The Curse of Connectivity: t-Total Vertex (Edge) Cover

Henning Fernau[1] and Fedor V. Fomin[2] and Geevarghese Philip[3] and Saket Saurabh[3]

[1] Universität Trier FB 4—Abteilung Informatik, 54286 Trier, Germany.
`fernau@uni-trier.de`
[2] Department of Informatics, University of Bergen, 5020 Bergen, Norway.
`fomin@ii.uib.no`
[3] The Institute of Mathematical Sciences, C.I.T Campus, Taramani, Chennai 600 113, India.
{`gphilip`|`saket`}`@imsc.res.in`

Abstract. We investigate the effect of certain natural connectivity constraints on the parameterized complexity of two fundamental graph covering problems, namely k-VERTEX COVER and k-EDGE COVER. Specifically, we impose the additional requirement that each connected component of a solution have at least t vertices (resp. edges from the solution), and call the problem t-TOTAL VERTEX COVER (resp. t-TOTAL EDGE COVER). We show that

- both problems remain fixed-parameter tractable with these restrictions, with running times of the form $\mathcal{O}^*\left(c^k\right)$ for some constant $c > 0$ in each case;
- for every $t \geq 2$, t-TOTAL VERTEX COVER has no polynomial kernel unless the Polynomial Hierarchy collapses to the third level;
- for every $t \geq 2$, t-TOTAL EDGE COVER has a linear vertex kernel of size $\frac{t+1}{t}k$.

These results significantly improve earlier work on these problems.

Our no-poly-kernel result for t-TOTAL VERTEX COVER, and the known NP-hardness result for t-TOTAL EDGE COVER, are in stark contrast to the fact that k-VERTEX COVER has a $2k$ vertex kernel, and that k-EDGE COVER is solvable in polynomial time. This illustrates how even the slightest connectivity requirement results in a drastic change in the tractability of problems — the curse of connectivity!

1 Introduction

The k-VERTEX COVER problem and the k-EDGE COVER problem are two related combinatorial problems that exhibit contrasting behavior in terms of their solvability. In the k-VERTEX COVER problem, we are given as input a graph G and a positive integer k, and are asked if there exists a set S of at most k vertices in G such that every edge in G is adjacent to at least one of the vertices in S; such an S is called a *vertex cover* of G. In k-EDGE COVER, we are again given as input a graph G and a positive integer k, and are asked if there exists a set S of at most k *edges* in G such that for each vertex v in G, there is at least one edge $e = \{u, v\} \in S$; such an S is called an *edge cover* of G.

The k-VERTEX COVER problem is one of the first problems that Karp showed to be NP-complete [17], and is one of the six basic NP-complete problems chosen for special mention by Garey and Johnson as being at the "core" of known NP-complete problems [14, Section 3.1]. The k-VERTEX COVER problem and its variants have been extensively investigated from the point of view of various algorithmic paradigms, including approximation and parameterized algorithms. In particular, k-VERTEX COVER is considered to be the drosophila of the field of parameterized complexity, where a large set of results of different kinds have been obtained on the problem and its variants [2, 8, 11, 13, 15, 19, 20, 23]. In sharp contrast, k-EDGE COVER has long been known to be solvable in polynomial time [26].

In this paper we investigate the parameterized complexity of variants of these problems where additional connectivity constraints are imposed on the solution S. More specifically, for each integer $t \geq 1$ we define variants of the two problems, named t-TOTAL VERTEX COVER and t-TOTAL EDGE COVER, respectively. In t-TOTAL VERTEX COVER (resp. t-TOTAL EDGE COVER), we ask whether there is a vertex cover (resp. edge cover) S of the input graph such that each connected component of the subgraph induced on S contains at least t vertices (resp. at least t edges from S). These problems were introduced by Fernau and Manlove [11] who initiated the study of the parameterized complexity of these problems. We significantly improve their results and obtain several new results: in particular, we complete the picture on how even the slightest connectivity requirement dramatically changes the complexity of these problems.

Independently, Małafiejski and Żylinski studied 2-TOTAL EDGE COVER as a model of weak cooperation of guards in an art gallery problem [22]. Both Fernau and Manlove [11] and Małafiejski and Żylinski [22] derived a Gallai type identity, namely, that under certain conditions, the sum of (i) the cardinality of the largest possible packing of a graph with vertex-disjoint copies of a path of length two and (ii) the size of the smallest 2-total edge cover, equals the number of vertices of the graph. A generalization of this result to all $t \geq 2$ was shown in [11]. Combining this with the result of Kirkpatrick and Hell [18], who proved that finding a packing of vertex-disjoint copies of trees on t edges in a graph is NP-hard, we get that for all $t \geq 2$, t-TOTAL EDGE COVER is NP-complete.

Our results. As noted above, k-EDGE COVER has been known to be solvable in polynomial time for over half a century. Fernau and Manlove [11] showed that the least possible connectivity requirement on S, namely that each connected component of the graph induced on S have at least 2 edges, makes the problem NP-hard.

We show a similar result for t-TOTAL VERTEX COVER, not in the context of classical complexity, but from the realm of parameterized complexity. It is a well-known result in parameterized complexity that k-VERTEX COVER has a so-called $2k$-sized vertex kernel(See Section 2). We show that adding a connectivity constraint results in a dramatic change in kernelizability: for any fixed $t \geq 2$, t-TOTAL VERTEX COVER has no polynomial-size kernel unless the Polynomial Hierarchy(PH) collapses to the third level, which is deemed unlikely in complexity theory. We complement the no-polynomial-kernel result with results on the fixed-parameter tractability of t-TOTAL VERTEX COVER and t-TOTAL EDGE COVER when parameterized by the solution size. More specifically, we show the following:

- t-TOTAL VERTEX COVER can be solved in $\mathcal{O}\left(16.1^{k+\mathcal{O}\left(\log^2 k\right)} \times n^{\mathcal{O}(1)}\right)$ time. To obtain these results we combine the classical result of Otter [27] on the number of unlabeled trees with a modification of the color-coding technique of Alon et al. [1].
- t-TOTAL EDGE COVER has a vertex kernel of size $\frac{t+1}{t}k$,
- t-TOTAL EDGE COVER can be solved in $\mathcal{O}\left(2^{\frac{t+1}{t}k+\mathcal{O}(\sqrt{k})} \times n^{\mathcal{O}(1)}\right)$ time. To obtain this result, we combine kernelization techniques with the classical result of Hardy & Ramanujan [16] and the Fast Fourier Transform [21].

These results significantly improve earlier work on these problems.

2 Preliminaries

In this section we state some basic definitions related to graph theory and parameterized complexity, and give an overview of the notation used in this paper. We use $G = (V, E)$ to refer to a graph with vertex set V (with $n := |V|$). A graph $G' = (V', E')$ is a *subgraph* of G if $V' \subseteq V$ and $E' \subseteq E$. The subgraph G' is called an *induced subgraph* of G if $E' = \{uv \in E \mid u, v \in V'\}$, in this case, G' is also called the subgraph *induced by* V' and is denoted $G[V']$. For a set $S \subseteq E$ of edges of G, we use $G\,(S)$ to denote the subgraph (V, S) of G. To describe running times of algorithms we sometimes use the $\mathcal{O}^*$ notation. Given $f : \mathbb{N} \rightarrow \mathbb{N}$, we define $\mathcal{O}^*(f(n))$ to be

$O(f(n) \cdot p(n))$, where $p(\cdot)$ is some polynomial function. That is, the $\mathcal{O}^*$ notation suppresses polynomial factors in the expression for the running time.

A parameterized problem Π is a subset of $\Sigma^* \times \mathbb{N}$, where Σ is a finite alphabet. An instance of a parameterized problem is a tuple (x, k), where k is called the parameter. A central notion in parameterized complexity is *fixed-parameter tractability (FPT)* which means, for a given instance (x, k), decidability in time $f(k) \cdot p(|x|)$, where f is an arbitrary function of k and p is a polynomial in the input size. The notion of *kernelization* is formally defined as follows.

Definition 1. [**Kernelization**] [25, 12] *A kernelization algorithm for a parameterized problem $\Pi \subseteq \Sigma^* \times \mathbb{N}$ is an algorithm that, given $(x, k) \in \Sigma^* \times \mathbb{N}$, outputs, in time polynomial in $|x| + k$, a pair $(x', k') \in \Sigma^* \times \mathbb{N}$ such that (a) $(x, k) \in \Pi$ if and only if $(x', k') \in \Pi$ and (b) $|x'|, k' \leq g(k)$, where g is some computable function. The output instance x' is called the kernel, and the function g is referred to as the size of the kernel. If $g(k) = k^{O(1)}$ (resp. $g(k) = O(k)$) then we say that Π admits a polynomial (resp. linear) kernel.*

3 Total Vertex Covers

For each positive integer t, the parameterized t-TOTAL VERTEX COVER problem is defined as follows:

t-TOTAL VERTEX COVER (TVC-t)

Input:	An undirected graph $G = (V, E)$, and a positive integer $k \geq t$.
Parameter:	k
Question:	Does there exist a set $S \subseteq V$ of at most k vertices of G such that (i) $G[V \setminus S]$ contains no edges (i.e., S is a vertex cover of G), and (ii) each connected component of $G[S]$ contains at least t vertices?

Such an S is called a t-total vertex cover (t-TVC) of G. The problem is NP-complete for all values of t: Clearly the problem is in NP for any value of t. For $t = 1$, t-TOTAL VERTEX COVER is the k-VERTEX COVER problem, and for $t = k$ it becomes the k-CONNECTED VERTEX COVER problem; these are two classical NP-complete problems [14, Problem GT1]. For $2 \leq t \leq k$, the t-TOTAL VERTEX COVER problem has been shown to be NP-hard by Fernau and Manlove [11, Theorem 3] by reduction from k-VERTEX COVER; we give an alternate proof of NP-hardness in Claim 1 below.

3.1 Kernelizability

In this section we investigate the kernelizability of t-TOTAL VERTEX COVER. Note that for $t = 1$, t-TOTAL VERTEX COVER is the k-VERTEX COVER problem, and for $t = k$ it becomes the k-CONNECTED VERTEX COVER problem. The former problem has a vertex kernel of size at most $2k$ [7], and the latter problem does not have a kernel of size k^c for any constant c unless PH collapses to the third level [10]. It turns out that this change in polynomial kernelizability occurs at the smallest value of t possible.

Theorem 1. *For each fixed $t \geq 2$, t-TOTAL VERTEX COVER has no kernel of size bounded by k^c, for any fixed constant c, unless* PH *collapses to the third level.*

To prove this, we need a few notions and results from the recently developed theory of kernel lower bounds [5, 6, 10]. We start off by defining a new problem:

RED-BLUE DOMINATING SET (RBDS)

Input:	An undirected bipartite graph $G = (R \uplus B, E)$, and a positive integer k.
Parameter:	$k + \|B\|$
Question:	Does there exist a set $D \subseteq R$ of at most k vertices of G such that every $v \in B$ is adjacent to some $u \in D$ (i.e., D is a dominating set of B)?

We will use RBDS as a source problem to show a kernel lower bound for t-TOTAL VERTEX COVER. Towards this end we state the following fact.

Fact 1. [10, Theorem 2] RED-BLUE DOMINATING SET *parameterized by* $(k + |B|)$ *does not admit a polynomial kernel unless* PH *collapses to the third level.*

We use a notion of reductions, similar in spirit to those used in classical complexity to show NP-hardness results, to show that t-TOTAL VERTEX COVER admits no polynomial kernel. We associate a classical decision problem with a parameterized problem in a natural way as follows:

Definition 2. [**Derived Classical Problem**] [6] *Let* $\Pi \subseteq \Sigma^* \times \mathbb{N}$ *be a parameterized problem, and let* $1 \notin \Sigma$ *be a new symbol. We define the* derived classical problem *associated with* Π *to be* $\{x1^k | \, (x, k) \in \Pi\}$.

Definition 3. [6] *Let* P *and* Q *be parameterized problems. We say that* P *is polynomial time and parameter reducible to* Q*, written* $P \leq_{Ptp} Q$*, if there exists a polynomial time computable function* $f : \Sigma^* \times \mathbb{N} \to \Sigma^* \times \mathbb{N}$*, and a polynomial* $p : \mathbb{N} \to \mathbb{N}$*, and for all* $x \in \Sigma^*$ *and* $k \in \mathbb{N}$*, if* $f((x, k)) = (x', k')$*, then* $(x, k) \in P$ *if and only if* $(x', k') \in Q$*, and* $k' \leq p(k)$*. We call* f *a polynomial parameter transformation (or a PPT) from* P *to* Q*.*

This notion of a reduction is useful in showing kernel lower bounds because of the following theorem:

Fact 2. [6, Theorem 3] *Let* P *and* Q *be parameterized problems whose derived classical problems are* P^c, Q^c*, respectively. Let* P^c *be* NP*-complete, and* $Q^c \in$ NP*. Suppose there exists a PPT from* P *to* Q*. Then, if* Q *has a polynomial kernel, then* P *also has a polynomial kernel.*

Now we are ready to prove Theorem 1.

Theorem 1. *For each fixed* $t \geq 2$*,* t-TOTAL VERTEX COVER *has no kernel of size bounded by* k^c*, for any fixed constant* c*, unless* PH *collapses to the third level* [4].

Proof. We begin by noting that by a simple reduction from the NP-complete SET COVER problem [14], the derived classical problem corresponding to RED-BLUE DOMINATING SET is NP-complete. Also, the derived classical problem corresponding to t-TOTAL VERTEX COVER is evidently in NP. Now suppose t-TOTAL VERTEX COVER has a polynomial kernel, and that there is a PPT from RED-BLUE DOMINATING SET to t-TOTAL VERTEX COVER. Then by Fact 2, RED-BLUE DOMINATING SET has a polynomial kernel, and hence by Fact 1 PH collapses to the third level. Thus t-TOTAL VERTEX COVER does not have polynomial kernel unless PH collapses to the third level. Hence to prove the theorem, it suffices to show that there is a PPT from RED-BLUE DOMINATING SET to t-TOTAL VERTEX COVER. We now proceed to give such a reduction.

---

[4] We note in passing that it was wrongly claimed in [11] that the problem has a kernel of size $\mathcal{O}(k(k + t))$.

Given an instance $(G = (R \uplus B, E), k)$ of RED-BLUE DOMINATING SET, we construct an instance of t-TOTAL VERTEX COVER as follows: If B contains isolated vertices then (G, k) is a NO instance, and in this case we construct a trivial NO instance of t-TOTAL VERTEX COVER. Otherwise, we add a distinct path of length (number of edges) $t - 1$ starting from each vertex $v \in B$. Thus, if $t = 2$, then we attach a new, distinct pendant vertex w_i to each $v_i \in B$; if $t = 3$, then we add a path $\left(v_i, u_i^1, w_i\right)$ to each $v_i \in B$. In general, for $t \geq 2$, we add a path $\left(v_i, u_i^1, u_i^2, \ldots u_i^{t-2}, w_i\right)$ to each $v_i \in B$, where the vertices u_i^j and w_i are all new and distinct: see Fig. 1 for an illustration. We call the resulting graph H and $(H, k + (t-1)|B|)$ is the constructed instance of t-TOTAL VERTEX COVER. To complete the proof, we need to show:

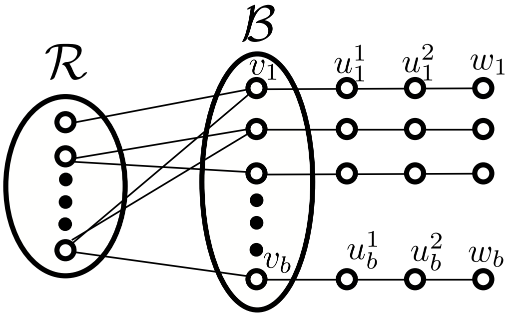

Fig. 1. The reduced instance for $t = 4$.

Claim 1. Let $(G = (R \uplus B, E), k)$ be an instance of RED-BLUE DOMINATING SET, and t a fixed positive integer. Let H be the graph constructed from G as described above. Then (G, k) is a YES instance of RED-BLUE DOMINATING SET if and only if $(H, k + (t-1)|B|)$ is a YES instance of t-TOTAL VERTEX COVER.

Proof. Let $(G = (R \uplus B, E), k)$ be a YES instance of RED-BLUE DOMINATING SET. Then there is an inclusion-minimal set $D \subseteq R$, $|D| \leq k$, that dominates B. Let S be the set of all new vertices added by the construction to H, except for the pendant vertex w_i in each new path. Thus, e.g., $S = \emptyset$ when $t = 2$, and in general $|S| = (t-2)|B|$. Define $C = D \cup B \cup S$. Now,

1. $|C| = |D| + |B| + |S| \leq k + |B| + (t-2)|B| \leq k + (t-1)|B|$.
2. C is a vertex cover of H: $B \subseteq C$ covers all original edges, and all new edges adjacent to vertices in B; S covers the rest of the new edges, if any.
3. Each connected component of $H[C]$ contains at least t vertices:
   (a) Since D dominates B in G, any vertex $v_i \in B$ has at least one neighbor $w \in D \subseteq C$ in $H[C]$. Then v_i, w, and the $t-2$ new vertices $\left\{u_i^1, u_i^2, \ldots u_i^{t-2}\right\} \subseteq S$ are all part of the same component in $H[C]$, as witnessed by the path $\left\langle w, v_i, u_i^1, u_i^2, \ldots u_i^{t-2}\right\rangle$. Thus each vertex $v_i \in B$ is part of a connected component of size at least t in $H[C]$.
   (b) D is an inclusion-minimal dominating set of B and $D \cup B \subseteq C$, and so each $w \in D$ has at least one neighbor $v_i \in B$ in the graph $H[C]$. It follows that each vertex $w \in D$ is part of a connected component of size at least t in $H[C]$, namely the component to which v_i belongs.
   (c) Each vertex $u_i^j \in S$ is in the same component in $H[C]$ as the vertex $v_i \in B$, and so u_i^j is part of a connected component of size at least t in $H[C]$, namely the component to which v_i belongs.

Thus C is a t-TVC of H, of size at most $k + (t-1)\,|B|$. This proves the forward direction.

To prove the reverse direction, suppose $(H, k + (t-1)\,|B|)$ is a YES instance of t-TOTAL VERTEX COVER, and let C be a t-TVC of H of size at most $k + (t-1)\,|B|$. Consider any path $P = \left(u_i^0 = v_i, u_i^1, u_i^2, \dots u_i^{t-2}, w_i\right)$ in H consisting of a vertex $v_i \in B$ and new vertices added by the construction. Since C is a vertex cover such that each connected component of $G[C]$ has at least t vertices, we have that $|P \cap C| \geq t-1$. Now suppose there exists a vertex $x \in P \setminus C$. If $x = u_i^j$ for some $0 \leq j \leq t-2$, then the vertices $u_i^{j+1}, u_i^{j+2}, \dots, u_i^{t-2}, w_i$ form a connected component of $H\,[C]$ of size strictly less than t, a contradiction. So one of the following holds:

1. All the t vertices of P are in C, or,
2. $\left\{v_i, u_i^1, u_i^2, \dots, u_i^{t-2}\right\} \in C$, and $w_i \notin C$.

Let p_t be the number of paths of the first kind in H, and let p_{t-1} be the number of such paths of the second kind. There is exactly one such path corresponding to each vertex of B, and so $p_t + p_{t-1} = |B|$. The total number of vertices contributed to C by these paths is $tp_t + (t-1)\,p_{t-1}$, and so the number of vertices in $C \cap R$ is at most $k + (t-1)\,|B| - (tp_t + (t-1)\,p_{t-1}) = k - p_t$. Now let P be a path of the second kind. By definition, $|P \cap C| = (t-1)$, and since each connected component of $G\,[C]$ has at least t vertices, there is at least one more vertex x in C that is adjacent in H to one of the vertices, say y, of $P \cap C$. The only possibility is that $x \in R$ and $y \in B$, and so at most $k - p_t$ vertices in $C \cap R$ dominate $p_{t-1} = |B| - p_t$ vertices in B. Since each vertex of B has a neighbor in R, it follows that at most k vertices in R dominate all of B.

This completes the proof of the claim and hence of the theorem. □

The reduction employed in the above argument also implies:

Corollary 1. *For each fixed $t \geq 2$,* t-TOTAL VERTEX COVER *restricted to bipartite graphs as input instances is* NP-*hard, and does not admit a polynomial kernel unless* PH *collapses to the third level.*

3.2 Fixed Parameter Tractability

We now investigate the fixed-parameter tractability of t-TOTAL VERTEX COVER. Two special cases of the problem, for the two extreme values $t = 1$ and $t = k$, have been studied extensively from the perspective of parameterized algorithms. As we saw earlier, these are the k-VERTEX COVER and k-CONNECTED VERTEX COVER problems, respectively. The k-VERTEX COVER problem is perhaps the most well-studied problem in parameterized algorithmics. After a long series of improvements, the current fastest FPT algorithm for this problem runs in time $\mathcal{O}^*\left(1.2738^k\right)$ [8]. Similarly k-CONNECTED VERTEX COVER also has a history of improvements, and the current fastest FPT algorithm for this problem runs in time $\mathcal{O}^*\left(2.7606^k\right)$ [23]. We show in this section that t-TOTAL VERTEX COVER is FPT parameterized by the solution size k, and give an $\mathcal{O}^*\left(c^k\right)$ time algorithm.

Let $G = (V, E)$ be the input graph. If $|V| \leq k$, then we can solve the problem in polynomial time by checking if each component of G has at least t vertices. Also, deleting isolated vertices does not affect the solution. Hence we assume without loss of generality that $|V| > k$, and that G has no isolated vertices. We start with a structural claim which is useful later.

Claim 2. [$\star$][5] Let $G = (V, E)\,, |V| > k$ be a graph in which there are no isolated vertices. Then G has a t-TVC of size at most k if and only if G has a t-TVC of size exactly k.

---

[5] Proofs of results labeled with a [$\star$] have been moved to the appendix due to space constraints.

We will use the following fact to prove the next lemma:

Fact 3. [4, 27] *The number of unlabeled trees on k vertices is at most 2.96^k. Moreover, all non-isomorphic unlabeled trees on k vertices can be enumerated in time $\mathcal{O}(2.96^k k^c)$ for some constant c independent of k.*

Lemma 1. $[\star]$ *All unlabeled* forests *on k vertices can be enumerated in $\mathcal{O}^*\left(2.96^k\right)$ time.*

Now we are ready to prove the main result of this section.

Theorem 2. *For every $t \geq 1$, the t-TOTAL VERTEX COVER problem is in FPT, and can be solved in time $\mathcal{O}^*\left(16.1^{k+\mathcal{O}\left(\log^2 k\right)}\right)$.*

Proof. Observe that any t-TVC, say S, of G is also a vertex cover of G and hence contains a minimal vertex cover $S' \subseteq S$ of G. The idea of our proof is to enumerate all the minimal vertex covers of G of size at most k and then try to expand each one to a t-TVC of G. We will use Fact 3 and the color-coding technique of Alon et al. [1] to do the expansion phase of our algorithm. More precisely, our algorithm is based on the following claim.

Claim 3. $[\star]$ A graph $G = (V, E)$ has a t-TVC of size k if and only if there exists a minimal vertex cover C of G of size at most k, and a subset $T \subseteq V \setminus C$ of size $k - |C|$, such that there exists a forest F on k vertices which is isomorphic to a spanning subgraph of $G[C \cup T]$, and in which each connected component has at least t vertices.

For the algorithm we essentially mimic the claim. First we enumerate all inclusion-minimal vertex covers of G of size at most k. This can be done in time $\mathcal{O}^*\left(2^k\right)$ by a simple 2-way branching on edges—for every edge at least one of its endpoints should be in any vertex cover. For each such vertex cover C, we do the following:

1. Color each $v \in C$ with a distinct color from $\{1, 2, \ldots, |C|\}$.
2. Let $l = k - |C|$. Color the vertices of the independent set $G\left[V \setminus C\right]$ uniformly at random with l new colors.

Let S be a fixed t-TVC of G of size at most k, if there exists one. Define a "good" coloring of V to be a coloring in which the vertices in $S \setminus C$ are all distinctly colored. The above procedure will yield a good coloring of V with probability $l!/l^l \geq e^{-l}$.

Next, we iterate through all unlabeled forests on k vertices, and check if at least one of these forests is isomorphic to a spanning forest $\mathcal{F}$ of $G\left[S\right]$, where each connected component of $\mathcal{F}$ has at least t vertices. By Lemma 1, we can iterate through all such forests in $\mathcal{O}^*\left(2.96^k\right)$ time. To check if a given forest F on k vertices is isomorphic to a witness for $G\left[S\right]$, we do the following:

1. We check if there is at least one tree in F that has less than t vertices. If yes, then we reject F.
2. Next we check if there is a colorful subgraph (one in which each vertex has a distinct color) isomorphic to F in the colored graph obtained above. Since F is of treewidth at most 1, this can be done in $\mathcal{O}\left(2^k \cdot k \cdot n^2\right)$ time [3, Corollary 12]. If such a subgraph is present, then F satisfies the requirements of Claim 3, and so we return YES. Otherwise we reject F.

The expected running time of the algorithm is

$$\begin{aligned} \mathcal{O}^*\left(\sum_{l=0}^{k} 2^{k-l} \times e^l \times 2.96^k \times 2^k\right) &= \mathcal{O}^*\left((2 \times 2.96 \times 2)^k \times \sum_{l=0}^{k}\left(\left(\frac{e}{2}\right)^l\right)\right) \\ &= \mathcal{O}^*\left(16.1^k\right). \end{aligned}$$

To obtain a deterministic algorithm we have to replace the randomized step of the algorithm, that is, where we color the vertices of $G[V \setminus C]$ uniformly at random by l colors, with a deterministic procedure. This is done by making use of (n, l, l)-*perfect hash families*. An (n, l, l)-perfect hash family, $\mathcal{H}$, is a set of functions from $\{1, \dots, n\}$ to $\{1, \dots, l\}$ such that for every subset $S \subseteq \{1, \dots, n\}$ of size l there exists a function $f \in \mathcal{H}$ such that f is injective on S. That is, for all $i, j \in S$, $f(i) \neq f(j)$. There exists a construction of (n, l, l)-perfect hash family of size $\mathcal{O}(e^l \cdot l^{\mathcal{O}(\log l)} \cdot \log n)$ and one can produce this family in time linear in the output size [24]. Using an (n, l, l)-perfect hash family of size $\mathcal{O}(e^l \cdot k^{\mathcal{O}(\log l)} \cdot \log n)$ instead of a random coloring, we obtain the desired deterministic algorithm. The running time of the derandomized algorithm is

$$\mathcal{O}^*\left(16.1^{k+\mathcal{O}\left(\log^2 k\right)}\right).$$

This concludes the proof of the theorem. □

4 Total Edge Covers

For each positive integer t, the parameterized t-TOTAL EDGE COVER problem is defined as follows:

t-TOTAL EDGE COVER (TEC-t)

Input:	An undirected graph $G = (V, E)$, and a positive integer k.
Parameter:	k
Question:	Does there exist a set $S \subseteq E$ of at most k edges of G such that (i) each vertex in V is incident to at least one edge in S (i.e., S is an edge cover of G), and (ii) each connected component of $G(S) = (V, S)$ contains at least t edges of S?

Such an S is called a t-Total Edge Cover (t-TEC) of G. For $t = 1$, t-TOTAL EDGE COVER is the k-EDGE COVER problem, which is solvable in polynomial time [26] while for $t \geq 2$ Fernau and Manlove [11, Theorem 3] showed that the problem is NP-complete. We state the following result which will be useful in obtaining an equivalent formulation of the problem.

Fact 4. [11, Theorem 16] *In any connected graph G with n vertices, and for any $t < n$, there exists a minimal t-TEC, say S, of G such that the graph $G(S)$ induced by the edge set S is acyclic.*

4.1 Kernelizability

Fernau and Manlove [11] observed the following simple vertex kernel of size at most $2k$ for t-TOTAL EDGE COVER: any edge in a graph covers exactly 2 vertices, and a YES instance of the problem has an edge cover of size (number of edges) at most k, and so such an instance cannot have more than $2k$ vertices. In other words, if the input instance has more than $2k$ vertices, then the answer is NO. Otherwise, the input instance itself forms a kernel on at most $2k$ vertices. We can improve this bound on the kernel size for larger values of t by observing the following:

Lemma 2. *Given a graph $G = (V, E)$, the t-*TOTAL EDGE COVER* problem is equivalent to the following problem: does there exist a partition of the vertex set V into q parts $V_1, \dots, V_q$, for some q, such that (i) $G[V_i]$ is connected, (ii) $|V_i| \geq t + 1$ for each $1 \leq i \leq q$, and (iii) $\sum_{i=1}^{q}(|V_i| - 1) \leq k$?*

Proof. Let $(G = (V, E), k)$ be an instance of t-TOTAL EDGE COVER and S be an edge-minimal t-TEC of G. Let $V_1, \ldots, V_q$ be the vertex sets of the connected components of $G(S) = (V, S)$. It directly follows from the properties of S given in the definition of t-TOTAL EDGE COVER and by Fact 4 that $V_1, \ldots, V_q$ satisfy all the conditions in the statement of the lemma. Conversely, if $V_1, \ldots, V_q$ satisfy all the conditions in the statement of the lemma, then let S_i be the edges of a spanning tree of $G[V_i]$, for $1 \leq i \leq q$, and let $S = \bigcup_{i=1}^{q} S_i$. It is easy to see that S has the properties stated in the definition of t-TOTAL EDGE COVER. ☐

This reformulation immediately yields:

Theorem 3. *t-TOTAL EDGE COVER admits a vertex kernel of size $\frac{t+1}{t}k$.*

Proof. Let $(G = (V, E), k)$ be a YES instance of t-TOTAL EDGE COVER. Hence, by Lemma 2 there exists a partition of V into q parts of the kind stated in the lemma. Now $|V_i| \geq t + 1 \implies |V_i| - 1 \geq t \implies \sum_{i=1}^{q} (|V_i| - 1) \geq qt$. But by the lemma, $\sum_{i=1}^{q} (|V_i| - 1) \leq k$, and so $qt \leq k$, and $q \leq \frac{k}{t}$. Also $\sum_{i=1}^{q} (|V_i| - 1) \leq k \implies \sum_{i=1}^{q} |V_i| \leq k + q \leq k + \frac{k}{t} = \frac{t+1}{t}k$, and so G has at most $\frac{t+1}{t}k$ vertices. ☐

Corollary 2. *If the t-TOTAL EDGE COVER problem has an exact exponential time algorithm that runs in $\mathcal{O}^*\left(c^{f(|V|)}\right)$ time on an input instance $(G = (V, E), k)$ for some function $f()$, then the problem has an FPT algorithm that runs in $\mathcal{O}^*\left(c^{f\left(\frac{t+1}{t}k\right)}\right)$ time.*

4.2 Fixed Parameter Tractability

We now present an exact exponential-time algorithm for the problem, with running time $\mathcal{O}^*\left(2^{n+\mathcal{O}(\sqrt{n})}\right)$ where n is the number of vertices in the input graph. By Corollary 2, this yields an FPT algorithm for the problem with running time $\mathcal{O}^*\left(c^k\right)$ for some fixed constant c. This is a significant improvement over the $\mathcal{O}^*\left((2k)^{2k}\right)$ bound obtained by Fernau and Manlove [11].

Let $(G = (V, E), k)$ be an input instance of t-TOTAL EDGE COVER. First, we enumerate all unordered partitions of n. By the Hardy-Ramanujan asymptotic formula, n has at most $2^{\mathcal{O}(\sqrt{n})}$ unordered partitions [16]. The partitions of n can be generated with constant average delay [28], and so we can enumerate all unordered partitions of n in $2^{\mathcal{O}(\sqrt{n})}$ time.

For each partition of n as $n = n_1 + n_2 + \cdots + n_q; 1 \leq q \leq n$, we check if there exists a partition of V into q parts $V_1, \ldots, V_q$ such that (i) $|V_i| = n_i$ for $1 \leq i \leq q$, and (ii) the partition satisfies the conditions of Lemma 2.

To do these checks, we construct the q lists

$$L_i = \{V' \subseteq V \mid |V'| = n_i \text{ and } G\left[V'\right] \text{ is connected}\}$$

for $1 \leq i \leq q$. For $1 \leq i \leq q$ we compute the polynomial

$$P_i = \sum_{V' \in L_i} x^{\chi(V')}$$

where x is a formal variable and $\chi(V')$ is the characteristic vector of $V' \subseteq V$. That is, let $V = \{v_1, v_2, \ldots, v_n\}$. Then $\chi(V')$ is a bit vector with $|V|$ bits where, for $1 \leq j \leq |V|$, the jth bit of $\chi(V')$ is 1 if and only if $v_j \in V'$. The lists L_i and the polynomials P_i can be computed in $\mathcal{O}^*(2^n)$ time, by enumerating all subsets of V. We now compute the product

$$Q = P_1 \times P_2 \times \cdots \times P_q$$

in the given order, with a small modification: given the partial product Q_i of the first i terms, we first compute $Q_i \times P_{i+1}$. Then we delete all those terms αx^β in $Q_i \times P_{i+1}$ where (the binary representation of) β does not contain exactly $\sum_{j=1}^{i+1} n_j$ 1s, and set Q_{i+1} to be the resulting polynomial. This pruning operation ensures that the partial product Q_i, for $1 \leq i \leq q$, represents exactly those sets of size $\sum_{j=1}^{i} n_j$ that can be obtained by taking the union of one set each from $L_1, L_2, \ldots, L_i$. It is easy to see that the product $Q_q = Q$ is non-zero if and only of there exists a partition of V into q parts satisfying the required conditions.

The degree of each polynomial involved in the multiplications is at most $2^{|V|} - 1 = 2^n - 1$, and so, using the Fast Fourier Transform, we can multiply two of these polynomials in $\mathcal{O}\left(2^n \log 2^n\right) = \mathcal{O}\left(n2^n\right)$ time [9, Chapter 30]. We have to perform at most $q \leq n$ such multiplications to compute Q, and so given the P_is we can compute Q in $\mathcal{O}\left(n^2 2^n\right) = \mathcal{O}^*\left(2^n\right)$ time. The running time of this algorithm is thus $2^{\mathcal{O}(\sqrt{n})} \times \left(\mathcal{O}^*\left(2^n\right) + \mathcal{O}^*\left(2^n\right)\right) = \mathcal{O}^*\left(2^{n+\mathcal{O}(\sqrt{n})}\right)$, and so we have:

Theorem 4. *t-TOTAL EDGE COVER can be solved in $\mathcal{O}^*\left(2^{n+O(\sqrt{n})}\right)$ time, where n is the number of vertices in the input graph.*

From this theorem and Corollary 2 we get:

Theorem 5. *t-TOTAL EDGE COVER can be solved in $\mathcal{O}^*(2^{\frac{t+1}{t}k+\mathcal{O}(\sqrt{k})})$ time.*

The above algorithm uses exponential space (e.g., for constructing the lists L_i). An approach similar to the one used in Section 3.2 results in an FPT algorithm that runs in polynomial space:

Theorem 6. *t-TOTAL EDGE COVER can be solved in time $\mathcal{O}^*\left(2^{O(k)}\right)$ time using polynomial space.*

Proof. [Sketch]Enumerate all unordered partitions $(n_1, \ldots n_q)$ such that $n_i \geq t+1$ and $\sum_{i=1}^{q}(n_i - 1) \leq k$ and then enumerate all trees of size n_i in time 2.96^i. Then, for each enumerated q-tuple of trees $(T_1, \ldots, T_q)$, test for subgraph isomorphism in time $\mathcal{O}^*(2^k)$. □

5 Conclusion

We investigated the parameterized complexity of two problems obtained by imposing certain connectivity constraints on two classical problems. In both cases we saw that adding a connectivity constraint (each component of the solution must have at least a certain number of vertices/edges from the solution) causes a drastic change in the computational complexity of the problem. In the case of t-TOTAL EDGE COVER, the shift is from polynomial-time computability to NP-hardness, and had been observed earlier [11]. We showed that a similar shift occurs in the case of the NP-hard problem t-TOTAL VERTEX COVER: for $t = 1$ the problem has a linear vertex kernel [7], and for any $t \geq 2$ the problem has no polynomial-size kernel unless PH collapses. We also showed that both these problems have FPT algorithms that run in time $\mathcal{O}^*\left(c^k\right)$ for different constants c.

One direction of future research would be to examine the effect of such connectivity constraints on other parameterized graph problems. Another would be to try to improve the base c of the exponent of the running times that we obtained for t-TOTAL VERTEX COVER and t-TOTAL EDGE COVER. In particular, the extreme cases of t-TOTAL VERTEX COVER, for the three special values $t = 1$, $t = 2$ and $t = k$, have much smaller values of c, namely $c = 1.2738$, $c = 2.3655$ and $c = 2.7606$, respectively [8, 11, 23]. It will be interesting to see if the value of c for the general case (currently 16.1: see Theorem 2) can be brought closer to these smaller values. Due to the mentioned possible applications of 2-TOTAL EDGE COVER, obtaining better algorithms for this particular problem seems to be most interesting.

References

1. N Alon, R Yuster, and U Zwick. Color-coding. *Journal of the Association for Computing Machinery*, 42(4):844–856, 1995.
2. O. Amini, F. V. Fomin, and S. Saurabh. Implicit branching and parameterized partial cover problems (extended abstract). In Ramesh Hariharan, Madhavan Mukund, and V. Vinay, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2008, December 9-11, 2008, Bangalore, India*, volume 2 of *LIPIcs*, pages 1–12. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2008.
3. O. Amini, F. V. Fomin, and S. Saurabh. Counting subgraphs via homomorphisms. In *Automata, Languages and Programming, 36th International Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part I*, volume 5555 of *Lecture Notes in Computer Science*, pages 71–82. Springer, 2009.
4. T. Beyer and S. M. Hedetniemi. Constant time generation of rooted trees. *SIAM Journal on Computing*, 9(4):706–712, 1980.
5. H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin. On problems without polynomial kernels (extended abstract). In *Proceedings of ICALP 2008*, LNCS, pages 563–574. Springer, 2008.
6. H. L. Bodlaender, S. Thomassé, and A. Yeo. Kernel bounds for disjoint cycles and disjoint paths. In *Proceedings of the 17th Annual European Symposium,on Algorithms (ESA 2009), Copenhagen, Denmark, September 7-9, 2009.*, volume 5757 of *Lecture Notes in Computer Science*, pages 635–646. Springer, 2009.
7. J. Chen, I. A. Kanj, and W. Jia. Vertex Cover: Further observations and further improvements. *Journal of Algorithms*, 41(2):280–301, 2001.
8. J. Chen, I. A. Kanj, and G. Xia. Improved parameterized upper bounds for Vertex Cover. In *Mathematical Foundations of Computer Science MFCS*, volume 4162 of *LNCS*, pages 238–249. Springer, 2006.
9. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, second edition, 2001.
10. M. Dom, D. Lokshtanov, and S. Saurabh. Incompressibility through Colors and IDs. In *Proceedings of ICALP 2009*, volume 5555 of *LNCS*, pages 378–389. Springer, 2009.
11. H. Fernau and D F. Manlove. Vertex and edge covers with clustering properties: Complexity and algorithms. *Journal of Discrete Algorithms*, 7:149–167, 2009.
12. J. Flum and M. Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, 2006.
13. F. V. Fomin, D. Lokshtanov, V. Raman, and S. Saurabh. Subexponential algorithms for partial cover problems. In Ravi Kannan and K Narayan Kumar, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2009)*, volume 4 of *LIPIcs*, pages 193–201. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2009.
14. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP–Completeness*. Freeman, San Francisco, 1979.
15. J. Guo, R. Niedermeier, and S. Wernicke. Parameterized complexity of vertex cover variants. *Theory of Computing Systems*, 41(3):501–520, 2007.
16. G. H Hardy and S Ramanujan. Asymptotic formulae in combinatory analysis. In *Proceedings of the London Mathematical Society*, volume 17, 1918.
17. R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Communications*, pages 85–103, 1972.
18. D. G. Kirkpatrick and P. Hell. On the completeness of a generalized matching problem. In *Proceedings of STOC '78: the 10th Annual ACM Symposium on Theory of Computing*, pages 240–245. ACM, 1978.
19. Joachim Kneis, Alexander Langer, and Peter Rossmanith. Improved upper bounds for partial vertex cover. In Hajo Broersma, Thomas Erlebach, Tom Friedetzky, and Daniël Paulusma, editors, *Graph-Theoretic Concepts in Computer Science, 34th International Workshop, WG 2008, Durham, UK, June 30 - July 2, 2008. Revised Papers*, volume 5344 of *Lecture Notes in Computer Science*, pages 240–251, 2008.
20. Joachim Kneis, Daniel Mölle, Stefan Richter, and Peter Rossmanith. Intuitive algorithms and t-vertex cover. In Tetsuo Asano, editor, *Algorithms and Computation, 17th International Symposium, ISAAC 2006, Kolkata, India, December 18-20, 2006, Proceedings*, volume 4288 of *Lecture Notes in Computer Science*, pages 598–607. Springer, 2006.
21. D. E. Knuth. *The art of computer programming*. Addison-Wesley, third edition, 1998. Vol. 3: Seminumerical algorithms.
22. M. Małafiejski and P. Żylinski. Weakly cooperative guards in grids. In O. Gervasi, M. L. Gavrilova, V. Kumar, A. Laganà, H. P. Lee, Y. Mun, D. Taniar, and C. J. K. Tan, editors, *Computational Science and Its Applications ICCSA (1)*, volume 3480 of *LNCS*, pages 647–656. Springer, 2005.
23. D. Mölle, S. Richter, and P. Rossmanith. Enumerate and Expand: Improved algorithms for Connected Vertex Cover and Tree Cover. *Theory of Computing Systems*, 43(2):234–253, 2008.

24. M. Naor, L. J. Schulman, and A. Srinivasan. Splitters and near-optimal derandomization. In *36th Annual Symposium on Foundations of Computer Science (FOCS'95)*, pages 182–193, Los Alamitos, 1995. IEEE Computer Society Press.
25. R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford Lecture Series in Mathematics and its Applications. Oxford University Press, 2006.
26. R. Z. Norman and M. O. Rabin. An algorithm for a minimum cover of a graph. *Proceedings of the American Mathematical Society*, 10:315–319, 1959.
27. R. Otter. The number of trees. *Annals of Mathematics*, 49(3):583–599, 1948.
28. A. Zoghbi and I. Stojmenovic. Fast algorithms for generating integer partitions. *International Journal of Computer Mathematics*, 70:319–332, 1998.

A Deferred Proofs

In this appendix we state those proofs that were omitted from the main body of the paper due to space constraints.

Claim. Let $G = (V, E)\,, |V| > k$ be a graph in which there are no isolated vertices. Then G has a t-TVC of size at most k if and only if G has a t-TVC of size exactly k.

Proof. If G has a t-TVC, say S, of size exactly k, then S itself is a t-TVC of G of size at most k. For the other direction, let S be a t-TVC of G size $l < k$. Consider any set of $k - l$ vertices $T \subseteq (V \setminus S)$. Since G has no isolated vertex, each $v \in T$ has at least one edge incident on it; since S is a vertex cover of G, the other end of this edge, say w, is in S. Now notice that every connected component of $G[S \cup T]$ has at least t vertices as each connected component of $G\,[S]$ has at least t vertices and every vertex of T gets attached to one of the components of $G\,[S]$. □

Lemma. All unlabeled *forests* on k vertices can be enumerated in $\mathcal{O}^*\left(2.96^k\right)$ time.

Proof. Let F be a forest on k vertices. Add a new vertex v and one edge from v to an arbitrary vertex of each tree in F, to obtain a tree T on $k+1$ vertices. Clearly, the forest F can be obtained by deleting one vertex (namely v) from T. It follows that a graph is a forest on k vertices if and only if it can be obtained by deleting a vertex from some tree on $k+1$ vertices.

To enumerate all forests on k vertices, we first enumerate all trees on $k+1$ vertices. From Fact 3, this can be done in $\mathcal{O}\left(2.96^{k+1}\,(k+1)^c\right)$ time where c is a constant independent of k. For each tree T on $k+1$ vertices obtained in this manner, we delete each of its $k+1$ vertices, one at a time, to obtain a set of forests. By the above observation, this procedure yields every forest on k vertices (some of them perhaps many times). The procedure takes $\mathcal{O}\left((k+1)\,2.96^{k+1}\,(k+1)^c\right) = \mathcal{O}^*\left(2.96^k\right)$ time. □

Claim. A graph $G = (V, E)$ has a t-TVC of size k if and only if there exists a minimal vertex cover C of G of size at most k, and a subset $T \subseteq V \setminus C$ of size $k - |C|$, such that there exists a forest F on k vertices which is isomorphic to a spanning subgraph of $G[C \cup T]$, and in which each connected component has at least t vertices.

Proof. If G has a t-TVC S of size k, then by definition S is a vertex cover of G. Let C be any minimal vertex cover contained in S, and let $T = S \setminus C$. Then $|T| = k - |C|$, and each connected component of $G\,[C \cup T] = G\,[S]$ has, from the definition of a t-TVC, at least t vertices. Let F be a forest formed by picking one spanning tree from each connected component of $G\,[S]$. Then F satisfies the conditions of the claim.

Conversely, let there exist a minimal vertex cover C of G, a set $T \subseteq V \setminus C$ of size $k - |C|$, and a forest F on k vertices that is isomorphic to a spanning subgraph $G' = (S = C \cup T, E')$ of $G[C \cup T]$, and in which each connected component has at least t vertices. Since $C \subseteq S$, S is a vertex cover of G. Also $|S| = |C \cup T| = k$. Now since G' is a subgraph of $G\,[S]$, and each connected component of G' contains at least t vertices, each connected component of $G\,[S]$ has at least t vertices. It follows that S is a t-TVC of G of size k. □