The Least-Dependency Constrained Spanning Tree Problem

Luiz Alberto Viana[1], **Manoel Campêlo**[1]

[1]Mestrado e Doutorado em Ciência da Computação – Universidade Federal do Ceará

Abstract. *We introduce the Least-Dependency Constrained Spanning Tree Problem, which consists of finding a spanning tree where each edge has at least one edge of its dependency set, if non-empty, also in the tree. The dependencies on the input graph G are described by a digraph D where the vertices are the edges of G, and the in-neighbors of a vertex are its dependency set. We show that the optimization problem is NP-hard even if G is a chordal cactus with maximum degree 3 or diameter at most 2, and D is a disjoint union of arborescences of height 2. The same complexity is proved when G is planar bipartite, and each component of D is an oriented cycle or an anti-arborescence of height 1. We also report two polynomial cases.*

1. Introduction

Finding a minimum spanning tree is known to be easy, but extra requirements for the tree can turn the problem NP-hard. One of the NP-hard variations occurs when conflict constraints over pairs of edges are imposed [Darmann et al. 2011, Zhang et al. 2011, Samer and Urrutia 2015]. These constraints are naturally described by an undirected simple graph, where each edge is a pair of conflicting edges of the input graph. Inspired by this problem, we introduce the Least-Dependency Constrained Spanning Tree problem, whose constraints are represented by a directed graph.

Let $G = (V, E)$ be a connected graph and $D = (E, A)$ be a digraph whose vertices are the edges of G. $e_1 \in E$ is a dependency of $e_2 \in E$ if $(e_1, e_2) \in A$. The *Least-Dependency Constrained Spanning Tree Problem* (L-DCST(G, D)) consists in deciding whether there is a spanning tree T of G such that each edge in T either has an empty dependency set or *at least* one of its dependencies is also in T. The corresponding optimization version, where a weighting function $w : E \to \mathbb{R}^+$ is considered and one wants to minimize the weight of tree, will be denoted L-DCMST(G, D, w).

Applications for this problem appear, for instance, in communication systems when a link can only be used if the message arrives through certain other links, due to protocol conversion restrictions on the nodes of the network [Viana 2016].

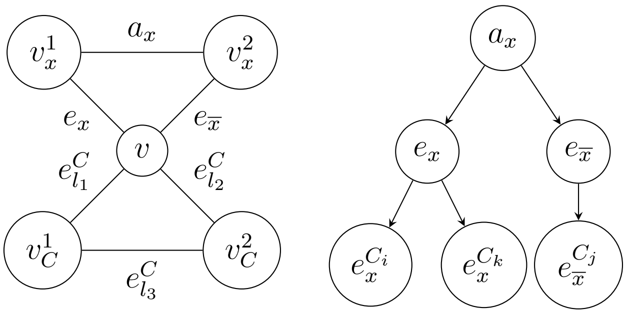
In this text, we use many concepts from graph theory. For those not familiar with them, we refer to classical books in the area such as [Douglas 2001].

2. NP-completeness

We show that L-DCST(G, D) is NP-complete even if the structure of both G and D are very simple. We use a reduction from 2 in 3 3-SAT.

An NP-complete variation of 3-SAT is 1 in 3 3-SAT which consists of deciding whether a formula can be satisfied with every clause having exactly one true literal [Gary and Johnson 1979]. We can define 2 in 3 3-SAT analogously. Note that 1 in 3 3-SAT can be reduced to 2 in 3 3-SAT by negating the literals of all clauses.

Given an instance of 2 in 3 3-SAT, we build an instance of L-DCST(G, D) as illustrated in Figure 1. We start G with a universal vertex v; for each variable x, v is connected to vertices v_x^1 and v_x^2 by edges e_x and $e_{\overline{x}}$, respectively, and v_x^1 and v_x^2 are neighbors by edge a_x; for each clause $C = l_1 \vee l_2 \vee l_3$, v is connected to vertices v_C^1 and v_C^2 by edges $e_{l_1}^C$ and $e_{l_2}^C$, respectively, while v_C^1 and v_C^2 are linked by edge $e_{l_3}^C$. We build D as follows: there are arcs from a_x to e_x and $e_{\overline{x}}$, for each variable x; there is an arc from e_l to e_l^C if literal l occurs in clause C. Note that G is a chordal cactus (actually, a union of triangles whose pairwise intersection is v), and D is a union of arborescences.



(a) Graph G. (b) Digraph D

Figure 1. Illustration of the 2 in 3 3-SAT **reduction.**

One can check that the 2 in 3 3-SAT instance has a "yes" answer if, and only if, the corresponding instance L-DCST(G, D) is feasible, which leads to the following theorem.

Theorem 1 *L-DCST(G, D) is NP-complete, even if G is a chordal cactus whose diameter is 2, and D is a union of arborescences whose height is 2.*

Notice that G is planar and has arbitrary $\Delta(G)$. We can rearrange its triangles to get $\Delta(G) = 3$. We make the triangles (related to clauses and variables) disjoint and link them as in Figure 2.

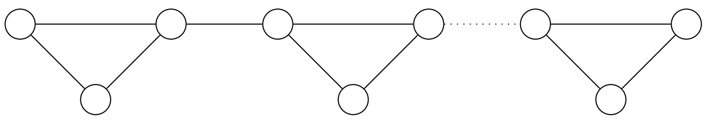
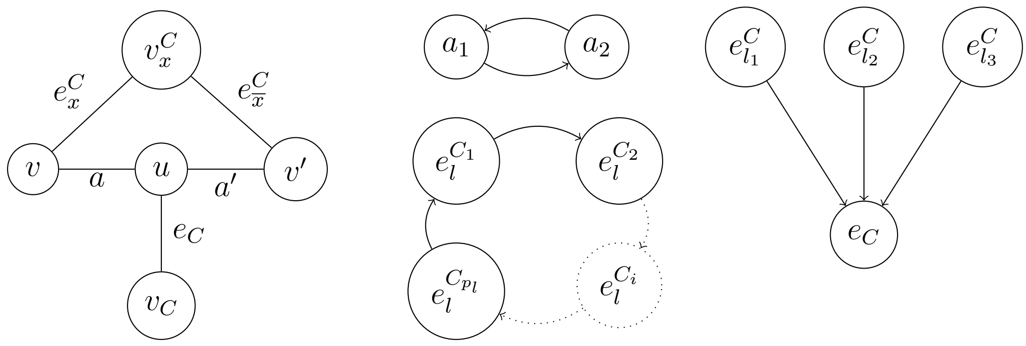


Figure 2. Illustration of the 2 in 3 3-SAT **reduction with** $\Delta(G) = 3$**.**

We now present a reduction to L-DCMST(G, D) from 3-SAT. It gives other insights on the hardness of the problem.

Given a formula, an instance of L-DCMST(G, D) is built as follows. G has vertices u, v and v', and edges $a = \{u, v\}$ and $a' = \{u, v'\}$; for each clause C, there is a vertex v_C and an edge $e_C = \{u, v_C\}$; for each variable x and each clause C containing x or $\overline{x}$, there is a vertex v_x^C and the edges $e_x^C = \{v, v_x^C\}$ and $e_{\overline{x}}^C = \{v', v_x^C\}$. In the digraph D, a and a' are mutually dependent; for each clause $C = l_1 \vee l_2 \vee l_3$, $e_{l_1}^C$, $e_{l_2}^C$ and $e_{l_3}^C$ are the dependencies of e_C; for each literal l, there is an oriented cycle with vertices e_l^C, for all C containing l. See Figure 3.

A truth valuation of the formula clearly translates into a feasible tree. Conversely, let T be a feasible tree. Due to connectivity of T and the mutual dependencies between



(a) Graph G.

(b) Digraph D. $C_1, \ldots, C_i, \ldots, C_{p_l}$ contain l.

Figure 3. Illustration of the 3-SAT reduction.

a and a', these two edges as well as e_C, for all clause C, must be chosen. Then, if $C = l_1 \vee l_2 \vee l_3$, the oriented star in D related to C ensures that e_l^C is in T, for some $l \in \{l_1, l_2, l_3\}$, and so $e_{\overline{l}}^C$ is not in T (since T is acyclic). Besides, the oriented cycle in D related to l guarantees that e_l^C is in T, for all C containing l. This indicates a consistent value for every variable that makes the formula satisfiable. Therefore, the following result holds.

Theorem 2 *L-DCMST(G, D) is NP-Complete, even if G is planar bipartite with diameter 3, and D is a union of oriented cycles and anti-arborescences of height 1.*

3. Polynomial cases

We present two cases where L-DCMST(G, D, w) is solvable in polynomial time. In both of them, we restrict the structure of D so that the problem can be decomposed in a polynomial number of Minimum Spanning Tree subproblems.

Theorem 3 *If D has $O(\log_2(|V(G)|))$ components, and each of them is either an oriented cycle or an arborescence whose subjacent graph is a star, DCMST(G, D, w) can be solved in polynomial time.*

Proof. If D is a directed cycle, either G is a tree or L-DCMST(G, D, w) is infeasible. When D is an arborescence whose subjacent graph is a star, we delete its root r and contract the related edge of G, obtaining G' (we do this again in case r is a leaf, obtaining G'') and solve the Minimum Spanning Tree Problem for the resulting graph G' (G'').

When D is a union of oriented cycles and directed stars, we have a choice to make for each of its (say, k) components: for each directed cycle, we decide to include none or all of its edges in the solution; for each directed star, we decide to include or not the root of D (and its unique descendant node, if the root is a leaf) and then allow or forbid all the other vertices of the star (edges of G) to take part of the solution. This way, we have 2^k subproblems to consider, each of them solvable in polynomial time. If $k = O(\log_2(|V(G)|))$, then all these subproblems can be solved in polynomial time. ∎

Notice the importance of restricting the number of components of D in the above theorem. Theorem 2 shows that the problem may become hard otherwise.

Theorem 4 *L-DCMST(G, D, w) can be solved in polynomial time, if D is an arborescence whose subjacent graph is a caterpillar.*

Proof. Let $D = (P \cup L, A)$ be an arborescence, where P induces the main path and L comprises the leaves of the subjacent caterpillar. First, suppose that D is rooted at $p_1 \in P$. Then, p_1 has exactly one or two neighbors in P.

In the first case, consider $P = \{p_1, p_2, \ldots, p_k\}$. Note that there is an arc from p_i to p_{i+1}, $1 \leq i \leq k-1$. Let $L_i \subseteq L$ be the set of leaves adjacent to p_i, $1 \leq i \leq k$. Observe that L-DCMST(G, D, w) can be decomposed into k subproblems. The ith subproblem is the Minimum Spanning Tree Problem where the edges from $\{p_1, p_2, \ldots, p_i\}$ *must* be chosen and the edges from $L_1 \cup \cdots \cup L_i$ *can* be chosen (it may be infeasible if the first set induces a cycle in G). Since each of these subproblems can be solved in polynomial time, and there are k of them, L-DCMST(G, D, w) can be solved in polynomial time with its optimal solution being the one whose cost is minimum among the optimal solutions of the feasible subproblems.

If p_1 has two neighbors in P, consider $P = \{p_1, p_1^1, p_2^1, \ldots, p_k^1, p_1^2, p_2^2, \ldots, p_l^2\}$, $k + l = |P| - 1$, such that $p_1, p_1^1, p_2^1, \ldots, p_k^1$ and $p_1, p_1^2, p_2^2, \ldots, p_l^2$ are directed paths in D. Also, consider $L_1 \subseteq L$ as the set of leaves incident to p_1, and let $L_i^1, L_j^2 \subseteq L$ be the analogously defined sets for p_i^1 and p_j^2, respectively, $1 \leq i \leq k$, $1 \leq j \leq l$. Note L-DCMST(G, D, w) can be decomposed into kl subproblems. We index the subproblems with tuples (i, j), $1 \leq i \leq k$, $1 \leq j \leq l$. The subproblem (i, j) is the Minimum Spanning Tree Problem for the subgraph of G containing p_1, p_a^1, p_b^2, L_a^1 and L_b^2 edges, $1 \leq a \leq i$, $1 \leq b \leq j$, which is infeasible if the p_1, p_a^1 and p_b^2 edges induce a cycle. Similarly as in the first case, L-DCMST(G, D, w) can be solved in polynomial time with its optimal solution being the one whose cost is minimum among the optimal solutions of the feasible subproblems.

To finish the proof, we consider D rooted at an L vertex. It is clear that the corresponding edge must be part of any feasible solution for DCMST(G, D, w). Thus, we contract it and fall back into the previous cases. ■

References

Darmann, A., Pferschy, U., Schauer, J., and Woeginger, G. J. (2011). Paths, trees and matchings under disjunctive constraints. *Discrete Applied Math.*, 159(16):1726 – 1735.

Douglas, B. (2001). *West. Graph Theory.* Prentice Hall, Upper Saddle River, NJ.

Gary, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-completeness.*

Samer, P. and Urrutia, S. (2015). A branch and cut algorithm for minimum spanning trees under conflict constraints. *Optimization Letters*, 9(1):41–55.

Viana, L. A. (2016). Árvore geradora com dependências mínima. Master's thesis, Federal University of Ceará.

Zhang, R., Kabadi, S., and Punnen, A. (2011). The minimum spanning tree problem with conflict constraints and its variations. *Discrete Optimization*, 8(2):191 – 205.