Proc. 9th WADS-05, Vol. 3608 in LNCS, pp. 36-48, Springer, 2005

Parameterized Complexity of Generalized Vertex Cover Problems

Jiong Guo*, Rolf Niedermeier*, and Sebastian Wernicke**

Institut für Informatik, Friedrich-Schiller-Universität Jena,
Ernst-Abbe-Platz 2, D-07743 Jena, Fed. Rep. of Germany
{guo,niedermr,wernicke}@minet.uni-jena.de

Abstract. Important generalizations of the Vertex Cover problem (Connected Vertex Cover, Capacitated Vertex Cover, and Maximum Partial Vertex Cover) have been intensively studied in terms of approximability. However, their parameterized complexity has so far been completely open. We close this gap here by showing that, with the size of the desired vertex cover as parameter, Connected Vertex Cover and Capacitated Vertex Cover are both fixed-parameter tractable while Maximum Partial Vertex Cover is W[1]-hard. This answers two open questions from the literature. The results extend to several closely related problems. Interestingly, although the considered generalized Vertex Cover problems behave very similar in terms of constant-factor approximability, they display a wide range of different characteristics when investigating their parameterized complexities.

1 Introduction

Given an undirected graph $G = (V, E)$, the NP-complete Vertex Cover problem is to find a set $C \subseteq V$ with $|C| \leq k$ such that each edge in E has at least one endpoint in C. In a sense, Vertex Cover could be considered the *Drosophila* of fixed-parameter algorithmics [17, 25]:

1. There is a long list of continuous improvements on the combinatorial explosion with respect to the parameter k when solving the problem exactly. The currently best exponential bound is below 1.28^k [8, 26, 14, 28, 12].
2. Vertex Cover has been a benchmark for developing sophisticated data reduction and problem kernelization techniques [1, 19].
3. It was the first parameterized problem where the usefulness of interleaving depth-bounded search trees and problem kernelization was proven [27].
4. Restricted to planar graphs, it was—besides Dominating Set—one of the central problems for the development of "subexponential" fixed-parameter algorithms and the corresponding theory of relative lower bounds [2, 4, 11].
5. Vertex Cover served as a testbed for algorithm engineering in the realm of fixed-parameter algorithms [1, 3, 13].

---

* Supported by the Deutsche Forschungsgemeinschaft, Emmy Noether research group PIAF (fixed-parameter algorithms), NI 369/4.

** Supported by Deutsche Telekom Stiftung and Studienstiftung des deutschen Volkes.

6. Studies of Vertex Cover led to new research directions within parameterized complexity such as counting [7], parallel processing [13], or using "vertex cover structure" as a general strategy to solve parameterized problems [30].

This probably incomplete list gives an impression of how important Vertex Cover was and continues to be for the whole field of parameterized complexity. However, research in this field to date appears to have neglected a closer investigation of recent significant generalizations and variants of Vertex Cover. These appear in various application scenarios such as drug design [22] and have so far only been studied in the context of their polynomial-time approximability. We close this gap here by providing several first-time parameterized complexity results, which also answers two open questions from the literature.

We are only aware of two papers that perform somewhat related research. First, Nishimura, Ragde, and Thilikos [29] also study generalizations of Vertex Cover. However, they follow a completely different route: Whereas we study concrete problems such as Capacitated Vertex Cover or Maximum Partial Vertex Cover on general graphs, their interest lies in recognizing general classes of graphs with a very special case of interest being the class of graphs with bounded vertex cover (refer to [29] for details). Second, Bläser [9] shows that some partial covering problems are fixed-parameter tractable when the parameter is the number of objects covered instead of the size of the covering set. (In this paper, as well as in the abovementioned studies, the parameter is always the size of the covering set.)

We deal with a whole list of vertex covering problems, all of them possessing constant-factor (mostly 2) polynomial-time approximation algorithms. Deferring their formal definitions to the next section, we now informally describe the studied problems and the known and new results. In the presentation of our results, n denotes the number of vertices and m denotes the number of edges of the input graph. The parameter k always denotes the size of the vertex cover.

1. For Connected Vertex Cover one demands that the vertex cover set is connected. This problem is known to have a factor-2 approximation [6]. We show that it can be solved in $O(6^k n + 4^k n^2 + 2^k n^2 \log n + 2^k nm)$ time. In addition, we derive results for the closely related variants Tree Cover and Tour Cover.
2. For Capacitated Vertex Cover, the "covering capacity" of each graph vertex is limited in that it may not cover all of its incident edges. This problem has a factor-2 approximation [22]. Addressing an open problem from [22], we show that Capacitated Vertex Cover can be solved in $O(1.2^{k^2} + n^2)$ time using an enumerative approach. We also provide a problem kernelization. Altogether, we thus show that Capacitated Vertex Cover—including two variants with "hard" and "soft" capacities—is fixed-parameter tractable.
3. For Maximum Partial Vertex Cover, one only wants to cover a specified number of edges (that is, not necessarily all) by at most k vertices. This problem is known to have a factor-2 approximation [10]. Answering an open question from [5], we show that this problem appears to be fixed-parameter intractable—it is W[1]-hard. The same is proven for its minimization version.

Proc. 9th WADS-05, Vol. 3608 in LNCS, pp. 36-48, Springer, 2005

Problem	Result	
Connected Vertex Cover	$6^k n + 4^k n^2 + 2^k n^2 \log n + 2^k nm$	Thm. 2
Tree Cover	$(2k)^k \cdot km$	Cor. 3
Tour Cover	$(4k)^k \cdot km$	Cor. 3
Capacitated Vertex Cover	$1.2^{k^2} + n^2$	Thm. 5
Soft Capacitated Vertex Cover	$1.2^{k^2} + n^2$	Thm. 10
Hard Capacitated Vertex Cover	$1.2^{k^2} + n^2$	Thm. 10
Maximum Partial Vertex Cover	W[1]-hard	Thm. 11
Minimum Partial Vertex Cover	W[1]-hard	Cor. 12

Table 1. New parameterized complexity results for some NP-complete generalizations of Vertex Cover shown in this work. The parameter k is the size of the desired vertex cover, m denotes the number of edges, and n denotes the number of vertices.

Summarizing, we emphasize that our main focus is on deciding between fixed-parameter tractability and W[1]-hardness for all of the considered problems. Interestingly, although all considered problems behave in more or less the same way from the viewpoint of polynomial-time approximability—all have factor-2 approximations—the picture becomes completely different from a parameterized complexity point of view: Maximum Partial Vertex Cover appears to be intractable and Capacitated Vertex Cover appears to be significantly harder than Connected Vertex Cover. Table 1 surveys all of our results.

2 Preliminaries and Previous Work

Parameterized complexity is a two-dimensional framework for studying the computational complexity of problems.[1] One dimension is the input size n (as in classical complexity theory) and the other one the *parameter* k (usually a positive integer). A problem is called *fixed-parameter tractable* (fpt) if it can be solved in $f(k) \cdot n^{O(1)}$ time, where f is a computable function only depending on k. A core tool in the development of fixed-parameter algorithms is polynomial-time preprocessing by *data reduction rules*, often yielding a *reduction to a problem kernel*. Here the goal is, given any problem instance x with parameter k, to transform it into a new instance x' with parameter k' such that the size of x' is bounded by some function only depending on k, (x, k) has a solution iff (x', k') has a solution, and $k' \leq k$. A formal framework to show *fixed-parameter intractability* was developed by Downey and Fellows [17] who introduced the concept of *parameterized reductions*. A parameterized reduction from a parameterized language L to another parameterized language L' is a function that, given an instance (x, k), computes in time $f(k) \cdot n^{O(1)}$ an instance (x', k') (with k' only depending on k) such that $(x, k) \in L \Leftrightarrow (x', k') \in L'$. The basic complexity class for fixed-parameter intractability is W[1] as there is good reason to believe that W[1]-hard problems are not fixed-parameter tractable [17].

[1] For a more detailed introduction see, e.g., [17, 19, 24].

In this work, we consider three directions of generalizing VERTEX COVER (VC), namely demanding that the vertices of the cover must be *connected* (Section 3), introducing *covering capacities* for the vertices (Section 4), and relaxing the condition that *all* edges in the graph must be covered (Section 5). Our corresponding parameterized complexity results are summarized in Table 1, the formal definitions of the problems follow.

CONNECTED VERTEX COVER: Given a graph $G = (V, E)$ and an integer $k \geq 0$, determine whether there exists a vertex cover C for G containing at most k vertices such that the subgraph of G induced by C is connected.

This problem is NP-complete and approximable within 2 [6]. Two variants are derived by introducing a weight function $w : E \to \mathbb{R}^+$ on the edges and requiring that the cover must induce a subgraph with a certain structure and minimum weight.

TREE COVER: Given a graph $G = (V, E)$ with edges weighted with positive real numbers, an integer $k \geq 0$, and a real number $W > 0$, determine whether there exists a subgraph $G' = (V', E')$ of G with $|V'| \leq k$ and $\sum_{e \in E'} w(e) \leq W$ such that V' is a vertex cover for G and G' is a tree.[2]

The closely related problem TOUR COVER differs from TREE COVER only in that the edges in G' should form a closed walk instead of a tree. Note that a closed walk can contain repeated vertices and edges. Both TREE COVER and TOUR COVER were introduced in [6] where it is shown that TREE COVER is approximable within 3.55 and TOUR COVER within 5.5. Könemann et al. [23] improved both approximation factors to 3.

Section 4 considers the CAPACITATED VERTEX COVER (CVC) problem and related variants. Here, each vertex $v \in V$ is assigned a *capacity* $c(v) \in \mathbb{N}^+$ that limits the number of edges it can cover when being part of the vertex cover.

Definition 1. *Given a capacitated graph $G = (V, E)$ and a vertex cover C for G. We call C* capacitated vertex cover *if there exists a mapping $f : E \to C$ which maps each edge in E to one of its two endpoints such that the total number of edges mapped by f to any vertex $v \in C$ does not exceed $c(v)$.*

CAPACITATED VERTEX COVER: Given a vertex-weighted (with positive real numbers) and capacitated graph G, an integer $k \geq 0$, and a real number $W \geq 0$, determine whether there exists a capacitated vertex cover C for G containing at most k vertices such that $\sum_{v \in C} w(v) \leq W$.

The CVC problem was introduced by Guha et al. [22] who also give a factor-2 approximation algorithm. Two special flavors of CVC exist in the literature that arise by allowing "copies" of a vertex to be in the capacitated vertex cover [22, 15, 20]. In that context, taking a vertex l times into the capacitated vertex cover causes the vertex to have l times its original capacity. The number of such copies is unlimited in the SOFT CAPACITATED VERTEX COVER (SOFT CVC) problem while it may be restricted for each vertex individually in the HARD CAPACITATED VERTEX COVER (HARD CVC) problem. For unweighted HARD CVC, the best

[2] TREE COVER is equivalent to CONNECTED VERTEX COVER for unweighted graphs.

known approximation algorithm achieves a factor of 2 [20]. The weighted version Hard CVC is at least as hard to approximate as Set Cover [15].

Section 5 considers a third direction of VC generalizations besides connectedness and capacitation. In the Maximum Partial Vertex Cover problem, we relax the condition that *all* edges must be covered.

> Maximum Partial Vertex Cover: Given a graph $G = (V, E)$ and two integers $k \geq 0$ and $t \geq 0$, determine whether there exists a vertex subset $V' \subseteq V$ of size at most k such that V' covers at least t edges.

This problem was introduced by Bshouty and Burroughs [10] who showed it to be approximable within 2. Further improvements can be found in [21]. Note that Maximum Partial Vertex Cover is fixed-parameter tractable with respect to the parameter t [9]. In case of Minimum Partial Vertex Cover we are asked for a vertex subset with *at least* k vertices covering *at most* t edges.

3 Connected Vertex Cover and Variants

In this section we show that Connected Vertex Cover is fixed-parameter tractable with respect to the size of the connected vertex cover. More precisely, it can be solved by an algorithm running in $O(6^k n + 4^k n^2 + 2^k n^2 \log n + 2^k nm)$ time where n and m denote the number of vertices and edges in the input graph and k denotes the size of the connected vertex cover. We modify this algorithm to also show the fixed-parameter tractability for two variants of Connected Vertex Cover, namely Tree Cover and Tour Cover.

We solve Connected Vertex Cover by using the Dreyfus-Wagner algorithm as a subprocedure for computing a Steiner minimum tree in a graph [18]. For an undirected graph $G = (V, E)$, a subgraph T of G is called a *Steiner tree* for a subset K of V if T is a tree containing all vertices in K such that all leaves of T are elements of K. The vertices of K are called the *terminals* of T. A *Steiner minimum tree* for K in G is a Steiner tree T such that the number of edges contained in T is minimum. Finding a Steiner minimum tree leads to an NP-complete problem. The Dreyfus-Wagner algorithm computes a Steiner minimum tree for a set of at most l terminals in $O(3^l n + 2^l n^2 + n^2 \log n + nm)$ time [18].

Our algorithm for Connected Vertex Cover consists of two steps:

1. Enumerate all minimal vertex covers with at most k vertices. If one of the enumerated minimal vertex covers is connected, then output it and terminate.
2. Otherwise, for each of the enumerated minimal vertex covers C, use the Dreyfus-Wagner algorithm to compute a Steiner minimum tree with C as the set of terminals. If one minimal vertex cover has a Steiner minimum tree T with at most $k - 1$ edges, then return the vertex set of T as output; otherwise, there is no connected vertex cover with at most k vertices.

Theorem 2. Connected Vertex Cover *can be solved in* $O(6^k n + 4^k n^2 + 2^k n^2 \log n + 2^k nm)$ *time.*

Proof. The first step of the algorithm is correct since each connected vertex cover (covc) contains at least one minimal vertex cover. For a given graph, there are at most 2^k minimal vertex covers with at most k vertices. We can enumerate all such minimal vertex covers in $O(2^k \cdot m)$ time. Then, the running time of the first step is $O(2^k \cdot m)$.

The correctness of the second step follows directly from the following easy to prove observation: For a set of vertices C, there exists a connected subgraph of G with at most k vertices which contains all vertices in C iff there exists a Steiner tree in G with C as the terminal set and at most $k-1$ edges. By applying the Dreyfus-Wagner algorithm on G with C as the terminal set, we can easily find out whether there are $k-|C|$ vertices from $V \setminus C$ connecting C and, hence, whether there is a covc with at most k vertices and containing C. Since $|C| < k$, the second step can be done in $O(2^k \cdot (3^k n + 2^k n^2 + n^2 \log n + nm)) = O(6^k n + 4^k n^2 + 2^k n^2 \log n + 2^k nm)$ time. □

The algorithm for CONNECTED VERTEX COVER can be modified to solve TREE COVER and TOUR COVER. The proof is omitted.

Corollary 3. TREE COVER *and* TOUR COVER *can be solved in $O((2k)^k \cdot km)$ and $O((4k)^k \cdot km)$ time, respectively.*

4 Capacitated Vertex Cover and Variants

In this section we present fixed-parameter algorithms for the CVC problem and its variants HARD CVC and SOFT CVC. In the case of CVC, the easiest way to show its fixed-parameter tractability is to give a reduction to a problem kernel. This is what we begin with here, afterwards complementing it with an enumerative approach for further improving the overall time complexity.

Proposition 4. *Given an n-vertex graph $G = (V, E)$ and an integer $k \geq 0$ as part of an input instance for* CVC*, then it is possible to construct an $O(4^k \cdot k^2)$-vertex graph $\tilde{G}$ such that G has a size-k solution for* CVC *iff $\tilde{G}$ has a size-k solution for* CVC*. In the special case of uniform vertex weights, $\tilde{G}$ has only $O(4^k \cdot k)$ vertices. The construction of $\tilde{G}$ can be performed in $O(n^2)$ time.*

Proof. We first assume uniform vertex weights, generalizing the approach to weighted graphs at the end of the proof.

Let $u, v \in V$, $u \neq v$, and $\{u, v\} \notin E$. The simple observation that lies at the heart of the data reduction rule needed for the kernelization is that if the open neighborhoods of u and v coincide (i.e., $N(u) = N(v)$) and $c(u) < c(v)$, then u is part of a minimum capacitated vertex cover only if v is as well. We can generalize this finding to a data reduction rule: Let $\{v_1, v_2, \ldots, v_{k+1}\} \subseteq V$ with the induced subgraph $G[\{v_1, v_2, \ldots, v_{k+1}\}]$ being edgeless, and $N(v_1) = N(v_2) = \cdots = N(v_{k+1})$. Call this the *neighbor set*. Then delete from G a vertex $v_i \in \{v_1, v_2, \ldots, v_{k+1}\}$ which has minimum capacity. This rule is correct because any size-k capacitated vertex cover C containing v_i can be modified by replacing v_i with a vertex from $\{v_1, v_2, \ldots, v_{k+1}\}$ which is not in C.

Based on this data reduction rule, $\tilde{G}$ can be computed from G as claimed by the following two steps:

1. Use the straightforward linear-time factor-2 approximation algorithm to find a vertex cover S for G of size at most $2k'$ (where k' is the size of a minimum vertex cover for G and hence $k' \leq k$). If $|S| > 2k$, then we can stop because then no size-k (capacitated) vertex cover can be found. Note that $V \setminus S$ induces an edgeless subgraph of G.
2. Examining $V \setminus S$, check whether there is a subset of $k + 1$ vertices that fulfill the premises of the above rule. Repeatedly apply the data reduction rule until it is no longer applicable. Note that this process continuously shrinks $V \setminus S$.

The above computation is clearly correct. The number of all possible neighbor sets can be at most 2^{2k} (the number of different subsets of S). For each neighbor set, there can be at most k neighboring vertices in $V \setminus S$; otherwise, the reduction rule would apply. Hence, in the worst case we can have at most $2^{2k} \cdot k$ vertices in the remaining graph $\tilde{G}$. The generalization to non-uniform vertex weights works as follows: We have $|S| \leq 2k$. Hence, the vertices in $V \setminus S$ may have maximum vertex degree $2k$ and the capacity of a vertex in $V \setminus S$ greater than $2k$ without any harm can be replaced by capacity $2k$. Therefore, without loss of generality, one may assume that the maximum capacity of vertices in $V \setminus S$ is $2k$. We then have to modify the reduction rule as follows. If there are vertices $v_1, v_2, \ldots, v_{2k^2+1} \in V$ with $N(v_1) = N(v_2) = \ldots = N(v_{2k^2+1})$, partition them into subsets of vertices with equal capacity. There are at most $2k$ of these sets. If such a set contains more than k vertices, delete the vertex with maximum weight. Altogether, we thus end up with a problem kernel of $2^{2k} \cdot 2k^2 = O(4^k \cdot k^2)$ vertices.

It remains to justify the polynomial running time. First, note that the trivial factor-2 approximation algorithm runs in time $O(|E|) = O(n^2)$. Second, examining the common neighborhoods can be done in $O(n^2)$ time by successively partitioning the vertices in $V \setminus S$ according to their neighborhoods. □

Clearly, a simple brute-force search within the reduced instance (with a size of only $O(4^k \cdot k^2)$ vertices) already yields the fixed-parameter tractability of CVC, albeit in time proportional to $\binom{4^k \cdot k^2}{k}$. As the next theorem shows, we can do much better concerning the running time.

Theorem 5. *The* CVC *problem can be solved in* $O(1.2^{k^2} + n^2)$ *time.*

The theorem is proved by first giving an algorithm to solve Capacitated Vertex Cover and then proving its running time. The basic idea behind the algorithm is as follows: We start with a minimal vertex cover $C = \{c_1, \ldots, c_i\} \subseteq V$ for the input graph $G = (V, E)$. Due to lack of capacities, C is not necessarily a *capacitated* vertex cover for G. Hence, if C is not a capacitated vertex cover, we need to add some additional vertices from $V \setminus C$ to C in order to provide additional capacities. More precisely, since for each vertex $v \in (V \setminus C)$ all of its neighbors are in C, adding v can be seen as "freeing" exactly one unit of capacity for as many as $c(v)$ neighbors of v. The algorithm uses an exhaustive search approach based on this observation by enumerating all possible patterns of capacity-freeing and for each pattern computing the cheapest set of vertices from $V \setminus C$ (if one exists) that matches it.

Definition 6. *Given a graph $G = (V, E)$ and a vertex cover $C = \{c_1, \ldots, c_i\} \subseteq V$ for G. A* capacity profile of length i *is a binary string* $s = s[1] \ldots s[i] \in \{0, 1\}^i$. *A vertex $w \in V \setminus C$ is said to* match a capacity profile s *if it is incident to each vertex $c_j \in C$ with $s[j] = 1$ and its capacity is at least the number of ones in s.*

Using Definition 6, the following pseudocode gives an algorithm for CVC.

Algorithm: Capacitated Vertex Cover
Input: A capacitated and vertex-weighted graph $G = (V, E)$, $k \in \mathbb{N}^+$, $W \in \mathbb{R}^+$
Output: "Yes" if G has a capacitated vertex cover of size at most k with weight $\leq W$; "No" otherwise

01 Perform the kernelization from Proposition 4 on G
02 **for** every minimal vertex cover C of G with size $i \leq k$ **do**
03   **if** C is a cap. vertex cover with weight $\leq W$ **then return** "Yes"
04   **for** every multiset M of $(k - i)$ capacity profiles of length i **do**
05   remove the all-zero profiles from M
06   find the cheapest set $\hat{C} \subseteq (V \setminus C)$ so that there exists a bijective mapping $f : \hat{C} \to M$ where each $\hat{c} \in \hat{C}$ matches the capacity profile of $f(\hat{c})$. Set $\hat{C} \leftarrow \emptyset$ if no such set exists
07   **if** $\hat{C} \neq \emptyset$, the weight of $\hat{C}$ is $\leq W$, and $C \cup \hat{C}$ is a capacitated vertex cover for G **then return** "Yes"
08 **return** "No"

Lemma 7. *The given algorithm for* Capacitated Vertex Cover *is correct.*

Proof. Preprocessing the graph in line *01* is correct according to Proposition 4. Since a capacitated vertex cover for a graph $G = (V, E)$ is also a vertex cover, its vertices can be partitioned into two sets C and $\hat{C}$ such that C is a minimal vertex cover for G. Each vertex in $\hat{C}$ gives additional capacity to a subset of the vertices in C, i.e., for every $\hat{c} \in \hat{C}$, we can construct a capacity profile $s_{\hat{c}}$ where $s_{\hat{c}}[j] = 1$ if and only if $\hat{c}$ uses its capacity to cover the edge to the j-th vertex in C. The correctness of the algorithm follows from its exhaustive nature: It tries all minimal vertex covers, all possible combinations of capacity profiles and for each combination determines the cheapest possible set $\hat{C}$ such that $C \cup \hat{C}$ is a capacitated vertex cover for G. □

Lemma 8. *The given algorithm for* CVC *runs in $O(1.2^{k^2} + n^2)$ time.*

Proof. The preprocessing in line *01* can be carried out in $O(n^2)$ time according to Proposition 4. This leads to a new graph containing at most $\tilde{n} = O(4^k \cdot k^2)$ vertices. Line *02* of the algorithm can be executed in $O(2^k) \cdot \tilde{n}^{O(1)}$ time and causes the subsequent lines *03*–*07* to be called at most 2^k times. Due to [15, Lemma 1], we can decide in $\tilde{n}^{O(1)}$ time whether a given vertex cover is also a capacitated vertex cover (lines *03* and *07*). For line *04*, note that for a given $0 \leq i \leq k$

there exist 2^i different capacity profiles of that length. Furthermore, it is well-known that given a set A where $|A| = a$, there exist exactly $\binom{a+b-1}{b}$ b-element multisets with elements drawn from A. Hence, line *04* causes lines *05–07* to be executed $\binom{2^i+(k-i)-1}{k-i}$ times. The delay between enumeration of two multisets can be kept constant. As it will be shown in Lemma 9, line *06* takes $\tilde{n}^{O(1)}$ time. Overall, the running time of the algorithm is bounded from above by

$$O(n^2) + 2^k \cdot \max_{1 \leq i \leq k} \binom{2^i + (k-i) - 1}{k-i} \cdot \tilde{n}^{O(1)}.$$

With some effort, we can bound this number by $O(n^2 + 1.2^{k^2})$. □

It remains to show the running time for line *06* of the algorithm.

Lemma 9. *Given a weighted, capacitated graph $G = (V, E)$, a vertex cover C of G of size $i \leq k$ for G, and a multiset M of $k - i$ capacity profiles of length i. Then, it takes $n^{O(1)}$ time to find the cheapest set $\hat{C} \subseteq (V \setminus C)$ so that there exists a bijective mapping $f : \hat{C} \to M$ where each $\hat{c} \in \hat{C}$ matches the capacity profile of $f(\hat{c})$ or determine that no such set $\hat{C}$ exists.*

Proof. Finding $\hat{C}$ is equivalent to finding a minimum weight maximum bipartite matching on the bipartite graph $G' = (V_1', V_2', E')$ where each vertex in V_1' represents a capacity profile from M, $V_2' = V \setminus C$, and two vertices $v \in V_1', u \in V_2'$ are connected by an edge in E' if and only if the vertex represented by u matches the profile represented by v (the weight of the edge is $w(u)$). Finding such a matching is well-known to be solvable in polynomial time [16]. □

It is possible to solve Soft CVC and Hard CVC by adapting the above algorithm for CVC: Observe that if we choose multiple copies of a vertex into the cover, each of these copies will have its own individual capacity profile. Thus, only line *06* of the CVC algorithm has to be adapted to solve Soft CVC and Hard CVC. The proof is omitted.

Corollary 10. Soft CVC *and* Hard CVC *are solvable in $O(1.2^{k^2} + n^2)$ time.*

5 Maximum and Minimum Partial Vertex Cover

All the Vertex Cover variants we studied in the previous sections are known to have a polynomial-time constant-factor approximation (mostly factor 2). All of them were shown to be fixed-parameter tractable. By way of contrast, we now present a result where a variant that has a polynomial-time factor-2 approximation is shown to be fixed parameter intractable. More precisely, we show that Maximum Partial Vertex Cover (MaxPVC) is W[1]-hard with respect to the size k of the partial vertex cover by giving a parameterized reduction from the W[1]-complete Independent Set problem [17] to MaxPVC. With the solution size as parameter, we also show the W[1]-hardness of its minimization version MinPVC by a reduction from Clique.

Independent Set: Given a graph $G = (V, E)$ and an integer $k \geq 0$, determine whether there is a vertex subset $I \subseteq V$ with at least k vertices such that the subgraph of G induced by I contains no edge.

An *independent set* in a graph is a set of pairwise nonadjacent vertices.

Theorem 11. Maximum Partial Vertex Cover *is W[1]-hard with respect to the size of the cover.*

Proof. We give a parameterized reduction from Independent Set to MaxPVC. Given an input instance $(G = (V, E), k)$ of Independent Set. For every vertex $v \in V$, let $\deg(v)$ denote the degree of v in G. We construct a new graph $G' = (V', E')$ in the following way: For each vertex $v \in V$ we insert $|V| - \deg(v)$ new vertices into G and connect each of these new vertices with v. In the following, we show that a size-k independent set in G one-to-one corresponds to a size-k partial vertex cover in G' which covers $t := k \cdot |V|$ edges.

Firstly, a size-k independent set in G also forms a size-k independent set in G'. Moreover, each of these k vertices has exactly $|V|$ incident edges. Then, these k vertices form a partial vertex cover covering $k \cdot |V|$ edges. Secondly, if we have a size-k partial vertex cover in G' which covers $k \cdot |V|$ edges, then we know that none of the newly inserted vertices in G' can be in this cover. Hence, this cover contains k vertices from V. Moreover, a vertex in G' can cover at most $|V|$ edges and two adjacent vertices can cover only $2|V| - 1$ edges. Therefore, no two vertices in this partial vertex cover can be adjacent, which implies that this partial cover forms a size-k independent set in G. □

In Minimum Partial Vertex Cover (MinPVC), we wish to choose *at least k* vertices such that *at most t* edges are covered. Through a parameterized reduction from the W[1]-complete Clique problem [17], it is possible to show analogously to MaxPVC that MinPVC is also W[1]-hard. This reduction works in a similar way as the reduction in the proof above. The proof is omitted.

Corollary 12. Minimum Partial Vertex Cover *is W[1]-hard with respect to the size of the cover.*

6 Conclusion

We extended and completed the parameterized complexity picture for natural variants and generalizations of Vertex Cover. Notably, whereas the fixed-parameter tractability of Vertex Cover immediately follows from a simple search tree strategy, this appears not to be the case for all of the problems studied here. Table 1 in Section 2 summarizes our results, all of which, to the best of our knowledge, are new in the sense that no parameterized complexity results have been known before for these problems. Our fixed-parameter tractability results clearly generalize to cases where the vertices have real weights $\geq c$ for some given constant $c > 0$ and the parameter becomes the weight of the desired vertex cover (see [28] for corresponding studies for Vertex Cover). Our work also complements the numerous approximability results for these problems. It is a task for future research to significantly improve on the presented worst-case running times (exponential factors in parameter k). In particular, it would be interesting to learn more about the amenability of the considered problems to problem kernelization by (more) efficient data reduction techniques.

Besides the significant interest (with numerous applications behind) in the studied problems on their own, we want to mention one more feature of our work that lies a little aside. Adding our results to the already known large arsenal of facts about Vertex Cover, this problem can be even better used and understood as a seed problem for parameterized complexity as a whole: New aspects now related to vertex covering by means of our results are issues such as enumerative techniques or parameterized reduction. This might be of particular use when learning or teaching parameterized complexity through basically *one* natural and easy to grasp problem—Vertex Cover—and its "straightforward" generalizations.

Acknowledgment. We are grateful to an anonymous referee of WADS 2005 for spotting a flaw in a previous proof of the fixed-parameter tractability of Connected Vertex Cover.

References

1. F. N. Abu-Khzam, R. L. Collins, M. R. Fellows, M. A. Langston, W. H. Suters, and C. T. Symons. Kernelization algorithms for the Vertex Cover problem: theory and experiments. In *Proc. ALENEX04*, pages 62–69. ACM/SIAM, 2004.
2. J. Alber, H. L. Bodlaender, H. Fernau, T. Kloks, and R. Niedermeier. Fixed parameter algorithms for Dominating Set and related problems on planar graphs. *Algorithmica*, 33(4):461–493, 2002.
3. J. Alber, F. Dorn, and R. Niedermeier. Experimental evaluation of a tree decomposition based algorithm for Vertex Cover on planar graphs. *Discrete Applied Mathematics*, 145(2):219–231, 2005.
4. J. Alber, H. Fernau, and R. Niedermeier. Parameterized complexity: exponential speed-up for planar graph problems. *Journal of Algorithms*, 52:26–56, 2004.
5. J. Alber, J. Gramm, and R. Niedermeier. Faster exact algorithms for hard problems: a parameterized point of view. *Discrete Mathematics*, 229:3–27, 2001.
6. E. M. Arkin, M. M. Halldórsson, and R. Hassin. Approximating the tree and tour covers of a graph. *Information Processing Letters*, 47(6):275–282, 1993.
7. V. Arvind and V. Raman. Approximation algorithms for some parameterized counting problems. In *Proc. 13th ISAAC*, volume 2518 of *LNCS*, pages 453–464. Springer, 2002.
8. R. Balasubramanian, M. R. Fellows, and V. Raman. An improved fixed-parameter algorithm for Vertex Cover. *Information Processing Letters*, 65(3):163–168, 1998.
9. M. Bläser. Computing small partial coverings. *Information Processing Letters*, 85(6):327–331, 2003.
10. N. H. Bshouty and L. Burroughs. Massaging a linear programming solution to give a 2-approximation for a generalization of the Vertex Cover problem. In *Proc. 15th STACS*, volume 1373 of *LNCS*, pages 298–308. Springer, 1998.

11. L. Cai and D. Juedes. On the existence of subexponential parameterized algorithms. *Journal of Computer and System Sciences*, 67(4):789–807, 2003.
12. L. S. Chandran and F. Grandoni. Refined memorisation for Vertex Cover. *Information Processing Letters*, 93(3):125–131, 2005.
13. J. Cheetham, F. Dehne, A. Rau-Chaplin, U. Stege, and P. J. Taillon. Solving large FPT problems on coarse-grained parallel machines. *Journal of Computer and System Sciences*, 67(4):691–706, 2003.
14. J. Chen, I. A. Kanj, and W. Jia. Vertex Cover: further observations and further improvements. *Journal of Algorithms*, 41:280–301, 2001.
15. J. Chuzhoy and J. S. Naor. Covering problems with hard capacities. In *Proc. 43rd FOCS*, pages 481–489. IEEE Computer Society Press, 2002.
16. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, 2nd Edition*. MIT Press, 2001.
17. R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer, 1999.
18. S. E. Dreyfus and R. A. Wagner. The Steiner problem in graphs. *Networks*, 1:195–207, 1972.
19. M. R. Fellows. New directions and new challenges in algorithm design and complexity, parameterized. In *Proc. 8th WADS*, volume 2748 of *LNCS*, pages 505–520. Springer, 2003.
20. R. Gandhi, E. Halperin, S. Khuller, G. Kortsarz, and A. Srinivasan. An improved approximation algorithm for Vertex Cover with hard capacities. In *Proc. 30th ICALP*, volume 2719 of *LNCS*, pages 164–175. Springer, 2003.
21. R. Gandhi, S. Khuller, and A. Srinivasan. Approximation algorithms for partial covering problems. In *Proc. 28th ICALP*, volume 2076 of *LNCS*, pages 225–236. Springer, 2001.
22. S. Guha, R. Hassin, S. Khuller, and E. Or. Capacitated vertex covering. *Journal of Algorithms*, 48(1):257–270, 2003.
23. J. Könemann, G. Konjevod, O. Parekh, and A. Sinha. Improved approximations for tour and tree covers. *Algorithmica*, 38(3):441–449, 2004.
24. R. Niedermeier. Ubiquitous parameterization—invitation to fixed-parameter algorithms. In *Proc. MFCS*, volume 3153 of *LNCS*, pages 84–103. Springer, 2004.
25. R. Niedermeier. *Invitation to Fixed-Parameter Algorithms.* Oxford University Press, forthcoming, 2005.
26. R. Niedermeier and P. Rossmanith. Upper bounds for Vertex Cover further improved. In *Proc. 16th STACS*, volume 1563 of *LNCS*, pages 561–570. Springer, 1999.
27. R. Niedermeier and P. Rossmanith. A general method to speed up fixed-parameter-tractable algorithms. *Information Processing Letters*, 73:125–129, 2000.
28. R. Niedermeier and P. Rossmanith. On efficient fixed-parameter algorithms for Weighted Vertex Cover. *Journal of Algorithms*, 47(2):63–77, 2003.
29. N. Nishimura, P. Ragde, and D. M. Thilikos. Fast fixed-parameter tractable algorithms for nontrivial generalizations of Vertex Cover. In *Proc. 7th WADS*, volume 2125 of *LNCS*, pages 75–86. Springer, 2001.
30. E. Prieto and C. Sloper. Either/or: using vertex cover structure in designing FPT-algorithms—the case of k-Internal Spanning Tree. In *Proc. 8th WADS*, volume 2748 of *LNCS*, pages 474–483. Springer, 2003.