Available online at [www.sciencedirect.com](http://www.sciencedirect.com/)

[Electronic Notes in Theoretical Computer Science 346 (2019) 711–723](https://doi.org/10.1016/j.entcs.2019.08.062)

[www.elsevier.com/locate/entcs](http://www.elsevier.com/locate/entcs)

The Generalized Dependency Constrained Spanning Tree Problem

Luiz Alberto do Carmo Viana [1](#_bookmark0)*,*[3](#_bookmark0)

*Campus de Crateu´s Universidade Federal do Cear´a Crateu´s, Brazil*

Manoel Campˆelo [1](#_bookmark0)*,*[2](#_bookmark0)*,*[4](#_bookmark0)

*Departamento de Estat´ıstica e Matem´atica Aplicada Universidade Federal do Cear´a*

*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 *∈ E* is a *D*-dependency of *e*2 if (*e*1*, e*2) *∈ A*. For each *e ∈ E*, we define its *D*-dependency set as *depD*(*e*)= *{ej ∈ E* : (*ej, e*) *∈ A}*. Also, let *l, u* : *E →* 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](mailto:luizalberto@crateus.ufc.br)

4 Email: [mcampelo@lia.ufc.br](mailto:mcampelo@lia.ufc.br)

<https://doi.org/10.1016/j.entcs.2019.08.062>

1571-0661/© 2019 The Author(s). Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

0 *∈* N). We say that a subgraph *H ⊆ G* of *G* (*l, u*)-satisfies *D* if, for each *e ∈ E*(*H*), *l*(*e*) *≤ |depD*(*e*) *∩ E*(*H*)*|≤ u*(*e*).

Let *fG* denote the set of all spanning trees of *G*. We define the Generalized De- pendency Constrained Spanning Tree Problem, abbreviated as G-DCST(*G, D, l, u*), as the problem of deciding if there is a *T ∈ fG* such that *T* (*l, u*)-satisfies *D*.

Let *w* : *E →* R+ be a weighting function over the edge set of *G*. For each sub- graph *H ⊆ G*, we define *w*(*H*) = Σ*e∈E*(*H*) *we*. 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 ∈ fG* that (*l, u*)-

satisfy *D*, a *T∗* with *w*(*T∗*) minimum.

The next sections develop results concerning G-DCST and G-DCMST. Section [2](#_bookmark1) es- tablishes relations between the present problems and generalized versions of span- ning tree problems found in the literature. Section [3](#_bookmark8) 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](#_bookmark15) ends the paper with comments and directions of future work.

# 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 partic- ular 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.

* 1. *Previous dependency constrained problems*

In a previous work [[15](#_bookmark30)], we have introduced and tackled problems similar to G-DCST. We have defined the Least-Dependency Constrained Spanning Tree Problem, abbre- viated as L-DCST(*G, D*), which consists of deciding whether there is a *T ∈ fG* such that, for each *e ∈ E*(*T* ) with *depD*(*e*) */*= *∅*, *at least one D*-dependency of *E* is also in *T* (i.e. *depD*(*e*) *∩ E*(*T* ) */*= *∅* if *depD*(*e*) */*= *∅*). 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 ∈ fG* such that, for each *e ∈ E*(*T* ), *all D*-dependencies of *e* are also in *T* (i.e. *depD*(*e*) *∩ E*(*T* )= *depD*(*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*{|depD*(*e*)*|,* 1*}* (if *depD*(*e*) = *∅*, *e* can freely take part in a solution) and *u*(*e*) = *|depD*(*e*)*|*, for all *e ∈ 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*) =

*|depD*(*e*)*|*, for all *e ∈ E*(*G*). *2*

As G-DCST generalizes both problems, it inherits their applications. For instance, dependency relations can model communication systems with protocol conversion

restrictions [[14](#_bookmark29)]. Besides, in Section [3](#_bookmark8), we profit from Theorem [2.1](#_bookmark2) and the compu- tational complexity study presented in [[15](#_bookmark30)] 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*) *→* 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,](#_bookmark2) they are particular cases of G-DCMST(*G, D, l, u, w*). In [[15](#_bookmark30)], 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.

* 1. *Conflict Constrained Minimum Spanning Tree Problem*

Let *G* = (*V, E*) be an undirected graph. Also, let *Gc* = (*E, Ec*) be a graph whose vertices are the edges of *G*. We say that *e*1*, e*2 *∈ E* are (*Gc*)-conflicting if *{e*1*, e*2*}∈ Ec*. We call *H ⊆ G* a (*Gc*)-conflicting free subgraph if no *e*1*, e*2 *∈ E*(*H*) are (*Gc*)-conflicting.

Given *w* : *E →* R+, the Conflict Constrained Minimum Spanning Tree Problem, abbreviated here as CCMST(*G, Gc, w*), consists of finding a (*Gc*)-conflicting free *T∗ ∈* *fG* with *w*(*T∗*) minimum. This problem was introduced in [[4](#_bookmark17)]. It was shown to be strongly NP-hard, even if *Gc* is the disjoint union of paths of lenght 2 [[4](#_bookmark17)], and it has a polynomial case when *Gc* is the disjoint union of cliques [[16](#_bookmark31)]. Besides, the problem is NP-hard even if *G* is a cactus [[16](#_bookmark31)]. Experiments with heuristics and a branch-and-cut algorithm were shown in [[16](#_bookmark31)] and [[12](#_bookmark27)], respectively. G-DCMST also generalizes this problem.

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

**Proof.** Given an instance (*G, Gc, w*) of CCMST, we build an instance (*Gj, D, l, u, w*) of G-DCMST in the following way: first, we make *Gj* = *G* and *D* = (*E*(*Gj*)*, A*), where *A* = *{*(*e*1*, e*2)*,* (*e*2*, e*1): *{e*1*, e*2*} ∈ E*(*Gc*)*}*; for each *e ∈ E*(*Gj*), *l*(*e*)= *u*(*e*) = 0. The construction of *D* is illustrated in Figure [1.](#_bookmark4)

Now, we prove that *T* is a solution for CCMST(*G, Gc, w*) if, and only if, *T* is a solution for G-DCMST(*Gj, D, l, u, w*).

Suppose that *T* is a solution for CCMST(*G, Gc, w*). This means that *T* is a span- ning tree of *G* and no *e*1*, e*2 *∈ E*(*T* ) are (*Gc*)-conflicting. Considering *D*, this implies that, for each *e*1*, e*2 *∈ E*(*T* ), *e*1 *∈/ depD*(*e*2) and *e*2 *∈/ depD*(*e*1). Then, for each *e ∈ E*(*T* ), we have *depD*(*e*) *∩E*(*T* )= *∅*, leading to *l*(*e*) *≤ |depD*(*e*) *∩E*(*T* )*|≤ u*(*e*). From this, we see that *T* (*l, u*)-satisfies *D*, so *T* is a solution for G-DCMST(*Gj, D, l, u, w*).

Conversely, take *T* as a solution for G-DCMST(*Gj, D, l, u, w*). This means that *T* is a spanning tree of *Gj* and *T* (*l, u*)-satisfies *D*. By definition, for each *e ∈ E*(*T* ), we have *depD*(*e*)*∩E*(*T* )= *∅*. Considering the construction of *D*, for each *e*1*, e*2 *∈ E*(*T* ), *e*1 and *e*2 are not (*Gc*)-conflicting, thus *T* is (*Gc*)-conflicting free. From this, *T* is a solution for CCMST(*G, Gc, w*).

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

value. *2*

(a) *Gc*. (b) *D*.



*e*1

*e*2



*e*1

*e*2

Fig. 1. Illustration of the CCMST reduction.

* 1. *Degree Constrained Spanning Tree Problems*

Let *G* = (*V, E*) be a graph. We denote by *dG*(*v*)*,v ∈ 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 ∈ V* a leaf in *G* if *dG*(*v*) = 1. For *V j ⊆ V* , let *G*[*V j*] be the subgraph of *G* induced by *V j*. We present relations between G-DCMST and spanning tree problems character- ized 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.

* + 1. *Max-Degree Constrained Minimum Spanning Tree*

Let *G* = (*V, E*) be a graph. Given *k* : *V →* N and *w* : *E →* 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 ∈ fG* such that *dT* (*v*) *≤ k*(*v*), for each *v ∈ V* , a *T∗* with minimum *w*(*T∗*). This NP-hard problem was first introduced in [[11](#_bookmark26)]. Since then, it has been extensively studied. Several heuristic, approximation and exact approaches have been proposed for the problem (see for example [[8,13,](#_bookmark28)[3](#_bookmark18)] 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 (*Gj, D, l, u, wj*) of G-DCMST as follows: *Gj* = (*V ∪ V j,E ∪ Ej*), where *V j* =

*{vj* : *v ∈ V }* is a set of artificial vertices, one for each vertex in *V* , and *Ej* = *{ev* = *{v, vj}* : *v ∈ V }* is a set of artificial cut edges, each one linking an original vertex *v ∈ V* and its corresponding artificial vertex *vj ∈ V j*; *D* = (*E ∪ Ej, A*), where *A* = *{*(*{u, v}, eu*)*,* (*{u, v}, ev*) : *{u, v} ∈ E}*; *l*(*e*) = *u*(*e*) = 0, for each *e ∈ E*, while

*l*(*ev*) = 0 and *u*(*ev*) = *k*(*v*), for each *v ∈ V* ; at last, *wj*

*e*

= *we*, for each *e ∈ E*, and

*wj* = 0, for each *e ∈ Ej*. This construction is illustrated in Figure [2](#_bookmark5). In particular, note that *depD*(*ev*) is the set of edges incident to *v* in *G*, for all *v ∈ V* .

*e*

We show that *T* is a solution for MaxDeg-MST(*G, k, w*) if, and only if, there is a solution *T j* for G-DCMST(*Gj, D, l, u, wj*) with *T* = *T j*[*V* ].

First, let *T* = (*V, ET* ) be a solution for MaxDeg-MST(*G, k, w*). We expand *T* into *T j* = (*V ∪ V j, ET ′* ) *⊆ Gj*, where *ET ′* = *ET ∪ Ej*. We need to show that *T j* is a solution for G-DCMST(*Gj, D, l, u, wj*). Since *T* is a spanning tree of *G*, it is clear that *T j* is a spanning tree of *Gj*. Each edge *e ∈ ET* has *depD*(*e*)= *∅*, so *l*(*e*)= *u*(*e*)=0 implies that *l*(*e*) *≤ |depD*(*e*) *∩ E*(*Tj*)*| ≤ u*(*e*). Now, let *e ∈ ET ′ \ ET* = *Ej*. Then, *e* = *ev* for some *v ∈ V* and, by the feasibility of *T* , *dT* (*v*) *≤ k*(*v*). This implies

that at most *k*(*v*) edges of *depD*(*ev*) are in *ET ′* . From this, it follows that *l*(*ev*) *≤*

*|depD*(*ev*) *∩ E*(*Tj*)*| ≤ u*(*ev*). Therefore, *T j* (*l, u*)-satisfies *D* and we have that *T j* is a solution for G-DCMST(*Gj, D, l, u, wj*).

Conversely, take *T j* = (*V ∪ V j, ET ′* ) as a solution for G-DCMST(*Gj, D, l, u, wj*). We show that *T* = *T j*[*V* ] is a solution for MaxDeg-MST(*G, k, w*). Let *v ∈ V* . As *ev* is a cut edge in *Gj*, then *ev ∈ ET ′* . By the construction of *D* and function *u*, this implies that there are at most *k*(*v*) edges *{u, v}∈ E* in *ET ′* . Since *T* is the subtree of *T j* induced by *V* , it follows that *dT* (*v*) *≤ k*(*v*). Moreover, since *T j* is a spanning tree of *Gj*, *T* is a spanning tree of *G*. Thus *T* is a solution for MaxDeg-MST(*G, k, w*). To finish the proof, notice that, since the edges in *Ej* have zero weight, cor- responding solutions of MaxDeg-MST(*G, k, w*) and G-DCMST(*Gj, D, l, u, wj*) have the same weight. This concludes that MaxDeg-MST(*G, k, w*), and G-DCMST(*Gj, D, l, u, wj*) have the same optimum value. *2*

(a) *G′*.



*uj*

*eu*

*vj*

*ev*

*u*

*v*

*{u, v}*

*eu*

*ev*

(b) *D*.

Fig. 2. Illustration of MaxDeg-MST reduction.

* + 1. *Generalized Degree Constrained Minimum Spanning Tree*

Let *G* = (*V, E*) be a graph. Given *κ ∈* N and *w* : *E →* R+, the Min-Degree Con- strained Minimum Spanning Tree Problem, abbreviated here as MD-MST(*G, κ, w*), consists of finding, among the *T ∈ fG* such that each nonleaf *v* of *T* has *dT* (*v*) *≥ κ*, a *T∗* with minimum *w*(*T∗*). This problem was introduced in [[2](#_bookmark19)], where it was shown to be NP-hard. Integer linear programs and solution methods were proposed in [[1,](#_bookmark16)[2,](#_bookmark19)[10](#_bookmark24)].

Let us introduce a generalized version of this problem, to be denoted GD-MST(*G, kj, k, w*), where we replace the scalar *κ* by functions *k, kj* : *V →* N and require each nonleaf *v* of *T* to satisfy *kj*(*v*) *≤ dT* (*v*) *≤ k*(*v*).

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

**Theorem 2.4** *GD-MST*(*G, kj, k, w*) *is a particular case of G-DCMST*(*G, D, l, u, w*)*.*

**Proof.** Given an instance (*G* = (*V, E*)*, kj, k, w*) of GD-MST, we build an instance (*Gj, D, l, u, wj*) of G-DCMST in the following way: first, *Gj* = (*V ∪ V j,E ∪ Ej*),

where *V j* = *{v*1*, v*2*, v*3 : *v ∈ V }* and *Ej* = *{ev*

1

= *{v, v*1*}, ev*

= *{v, v*2*}, ev* =

*{v*1*, v*3*}, ev* = *{v*2*, v*3*}* : *v ∈ V }*; we make *D* = (*E ∪Ej, A*), where *A* = *A*1 *∪A*2, *A*1 =

2

3

4

*{*(*{u, v}, ev*)*,* (*{u, v}, ev*) : *v ∈ V, u ∈ NG*(*v*)*}* and *A*2 = *{*(*ev, ev*)*,* (*ev, ev*): *v ∈ V }*;

1 2 3 4 4 3

for each *e ∈ E*, *l*(*e*) = *u*(*e*) = 0; for each *v ∈ V* , *l*(*ev*) = *kj*(*v*), *u*(*ev*) = *k*(*v*),

1 1

*l*(*ev*) = *u*(*ev*) = 1, 2 *≤ i ≤* 4; at last, *wj* = *we*, if *e ∈ E*, and *wj*

= 0, otherwise.

*i i e e*

This construction is illustrated in Figure [3](#_bookmark7). Observe that *depD*(*ev*) = *depD*(*ev*) is

1 2

the set of edges incident to *v*, for every *v ∈ V* .

We show that *T* is a solution for GD-MST(*G, kj, k, w*) if, and only if, there is a solution *T j* for G-DCMST(*Gj, D, l, u, wj*) with *T* = *T j*[*V* ].

First, let *T* = (*V, ET* ) be a solution for GD-MST(*G, kj, k, w*). We expand *T* into *T j* = (*V ∪V j, ET ′* ) *⊆ Gj*, where *ET ′* is equal to *ET* together with the following edges: for each *v ∈ V* , *ev* and *ev*; for each *v ∈ V* , either *ev* or *ev*, depending whether *v* is

3 4 2 1

a leaf in *T* or not, respectively. Clearly, *T* = *T j*[*V* ]. It remains to show that *T j* is a

solution for G-DCMST(*Gj, D, l, u, wj*). Since *T* is a spanning tree of *G* and, for each *v ∈ V* , exactly three of *ev,i ∈* [4], are in *ET ′* , *T j* is a spanning tree of *Gj*. Now, we show that the *D*-dependencies are satisfied. Every edge *e ∈ ET* has *depD*(*e*) = *∅* and *l*(*e*) = *u*(*e*) = 0, so *l*(*e*) *≤ |depD*(*e*) *∩ E*(*Tj*)*| ≤ u*(*e*) trivially follows. The remaining edges in *ET ′ \ ET* can be grouped as follows:

*i*

1. *ev* and *ev*, for each *v ∈ V* : 1 = *l*(*ev*) *≤ |depD*(*ev*) *∩ E*(*Tj*)*| ≤ u*(*ev*) = 1,

3 4 *i i* *i*

1. *≤ i ≤* 4, is immediate because *ev* is the unique dependency of *ev*, and vice-

3 4

versa;

1. *ev*, for each leaf *v* in *T* : since exactly one edge of *depD*(*ev*) is in *ET ′* , we have

2 2

that 1 = *l*(*ev*) *≤ |depD*(*ev*) *∩ E*(*Tj*)*|≤ u*(*ev*)= 1;

2 2 2

1. *ev*, for each nonleaf *v* in *T* : from the feasibility of *T* , *kj*(*v*) *≤ dT* (*v*) *≤ k*(*v*),

1

i.e. at least *kj*(*v*) and at most *k*(*v*) edges of *depD*(*ev*) are in *ET ⊆ ET ′* , which implies that *kj*(*v*)= *l*(*ev*) *≤ |depD*(*ev*) *∩ E*(*Tj*)*|≤ u*(*ev*)= *k*(*v*).

1

1 1 1

Therefore, *T j* (*l, u*)-satisfies *D*, and we have that *T j* is a solution for

G-DCMST(*Gj, D, l, u, wj*).

Conversely, suppose that *T j* = (*V ∪ V j, ET ′* ) is a solution for G-DCMST(*Gj, D, l, u, wj*). We show that *T* = *T j*[*V* ] is a solution for GD-MST(*G, kj, k, w*). Due to dependency constraints, *ev* and *ev* are in *T j*, for each

3 4

*v ∈ V* . From this, and since *ev* and *ev* are a cut in *Gj*, exactly one of *ev* and *ev* is

1 2 1 2

in *T j*, for each *v ∈ V* . Take *v ∈ V* . If *ev* is in *T j*, then there are from *kj*(*v*) to *k*(*v*)

1

edges *{u, v}∈ E* in *ET ′* . If *ev* is in *T j*, then there is exaclty one edge *{u, v}∈ E* in *ET ′* . Therefore, either *kj*(*v*) *≤ dT* (*v*) *≤ k*(*v*) or *dT* (*v*) = 1. Since *T j* is a spanning tree of *Gj*, *T* is a spanning tree of *G*, and thus *T* is a solution of GD-MST(*G, kj, k, w*). To finish the proof, observe that corresponding solutions of GD-MST(*G, kj, k, w*) and G-DCMST(*Gj, D, l, u, wj*) have the same weight, since *Ej* edges have zero weight. This concludes that GD-MST(*G, kj, k, w*) and G-DCMST(*Gj, D, l, u, wj*) have the same optimum value. *2*

2

An interesting variation of MD-MST(*G* = (*V, E*)*, k, w*) is obtained when the leaves are fixed a priori. Precisely, given *C ⊆ V* and *d* : *C →* Z+, it consists in finding a *T∗* among the *T ∈ fG* such that *dT* (*u*) *≥ d*(*u*), for each *u ∈ C*, and *dT* (*v*) = 1, for each *v ∈ V \C*, with *w*(*T∗*) minimum. This problem is abbreviated as FMD-MST(*G, C, d, w*) and was introduced in [[5](#_bookmark20)]. Similarly, let us introduce a generalized version, to be denoted FGD-MST(*G, C, dj, d, w*), where we add a lower bounding function *dj* : *V →* N



*eu*

3

*u*3

*eu*

4

*ev*

3

*v*3

*ev*

4

*u*1

*u*2

*v*1

*v*2

*eu*

1

*u*

*eu*

2

*ev*

1

*v*

*ev*

2

(a) *G′*.



*eu*

1

*{u, v}*

*ev*

1

*eu*

2

*ev*

2



*eu*

3

*eu*

4

(b) *D*.



*ev*

3

*ev*

4

Fig. 3. Illustration of MD-MST reduction.

and require each vertex *v ∈ C* to satisfy *dj*(*u*) *≤ dT* (*u*) in any feasible tree *T* (besides the constraints of FMD-MST).

We can prove that FGD-MST(*G, C, dj, 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](#_bookmark6). Given *Gj* built as described in Theorem [2.4](#_bookmark6), we build *Gjj* removing edges from *Gj*: for each *v ∈ C*, we remove the edge *ev*; for each *v ∈ V \ C*, we remove the edge *ev*. An obvious adaptation on *D* is also required. An argument similar to the proof of Theorem [2.4](#_bookmark6) is sufficient to conclude the following theorem.

2

1

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

# Complexity

* 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](#_bookmark30)]. 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* Δ(*G*) *≤* 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](#_bookmark2), [3.1](#_bookmark9) and [3.2,](#_bookmark10) we can establish the NP-

completeness of G-DCST as follows. The last observation is due to the proof of Theorem [2.1](#_bookmark2).

**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*](#_bookmark9) *or* [*3.2*](#_bookmark10)*. 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](#_bookmark30)] we prove that L-DCMST and A-DCMST keep an ln(*n*) inapproximability threshold for very restricting assump- tions. 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 *−* Ω(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,](#_bookmark2) 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*](#_bookmark11) *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.*

* 1. *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.

* + 1. *l* =0

When we take instances (*G, D, l, u, w*) of G-DCMST where *l*(*e*) = 0, for each *e ∈ E*(*G*), we allow the inclusion of an edge *e ∈ 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*) *≥*

*|depD*(*e*)*|*, for each *e ∈ 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 *|depD*(*e*)*| ≤ |E*(*G*)*|*, we see that there is no harm in considering *u*(*e*) *≤ |E*(*G*)*|*, for each *e ∈ 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 *κ >* 0 be an integer. Given an instance (*G* = (*V, E*)*, Gc* = (*E, Ec*)*, w*) of CCMST we describe a cost preserving reduction to G-DCMST(*Gj, D, l, u, wj*) in the

following way: *Gj* = (*V ∪ V j,E ∪ Ej*), where *V j* = *{p}∪ {pi*

*e*

: *e ∈ E, i ∈* [*κ*]*}* and

*Ej* = *{{p, q}} ∪ {ai* = *{p, pi }* : *e ∈ E, i ∈* [*κ*]*}*, for some *q ∈ V* ; *D* = (*E ∪ Ej, A*),

*e*

*e*

where *A* = *A*1 *∪ A*2, *A*1 = *{*(*e*1*, e*2)*,* (*e*2*, e*1): *{e*1*, e*2*} ∈ Ec}* and *A*2 = *{*(*ai , e*): *e ∈*

*e*

*E, i ∈* [*κ*]*}*; *l*(*e*) = 0, for each *e ∈ E ∪ Ej*; *u*(*e*)= *κ*, for each *e ∈ E ∪ Ej*; *wj* = *we*,

*e*

for each *e ∈ E*, and *wj* = 0, for each *e ∈ Ej*. Since we can restrict ourselves to

*e*

the instances with *|E|≥ κ* to prove NP-hardness, the reduction is polynomial. See Figure [4](#_bookmark12) for an illustration. Note that *Gj*[*V j ∪ {q}*] is a tree.

Now, we show that *T* is a solution for CCMST(*G, Gc, w*) if, and only if, there is a solution *T j* for G-DCMST(*Gj, D, l, u, wj*) with *T* = *T j*[*V* ].

First, take *T* = (*V, ET* ) as a solution for CCMST(*G, Gc, w*). We expand *T* into *T j* = (*V ∪ V j, ET ′* ), defining *ET ′* = *ET ∪ Ej*. Clearly, *T* = *T j*[*V* ]. We need to show that *T j* is a solution for G-DCMST(*Gj, D, l, u, wj*). Since *T* is a spanning tree of *G* and every edge in *Ej* is a cut edge in *Gj*, we see that *T j* is a spanning tree of *Gj*. By the feasibility of *T* , for each *e*1*, e*2 *∈ ET* , we have that *e*1 *∈/ depD*(*e*2) and *e*2 *∈/ depD*(*e*1). Then, for each *e ∈ ET* , *|depD*(*e*) *∩ ET ′ |* = *κ*, as it is clear that *|depD*(*e*) *∩ Ej|* = *κ*. From this, it follows that, for each *e ∈ ET* , *l*(*e*) *≤ |depD*(*e*) *∩ E*(*Tj*)*| ≤ u*(*e*). Notice that, for each *e ∈ Ej*, *depD*(*e*) = *∅*, so it is immediate that, for any *κ ≥* 0, *l*(*e*) *≤ |depD*(*e*) *∩ E*(*Tj*)*|≤ u*(*e*). Therefore, *T j* (*l, u*)-satisfies *D*, so *T j* is a solution for G-DCMST(*Gj, D, l, u, wj*).

Conversely, let *T j* = (*V ∪ V j, ET ′* ) be a solution for G-DCMST(*Gj, D, l, u, wj*). We show that *T* = *T j*[*V* ] is a solution for CCMST(*G, Gc, w*). Since each edge in *Ej* is a cut edge in *Gj*, we see that *Ej ⊆ ET ′* . Then, by the construction of *D*, it follows that *|depD*(*e*) *∩ ET ′ |≥ |depD*(*e*) *∩ Ej|* = *κ*, for each *e ∈ E*. Let *e ∈ ET ′ \ Ej*. By the feasibility of *T j*, we have *|depD*(*e*) *∩ ET ′ |≤ u*(*e*)= *κ*, and so *|depD*(*e*) *∩ ET ′ |* = *u*(*e*). From these observations, we conclude that *|depD*(*e*) *∩* (*ET ′ \ Ej*)*|* = 0. In other terms, if *e*1*, e*2 *∈ ET ′ \ Ej*, we have that *e*1 *∈/ depD*(*e*2) and *e*2 *∈/ depD*(*e*1). Now, take *T* = *T* [*V j*] = (*V, ET ′ \ Ej*). It follows that *T* is (*Gc*)-conflicting free. At last, since *T j* is a spanning tree of *Gj*, *T* is a spanning tree of *G*. This concludes that *T* is a solution for CCMST(*G, Gc, w*).

To finish the proof, notice that *Ej* edges have zero weight. This implies that cor- responding solutions of CCMST(*G, Gc, w*) and G-DCMST(*Gj, D, l, u, wj*) have the same weight, thus CCMST(*G, Gc, w*) and G-DCMST(*Gj, D, l, u, wj*) have the same optimum value. *2*

In [[16](#_bookmark31)], CCMST(*G, Gc, w*) is proven to be solvable in polynomial time when *Gc* 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 *∪ D*2 *∪*



*u*

*e*1

*v*

*e*2

*q*

*G*

*p*

*a*1 *aκ a*1 *aκ*

*e*1 *e*1 *e*2

*e*2

*p*1 *pκ*

*e*1

*e*1

*p*1 *pκ*

*e*2

*e*2



*e*1

*e*2

* + - 1. *G′*.
      2. *Gc*.



*a*1

*e*1

*aκ*

*e*1

*e*1

*e*2

*a*1

*e*2

*aκ*

*e*2

* + - 1. *D*.

Fig. 4. Illustration of CCMST reduction.

*... ∪ Dk* 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 ∈ V* (*Di*), *i ∈* [*k*], assume that *u*(*e*) = *ui −* 1, for some 1 *≤ ui ≤ |V* (*Di*)*|*. From this construction, a solution for G-DCMST(*G, D, l, u, w*) can have at most *ui* edges from *Di*, *i ∈* [*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](#_bookmark21)]. 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*) *→* R+, we consider *wj* = *M −we*, for each *e ∈ E*(*G*), with *M* choosen such that *wj* is a positive function. This way, MST(*G, w*) corresponds to finding a maximum weight basis of the graphic matroid of *G* weighted by *wj*.

*e*

A partition matroid *M* = (*E, I*) is based on a partition *E* = *E*1 *∪E*2 *∪... ∪Ek* of its elements and integers *di ≤ |Ei|*, *i ∈* [*k*]. A subset *S* of *E* is in *I* iff *|S ∩ Ei|≤ di*, for each *i ∈* [*k*]. This definition is from [[9](#_bookmark25)].

In [[7](#_bookmark22)], 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 *wj*, 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 *∪ D*2 *∪ ... ∪ Dk is the union of k disjoint complete digraphs and, for each*

*e ∈ V* (*Di*)*, i ∈* [*k*]*, u*(*e*)= *ui −* 1*, for some* 1 *≤ ui ≤ |V* (*Di*)*|.*

* + 1. *l >* 0

When we consider instances (*G, D, l, u, w*) of G-DCMST where *l*(*e*) *>* 0, for each *e ∈ E*(*G*), we allow an edge *e ∈ E*(*G*) to take part in a solution only if a certain (positive) number of edges in its *D*-dependency set *depD*(*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](#_bookmark30)] 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](#_bookmark13) (besides artificial vertices *v*, *vP* and *vP* , each element *i* of the SCP instance is represented by *vi* and each subset *S* is represented by *vS*; also, *i ∈ S* iff *eS ∈ E*(*G*)). Observe that every edge *e ∈ E*(*G*) has *|depD*(*e*)*|∈ {*0*,* 1*}*. We can extend *D* into *Dj*

*i*

with arcs (*e, e*), for each *e ∈ E*(*G*) with *depD*(*e*)= *∅*, so every edge has exaclty one *Dj*-dependency. Notice that L-DCST(*G, Dj*) is equivalent to L-DCST(*G, D*). Also, since every edge has exactly one *Dj*-dependency, *T* is a solution of L-DCST(*G, Dj*) if, and only if, *T* is a solution of G-DCST(*G, Dj, l, u*), where *l*(*e*)= *u*(*e*) = 1, for each *e ∈ E*(*G*). From this, we conclude the following theorem.



*a*1

*vP*

*v*

0 0

*a*2

*vP*

*eS*

1 0

*eS*

*vS*

0

*eS*

*i*

*vi*



*a*1

*a*2



*eS*

*eS*

*i*

*eS*

(a) Graph *G*. (b) Digraph *D*.

Fig. 5. Illustration of the L-DCMST reduction, presented in [[15](#_bookmark30)].

**Theorem 3.8** *G-DCST*(*G, D, l, u*) *is NP-complete, even if l*(*e*)= *u*(*e*)= 1*, for each*

*e ∈ E*(*G*)*.*

We remark that G-DCST(*G, D, l, u*) has no solution when *l*(*e*) *> u*(*e*), for some *e ∈ E*(*G*). Thus, in order to complement Theorem [3.8,](#_bookmark14) we consider the case where 0 *< l < u*.

Consider the reduction proposed in the proof of Theorem [2.2](#_bookmark3). We extend this reduction as follows. *Gjj* is created from *Gj*, adding vertices *a*1 and *a*2, for each *e ∈*

*e e*

*E*(*Gj*), and edges *f* 1 = *{v, a*1*}* and *f* 2 = *{v, a*2*}*, for some *v ∈ V* (*Gj*). Clearly, the *a*

*e e e e*

vertices are leaves in *Gjj*, so the *f* edges are cut edges. *Dj* is created from *D*, adding

arcs (*fi, e*) and (*fi,fi*)*,i ∈* [2], for each *e ∈ E*(*Gj*). This way, each edge of *Gj* has

*e e e*

now two new artificial *Dj*-dependencies in *Gjj*, which take part in any spanning tree of *Gjj*. We also define *lj*(*e*)=1 and *uj*(*e*) = 2, for each *e ∈ E*(*Gjj*). At last, *wj* = *we*,

*e*

if *e ∈ E*(*Gj*), and *wj* = 0, otherwise. It is easy to see that G-DCMST(*Gj, D, l, u, w*)

*e*

is equivalent to G-DCMST(*Gjj, Dj, lj, uj, wj*), since the *D*-dependencies, forbidden by *l* and *u*, are also *Dj*-dependencies, this time forbidden by the *f* edges, *lj* and *uj*. Notice that *lj* and *uj* 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 con- stant functions with l < u.*

# 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 func- tions *l* and *u*. Each cell in the table leads to an infeasible problem (INF) or in- cludes the indicated problem as a special case. The rows are related to *l*(*e*) = 0, *l*(*e*) *∈ {*0*,* 1*}*, 1 *≤ l*(*e*) *< depD*(*e*), or *l*(*e*)= *depD*(*e*), for every *e ∈ E*. The columns are related to similar cases for *u*(*e*), for each *e ∈ E*. DCMST stands for problem L-DCMST (or equivalently A-DCMST) when the maximum in-degree of *D* is at most 1 [[15](#_bookmark30)]. 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*.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *u*(*e*)  *l*(*e*) | 0 | *{*0*,* 1*}* | *< depD*(*e*) | *depD*(*e*) |
| 0 | CCMST | NP-hard | MaxDeg-MST | MST |
| *{*0*,* 1*}* | INF | DCMST | NP-hard | L-DCMST |
| *< depD*(*e*)  *depD*(*e*) | INF  INF | INF  INF | GD-MST  INF | NP-hard  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. Akgu¨n, 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.

1. 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.
2. Dias, F. C., M. Campˆelo, 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.
3. Edmonds, J., *Matroids and the greedy algorithm*, Mathematical programming **1** (1971), pp. 127–136.
4. Edmonds, J., *Matroid intersection*, Annals of discrete Mathematics **4** (1979), pp. 39–49.
5. 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.
6. Lawler, E. L., *Combinatorial optimization: Networks and matroids*, New York (1976).
7. 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.
8. Narula, S. and C. Ho, *Degree-constrained minimum spanning tree*, Computers & Operations Research

**7** (1980), pp. 239–249.

1. Samer, P. and S. Urrutia, *A branch and cut algorithm for minimum spanning trees under conflict constraints*, Optimization Letters **9** (2015), pp. 41–55.
2. 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.
3. Viana, L. A., “A´rvore geradora com dependˆencias m´ınima,” Master’s thesis, Federal University of Cear´a (2016).

URL <http://mdcc.ufc.br/teses/doc_download/307-234-luiz-alberto-do-carmo-viana>

1. Viana, L. A. C. and M. Campˆelo, *Two dependency constrained spanning tree problems*, International Transactions in Operational Research (2019), to appear.
2. 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.