Fernau, H. and Manlove, D. F. (2006) Vertex and Edge Covers With Clustering Properties: Complexity and Algorithms. In: Algorithms and Complexity in Durham 2006: Proceedings of the Second ACiD Workshop, Durham, UK, 18-20 Sep 2006, pp. 69-84. ISBN 9781904987383.

There may be differences between this version and the published version. You are advised to consult the publisher's version if you wish to cite from it.

http://eprints.gla.ac.uk/150584/

Deposited on: 25 October 2017

Enlighten – Research publications by members of the University of Glasgow
http://eprints.gla.ac.uk

Vertex and Edge Covers with Clustering Properties: Complexity and Algorithms

Henning Fernau[1,*] and David F. Manlove[2,†]

[1] *FB 4—Abteilung Informatik, Universität Trier, 54286 Trier, Germany*
Email: `fernau@informatik.uni-trier.de`.

[2] *Department of Computing Science, University of Glasgow, Glasgow G12 8QQ, UK*
Email: `davidm@dcs.gla.ac.uk`.

Abstract

We consider the concepts of a t-*total vertex cover* and a t-*total edge cover* ($t \geq 1$), which generalize the notions of a vertex cover and an edge cover, respectively. A t-total vertex (respectively edge) cover of a connected graph G is a vertex (edge) cover S of G such that each connected component of the subgraph of G induced by S has least t vertices (edges). These definitions are motivated by combining the concepts of clustering and covering in graphs. Moreover they yield a spectrum of parameters that essentially range from a vertex cover to a connected vertex cover (in the vertex case) and from an edge cover to a spanning tree (in the edge case). For various values of t, we present $\mathcal{NP}$-completeness and approximability results (both upper and lower bounds) and $\mathcal{FPT}$ algorithms for problems concerned with finding the minimum size of a t-total vertex cover, t-total edge cover and connected vertex cover, in particular improving on a previous $\mathcal{FPT}$ algorithm for the latter problem.

1 Introduction

In graph theory, the notion of covering vertices or edges of graphs by other vertices or edges has been extensively studied (see [24] for a survey). For instance, covering vertices by other vertices leads to parameters concerned with vertex domination [20, 21]. When edges are to be covered by vertices we obtain parameters connected with the classical vertex covering problem [19, p.94]. Covering vertices by edges, i.e. finding edge covers, was first considered by Norman and Rabin [30]. Finally, when edges are to cover other edges, we obtain parameters associated with edge domination (introduced by Mitchell and Hedetniemi [25]). These problems have long been a testbed for the design of parameterized algorithms (or for showing

*Part of this work was carried out whilst visiting the University of Glasgow, supported by Engineering and Physical Sciences Research Council grant EP/D030110/1.

†Supported by Engineering and Physical Sciences Research Council grant GR/R84597/01 and by a Royal Society of Edinburgh / Scottish Executive Personal Research Fellowship. Email

the limitations of that approach) [9]. However, in particular only recently has a systematic study of variants of vertex cover problems been initiated with respect to parameterized complexity [18, 29].

Clustering in graphs is another fundamental concept with a large range of practical applications [14]. Connectedness can be seen as one of the weakest notions of clustering: it is reasonable to assert that a vertex set can be termed a cluster only if it is connected. When being used for classification purposes, there is rarely only one cluster, but rather a number of them, each representing some concept, i.e., one is looking for connected components. In order to exclude trivial cases and to define meaningful concepts, it may often be appropriate to impose a lower bound on the number of elements per cluster.

In this paper we consider a synergy of the notion of clustering with each of the concepts of vertex and edge covering. Throughout we assume that $G = (V, E)$ is a connected graph, where $n = |V|$ and $m = |E| \geq 1$. For $1 \leq t \leq n$, a *t-total vertex cover* (henceforth a *t-tvc*) in G is a vertex cover S in G such that each connected component of $G[S]$, the subgraph of G induced by S, has at least t vertices. Similarly, for $1 \leq t \leq m$, a *t-total edge cover* (henceforth a *t-tec*) in G is an edge cover S of G (i.e. each vertex of G is incident to an edge in S) such that each connected component of $G[S]$, the subgraph of G induced by S, has at least t edges. Hence, if S is a t-tvc or t-tec, then S is a vertex cover or edge cover respectively such that each member of S belongs to a "cluster" containing at least t elements of S.

The concept of a *total dominating set* in a graph, first defined and studied by Cockayne et al. [6], illustrates one case where the notions of clustering and covering (vertices by vertices) have already been brought together. A set of vertices S is a *total dominating set* of G if (i) S is a dominating set (i.e. every vertex in $V \backslash S$ is adjacent to a vertex in S), and (ii) each connected component of $G[S]$ has at least two vertices.

The notion of a 2-tvc was first defined by Jean Blair [2] using the terminology *total vertex cover* (by analogy to the term *total dominating set*). It is straightforward to present relationships between the minimum size of a t-tvc (respectively t-tec) for various values of t and established parameters concerned with vertex covering (respectively edge covering) in G. Throughout this paper, our notation follows and extends that of Harary [19]. Let $\alpha_0(G)$ denote the minimum size of a vertex cover in G. A *connected vertex cover* (henceforth a *cvc*) in G is a vertex cover S in G such that $G[S]$ is connected. Let $\alpha_0^c(G)$ denote the minimum size of a cvc in G. It follows that a 1-tvc is simply a vertex cover (recall that $m \geq 1$), whilst a cvc of size t is a t-tvc. For $t \geq 1$, let $\alpha_{0,t}(G)$ denote the minimum size of a t-tvc in G. Then $\alpha_{0,1}(G) = \alpha_0(G)$. The parameters $\alpha_{0,t}(G)$ for $t \geq 2$ do not appear to have been studied in the literature previously. In Section 2, we present some additional relationships involving the parameters $\alpha_0(G)$, $\alpha_{0,t}(G)$ and $\alpha_0^c(G)$.

Now let $1 \leq t \leq m$ – we turn to the concept of a t-tec. It follows that a 1-tec is simply an edge cover (again recall that $m \geq 1$), whilst a minimum $(n-1)$-tec is a minimum connected edge cover, i.e. a spanning tree. Let $\alpha_{1,t}(G)$ denote the minimum size of a t-tec of G, and let $\alpha_1(G)$ denote the minimum size of an edge cover of G. Then $\alpha_{1,1}(G) = \alpha_1(G)$. The parameters $\alpha_{1,t}(G)$ for $t \geq 2$ do not appear to have been studied in the literature previously. In Section 2, we present some additional relationships between the parameters $\alpha_1(G)$ and $\alpha_{1,t}(G)$.

We remark that, for a t-tvc (respectively t-tec) to exist, it is sufficient that each connected component of G has at least t vertices (edges), and the results in this paper also hold in such a setting. However for ease of exposition, and due to the correspondence between t-tvcs and t-tecs with connected vertex covers and spanning trees respectively, we choose to assert throughout that G is connected.

Given $t \geq 1$, let VC, t-TVC, t-TEC and CVC denote the problems of computing $\alpha_0(G)$, $\alpha_{0,t}(G)$, $\alpha_{1,t}(G)$ and $\alpha_0^c(G)$ respectively, given a connected graph G where $n = |V|$ and $m = |E| \geq 1$ (additionally $n \geq t$ in the case of t-TVC and $m \geq t$ in the case of t-TEC). Let VC-D, t-TVC-D, t-TEC-D and CVC-D denote the decision versions of VC, t-TVC, t-TEC and CVC, respectively. Hence, the question is, given a graph G and a parameter k, whether there is a cover C (with the additional properties specified by the problem) such that $|C| \leq k$.

For each $t \geq 2$, we show in Section 3 that t-TVC is $\mathcal{NP}$-hard and not approximable within an asymptotic performance ratio of $10\sqrt{5} - 21 - \delta$ (> 1.3606), for any $\delta > 0$, unless $\mathcal{P}$=$\mathcal{NP}$. However on the other hand we prove that t-TVC is approximable within 2. We also prove that t-TVC-D is $\mathcal{NP}$-complete, even for planar bipartite graphs of maximum degree 3. Moreover we show that there exists a constant $\delta_t > 1$ such that t-TVC in bipartite graphs of maximum degree 3 is not approximable within δ_t unless $\mathcal{P} = \mathcal{NP}$. Finally, we give a parameterized algorithm for 2-TVC-D with complexity $\mathcal{O}^*(2.3655^k)$. Here the parameter is the size of the 2-tvc.

CVC is $\mathcal{NP}$-hard, even for planar graphs of maximum degree 4 [16], though polynomial-time solvable for graphs of maximum degree 3 [34]. For a tree T, finding a minimum cvc is trivial (if $T = K_2$, one vertex will suffice, otherwise the set of non-leaf nodes is a minimum cvc in T). It is known that CVC is approximable within 2 [33, 1]. In Section 4, we show that CVC is not approximable within an asymptotic performance ratio of $10\sqrt{5} - 21 - \delta$, for any $\delta > 0$, unless $\mathcal{P}$=$\mathcal{NP}$. The complexity of CVC in bipartite graphs does not seem to have been considered in the literature so far. We show that CVC-D is $\mathcal{NP}$-complete, even for planar bipartite graphs of maximum degree 4. We also present a parameterized algorithm for CVC-D with complexity $\mathcal{O}^*(2.9316^k)$, improving on a previous algorithm due to Guo et al. [18], having complexity $\mathcal{O}^*(6^k)$. Here the parameter is the size of the cvc. We remark that, independently and by using different techniques, Moelle et al. [26] present a parameterized algorithm for CVC-D with complexity $\mathcal{O}^*(3.2361^k)$. Furthermore, following that approach, an improved algorithm for the same problem, having complexity $\mathcal{O}^*(2.7606^k)$, will be reported [27].

1-TEC, i.e. the problem of finding a minimum edge cover, is polynomial-time solvable [30]. In Section 5, we give a Gallai identity involving $\alpha_{1,t}(G)$ for each $t \geq 1$. We use this to prove that t-TEC-D is $\mathcal{NP}$-complete for each $t \geq 2$. We also show that t-TEC is approximable within 2 for each $t \geq 2$, though there exists some $\delta > 1$ such that 2-TEC is not approximable within δ unless $\mathcal{P}$=$\mathcal{NP}$. Finally we show that t-TEC-D is in $\mathcal{FPT}$ for each $t \geq 2$ (where the parameter is the size of the t-tec) and the parametric dual of 2-TEC-D is also in $\mathcal{FPT}$. This gives one of the few examples where both a problem and its dual belong to $\mathcal{FPT}$.

2 Preliminary observations involving $\alpha_{0,t}(G)$ and $\alpha_{1,t}(G)$

We begin this section by presenting some relationships involving the parameters $\alpha_0(G)$, $\alpha_{0,t}(G)$ and $\alpha_0^c(G)$.

Proposition 1. *Let $G = (V, E)$ be a connected graph where $n = |V|$, $m = |E| \geq 1$, and let $1 \leq t \leq n$. Then:*

1. *$\alpha_0(G) \leq \alpha_{0,t}(G)$, and for $t < n$, $\alpha_{0,t}(G) \leq \alpha_{0,t+1}(G)$;*
2. *$\alpha_{0,t}(G) \geq t$;*
3. *for $\alpha_0^c(G)/2 < t \leq \alpha_0^c(G)$, $\alpha_{0,t}(G) = \alpha_0^c(G)$;*
4. *for $t \geq \alpha_0^c(G)$, $\alpha_{0,t}(G) = t$;*
5. *the minimum t such that $\alpha_{0,t}(G) = t$ satisfies $t = \alpha_0^c(G)$.*

Proof. 1. If S is a $(t+1)$-tvc then clearly S is a t-tvc. Moreover clearly any t-tvc is a vertex cover.

2. If S is any t-tvc, then as $m \geq 1$, it follows that $G[S]$ has at least one connected component, which contains at least t vertices.

3. Let S be a minimum t-tvc and let C be a minimum cvc. Then C is a t-tvc, so that $|S| \leq |C| = \alpha_0^c(G)$. Now suppose that $G[S]$ contains at least two connected components. Then $|S| \geq 2t > \alpha_0^c(G)$, a contradiction. Hence S is a cvc, so that $|C| \leq |S|$. Hence $\alpha_{0,t}(G) = \alpha_0^c(G)$.

4. Let C be a minimum cvc and let $t' = t - |C|$. As G is connected we may construct a t-tvc S by adding t' vertices to C. Then $|S| = t$, so that $\alpha_{0,t}(G) \leq t$. Hence $\alpha_{0,t}(G) = t$ by Part 2.

5. Let $t = \alpha_0^c(G)$. By Part 4, $\alpha_{0,t}(G) = t$. Now suppose that $t' < t$ and $\alpha_{0,t'}(G) = t'$. Let S be a t'-tvc such that $|S| = t'$. Then $G[S]$ contains one connected component, for otherwise $|S| \geq 2t'$, a contradiction. Hence S is a cvc such that $|S| = t' < \alpha_0^c(G)$, a contradiction. □

We next present some relationships involving the parameters $\alpha_1(G)$ and $\alpha_{1,t}(G)$.

Proposition 2. *Let $G = (V, E)$ be a connected graph where $n = |V|$, $m = |E| \geq 1$, and let $1 \leq t \leq m$. Then:*

1. *$\alpha_1(G) \leq \alpha_{1,t}(G)$, and for $t < m - 1$, $\alpha_{1,t}(G) \leq \alpha_{1,t+1}(G)$;*
2. *$\alpha_{1,t}(G) \geq t$;*
3. *for $\frac{n-1}{2} < t \leq n - 1$, $\alpha_{1,t}(G) = n - 1$;*
4. *for $t \geq n - 1$, $\alpha_{1,t}(G) = t$.*
5. *the minimum t such that $\alpha_{1,t}(G) = t$ satisfies $t = n - 1$.*

Proof. 1. If S is a $(t+1)$-tec then clearly S is a t-tec. Moreover clearly any t-tec is an edge cover.

2. If S is any t-tec, then as $m \geq 1$, it follows that $G[S]$ has at least one connected component, which contains at least t edges.

3. Let S be a minimum t-tec and let T be a spanning tree of G. Then T is a t-tec, so that $|S| \leq |T| = n-1$. Now suppose that $G[S]$ contains at least two connected components. Then $|S| \geq 2t > n-1$, a contradiction. Hence $G[S]$ is connected, so that $|S| \geq n-1$. Thus $\alpha_{1,t}(G) = n-1$.

4. Let T be a spanning tree of G and let $t' = t-(n-1)$. As G is connected we may construct a t-tec S by adding t' edges to T. Then $|S| = t$, so that $\alpha_{1,t}(G) \leq t$. Hence $\alpha_{1,t}(G) = t$ by Part 2.

5. Let $t = n-1$. By Part 4, $\alpha_{1,t}(G) = t$. Now suppose that $t' < t$ and $\alpha_{1,t'}(G) = t'$. Let S be a t'-tec such that $|S| = t'$. Then $G[S]$ contains one connected component, for otherwise $|S| \geq 2t'$, a contradiction. Hence S is a spanning tree such that $|S| = t' < n-1$, a contradiction. ☐

3 Complexity and approximability of t-TVC

We begin with a lower bound for the approximability of t-TVC in general graphs.

Theorem 3. *For each $t \geq 1$, t-TVC is $\mathcal{NP}$-hard and not approximable within an asymptotic performance ratio of $10\sqrt{5}-21-\delta$, for any $\delta > 0$, unless $\mathcal{P}$=$\mathcal{NP}$.*

Proof. For $t = 1$ the result follows by [8]. Now assume that $t \geq 2$. Let $G = (V, E)$ be an instance of VC. We lose no generality in assuming that G is connected and $|V| \geq 2$. Create a new graph $G' = (V', E')$ such that $V' = V \cup W$ and $E' = E \cup E_1 \cup E_2$, where $W = \{w_i : 1 \leq i \leq t\}$ is a set of new vertices, $E_1 = \{\{v, w_1\} : v \in V\}$ and $E_2 = \{\{w_i, w_{i+1}\} : 1 \leq i \leq t-1\}$. Let $W' = W \backslash \{v_t\}$. It is straightforward to verify that if S is a minimum vertex cover in G, then $S \cup W'$ is a t-tvc in G'. Conversely if S' is a minimum t-tvc in G', then $S' \cap W = W'$, and $S' \cap V$ is a vertex cover in G. Hence $\alpha_{0,t}(G') = \alpha_0(G) + t - 1$. The result follows by [8]. ☐

We now present an upper bound for the approximability of t-TVC.

Theorem 4. *For each $t \geq 1$, t-TVC is approximable within 2.*

Proof. Let $G = (V, E)$ be an instance of t-TVC (then G is a connected graph, where $n = |V| \geq t$ and $m = |E| \geq 1$). Savage [33] presents an approximation algorithm for CVC: the algorithm computes a cvc S in G such that $|S| \leq 2\alpha_0(G)$. Suppose firstly that $t \leq |S|$. Then S is a t-tvc, and $|S| \leq 2\alpha_0(G) \leq 2\alpha_{0,t}(G)$ by Proposition 1, as required. Now suppose that $t > |S|$. Let $t' = t - |S|$. As G is connected, we may construct a t-tvc S' in G by adding t' vertices to S. Then $|S'| = t$, so that S' is in fact a minimum t-tvc by Proposition 1. ☐

The next two results concern the complexity and approximability of t-TVC in bounded degree bipartite graphs, for each $t \geq 2$.

Theorem 5. *For each $t \geq 2$, t-TVC-D is $\mathcal{NP}$-complete for planar bipartite graphs of maximum degree 3.*

Proof. Clearly t-TVC-D belongs to $\mathcal{NP}$. To show $\mathcal{NP}$-hardness, we give a reduction from the $\mathcal{NP}$-complete restriction of VC-D to planar graphs of maximum degree 3 [17, 16]. Hence let $G = (V, E)$ (a planar graph of maximum degree 3) and k (a positive integer) be an instance of this problem. Let $E = \{e_1, e_2, \ldots, e_m\}$ for some m. We define an instance of t-TVC-D as follows. Construct a graph $G' = (V', E')$ by letting $V' = V \cup W$, where $W = \{w_{i,j} : 1 \leq i \leq m \wedge 1 \leq j \leq t\}$. For each i ($1 \leq i \leq m$), suppose that $e_i = \{u, v\}$ for some $u, v \in V$. Add the edges $\{u, w_{i,1}\}$, $\{w_{i,j}, w_{i,j+1}\}$ ($1 \leq j \leq t-1$) and $\{w_{i,1}, v\}$ to E'. Clearly G' can be constructed in polynomial time from G, and G' is planar, bipartite and has maximum degree 3. Let $k' = k + (t-1)m$. We claim that G has a vertex cover of size at most k if and only if G' has a t-tvc of size at most k'.

For, suppose that G has a vertex cover S of size at most k. Let $S' = S \cup W'$, where $W' = W \backslash \{w_{i,t} : 1 \leq i \leq m\}$. Then it may be verified that S' is a t-tvc of G', and $|S'| = |S| + (t-1)m \leq k + (t-1)m = k'$.

Conversely suppose that G' has a t-tvc of size at most k'. Choose S' to be such a set that minimizes $|S' \cap W|$. It is straightforward to verify that $W' \subseteq S'$, since $t \geq 2$. Also, $S' \cap W = W'$. For, suppose that $w_{i,t} \in S'$ for some i ($1 \leq i \leq m$). Let $e_i = \{u, v\}$ for some $u, v \in V$. Define $S'' = (S' \backslash \{w_{i,t}\}) \cup \{u\}$. Then S'' is a t-tvc of G', $|S''| \leq |S'| \leq k'$, and $|S'' \cap W| < |S' \cap W|$, contradicting the choice of S'. Hence the claim is established. Let $S = S' \cap V$. Then it may be verified that S is a vertex cover of G, and $|S| = |S'| - (t-1)m \leq k' - (t-1)m = k$. □

Corollary 6. *For each $t \geq 2$, t-TVC in bipartite graphs of maximum degree 3 is not approximable within $1 + \frac{1}{500t-400}$ unless $\mathcal{P}$=$\mathcal{NP}$.*

Proof. VC in cubic graphs is not approximable within $\frac{100}{99}$ unless $\mathcal{P}$=$\mathcal{NP}$ [5]. By considering this problem as the starting point for the same reduction as in the proof of Theorem 5, it again follows that $\alpha_{0,t}(G') = \alpha_0(G) + (t-1)m$. Now $\alpha_0(G) \geq \beta_1(G) \geq \frac{m}{5}$ [35, Theorem 60], where $\beta_1(G)$ is the size of a maximum matching in G, since G is cubic. It follows that $\alpha_{0,t}(G') \leq (5t-4)\alpha_0(G)$. Hence the reduction of Theorem 5 is an L-reduction (defined in [31]) with parameters $\alpha = 5t - 4$ and $\beta = 1$. The result follows by [35, Theorem 63]. □

The next result concerns the parameterized complexity of 2-TVC.

Theorem 7. *2-TVC-D is in $\mathcal{FPT}$ and can be solved in time $\mathcal{O}^*(3.2361^k)$, where k is the size of the 2-tvc.*[1]

Proof. Let $G = (V, E)$ be a connected graph. Firstly, we describe an algorithm running in time $\mathcal{O}^*(4^k)$. It is known (see [9, 7]) that all minimal vertex covers of size at most k can be enumerated in time $\mathcal{O}^*(2^k)$. Of course, a (minimal) vertex cover need not be a 2-tvc, but all 2-tvcs of size k can be obtained by "extending" minimal vertex covers of size at most k.

For each minimal vertex cover C of G described at a leaf of the search tree, we construct a hypergraph H as follows: $V' = V \setminus C$ are the vertices of the hypergraph, and the hyperedges E' are the open neighbourhoods of the vertices in C that do not contain (other) vertices from C. Now, a "minimum extension" of C to a valid 2-tvc corresponds to a minimum hitting set in H. Since $|E'| \leq k$, this can be done in time

[1] In [11], we show how this upper bound can be further improved to $\mathcal{O}^*(2.3655^k)$.

$\mathcal{O}^*(2^k)$ according to [12]; see also [9, Theorem 8.1]. This shows that 2-TVC-D can be solved in time $\mathcal{O}^*(4^k)$.

To improve on this running time, observe that the degree-0 and degree-1 reduction rules of VC-D (c.f. [9, p.21]) are also valid for 2-TVC-D with some variation, since we should now deal with instances in which some vertices are already marked. More precisely, we say that a vertex is 1-marked if it is known to belong to the vertex cover, and it is 2-marked if, in addition, also one of its neighbours is known to belong to the vertex cover. Moreover, a vertex is 0-marked if it is unknown if it belongs to the vertex cover, but one of its neighbours does belong to the vertex cover. Notice that the particular neighbour that testifies why a vertex has become say 0-marked may be later deleted since the necessary information is kept in the markings. To be coherent with the search tree part, we manage the parameter budget in a way that the 1- and 2-marked vertices are already taken into account; hence we have a NO-instance (corresponding to this branch of the search tree) if the parameter budget falls below zero.

We use the following *colouring handling rules*:

- If vertex x is unmarked but neighbour of a 1-marked or 2-marked vertex, then 0-mark x.
- If vertex x is 1-marked and neighbour of a 1-marked or 2-marked vertex, then 2-mark x.
- Merge two 2-marked vertices.
- If the parameter drops below zero, we have a NO-instance.

These rules guarantee that the 1-marked and 2-marked vertices form an independent set in the graph, and that there is at most one 2-marked vertex in a reduced instance.

In the following, we always assume that we have already exhaustively executed the colouring handling rules.

Let us first consider the case that x is a vertex of degree 0.

- If x is unmarked or 0-marked, then delete x.
- If x is 1-marked, then we have a NO-instance.
- If x is 2-marked, then delete x.

If x is a vertex of degree 1 with unique neighbour y, then do the following:

- If x is unmarked or 0-marked, we distinguish two subcases:
  - If y has degree 1, decrease the parameter by 2 if x and y are both unmarked, and decrease the parameter by 1 if x or y is 0-marked. In both cases, delete both x and y.
  - Otherwise, y has degree at least 2. If y is unmarked, 1-mark y. If y is 0-marked, 2-mark y. In both cases delete x and decrement the parameter. (For if C is a minimum 2-tvc that includes x, then $(C \setminus \{x\}) \cup \{z\}$ is a minimum 2-tvc excluding x, where $z \neq x$ is adjacent to y.)
- If x is 1-marked, then 2-mark y and decrement the parameter.

- Finally, if x is 2-marked, then delete x.

If the vertices that are unmarked or 0-marked form an independent set, a vertex cover has been found; without any 1-marked vertices, we have a 2-tvc.

When we branch in the first phase of the algorithm (that basically enumerates the minimal vertex covers), the branching itself must also respect the markings of the vertices. Of course, only 0-marked or unmarked vertices will be considered for branching. If x is an unmarked vertex chosen for branching, then in the case that x is put into the cover, it will show up as a 1-marked vertex in the recursion. If x is 0-marked, it will show up as a 2-marked vertex in the recursion. In the case that x is not taken into the cover, it will be deleted from the instance (as usual), irrespective of whether x is 0-marked or unmarked. At the beginning of each recursive call, all reduction rules will be exhaustively applied.

We arrange our search tree so that it always tries to branch at an unmarked vertex if such a vertex x exists. By the reduction rules, x has degree at least 2, and none of its neighbours is 1-marked or 2-marked. Hence, we either take x into the cover (reducing the parameter by 1) or all of its neighbours go into the cover (reducing the parameter by at least 2). Therefore, the running time of this part of the enumeration can be estimated by the recurrence relation $T(k) \leq T(k-1)+T(k-2)$. The partial search tree T' based on branching at unmarked vertices therefore has $\mathcal{O}^*(1.618^k)$ nodes.

After having branched at all unmarked vertices, we next consider an edge whose endpoints x, y are both 0-marked. Since it might well be that all other neighbours of x and y are 1-marked or 2-marked, we cannot guarantee to reduce the parameter by more than 1 in the branching that takes either x or all of its neighbours into the cover. Hence a leaf node of T' at parameter height i will be replaced by a tree of height $\mathcal{O}^*(2^{k-i})$ in the complete search tree T.

We finally consider the Hitting Set phase of the algorithm. Any leaf node x of T (corresponding to a vertex cover but not necessarily a 2-tvc) has an ancestor y that is a leaf node of T' at parameter height i. Recall that branching on a vertex that is already 0-marked produces a 2-marked vertex. Such vertices do not need to be taken into account in the Hitting Set instance. That is, the Hitting Set instance corresponding to node x involves only the i vertices that are 1-marked, created at y and above in T'. Since at most i hyperedges will be created, the Hitting Set phase at node x runs in $\mathcal{O}^*(2^i)$ time. Hence if T' has l_i leaf nodes at parameter height i, the overall complexity of our algorithm can be estimated as $\mathcal{O}^*(\sum_{i=0}^{k}(l_i 2^{k-i}2^i)) = \mathcal{O}^*(2^k \sum_{i=0}^{k} l_i)) = \mathcal{O}^*(3.2361^k)$ as claimed. □

4 Complexity and approximability of CVC

We begin with two results concerning the complexity and approximability of CVC in general graphs and planar bipartite graphs of bounded degree.

Theorem 8. CVC *is not approximable within an asymptotic performance ratio of* $10\sqrt{5} - 21 - \delta$, *for any* $\delta > 0$, *unless* $\mathcal{P}$=$\mathcal{NP}$.

Proof. The result follows using the construction in the proof of Theorem 3 for the case that $t = 2$. □

Theorem 9. CVC-D *is $\mathcal{NP}$-complete for planar bipartite graphs of maximum degree 4.*

Proof. Clearly CVC-D belongs to $\mathcal{NP}$. To show $\mathcal{NP}$-hardness, we use the same reduction as in Theorem 5 with $t = 2$, however in this case we reduce from the $\mathcal{NP}$-complete restriction of CVC-D to planar graphs of maximum degree 4 [16]. The graph G' so constructed is then also a planar bipartite graph of maximum degree 4. If S is a connected vertex cover of size at most k in G, then $S \cup W'$ is a connected vertex cover of size at most k' in G'. Conversely if S' is a minimum connected vertex cover of size at most k' in G', then $S \cap W = W'$. It follows that $S' \cap V$ is a connected vertex cover in G of size at most k. □

We now consider the parameterized complexity of CVC. Combining the recently improved Steiner tree algorithm [13, 28] with a colouring technique similar to that described in detail for 2-TVC-D in the proof of Theorem 7, we are able to achieve the following result (see [11] for the full proof):

Theorem 10. CVC-D *is in $\mathcal{FPT}$ and can be solved in time $\mathcal{O}^*(2.9316^k)$, where k is the size of the cvc.*

5 Complexity and approximability of t-TEC

Let $G = (V, E)$ be a connected graph, where $n = |V|$, $m = |E| \geq 1$, and let $1 \leq t \leq n - 1$. We begin this section by presenting a Gallai identity involving the concepts of a t-tec and a *t-tree packing*. A *t-tree packing* of G is a collection $\mathcal{P} = \{G_1, \dots, G_k\}$ of vertex-disjoint (non-induced) subgraphs of G, each of which is a tree containing exactly t edges. The value k is defined to be the *size* of $\mathcal{P}$. Let $\beta_{1,t}(G)$ denote the maximum size of a t-tree packing of G. Then $\beta_{1,1}(G) = \beta_1(G)$, the size of a maximum matching in G. The following result gives a Gallai identity involving $\alpha_{1,t}(G)$ and $\beta_{1,t}(G)$.

Theorem 11. *Let $G = (V, E)$ be a connected graph, where $n = |V|$, $m = |E| \geq 1$, and let $1 \leq t \leq n - 1$. Then $\alpha_{1,t}(G) + \beta_{1,t}(G) = n$.*

Proof. Let $\mathcal{P} = \{G_1, \dots, G_k\}$ be a t-tree packing of G such that $k = \beta_{1,t}(G)$. Let S initially contain the edges belonging to the subgraphs in $\mathcal{P}$. Then $|S| = kt$ and S covers $k(t+1)$ vertices of G, so that $n - k(t+1)$ vertices are as yet uncovered. Pick any uncovered vertex v. Then v is at distance at most t from a covered vertex w, for otherwise we contradict the maximality of $\mathcal{P}$. Let $v_0 = v$, and let $v_0, v_1, \dots, v_s$ be the vertices (in order) on a path in G from v_0 to v_s, where v_s is covered, v_i is uncovered $(1 \leq i \leq s-1)$, and $1 \leq s \leq t$. Add $\{v_i, v_{i+1}\}$ to S $(0 \leq i \leq s-1)$. Continue in this way until all vertices are covered. Then S is a t-tec of G. Moreover we add one edge for every additional vertex that we cover, so that $|S| = kt + (n - k(t+1)) = n - k$, i.e. $\alpha_{1,t}(G) \leq n - \beta_{1,t}(G)$.

Conversely let $\mathcal{S} = \{S : S \text{ is a } t\text{-tec in } G \text{ and } |S| = \alpha_{1,t}(G)\}$. Choose $S \in \mathcal{S}$ such that $G[S]$ contains the fewest number of cycles. Let $G_i = (V_i, S_i)$ $(1 \leq i \leq k)$ be the connected components of $G[S]$, for some $k \geq 1$. Let i $(1 \leq i \leq k)$ be given. Then by definition of S, it follows that G_i contains at least t edges. Now suppose that G_i contains a cycle, and let e be any edge on this cycle. If $k = 1$ then $S' = S \backslash \{e\}$ is a connected subgraph of G that spans V, and hence $|S'| \geq n-1$, so that S' is a t-tec, contradicting the minimality of S. Hence $k \geq 2$. Since S is an edge cover, there exists an edge $e' = \{u, v\}$ in G such that u is covered by G_i and v is covered by some G_j $(1 \leq j \neq i \leq k)$. Let $S' = (S \backslash \{e\}) \cup \{e'\}$. Then S' is a t-tec, $|S'| = |S|$ and S' has one fewer cycle than S, contradicting the choice of S. Hence G_i is acyclic. It follows that $|S_i| = |V_i| - 1$, so that

$$|S| = \sum_{i=1}^{k} |S_i| = \sum_{i=1}^{k} (|V_i| - 1) = n - k.$$

Let $\mathcal{P} = \{H_1, \ldots, H_k\}$ be formed by "pruning" each G_i in order to form a tree H_i containing exactly t edges (this may be carried out by repeatedly deleting edges incident to vertices of degree 1 in G_i, until exactly t edges remain). Then $\mathcal{P}$ is a t-tree packing of G, and $|\mathcal{P}| = k = n - \alpha_{1,t}(G)$, so that $\beta_{1,t}(G) \geq n - \alpha_{1,t}(G)$. □

We remark that, in the case $t = 1$, Theorem 11 gives the familiar Gallai identity $\alpha_1(G) + \beta_1(G) = n$ [15].

For each $t \geq 1$, let t-TREE PACKING denote the problem of computing $\beta_{1,t}(G)$, given a connected graph $G = (V, E)$, where $n = |V| \geq t+1$. Let t-TREE PACKING-D denote the decision version of t-TREE PACKING. Kirkpatrick and Hell [23] proved the following result concerning t-TREE PACKING-D.

Theorem 12 ([23]). *For each $t \geq 2$, t-*TREE PACKING-D *is $\mathcal{NP}$-complete.*

The following is an immediate consequence of Theorems 12 and 11.

Corollary 13. *For each $t \geq 2$, t-*TEC-D *is $\mathcal{NP}$-complete.*

The next two results concern the approximability of t-TEC for $t \geq 2$.

Theorem 14. *For each $t \geq 2$, t-*TEC *is approximable within 2.*

Proof. Let $G = (V, E)$ be an instance of t-TEC (a connected graph where $n = |V|$ and $m = |E| \geq t$). Any edge cover S of G satisfies $|S| \geq \frac{n}{2}$, since each edge of S covers 2 vertices of G. Now let T be a spanning tree of G. Suppose firstly that $t \leq n-1$. Then T is a t-tec of G and $|T| = n - 1 \leq 2\alpha_1(G) \leq 2\alpha_{1,t}(G)$ by Proposition 2, as required. Now suppose that $t > n-1$. Let $t' = t - (n-1)$. As G is connected, we may construct a t-tec S by adding t' edges to T. Then $|S| = t$, so that S is in fact a minimum t-tec by Proposition 2. □

Theorem 15. 2-TEC *in bounded degree graphs is not approximable within some $\delta > 1$ unless $\mathcal{P}$=$\mathcal{NP}$.*

Proof. 2-TREE PACKING in graphs of maximum degree B is not approximable within some $\varepsilon > 1$ unless $\mathcal{P}$=$\mathcal{NP}$ [22]. We may consider this problem as the starting point for a reduction to 2-TEC that essentially follows the same lines as the proof of Theorem 11 in the case that $t = 2$ and $G = (V, E)$ is a connected graph of maximum degree B, where $n = |V|$ and $m = |E|$. Now $\beta_{1,2}(G) \geq m/(3B-1)$, since any P_3 in a 2-tree packing $\mathcal{P}$ of G rules out at most $3B - 1$ edges for inclusion in some other P_3 in $\mathcal{P}$. By Theorem 11, $\alpha_{1,2}(G) + \beta_{1,2}(G) = n \leq m$, since the graph constructed by Kann's reduction [22] is not acyclic. Hence the reduction described here is an L-reduction (see [31]) with parameters $\alpha = 3B - 2$ and $\beta = 1$. The result follows by [35, Theorem 63]. □

We now consider the parameterized complexity of t-TEC ($t \geq 2$).

Theorem 16. *For each $t \geq 2$, t-TEC-D is in $\mathcal{FPT}$.*

Proof. Let $\langle G, k \rangle$ be an instance of t-TEC-D. Then k is a parameter and $G = (V, E)$ is a connected graph where $n = |V|$ and $m = |E| \geq t$. As observed in the proof of Theorem 14, $k \geq \frac{n}{2}$ or else $\langle G, k \rangle$ is a NO-instance. Hence $n \leq 2k$, so $m \leq (2k)^2$. Generating every subset S of E with at most k edges and verifying whether S is a t-tec is a process that takes $\mathcal{O}^*((2k)^{2k})$ overall time. □

We now consider the concept of *parametric duality* (see [4, 9] for a recent exposition), which is in a sense quite related to the family of Gallai identities proved above. Define DUAL-t-TEC-D to be the problem of deciding, given a connected graph $G = (V, E)$ where $n = |V|$ and $m = |E| \geq t$, and a (dual) parameter k_d, whether there a t-tec of size at most $n - k_d$. Using the fact that 2-TREE-PACKING-D is in $\mathcal{FPT}$ and solvable in time $\mathcal{O}^*(2^{5.3k})$, where k is the size of the 2-tree-packing [32], Theorem 11 implies the following result.

Theorem 17. DUAL-*2*-TEC-D *is in $\mathcal{FPT}$ and can be solved in time $\mathcal{O}^*(2^{5.3k_d})$.*

Theorems 16 and 17 therefore imply that both 2-TEC-D and DUAL-2-TEC-D are in $\mathcal{FPT}$, a result rarely observed in the context of parameterized complexity. However, in the case of t-TVC and CVC, we can show:

Theorem 18. DUAL-*2*-TVC-D *is $\mathcal{W}$[1]-complete. Also* DUAL-t-TVC-D *($t \geq 3$) and* DUAL-CVC-D *are $\mathcal{W}$[1]-hard.*

Proof. To show membership in $\mathcal{W}[1]$ of DUAL-2-TVC-D, we employ the "Turing way" [3, 9]. That is, we exhibit a Turing machine whose $f(k_d)$-step halting problem is solvable if and only if the given instance of DUAL-2-TVC-D is a YES-instance.

A 1-tape nondeterministic Turing machine M_G for graph $G = (V, E)$ would work as follows. The alphabet is $V \times \{0, 1\}$ (plus the end markers).

1. Guess k_d letters from $V \times \{0\}$ and write them on the tape.

2. Sweep back and forth on the tape and verify that the vertices are independent. (If two vertices u, v have been guessed with $u \in N(v)$, then the Turing machine would enter an infinite loop.)

   The second part of the tape alphabet can be used to protocol which two vertices are tested. If all pairs have been tested, then the tape contains an independent set I.

3. Now use the second part of the tape alphabet to cycle through all subsets of I. For each subset $\emptyset \neq X \subseteq I$, we have to test whether $X = N(v)$ for some $v \notin I$. If this is the case, then we have detected a vertex from the vertex cover $V \setminus I$ that has no neighbour from $V \setminus I$.

   To this end, an n-bit internal memory is used. Initially, this is an all-zero vector. Upon reading X off the tape, at most k_d bits are set to 1. Then, by the internal memory bit vector $X = N(v)$ can be checked in one further step.

If the infinite loop is not entered (i.e., $X \neq N(v)$ for all $v \in V \setminus I$), then the k_d bits are set to 0 again, and then the "next set" is selected by the bit vector counter on the tape.

Finally, the bit vector counter on the tape contains only ones, and then the machine will stop.

Hence, there is a function $f(k_d)$ such that G has a total vertex cover of size $n - k_d$ iff M_G stops in at most $f(k_d)$ steps.

We now show that DUAL-t-TVC-D is $\mathcal{W}[1]$-hard, for each $t \geq 2$. We use the same reduction as in Theorem 3, where $G = (V, E)$ is a connected graph with $n = |V| \geq 2$ and k_d is a parameter, given as an instance of INDEPENDENT SET-D. Then G has an independent set of size k_d if and only if the $(n + t)$-vertex graph G' has a t-tvc of size $n - k_d + (t - 1) = (n + t) - (k_d + 1)$.

In the case of DUAL-CVC-D, the proof is similar; the same reduction may be used with $t = 2$. □

6 Concluding remarks

In this paper we have defined the concepts of a t-tvc and a t-tec for $t \geq 1$, which are motivated by the notions of covering and clustering in graphs. We have presented $\mathcal{NP}$-completeness, approximability and parameterized complexity results for associated optimization and decision problems.

Until now, enumeration-based solutions to parameterized decision problems seemed to be doomed to give rise to a complexity function $\mathcal{O}^*(C^k)$ where C is quite large. Our $\mathcal{FPT}$ algorithms in this paper demonstrate how this can be overcome by introducing appropriate "colourings" and corresponding reduction rules within the search tree algorithm. A further example is the EDGE DOMINATING SET algorithm described in [10]. Moreover, a novel way of analyzing search trees that can be decomposed into two phases is exhibited; this has proved to be highly effective in the case of 2-TVC-D and CVC-D, and should also be applicable in improving the analysis of other fixed-parameter algorithms.

In Section 4, we described very briefly in outline an $\mathcal{O}^*(2.9316^k)$ algorithm for CVC-D. As mentioned in Section 1, an improved $\mathcal{O}^*(2.7606^k)$ algorithm for CVC-D will be reported in [27]. It is likely that a further improvement could be obtained by combining the approach of Moelle et al. with the reduction rules that we employ for our CVC-D algorithm.

The results in this paper leave open the following problems, among others, that are worthy of further consideration: (1) Formulate polynomial-time algorithms for t-TVC and t-TEC in restricted classes of graphs. (2) Formulate (if possible) $\mathcal{FPT}$ algorithms for t-TVC-D ($t > 2$). Are the corresponding parametric dual problems in $\mathcal{W}[1]$? (3) Consider "clustering" variants of vertex domination and edge domination.

Acknowledgements

The second author would like to thank Michele Zito for helpful discussions regarding 2-total vertex covers, and Pavol Hell for drawing our attention to reference [23] in connection with t-tree packings.

References

[1] E.M. Arkin, M.M. Halldórsson, and R. Hassin. Approximating the tree and tour covers of a graph. *Information Processing Letters*, 47:275–282, 1993.

[2] J.R.S. Blair. Personal communication, 2001.

[3] M. Cesati. The Turing way to parameterized complexity. *Journal of Computer and System Sciences*, 67:654–685, 2003.

[4] J. Chen, H. Fernau, I. A. Kanj, and G. Xia. Parametric duality and kernelization: Lower bounds and upper bounds on kernel size. In *Proceedings of STACS 2005: the 22nd Annual Symposium on Theoretical Aspects of Computer Science*, volume 3404 of *Lecture Notes in Computer Science*, pages 269–280. Springer, 2005.

[5] M. Chlebík and J. Chlebíková. Complexity of approximating bounded variants of optimization problems. *Theoretical Computer Science*, 354(3):320–338, 2006.

[6] E.J. Cockayne, R.M. Dawes, and S.T. Hedetniemi. Total domination in graphs. *Networks*, 10:211–219, 1980.

[7] P. Damaschke. Parameterized enumeration, transversals, and imperfect phylogeny reconstruction. *Theoretical Computer Science*, 351:337–350, 2006.

[8] I. Dinur and S. Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, 162(1):439–485, 2005.

[9] H. Fernau. Parameterized algorithmics: A graph-theoretic approach. Habilitationsschrift, University of Tübingen, 2005.

[10] H. Fernau. Edge dominating set: efficient enumeration-based exact algorithms. In *Proceedings of IWPEC 2006: the Third International Workshop on Parameterized and Exact Computation*, volume 4169 of *Lecture Notes in Computer Science*, pages 142–153. Springer, 2006.

[11] H. Fernau and D.F. Manlove. Vertex and edge covers with clustering properties: Complexity and algorithms. Technical Report TR-2006-210, University of Glasgow, Department of Computing Science, 2006.

[12] F. Fomin, D. Kratsch, and G. Woeginger. Exact (exponential) algorithms for the dominating set problem. In *Proceedings of WG '04: the 30th International Workshop on Graph-Theoretic Concepts in Computer Science*, volume 3353 of *Lecture Notes in Computer Science*, pages 245–256. Springer, 2004.

[13] B. Fuchs, W. Kern, Daniel Mölle, S. Richter, P. Rossmanith, and X. Wang. Dynamic programming for minimum Steiner trees. Technical Report zaik2005-492, University of Cologne, 2005.

[14] M. Gaertler. Clustering. In U. Brandes and T. Erlebach, editors, *Network Analysis*, volume 3418 of *Lecture Notes in Computer Science*, chapter 8, pages 178–215. Springer, 2005.

[15] T. Gallai. Über extreme Punkt-und Kantenmengen. *Ann. Univ. Sci. Budapest, Eötvös Sect. Math.*, 2:133–138, 1959.

[16] M.R. Garey and D.S. Johnson. The rectilinear Steiner tree problem is NP-complete. *SIAM Journal on Applied Mathematics*, 32(4):826–834, 1977.

[17] M.R. Garey, D.S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1:237–267, 1976.

[18] J. Guo, R. Niedermeier, and S. Wernicke. Parameterized complexity of generalized vertex cover problems. In *Proceedings of WADS 2005: the 9th International Workshop on Algorithms and Data Structures*, volume 3608 of *Lecture Notes in Computer Science*, pages 36–48. Springer, 2005.

[19] F. Harary. *Graph Theory*. Addison-Wesley, 1969.

[20] T. Haynes, S.T. Hedetniemi, and P.J. Slater, editors. *Domination in Graphs: Advanced Topics*. Marcel Dekker, 1998.

[21] T. Haynes, S.T. Hedetniemi, and P.J. Slater. *Fundamentals of Domination in Graphs*. Marcel Dekker, 1998.

[22] V. Kann. Maximum bounded H-matching is MAX SNP-complete. *Information Processing Letters*, 49:309–318, 1994.

[23] 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.

[24] D.F. Manlove. On the algorithmic complexity of twelve covering and independence parameters of graphs. *Discrete Applied Mathematics*, 91:155–175, 1999.

[25] S. Mitchell and S.T. Hedetniemi. Edge domination in trees. In *Proceedings of the 8th South-Eastern Conference on Combinatorics, Graph Theory and Computing*, pages 489–509. Utilitas Mathematica, 1977.

[26] D. Mölle, S. Richter, and P. Rossmanith. Enumerate and expand: Improved algorithms for connected vertex cover and tree cover. In *Proceedings of CSR '06: International Computer Science Symposium in Russia*, volume 3967 of *Lecture Notes in Computer Science*, pages 280–290. Springer, 2006.

[27] D. Mölle, S. Richter, and P. Rossmanith. Enumerate and expand: New runtime bounds for vertex cover variants. In *Proceedings of COCOON 2006: the 12th Annual International Computing and Combinatorics Conference* (to appear), Lecture Notes in Computer Science. Springer, 2006.

[28] D. Mölle, S. Richter, and P. Rossmanith. A faster algorithm for the Steiner Tree problem. In *Proceedings of STACS 2006: the 23rd Annual Symposium on Theoretical Aspects of Computer Science*, volume 3884 of *Lecture Notes in Computer Science*, pages 561–570. Springer, 2006.

[29] N. Nishimura, P. Ragde, and D.M. Thilikos. Fast fixed-paramter tractable algorithms for nontrival generalizations of vertex cover. *Discrete Applied Mathematics*, 152:229–245, 2005.

[30] 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.

[31] C.H. Papadimitriou and M. Yannakakis. Optimization, approximation and complexity classes. *Journal of Computer and System Sciences*, 43(3):425–440, 1991.

[32] E. Prieto and C. Sloper. Looking at the stars. *Theoretical Computer Science*, 351:437–445, 2006.

[33] C. Savage. Depth-first search and the vertex cover problem. *Information Processing Letters*, 14:233–235, 1982.

[34] S. Ueno, Y. Kajitani, and S. Gotoh. On the nonseparating independent set problem and feedback set problem for graphs with no vertex degree exceeding three. *Discrete Mathematics*, 72:355–360, 1988.

[35] M. Zito. *Randomised Techniques in Combinatorial Algorithms*. PhD thesis, University of Warwick, Department of Computer Science, 1999.