Extension of vertex cover and independent set in some classes of graphs

Katrin Casel [1,2], Henning Fernau [2], Mehdi Khosravian Ghadikoalei [3], Jérôme Monnot[3], and Florian Sikora[3]

[1] Hasso Plattner Institute, University of Potsdam, 14482 Potsdam, Germany
[2] Universität Trier, Fachbereich 4, Informatikwissenschaften, 54286 Trier, Germany, {casel,fernau}@informatik.uni-trier.de
[3] Université Paris-Dauphine, PSL University, CNRS, LAMSADE, 75016 Paris, France, {mehdi.khosravian-ghadikolaei,jerome.monnot, florian.sikora}@lamsade.dauphine.fr

Abstract. We study extension variants of the classical problems Vertex Cover and Independent Set. Given a graph $G = (V, E)$ and a vertex set $U \subseteq V$, it is asked if there exists a *minimal* vertex cover (resp. *maximal* independent set) S with $U \subseteq S$ (resp. $U \supseteq S$). Possibly contradicting intuition, these problems tend to be NP-complete, even in graph classes where the classical problem can be solved efficiently. Yet, we exhibit some graph classes where the extension variant remains polynomial-time solvable. We also study the parameterized complexity of theses problems, with parameter $|U|$, as well as the optimality of simple exact algorithms under ETH. All these complexity considerations are also carried out in very restricted scenarios, be it degree or topological restrictions (bipartite, planar or chordal graphs). This also motivates presenting some explicit branching algorithms for degree-bounded instances. We further discuss the *price of extension*, measuring the distance of U to the closest set that can be extended, which results in natural optimization problems related to extension problems for which we discuss polynomial-time approximability.

Key words: extension problems, special graph classes, approximation algorithms, NP-completeness

1 Introduction

We will consider *extension problems* related to the classical graph problems Vertex Cover and Independent Set. Informally in the extension version of Vertex Cover, the input consists of both a graph G and a subset U of vertices, and the task is to extend U to an inclusion-wise minimal vertex cover of G (if possible). With Independent Set, given a graph G and a subset U of vertices, we are looking for an inclusion-wise maximal independent set of G contained in U.

Studying such version is interesting when one wants to develop efficient enumeration algorithms or also for branching algorithms, to name two examples of a list of applications given in [6].

Related work In [5], it is shown that extension of partial solutions is NP-hard for computing prime implicants of the dual of a Boolean function; a problem which can also be seen as trying to find a minimal hitting set for the prime implicants of the input function. Interpreted in this way, the proof from [5] yields NP-hardness for the minimal extension problem for 3-Hitting Set (but polynomial-time solvable if $|U|$ is constant). This result was extended in [2] to prove NP-hardness for computing the extensions of vertex sets to minimal dominating sets (Ext DS), even restricted to planar cubic graphs. Similarly, it was shown in [1] that extensions to minimal vertex covers restricted to planar cubic graphs is NP-hard. The first *systematic* study of this type of problems was exhibited in [6] providing quite a number of different examples of this type of problem.

An *independent system* is a set system $(V, \mathcal{E})$, $\mathcal{E} \subseteq 2^V$, that is hereditary under inclusion. The extension problem Ext Ind Sys (also called Flashlight) for independent system was proposed in [17]. In this problem, given as input $X, Y \subseteq V$, one asks for the existence of a maximal independent set including X and that does not intersect with Y. Lawler et al. proved that Ext Ind Sys is NP-complete, even when $X = \emptyset$ [17]. In order to enumerate all (inclusion-wise) minimal dominating sets of a given graph, Kanté et al. studied a restriction of Ext Ind Sys: finding a minimal dominating set containing X but excluding Y. They proved that Ext DS is NP-complete, even in special graph classes like split graphs, chordal graphs and line graphs [15, 14]. Moreover, they proposed a linear algorithm for split graphs when X, Y is a partition of the clique part [13].

Organization of the paper After some definitions and first results in Section 2, we focus on bipartite graphs in Section 3 and give hardness results holding with strong degree or planarity constraints. We also study parameterized complexity at the end of this section and comment on lower bound results based on ETH. In Section 4, we give positive algorithmic results on chordal graphs, with a combinatorial characterization for the subclass of trees. We introduce the novel concept of *price of extension* in Section 5 and discuss (non-)approximability for the according optimization problems. In Section 6, we prove several algorithmic results for bounded-degree graphs, based on a list of reduction rules and simple branching. Finally, in Section 7, we give some prospects of future research.

2 Definitions and preliminary results

Throughout this paper, we consider simple undirected graphs only, to which we refer as *graphs*. A graph can be specified by the set V of vertices and the set E of edges; every edge has two endpoints, and if v is an endpoint of e, we also say that e and v are *incident*. Let $G = (V, E)$ be a graph and $U \subseteq V$; $N_G(U) = \{v \in V : \exists u \in U (vu \in E)\}$ denotes *the neighborhood* of U in G and $N_G[U] = U \cup N_G(U)$ denotes *the closed neighborhood* of U. For singleton sets $U = \{u\}$, we simply write $N_G(u)$ or $N_G[u]$, even omitting G if clear from the context. The cardinality of $N_G(u)$ is called *degree* of u, denoted $d_G(u)$. A graph where all vertices have degree k is called *k-regular*; 3-regular graphs are called *cubic*. If 3 upper-bounds the degree of all vertices, we speak of *subcubic graphs*.

A vertex set U induces the graph $G[U]$ with vertex set U and $e \in E$ being an edge in $G[U]$ iff both endpoints of e are in U. A vertex set U is called *independent* if $U \cap N_G(U) = \emptyset$; U is called *dominating* if $N_G[U] = V$; U is a *vertex cover* if each edge e is incident to at least one vertex from U. A graph is called *bipartite* if its vertex set decomposes into two independent sets. A vertex cover S is *minimal* if any proper subset $S' \subset S$ of S is not a vertex cover. Clearly, a vertex cover S is minimal iff each vertex v in S possesses a *private edge*, i.e., an edge vu with $u \notin S$. An independent set S is *maximal* if any proper superset $S' \supset S$ of S is not an independent set. The two main problems discussed in this paper are:

Ext VC
Input: A graph $G = (V, E)$, a set of vertices $U \subseteq V$.
Question: Does G have a minimal vertex cover S with $U \subseteq S$?

Ext IS
Input: A graph $G = (V, E)$, a set of vertices $U \subseteq V$.
Question: Does G have a maximal independent set S with $S \subseteq U$?

For Ext VC, the set U is also referred to as the set of *required* vertices.

Remark 1. (G, U) is a *yes*-instance of Ext VC iff $(G, V \setminus U)$ is a *yes*-instance of Ext IS, as complements of maximal independent sets are minimal vertex covers.

Since adding or deleting edges between vertices of U does not change the minimality of feasible solutions of Ext VC, we can first state the following.

Remark 2. For Ext VC (and for Ext IS) one can always assume the required vertex set (the set $V \setminus U$) is either a clique or an independent set.

The following theorem gives a combinatorial characterization of *yes*-instances of Ext VC that is quite important in our subsequent discussions.

Theorem 3. *Let $G = (V, E)$ be a graph and $U \subseteq V$ be a set of vertices. The three following conditions are equivalent:*

(i) (G, U) is a yes-instance of Ext VC.
(ii) $(G[N_G[U]], N_G[U] \setminus U)$ is a yes-instance of Ext IS.
(iii) There exists an independent dominating set $S' \subseteq N_G[U] \setminus U$ of $G[N_G[U]]$.

3 Bipartite graphs

In this section, we focus on bipartite graphs. We prove that Ext VC is NP-complete, even if restricted to cubic, or planar subcubic graphs. Due to Remark 1, this immediately yields the same type of results for Ext IS. We add some algorithmic notes on planar graphs that are also valid for the non-bipartite case. Also, we discuss results based on ETH. We conclude the section by studying the parameterized complexity of Ext VC in bipartite graphs when parameterized by the size of U.

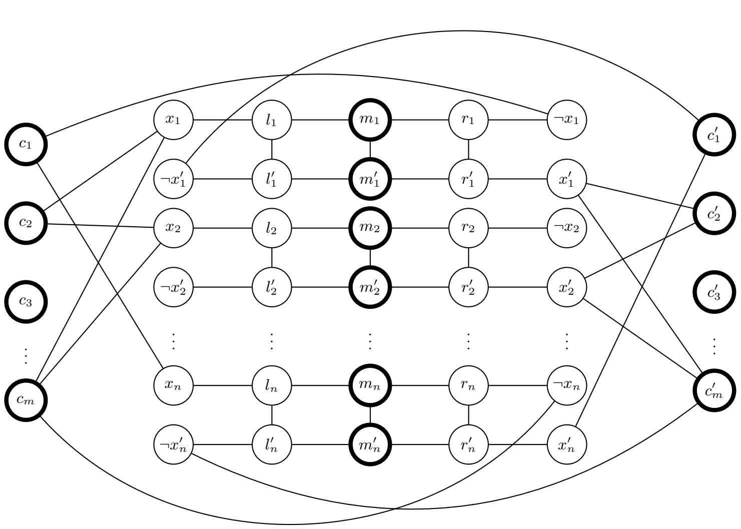

Fig. 1. Graph $G = (V, E)$ for Ext VC built from I. Vertices of U have a bold border.

Theorem 4. Ext VC *(and* Ext IS*) is* NP*-complete in cubic bipartite graphs.*

Proof. We reduce from 2-balanced 3-SAT, denoted $(3, B2)$-SAT, which is NP-hard by [3, Theorem 1], where an instance I is given by a set C of CNF clauses over a set X of Boolean variables such that each clause has exactly 3 literals and each variable appears exactly 4 times, twice negative and twice positive. The bipartite graph associated to I is $BP = (C \cup X, E(BP))$ with $C = \{c_1, \ldots, c_m\}$, $X = \{x_1, \ldots, x_n\}$ and $E(BP) = \{c_j x_i \colon x_i \text{ or } \neg x_i \text{ is literal of } c_j\}$.

For an instance $I = (C, X)$ of $(3, B2)$-SAT, we build a cubic bipartite graph $G = (V, E)$ by duplicating instance I (here, vertices $C' = \{c'_1, \ldots, c'_m\}$ and $X' = \{x'_1, \ldots, x'_n\}$ are the duplicate variants of vertices $C = \{c_1, \ldots, c_m\}$ and $X = \{x_1, \ldots, x_n\}$) and by connecting gadgets as done in Figure 1. We also add the following edges between the two copies: $l_i l'_i$, $m_i m'_i$ and $r_i r'_i$ for $i = 1, \ldots, n$. The construction is illustrated in Figure 1 and clearly, G is a cubic bipartite graph. Finally we set $U = \{c_i, c'_i \colon i = 1, \ldots, m\} \cup \{m_j, m'_j \colon j = 1, \ldots, n\}$.

We claim that I is satisfiable iff G admits a minimal vertex cover containing U.

Assume I is satisfiable and let T be a truth assignment which satisfies all clauses. We set $S = \{\neg x_i, l_i, \neg x'_i, r'_i \colon T(x_i) = \mathit{true}\} \cup \{x_i, r_i, x'_i, l'_i \colon T(x_i) = \mathit{false}\} \cup U$. We can easily check that S is a minimal vertex cover containing U.

Conversely, assume that G possesses a minimal vertex cover S containing U. For a fixed i, we know that $|\{l_i, l'_i, r_i, r'_i\} \cap S| \geq 2$ to cover the edges $l_i l'_i$ and $r_i r'_i$. If $\{l_i, r_i\} \subseteq S$ (resp. $\{l'_i, r'_i\} \subseteq S$), then S is not a minimal vertex cover because m_i (resp. m'_i) can be deleted, a contradiction. If $\{l_i, l'_i\} \subseteq S$ (resp. $\{r_i, r'_i\} \subseteq S$), then S must contain another vertex to cover $r_i r'_i$ (resp. $l_i l'_i$), leading to the previous case, a contradiction. Hence, if $\{l_i, r'_i\} \subseteq S$ (resp., $\{r_i, l'_i\} \subseteq S$), then $\{\neg x_i, \neg x'_i\} \subseteq S$ (resp., $\{x_i, x'_i\} \subseteq S$), since the edges $l'_i \neg x'_i$ and $r_i \neg x_i$ (resp., $l_i x_i$ and $r'_i x_i$) must be covered. In conclusion, by setting $T(x_i) = \textit{true}$ if $\neg x_i \in S$ and $T(x_i) = \textit{false}$ if $x_i \in S$ we obtain a truth assignment T which satisfies all clauses, because $\{C_i, C'_i \colon i = 1, \ldots, m\} \subseteq U \subseteq S$. □

Theorem 5. Ext IS *is* NP*-complete on planar bipartite subcubic graphs.*

Algorithmic notes for the planar case By distinguishing between whether a vertex belongs to the cover or not and further, when it belongs to the cover, if it already has a private edge or not, it is not hard to design a dynamic programming algorithm that decides in time $\mathcal{O}^*(c^t)$ if (G, U) is a *yes*-instance of Ext VC or not, given a graph G together with a tree decomposition of width t. With some more care, even $c = 2$ can be achieved, but this is not so important here. Rather, below we will make explicit another algorithm for trees that is based on several combinatorial properties and hence differs from the DP approach sketched here for the more general notion of treewidth-bounded graphs.

Moreover, it is well-known that planar graphs of order n have treewidth bounded by $\mathcal{O}(\sqrt{n})$. In fact, we can obtain a corresponding tree decomposition in polynomial time, given a planar graph G. Piecing things together, we obtain:

Theorem 6. Ext VC *can be solved in time* $\mathcal{O}^*(2^{\mathcal{O}(\sqrt{n})})$ *on planar graphs.*

Remarks on the Exponential Time Hypothesis Assuming ETH, there is no $2^{o(n+m)}$-algorithm for solving n-variable, m-clause instances of $(3, B2)$-SAT. As our reduction from $(3, B2)$-SAT increases the size of the instances only in a linear fashion, we can immediately conclude:

Theorem 7. *There is no $2^{o(n+m)}$-algorithm for n-vertex, m-edge bipartite subcubic instances of* Ext VC*, unless ETH fails.*

This also motivates us to further study exact exponential-time algorithms. We can also deduce optimality of our algorithms for planar graphs based on the following auxiliary result.

Proposition 8. *There is no algorithm that solves* 4-Bounded Planar 3-Connected SAT *(see [16]) on instances with n variables and m clauses in time $2^{o(\sqrt{n+m})}$, unless ETH fails.*

Corollary 9. *There is no $2^{o(\sqrt{n})}$ algorithm for solving* Ext VC *on planar instances of order n, unless ETH fails.*

Remarks on Parameterized Complexity We now study our problems in the framework of parameterized complexity where we consider the size of the set of fixed vertices as *standard parameter* for our extension problems.

Theorem 10. Ext VC *with standard parameter is* W[1]*-complete, even when restricted to bipartite instances.*

Theorem 11. Ext VC *with standard parameter is in* FPT *on planar graphs.*

4 Chordal and Circular-arc graphs

An undirected graph $G = (V, E)$ is *chordal* iff each cycle of G with a length at least four has a chord (an edge linking two non-consecutive vertices of the cycle) and G is *circular-arc* if it is the intersection graph of a collection of n arcs around a circle. We will need the following problem definition.

Minimum Independent Dominating Set (MinISDS for short)
Input: A graph $G = (V, E)$.
Solution: Subset of vertices $S \subseteq V$ which is independent and dominating.
Output: Solution S that minimizes $|S|$.

Weighted Minimum Independent Dominating Set (or WMinISDS for short) corresponds to the vertex-weighted variant of MinISDS, where each vertex $v \in V$ has a non-negative weight $w(v) \geq 0$ associated to it and the goal consists in minimizing $w(S) = \sum_{v \in S} w(v)$. If $w(v) \in \{a, b\}$ with $0 \leq a < b$, the weights are called *bivaluate*, and $a = 0$ and $b = 1$ corresponds to *binary weights*.

Remark 12. MinISDS for chordal graphs has been studied in [10], where it is shown that the restriction to binary weights is solvable in polynomial-time. Bivalued MinISDS with $a > 0$ however is already NP-hard on chordal graphs, see [7]. WMinISDS (without any restriction on the number of distinct weights) is also polynomial-time solvable in circular-arc graphs [8].

Corollary 13. Ext VC *is polynomial-time decidable in chordal and in circular-arc graphs.*

Farber's algorithm [10] (used in Corollary 13) runs in linear-time and is based on the resolution of a linear programming using primal and dual programs. Yet, it would be nice to find a (direct) combinatorial linear-time algorithm for chordal and circular-arc graphs, as this is quite common in that area. We give a first step in this direction by presenting a characterization of *yes*-instances of Ext VC on trees. Consider a tree $T = (V, E)$ and a set of vertices U. A subtree $T' = (V', E')$ (i.e., a connected induced subgraph) of a tree T is called *edge full with respect to* (T, U) if $U \subseteq V'$, $d_{T'}(u) = d_T(u)$ for all $u \in U$. A subtree $T' = (V', E')$ is *induced edge full with respect to* (T, U) if it is edge full with respect to $(T, U \cap V')$.

For our characterization, we use a coloring of vertices with colors black and white. If $T = (V, E)$ is a tree and $X \subseteq V$, we use $T[X \to \text{black}]$ to denote the colored tree where exactly the vertices from X are colored black. Further define the following class of black and white colored trees $\mathcal{T}$, inductively as follows.
Base case: A tree with a single vertex x belongs to $\mathcal{T}$ if x is black.
Inductive step: If $T \in \mathcal{T}$, the tree resulting from the addition of a P_3 (3 new vertices that form a path p), one endpoint of p being black, the two other vertices being white and the white endpoint of p linked to a black vertex of T, is in $\mathcal{T}$.

The following theorem can be viewed as an algorithm for Ext VC on trees.

Theorem 14. *Let $T = (V, E)$ be a tree and $U \subseteq V$ be an independent set. Then, (T, U) is a yes-instance of* Ext VC *iff there is no subtree $T' = (V', E')$ of T that is induced edge full with respect to (T, U) such that $T'[U \to \text{black}] \in \mathcal{T}$.*

5 Price of extension

Considering the possibility that some set U might not be extendible to any minimal solution, one might ask how wrong U is as a choice for an extension problem. One idea to evaluate this, is to ask how much U has to be altered when aiming for a minimal solution. Described differently for our extension problems at hand, we want to discuss how many vertices of U have to be deleted for Ext VC (added for Ext IS) in order to arrive at a *yes*-instance of the extension problem. The magnitude of how much U has to be altered can be seen as the price that has to be paid to ensure extendibility. To formally discuss this concept, we consider according optimization problems. From an instance $I = (G, U)$ of Ext VC or Ext IS, we define the two NPO problems:

Max Ext VC
Input: A graph $G = (V, E)$, a set of vertices $U \subseteq V$.
Solutions: Minimal vertex cover S of G.
Output: Solution S that maximizes $|S \cap U|$.

Min Ext IS
Input: A graph $G = (V, E)$, a set of vertices $U \subseteq V$.
Solutions: Maximal independent set S of G.
Output: Solution S that minimizes $|U| + |S \cap (V \setminus U)|$.

For $\Pi =$Max Ext VC or Min Ext IS, we denote by $opt_{\Pi}(I, U)$ the value of an optimal solution of Max Ext VC or Min Ext IS, respectively. Since for both of them, $opt_{\Pi}(I, U) = |U|$ iff (G, U) is a *yes*-instance of Ext VC or Ext IS, respectively, we deduce that Max Ext VC and Min Ext IS are NP-hard as soon as Ext VC and Ext IS are NP-complete. Alternatively, we could write $opt_{\text{Max Ext VC}}(G, U) = \arg\max\{U' \subseteq U : (G, U') \text{ is a } \textit{yes}\text{-instance of Ext VC}\}$, $opt_{\text{Min Ext IS}}(G, U) = \arg\min\{U' \supseteq U : (G, U') \text{ is a } \textit{yes}\text{-instance of Ext IS}\}$.

Similarly to Remark 1, one observes that the decision variants of Max Ext VC and Min Ext IS are equivalent, more precisely:

$$opt_{\text{Max Ext VC}}(G, U) + opt_{\text{Min Ext IS}}(G, V \setminus U) = |V| \,. \tag{1}$$

We want to discuss polynomial-time approximability of Max Ext VC and Min Ext IS. Considering Max Ext VC on $G = (V, E)$ and the particular subset $U = V$ (resp., Min Ext IS with $U = \emptyset$), we obtain two well known optimization problems called upper vertex cover (UVC for short, also called maximum minimal vertex cover) and maximum minimal independent set (ISDS for short). In [18], the computational complexity of these problems are studied (among 12 problems), and (in)approximability results are given in [19, 4] for UVC and in [11] for ISDS where lower bounds of $O(n^{\varepsilon-1/2})$ and $O(n^{1-\varepsilon})$, respectively, for graphs on n vertices are given for every $\varepsilon > 0$. Analogous bounds can be derived depending on the maximum degree Δ. In particular, we deduce:

Corollary 15. *For any constant $\varepsilon > 0$, any $\rho \in \mathcal{O}\left(n^{1-\varepsilon}\right)$ and $\rho \in \mathcal{O}\left(\Delta^{1-\varepsilon}\right)$, there is no polynomial-time ρ-approximation for* Min Ext IS *on graphs of n vertices and maximum degree Δ, even when $U = \emptyset$, unless* P = NP.

Theorem 16. Max Ext VC *is as hard as* MaxIS *to approximate even if the set U of required vertices forms an independent set.*

Sketch. Let $G = (V, E)$ be an instance of MaxIS. Construct $H = (V_H, E_H)$ from G, where vertex set V_H contains two copies of V, V and $V' = \{v' : v \in V\}$. Let $E_H = E \cup \{vv' : v \in V\}$. Consider $I = (H, U)$ as instance of Max EXT VC, where the required vertex subset is given by $U = V'$.
We claim: H has a minimal vertex cover containing k vertices from U iff G has a maximal independent set of size k.

Using the strong inapproximability results for MaxIS given in [20, 21], observing $\Delta(H) = \Delta(G) + 1$ and $|V_H| = 2|V|$, we deduce the following result.

Corollary 17. *For any constant $\varepsilon > 0$, any $\rho \in \mathcal{O}\left(\Delta^{1-\varepsilon}\right)$ and $\rho \in \mathcal{O}\left(n^{1-\varepsilon}\right)$, there is no polynomial-time ρ-approximation for* Max Ext VC *on graphs of n vertices and maximum degree Δ, unless* $\mathsf{P} = \mathsf{NP}$.

In contrast to the hardness results on these restricted graph classes from the previous sections, we find that restriction to bipartite graphs or graphs of bounded degree improve approximability of Max Ext VC. For the following results, we assume, w.l.o.g., that the input graph is connected, non-trivial and therefore without isolated vertices, as we can solve our problems separately on each connected component and then combine the results. By simply selecting the side containing the largest number of vertices from U, we can show the following.

Theorem 18. *A 2-approximation for* Max Ext VC *on bipartite graphs can be computed in polynomial time.*

Theorem 19. *A Δ-approximation for* Max Ext VC *on graphs of maximum degree Δ can be computed in polynomial time.*

Proof. Let $G = (V, E)$ be connected of maximum degree Δ, and $U \subseteq V$ be an instance of Max Ext VC. If $\Delta \leq 2$, or if $G = K_{\Delta+1}$ (the complete graph on $\Delta + 1$ vertices), it is easy to check Max Ext VC is polynomial-time solvable; actually in these two cases, G is either chordal or circular-arc and Theorem 20 gives the conclusion. Hence, assume $\Delta \geq 3$ and $G \neq K_{\Delta+1}$. By Brooks's Theorem, we can color G properly with at most Δ colors in polynomial-time (even linear). Let $(S_1, \ldots, S_\ell)$ be such coloring of G with $\ell \leq \Delta$. For $i \leq \ell$, set $U_i = U \cap N_G(S_i)$ where we recall $N_G(S_i)$ is the open neighborhood of S_i. By construction, S_i is an independent set which dominates U_i in G so it can be extended to satisfy (*iii*) of Theorem 3, so (G, U_i) is a *yes*-instance of Ext VC. Choosing $U' = \arg\max |U_i|$ yields a Δ-approximation, since on the one hand $\sum_{i=1}^{\ell} |U_i| \geq |U \cap \left(\cup_{i=1}^{\ell} N_G(S_i)\right)| = |U \cap V|$ and on the other hand $\Delta \times |U'| \geq \sum_{i=1}^{\ell} |U_i| \geq |U| \geq opt_{\textsc{Max Ext VC}}(G, U)$. □

Along the lines of Corollary 13 with more careful arguments, we can prove:

Theorem 20. Max Ext VC *can be solved optimally for chordal graphs and circular-arc graphs in polynomial time.*

Proof. Let (G, U) be an instance of Max Ext VC where $G = (V, E)$ is a chordal graph (resp., a circular-arc graph) and U is an independent set. We build a weighted graph G' for WMinISDS such that G' is the subgraph of G induced by $N_G[U]$ and the weights on vertices are given by $w(v) = 1$ if $v \in U$ and $w(v) = 0$ for $v \in N_G[U] \setminus U$. Thus, we get: $opt_{\text{WMinISDS}}(G', w) = |U| - opt_{\text{Max Ext VC}}(G, U)$. □

6 Bounded degree graphs

Our NP-hardness results also work for the case of graphs of bounded degree, hence it is also interesting to consider Ext VC with standard parameter with an additional degree parameter $\varDelta$.

Theorem 21. Ext VC *is in* FPT *when parameterized both by the standard parameter and by the maximum degree $\varDelta$ of the graph.*

Sketch. Recursively, the algorithm picks some $u \in U$ and branches on every neighbor $x \in N(u) \setminus U$ to be excluded from the vertex cover to ensure a private edge xu for u. This is a limited choice of at most $\varDelta$ neighbors, and considering the new instance $(G - N[x], U \setminus N[x])$, this yields a running time in $\mathcal{O}^*(\varDelta^k)$.

Let us look at this algorithm more carefully in the case of $\varDelta = 3$ analyzing it from the standpoint of exact algorithms, i.e., dependent on the number of vertices n of the graph. Our algorithm has a branching vector of $(2, 2, 2)$ (in each branch, u and a neighbor of u is removed, so n reduces by 2), resulting in a branching number upper-bounded by 1.733. However, the worst case is a vertex in U that has three neighbors of degree one. Clearly, this can be improved. We propose the following reduction rules for Ext VC on an instance (G, U), $G = (V, E)$, which have to be applied exhaustively and in order:

0. If $U = \emptyset$, then answer *yes*.
1. If some $u \in U$ is of degree zero, then (G, U) is a *no*-instance.
2. If some $x \notin U$ is of degree zero, then delete x from V.
3. If $u, u' \in U$ with $uu' \in E$, then delete uu' from E.
4. If $u \in U$ is of degree one, then the only incident edge $e = ux$ must be private, hence we can delete $N[x]$ from V and all u' from U that are neighbors of x.
5. If $u \in U$ has a neighbor x that is of degree one, then assume $e = ux$ is the private edge of u, so that we can delete u and x from V and u from U.

After executing the reduction rules exhaustively, the resulting graph has only vertices of degree two and three (in the closed neighborhood of U) if we start with a graph of maximum degree three. This improves the branching vector to $(3, 3, 3)$, resulting in a branching number upper-bounded by 1.443. However, the rules are also valid for arbitrary graphs, as we show in the following.

Lemma 22. *The reduction rules are sound for general graphs when applied exhaustively and in order.*

Theorem 23. Ext VC *can be solved in time* $\mathcal{O}^*((\sqrt[3]{\Delta})^n)$ *on graphs of order* n *with maximum degree* Δ.

This gives interesting branching numbers for $\Delta = 3$: 1.443, $\Delta = 4$: 1.588, $\Delta = 5$: 1.710, etc., but from $\Delta = 8$ on this is no better than the trivial $\mathcal{O}^*(2^n)$-algorithm.

Let us remark that the same reasoning that resulted in Rule 5 is valid for:

5'. If $x \notin U$ satisfies $N(x) \subseteq U$, then delete $N[x]$ from V and from U.
6. Delete $V \setminus N_G[U]$. (inspired by Theorem 3)

We now run the following branching algorithm:

1. Apply all reduction rules exhaustively in the order given by the numbering.
2. On each connected component, do:
   - Pick a vertex v of lowest degree.
   - If $v \in U$: Branch on all possible private neighbors.
   - If $v \notin U$: Branch on if v is not in the cover or one of its neighbors.

A detailed analysis of the suggested algorithm gives the following result.

Theorem 24. Ext VC *on subcubic graphs can be solved in time* $\mathcal{O}^*(1.26^n)$ *on graphs of order* n.

Corollary 25. Ext VC *on subcubic graphs can be solved in time* $\mathcal{O}^*(2^{|U|})$ *with fixed vertex set* U.

Our reduction rules guarantee that each vertex not in U (and hence in $N_G(U)$) has one or two neighbors in U, and each vertex in U has two or three neighbors in $N_G(U)$. Hence, $|N_G(U)| \leq 3|U|$. In general, due to Rule 6:

Theorem 26. Ext VC *on graphs of maximum degree* Δ *allows for a vertex kernel of size* $(\Delta + 1)|U|$, *parameterized by the size of the given vertex set* U.

Looking at the dual parameterization (i.e., Ext IS with standard parameter), we can state due to all reduction rules:

Theorem 27. Ext VC *on graphs of maximum degree* Δ *allows for a vertex kernel of size* $\frac{\Delta-1}{2}|V \setminus U|$, *parameterized by* $|V \setminus U|$.

For $\Delta = 3$, we obtain vertex kernel bounds of $4|U|$ and $2|V \setminus U|$, respectively. With the computations of [9, Cor. 3.3 & Cor. 3.4], we can state the following.

Corollary 28. *Unless* $\mathsf{P} = \mathsf{NP}$, *for any* $\varepsilon > 0$, *there is no size* $(2-\varepsilon)|U|$ *and no size* $(\frac{4}{3} - \varepsilon)|V \setminus U|$ *vertex kernel for* Ext VC *on subcubic graphs, parameterized by* $|U|$ *or* $|V \setminus U|$, *respectively.*

This shows that our (relatively simple) kernels are quite hard to improve on.

Remark 29. Note that the arguments that led to the FPT-result for Ext VC on graphs of bounded degree (by providing a branching algorithm) also apply to graph classes that are closed under taking induced subgraphs and that guarantee the existence of vertices of small degree. This idea leads to a branching algorithm with running time $\mathcal{O}^*(5^{|U|})$ or $\mathcal{O}^*(1.32^{|V|})$.

Remark 30. Let us mention that we also derived several linear-time algorithms for solving Ext VC (and hence Ext IS) on trees in this paper. (1) A simple restriction of the mentioned DP algorithm on graphs of bounded treewidth solves this problem. (2) Apply our reduction rules exhaustively. (3) Check the characterization given in Theorem 14. Also, Theorem 20 provides another polynomial-time algorithm on trees.

7 Conclusions

We have found many graph classes where Ext VC (and hence also Ext IS) remains NP-complete, but also many classes where these problems are solvable in poly-time. The latter findings could motivate looking into parameterized algorithms that consider the distance from favorable graph classes in some way.

It would be also interesting to study further optimization problems that could be related to our extension problems, for instance the following ones, here formulated as decision problems (a) Given G, U, k, is it possible to delete at most k vertices from the graph such that (G, U) becomes a *yes*-instance of Ext VC? Clearly, this problem is related to the idea of the price of extension discussed in this paper, in particular, if one restricts the possibly deleted vertices to be vertices from U. (b) Given G, U, k, is it possible to add at most k edges from the graph such that (G, U) becomes a *yes*-instance of Ext VC? Recall that adding edges among vertices from U does not change our problem, as they can never be private edges, but adding edges elsewhere might create private edges for certain vertices. Such problems would be defined according to the general idea of graph editing problems studied quite extensively in recent years. These problems are particularly interesting in graph classes where Ext VC is solvable in poly-time.

Considering the underlying classical optimization problems, it is also a rather intriguing question to decide for a given set U if it can be extended not just to any inclusion minimal vertex cover but to a globally smallest one, as a kind of *optimum-extension* problem. However, it has been shown in [12, Cor. 4.13] that the Vertex Cover Member problem (given a graph G and a vertex v, does there exist a vertex cover of minimum size that has v as a member, or, in other words, that extends $\{v\}$ is complete for the complexity class $\mathsf{P}_{\|}^{\mathsf{NP}}$, which is above NP and co-NP.

Acknowledgements The first author was partially supported by the Deutsche Forschungsgemeinschaft DFG (FE 560/6-1).

References

1. C. Bazgan, L. Brankovic, K. Casel, and H. Fernau. On the complexity landscape of the domination chain. In S. Govindarajan and A. Maheshwari, editors, *Algorithms and Discrete Applied Mathematics - Second International Conference, CALDAM*, volume 9602 of *LNCS*, pages 61–72. Springer, 2016.

2. C. Bazgan, L. Brankovic, K. Casel, H. Fernau, K. Jansen, K. Klein, M. Lampis, M. Liedloff, J. Monnot, and V. T. Paschos. The many facets of upper domination. *Theor. Comput. Sci.*, 717:2–25, 2018.
3. P. Berman, M. Karpinski, and A. D. Scott. Approximation hardness of short symmetric instances of MAX-3SAT. *ECCC*, (049), 2003.
4. N. Boria, F. D. Croce, and V. T. Paschos. On the max min vertex cover problem. *Disc. Appl. Math.*, 196:62–71, 2015.
5. E. Boros, V. Gurvich, and P. L. Hammer. Dual subimplicants of positive Boolean functions. *Optim. Meth. Softw.*, 10(2):147–156, 1998.
6. K. Casel, H. Fernau, M. K. Ghadikolaei, J. Monnot, and F. Sikora. On the complexity of solution extension of optimization problems. *CoRR*, abs/1810.04553, 2018.
7. G. J. Chang. The weighted independent domination problem is NP-complete for chordal graphs. *Disc. Appl. Math.*, 143(1-3):351–352, 2004.
8. M. Chang. Efficient algorithms for the domination problems on interval and circular-arc graphs. *SIAM J. Comp.*, 27(6):1671–1694, 1998.
9. J. Chen, H. Fernau, I. A. Kanj, and G. Xia. Parametric duality and kernelization: Lower bounds and upper bounds on kernel size. *SIAM J. Comp.*, 37(4):1077–1106, 2007.
10. M. Farber. Independent domination in chordal graphs. *Oper. Res. Lett.*, 4(1):134–138, 1982.
11. M. M. Halldórsson. Approximating the minimum maximal independence number. *Inf. Proc. Lett.*, 46(4):169–172, 1993.
12. E. Hemaspaandra, H. Spakowski, and J. Vogel. The complexity of Kemeny elections. *Theor. Comput. Sci.*, 349(3):382–391, 2005.
13. M. M. Kanté, V. Limouzy, A. Mary, and L. Nourine. On the enumeration of minimal dominating sets and related notions. *SIAM J. Disc. Math.*, 28(4):1916–1929, 2014.
14. M. M. Kanté, V. Limouzy, A. Mary, L. Nourine, and T. Uno. Polynomial delay algorithm for listing minimal edge dominating sets in graphs. In F. Dehne, J. Sack, and U. Stege, editors, *Workshop on Algorithms and Data Structures, WADS*, volume 9214 of *LNCS*, pages 446–457. Springer, 2015.
15. M. M. Kanté, V. Limouzy, A. Mary, L. Nourine, and T. Uno. A polynomial delay algorithm for enumerating minimal dominating sets in chordal graphs. In E. W. Mayr, editor, *International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2015*, volume 9224 of *LNCS*, pages 138–153. Springer, 2016.
16. J. Kratochvíl. A special planar satisfiability problem and a consequence of its NP-completeness. *Disc. Appl. Math.*, 52:233–252, 1994.
17. E. L. Lawler, J. K. Lenstra, and A. H. G. R. Kan. Generating all maximal independent sets: NP-hardness and polynomial-time algorithms. *SIAM J. Comp.*, 9:558–565, 1980.
18. D. F. Manlove. On the algorithmic complexity of twelve covering and independence parameters of graphs. *Disc. Appl. Math.*, 91(1-3):155–175, 1999.
19. S. Mishra and K. Sikdar. On the hardness of approximating some NP-optimization problems related to minimum linear ordering problem. *RAIRO Inf. théor. Appl.*, 35(3):287–309, 2001.
20. L. Trevisan. Non-approximability results for optimization problems on bounded degree instances. In J. S. Vitter, P. G. Spirakis, and M. Yannakakis, editors, *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, STOC*, pages 453–461. ACM, 2001.
21. D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Th. Comp. Syst.*, 3(1):103–128, 2007.