G', D, l, u, w')$.

Conversely, suppose that $T' = (V \cup V', E_{T'})$ is a solution for $\texttt{G-DCMST}(G', D, l, u, w')$. We show that $T = T'[V]$ is a solution for $\texttt{GD-MST}(G, k', k, w)$. Due to dependency constraints, e_3^v and e_4^v are in T', for each $v \in V$. From this, and since e_1^v and e_2^v are a cut in G', exactly one of e_1^v and e_2^v is in T', for each $v \in V$. Take $v \in V$. If e_1^v is in T', then there are from $k'(v)$ to $k(v)$ edges $\{u, v\} \in E$ in $E_{T'}$. If e_2^v is in T', then there is exaclty one edge $\{u, v\} \in E$ in $E_{T'}$. Therefore, either $k'(v) \leq d_T(v) \leq k(v)$ 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 $\texttt{GD-MST}(G, k', k, w)$.

To finish the proof, observe that corresponding solutions of $\texttt{GD-MST}(G, k', k, w)$ and $\texttt{G-DCMST}(G', D, l, u, w')$ have the same weight, since E' edges have zero weight. This concludes that $\texttt{GD-MST}(G, k', k, w)$ and $\texttt{G-DCMST}(G', D, l, u, w')$ have the same optimum value. □

An interesting variation of $\texttt{MD-MST}(G = (V, E), k, w)$ is obtained when the leaves are fixed a priori. Precisely, given $C \subseteq V$ and $d : C \to \mathbb{Z}^+$, it consists in finding a T^* among the $T \in \mathcal{T}_G$ such that $d_T(u) \geq d(u)$, for each $u \in C$, and $d_T(v) = 1$, for each $v \in V \setminus C$, with $w(T^*)$ minimum. This problem is abbreviated as $\texttt{FMD-MST}(G, C, d, w)$ and was introduced in [5]. Similarly, let us introduce a generalized version, to be denoted $\texttt{FGD-MST}(G, C, d', d, w)$, where we add a lower bounding function $d' : V \to \mathbb{N}$



(b) D.

Fig. 3. Illustration of MD-MST reduction.

and require each vertex $v \in C$ to satisfy $d'(u) \leq d_T(u)$ in any feasible tree T (besides the constraints of FMD-MST).

We can prove that FGD-MST(G, C, d', d, w) is also particular case of G-DCMST(G, D, l, u, w) with a little modification in the construction made in the proof of Theorem 2.4. Given G' built as described in Theorem 2.4, we build G'' removing edges from G': for each $v \in C$, we remove the edge e_2^v; for each $v \in V \setminus C$, we remove the edge e_1^v. An obvious adaptation on D is also required. An argument similar to the proof of Theorem 2.4 is sufficient to conclude the following theorem.

Theorem 2.5 *FGD-MST(G, C, d', d, w) is a particular case of G-DCMST(G, D, l, u, w).*

3 Complexity

3.1 Analysis in terms of G and D

In this subsection, we analyse the complexity of G-DCST(G, D, l, u) and G-DCMST(G, D, l, u) as a function of the input graphs G and D. We show NP-completeness results for G-DCST. Also, we establish an inapproximability threshold for G-DCMST.

The following results were proven in [15]. Notice that the hardness holds for very limiting conditions over G and D.

Theorem 3.1 *L-DCST(G, D) and A-DCST(G, D) are NP-complete, even if G is a chordal graph whose diameter is 2, and D is a union of arborescences of height 2 or an arborescence of height 3.*

Theorem 3.2 *L-DCST(G, D) and A-DCST(G, D) are NP-complete, even if G is a chordal graph with $\Delta(G) \leq 3$, and D is a union of arborescences of height 2 or an arborescence of height 3.*

As a direct consequence of theorems 2.1, 3.1 and 3.2, we can establish the NP-completeness of `G-DCST` as follows. The last observation is due to the proof of Theorem 2.1.

Corollary 3.3 *`G-DCST`(G, D, l, u) is NP-complete, even if G and D are under the assumptions of either Theorems 3.1 or 3.2. This result holds even when $l = u$.*

It is remarkable that, since `G-DCST` is NP-complete, `G-DCMST` is inapproximable. The same holds for `L-DCMST` and `A-DCMST`. Actually, in [15] we prove that `L-DCMST` and `A-DCMST` keep an $\ln(n)$ inapproximability threshold for very restricting assumptions. This result is presented here in the following theorem.

Theorem 3.4 *`L-DCMST`(G, D, w) and `A-DCMST`(G, D, w) are APX-hard, not being approximable to $(1 - \Omega(1)) \ln(|V(G)|)$ unless $P = NP$. Moreover, they are W[2]-hard parameterized by the cost of the solution tree. The results hold even if G is bipartite, the dependency relations occur only between adjacent edges of G, and each weak component of D has diameter 1.*

Based on Theorem 2.1, it is easy to see that `L-DCMST`(G, D, w) and `A-DCMST`(G, D, w) are particular cases of `G-DCMST`(G, D, l, u, w). From this, we have the following corollary.

Corollary 3.5 *Theorem 3.4 is valid for `G-DCMST`(G, D, l, u, w), with G and D under the same assumptions. This result holds even when $l = u$.*

3.2 Analysis in terms of l and u

In this subsection, `G-DCMST`(G, D, l, u, w) is examined by mainly focusing on the functions l and u. By following this approach, we attempt to obtain a deeper understanding on the hardness of `G-DCMST` and spot cases where this problem could be treated in a reasonable amount of time. In particular, a polynomial case for `G-DCMST` is identified via matroid intersection.

3.2.1 $l = 0$

When we take instances (G, D, l, u, w) of `G-DCMST` where $l(e) = 0$, for each $e \in E(G)$, we allow the inclusion of an edge $e \in E(G)$ together with at most $u(e)$ of its D-dependencies. It is natural to think that, for this kind of instance, `G-DCMST` seems to be a "weaker" version of `CCMST`, since in the latter problem an edge is allowed only together with none of its relatives. From this, one could imagine a possible relation between `CCMST` and `G-DCMST` under $l = 0$.

Before proceding, we notice that `G-DCMST`(G, D, l, u, w) with $l = 0$ and $u(e) \geq |dep_D(e)|$, for each $e \in E(G)$, is an easily solvable problem. Since every spanning tree of G trivially (l, u)-satisfies D, it suffices to find one with minimum weight according to w. Besides, from $|dep_D(e)| \leq |E(G)|$, we see that there is no harm in considering $u(e) \leq |E(G)|$, for each $e \in E(G)$.

The following theorem establishes that the hardness of `G-DCMST`(G, D, l, u, w) under $l = 0$ is a direct consequence of the hardness of `CCMST`. In other words, it can be seen that CCMST holds its NP-hardness even if we "weaken" its conflict constraints.

Theorem 3.6 *G-DCMST(G, D, l, u, w) is NP-hard, even if $l = 0$ and u is a positive constant function.*

Proof. Let $\kappa > 0$ be an integer. Given an instance $(G = (V, E), G_c = (E, E_c), w)$ of CCMST we describe a cost preserving reduction to G-DCMST(G', D, l, u, w') in the following way: $G' = (V \cup V', E \cup E')$, where $V' = \{p\} \cup \{p_e^i : e \in E, i \in [\kappa]\}$ and $E' = \{\{p, q\}\} \cup \{a_e^i = \{p, p_e^i\} : e \in E, i \in [\kappa]\}$, for some $q \in V$; $D = (E \cup E', A)$, where $A = A_1 \cup A_2$, $A_1 = \{(e_1, e_2), (e_2, e_1) : \{e_1, e_2\} \in E_c\}$ and $A_2 = \{(a_e^i, e) : e \in E, i \in [\kappa]\}$; $l(e) = 0$, for each $e \in E \cup E'$; $u(e) = \kappa$, for each $e \in E \cup E'$; $w'_e = w_e$, for each $e \in E$, and $w'_e = 0$, for each $e \in E'$. Since we can restrict ourselves to the instances with $|E| \geq \kappa$ to prove NP-hardness, the reduction is polynomial. See Figure 4 for an illustration. Note that $G'[V' \cup \{q\}]$ is a tree.

Now, we show that T is a solution for CCMST(G, G_c, w) if, and only if, there is a solution T' for G-DCMST(G', D, l, u, w') with $T = T'[V]$.

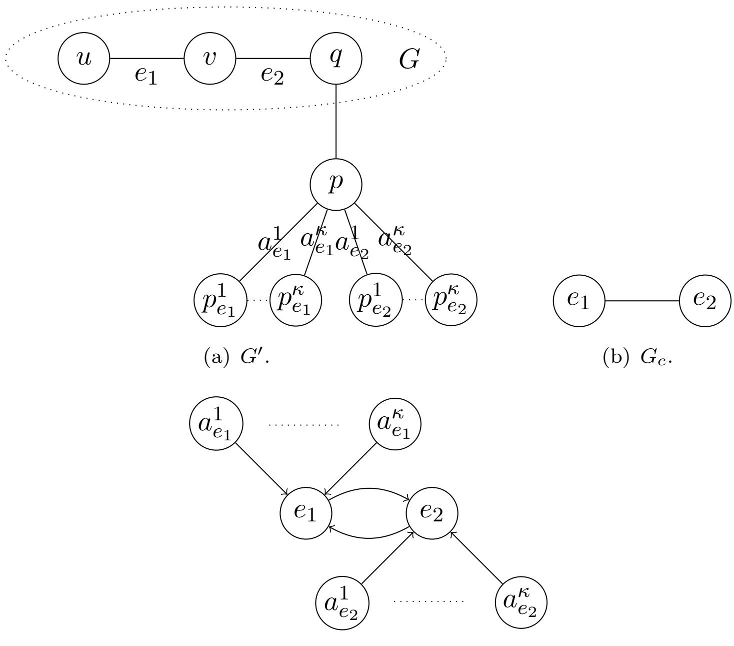
First, take $T = (V, E_T)$ as a solution for CCMST(G, G_c, w). We expand T into $T' = (V \cup V', E_{T'})$, defining $E_{T'} = E_T \cup E'$. Clearly, $T = T'[V]$. We need to show that T' is a solution for G-DCMST(G', D, l, u, w'). Since T is a spanning tree of G and every edge in E' is a cut edge in G', we see that T' is a spanning tree of G'. By the feasibility of T, for each $e_1, e_2 \in E_T$, we have that $e_1 \notin dep_D(e_2)$ and $e_2 \notin dep_D(e_1)$. Then, for each $e \in E_T$, $|dep_D(e) \cap E_{T'}| = \kappa$, as it is clear that $|dep_D(e) \cap E'| = \kappa$. From this, it follows that, for each $e \in E_T$, $l(e) \leq |dep_D(e) \cap E(T')| \leq u(e)$. Notice that, for each $e \in E'$, $dep_D(e) = \emptyset$, so it is immediate that, for any $\kappa \geq 0$, $l(e) \leq |dep_D(e) \cap E(T')| \leq u(e)$. Therefore, T' (l, u)-satisfies D, so T' is a solution for G-DCMST(G', D, l, u, w').

Conversely, let $T' = (V \cup V', E_{T'})$ be a solution for G-DCMST(G', D, l, u, w'). We show that $T = T'[V]$ is a solution for CCMST(G, G_c, w). Since each edge in E' is a cut edge in G', we see that $E' \subseteq E_{T'}$. Then, by the construction of D, it follows that $|dep_D(e) \cap E_{T'}| \geq |dep_D(e) \cap E'| = \kappa$, for each $e \in E$. Let $e \in E_{T'} \setminus E'$. By the feasibility of T', we have $|dep_D(e) \cap E_{T'}| \leq u(e) = \kappa$, and so $|dep_D(e) \cap E_{T'}| = u(e)$. From these observations, we conclude that $|dep_D(e) \cap (E_{T'} \setminus E')| = 0$. In other terms, if $e_1, e_2 \in E_{T'} \setminus E'$, we have that $e_1 \notin dep_D(e_2)$ and $e_2 \notin dep_D(e_1)$. Now, take $T = T[V'] = (V, E_{T'} \setminus E')$. It follows that T is (G_c)-conflicting free. At last, since T' is a spanning tree of G', T is a spanning tree of G. This concludes that T is a solution for CCMST(G, G_c, w).

To finish the proof, notice that E' edges have zero weight. This implies that corresponding solutions of CCMST(G, G_c, w) and G-DCMST(G', D, l, u, w') have the same weight, thus CCMST(G, G_c, w) and G-DCMST(G', D, l, u, w') have the same optimum value. □

In [16], CCMST(G, G_c, w) is proven to be solvable in polynomial time when G_c is a union of disjoint cliques. Due to its similarity with CCMST, we can prove an analogous result for G-DCMST(G, D, l, u, w) under $l = 0$.

Let (G, D, l, u, w) be an instance of G-DCMST, where $l = 0$ and $D = D_1 \cup D_2 \cup$



(a) G'.

(b) G_c.

(c) D.

Fig. 4. Illustration of CCMST reduction.

$\ldots \cup D_k$ is the union of k disjoint complete digraphs, that is, digraphs with arcs in both directions between any pair of vertices. Also, for each $e \in V(D_i)$, $i \in [k]$, assume that $u(e) = u_i - 1$, for some $1 \leq u_i \leq |V(D_i)|$. From this construction, a solution for G-DCMST(G, D, l, u, w) can have at most u_i edges from D_i, $i \in [k]$. We show that such an instance corresponds to a matroid optimization problem that can be solved in polynomial time.

The problem of finding a maximum weight basis of a weighted matroid can be solved in polynomial time [6]. Since a spanning tree of a graph G corresponds to a basis of the graphic matroid of G, the classical Minimum Spanning Tree Problem (MST) can be solved in polynomial time. If G is weighted by $w : E(G) \to \mathbb{R}^+$, we consider $w'_e = M - w_e$, for each $e \in E(G)$, with M choosen such that w' is a positive function. This way, MST(G, w) corresponds to finding a maximum weight basis of the graphic matroid of G weighted by w'.

A partition matroid $M = (E, \mathcal{I})$ is based on a partition $E = E_1 \cup E_2 \cup \ldots \cup E_k$ of its elements and integers $d_i \leq |E_i|$, $i \in [k]$. A subset S of E is in $\mathcal{I}$ iff $|S \cap E_i| \leq d_i$, for each $i \in [k]$. This definition is from [9].

In [7], the problem of finding a maximum weight common independent set of two weighted matroids M_1 and M_2 is shown to be solvable in polynomial time. Since a solution T for G-DCMST(G, D, l, u, w) corresponds to a basis of the graphic matroid of G, say M_1, and also to an independent set of a partition matroid related to D and u, say M_2, T corresponds to a common independent set of M_1 and M_2. As both M_1 and M_2 can be weighted according to w', this leads to the following theorem.

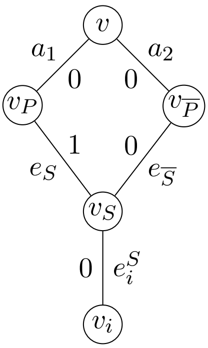
Theorem 3.7 *G-DCMST(G, D, l, u, w) can be solved in polynomial time, if $l = 0$,*

$D = D_1 \cup D_2 \cup \ldots \cup D_k$ is the union of k disjoint complete digraphs and, for each $e \in V(D_i)$, $i \in [k]$, $u(e) = u_i - 1$, for some $1 \le u_i \le |V(D_i)|$.

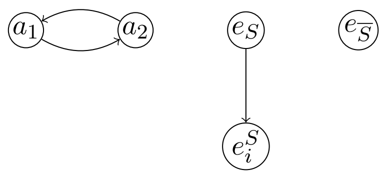
3.2.2 $l > 0$

When we consider instances (G, D, l, u, w) of `G-DCMST` where $l(e) > 0$, for each $e \in E(G)$, we allow an edge $e \in E(G)$ to take part in a solution only if a certain (positive) number of edges in its D-dependency set $dep_D(e)$ take part as well. In this case, it is easy to establish the NP-hardness of `G-DCMST`. Additionally, we further constrain l to obtain stronger hardness results.

Let us consider the reduction from the Set Cover Problem (`SCP`) used in [15] to prove that `L-DCMST` is APX-hard. Naturally, this reduction also proves that `L-DCST` is NP-complete. Such reduction builds an instance (G, D) as illustrated in Figure 5 (besides artificial vertices v, v_P and $v_{\overline{P}}$, each element i of the `SCP` instance is represented by v_i and each subset S is represented by v_S; also, $i \in S$ iff $e_i^S \in E(G)$). Observe that every edge $e \in E(G)$ has $|dep_D(e)| \in \{0, 1\}$. We can extend D into D' with arcs (e, e), for each $e \in E(G)$ with $dep_D(e) = \emptyset$, so every edge has exaclty one D'-dependency. Notice that `L-DCST`(G, D') is equivalent to `L-DCST`(G, D). Also, since every edge has exactly one D'-dependency, T is a solution of `L-DCST`(G, D') if, and only if, T is a solution of `G-DCST`(G, D', l, u), where $l(e) = u(e) = 1$, for each $e \in E(G)$. From this, we conclude the following theorem.



(a) Graph G.



(b) Digraph D.

Fig. 5. Illustration of the `L-DCMST` reduction, presented in [15].

Theorem 3.8 *`G-DCST`(G, D, l, u) is NP-complete, even if $l(e) = u(e) = 1$, for each $e \in E(G)$.*

We remark that `G-DCST`(G, D, l, u) has no solution when $l(e) > u(e)$, for some $e \in E(G)$. Thus, in order to complement Theorem 3.8, we consider the case where $0 < l < u$.

Consider the reduction proposed in the proof of Theorem 2.2. We extend this reduction as follows. G'' is created from G', adding vertices a_e^1 and a_e^2, for each $e \in E(G')$, and edges $f_e^1 = \{v, a_e^1\}$ and $f_e^2 = \{v, a_e^2\}$, for some $v \in V(G')$. Clearly, the a vertices are leaves in G'', so the f edges are cut edges. D' is created from D, adding arcs (f_e^i, e) and (f_e^i, f_e^i), $i \in [2]$, for each $e \in E(G')$. This way, each edge of G' has now two new artificial D'-dependencies in G'', which take part in any spanning tree of G''. We also define $l'(e) = 1$ and $u'(e) = 2$, for each $e \in E(G'')$. At last, $w'_e = w_e$, if $e \in E(G')$, and $w'_e = 0$, otherwise. It is easy to see that G-DCMST(G', D, l, u, w) is equivalent to G-DCMST(G'', D', l', u', w'), since the D-dependencies, forbidden by l and u, are also D'-dependencies, this time forbidden by the f edges, l' and u'. Notice that l' and u' are positive constant functions. This and the NP-hardness of CCMST imply the following corollary.

Corollary 3.9 *G-DCMST(G, D, l, u, w) is NP-hard, even if l and u are positive constant functions with $l < u$.*

4 Conclusion

We have introduced G-DCMST and established relations with NP-hard spanning tree problems. The table below presents particular cases of G-DCMST, for specific functions l and u. Each cell in the table leads to an infeasible problem (INF) or includes the indicated problem as a special case. The rows are related to $l(e) = 0$, $l(e) \in \{0, 1\}$, $1 \leq l(e) < dep_D(e)$, or $l(e) = dep_D(e)$, for every $e \in E$. The columns are related to similar cases for $u(e)$, for each $e \in E$. DCMST stands for problem L-DCMST (or equivalently A-DCMST) when the maximum in-degree of D is at most 1 [15]. Observe that every cell not marked with INF defines an NP-hard scenario, except for the Minimum Spanning Tree Problem (MST). Actually, it was shown that G-DCMST keeps its hardness even when very strict assumptions are taken either for G and D or for the functions l and u.

$l(e)$ \ $u(e)$	0	$\{0, 1\}$	$< dep_D(e)$	$dep_D(e)$
0	CCMST	NP-hard	MaxDeg-MST	MST
$\{0, 1\}$	INF	DCMST	NP-hard	L-DCMST
$< dep_D(e)$	INF	INF	GD-MST	NP-hard
$dep_D(e)$	INF	INF	INF	A-DCMST

We believe that G-DCMST(G, D, l, u, w) under $l = 0$ is a promising particular case to explore. It generalizes CCMST, which is a recently studied problem, and we have the impression that some results of CCMST can be generalized to this particular case of G-DCMST. We intend to tackle G-DCMST via integer linear programming and develop a polyhedral study of the dependency constraints. We also plan to use defective coloring results to obtain heuristic methods and approximation results for G-DCMST(G, D, l, u, w) under $l = 0$.

References

[1] Akgün, I. and B. Tansel, *Min-degree constrained minimum spanning tree problem: New formulation via millertuckerzemlin constraints*, Computers & Operations Research **37** (2010), pp. 72 – 82.

[2] Almeida, A. M., P. Martins and M. C. de Souza, *Min-degree constrained minimum spanning tree problem: complexity, properties, and formulations*, International Transactions in Operational Research **19** (2012), pp. 323–352.

[3] Bicalho, L. H., A. S. da Cunha and A. Lucena, *Branch-and-cut-and-price algorithms for the degree constrained minimum spanning tree problem*, Computational Optimization and Applications **63** (2016), pp. 755–792.

[4] Darmann, A., U. Pferschy, J. Schauer and G. J. Woeginger, *Paths, trees and matchings under disjunctive constraints*, Discrete Applied Mathematics **159** (2011), pp. 1726–1735.

[5] Dias, F. C., M. Campêlo, C. Souza and R. Andrade, *Min-degree constrained minimum spanning tree problem with fixed centrals and terminals: Complexity, properties and formulations*, Computers & Operations Research **84** (2017), pp. 46–61.

[6] Edmonds, J., *Matroids and the greedy algorithm*, Mathematical programming **1** (1971), pp. 127–136.

[7] Edmonds, J., *Matroid intersection*, Annals of discrete Mathematics **4** (1979), pp. 39–49.

[8] Krishnamoorthy, M., A. T. Ernst and Y. M. Sharaiha, *Comparison of algorithms for the degree constrained minimum spanning tree*, Journal of Heuristics **7** (2001), pp. 587–611.

[9] Lawler, E. L., *Combinatorial optimization: Networks and matroids*, New York (1976).

[10] Martinez, L. C. and A. S. da Cunha, *The min-degree constrained minimum spanning tree problem: Formulations and branch-and-cut algorithm*, Discrete Applied Mathematics **164** (2014), pp. 210 – 224.

[11] Narula, S. and C. Ho, *Degree-constrained minimum spanning tree*, Computers & Operations Research **7** (1980), pp. 239–249.

[12] Samer, P. and S. Urrutia, *A branch and cut algorithm for minimum spanning trees under conflict constraints*, Optimization Letters **9** (2015), pp. 41–55.

[13] Singh, M. and L. C. Lau, *Approximating minimum bounded degree spanning trees to within one of optimal*, J. ACM **62** (2015), pp. 1:1–1:19.

[14] Viana, L. A., "Árvore geradora com dependências mínima," Master's thesis, Federal University of Ceará (2016).
URL http://mdcc.ufc.br/teses/doc_download/307-234-luiz-alberto-do-carmo-viana

[15] Viana, L. A. C. and M. Campêlo, *Two dependency constrained spanning tree problems*, International Transactions in Operational Research (2019), to appear.

[16] Zhang, R., S. N. Kabadi and A. P. Punnen, *The minimum spanning tree problem with conflict constraints and its variations*, Discrete Optimization **8** (2011), pp. 191 – 205.