# On the Parameterized Complexity of Spanning Trees with Small Vertex Covers

Chamanvir Kaur and Neeldhara Misra(✉)

Indian Institute of Technology, Gandhinagar, Gujarat, India
{chamanvir.kaur,neeldhara.m}@iitgn.ac.in

**Abstract.** We consider the minimum power spanning tree (MPST) problem with general and unit demands from a parameterized perspective. The case of unit demands is equivalent to the problem of finding a spanning tree with the smallest possible vertex cover (MCST). We show that MPST is W[1]-hard when parameterized by the vertex cover of the input graph, and is W[2]-hard when parameterized by the solution size—the latter holds even in the case of unit demands. For the special case of unit demands, however, we demonstrate an FPT algorithm when parameterized by treewidth. In the context of kernelization, we show that even MCST is unlikely to admit a polynomial kernel under standard complexity-theoretic assumptions when parameterized by the vertex cover of the input graph.

**Keywords:** Vertex cover · Spanning trees · Treewidth

## 1 Introduction

A spanning tree is a minimally connected subgraph of a graph that spans all of its vertices. The problem of finding spanning trees of minimum weight is a fundamental algorithmic question and has received much attention. Variations of this question have also attracted a lot of interest, wherein one is interested in spanning trees with special structural properties, such as having bounded diameter [10], many leaves [4], or minimum poise (the sum of the diameter and the maximum degree [11]).

Our focus, in this work, is on the problem of finding spanning trees with small vertex covers. This is a special case of a more general problem which we also address, namely the minimum power spanning tree problem (MPST). Here, we are given an edge-weighted graph $\mathsf{G} = (\mathsf{V}, \mathsf{E})$, where the weights can be thought of as "demands", and the objective is to find a spanning tree and to assign "power" values to vertices such that all the edges of the spanning tree are covered. An edge is *covered* if the assigned power in one of its extremities is at least its demand. The goal is to minimize the sum of powers over all vertices.

This work is supported by the Science and Engineering Research Board (SERB), India.


M. Changat and S. Das (Eds.): CALDAM 2020, LNCS 12016, pp. 427–438, 2020.
https://doi.org/10.1007/978-3-030-39219-2_34

Observe that if all edges have unit demands, then this question is equivalent to finding a spanning tree with the smallest possible vertex cover. As an example, consider the complete graph, which has two spanning trees that are extreme from this perspective: the first is a star, which has a vertex cover of size one and the other is a Hamiltonian path, where we need roughly half the vertices of the graph to cover all the edges involved.

The minimum power variation of various optimization problems—notably, vertex cover (power vertex cover), Steiner trees (minimum power Steiner trees), and cut problems—are widely studied for their suitability in application scenarios. As a consequence of being more general, these problems usually model more sophisticated scenarios. Several applications arise in the context of connectivity questions in the domain of wireless networks and placements of sensors and cameras on road and home networks.

*Our Contributions.* In this contribution, we explore the parameterized complexity of the MPST problem and special cases. Our main result is that MPST is W[1]-hard when parameterized by the vertex cover of the input graph (Theorem 2), even when the weights are polynomially bounded in the size of the input. In fact, assuming ETH (Exponential Time Hypothesis), this also implies that there is no algorithm with a running time of $n^{o(\ell)}$, where $\ell$ denotes the size of the vertex cover.

Motivated by this intractibility, we turn to the special case with unit demands, which we refer to as the Minimum Cover Spanning Tree problem (MCST). We show that MCST is W[2]-hard when parameterized by the solution size (Theorem 1). On the other hand, we show an FPT algorithm when parameterized by the treewidth of the input graph (Theorem 3). In the context of kernelization, we show that even MCST is unlikely to admit a polynomial kernel under standard complexity-theoretic assumptions when parameterized by the vertex cover of the input graph (Corollary 3).

*Related Work.* The MPST problem, in the form that we propose and study it here, was considered by Angel et al. [2], who establish the hardness of approximation for this problem by a reduction from Dominating Set. In fact, our FPT reduction showing that MCST is W[2]-hard is in similar spirit. For a treatment of MCST from the perspective of approximation algorithms and on special classes of graphs, see [8].

Our result showing the hardness of MPST when parameterized by the vertex cover uses some ideas from the reduction employed by Angel et al. [3] to demonstrate that the Power Vertex Cover problem is also similarly intractable when parameterized by treewidth. The Power Vertex Cover problem is the natural "power"-based analog of the traditional vertex cover problem, where we seek an assignment of power values to the vertices, minimizing the total power assigned, so that the demand of every edge is met.

Closely related to MPST and MCST is the minimum power analog of the Steiner Tree problem, which has also been studied quite extensively (see, for instance, [1,9]). However, we remark that in these treatments, the demand of an

edge is met only if both of its endpoints receive a power value that is at least as much as its demand. For recent developments in approximation algorithms for power covering problems, see [5].

## 2 Preliminaries

We use $[n]$ to denote the set $\{1, 2, \ldots, n\}$. We follow standard notation and terminology from parameterized complexity [6] and graph theory [7]. We recall some of the definitions that will be relevant to our discussions.

The *neighborhood* of a vertex is denoted $N(v)$ and consists of all vertices $u$ adjacent to $v$. The *closed neighborhood* of a vertex is denoted $N[v]$ and is defined as $N(v) \cup \{v\}$. A *tree* is a connected and acyclic graph. Given a graph $G$, a *spanning tree* $T$ is a subgraph of $G$ such that $V(T) = V(G)$ and $T$ is a tree. For any two vertices $u$ and $v$, we let $d(u, v)$ denote the length of the shortest path between the vertices. For $S \subseteq V$, $G[S]$ denotes the graph *induced* by $S$ in $G$. The vertex set of $G[S]$ is $S$, and the edge set is $\{(u, v) \mid u \in S, v \in S \text{ and } (u, v) \in E\}$. We say that a vertex $v$ is *global* to a set $S$ of vertices if $v$ is adjacent to every vertex in $S$. We now turn to the description of some of the problems that will be considered in this contribution.

The Minimum Power Spanning Tree (MPST) problem is the following.

Minimum Power Spanning Tree (MPST)
**Input:** A graph $G$, a demand function $w : E \to \mathbb{R}^+$, and $k \in \mathbb{Z}^+$.
**Question:** Does $G$ admit a spanning tree $T$ and an assignment of power values $\rho : V(G) \to \mathbb{R}^+$ such that $\sum_{v \in V(G)} \rho(v) \leqslant k$ and for every edge $e \in E(T)$ with endpoints $u$ and $v$, $\max(\rho(u), \rho(v)) \geqslant w(e)$?

We now consider this problem in the context of unit demands. Noting that this is equivalent to finding a spanning tree with the smallest vertex cover, we refer to this as Minimum Cover Spanning Tree (MCST) problem.

Minimum Cover Spanning Tree (MCST)
**Input:** A graph $G$ and a positive integer $k$.
**Question:** Does $G$ admit a spanning tree $T$ with a vertex cover of size at most $k$?

The problems RBDS and MCIS (defined below) are known to be $W[2]$-hard and $W[1]$-hard, respectively, when parameterized by the solution size [6].

Red-Blue Dominating Set (RBDS)
**Input:** A bipartite graph $G = (R \cup B, E)$ and a positive integer $k$.
**Question:** Does there exist a subset $S \subseteq R$ of size at most $k$ such that for every $v \in B$, $|N(v) \cap S| \geqslant 1$?

MULTI-COLORED INDEPENDENT SET (MCIS)
**Input:** A $G = (V, E)$ and a partition of $V = (V_1, \ldots, V_k)$ into $k$ parts.
**Question:** Does there exist a subset $S \subseteq V$ such that $S$ is independent in $G$ and for every $i \in [k]$, $|V_i \cap S| = 1$?

We now turn to the notion of the treewidth of a graph.

**Definition 1.** *Let* $G$ *be a graph. A* tree-decomposition *of* $G$ *is a pair* $\mathbb{T} = (T, (B_t)_{t \in V(T)})$*, where* $T$ *is a rooted tree, and for all* $t \in V(T)$*,* $B_t \subseteq V(G)$ *such that*

- $\triangleright$ $\cup_{t \in V(T)} B_t = V(G)$,
- $\triangleright$ *for every edge* $xy \in E(G)$ *there is a* $t \in V(T)$ *such that* $\{x, y\} \subseteq B_t$*, and*
- $\triangleright$ *for every vertex* $v \in V(G)$ *the subgraph of* $T$ *induced by the set* $\{t \mid v \in B_t\}$ *is connected.*

*The* width *of a tree decomposition is* $\max_{t \in V(T)} |B_t| - 1$ *and the* treewidth *of* $G$ *is the minimum width over all tree decompositions of* $G$ *and is denoted by* $\mathbf{tw}(G)$.

For completeness, we also define here the notion of a *nice tree decomposition with introduce edge nodes*, as this is what we will work with in due course. We note that a given tree decomposition can be modified in linear time to fulfill the above constraints; moreover, the number of nodes in such a tree decomposition of width $w$ is $O(w \cdot n)$ [12].

**Definition 2.** *A tree decomposition* $\mathbb{T} = (T, (B_\alpha)_{\alpha \in V(T)})$ *is a* nice tree decomposition with introduce edge nodes *if the following conditions hold.*

1. *The tree* $T$ *is rooted and binary.*
2. *For all edges in* $E(G)$ *there is exactly one* introduce edge node *in* $T$*, where an introduce edge node is a node* $\alpha$ *in the tree decomposition* $T$ *of* $G$ *labeled with an edge* $\{u, v\} \in E(G)$ *with* $u, v \in B_\alpha$ *that has exactly one child node* $\alpha'$*; furthermore* $B_\alpha = B_{\alpha'}$.
3. *Each node* $\alpha \in V(T)$ *is of one of the following types:*
   - $\triangleright$ introduce edge node*: as described above;*
   - $\triangleright$ leaf node*:* $\alpha$ *is a leaf of* $T$ *and* $B_\alpha = \emptyset$*;*
   - $\triangleright$ introduce vertex node*:* $\alpha$ *is an inner node of* $T$ *with exactly one child node* $\beta \in V(T)$*; furthermore* $B_\beta \subseteq B_\alpha$ *and* $|B_\alpha \setminus B_\beta| = 1$*;*
   - $\triangleright$ forget node*:* $\alpha$ *is an inner node of* $T$ *with exactly one child node* $\beta \in V(T)$*; furthermore* $B_\alpha \subseteq B_\beta$ *and* $|B_\beta \setminus B_\alpha| = 1$*;*
   - $\triangleright$ join node*:* $\alpha$ *is an inner node of* $T$ *with exactly two child nodes* $\beta, \gamma \in V(T)$*; furthermore* $B_\alpha = B_\beta = B_\gamma$.

## 3 The Standard Parameter

In this section, we show that MCST is W[2]-hard with respect to the standard parameter (i.e, the solution size). In particular, we describe an FPT reduction from RBDS to MCST.

**Theorem 1.** MCST *is* W[2]*-hard when parameterized by the solution size.*

*Proof.* Let $(G, k)$ be an instance of RBDS where $G = (R \cup B, E)$. Without loss of generality, we assume that every vertex in $B$ has at least one neighbor in $R$, because if this is not the case, we may return a trivial No instance. We now describe the transformed instance of MCST, which we denote by $(H, k')$. The graph $H$ is obtained from $G$ by adding a global vertex to $R$, which in turn has $(k+2)$ new neighbors of degree one, which we refer to as *guards*. In particular, we have:

$$V(H) = V(G) \cup \{g, g_1, \ldots, g_{k+2}\},$$

and:

$$E(H) = E(G) \cup \{(g, g_1), (g, g_2), \ldots, (g, g_{k+2})\} \cup \{(g, v) \mid v \in R\}.$$

We let $k' = k + 1$. This completes the description of the reduction, and we now turn to a proof of the equivalence of the instances.

*The Forward Direction.* Let $S \subseteq R$ be a dominating set. For a vertex $v \in B$, let $p_v \in R$ denote a vertex from $S$ such that $p_v \in N(v)$. In the event that there are multiple vertices in $S$ that are adjacent to $v$, the choice of $p_v$ is arbitrary. Consider the spanning tree $T$ given by the following edges:

$$E(T) = \{(g, g_1), (g, g_2), \ldots, (g, g_{k+2})\} \cup \{(g, v) \mid v \in R\} \cup \{(v, p_v) \mid v \in B\}.$$

It is easy to verify that $\{g\} \cup S$ is a vertex cover of size $(k+1)$ for the edges of $T$. Indeed, it is evident that any edge in $T$ that is not incident to $g$ is incident to a vertex from $S$ and this concludes the argument in the forward direction.

*The Reverse Direction.* Let $T$ be a spanning tree of $H$ with a vertex cover $S$ of size $(k+1)$. Observe that the edge $(g, g_i)$ belongs to $T$ for any $1 \leqslant i \leqslant k+2$: if not, the vertex $g_i$ would be isolated in $T$. Therefore, we also conclude that $g \in S$, since it would otherwise be too expensive for $S$ to account for covering the edges incident to all the guards.

Let $S' := S \cap (R \cup B)$. By the argument above, we have that $|S'| \leqslant k$. For any $v \in B \cap S'$, let $p_v$ denote an arbitrarily chosen neighbor of $v$ in $R$. Now consider the set given by:

$$S^\star := (S \cap R) \ \cup \ \{p_v \mid v \in S \cap B\}.$$

It is easy to see that $|S^\star| \leqslant k$ and that $S^\star \subseteq R$ dominates all vertices in $B$. Indeed, consider any $v \in B$. If $v \in S \cap B$, then $p_v \in S^\star$ dominates $v$. If not, then observe that for some vertex $u \in N(v)$, the edge $(u, v)$ must belong to $T$ (if not, $v$ is isolated in $T$, a contradiction). By the case we are in, $v \notin S$, so to cover the edge $(u, v)$, we must have that $u \in S$. Further, since $G$ is bipartite, $u \in R$, thus $u \in (S \cap R)$. This implies, by construction, that $u \in S^\star$, and $u$ dominates $v$, as required. This concludes the proof. □

## 4 Vertex Cover and Treewidth

We begin by establishing that MPST is W[1]-hard when parameterized even by the vertex cover of the input graph (and therefore also its treewidth). Thereafter, we show a FPT algorithm for the special case of unit demands, i.e, the MCST problem, when parameterized by treewidth. We remark that the ideas in the reduction demonstrating the hardness parameterized by vertex cover are inspired by the construction used for showing the hardness of the power vertex cover problem when parameterized by treewidth [3].

**Theorem 2.** MPST *is* W[1]*-hard when parameterized by the vertex cover of the input graph.*

*Proof.* We reduce from MCIS. Let $(G, k)$ be an instance of MCIS where $G = (V, E)$ and further, let $V = (V_1, \ldots, V_k)$ denote the partition of the vertex set $V$. We assume, without loss of generality, that $|V_i| = n$ for all $i \in [k]$. Specifically, we denote the vertices of $V_i$ by $\{v_1^i, \ldots, v_n^i\}$. We are now ready to describe the transformed instance of MCST, which we denote by $(H, w, k')$. The construction is a combination of *choice gadgets* and *checker gadgets*.

*Choice Gadgets.* For each $1 \leqslant i \leqslant k$, introduce two vertices $x_i$ and $y_i$, and further, introduce $n$ vertices $z_1^i, \ldots, z_n^i$. We add the edges $(x_i, z_j^i)$ and $(y_i, z_j^i)$ for all $j \in [n]$. The weight function is defined as follows, for all $1 \leqslant j \leqslant n$.

$$w(u, z_j^i) = \begin{cases} j & \text{if } u = x_i, \\ n - j + 1 & \text{if } u = y_i. \end{cases}$$

In other words, for all $1 \leqslant i \leqslant k$ and $1 \leqslant j \leqslant n$, the vertex $z_j^i$ is adjacent to $x_i$ with an edge of weight $j$ and to $y_i$ with an edge of weight $(n-j+1)$. We refer to $x_i$ and $y_i$ as *anchors* and their neighbors in the choice gadget as *guards*.

*Checker Gadgets.* For each edge $e = (v_p^a, v_q^b) \in E(G)$, we introduce a vertex $c_e$, which is adjacent to the vertices $x_a, y_a, x_b, y_b$ in the choice gadgets. The edge weights are given by:

$$w(c_e, u) = \begin{cases} p + 1 & \text{if } u = x_a, \\ n - p + 1 & \text{if } u = y_a, \\ q + 1 & \text{if } u = x_b, \\ n - q + 1 & \text{if } u = y_b. \end{cases}$$

Our transformed instance comprises of all the $k$ choice gadgets and $m$ checker gadgets. Further, we add a vertex $g$ universal to all the anchor vertices and introduce $b$ vertices $\{g_1, \ldots, g_b\}$ that are adjacent only to $g$, where $b = nk + 2$. All edges incident to $g$ have a weight of one. This completes the description of the graph $H$. We refer the reader to Fig. 1 for a schematic depiction of the graph $H$.

The target total power, $k'$, is set to $nk+1$. Observe that the anchor vertices along with the vertex $g$ form a vertex cover of size $(2k+1)$ for $H$—indeed, all edges not incident to the universal vertices are either edges between anchors and guards, or between anchors and the vertices in the checker gadgets representing the edges of $G$. We now turn to a proof of equivalence.

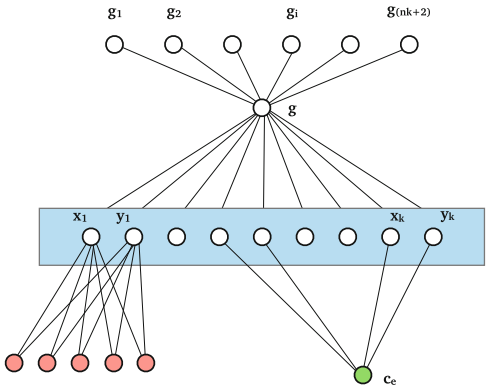


**Fig. 1.** An overview of the reduction used in the proof of Theorem 2. The anchor vertices are placed in the blue box, and along with the vertex $g$, form a vertex cover of size $2k+1$. The red vertices are the guards while the green vertex is an example of a vertex from a checker gadget. (Color figure online)

*Forward Direction.* Let $S \subseteq V$ be a multicolored independent set of $G$. Further, let $\tau : [k] \to [n]$ denote the choice of $S$ from the parts of $V$. Specifically:

$$S \cap V_i := \{v^i_{\tau(i)}\}.$$

We now describe our spanning tree $T$ and a power assignment $\rho$ of total cost at most $k' = nk+1$. First, we choose all edges incident on $g$. Next, from the choice gadget containing vertices $x_i$ and $y_i$, we pick the following edges:

$$\{(x_i, z^i_1), \ldots, (x_i, z^i_t), (y_i, z^i_{t+1}), \ldots, (y_i, z^i_n)\},$$

where $t := \tau(i)$. Before describing how we will connect $c_e$ to the structure developed so far, it will be useful to describe the power value assignment $\rho$. We have the following, where $i \in [k]$:

$$\rho(u) = \begin{cases} 1 & \text{if } u = g, \\ \tau(i) & \text{if } u = x_i, \\ n - \tau(i) & \text{if } u = y_i, \\ 0 & \text{for all other vertices.} \end{cases}$$

We now consider again the checker gadgets. Let $e = (v_p^a, v_q^b) \in E(G)$. Consider the case when $v_p^a \in S$ (the other scenarios are symmetric). Then $v_q^b \notin S$. Consider $\rho(x_b)$. We know that $\rho(x_b) \neq q$. If $\rho(x_b) < q$, then:

$$\rho(y_b) = n - \rho(x_b) > n - q \geqslant n - q + 1 = w(c_e, y_b),$$

and we choose the edge $(c_e, y_b)$ in our spanning tree, noting that its demand is met by $\rho$. On the other hand, if $\rho(x_b) > q$, then:

$$\rho(x_b) > q \geqslant q + 1 = w(c_e, x_b),$$

and we choose the edge $(c_e, y_b)$ in our spanning tree in this case. It is easy to verify that $T$ is indeed a spanning tree and that $\rho$ accounts for the demands of all the edges in $T$. Further, observe that the total power assigned by $\rho$ is $(nk + 1)$, as required, and this completes the argument in the forward direction.

*Reverse Direction.* Let $T$ be a spanning tree and let $\rho$ be a assignment of power values with total power value at most $(nk + 1)$. Notice that by an argument similar to the one used in the proof of Theorem 1, we have that:

$$\sum_{v \in V(H) \setminus \{g, g_1, \ldots, g_b\}} \rho(v) \leqslant nk. \tag{1}$$

This is to say that any valid assignment of power values must assign a power value of one or more to the vertex $g$. Therefore, we are left with $nk$ "units of power" to distribute amongst the $k$ choice gadgets and the $m$ checker gadgets. Our next claim is that if $C_i \subseteq V(H)$ denotes the set of vertices used in the $i^{th}$ choice gadget, then:

$$\sum_{v \in C_i} \rho(v) \geqslant n. \tag{2}$$

Indeed, suppose not. Fix $i \in [k]$ and suppose $\rho(x_i) = p$ and $\rho(y_i) = q$. If $p + q \geqslant n$, then there is nothing to prove, therefore, assume that $p + q < n$. This implies that for vertices $z_j^i$, where $p < j < n - q + 1$, $\rho(z_j^i) \geqslant 1$, since the assignment to the vertices $x_i$ and $y_i$ will not be enough to meet the demands of either of the edges incident to $z_j^i$. This implies that the total power assigned by $\rho$ in this gadget is:

$$p + q + ((n - q + 1) - p) - 1 = n,$$

and the claim follows. The inequalities (1) and (2) imply that every choice gadget is assigned a total power of $n$ according to $\rho$, which also gives us that $\rho(c_e) = 0$ for all $e \in E$.

Now, we choose our independent set as follows. Let $\tau(i)$ denote $\rho(x_i)$. Then from $V_i$ we choose the vertex $v_{\tau(i)}^i$. We claim that these chosen vertices must be independent in $G$. Indeed, suppose not, and in particular, suppose the vertices chosen from parts $a, b \in [k]$ are adjacent in $G$. Let the chosen vertices be $v_p^a$ and

$v_q^b$. Note that the power assignments made by $\rho$ on the relevant choice gadget can be inferred to be:

$$\rho(u) = \begin{cases} p & \text{if } u = x_a, \\ q & \text{if } u = x_b, \\ p' & \text{if } u = y_a, \\ q' & \text{if } u = y_b. \end{cases}$$

Here, we know that $p' \leqslant n - p$ and $q' \leqslant n - q$. Now consider the vertex $c_e$. The spanning tree $T$ must include one of the following edges, since these are the only edges incident to $c_e$:

$$\{(c_e, x_a), (c_e, y_a), (c_e, x_b), (c_e, y_b)\}.$$

However, recalling the weights of these edges from the construction, the fact that $\rho(c_e) = 0$, and given what we have inferred about the power values on the vertices $x_a, y_a, x_b$ and $y_b$, it is clear that $\rho$ does not meet the demand of any of these possible edges that is used by $T$ to span the vertex $c_e$. This is a contradiction, implying that the chosen vertices indeed form an independent set. This argument concludes the proof. □

Observing that the vertex cover of the reduced graph in the proof of Theorem 2 was bounded linearly in $k$, and based on the fact that there is no $n^{o(k)}$ algorithm for MCIS unless the ETH is false, we obtain the following consequence.

**Corollary 1.** *If there exists an algorithm which, given an instance* $(G = (V, E), w)$ *of* MPST *where* $G$ *has a vertex cover of size* $\ell$*, computes an optimal solution in time* $|V|^{o(\ell)}$*, then the ETH is false. This holds even if all weights are polynomially bounded.*

Our next result shows that when we restrict our attention to unit demands, then the problem of finding a minimum power spanning tree (equivalently, MCST) becomes tractable. Our approach is quite similar to the one used to show that Steiner Tree is FPT parameterized by treewidth, with two main differences: in the Steiner Tree problem, we need to explicitly track the subset of vertices that are "touched" by the solution, which we do not need to do since every vertex is effectively a terminal for the spanning tree problem. On the other hand, for the Steiner Tree problem, the cost of a solution is linked to its size in a rather straightforward manner, while for MCST, the structure of the edges involved in the spanning tree (or, for partial solutions, the spanning forests) is rather relevant. Therefore, we explicitly store not only the partition induced on a bag by a spanning forest, but also guess the specific edges from the bag that participate in the forest. This helps us keep track of the vertex cover as we go along, but it does affect the running time since our dynamic programming (DP) tables have $2^{\mathcal{O}(w^2)}$ rows in contrast with the $2^{\mathcal{O}(w \log w)}$ that turn out to be enough for Steiner Tree.

**Theorem 3.** MCST *admits an algorithm with running time* $2^{\mathcal{O}(w^2)}$ *where* $w$ *denotes the treewidth of the input graph.*

*Proof. (Sketch.)* Let $(G, k)$ be an instance of MCST with $G = (V, E)$, where $n := |V|$ and $m := |E|$. Let $\mathbb{T} = (T, \{B_t\}_{t \in V(T)})$ be a nice tree decomposition of $G$ (with introduce edge nodes) of width $w$. Let $t$ be a node of $V(T)$ and $B_t$ be the bag associated with it. Note that $|B_t| \leqslant w + 1$.

We use $G_t$ to denote the graph induced by the vertex set $\bigcup_{t'} B_{t'}$, where $t'$ ranges over all descendants of $t$, including $t$. By $E(B_t)$ we denote the edges present in $G[B_t]$. We use $H_t$ to denote the graph on vertex set $V(G_t)$ and the edge set $E(G_t) \setminus E(B_t)$.

Let $H$ be a subgraph of $G$. For a subset of vertices $X \subseteq V(H)$ and a partition $\mathcal{P}$ of $X$ into $s$ parts $X_1, \ldots, X_s$, we say that a spanning forest $F$ of $H$ is *compatible* with $\mathcal{P}$ if $F$ has exactly $s$ components $C_1, \ldots, C_s$ such that $X_i \subseteq C_i$ for all $i \in [s]$. We are now ready to describe the semantics of our DP table. For each $t \in V(T)$, for all possible partitions $\mathcal{P}$ of $B_t$, for each $F \subseteq E(B_t)$, $X \subseteq V(B_t)$ and $k \in [n]$, we let:

$$D[t, \mathcal{P}, F, X, k] = \begin{cases} 1 & \text{if there exists a spanning forest } F^\star \text{ of } G_t \text{ compatible} \\ & \text{with } \mathcal{P} \text{ with a vertex cover } S \text{ of size at most } k, \\ & \text{such that } F^\star \cap E(B_t) = F \text{ and } S \cap B_t = X, \\ 0 & \text{otherwise.} \end{cases}$$

Note that the solution to the problem is the smallest $k$ for which:

$$D[t, \mathcal{P}, F, X, k] = 1,$$

for any choice of $F$ and $X$, where $t$ is the root node and $\mathcal{P}$ is the partition with exactly one part. Also observe that the size of the DP table is $2^{\mathcal{O}^\star(w^2)} \cdot n^{\mathcal{O}^\star(1)}$. The recursive relationships are described in a standard fashion, but we briefly sketch some of the interesting cases.

The calculations for the leaf nodes and the introduce vertex nodes are trivial. When at an introduce edge node, if the introduced edge belongs to $F$, then it is important to check that $X$ includes one of its endpoints. At a forget node, if $v$ is the forgotten vertex and $\mathcal{P}$ is the partition under consideration, we review all rows in the child bag consistent with $\mathcal{P}$ on the vertices in $B_t$ but where $v$ belongs to one of the given parts, and ignore rows where $v$ is a standalone part. The value of $k$ will also have to be adjusted appropriately depending on whether $v$ was chosen in the vertex cover or not, and in all the cases where $v$ was not in the vertex cover, we need to run a sanity check to ensure that edges incident to $v$ as given by the row of the child node under consideration are covered by the current choice of vertex cover in $B_t$.

The most non-trivial case is the join nodes. Intuitively, the procedure for the join node involves "patching" the solutions from the two child nodes, while adjusting for the double-counting of the cost of vertex cover vertices in $X$. However, a straightforward patch may lead to cycles in our combined solution, which might also lead to suboptimal costs for the corresponding vertex covers. It can be verified that in this case, we can follow an approach similar to what is used in the case of Steiner Trees, except that the auxiliary graphs used for "acyclic

merges" have to now account for the edges given in F. Due to lack of space, defer a detailed description to a full version of this manuscript. □

We conclude this section by making some observations about the relationship between spanning trees that have small vertex covers and the vertex covers of the underlying graph.

**Observation 1.** *Every connected graph* G *admits a spanning tree with a vertex cover of size at most* $\kappa$*, where* $\kappa$ *is the size of a smallest connected vertex cover of* G*.*

*Proof.* Let $S \subseteq V(G)$ be a connected vertex cover of size at most $\kappa$. Let $T$ be a spanning tree for $G[S]$. For any vertex $v \in V \setminus S$, let $s_v$ denote an arbitrarily chosen vertex from $N(v)$. Note that $N(v) \subseteq S$ (since $S$ is a vertex cover) and $N(v) \neq \emptyset$ (since $G$ is connected). Consider the tree $T'$ obtained from $T$ by adding the edges $(v, s_v)$ for all $v \in V \setminus S$. Observe that $T'$ is a spanning tree and $S$ covers all edges of $T$, which is $|S| \leqslant \kappa$, as desired. This concludes the proof. □

Combined with the well-known fact that any connected graph that has a vertex cover of size $\ell$ also has a connected vertex cover of size at most $2\ell$ (for instance, by choosing all the non-leaf vertices of a depth-first search traversal, which is known to be a two-approximate vertex cover), we have the following.

**Corollary 2.** *Every graph admits a spanning tree with a vertex cover of size at most* $2\ell$*, where* $\ell$ *is the size of a smallest vertex cover of* G*.*

Although far from optimal—for instance, consider the complete graphs—the bound in Observation 1 is tight, as witnessed by the graph consisting of two stars on three leaves whose centers are connected by an edge. We remark that the size of the vertex cover of the transformed instance constructed in the proof of Theorem 1 is bounded by $(|R| + 1)$, which implies the following consequence in the context of kernelization when parameterized by the size of the vertex cover.

**Corollary 3.** *When parameterized by the solution size,* MCST *does not admit a polynomial kernel unless coNP* $\subseteq$ *NP/poly.*

## 5 Concluding Remarks

We showed that MPST is W[1]-hard when parameterized by the vertex cover of the input graph, and is W[2]-hard when parameterized by the solution size. For the special case of unit demands, however, we demonstrate an FPT algorithm when parameterized by treewidth. We also demonstrated the hardness of polynomial kernelization for MCST when parameterized by the vertex cover. For MPST, the dynamic programming algorithm presented would also give us a FPT algorithm for the combined parameter $(w, M)$, where $M$ is the maximum demand and $w$ is the treewidth. To achieve this, we would have to store all possible power assignments to the vertices in the bags.

Our contributions here for the MCST problem leave open several directions for future work. The most natural question is if we can improve the running time of the dynamic programming algorithm when parameterized by treewidth, possibly using randomized approaches such as "Cut and Count", which have proven to be successful for the related Steiner Tree problem. We also leave open the question of whether there is a deterministic single-exponential algorithm when parameterized by the vertex cover of the input graph. This seems to be an intuitively appealing possibility, and indeed, it is promising to pursue the natural approach where we guess the interaction of a possible optimal solution with a given vertex cover. We also leave open the question of obtaining Turing or lossy kernels for MPST or MCST when parameterized by vertex cover. Finally, we believe that it would be interesting to study these problems on special classes of graphs, especially those relevant to application scenarios, from a parameterized perspective. We note that the reduced instance in Theorem 1 is bipartite.

## References

1. Althaus, E., Călinescu, G., Mandoiu, I.I., Prasad, S.K., Tchervenski, N., Zelikovsky, A.: Power efficient range assignment for symmetric connectivity in static ad hoc wireless networks. Wireless Netw. **12**(3), 287–299 (2006)
2. Angel, E., Bampis, E., Chau, V., Kononov, A.: Min-power covering problems. In: Elbassioni, K., Makino, K. (eds.) ISAAC 2015. LNCS, vol. 9472, pp. 367–377. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48971-0_32
3. Angel, E., Bampis, E., Escoffier, B., Lampis, M.: Parameterized power vertex cover. In: Heggernes, P. (ed.) WG 2016. LNCS, vol. 9941, pp. 97–108. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53536-3_9
4. Bonsma, P.S., Zickfeld, F.: Improved bounds for spanning trees with many leaves. Discrete Math. **312**(6), 1178–1194 (2012)
5. Calinescu, G., Kortsarz, G., Nutov, Z.: Improved approximation algorithms for minimum power covering problems. In: Epstein, L., Erlebach, T. (eds.) WAOA 2018. LNCS, vol. 11312, pp. 134–148. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-04693-4_9
6. Cygan, M., et al.: Parameterized Algorithms. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21275-3
7. Diestel, R.: Graph Theory. GTM, vol. 173. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-53622-3
8. Fukunaga, T., Maehara, T.: Computing a tree having a small vertex cover. In: Chan, T.-H.H., Li, M., Wang, L. (eds.) COCOA 2016. LNCS, vol. 10043, pp. 77–91. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48749-6_6
9. Grandoni, F.: On min-power steiner tree. In: Epstein, L., Ferragina, P. (eds.) ESA 2012. LNCS, vol. 7501, pp. 527–538. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33090-2_46
10. Ho, J., Lee, D.T., Chang, C., Wong, C.K.: Minimum diameter spanning trees and related problems. SIAM J. Comput. **20**(5), 987–997 (1991)
11. Iglesias, J., Rajaraman, R., Ravi, R., Sundaram, R.: Plane gossip: approximating rumor spread in planar graphs. LATIN 2018. LNCS, vol. 10807, pp. 611–624. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-77404-6_45
12. Kloks, T. (ed.): Treewidth. LNCS, vol. 842. Springer, Heidelberg (1994). https://doi.org/10.1007/BFb0045375