An FPT algorithm for node-disjoint subtrees problems parameterized by treewidth

Julien Baste — Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
Dimitri Watel — ENSIIE, 1 square de la Résistance, 91000, Evry, France — SAMOVAR, 9 Rue Charles Fourier, 91000, Évry, France

Abstract

We introduce a problem called MINIMUM SUBTREE PROBLEM WITH DEGREE WEIGHTS, or MTDW, that generalizes covering tree problems. It consists, given an undirected graph $G = (V, E)$, in the search of a forest $(T_1, T_2, \dots, T_\ell)$ containing ℓ node-disjoint trees of G satisfying some constraints and minimizing an objective function that depend only on the degree of the nodes in the trees.

We investigate MTDW with regard to parameterized complexity with respect to four parameters that are the number of constraints, the maximum degree after which the constraints are constant, the value ℓ, and the treewidth of the input graph G. For this problem, we provide a first dichotomy P versus NP-hard and a second dichotomy FPT versus W[1]-hard depending whether each of these parameters is constant, considered as a parameter, or disregard.

Particularly, our results show that most of the subproblems of MTDW are FPT by treewidth.

1 Introduction

There exists a real variety of spanning tree, Steiner tree and more generally covering tree problems that can be found in the literature. These problems find applications in particular in network routing. In each such problem, the objective is to find a subtree in a graph satisfying some constraints and minimizing an objective. Well-known examples are the MINIMUM UNDIRECTED STEINER TREE problem (UST); the k-MINIMUM SPANNING TREE problem (k-MST) [1]; the PRIZE COLLECTING STEINER TREE problem (PCST) [3]; the MINIMUM BRANCH VERTICES problem (MBV) [4]; the BUDGET STEINER TREE PROBLEM WITH PROFITS (BSTP) [5]; the k-BOTTLENECK STEINER TREE (k-BST) [2]; or the MINIMUM LEAF SPANNING TREE problem (MLST) [6].

A natural question to ask is how hard are those problems and their variants when the graph is close to a tree. A way to describe the distance between a graph and a tree is the treewidth, introduced by Robertson and Seymour [7], and actively used in parameterized complexity of graph optimization problems [8, 9]. It was proved, for instance, that UST, PCST, MLST and k-MST are FPT with respect to the treewidth [10, 11, 1]. No such result seems to exist for BSTP, k-BST or MBV. However, note that MBV is a generalization of the HAMILTONIAN PATH problem which is also FPT in the treewidth [9]. This paper aims to explore the fact that all those problems can be described (or rewritten) only by looking at the degree of the nodes of the graph in the tree. As shown in the following sections, this common property makes all those problems, and most of their variants including their node-disjoint counterpart, FPT by treewidth.

The Minimum node-disjoint subTrees problem with Degree Weights (MTDW). We introduce the MINIMUM NODE-DISJOINT SUBTREES PROBLEM WITH DEGREE WEIGHTS (MTDW) in Definition 1. This problem encodes many kinds of constraints (for instance spanning, bounded degree or cost constraints), that must be satisfied by a feasible forest, by associating to each node a set of scores depending on the degree of the node in the forest. We then get a set of scores of the tree by summing the scores of the nodes. One of the scores is used to define an objective function that must be minimized, and the others are used to define a set of constraints.

Given an undirected graph G and a node of v, we denote by $d_G(v)$ and $\gamma_G(v)$ the degree and the set incident edges of v in G respectively.

Definition 1 (MTDW problem)**.** *Given an undirected graph $G = (V, E)$ with n nodes, two integers $m, \ell \in \mathbb{N}$, m mappings $C_j : V \times \mathbb{N} \to \mathbb{Z}$, $j \in [\![1, m]\!]$, one mapping $D : V \times \mathbb{N} \to \mathbb{Z}$, and m integers $K_j \in \mathbb{Z}$ for $j \in [\![1, m]\!]$, we search for ℓ node-disjoint trees $T_1, T_2, \dots T_\ell$ such that for each $(i, j) \in [\![1, \ell]\!] \times [\![1, m]\!]$, $\sum_{v \in V} C_j(v, d_{T_i}(v)) \leq K_j$, minimizing $\sum_{i=1}^{\ell} \sum_{v \in V} D(v, d_{T_i}(v))$.*

Note that all the nodes of the graph intervene in the formulae, including those for which $d_{T_i}(v) = 0$, $i \in [\![1, \ell]\!]$. For instance, Minimum Leaf Spanning Tree can be rewritten as a subproblem of MTDW with $m = 1$ and $\ell = 1$. C_1 is a spanning tree constraint: $C_1(v, 0) = 1$, $C_1(v, d \geq 1) = 0$ and $K_1 = 0$. We minimize the number of leaves with D: $D(v, 1) = 1$ and $D(v, d \neq 1) = 0$.

The search of a feasible solution of an instance of MTDW. Every hardness result of this paper deals with the search of such a solution. In the following, we call MTDW_{FS} the decision problem in which, given an instance of MTDW, we must decide whether there exists a feasible solution *i.e.* ℓ node-disjoint trees $T_1, T_2, \dots T_\ell$ such that for each $(i, j) \in [\![1, \ell]\!] \times [\![1, m]\!]$, $\sum_{v \in V} C_j(v, d_{T_i}(v)) \leq K_j$, whatever the cost $\sum_{i=1}^{\ell} \sum_{v \in V} D(v, d_{T_i}(v))$ of that solution is.

Note that, when $\ell = 1$, the decision version of MTDW (deciding whether there exists a solution of cost at most K for some value K) can be easily described by replacing D with a constraint (m is then increased by 1). An interesting property is that this problem is equivalent to MTDW_{FS}.

The parameters. In this paper, we analyze the MTDW problem with regard to four parameters: m (the number of constraints), Δ (the minimum degree above which all the mappings C_j and D are constant: for every $j \in [\![1, m]\!]$, $v \in V$ and $d \geq \Delta$, $C_j(v, d) = C_j(v, \Delta)$, and $D(v, d) = D(v, \Delta)$), ℓ (the number of node-disjoint trees we are looking for), and tw (the treewidth of G).

Throughout the paper, we distinguish three possible cases for a parameter of MTDW. Either it is fixed as a constant, it is a parameter, with regard to the parameterized complexity point of view, or it is disregarded in the complexity analysis.

The encoding of the constraints. A last element that affects the complexity results in this paper is the encoding of the values K_j and $C_j(v, d)$ for every $j \in [\![1, m]\!]$, $v \in G$ and $d \leq d_G(v)$.

Let $\mu = \max_{j=1}^{m} \sum_{v \in V} \sum_{d \leq d_G(v)} |C_j(d, v)|$.

2 Results

In Theorem 1, we show that when μ is written in binary, the MTDW problem is hard, even when restricted to instances where ℓ, m, tw, and Δ are constants.

Theorem 1. MTDW_{FS} *is* NP*-hard when μ is written in binary, even when restricted to instances where ℓ, m,* tw*, and Δ are bounded by constants.*

In the following, we assume that μ is unary encoded. Moreover we may assume, without loss of generality, that $|K_j| \leq \mu$ for every $j \in [\![1, m]\!]$, as, otherwise, either the $j - th$ constraint is necessarily satisfied or necessarily unsatisfied, thus the encoding of those integers is never given. Note also that the values of the objective function D may be binary encoded and does not depend on the encoding of μ.

Classical complexity analysis. We show that the problem is NP-hard even when some of the parameters we considered are fixed to be constant. We also provide a generic algorithm whose running time is expressed with regard to ℓ, m, tw, Δ, n and μ.

Theorem 2. MTDW$_{FS}$ *is NP-hard even when ℓ, m, and Δ are constants.*

Theorem 3. MTDW$_{FS}$ *is NP-hard even when ℓ, Δ, and tw are constants.*

Theorem 4. *There exists an algorithm solving* MTDW *in time*

$$\mathcal{O}(\ell^2 \cdot \min(\ell, \mathit{tw})^2 \cdot (2\mu+1)^{2\cdot m\cdot\min(\ell,\mathit{tw})} \cdot (\Delta+1)^{2\cdot \mathit{tw}} \cdot \mathit{tw}^{4\mathit{tw}+1} \cdot n).$$

By combining Theorems 2, 3 and 4, we obtain the dichotomy for the MTDW problem depending on whether ℓ, m, tw, and Δ are fixed to be constant or not summarized in Table 1.

	tw disregarded	tw constant
m constant	NP-hard	polynomial
m disregarded	NP-hard	NP-hard

Table 1: The complexity of MTDW depending on whether m and tw are constant or not. The results remain the same, whatever the status of Δ and ℓ is.

Parameterized complexity analysis. In this paragraph, we will focus on the parameterized point of view when the considered parameters are ℓ, m, tw, and Δ. Regarding Table 1, it remains to study the cases where either m is a parameter or tw is a parameter.

Theorem 5. MTDW$_{FS}$ *is W[1]-hard parameterized by m, even when ℓ, Δ, and tw are constants.*

Because of Theorem 2, we know that if tw is not a parameter, the problem is NP-hard, while, using Theorem 4, if tw is a parameter together with m, then there is an XP algorithm to solve MTDW. This is summarized in Table 2.

	tw disregarded	tw as a parameter	tw constant
m as parameter	NP-hard even for fixed m	W[1]-hard XP algorithm	W[1]-hard XP algorithm

Table 2: The parameterized complexity of MTDW$_{\mathrm{FS}}$ when m is a parameter. The results remain the same, whatever the status of Δ and ℓ is.

With regard to the previous discussion, we now assume that m is constant. Using Table 1, it remains to deal with the case where tw is a parameter. We present two new hardness results.

Theorem 6. MTDW$_{FS}$ *is W[1]-hard parameterized by tw and ℓ, even when Δ and m are constants.*

	ℓ disregarded	ℓ as a parameter	ℓ constant
Δ disregarded	W[1]-hard	W[1]-hard	W[1]-hard
Δ as a parameter	W[1]-hard	W[1]-hard	FPT
Δ constant	W[1]-hard	W[1]-hard	FPT

Table 3: The parameterized complexity of MTDW when m is a constant and tw is a parameter. For all cases, the problem is NP-hard and can be solved by an XP algorithm.

Theorem 7. MTDW_{FS} *is W[1]-hard parameterized by* tw*, even when* ℓ *and* m *are constants.*

By combining Theorem 4, Theorem 6, and Theorem 7, we obtain the dichotomy of Table 3.

Altogether, Tables 1, 2, and 3 provide a complete analysis of the complexity of the MTDW problem with regard to the classical complexity and the parameterized complexity when taking into account ℓ, m, tw, and Δ as parameters. In particular, every subproblem of MTDW for which μ is unary encoded and ℓ, Δ and m are constants is FPT by tw. This is, for instance, the case of all the aforedmentioned subproblems of MTDW.

References

[1] R. Ravi, R. Sundaram, M. V. Marathe, D. J. Rosenkrantz, and S. S. Ravi. Spanning trees - Short or small. SIAM Journal on Discrete Mathematics, 9(2):178–200, 1996.

[2] A. K. Abu-Affash, P. Carmi, and M. J. Katz. Bottleneck Steiner tree with bounded number of Steiner vertices. Journal of Discrete Algorithms, 30:96–100, 2015.

[3] D. Bienstock, M. X. Goemans, D. Simchi-Levi, and D. Williamson. A note on the prize collecting traveling salesman problem. Mathematical Programming, 59(1-3):413–420, 1993.

[4] L. Gargano, P. Hell, L. Stacho, and U. Vaccaro. Spanning trees with bounded number of branch vertices. ICALP, pages 355–365, 2002.

[5] D. S. Johnson, M. Minkoff, and S. Phillips. Prize Collecting Steiner Tree problem: Theory and practice. Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms, pages 760–769, 2000.

[6] G. Salamon and G. Wiener. On finding spanning trees with few leaves. Information Processing Letters, 105(5):164–169, 2008.

[7] N. Robertson and P. D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. Journal of Algorithms, 7(3):309–322, 1986.

[8] M. Cygan, F. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. Parameterized Algorithms. Springer, Cham, 2015.

[9] R. Downey and M. Fellows. Parameterized complexity. Springer-Verlag New York, 1999.

[10] S. Arnborg, J. Lagergren, and D. Seese. Easy problems for treedecomposable graphs. Journal of Algorithms, 12(2):308–340, 1991.

[11] M. Chimani, P. Mutzel, and B. Zey. Improved Steiner tree algorithms for bounded treewidth. In Journal of Discrete Algorithms, volume 16, pages 67–78, 2012.