On the Complexity of Solution Extension of Optimization Problems

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

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

Abstract. The question if a given partial solution to a problem can be extended reasonably occurs in many algorithmic approaches for optimization problems. For instance, when enumerating minimal vertex covers of a graph $G = (V, E)$, one usually arrives at the problem to decide for a vertex set $U \subseteq V$ (*pre-solution*), if there exists a *minimal* vertex cover S (i. e., a vertex cover $S \subseteq V$ such that no proper subset of S is a vertex cover) with $U \subseteq S$ (*minimal extension* of U). We propose a general, partial-order based formulation of such extension problems which allows to model parameterization and approximation aspects of extension, and also highlights relationships between extension tasks for different specific problems. As examples, we study a number of specific problems which can be expressed and related in this framework.
In particular, we discuss extension variants of the problems dominating set and feedback vertex/edge set. All these problems are shown to be NP-complete even when restricted to bipartite graphs of bounded degree, with the exception of our extension version of feedback edge set on undirected graphs which is shown to be solvable in polynomial time. For the extension variants of dominating and feedback vertex set, we also show NP-completeness for the restriction to planar graphs of bounded degree. As non-graph problem, we also study an extension version of the bin packing problem. We further consider the parameterized complexity of all these extension variants, where the parameter is a measure of the pre-solution as defined by our framework.

1 Introduction

The very general problem of determining the quality of a given partial solution occurs basically in every algorithmic approach which computes solutions in some sense gradually. Pruning search-trees, proving approximation guarantees or the efficiency of enumeration strategies usually requires a suitable way to decide if a partial solution is a reasonable candidate to pursue. Consider for example the classical concept of minimal vertex covers for graphs. The task of finding a maximum cardinality minimal vertex cover (or an approximation of it) as well as enumerating all minimal vertex covers naturally leads to solving the following extension problem: Given a graph $G = (V, E)$ and a vertex set $U \subseteq V$, does there exist a *minimal* vertex cover S with $U \subseteq S$?

In this paper, we want to consider these kinds of subproblems which we call *extension problems*. Informally, in an extension version of a problem, we are given in the input a partial solution (that we call a pre-solution) to be extended into a *minimal* or a *maximal* solution for the problem (but not necessarily to a solution of globally minimum or maximum value, i. e., we consider a certain partial ordering on the set of pre-solutions). Extension problems as studied in this paper are encountered for many computational tasks/strategies. Let us go back to the prototype classical problem Vertex Cover:

– When running a search tree algorithm, usually parts of the constructed solution are fixed, i. e., some vertices are picked to be part of the solution. It is highly desirable to be able to prune branches of the search tree as early as possible. Hence, it would be very nice to tell (in polynomial time) if such a solution part can be part of a minimal vertex cover.

– The same type of reasoning is especially relevant if one aims to enumerate or count all minimal dominating sets [22–24], transversals in a hypergraph [7, 15], cliques [38], or similar structures [5, 6, 17, 30, 31, 39]. Note that building up from only relevant pre-solutions yields the possibility to define an order on solutions and hence enumerate without repetition. It was this scenario of enumeration where the question of extension was asked for Vertex Cover Extension in [15].
– Analyzing any algorithmic strategy that greedily builds a solution in a stepwise fashion essentially always boils down to proving some quality of the intermediate pre-solutions, where "quality" translates to "possibility to be extended".
– Especially greedily solving the task of finding (and also approximating) a minimal vertex cover lower-bounded by a given number k (a minimal vertex cover of maximum size) requires knowing that (at least a significant part of) the pre-solution remains in a minimal solution. Considering the strategies that give good approximations for the minimum vertex cover problem, it is the difficulty of deciding extendability of pre-solutions that hinders using them for finding a minimal vertex cover of maximum size.

Another popular strategy for problem solving is local search. In a more general setting, this can be viewed as a strategy to move in the space of pre-solutions in order to find a good (if not optimum) solution. The danger of local search is to get stuck at a pre-solution that cannot yield any minimal solution, let alone a minimum solution. It is desirable to detect such a situation where the search was trapped as early as possible.

On a more general note, the following question was already asked in 1956 by Kurt Gödel in a famous letter to Joh(an)n von Neumann [40]: *"It would be interesting to know* [...] *how strongly in general the number of steps in finite combinatorial problems can be reduced with respect to simple exhaustive search."* The mentioned pruning of search branches and hence the question of finding (pre-)solution extensions lies at the heart of this question.

Related work. This paper is not the first one to consider extension problems, however, we are not aware of a *systematic* study of this type of problems. The question of finding extensions to minimal solutions was encountered in the context of proving hardness results for (efficient) enumeration algorithms for Boolean formulae, in the context of matroids and similar situations; see [6, 31]. More precisely, it is NP-hard to decide if a partial solution can be extended for the problem of computing prime implicants of the dual of a Boolean function; a problem which can also be seen as finding a minimal hitting set for the set of prime implicants of the input function. Interpreted in this way, the proof from [6] yields NP-hardness for an extension version of the 3-Hitting Set problem; see also [36, Théorème 2.16]. Also, notice that the whole of Sec. 6 of [37] (a rather recent paper on the efficient enumeration of solutions) is dedicated to the enumeration of minimal respectively maximal solutions.

Another historical reference concerning extension problems that can be found in the literature considers independence systems. An *independence system* is a set system $(E, \mathcal{I})$, with $\mathcal{I} \subseteq 2^E$ being closed under taking subsets. Elements of $\mathcal{I}$ are also called independent sets. In the extension problem Ext Ind Sys, given as input[4] $(E, \mathcal{I}, U)$ where $U \subseteq E$, one asks for the existence of an inclusion-wise maximal independent set contained in U. In [33], it has been proved in Theorem 1 that, unless $\mathsf{P} = \mathsf{NP}$, there is no algorithm that generates all K maximal independent sets of $(E, \mathcal{I})$, running in time polynomial in $|E|$ and K. The proof of this result reduces from Satisfiability (SAT) and can be modified to produce an NP-hardness result for Ext Ind Sys. More precisely, with the notations from [33], define $E = \{T_1, F_1, \dots, T_N, F_N\}$ and let the independence system $(E, \mathcal{I})$ be derived from the Boolean formula F as in [33]. Consider $U_1 = E \setminus \{T_1, T_2\}$, $U_2 = E \setminus \{T_1, F_2\}$, $U_3 = E \setminus \{F_1, T_2\}$, $U_4 = E \setminus \{F_1, F_2\}$. Observe that F is satisfiable if and only if (at least) one of the four Ext Ind Sys instances $(E, \mathcal{I}, U_i)$ is a *yes*-instance. As independence systems can be used to model, for instance, the independence property in graphs, nowadays we have even a number of (other) NP-hardness proofs concerning Ext Ind Sys. For more discussions on extension problem variants of dependence and independence systems, we refer to Sec. 5 of [36].

[4] Here, $\mathcal{I}$ is not given as an explicit input, rather, it is assumed that membership in $\mathcal{I}$ can be efficiently detected by a suitable algorithm.

In order to enumerate all (inclusion-wise) minimal dominating sets of a given graph, Kanté et al. [28] studied the problem of deciding, given a subset $U \subseteq V$, if there exists a minimal dominating set D containing U (denoted by Ext DS here); sometimes, another set Y of vertices is given as input that should not intersect D. Mary proved in [36, Proposition 3.39] that Ext DS is NP-complete (on general graphs, with $Y = \emptyset$, as in our setting). Kanté et al. proved that the variation of Ext DS where also another set Y of vertices is given as input is NP-complete, even in special graph classes like split graphs, chordal graphs and line graphs [27, 28]. Further, in [3] it is shown that Ext DS remains NP-hard even when restricted to planar cubic graphs.

Recently, extension variants of the classical problems vertex cover (denoted Ext VC here) and independent set were studied in [11]. While these problems are NP-complete on planar bipartite sub-cubic graphs, they are polynomial-time decidable in chordal and in circular-arc graphs. Also, Ext VC remains NP-hard, even restricted to planar cubic graphs, see [2].

In [21], Khosravian et al. studied the following extension variant of the Connected Vertex Cover problem, denoted by Ext CVC: given a connected graph $G = (V, E)$ together with a subset $U \subseteq V$ of vertices, the goal is to decide whether there exists a minimal connected vertex cover of G containing U. It is shown that Ext CVC is polynomial-time decidable in chordal graphs while it is NP-complete on bipartite graphs of maximum degree 3 even if U is an independent set.

Extension variants of three edge graph problems, namely Edge Cover, Edge Matching and Edge Dominating Set, (here denoted by Ext EC, Ext EM and Ext EDS, respectively, and formally defined in Section 2) were studied in [10]; it is shown that all these problems are NP-hard in planar bipartite graphs of maximum degree 3. In all these problems, given a graph and a subset of its edges U, the task is to find, respectively, a minimal edge cover containing U, a maximal edge matching contained in U and a minimal edge dominating set containing U. Further, Ext EM is polynomial-time decidable when the forbidden edges $\overline{U} = E \setminus U$ form an induced matching.

In [11] and [10], extension variants of some classical graph problems were also studied from a parameterized complexity point of view, in particular under the parameterization by the cardinality of the fixed pre-solution U (standard), or by the cardinality of its complement, i. e., $|V| - |U|$ for $U \subseteq V$, and $|E| - |U|$ for $U \subseteq E$ (dual). A summary of these parameterized results is presented in Table 1. Further, the paper [11] contains some complexity lower bounds for extension problems assuming the Exponential Time Hypothesis (ETH)[5].

Admittedly, it is not that clear if the standard parameter $|U|$ or the dual parameter are small in practical situations. If one thinks of a traditional branching algorithm that should exactly solve a minimization problem, then U is typically very small at the beginning and grows when moving towards the leaves of the branching tree. Cutting off branches at an early stage would be highly desirable, but our results indicate that this is in fact difficult to achieve; see Tables 1 and 2. Conversely, close to the leaves of the branching tree, the dual parameter becomes relatively small, in particular, if the graph parameter that is considered could be relatively large, as it is the case, for instance, for Vertex Cover.

Param. \ Ext. of	EC	EM	EDS	IS	VC
standard	FPT	FPT	W[1]-hard	FPT	W[1]-complete
dual	FPT	FPT	FPT	W[1]-complete	FPT

Table 1. Summary of parameterized complexity results for extension problems from [11] and [10].

Finally, we also study an approximation question related to extension problems, called *price of extension*, formally described below. The idea is that, if a certain pre-solution U cannot be

[5] ETH is a conjecture implying that there is no $2^{o(n)}$ (i. e., no sub-exponential) algorithm for solving 3-SAT, where n is the number of variables; the number of clauses is somehow subsumed into this expression, as this number can be assumed to be linear in n (after applying the famous sparsification procedure); cf. [26].

extended to a minimal solution, then it would be good to find a pre-solution that is as close as possible to U and that can be extended. The usefulness of the price of extension could be best seen for algorithms that use a local search approach to explore the solution space. Then, if the algorithm has detected that its current pre-solution has led to a dead end, because it cannot be extended, some of the decisions that led to this pre-solution have to be undone, but this sort of back-tracking should keep as many as possible of these previous decisions.

Summary of results and organization. In an attempt to study the nature of extension tasks, we propose a general framework to express a broad class of what we refer to as extension problems. This framework is based on a partial order approach, reminiscent of what has been endeavored for *maximin* problems in [35]. In essence, we consider optimization problems in NPO with an additionally specified set of partial solutions which we call *pre-solutions* (including the set of solutions) and a partial order $\preceq$ on those. The partial order reflects not only the notion of *extension* but also of *minimality* as follows. For a pre-solution U and a solution S, S *extends* U if $U \preceq S$. A solution S is *minimal* if there exists no solution $S' \neq S$ with $S' \preceq S$. The resulting extension problem is formally defined as the task to decide, for a given pre-solution U, if there exists a minimal solution S which extends U. We give more detailed and formal definitions for our broad notion of extension together with some general properties in Section 2. This section also contains a list of specific examples of extension problems.

We further study some of these specific examples. In particular, we consider extension version of the dominating set problem (Ext DS) in Section 3, and extension versions of feedback vertex/edge set (Ext FVS / Ext FES), as well as their directed version (Ext DFVS/Ext DFES) in Section 4. An extension version of the bin packing problem (Ext BP) is discussed in Section 5. For the graph problems dominating set and feedback vertex set (on directed or undirected graphs), we consider extension versions that model gradually adding vertices to build up a solution. This results in the question if a given subset of vertices (a pre-solution U in our setting) is a subset of an inclusion minimal dominating set or feedback vertex set (a minimal solution S that extends U), respectively.Similarly, for feedback edge set (on directed or undirected graphs), we consider gradually adding edges, hence pre-solutions U are subsets of edges. For bin packing, we consider the strategy of starting with the whole ground set and then gradually splitting up sets that are still too large (for formal definitions of the resulting partial orderings and the according notion of minimality, see Section 2).

For the graph problems considered here, we discuss in particular the restrictions to specific graph classes. We consider restrictions to bipartite graphs, planar graphs, and bounded maximum degree. We show that Ext DS, Ext FVS, and Ext DFVS are NP-complete even when restricted to planar bipartite graphs of bounded degree. For Ext DFES we show NP-completeness for the restriction to bipartite graphs of bounded degree (without planarity). The respective bounds on the degree for these results are given in Table 2. In contrast, we show that Ext FES is solvable in polynomial time.

We also consider the parameterized complexity of these extension problems, where the parameter is the size of the given pre-solution. A summary of the respective results we obtain is also given in Table 2. Note that the dual parameterization (parameter $|V|-|U|$), as discussed in [11] and [10] and also stated in Table 1, is not particularly interesting for the graph problems we consider here, as this parameterization trivially gives membership in FPT, by guessing and checking the set that extends the pre-solution to a solution. We will state this observation in a more general terminology in Corollary 1 below. The dual parameterization for Ext BP is discussed at the end of Section 5.

Fixing notions. We use standard notations from graph theory. A graph can be specified as $G = (V, E)$ and for two vertices $u, v \in V$, we denote by uv and (u, v) an (undirected) edge and a directed edge, also called an arc, from vertex u to vertex v, respectively. We use $N[v]$ to denote in undirected graphs the closed neighborhood of v, i. e., $N[v] = \{w \in V \mid vw \in E\} \cup \{v\}$. Further, for $U \subseteq V$ we write $N[U] = \bigcup_{v \in U} N[v]$.

The *degree* of a vertex v in an undirected graph is its number of neighbors $|\{w \in V \mid vw \in E\}|$. For a vertex v in a directed graph, we separate the degree-notion and use *out-degree* for the number of arcs leaving v (i. e. $|\{(v, u) \mid (v, u) \in E\}|$) and *in-degree* for the number of arcs pointing to v

Ext. of / graph cl.	DS	FVS	DFVS	FES	DFES	BP
bipartite+planar	$\Delta \leq 3$	$\Delta \leq 6$	$\Delta \leq 3$	P	$\Delta \leq 4$ (non-planar)	-
param-compl	W[3]-complete	W[1]-hard	W[1]-hard	P	?	para-NP

Table 2. Summary of results where the first row (bipartite+planar), Δ gives the restriction of maximum degree that still gives NP-hardness for restriction to planar bipartite graphs (only bipartite graphs for Ext DFES). The second row (param-compl) gives the classification of the parameterized complexity w. r. t. standard parameterization (size of the pre-solution U).

(i. e. $|\{(u, v) \mid (u, v) \in E\}|$). A directed graph has *bounded degree* if both in-degree and out-degree of each vertex is bounded by a constant, where we list the specific bounds separately in according results.

For sets $U \subseteq V$, $G[U]$ denotes the subgraph induced by U. For $X \subseteq E$, $G[X]$ denotes the graph $(V(X), X)$ where $V(X)$ denotes the set of vertices incident to edges in X.

A *parameterized problem* is a decision problem specified together with a *parameter*, that is, an integer k depending on the instance. A problem is *fixed-parameter tractable* (FPT for short) if it can be solved in time $f(k) \cdot |I|^c$ (often briefly referred to as FPT-*time*) for an instance I of size $|I|$ with parameter k, where f is a computable function and c is a constant. An FPT-*reduction* between two parameterized problems P and Q is a function mapping an instance (I, k) of P to an instance (I', k') of Q such that $k' \leq g(k)$, the running time is $f(k)|I|^{O(1)}$, for some computable functions f and g, and where (I, k) is a *yes*-instance of P if and only if (I', k') is a *yes*-instance of Q.

If a parameterized problem P is C-hard for a parameterized complexity class C and there exists an FPT-reduction from P to a parameterized problem Q, then Q is also C-hard. We will encounter three particular parameterized complexity classes in this paper: W[1], W[3] and para-NP; see [16]. Hardness for any of these classes implies that it is unlikely to find an FPT algorithm under some complexity assumptions; we refer the reader to the textbooks [14, 16] for details.

A short version of this paper, together with some more examples, was presented at an invited paper at CIAC in 2021; see [12].

2 A General Framework of Extension Problems

In order to formally define our concept of minimal extension, we define what we call *monotone problems* which can be thought of as problems in NPO with the addition of a set of pre-solutions (which includes the set of solutions) together with a partial ordering on this new set. Formally, we define such monotone problems as 5-tuples $\Pi = (\mathcal{I}, presol, sol, \preceq, m)$ (where $\mathcal{I}, sol, m$ with an additional $goal \in \{\min, \max\}$ yields an NPO problem, see [25] for definitions of NPO):

- $\mathcal{I}$ is the set of instances, recognizable in polynomial time.
- For $I \in \mathcal{I}$, $presol(I)$ is the set of *pre-solutions* and, in a reasonable representation of instances and pre-solutions, the length of the encoding of any $y \in presol(I)$ is polynomially bounded in the length of the encoding of I.[6]
- For $I \in \mathcal{I}$, $sol(I)$ is the *set of solutions*, which is a subset of $presol(I)$.
- There exists an algorithm which, given (I, U), decides in polynomial time if $U \in presol(I)$; similarly there is an algorithm which decides in polynomial time if $U \in sol(I)$.
- For $I \in \mathcal{I}$, $\preceq$ is a partial ordering on $presol(I)$ and there exists an algorithm that, given an instance I and $U, U' \in presol(I)$, can decide in polynomial time if $U' \preceq U$.
- For each $I \in \mathcal{I}$, the set of solutions $sol(I)$ is upward closed with respect to $\preceq$, i. e., $U \in sol(I)$ implies $U' \in sol(I)$ for all $U, U' \in presol(I)$ with $U \preceq U'$.

[6] This condition is not only found in NPO problems, but also in the context of enumeration problems (*polynomially balanced*), see [42], leading to the class EnumP.

- m is a polynomial-time computable function which maps pairs (I, U) with $I \in \mathcal{I}$ and $U \in presol(I)$ to non-negative rational numbers; $m(I, U)$ is the *value of* U.
- For $I \in \mathcal{I}$, $m(I, \cdot)$ is *monotone* with respect to $\preceq$, meaning that the property $U' \preceq U$ for some $U, U' \in presol(I)$ either always implies $m(I, U') \leq m(I, U)$ or $m(I, U') \geq m(I, U)$.

Given a monotone problem $\Pi = (\mathcal{I}, presol, sol, \preceq, m)$, we denote by $\mu(sol(I))$ the set of *minimal feasible solutions of* I, formally given by

$$\mu(sol(I)) = \{S \in sol(I) \colon ((S' \preceq S) \wedge (S' \in sol(I))) \rightarrow S' = S\}.$$

Further, given $U \in presol(I)$, we define $ext(I, U) = \{U' \in \mu(sol(I)) \colon U \preceq U'\}$ to be the set of *extensions* of U. Sometimes, $ext(I, U) = \emptyset$, which makes the question of the existence of such extensions interesting. Hence, finally, the *extension problem* for Π, written $\textsc{Ext}\ \Pi$, is defined as follows: An instance of $\textsc{Ext}\ \Pi$ consists of an instance $I \in \mathcal{I}$ together with some $U \in presol(I)$, and the associated decision problems asks if $ext(I, U) \neq \emptyset$.

With these formal definitions, we try to capture aspects of extension that could be used to transfer properties among different specific extension problems. The requirement that the set of solutions is upward closed with respect to the partial ordering relates to *independence systems*, see [33]. This choice also models greedy strategies that attempt to build up solutions gradually by stepwise improvements towards feasibility. Note that such greedy approaches usually do not employ steps that transform a solution back into a pre-solution that is not feasible.

Adding the function m to the formal description of a monotone problem is on the one hand reminiscent of the problem class NPO, but it also allows to study approximate extension as follows. For a monotone problem Π one might ask for input (I, U) not to extend exactly U but a pre-solution as close as possible to U. Formally this yields the task to find a pre-solution $U' \in presol(I)$ with $U' \preceq U$ and $ext(I, U') \neq \emptyset$ that optimizes the value $m(I, U')$, where we choose maxization (minimization) if m is monotonically increasing (decreasing) w. r. t. $\preceq$, i. e., $U' \preceq U$ implies $m(I, U') \leq m(I, U)$ ($m(I, U') \geq m(I, U)$). Such an optimization formulation was studied for extension versions of the vertex cover and independent set problem under the notion *price of extension* in [11]. Further, the function m allows to discuss the parameterized complexity of extension problems, where we define the *standard parameter* for an extension problem $\textsc{Ext}\ \Pi$ for a monotone problem $\Pi = (\mathcal{I}, presol, sol, \preceq, m)$ with m mapping to integers, to be the value of the given pre-solution, i. e., the parameter for instance (I, U) of $\textsc{Ext}\ \Pi$ is $m(I, U)$. The *dual* parameterizations as discussed in [11] and [10] to derive the results summarized in Table 1, can be modeled as follows in this framework. The dual parameter is given by the difference between the value of the given pre-solution and the maximum $m_{max}(I) := \max\{m(I, y) \colon y \in presol(I)\}$, so the parameter for instance (I, U) of $\textsc{Ext}\ \Pi$ is $m_{max}(I) - m(I, U)$. In this case we say that Π *admits a dual parameterization*, and observe the following.

Corollary 1. *Let $\Pi = (\mathcal{I}, presol, sol, \preceq, m)$ be a monotone problem that admits a dual parameterization. If, for all $I \in \mathcal{I}$ and $U \in presol(I)$, the set $Above(U) = \{V \colon V \in sol(I), U \preceq V\}$ can be constructed in* FPT*-time, parameterized by $m_{max}(I) - m(I, U)$, then $\textsc{Ext}\ \Pi$ with dual parameterization is in* FPT.

In order to compute $Above(U)$, it is often easiest to actually list (if possible in FPT-time) the superset $\{V \colon V \in presol(I), U \preceq V\}$ instead and then filter this set by checking which of the listed pre-solutions are solutions, which can be done in polynomial time in our framework.

Although we strongly linked the definition of monotone problems to NPO, the corresponding extension problems do not generally belong to NP (in contrast to the canonical decision problems associated to NPO problems). Consider for example the following monotone problem $\Pi_\tau = (\mathcal{I}, presol, sol, \preceq, m)$ with:

- $\mathcal{I} = \{F \colon F \text{ is a Boolean formula}\}$.
- $presol(F) = sol(F) = \{\phi \mid \phi \colon \{1, \ldots, n\} \rightarrow \{0, 1\}\}$ for a formula $F \in \mathcal{I}$ on n variables.
- For $\phi, \psi \in presol(F)$, $\phi \preceq \psi$ if either $\phi = \psi$, or assigning variables according to ψ satisfies F while an assignment according to ϕ does not.

– $m \equiv 1$ (plays no role for the extension problem)

The associated extension problem Ext Π_τ corresponds to the co-NP-complete problem Tautology in the following way: Given a Boolean formula F which, w.l.o.g., is satisfied by the all-ones assignment $\psi_1 \equiv 1$, it follows that (F, ψ_1) is a *yes*-instance for Ext Π_τ if and only if F is a tautology, as ψ_1 is in $\mu(sol(F))$ if and only if there does not exist some $\psi_1 \neq \phi \in sol(F)$ with $\phi \preceq \psi_1$, so, by definition of the partial ordering, an assignment ϕ which does not satisfy F. Consequently, Ext Π_τ is not in NP, unless co-NP = NP.

Still, we can prove a general upper bound as follows. Recall that $\Sigma_1^p = \mathsf{NP}$ and that co-NP $\subseteq \Sigma_2^p$ in the usual terminology regarding the first levels of the polynomial-time hierarchy, see for example the book of Arora and Barak for more definitions [1].

Proposition 2. *If Π is a monotone problem, then* Ext Π *is in Σ_2^p.*

Proof. Deciding if (I, U) is a *yes*-instance of Ext Π can be done by evaluating the expression:

$$\exists U' \forall U'' (U' \in sol(I)) \wedge (U \preceq U') \wedge ((U'' \preceq U') \wedge (U'' \in sol(I)) \rightarrow (U'' = U')) \tag{1}$$

The number of bits required to express U' and U'' in the expression (1) above is polynomial in the encoding length of I, by our assumption on pre-solutions for monotone problems. Note also, that by the conditions of monotone problems, the tests for inclusion with respect to $\preceq$ and membership in $sol(I)$ can be performed in polynomial time. This explains membership in Σ_2^p. □

One of the consequences is that we cannot expect to obtain PSPACE-hard extension problems within our framework.

Under certain circumstances, there is a more efficient algorithm. To this end, consider the finer structure of the ordering $\preceq$ defined on $presol(I)$ for an instance I of Π. For $U, U' \in presol(I)$, call U' an *immediate predecessor* of U if $U' \preceq U$ and U' is a maximal element in $Below(U) = \{X \in presol(I) \colon X \preceq U \wedge X \neq U\}$, i. e., there exists no $U'' \neq U'$ with $U'' \in Below(U)$ and $U' \preceq U''$. We say that a monotone problem Π *admits polynomial computation of predecessors* if there exists a polynomial-time algorithm that, given any instance I of Π and $U \in presol(I)$, computes the set of all immediate predecessors of U in polynomial time.

Proposition 3. *If Π is a monotone problem that admits polynomial computation of predecessors, then* Ext Π *is in $\Sigma_1^p = \mathsf{NP}$.*

Proof. Given an instance (I, U) of Ext Π, we can perform the following steps.

1. Guess a solution U' of I.
2. Verify that $U \preceq U'$ holds, i. e., that U' is an extension of U.
3. Check for all immediate predecessor U'' of U', if $U'' \in sol(I)$.
4. If Step 4. does not find a predecessor with $U'' \neq U'$ and $U'' \in sol(I)$, then return U'.

By our assumptions, solutions are of polynomial size with respect to I, so that the first step means to guess a polynomial number of bits. Also, we can check if a guessed bitstring represents a solution U' in polynomial time. Furthermore, we can check in polynomial time if $U \preceq U'$ holds. Although all solutions of I will be exponentially many in general, they can be represented by bitstrings of polynomial size. As Π admits polynomial computation of predecessors and checking if $U'' \in sol(I)$ can also be done in polynomial time, Step 4 requires polynomial time.

For correctness, note that if there exists some $U'' \preceq U'$ that is a solution with $U'' \neq U'$ (disproving minimality of U') and if U'' is not an immediate predecessor of U', then there is a pre-solution $\hat{U}$ with $U'' \preceq \hat{U} \preceq U'$ that is an immediate predecessor of U'. As the set of solutions is upward closed, $\hat{U}$ is also a solution. Hence, minimality of a solution can be checked by cycling through all (polynomially many) immediate predecessors of U'. □

Some examples. Let us mention some well-known graph problems, that can quite naturally be modeled as monotone problems with $\mathcal{I}$ always as the set of undirected graphs, denoting instances by $G = (V, E)$, and the simple cardinality function as objective, i. e., $m(G, U) = |U|$ for all $U \in presol(G)$:

– Vertex Cover (VC): $\preceq = \subseteq$, $presol(G) = 2^V$, $C \in sol(G)$ iff each $e \in E$ is incident to at least one $v \in C$;
– Edge Cover (EC): $\preceq = \subseteq$, $presol(G) = 2^E$, $C \in sol(G)$ iff each $v \in V$ is incident to at least one $e \in C$;
– Independent Set (IS): $\preceq = \supseteq$, $presol(G) = 2^V$, $S \in sol(G)$ iff $G[S]$ contains no edges;
– Edge Matching (EM): $\preceq = \supseteq$, $presol(G) = 2^E$, $S \in sol(G)$ iff none of the vertices in V is incident to more than one edge in S;
– Dominating Set (DS): $\preceq = \subseteq$, $presol(G) = 2^V$, $D \in sol(G)$ iff $N[D] = V$;
– Edge Dominating Set (EDS): $\preceq = \subseteq$, $presol(G) = 2^E$, $D \in sol(G)$ iff each edge belongs to D or shares an endpoint with some $e \in D$.

When studying monotone graph problems restricted to some particular graph classes, this formally means that the instance set $\mathcal{I}$ contains only graphs that fall into the graph class under consideration. We hence arrive at problems like Ext VC (or Ext IS, resp.), where the instance is specified by a graph $G = (V, E)$ and a vertex set U, and the question is if there is some *minimal* vertex cover $C \supseteq U$ (or some *maximal* independent set $I \subseteq U$). Notice that the instance (G, V) of Ext IS can be solved by the exhaustive greedy approach that, starting from $\emptyset$, gradually adds vertices and deletes their closed neighborhood. Note that this gives an independent set U that trivially satisfies $U \subset V$ so $V \preceq U$. Further, for any $w \in V \setminus U$, the set $U \cup \{w\}$ is not an independent set by the construction of U, which means that there exists no independent set $U' \neq U$ with $U \subset U'$ (i. e., $U' \preceq U$), as $\{U \cup \{w\} \mid w \in V \setminus U\}$ is the set of immediate predecessors of U. Similarly, $(G, \emptyset)$ is an easy instance of Ext VC. We will show that this impression changes for other instances.

Notice that, as illustrated for Ext IS, each of the above monotone graph problems admits polynomial computation of predecessors. Therefore, the corresponding extension problems all lie in NP. It is instructive to have another look at the monotone problem Π_τ whose extension variant corresponds to Tautology. Here, the partial order $\preceq$ on $\{1, \dots, n\}^{\{0,1\}}$ can be also described as follows (with respect to a given Boolean formula F):

– All assignments that do not satisfy F are mutually incomparable, while
– each of them is strictly smaller (with respect to $\preceq$) than any assignments that satisfy F,
– which are again incomparable amongst themselves.

As a formula may possess exponentially many non-satisfying assignments, Π_τ does not admit polynomial computation of predecessors. In view of our earlier findings, this is a pre-requisite to prove co-NP-hardness of the extension variant.

So far, it might appear that every classical decision problem yields exactly one corresponding extension problem. However, different algorithmic (greedy) strategies for a classical problem result in different corresponding sets of pre-solutions and orderings, hence different extension problems. Consider for example the following two greedy strategies of finding a proper vertex coloring. Formally, vertex colorings of a graph $G = (V, E)$ are functions $c \colon V \to \{1, \dots, k\}$ for some $k \in \mathbb{N}$, and they are proper (hence a solution to the graph coloring problem) if $c(u) \neq c(v)$ for all edges $uv \in E$. Starting from a base coloring c that assigns the same color to all vertices, formally $c \colon V \to \{1\}$, consider the following two options as greedy improvement strategies for a coloring $c \colon V \to \{1, \dots, k\}$:

(a) Pick an index $i \in \{1, \dots, k\}$ and a subset C_i of $\{v \in V \mid c(v) = i\}$. Define the improved coloring c' on $\{1, \dots, k\}$ by $c'(v) = k + 1$ for all $v \in C_i$, and $c'(v) = c(v)$ for all $v \in V \setminus C_i$. *(split one color class into two)*
(b) Pick an independent set $C \subseteq V$ and define the improved coloring c' on $\{1, \dots, k\}$ by $c'(v) = k + 1$ for all $v \in C$, and $c'(v) = c(v)$ for all $v \in V \setminus C$. *(recolor an independent set)*

These two ideas, expressed as partial orderings towards and among feasible solutions, yield the partial orderings denoted *a-chromatic* and *b-chromatic* in [35], that can be expressed as the transitive closure of the following relations. Two colorings c_1, c_2 for a graph G satisfy $c_1 \preceq c_2$ if c_2 uses exactly one color more than c_1, i. e., $c_1 \colon V \to \{1, \ldots, k\}$ and $c_2 \colon V \to \{1, \ldots, k+1\}$ with the following conditions:

a-chromatic there exists a color i such that $c_1(v) \neq c_2(v)$ only for v with $c_1(v) = i$ and $c_2(v) = k+1$ (split one color into two)

b-chromatic $c_1(v) \neq c_2(v)$ only for v with $c_2(v) = k+1$ AND the color class $k+1$ forms an independent set (recolor an independent set)

The reader is also referred to [35] as a rich source of other examples for instance orderings.

Bin Packing. As a non-graph example for extension that fits within our framework, we will discuss in more detail in Section 5 an extension version of the bin packing problem. We consider as ordering the so-called partition ordering. Bin packing can be modeled as monotone problem as follows. Instances in $\mathcal{I}$ are sets $X = \{x_1, \ldots, x_n\}$ of items and a weight function w that associates rational numbers $w(x_i) \in (0, 1)$ to items, $presol(X)$ contains all partitions of X, and a partition π of X is in $sol(X)$, if for each set $Y \in \pi$, $\sum_{y \in Y} w(y) \leq 1$. For two partitions π_1, π_2 of X, we define the partial ordering $\preceq$ by $\pi_1 \preceq \pi_2$ iff π_2 is a refinement of π_1, i. e., π_2 can be obtained from π_1 by splitting up its sets into a larger number of smaller sets. The traditional aim of the bin packing problem is to find a feasible π such that $|\pi|$, the number of bins, is minimized, hence we set $m(X, \pi) = |\pi|$.

Notice that $\{X\}$ is the smallest partition with respect to $\preceq$. Clearly, the set of solutions is upward closed. Now, a solution is minimal if merging any two of its sets into a single set yields a partition π such that there is some $Y \in \pi$ with $w(Y) := \sum_{y \in Y} w(y) > 1$. Aside from modeling the greedy strategy that gradually splits up bins that are too large, fixing a pre-solution can be interpreted as encoding knowledge about which items should not be put together in one bin.

Synchronizing Words. In [18], extension variants of one of the most famous combinatorial problems in automata theory, namely the Synchronizing Word problem for deterministic finite automata (DFA), were considered. As this reveals certain interesting aspects of the general framework that we present, let us explain this in more detail. Recall that a deterministic finite semi-automaton $A = (S, \Sigma, \delta)$ (or DFSA) can be specified by its state alphabet S, its input alphabet Σ, and the total transition function $\delta : S \times \Sigma \to S$, which can be extended to $\delta^* \colon S \times \Sigma^* \to S$. A word $w \in \Sigma^*$ is *synchronizing* A if $|\{\delta^*(s, w) \mid s \in S\}| = 1$. The decision problem Synchronizing Word expects a DFSA A and an integer $k \geq 0$ and asks if there exists a synchronizing word for A of length at most k. There are quite a number of partial orders $\preceq$ on Σ^* that could be studied to model this as a monotone problem. From a search-space perspective, the following choices look most promising.

- $y \sqsubseteq w$, meaning that y is a *prefix* of w, i. e., there is a word z such that $w = yz$;
- $y \sqsupseteq w$, meaning that y is a *suffix* of w, i. e., there is a word x such that $w = xy$;
- $y \; sub \; w$, meaning that y is a *subword* of w, i. e., there are words x, z such that $w = xyz$;

This motivates to model extension problems Ext DFA-SW-$\preceq$, depending on the partial order $\preceq$ on input words as follows. Given a DFSA A and some word u over the input alphabet Σ, the question is if there exists a word w $\preceq$-extending u, i. e., with $u \preceq w$, such that w is minimal for the set of synchronizing words for A with respect to $\preceq$. The complexity picture of these problems is rather diverse, as the subsequence ordering $|$, the lexicographical ordering $\leq_{lex}$ (also of interest in a search-space perspective) and the length-lexicographical ordering $\leq_{ll}$ were also considered in [18].

- Ext DFA-SW-$\sqsubseteq$, Ext DFA-SW-$\sqsupseteq$ and Ext DFA-SW-$\leq_{lex}$ are solvable in polynomial time.
- The complexity status of Ext DFA-SW-sub is open.
- Ext DFA-SW-$|$ is NP-hard, even for several restricted classes of automata (see [8]).
- Ext DFA-SW-$\leq_{ll}$ is co-NP-hard, even when restricted to binary input alphabets.

This complexity diversity might also be reflected by the fact that these types of extension problems do not completely fit into the presented framework. Observe that none of the two natural picks of defining the set of pre-solutions to a DFSA A, namely either the set of all input words, or the set of all words synchronizing A, is polynomially bounded in the size of the input. But even if we impose some length bound on the set of words we care about, we face the problem that the solution space need not be upward closed. For instance, it could well be that $u|v$, $u \neq v$, u is synchronizing, but v is not. Only with $\sqsubseteq$, $\sqsupseteq$ and *sub*, this problem does not exist. Regarding the subsequence order $|$, interestingly the question if a word w synchronizing A is minimal is complete for co-NP.[7] This discussion might explain some of the difficulties that were found when considering extension models for synchronizing words. For further parameterized complexity results for Ext DFA-SW-$|$ in particular, we refer to [9].

3 Dominating Set

Recall that we defined for the dominating set problem a monotone problem version denoted DS$=(\mathcal{I}, presol, sol, \preceq, m)$ with $\mathcal{I}$ as the set of undirected graphs, denoting instances by $G = (V, E)$, $\preceq = \subseteq$, $presol(G) = 2^V$, $D \in sol(G)$ iff $N[D] = V$ and $m(G, U) = |U|$ for all $U \in presol(G)$.

We study the resulting extension problem Ext DS, formally defined as follows:

> Ext DS
> **Input:** A graph $G = (V, E)$, a set $U \subseteq V$ (i. e., $U \in presol(G)$) .
> **Question:** Is $ext(G, U) \neq \emptyset$?

Note that $ext(G, U) \neq \emptyset$ holds if and only if there is a minimal dominating set D (in the sense that $N[D'] \neq V$ for any $D' \subset D$) such that $U \subseteq D$. We in particular study the complexity of Ext DS when $\mathcal{I}$ is restricted to graphs of bounded degree with the additional properties bipartite and planar. Additionally, in this section we prove a W[3]-completeness result for the standard parameterization of Ext DS, a class rarely met when studying parameterized complexity.

We present a reduction from a variant of Satisfiability (SAT) named 4-Bounded Planar 3-Connected SAT (or 4P3C3SAT for short) that remains NP-hard by [32]. An instance $I = (C, X)$ of 4P3C3SAT is given by a set C of CNF clauses defined over a set X of Boolean variables such that each clause has exactly 3 literals, and such that each variable occurs in at most four clauses (at least one time negated and one time unnegated). The graph associated to instance $I = (C, X)$ is the *variable-clause graph* $VC = (C \cup X, E(VC))$ with $C = \{c_1, \ldots, c_m\}$, $X = \{x_1, \ldots, x_n\}$ and $E(VC) = \{c_j x_i \colon x_i \text{ or } \neg x_i \text{ is a literal of } c_j\}$. I is an instance of 4P3C3SAT if VC is planar and bipartite of maximum degree 4.

Theorem 4. Ext DS *is* NP*-complete for planar bipartite graphs of maximum degree* 3.

Proof. Membership in NP follows directly from Proposition 3; note that for any instance G of Ext DS, the set of immediate predecessors of any $U \in presol(G)$ is equal to $\{U \setminus \{v\} \mid v \in U\}$.

To show NP-hardness, we give a reduction from 4P3C3Sat. For an instance $I = (C, X)$ of 4P3C3Sat with clause set $C = \{c_1, \ldots, c_m\}$ and variable set $X = \{x_1, \ldots, x_n\}$, we build an instance (H, U) of Ext DS, where $H = (V_H, E_H)$ is a planar bipartite graph with maximum degree 3 and $U \subseteq V_H$.

Informally, in order to build H, we start from an embedding of the variable-clause graph $VC = (C \cup X, E(VC))$ with $C = \{c_1, \ldots, c_m\}$, $X = \{x_1, \ldots, x_n\}$ and $E(VC) = \{c_j x_i \colon x_i \text{ or } \neg x_i \text{ is a literal of } c_j\}$, then delete all edges of VC and replace the vertices of VC by subgraphs with inerconnections preserving planarity, where for replacing a variable vertex x_i the construction distinguishes how x_i appears negated.

Recall that by the definition of the problem 4P3C3Sat, any variable x_i appears in at most four clauses. Assume that x_i appears in the clauses c_1, c_2, c_3, c_4 of the original instance I such that in the induced (embedded) subgraph $G_i = G[\{x_i, c_1, c_2, c_3, c_4\}]$ an anti-clockwise ordering of edges

[7] shown by Petra Wolf (personal communication)

around x_i is given by c_1x_i, c_2x_i, c_3x_i, c_4x_i. By looking at G_i and considering the constellation in which x_i appears negated or non-negated in the four clauses c_1, c_2, c_3, c_4 in I, the construction distinguishes the following three cases:

- case 1: $x_i \in c_1, c_2$ and $\neg x_i \in c_3, c_4$,
- case 2: $x_i \in c_1, c_3$ and $\neg x_i \in c_2, c_4$,
- case 3: $x_i \in c_1, c_2, c_3$ and $\neg x_i \in c_4$.

Note that all other cases are included in these 3 cases by rotations and / or replacing $x_i(\neg x_i)$ with $\neg x_i(x_i)$.

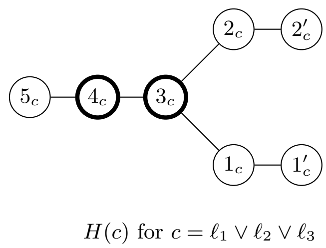

Fig. 1. The Gadget $H(c)$ for EXT DS. Vertices in pre-solution illustrated by their bold border.

Considering the above explanation, we build an instance (H, U) of EXT DS as follows:

- For each clause $c = \ell_1 \vee \ell_2 \vee \ell_3$, where ℓ_1, ℓ_2, ℓ_3 are literals, we introduce the subgraph $H(c) = (V_c, E_c)$ with 7 vertices and 6 edges as illustrated in Figure 1. The vertices $1'_c$ and $2'_c$ represent literals in clause c ($1'_c$ represents literals ℓ_1 and ℓ_2 while $2'_c$ represents ℓ_3) and the vertex set $U_c = \{3_c, 4_c\}$ is included in the pre-solution.
- For each of the three cases of variable x_i, we choose a different subgraph, denoted by $H(x_i)$, to replace x_i in VC to build H. The resulting three gadgets are illustrated in Figure 2, where the vertices among the subgraph $H(x_i)$ we put into the pre-solution, denoted U_{x_i}, are illustrated in black.
- We connect the subgraphs $H(x)$ and $H(c)$ by edges in the following way: for each clause c with literals ℓ_1, ℓ_2, ℓ_3, corresponding to variables x_1, x_2, x_3, respectively, connect $1'_c$ (representing ℓ_1 and ℓ_2) to the determined vertices in $H(x_1)$ and $H(x_2)$ and connect $2'_c$ (representing ℓ_3) to the determined vertex in $H(x_3)$ as illustrated in Figure 2 according to which of the three defined cases variable x_i complies with.
- At last, we set the pre-solution to $U = (\bigcup_{x_i \in X} U_{x_i}) \cup (\bigcup_{c_j \in C} U_{c_j})$.

Note that this construction can be build in polynomial time and that the resulting graph H is planar, bipartite and of maximum degree 3. To see that planarity is preserved, recall that we replaced vertices of the variable-clause graph by subgraphs that are interconnected in the same way as the previous vertices were. We claim that (H, U) is a *yes*-instance of EXT DS if and only if I has a satisfying assignment T.

Suppose T is a truth assignment of I which satisfies all clauses. We construct a dominating set S from U as follows:

- For each variable gadget $H(x_i)$ that complies with "case 1", add t_i (or f_i, respectively) to S if $T(x_i) = \texttt{true}$ (or $T(x_i) = \texttt{false}$, respectively).
- For each variable gadget $H(x_i)$ that complies with "case 2" add t_i^1, t_i^2 (or f_i^1, f_i^2, respectively) to S if $T(x_i) = \texttt{true}$ (or $T(x_i) = \texttt{false}$, respectively).
- For each variable gadget $H(x_i)$ that complies with "case 3" add t_i^1, t_i^2, m_i (or f_i, l_i^2, r_i^2, respectively) to S if $T(x_i) = \texttt{true}$ (or $T(x_i) = \texttt{false}$, respectively).

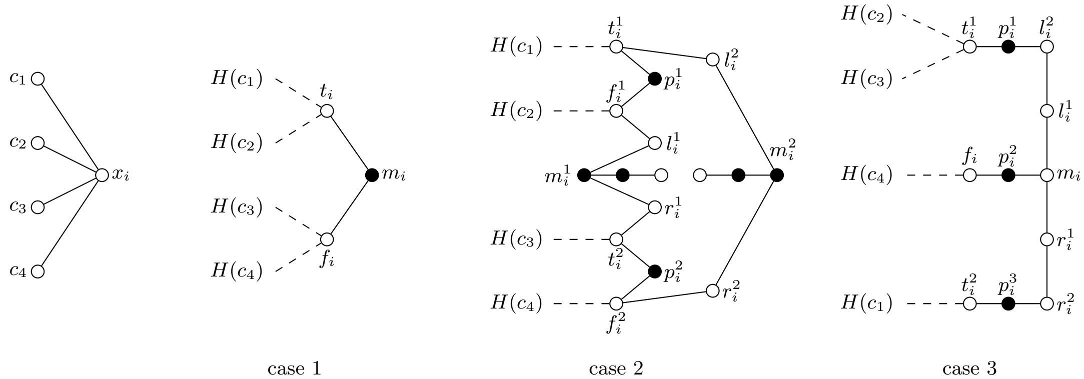

Fig. 2. Variable gadgets $H(x_i)$ of Theorem 4. On the left: A variable x_i appearing in four clauses c_1, c_2, c_3, c_4 in I. On the right, cases 1, 2, 3 are corresponding to $H(x_i)$, depending on how x_i appears (negated or non-negated) in the four clauses (Here case 3 is rotated). Black vertices denote elements of U_{x_i}. Crossing edges are marked with dashed lines. In case a variable only occurs in $\ell < 4$ clauses, there are no dashed edges to $H(c_t)$ for $4 \geq t > \ell$.

- For each clause $c \in C$, add vertex 1_c to S if $1'_c$ is not dominated by a variable vertex of S and add 2_c to S if $2'_c$ is not dominated by a variable vertex of S.

We claim that there exists a minimal dominating set $S' \subset S$ such that $U \subset S'$. To this end, we show that S is a dominating set of H and that for any $u \in U$, $S \setminus \{u\}$ is not a dominating set. Clearly $U \subseteq S$ by construction, because we start from U and add some vertices based on the above rules, which means that successively removing $v \in S \setminus U$ is still a dominating set will then result in the claimed minimal dominating set S' with $U \subseteq S'$.

Based on the constructions of $H(x)$ depicted in Figure 2, the vertex set U_{x_i} when variable x_i complies with "case 1" and "case 2" and the vertex set $U_x \cup \{l_i^2, r_i^2\}$ or $U_x \cup m_i$ when x_i complies with "case 3" is a minimal dominating set for the subgraph of the variable gadget $H(x)$. Further, observe that no neighbors of vertices in $H(x)$ outside of this gadget are picked to be in S. This means in particular that for any vertex $u \in U_x$, for some variable x, the set $S \setminus \{u\}$ is not a dominating set for H. Further, S is dominating for all vertices in the variable-gadgets. In each clause gadget $H(c)$, for some clause c, only the vertices $1'_c, 2'_c$ are not already dominated by U, and in the construction of S we include corresponding neighbors to ensure that S also dominates $1'_c$ and $2'_c$. Hence S is also a dominating set for $H(c)$. Since T is a satisfying assignment, for each clause gadget $H(c)$, at least one of $1'_c, 2'_c$ is dominated by a variable vertex of S. Thus, for each $H(c)$, at most one of the vertices 1_c and 2_c is added to S, consequently, $S \setminus \{3_c\}$ does not dominate either 1_c or 2_c. Further, $S \setminus \{4_c\}$ does not dominate 5_c. This shows that $S \setminus \{u\}$ is not a dominating set for $u \in U_c$. In summary, S is a dominating set for H and $S \setminus \{u\}$ is not a dominating set for any $u \in U$, which yields the existence of the claimed minimal dominating set S'.

Conversely, suppose S is a minimal dominating set of G with $U \subseteq S$. Because of minimality, $S \setminus \{3_c\}$ does not dominate either 1_c or 2_c. Hence, S contains at most one vertex in $\{1_c, 1'_c, 2_c, 2'_c\}$ for each clause gadget $H(c)$. In particular, there is at least one vertex among $\{1'_c, 2'_c\}$ which needs to be dominated by a literal vertex (a vertex in a variable gadget $H(x)$), thus, there is an assignment T which satisfies all clauses of I. We now show that T is a valid assignment, and in order to do this, we consider the three types of variable gadgets independently:

- If $H(x_i)$ complies with case 1, by minimality, S cannot contain both t_i, f_i, since otherwise $S \setminus \{m_i\}$ is also a dominating set. So we set $T(x_i) = \texttt{true}$ if $\{f_i\} \cap S = \emptyset$ and otherwise we set $T(x_i) = \texttt{false}$.
- If $H(x_i)$ complies with case 2, by minimality, S cannot contain both vertices in each pair $(t_i^1, f_i^1), (t_i^1, f_i^2), (t_i^2, f_i^1), (t_i^2, f_i^2)$, since otherwise, we can remove the vertices $p_i^1, m_i^2, m_i^1, p_i^2$ from S, respectively. So we set $T(x_i) = \texttt{true}$ if $S \cap \{f_i^1, f_i^2\} = \emptyset$ and otherwise we set $T(x_i) = \texttt{false}$.

- If $H(x_i)$ complies with case 3, by minimality, S cannot contain both vertices in each pair $(t_i^1, f_i), (t_i^2, f_i)$, since otherwise we can remove one of vertices in the pairs $(p_i^1, p_i^2), (p_i^1, p_i^3)$ from S, respectively. Note here that in order to dominate l_i^1 and r_i^2, $S \cap \{l_i^1, l_i^2, m_i\} \neq \emptyset$ and $S \cap \{m_i, r_i^1, r_i^2\} \neq \emptyset$. Hence, we set $T(x_i) = \texttt{true}$ if $S \cap \{f_i\} = \emptyset$ and otherwise, we set $T(x_i) = \texttt{false}$. $\square$

Ext DS gives an interesting hardness result in the framework of parameterized complexity with the size of pre-solution $|U|$ as parameter (our *standard* parameterization for extension problems); it happens to be one of the still rather few known problems that are complete for the class W[3], also cf. the discussion in [13]. To show this result, we give a reduction from an extension variant of Hitting Set to Ext DS. Minimum Hitting Set as an NPO problem is defined by $\mathcal{I}$ being the set of hypergraphs with instances denoted $I = (X, \mathcal{S})$ where X is a finite ground set and $\mathcal{S} = \{S_1, \ldots, S_m\}$ is a collection of sets $S_i \subseteq X$ (usually referred to as *hyperedges*), feasible solutions are subsets $H \subseteq X$ such that $H \cap S_i \neq \emptyset$ for all $i \in \{1, \ldots, m\}$ and $m(I, H) = |H|$. This can be seen as a monotone problem with $presol(I) = 2^X$ and $\preceq = \subseteq$. In [4], the extension problem associated to this monotone formulation, in the following referred to as Ext HS, appears as a subproblem for the enumeration of minimal hitting sets in lexicographical order, and Ext HS parameterized by $m(I, U) = |U|$ is shown to be W[3]-complete. By a slight adjustment of the classical reduction from the Hitting Set problem to Dominating Set, this result transfers and formally yields:

Theorem 5. Ext DS *with standard parameter is* W[3]*-complete, even when restricted to bipartite instances.*

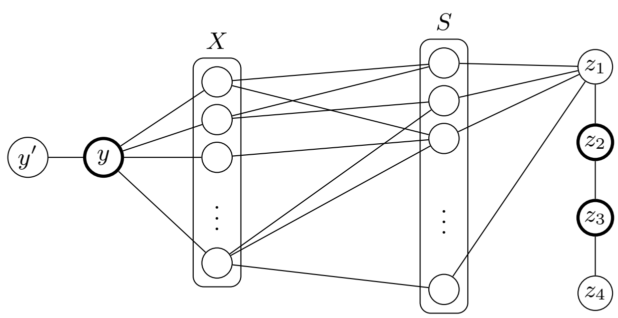

Fig. 3. The graph $G = (V, E)$ for Ext DS, vertices in the pre-solution set U' are drawn bold.

Proof. Ext DS can obviously be modeled as special case of Ext HS by interpreting the closed neighborhoods as subsets of the ground set of vertices. This immediately gives membership in W[3] for Ext DS.

Conversely, given an instance (I, U) with $I = (X, \mathcal{S})$, $\mathcal{S} = \{S_1, \ldots, S_m\}$ for Ext HS we create a graph for the corresponding instance for Ext DS as follows (This construction is also illustrated in Figure 3.):

- Start with the bipartite graph on vertices $X \cup \{s_1, \ldots, s_m\}$ containing edges xs_i if and only if $x \in S_i$.
- Add two new vertices y, y' with edges $y'y$ and xy for all $x \in X$.
- Add four new vertices z_1, z_2, z_3, z_4 with edges z_1z_2, z_2z_3, z_3z_4 and z_1s_i for all $1 \leq i \leq m$.

Let $G = (V, E)$ denote the graph created in this way, and observe that G is bipartite. With the set U' containing the vertex y to dominate X, z_2 and z_3 to forbid including any vertex s_i in the extension (as this would make z_2 obsolete) and the vertices corresponding to the pre-solution U for Ext HS, it is not hard to see that (G, U') is a *yes*-instance for Ext DS if and only if (I, U) is a *yes*-instance for Ext HS. As the parameters relate by $|U'| = |U| + 3$, this reduction transfers the W[3]-hardness of Ext HS to Ext DS on bipartite graphs. $\square$

4 Feedback Vertex/Edge Set

A *feedback vertex (edge) set* in a graph $G = (V, E)$ is a subset S of vertices (F of edges) such that $G[V \setminus S]$ (the graph $(V, E \setminus F)$) is acyclic. These two problems are studied both for directed and undirected graphs; in case of the problem feedback edge set, the variant for directed graphs is often called *feedback arc set*.

In [19], it is shown that it is NP-hard to find a feedback vertex set (in directed or undirected graphs) or a feedback arc set of minimum size for a given graph. Computing a minimum size feedback edge set, however, is equivalent to finding a spanning tree of maximum size for a given edge weighted graph; a problem that can be solved in polynomial time (see [20] for details). With respect to restricted graph classes, it is known that it is NP-hard to find a minimum feedback arc set in graphs of maximum in-degree and out-degree 3. This is somewhat in contrast to the situation in the undirected case, as minimum feedback edge sets can be found in polynomial time anyways, but even minimum feedback vertex sets can be determined in polynomial time in (sub-)cubic graphs; see [41].

We will use FVS and FES to denote the problems feedback vertex set and feedback edge set (both on undirected graphs), respectively, and we use DFVS and DFES to denote the versions on directed graphs. We consider corresponding monotone problem versions, defined as follows.

FVS $= (\mathcal{I}, presol, sol, \preceq, m)$, with $\mathcal{I}$ being the set of undirected graphs denoting instances by $G = (V, E)$, $presol(G) = 2^V$, $D \in sol(G)$ iff $G[V \setminus D]$ is acyclic, $\preceq\, = \,\subseteq$ and $m(G, U) = |U|$ for all $U \in presol(G)$. DFVS is defined like FVS, except that $\mathcal{I}$ is the set of directed graphs, where we denote instances by $G = (V, A)$.

This yields the following extension versions:

EXT FVS **Input:** A graph $G = (V, E)$ and $U \in presol(G)$ (i. e., $U \subseteq V$). **Question:** Is $ext(G, U) \neq \emptyset$? (Does G have a minimal feedback vertex set S with $U \subseteq S$?)

EXT DFVS **Input:** A directed graph $G = (V, A)$ and $U \in presol(G)$ (i. e., $U \subseteq V$). **Question:** Is $ext(G, U) \neq \emptyset$? (Does G have a minimal feedback vertex set S with $U \subseteq S$?)

For feedback edge set, we define the monotone problem FES $= (\mathcal{I}, presol, sol, \preceq, m)$ with $\mathcal{I}$ being the set of undirected graphs, $presol(G) = 2^E$, $D \in sol(G)$ iff $G[E \setminus D]$ is acyclic, $\preceq\, = \,\subseteq$ and $m(G, U) = |U|$ for all $U \in presol(G)$. DFES (sometimes denoted FAS in the literature) is defined like FES, except that $\mathcal{I}$ is the set of directed graphs $G = (V, A)$ (consequently, $presol(G)$ and also $sol(G)$ contains subsets of the arcs A). As corresponding extension problems, this gives the following.

EXT FES **Input:** A graph $G = (V, E)$ and $U \in presol(G)$ (i. e., $U \subseteq E$). **Question:** Is $ext(G, U) \neq \emptyset$? (Does G have a minimal feedback edge set S with $S \supseteq U$?)

EXT DFES **Input:** A graph $G = (V, A)$ and $U \in presol(G)$ (i. e., $U \subseteq A$). **Question:** Is $ext(G, U) \neq \emptyset$? (Does G have a minimal feedback arc set S with $S \supseteq U$?)

We start by looking at the extension problems EXT FVS, EXT DFVS and EXT DFES and show that they are all NP-hard in graphs of bounded degree. For these results, we exploit the relationships highlighted by our framework and reduce from another extension problem. More precisely, we reduce from EXT VC, the extension version of the VERTEX COVER problem as defined as monotone problem above. EXT VC was studied in [11] and shown to be NP-complete even when restricted to planar bipartite graphs of maximum degree 3.

Theorem 6. EXT FVS *is* NP-*complete in planar bipartite graphs of maximum degree* 6.

Proof. Membership in NP follows from Proposition 3; note that for any instance G of Ext FVS, the set of immediate predecessors of any $U \in \mathit{presol}(G)$ is equal to $\{U \setminus \{v\} \mid v \in U\}$.

For NP-hardness, we give a reduction from Ext VC. Let (G, U) be an instance of Ext VC, where $G = (V, E)$ is a bipartite graph of maximum degree 3 and $U \subseteq V$ is the pre-solution. We construct a new graph $G' = (V', E')$ by adding the gadget $H(e) = (V_e, E_e)$ containing 4 new vertices $v_1^e, v_2^e, v_3^e, v_4^e$ and 6 new edges $xv_1^e, yv_2^e, v_1^e v_2^e, v_2^e v_3^e, v_3^e v_4^e, v_4^e v_1^e$ for each edge $e = xy \in E$ to G. Formally:

- $V' = V \cup \{v_1^e, v_2^e, v_3^e, v_4^e \colon e \in E\}$,
- $E' = E \cup \{xv_1^e, yv_2^e, v_1^e v_2^e, v_2^e v_3^e, v_3^e v_4^e, v_4^e v_1^e \colon e = xy \in E\}$.

An illustration of the gadget $H(e)$ for $e = xy$ is given in Figure 4. Observe that G' is a planar bipartite graph of maximum degree 6 and can be constructed from G in polynomial time.

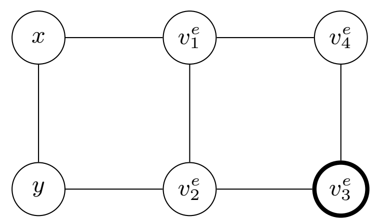

Fig. 4. Gadget $H(e)$ for $e = xy$. The vertex v_3^e is in the pre-solution U'.

We claim that (G, U) is a *yes*-instance of Ext VC if and only if (G', U') with $U' = U \cup \{v_3^e : e \in E\}$ is a *yes*-instance of Ext FVS. Suppose (G, U) is a *yes*-instance of Ext VC, i.e., there exists a minimal vertex cover S with $U \subseteq S$. We claim that $S' = S \cup \{v_3^e \colon e \in E\}$ is a minimal feedback vertex set of G' with $U' \subseteq S'$. For this, we prove separately that: S' is a solution, S' is minimal, and S' contains U':

- $S' \in \mathit{sol}(G')$ (S' is a solution): Assume on the contrary, that $G'[V \setminus S']$ still contains a cycle C. Since S is a vertex cover for G, it follows that C contains no edges from E. Further, all cycles containing only vertices from $V' \setminus V$ are covered by $\{v_3^e : e \in E\} \subseteq S'$. Hence, C has to contain vertices from both V and $V' \setminus V$. Any such cycle in G' that does not include a vertex from U' has to traverse a path of the form $x - v_1^e - v_2^e - y$ for some edge $e = xy \in E$, which means that $x, y \notin S$ for an edge $xy \in E$, hence a contradiction to S being a vertex cover for G.
- $S'' \in \mathit{sol}(G')$ with $S'' \subseteq S'$ implies $S'' = S'$ (S' is minimal): It suffices to show that each immediate predecessor $S'' = S' \setminus \{v\}$, $v \in S'$, is not a feedback vertex set for G'. For $v = v_3^e \in U'$ for some $e \in E$, it follows that $G'[V \setminus S'']$ contains the 4-cycle $(v_1^e, v_2^e, v_3^e, v_4^e)$, and so S'' is not in $\mathit{sol}(G')$. For $v \in S$ (i.e., $v \in U'$ with $v \neq v_3^e$ for some $e \in E$), the minimality of S implies that $S \setminus \{v\}$ is not a vertex cover in G, i.e., $G[(V \setminus S) \cup \{v\}]$ contains at least one edge e. Consequently, e is also an edge in $G'[V \setminus S'']$ (note that this is a supergraph of $G[(V \setminus S) \cup \{v\}]$). With $e = xy$, it follows that the vertices x, y, v_1^e, v_2^e form a 4-cycle in $G'[V \setminus S'']$, hence $S'' \notin \mathit{sol}(G')$.
- $U' \subseteq S'$: This follows immediately from $U \subseteq S$ and from the definitions $U' = U \cup \{v_3^e : e \in E\}$ and $S' = S \cup \{v_3^e : e \in E\}$.

Conversely, suppose that (G', U') is a *yes*-instance of Ext FVS, so there exists a minimal feedback vertex set S' for G' with $U' \subseteq S'$. We claim that $S = V \cap S'$ is a minimal vertex cover for G' with $U \subseteq S$. Again we prove the three properties of being a solution, minimality and extension separately, with the following property derived from the minimality of U'. By the requirement $\{v_3^e : e \in E\} \subseteq U' \subseteq S'$, minimality requires that $U' \setminus \{v_3^e\}$ is not a feedback vertex set. Hence for every edge e there has to be a cycle C such that $C \cap U' = \{v_3^e\}$ (a cycle uniquely covered by v_3^e). Looking at the structure of G', this implies that $\{v_1^e, v_2^e : e \in E\} \cap S' = \emptyset$, since all cycles containing v_3^e for some $e \in E$ contain both v_1^e and v_2^e.

- $S \in sol(G)$: For each $e = xy \in E$, it follows that either x or y has to be in S' to cover the cycle built from x, y, v_1^e, v_2^e, which means that S is a vertex cover for G.
- $S^* \in sol(G)$ with $S^* \subseteq S$ implies $S^* = S$: Assume towards contradiction that $S^* = S \setminus \{v\}$ for some $v \in S$ is a vertex cover for G. By the forward direction of the proof, it follows that $S'' = S^* \cup \{v_3^e : e \in E\}$ is a feedback vertex set for G', contradicting the minimality of S', since $S'' \subset S \cup \{v_3^e : e \in E\} \subset S'$ and $v \in S' \setminus S''$, so $S' \neq S''$.
- $U \subseteq S$ follows directly from $U \subseteq U'$ and $U' \subseteq S'$. □

Theorem 7. Ext DFVS *is* NP*-complete in planar bipartite graphs of maximum in-degree and out-degree* 3.

Proof. Membership in NP follows again directly from Proposition 3. To show NP-hardness, we give a reduction from Ext VC adapted from Karp's reduction [29]. Let (G, U) with $G = (V, E)$ and $U \subseteq V$ be an instance of Ext VC. We transform G into a digraph $G' = (V, A)$, with the same vertices as G and with the two arcs (u, v) and (v, u) in A for each edge $uv \in E$. Note that G' is bipartite of maximum in-degree and out-degree 3 if G is bipartite of maximum degree 3.

We claim that (G, U) is a *yes*-instance for Ext VC if and only if (G', U) is a *yes*-instance for Ext DFVS. Suppose (G, U) is a *yes*-instance for Ext VC, so there exists a minimal vertex cover S for G with $U \subseteq S$. We claim that $S' = S$ is a minimal feedback vertex set for G':

- $S' \in sol(G')$: Since S is a vertex cover in G, it follows that $G[V \setminus S]$ is edgeless, which by the definition of G' makes $G'[V \setminus S]$ arcless, so in particular acyclic.
- $S'' \in sol(G')$ with $S'' \subseteq S'$ implies $S'' = S'$: Assume towards contradiction that $S'' = S' \setminus \{v\}$ for some $v \in S'$ is a feedback vertex set for G'. By the structure of G', this means that $u \in S'$ for all $u \in V$ with $uv \in E$, since otherwise $G'[V \setminus S'']$ contains the cycle formed by the arcs $(u, v), (v, u) \in A$. This however means that $S \setminus \{v\}$ is also a vertex cover for G, a contradiction to the minimality of S.
- $U \subseteq S'$ follows directly from $U \subseteq S$ and $S' = S$.

Suppose (G, U') is a *yes*-instance for Ext DFVS, so there exists a minimal feedback vertex set S' for G' with $U \subseteq S'$. We claim that $S = S'$ is a minimal vertex cover for G:

- $S \in sol(G)$: For each $e = xy \in E$, it follows that either x or y has to be in S' to cover the cycle built from the arcs $(x, y), (y, x) \in A$, which means that S is a vertex cover for G.
- $S^* \in sol(G)$ with $S^* \subseteq S$ implies $S^* = S$: Assume towards contradiction that $S^* = S \setminus \{v\}$ for some $v \in S$ is a vertex cover for G. By the forward direction of the proof, it follows that S^* is also a feedback vertex set for G', contradicting the minimality of S'.
- $U \subseteq S$ follows directly from $U \subseteq S'$ and $S = S'$. □

Theorem 8. Ext DFES *is* NP*-complete in bipartite graphs of maximum in-degree and out-degree* 4.

Proof. Membership in NP follows again directly from Proposition 3. To show NP-hardness, we give a reduction from Ext VC adapted from the reduction given in [29]. Let (G, U) with $G = (V, E)$ and $U \subseteq V$ be an instance of Ext VC, we transform G into a digraph $G_A = (V_A, A)$ as follows. We build the vertex set $V_A = V \cup V'$ where $V' = \{v' : v \in V\}$ is a copy of V, and the arc set $A = A' \cup A''$ where $A' = \{(v, v') : v \in V\}$ and $A'' = \{(u, v'), (v', u), (v, u'), (u', v) : e = uv \in E\}$. An illustration of this reduction for an edge $e = uv$ in depicted in Figure 5. Finally, we define the pre-solution as $U_A = \{(u, v'), (v, u') : e = uv \in E\} \cup \{(u, u') : u \in U\}$. Obviously, G_A is bipartite of maximum in-degree and out-degree 4 if G is bipartite of maximum degree 3.

We claim that (G, U) is a *yes*-instance of Ext VC if and only if (G_A, U_A) is a *yes*-instance of Ext DFES. Observe that $(V_A, A \setminus U_A)$ only contains edges of the form (v', u) where $v' \in V'$ and $u \in V$ and $vu \in E$, or $(u, u') \in V \times V'$. Hence, all cycles in $(V_A, A \setminus U_A)$ alternate between vertices in V and V', and the arcs alternate between edge-arcs ((v', u) with $vu \in E$) and vertex-arcs ($(u, u') \in V \times V'$).

Suppose (G, U) is a *yes*-instance of Ext VC, i. e., there exists a minimal vertex cover S with $U \subseteq S$. We claim that $S' = \{(u, u') : u \in S\} \cup U_A$ is a minimal directed feedback edge set of G_A with $U_A \subseteq S'$:

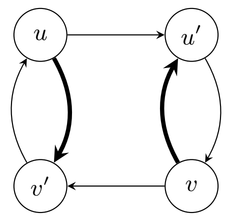

Fig. 5. Example of construction of G_A if G is a single edge $e = uv$. Arcs drawn in bold are in U_A.

- $S' \in sol(G)$: Suppose towards contradiction that $(V_A, A \setminus S')$ contains a cycle. By the structure of $(V_A, A \setminus U_A)$ (which is a supergraph of $(V_A, A \setminus S')$), it follows that the cycle is of length at least 4 and contains a sequence of vertices of the form u, u', v, v', where $uv \in E$ (recall the limited types of edges). This means that $(u, u'), (v, v') \notin S'$, hence $u, v \notin S$ for $uv \in E$, which contradicts S being a vertex cover for G.
- $S'' \in sol(G_A)$ with $S'' \subseteq S'$ implies $S'' = S'$: Again, it suffices to show that $S'' = S' \setminus \{a\}$ for every arc $a \in A$ is not a feedback arc set for G'. For $a = (u, v')$ with $(u, v') \in U_A$, the 2-cycle built with the arc $(v', u) \in A \setminus S'$ shows that $S'' = S' \setminus \{a\} \notin sol(G_A)$. For $a = (u, u') \in S'$, it follows by construction of S' that $u \in S$ and the minimality of S implies that $G[(V \setminus S) \cup \{v\}]$ contains at least one edge e. Consequently, for $e = uv$ it follows that $v \notin S$ and hence $(v, v') \notin S'$, and further $(v', u), (u', v) \in A$. Since S' contains no arcs of the form (u', v) with $v \neq u$, the 4-cycle on the vertices u, u', v, v' is contained in $(V_A, A \setminus S'')$, hence $S'' \notin sol(G_A)$.
- $U_A \subseteq S'$: This follows from setting $S' = \{(u, u') \colon u \in S\} \cup U_A$.

Conversely, suppose that (G_A, U_A) is a *yes*-instance of Ext DFES, so there exists a minimal feedback vertex set S' for G' with $U_A \subseteq S'$. We claim that $S = \{u \in V \mid (u, u') \in S'\}$ is a minimal vertex cover for G' with $U \subseteq S$.

- $S \in sol(G)$: Assume towards contradiction that there is an edge $e = uv \in E$ such that $u, v \notin S$, which means $(u, u'), (v, v') \notin S'$. Since u, u', v, v' is a 4-cycle in G_A, it follows that at least one of the arcs $(u', v), (v', u)$ has to be in S', so assume $(u', v) \in S'$. By minimality of S', it follows that for the arc $(v, u') \in S'$, $S' \setminus \{(v, u')\}$ is not a feedback arc set. Hence there exists a path p_1 in $(V_A, A \setminus S')$ from u' to v. Since $(u, u'), (v, v') \notin S'$, it follows that also $(v', u) \in S'$, since otherwise u, u', p_1, v, v' builds a cycle in $(V_A, A \setminus S')$. Minimality again implies that also $S' \setminus \{(u, v')\}$ is not a feedback arc set, which yields a path p_2 in $(V_A, A \setminus S')$ from v' to u. This however gives a cycle from u to u' via p_1 to v to v' and via p_2 back to u (note that if p_1 and p_2 share arcs, the described cycle can be cut to be simple), which is a contradiction to S' being a feedback arc set.
- $S^* \in sol(G)$ with $S^* \subseteq S$ implies $S^* = S$: Assume towards contradiction that $S^* = S \setminus \{v\}$ for some $v \in S$ is a vertex cover for G. Since $v \in S$ we know that $(v, v') \in S'$, and by minimality there exists a path p from v' to v in $(V_A, A \setminus S')$. By the structure of the paths in $(V_A, A \setminus U_A)$, this path p has to start with an edge-arc (v', u) for some u with $uv \in E$, followed by the only outgoing arc from u which is the vertex-arc (u, u'). This means that $(u, u') \notin S'$ and hence $u \notin S$ for a vertex u with $uv \in E$, a contradiction to S^* being a vertex cover for G, so $S^* \notin sol(G)$.
- $U \subseteq S$ follows from the definitions $U_A = \{(u, v'), (v, u') \colon e = uv \in E\} \cup \{(u, u') \colon u \in U\}$ and $S = \{u \in V \mid (u, u') \in S'\}$. ☐

Note that our construction destroys the planarity of the input graph. Since we were not able to mend this by a different construction, we leave the complexity of Ext DFES restricted to planar graphs as an open problem. From our failed attempts to show hardness, it seems that Ext VC is not a good choice as problem to reduce from. Also, the fact that feedback arc set is solvable in polynomial time when restricted to planar graphs [34], might indicate that the same is true for the extension version Ext DFES.

With respect to parameterized complexity, [11] also shows that Ext VC with standard parameterization is W[1]-complete. The reduction given to prove Theorem 6 does not transfer this parameterized hardness, since the value of the pre-solution of the constructed instance for Ext FVS, and hence the parameter, is $|U| + |E|$. However, we can give a different construction that transfers the $\mathsf{W}[1]$-hardness of Ext VC to Ext FVS:

Theorem 9. Ext FVS *with standard parameterization is* $\mathsf{W}[1]$*-hard.*

Proof. We give a reduction from Ext VC to Ext FVS. Let (G, U) be an instance of Ext VC where $U \subseteq V$ is the pre-solution. We construct a new graph $G' = (V', E')$ by adding a universal vertex u to G and attaching a triangle on two new vertices to u, formally

- $V' = V \cup \{u, u_1, u_2\}$,
- $E' = E \cup \{uv \mid v \in V\} \cup \{uu_1, uu_2, u_1u_2\}$.

We claim that (G, U) is a *yes*-instance of Ext VC if and only if (G', U') with $U' = U \cup \{u_1\}$ is a *yes*-instance of Ext FVS. Suppose (G, U) is a *yes*-instance of Ext VC, i. e., there exists a minimal vertex cover S with $U \subseteq S$. We claim that $S' = S \cup \{u_1\}$ is a minimal feedback vertex set of G' with $U' \subseteq S'$:

- $S' \in sol(G')$ (S' is a solution): Since S is a vertex cover for G, it follows that $G[V \setminus S]$ is edgeless, hence $G'[V \setminus S']$ is a star with center u, so in particular acyclic.
- $S'' \in sol(G')$ with $S'' \subseteq S'$ implies $S'' = S'$ (S' is minimal): Again, it suffices to show that $S'' = S' \setminus \{v\}$ for every $v \in S'$ is not a feedback vertex set for G'. For $v = u_1$, $G'[V \setminus S'']$ contains the triangle formed by u, u_1, u_2, so $S'' \notin sol(G')$. For $v \in S$, the minimality of S implies that $S \setminus \{v\}$ is not a vertex cover in G, i. e., $G[(V \setminus S) \cup \{v\}]$ contains at least one edge e. Consequently, e is also an edge in $G'[V \setminus S'']$. With $e = xy$, it follows that the vertices x, y, u form a triangle in $G'[V \setminus S'']$, hence $S'' \notin sol(G')$.
- $U' \subseteq S'$: This follows immediately from $U \subseteq S$ and the definitions $U' = U \cup \{u_1\}$ and $S' = S \cup \{u_1\}$.

Conversely, suppose that (G', U') is a *yes*-instance of Ext FVS, so there exists a minimal feedback vertex set S' for G' with $U' \subseteq S'$. We claim that $S = V \cap S'$ is a minimal vertex cover for G' with $U \subseteq S$. Note that minimality of S' implies that $S = S' \setminus \{u_1\}$, since the only cycle covered by u_1 is the triangle among the vertices u, u_1, u_2. This means that S satisfies:

- $S \in sol(G)$: For each $e = xy \in E$, it follows that either x or y has to be in S' to cover the triangle built from x, y, u, which means that S is a vertex cover for G.
- $S^* \in sol(G)$ with $S^* \subseteq S$ implies $S^* = S$: Assume towards contradiction that $S^* = S \setminus \{v\}$ for some $v \in S$ is a vertex cover for G. By the forward direction of the proof, it follows that $S'' = S^* \cup \{u_1\}$ is a feedback vertex set for G', contradicting the minimality of S', since $S'' \subset S \cup \{u_1\} \subset S'$ and $v \in S' \setminus S''$, so $S' \neq S''$.
- $U \subseteq S$ follows directly from $U \subseteq U'$ and $U' \subseteq S'$.

Finally, note that this construction is a parameterized reduction as the parameter of the constructed instance $m(G', U')$ satisfies $m(G', U') = m(G, U) + 1$. This overall transfers the $\mathsf{W}[1]$-hardness of Ext VC to Ext FVS. ◻

Observe that we cannot use the construction of Theorem 9 for the NP-hardness result of Theorem 6, since the universal vertex destroys both the degree bound and planarity.

Theorem 7 is proven by a parameterized reduction (the pre-solution U for Ext VC remains the pre-solution for Ext DFVS, which gives the same parameter value). Hence we can conclude:

Corollary 10. Ext DFVS *with standard parameterization is* $\mathsf{W}[1]$*-hard.*

The construction used to prove Theorem 8 is again not a parameterized reduction, as it blows up the parameter. We leave the classification of the parameterized complexity of Ext DFES with standard parameterization as an open problem. Further, Theorem 9 and Corollary 10 only show hardness for the class $\mathsf{W}[1]$ while we are not able to show a corresponding membership result. The full classification of the parameterized complexity of Ext FVS and Ext DFVS with standard parameterization hence also remains open. However, as we can list all supersets of a given set $U \subseteq X$ in time $\mathcal{O}(2^{|X|-|U|})$, we can easily conclude with Corollary 1:

Corollary 11. Ext FVS*,* Ext DFVS*, and* Ext DFES *with dual parameter are in* FPT*.*

At last, we have not yet considered the problem Ext FES. This problem turns out to be easy, and it might be tempting to think that this simply follows from the polynomial solvability of the related decision problem. Recall that there are numerous examples of polynomial-time solvable decision problems, for which a corresponding monotone formulation results in an NP-hard extension problem, e. g., Ext EM (an extension version of the maximum matching problem) is shown to be NP-hard in [10], and also Ext VC restricted to bipartite graphs, shown to be NP-hard in [11]. For our monotone version of the feedback edge set problem, the nice properties of spanning trees in graphs however quickly give the following.

Theorem 12. Ext FES *is solvable in polynomial time.*

Proof. Note that a feedback edge set S of a given connected graph $G=(V,E)$ is minimal if and only if the graph $(V,E\setminus S)$ is a tree. Hence, for a given connected graph $G=(V,E)$ and a pre-solution $U\subseteq E$, there is a minimal feedback edge set of G containing U if and only if the graph $(V,E\setminus U)$ is connected.

Generally, let (G,U) be an instance of Ext FES where $G=(V,E)$ and $U\subseteq E$, then the answer is *yes* if and only if the number of connected components of the graph $(V,E\setminus U)$ is equal to the number of connected components of G. □

5 Bin Packing

In this section we study the extension version of the bin packing problem Ext BP. Recall our definition introduced in Section 2: The monotone problem BP$=(\mathcal{I},\mathit{presol},\mathit{sol},\preceq,m)$ is given by $\mathcal{I}$ being sets X of items with associated weights given by a function $w\colon X\to(0,1)$, $\mathit{presol}(X)$ contains all partitions of X and such a partition $\pi\in\mathit{presol}(X)$ is in $\mathit{sol}(X)$ if $\sum_{y\in Y}w(y)\leq 1$ for each set $Y\in\pi$. For the partial ordering $\preceq$, two pre-solutions $\pi_1,\pi_2\in\mathit{presol}(X)$ satisfy $\pi_1\preceq\pi_2$ iff π_2 is a refinement of π_1. At last, $m(X,\pi)=|\pi|$, where $|\pi|$ counts the number of sets in the partition. This monotone problem formulation yields the following extension problem:

Ext BP
Input: A set of items X with weight function w, a partition π_U of X (i. e., $\pi_U\in\mathit{presol}(X)$).
Question: Is $\mathit{ext}(X,\pi_U)\neq\emptyset$?

It turns out that this extension problem is already NP-hard for the very restricted case that π_U only contains two sets. Hence, Ext BP with standard parameterization is para-NP-hard, since NP-hardness already holds for the restriction to instances (X,π_U) with parameter value $m(X,\pi_U)=2$.

Theorem 13. Ext BP *is* NP*-complete, even if the pre-solution π_U contains only two sets.*

Proof. Observe that BP admits polynomial computation of predecessors and hence Ext BP is in NP by Proposition 3.

The proof then consists of a reduction from 3-Partition which is defined as follows: given a multiset $S=\{s_1,\ldots,s_{3m}\}$ of positive integers and a positive integer b as input, decide if S can be partitioned into m triples $S_1,\ldots,S_m$ such that the sum of each subset equals b. 3-Partition is NP-complete even if each integer satisfies $b/4<s_i<b/2$; see [19].

Let $(S=\{s_1,\ldots,s_{3m}\},b)$ be the input of 3-Partition, where $b/4<s_i<b/2$ for each $1\leq i\leq 3m$. We build a set $X=\{x_0,x_1,\ldots,x_{3m}\}$ of items and a weight function w where $w(x_0)=\frac{1}{b}$ and $w(x_i)=\frac{s_i}{b}$ for each $1\leq i\leq 3m$ and set $\pi_U=\{\{x_0\},\{x_1,\ldots,x_{3m}\}\}$ as a partial partition of X.

Claim: (S,b) is a *yes*-instance of 3-Partition if and only if (X,π_U) is a *yes*-instance of Ext BP.

Suppose first that S can be partitioned into m triples $S_1,\ldots,S_m$, where $\sum_{s_j\in S_i}s_j=b$ for each $S_i\in S$. We build a set $X_i=\{x_j\colon 1\leq j\leq 3m, s_j\in S_i\}$, $1\leq i\leq m$. Considering π_U, $\pi'_U=\{\{x_0\},X_1,\ldots,X_m\}$ is a feasible partition and since for each $S_i\in S$, $\sum_{s_j\in S_i}s_j=b$, we have

$w(X_i) = 1$ for each $X_i \in \pi'_U$. Hence π'_U is not the refinement of any other feasible partition for (S, b), as especially x_0 cannot be added to any subset $X_i \in \pi'_U$. Since π'_U is obviously a refinement of π_U, π'_U is a minimal feasible partition with $\pi_U \preceq \pi'_U$.

Conversely, assume that π'_U is a minimal partition of X as a refinement of π_U. As the set $\{x_0\}$ in the partition π_U can not be split up further, it follows that the extension π'_U is of the form $\{\{x_0\}, X_1, \ldots, X_k\}$. By using the minimality of π'_U, it follows especially that $\sum_{x_l \in X_i} w(x_l) + w(x_0) > 1$ for all $i \in \{1, \ldots, k\}$, as otherwise $\pi''_U = \{X_1, \ldots, X_{i-1}, X_i \cup \{x_0\}, X_{i+1}, \ldots, X_k\}$ would be a feasible partition of X with $\pi''_U \preceq \pi'_U$. This implies that $w(X_i) = 1$ for all $1 \leq i \leq k$, since all values $w(x_j)$ are multiples of $\frac{1}{b}$, which means that $w(X_i) < 1$ yields $w(X_i) \leq 1 - \frac{1}{b} = 1 - w(x_0)$, a contradiction to minimality. Consider the collection of the sets $S_i = \{s_j : 1 \leq j \leq 3m, x_j \in X_i\}$, $1 \leq i \leq k$, as a partition for S. Since $w(X_i) = 1$ it follows that $\sum_{s_l \in S_i} s_l = b$ for each $i \in \{1, \ldots, m\}$. The requirement $b/4 < s_i < b/2$ for each $1 \leq i \leq 3m$ then implies that the size of each X_i equals 3, which overall means that $S_1, \ldots, S_m$ is a solution for 3-Partition on (S, b). □

Remark 14. At last, we like to note that dual parameterization for EXT BP easily yields membership in FPT by kernelization. Recall that by our definitions, the dual parameter for instance (X, π_U) is $|X| - |\pi_U|$, as putting each object in its own set is the pre-solution with largest value with respect to m. Consider the reduction rule that, for a partition π_U of X given by sets $X_1, \ldots, X_k$, removes for all i with $X_i = \{x_i\}$ the elements x_i from X and X_i from π_U, leaving the dual parameter $k_d = n - k$ unaffected. An irreducible instance is then a partition $\pi_U = \{X_1, \ldots, X_k\}$ of X, $|X| = n$, with $|X_i| \geq 2$ and hence $2k \leq n = |X| = \sum_{i=1}^{k} |X_i|$, so that $k_d = n - k \geq \frac{1}{2}n$. It is known that the number of partitions of an m-element set is given by the m^{th} Bell number, which again is upper-bounded by $\mathcal{O}(m^m)$. Hence, by simple brute-force, an instance (X, π_U) can be solved in time $\mathcal{O}^*(k_d^{k_d})$.

6 Conclusions

This paper gives a general framework to capture the problem of minimal extension with the aim to highlight useful relationships between different specific extension problems on the one hand, but also to capture broader aspects like parameterized and approximation approaches. In view of the richness of combinatorial problems, many other areas could be looked into with our extension model. Further, it would be interesting to investigate to what extent enumeration problems can be improved by a clever solution to extension or, conversely, how the difficulty of extension implies bounds on enumeration problems. Also, it might be interesting to investigate further additional or alternative parameters for extension problems. For instance, in the case of minimization problems, it might be known that $|V|$ is not reachable by any extension of U concerning a typical graph extension instance (V, E, U). Then, having a better estimate $p_U < |V|$ for the size of a possible extension would provide a more interesting parameter $p_U - |U|$ that is always smaller than the dual parameter that we considered. Again, this area is widely open for research.

Let us also give one concrete open question in the spirit of the mentioned letter of Gödel to von Neumann: Is it possible to design an exact algorithm for Upper Domination (the task to find a minimal dominating set of maximum cardinality) that avoids enumerating all minimal dominating sets? This still unsolved question already triggered quite some research; see [2, 3].

From the specific extension problems for feedback vertex/edge set studied here, some open problems are the following ones:

- Does Ext DFES remain NP-hard when restricted to planar graphs, or can planarity be exploited (like for the decision problem) to find an efficient algorithm?
- Is Ext DFES with standard parameterization in FPT?
- To which class in the W-hierarchy does Ext FVS (and also Ext DFVS) with standard parameterization belong to? Note that we only showed W[1]-hardness but no membership, and not even W[2]-membership appears to be apparent.

At last, let us give a remark concerning the exact exponential complexity of the specific extension problems discussed here. Assuming the ETH, it is known that there is no $2^{o(n+m)}$-algorithm for solving n-variable, m-clause instances of $(3, B2)$-SAT and no $2^{o(n+m)}$-algorithm for n-vertex, m-edge bipartite subcubic instances of Ext VC [11]. Therefore, by reducing from these problems, we can conclude the following by observing that our reductions increase the size of the instances only linearly.

Corollary 15. *Under the ETH, there is no $2^{o(n+m)}$-algorithm for n-vertex, m-edge bipartite instances of* Ext DS*,* Ext FVS*,* Ext DFVS*,* Ext DFES*.*

Looking again at the reductions, this result is also valid for degree-restricted instances, but different formulations would apply for the different problems, so that we do not integrate it into the corollary.

Acknowledgements

This paper is dedicated to the memory of Prof. Jérôme Monnot, who untimely passed away while we were working on this project. It was in fact Jérôme who first brought our attention to the question of the complexity of minimal extension (for the dominating set problem) which is why in the first couple of months working on this, we kept referring to Ext DS as *Jérôme's problem.*

References

1. S. Arora and B. Barak. *Computational Complexity - A Modern Approach.* Cambridge University Press, 2009.
2. C. Bazgan, L. Brankovic, K. Casel, and H. Fernau. Domination chain: Characterisation, classical complexity, parameterised complexity and approximability. *Discrete Applied Mathematics*, 280:23–42, 2020.
3. 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. *Theoretical Computer Science*, 717:2–25, 2018.
4. T. Bläsius, T. Friedrich, J. Lischeid, K. Meeks, and M. Schirneck. Efficiently enumerating hitting sets of hypergraphs arising in data profiling. In S. G. Kobourov and H. Meyerhenke, editors, *Proceedings of the Twenty-First Workshop on Algorithm Engineering and Experiments, ALENEX*, pages 130–143. SIAM, 2019.
5. E. Boros, K. M. Elbassioni, V. Gurvich, L. Khachiyan, and K. Makino. On generating all minimal integer solutions for a monotone system of linear inequalities. In F. Orejas, P. G. Spirakis, and J. van Leeuwen, editors, *Automata, Languages and Programming, 28th International Colloquium, ICALP*, volume 2076 of *Lecture Notes in Computer Science*, pages 92–103. Springer, 2001.
6. E. Boros, V. Gurvich, and P. L. Hammer. Dual subimplicants of positive Boolean functions. *Optimization Methods and Software*, 10(2):147–156, 1998.
7. E. Boros, V. Gurvich, L. Khachiyan, and K. Makino. Dual-bounded generating problems: Partial and multiple transversals of a hypergraph. *SIAM Journal on Computing*, 30(6):2036–2050, 2000.
8. J. Bruchertseifer and H. Fernau. Synchronizing series-parallel automata with loops. In R. Freund, M. Holzer, and J. M. Sempere, editors, *Eleventh Workshop on Non-Classical Models of Automata and Applications, NCMA*, pages 63–78. Österreichische Computer Gesellschaft, 2019.
9. J. Bruchertseifer and H. Fernau. Synchronizing words and monoid factorization: A parameterized perspective. In J. Chen, Q. Feng, and J. Xu, editors, *16th Annual Conference on Theory and Applications of Models of Computation, TAMC*, volume 12337 of *Lecture Notes in Computer Science*, pages 352–364. Springer, 2020.
10. K. Casel, H. Fernau, M. K. Ghadikolaei, J. Monnot, and F. Sikora. Extension of some edge graph problems: Standard and parameterized complexity. In L. A. Gasieniec, J. Jansson, and C. Levcopoulos, editors, *Fundamentals of Computation Theory - 22nd International Symposium, FCT*, volume 11651 of *Lecture Notes in Computer Science*, pages 185–200. Springer, 2019.
11. K. Casel, H. Fernau, M. K. Ghadikolaei, J. Monnot, and F. Sikora. Extension of vertex cover and independent set in some classes of graphs. In P. Heggernes, editor, *Algorithms and Complexity - 11th International Conference, CIAC*, volume 11485 of *Lecture Notes in Computer Science*, pages 124–136. Springer, 2019.
12. K. Casel, H. Fernau, M. K. Ghadikolaei, J. Monnot, and F. Sikora. Abundant extensions. In T. Calamoneri and F. Corò, editors, *Algorithms and Complexity - 12th International Conference, CIAC*, volume 12701 of *Lecture Notes in Computer Science*, pages 3–17. Springer, 2021.

13. J. Chen and F. Zhang. On product covering in 3-tier supply chain models: Natural complete problems for W[3] and W[4]. *Theoretical Computer Science*, 363(3):278–288, 2006.
14. M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.
15. P. Damaschke. Parameterized enumeration, transversals, and imperfect phylogeny reconstruction. *Theoretical Computer Science*, 351(3):337–350, 2006.
16. R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.
17. H. Fernau. On parameterized enumeration. In O. H. Ibarra and L. Zhang, editors, *Computing and Combinatorics, COCOON*, volume 2387 of *Lecture Notes in Computer Science*, pages 564–573. Springer, 2002.
18. H. Fernau and S. Hoffmann. Extensions to minimal synchronizing words. *Journal of Automata, Languages and Combinatorics*, 24:287–307, 2019.
19. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.
20. F. Gavril. Some NP-complete problems on graphs. In *Proc. Conf. on Inform. Sci. and Systems, 1977*, pages 91–95, 1977.
21. M. K. Ghadikolaei, N. Melissinos, J. Monnot, and A. Pagourtzis. Extension and its price for the connected vertex cover problem. In C. J. Colbourn, R. Grossi, and N. Pisanti, editors, *Combinatorial Algorithms - 30th International Workshop, IWOCA*, volume 11638 of *Lecture Notes in Computer Science*, pages 315–326. Springer, 2019.
22. P. A. Golovach, P. Heggernes, M. M. Kanté, D. Kratsch, and Y. Villanger. Enumerating minimal dominating sets in chordal bipartite graphs. *Discrete Applied Mathematics*, 199:30–36, 2016.
23. P. A. Golovach, P. Heggernes, M. M. Kanté, D. Kratsch, and Y. Villanger. Minimal dominating sets in interval graphs and trees. *Discrete Applied Mathematics*, 216:162–170, 2017.
24. P. A. Golovach, P. Heggernes, D. Kratsch, and Y. Villanger. An incremental polynomial time algorithm to enumerate all minimal edge dominating sets. *Algorithmica*, 72(3):836–859, 2015.
25. J. Hromkovic. *Algorithmics for Hard Problems - Introduction to Combinatorial Optimization, Randomization, Approximation, and Heuristics, Second Edition*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004.
26. R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.
27. 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 *Lecture Notes in Computer Science*, pages 446–457. Springer, 2015.
28. 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 *Lecture Notes in Computer Science*, pages 138–153. Springer, 2016.
29. R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972.
30. L. Khachiyan, E. Boros, K. M. Elbassioni, and V. Gurvich. On enumerating minimal dicuts and strongly connected subgraphs. *Algorithmica*, 50(1):159–172, 2008.
31. L. G. Khachiyan, E. Boros, K. M. Elbassioni, V. Gurvich, and K. Makino. On the complexity of some enumeration problems for matroids. *SIAM Journal of Discrete Mathematics*, 19(4):966–984, 2005.
32. J. Kratochvíl. A special planar satisfiability problem and a consequence of its NP-completeness. *Discrete Applied Mathematics*, 52:233–252, 1994.
33. E. L. Lawler, J. K. Lenstra, and A. H. G. R. Kan. Generating all maximal independent sets: NP-hardness and polynomial-time algorithms. *SIAM Journal on Computing*, 9:558–565, 1980.
34. C. L. Lucchesi and D. H. Younger. A minimax theorem for directed graphs. *Journal of the London Mathematical Society*, 2(17):369374, 1978.
35. D. F. Manlove. *Minimaximal and maximinimal optimisation problems: a partial order-based approach*. PhD thesis, University of Glasgow, Computing Science, 1998.
36. A. Mary. *Énumération des dominants minimaux d'un graphe*. PhD thesis, LIMOS, Université Blaise Pascal, Clermont-Ferrand, France, Nov. 2013.
37. A. Mary and Y. Strozecki. Efficient enumeration of solutions produced by closure operations. *Discrete Mathematics and Theoretical Computer Science*, 21(3), 2019.
38. J. W. Moon and L. Moser. On cliques in graphs. *Israel Journal of Mathematics*, 3:23–28, 1965.

39. B. Schwikowski and E. Speckenmeyer. On enumerating all minimal solutions of feedback problems. *Discrete Applied Mathematics*, 117:253–265, 2002.
40. M. Sipser. The history and status of the P versus NP question. In S. R. Kosaraju, M. Fellows, A. Wigderson, and J. A. Ellis, editors, *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, STOC*, pages 603–618. ACM, 1992.
41. E. Speckenmeyer. On feedback vertex sets and nonseparating independent sets in cubic graphs. *Journal of Graph Theory*, 12(3):405–412, 1988.
42. Y. Strozecki. Enumeration complexity. *Bulletin of the EATCS*, 129, 2019.