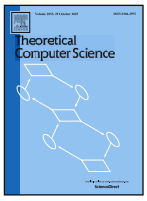
Theoretical Computer Science 1069 (2026) 115820



Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs





The complexity of blocking all solutions

Christoph Grüne [a,*], Lasse Wulf [b]

[a] Department of Computer Science, RWTH Aachen University, Ahornstr. 55, Aachen, 52074, Germany
[b] Section of Theoretical Computer Science, IT University of Copenhagen, Rued Langgaards Vej 7, Copenhagen, 2300, Denmark

ARTICLE INFO

Prof. Seth Pettie
Handling editor: Katie Harris

Keywords:
Computational complexity
Robust optimization
Most vital elements
Most vital nodes
Most vital vertex
Most vital edges
Blocker problems
Vertex blocker
Node blocker
Edge blocker
Interdiction problems
Polynomial hierarchy
Sigma-2

ABSTRACT

We consider the general problem of blocking all solutions of some given combinatorial problem with only few elements. For example, the problem of destroying all Hamiltonian cycles of a given graph by forbidding only few edges; or the problem of destroying all maximum cliques of a given graph by forbidding only few vertices. Problems of this kind are so fundamental that they have been studied under many different names in many different disjoint research communities already since the 90s. Depending on the context, they have been called the interdiction, most vital vertex, most vital edge, blocker, or vertex deletion problem.

Despite their apparent popularity, surprisingly little is known about the computational complexity of interdiction problems in the case where the original problem is already NP-complete. In this paper, we fill that gap of knowledge by showing that a large amount of interdiction problems are even harder than NP-hard. Namely, they are complete for the second stage of Stockmeyer's polynomial hierarchy, the complexity class Σ_2^p. Such complexity insights are important because they imply that all these problems cannot be modelled by a compact integer program (unless the unlikely conjecture NP $= \Sigma_2^p$ holds). Concretely, we prove Σ_2^p-completeness of the following interdiction problems: satisfiability, dominating set, set cover, hitting set, feedback vertex set, feedback arc set, uncapacitated facility location, p-center, p-median, independent set, clique, subset sum, knapsack, Hamiltonian path/cycle (directed/undirected), TSP, k-directed vertex disjoint path ($k \geq 2$), Steiner tree. We show that all of these problems share an abstract property which implies that their interdiction counterpart is Σ_2^p-complete. Thus, all of these problems are Σ_2^p-complete "for the same reason". Our result extends a recent framework by Grüne and Wulf.

1. Introduction

This paper is concerned with the *minimum cardinality interdiction problem*, by which we understand the following task: Given some base problem (the so-called *nominal problem*) we wish to find a small subset of elements such that this subset has a non-empty intersection with every optimal solution of the base problem. The concept of interdiction is so natural that is has reappeared under many different names in different research communities. Depending on the context, the interdiction problem (or slight variants of it) has been called the *most vital node/most vital edge* problem, the *blocker* problem, and *node deletion/edge deletion* problem. As an example for the type of problems that this paper is concerned with, consider the following two problems:

* Corresponding author.
E-mail address: gruene@algo.rwth-aachen.de (C. Grüne).

https://doi.org/10.1016/j.tcs.2026.115820
Received 11 July 2025; Received in revised form 20 November 2025; Accepted 10 February 2026
Available online 12 February 2026
0304-3975/© 2026 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Min Cardinality Clique Interdiction
Input: Graph $G = (V, E)$
Task: Find a minimum-size subset $V' \subseteq V$ such that every maximum clique shares at least one vertex with V'.

Min Cardinality Hamiltonian Cycle Interdiction
Input: Graph $G = (V, E)$
Task: Find a minimum-size subset $E' \subseteq E$ such that every Hamiltonian cycle shares at least one edge with E'.

In particular, if in the above examples the set V' (respectively the set E') is deleted from the graph, the maximum clique size decreases (respectively the graph becomes Hamiltonian-cycle-free). Hence the interdiction problem can be interpreted as the minimal effort required to destroy all optimal solutions. Clearly, analogous problems can be defined and analyzed for a wealth of different nominal problems. Indeed, this has been done extensively by past researchers. The following is a non-exhaustive list: Interdiction-like problems have been considered already since the90ıs for a large amount of problems, among others for shortest path [1–3], matching [4], minimum spanning tree [5], or maximum flow [6]. Note that in all these cases the nominal problem can be solved in polynomial time. Interdiction for nominal problems that are NP-complete has also been extensively considered, for example for vertex covers [7,8], independent sets [7–12], colorings [9,12,13], cliques [13–16], knapsack [17,18], dominating sets [19,20], facility location [21], 1- and p-center [22,23], and 1- and p-median [22,23]. A general survey is provided by Smith, Prince, and Geunes [24].

This large interest is due to the fact that interdiction problems are well-motivated from many different directions. In the area of robust optimization, interdiction is studied because it concerns robust network design, defense against (terrorist) attacks, and sensitivity analysis [25]. In particular, we want to find the most vital nodes/edges of a given network in order to identify its most vulnerable points and understand where small changes have the largest impact. Interdiction in these contexts is often interpreted as a min-max optimization problem or alternatively as a game between a network interdictor (attacker) and a network owner (defender) with competing goals. In the area of bilevel optimization, interdiction-like problems arise naturally from the dynamic between two independent hierarchical agents [17]. In the area of pure graph theory, interdiction problems are usually called vertex and edge blocker problems. They relate to the important concepts of maximum induced subgraphs, critical vertices and edges, cores, and transversals (with respect to some fixed property) [12]. In the area of (parameterized) complexity, interdiction-like problems are usually called vertex deletion problems. They arise from the desire to delete a constant number of vertices until the resulting graph has some desirable property, for example so that it can be handled by an efficient algorithm. For instance, Lewis and Yannakakis showed that the vertex deletion problem for hereditary graph properties is NP-complete [26] and Bannach, Chudigiewitsch and Tantau analyzed the parameterized complexity for properties formulatable by first order formulas [27].

The natural complexity of minimum cardinality interdiction. In this paper, we are mainly concerned with interdiction problems where the nominal problem is already NP-complete. From a complexity-theoretic point of view, such interdiction problems are oftentimes even harder than NP-complete, namely they are complete for the second stage of the *polynomial hierarchy* [28]. A problem complete for the second stage of the hierarchy is called Σ_2^p-complete. The theoretical study of Σ_2^p-complete problems is important: If a problem is found to be Σ_2^p-complete, it means that, under some basic complexity-theoretic assumptions[1], it is not possible to find a mixed-integer linear programming formulation of the problem of polynomial size [29] (also called a *compact* model). This means that no matter how cleverly a decision maker tries to design their mixed-integer linear programming model, it must inherently have a huge number of constraints and/or variables, and may be much harder to solve than even NP-complete problems. Furthermore, for the type of interdiction problems discussed here, where the nominal problem is NP-complete, under the same assumption NP $\neq \Sigma_2^p$, one can show that (the decision variant of) the interdiction problem is oftentimes actually not contained in the complexity class NP, only in the class Σ_2^p. Hence the class Σ_2^p is the natural class for this type of problem.

Even though this fact makes the study of Σ_2^p-complete problems compelling, and even though interdiction-like problems have received a large amount of attention in recent years, surprisingly few Σ_2^p-completeness results relevant to the area of interdiction were known until recently. While the usual approach to prove Σ_2^p-completeness (or NP-completeness) is to formulate a new proof for each single problem, a recent paper by Grüne & Wulf [37], extending earlier ideas by Johannes [30] breaks with this approach. Instead, it is shown that there exists a large 'base list' of problems (called *SSP-NP-complete* problems in [37]), including many classic problems like satisfiability, vertex cover, clique, knapsack, subset sum, Hamiltonian cycle, etc. Grüne and Wulf show that for each problem from the base list, some corresponding min-max version is Σ_2^p-complete, and some corresponding min-max-min version is Σ_3^p-complete. This approach has three main advantages: 1.) It uncovers a large number of previously unknown Σ_2^p-complete problems. 2.) It reveals the theoretically interesting insight, that for all these problems the Σ_2^p-completeness follows from essentially the same argument. 3.) It can simplify future proofs, since heuristically it seems to be true that for a new problem it is often easier to show that the nominal problem belongs to the list of SSP-NP-complete problems, than to find a Σ_2^p-completeness proof from scratch.

Our results. We introduce a meta-theorem that simultaneously classifies the complexity of minimum cardinality interdiction problems, which is arguably the most natural variant of interdiction. For proving this meta-theorem, we extend the framework of Grüne & Wulf [37]. This framework already provides such a result in the case where the action of interdicting an element is associated with so-called interdiction costs, which may be different for each element. We show that in many problem settings it is possible via an easy argument to reduce the *minimum-cost* interdiction problem to the *unit-cost* case. Our paper therefore offers an easy approach to researchers who wish to prove that some unit-cost interdiction problem is Σ_2^p-hard. This proof strategy may be easier than starting

[1] More specifically, we assume here that NP $\neq \Sigma_2^p$, i.e. we assume the polynomial hierarchy does not collapse to the first level. Similar to the famous P $\neq$ NP conjecture, this is believed to be unlikely by experts. However, the true status of the conjecture is not known (see e.g. [29]).

a complicated reduction chain from some Σ_2^p-hard problem from scratch. Additionally, we provide a set of sufficient conditions that specify that a unit-cost interdiction problem is likely not Σ_2^p-complete but belongs to the class coNP. We discuss these cases in more detail in Section 6.

Concretely, we define a general minimum cardinality interdiction problem by embedding it in the SSP framework of [37]. Then, we show that the minimum cardinality interdiction problem is Σ_2^p-complete for the following nominal problems:

> satisfiability, dominating set, set cover, hitting set, feedback vertex set, feedback arc set, uncapacitated facility location, p-center, p-median, independent set, clique, subset sum, knapsack, Hamiltonian path/cycle (directed/undirected), TSP, k-directed vertex disjoint path ($k \geq 2$), Steiner tree.

We remark that Σ_2^p-completeness was already known in the case of clique/independent set, and knapsack [25,31]. Hence our work is a generalization of these results.

Related Work. Usually in the literature, the complexity of interdiction problems is not discussed beyond NP-hardness in the context of the polynomial hierarchy. However, there are the following exceptions: Rutenburg [31] proves Σ_2^p-completeness for unit-cost clique interdiction. Caprara, Carvalho, Lodi & Woeginger [17] consider different bilevel knapsack formulations and prove Σ_2^p-completeness of the DeNegre [32] knapsack variant, which can be interpreted as a min-cost interdiction knapsack variant. An analogous min-cost interdiction knapsack result is obtained by Nabli, Carvalho, and Hosteins [33], along some complexity results about the min-cost multilevel critical node problem. Boggio Tomasaz, Carvalho, Cordone & Hosteins [25] consider interdiction-fortification games and prove Σ_2^p-completeness of the unit-cost interdiction knapsack thus generalizing the results from [17,33] to the unit cost case. Fröhlich and Ruzika prove Σ_2^p-completeness of a unit-cost facility location interdiction problem on graphs (in contrast to our work, the interdictor attacks edges instead of vertices) [21, Section 4]. Our work extends these results to more problem classes. Finally, in a seminal paper, Lewis & Yannakakis prove the very general result that the (unit-cost) most vital vertex problem is NP-hard for every nontrivial hereditary graph property [26]. Our work adds to these results by showing that in many cases, interdiction is even harder than NP-hard. As already mentioned, our work is based on the framework by Grüne and Wulf [37], which itself is based on earlier ideas by Johannes [30].

2. Preliminaries

A *language* is a set $L \subseteq \{0,1\}^*$. A language L is contained in Σ_k^p iff there exists some polynomial-time computable function V (verifier), and $m_1, m_2, \dots, m_k = |w|^{O(1)}$ such that for all $w \in \{0,1\}^*$

$$w \in L \Leftrightarrow \exists y_1 \in \{0,1\}^{m_1}\ \forall y_2 \in \{0,1\}^{m_2} \dots\ Q y_k \in \{0,1\}^{m_k} : V(w, y_1, y_2, \dots, y_k) = 1,$$

where $Q = \exists$, if k is odd, and $Q = \forall$, if k even. An introduction to the polynomial hierarchy and the classes Σ_k^p can be found in the book by Papadimitriou [34] or in the article by Jeroslow [35]. An introduction specifically in the context of bilevel optimization can be found in the article of Woeginger [29].

A *many-one-reduction* or *Karp-reduction* from a language L to a language L' is a map $f : \{0,1\}^* \to \{0,1\}^*$ such that $w \in L$ iff $f(w) \in L'$ for all $w \in \{0,1\}^*$. A language L is Σ_k^p-hard, if every $L' \in \Sigma_k^p$ can be reduced to L with a polynomial-time many-one reduction. If L is both Σ_k^p-hard and contained in Σ_k^p, it is Σ_k^p-complete.

For some cost function $c : U \to \mathbb{R}$, and some subset $U' \subseteq U$, we define the cost of the subset U' as $c(U') := \sum_{u \in U'} c(u)$. For a map $f : A \to B$ and some subset $A' \subseteq A$, we define the image of the subset A' as $f(A') = \{f(a) : a \in A'\}$.

3. Framework

Since this work is an extension of the framework of Grüne & Wulf, it becomes necessary to reintroduce the most important concepts of the framework. A more in-depth explanation of these concepts and their motivation is provided in the original paper [37]. We provide only the concepts, which are necessary for this paper. Grüne & Wulf start by giving a precise definition of the objects they are interested in, linear optimization problems (LOP). An example of an LOP problem is the vertex cover problem.

Definition 1 (Linear Optimization Problem, from [37]). A linear optimization problem (or in short LOP) Π is a tuple $(\mathcal{I}, \mathcal{U}, \mathcal{F}, d, t)$, such that

- $\mathcal{I} \subseteq \{0,1\}^*$ is a language. We call $\mathcal{I}$ the set of instances of Π.
- To each instance $I \in \mathcal{I}$, there is some
  - set $\mathcal{U}(I)$ which we call the universe associated to the instance I.
  - set $\mathcal{F}(I) \subseteq 2^{\mathcal{U}(I)}$ that we call the feasible solution set associated to the instance I.
  - function $d^{(I)} : \mathcal{U}(I) \to \mathbb{Z}$ mapping each universe element e to its costs $d^{(I)}(e)$.
  - threshold $t^{(I)} \in \mathbb{Z}$.

For $I \in \mathcal{I}$, we define the solution set $\mathcal{S}(I) := \{S \in \mathcal{F}(I) : d^{(I)}(S) \leq t^{(I)}\}$ as the set of feasible solutions below the cost threshold. The instance I is a Yes-instance, if and only if $\mathcal{S}(I) \neq \emptyset$. We assume (for LOP problems in NP) that it can be checked in polynomial time in $|I|$ whether some proposed set $F \subseteq \mathcal{U}(I)$ is feasible.

VERTEX COVER
Instances: Graph $G = (V, E)$, number $k \in \mathbb{N}$.
Universe: Vertex set $V =: \mathcal{U}$.
Feasible solution set: The set of all vertex covers of G.
Solution set: The set of all vertex covers of G of size at most k.

It turns out that oftentimes the mathematical discussion is a lot clearer, when one omits the concepts $\mathcal{F}, d^{(I)}$, and $t^{(I)}$, since for the abstract proof of the theorems only $\mathcal{I}$, $\mathcal{U}$, $\mathcal{S}$ are important. This leads to the following abstraction from the concept of an LOP problem:

Definition 2 (Subset Search Problem (SSP), from [37]). A subset search problem (or short SSP problem) Π is a tuple $(\mathcal{I}, \mathcal{U}, \mathcal{S})$, such that

- $\mathcal{I} \subseteq \{0,1\}^*$ is a language. We call $\mathcal{I}$ the set of instances of Π.
- To each instance $I \in \mathcal{I}$, there is some set $\mathcal{U}(I)$ that we call the universe associated to instance I.
- To each instance $I \in \mathcal{I}$, there is some (potentially empty) set $\mathcal{S}(I) \subseteq 2^{\mathcal{U}(I)}$ that we call the solution set associated to instance I.

An instance of an SSP problem is a yes-instance, if $\mathcal{S}(I) \neq \emptyset$. Every LOP problem becomes an SSP problem with the definition $\mathcal{S}(I) := \{S \in \mathcal{F}(I) : d^{(I)}(S) \leq t^{(I)}\}$. We call this the *SSP problem derived from an LOP problem*. Some problems are more naturally modelled as an SSP problem to begin with, rather than as an LOP problem. For example, the satisfiability problem becomes an SSP problem with the following definition.

SATISFIABILITY
Instances: Literal set $L = \{\ell_1, \dots, \ell_n\} \cup \{\overline{\ell}_1, \dots, \overline{\ell}_n\}$, clause set $C = \{C_1, \dots, C_m\}$ such that $C_j \subseteq L$ for all $j \in \{1, \dots, m\}$.
Universe: $L =: \mathcal{U}$.
Solution set: The set of all subsets $L' \subseteq \mathcal{U}$ of the literals such that for all $i \in \{1, \dots, n\}$ we have $|L' \cap \{\ell_i, \overline{\ell}_i\}| = 1$, and such that $|L' \cap C_j| \geq 1$ for all clauses $C_j \in C$.

Grüne & Wulf introduce a new type of reduction, called *SSP reduction*. Roughly speaking, a usual polynomial-time reduction from some problem Π to another problem Π' has the SSP property, if it comes with an additional injective map $f : \mathcal{U} \to \mathcal{U}'$ which embeds the universe $\mathcal{U}$ of Π into the universe $\mathcal{U}'$ of Π', in such a way that Π can be interpreted as a "subinstance" of Π' and the topology of solutions is maintained in the subset that is induced by the image of f. More formally, let W denote the image of f. We interpret W as the subinstance of Π contained in the instance of Π' and we want the following two conditions to hold: 1.) For, every solution S of Π, the set $f(S)$ is a partial solution of Π' and can be completed to a solution by using elements not in W. 2.) For every solution S' of Π', the set $f^{-1}(S' \cap W)$ is a solution of Π. These two conditions together are summarized in the single Eq. (1). We write $\Pi \leq_{\text{SSP}} \Pi'$ to denote that such a reduction exists. We refer the reader to [37] for a more intuitive explanation of these properties and an example reduction of 3SAT $\leq_{\text{SSP}}$ VERTEX COVER.

Definition 3 (SSP Reduction, from [37]). Let $\Pi = (\mathcal{I}, \mathcal{U}, \mathcal{S})$ and $\Pi' = (\mathcal{I}', \mathcal{U}', \mathcal{S}')$ be two SSP problems. We say that there is an SSP reduction from Π to Π', and write $\Pi \leq_{\text{SSP}} \Pi'$, if

- There exists a function $g : \mathcal{I} \to \mathcal{I}'$ computable in polynomial time in the input size $|I|$, such that I is a Yes-instance iff $g(I)$ is a Yes-instance (i.e. $\mathcal{S}(I) \neq \emptyset$ iff $\mathcal{S}'(g(I)) \neq \emptyset$).
- There exist functions $(f_I)_{I \in \mathcal{I}}$ computable in polynomial time in $|I|$ such that for all instances $I \in \mathcal{I}$, we have that $f_I : \mathcal{U}(I) \to \mathcal{U}'(g(I))$ is an injective function mapping from the universe of the instance I to the universe of the instance $g(I)$ such that

$$\{f_I(S) : S \in \mathcal{S}(I)\} = \{S' \cap f_I(\mathcal{U}(I)) : S' \in \mathcal{S}'(g(I))\}. \tag{1}$$

It is shown in [37] that SSP reductions are transitive, i.e. $\Pi_1 \leq_{\text{SSP}} \Pi_2$ and $\Pi_2 \leq_{\text{SSP}} \Pi_3$ implies $\Pi_1 \leq_{\text{SSP}} \Pi_3$. The class of SSP-NP-complete problems is denoted by SSP-NPc and consists of all SSP problems Π that are polynomial-time verifiable and such that SATISFIABILITY $\leq_{\text{SSP}} \Pi$. The main observation in [37] is that many classic problems are contained in the class SSP-NPc, and that this fact can be used to prove that their corresponding min-max versions are Σ_2^p-complete.

4. Minimum cardinality interdiction problems

In this section, we prove our Σ_2^p-completeness results regarding the minimum cardinality interdiction problem. Since we want to prove the theorem simultaneously for multiple problems at once, we require an abstract definition of the interdiction problem. For this, consider the following definition.

Definition 4 (Minimum Cardinality Interdiction Problem). Let an SSP problem $\Pi = (\mathcal{I}, \mathcal{U}, \mathcal{S})$ be given. The minimum cardinality interdiction problem associated to Π is denoted by MIN CARDINALITY INTERDICTION-Π and defined as follows: The input is an instance $I \in \mathcal{I}$ together with a number $k \in \mathbb{N}_0$. The question is whether

$$\exists B \subseteq \mathcal{U}(I),\ |B| \leq k : \forall S \in \mathcal{S}(I) : B \cap S \neq \emptyset.$$

For the remainder of the paper, it is helpful to imagine this problem as a game between two players: the *attacker* and the *defender*. That is, interdiction is an action performed by an attacker (or interdictor), who wishes to select a blocker of few elements to destroy all solutions. On the other hand, the defender wants to find a solution to the problem after the attacker selected a blocker. This leads to the following interpretation:

- The set $\mathcal{U}(I)$ contains all the elements the attacker is allowed to attack.
- The set $\mathcal{S}(I)$ contains all the solutions the attacker wants to destroy such that the defender is not able to find any solution. For example, this could be the set of all Hamiltonian cycles, the set of all cliques of a certain size, etc.

Therefore, the formulation of the base problem as SSP problem $(\mathcal{I}, \mathcal{U}, \mathcal{S})$ determines which elements the attacker can attack, which he cannot attack (e.g. edges/vertices of a graph), and what the attacker's goal is. We note that different formulations $(\mathcal{I}, \mathcal{U}, \mathcal{S})$ of the same problem are formally different SSP problems. They might be both SSP-NP-complete independent of each other, but require their own SSP-NP-completeness proof each. For all the concrete problems studied in this paper, our complexity results hold for the natural choices of $(\mathcal{I}, \mathcal{U}, \mathcal{S})$ formally given in Appendix A. Finally, note that if the base problem is an LOP problem, then by definition $\mathcal{S}(I)$ is the set of feasible solutions below some threshold specified in the input. For example, applying Definition 4 to $\Pi = \textsc{Clique}$ yields the following decision problem:

> Min Cardinality Interdiction-Clique
> **Input**: Graph $G = (V, E)$, numbers $k, t \in \mathbb{N}_0$
> **Question**: Does there exist a subset $B \subseteq V$ of size $|B| \leq k$ such that every clique of size at least t shares at least one vertex with B?

Some more technical details, concerning the subtle differences between different variants of interdiction referenced in the literature as well as concerning the question whether t can be chosen to be optimal are discussed in Section 4.2.

We start the complexity analysis of minimum cardinality interdiction problems by showing the containment in the class Σ_2^p, if the nominal problem is in NP. We then proceed with presenting the main result.

Theorem 1. *Let $\Pi = (\mathcal{I}, \mathcal{U}, \mathcal{S})$ be an SSP problem in SSP-NP, then Min Cardinality Interdiction-Π is in Σ_2^p.*

Proof. We provide a polynomial time algorithm V that verifies a specific solution y_1, y_2 of polynomial size for instance I such that

$$I \in L \Leftrightarrow \exists y_1 \in \{0,1\}^{m_1}\ \forall y_2 \in \{0,1\}^{m_2} : V(I, y_1, y_2) = 1.$$

With the $\exists$-quantified y_1, we encode the blocker $B \subseteq \mathcal{U}(I)$. The encoding size of y_1 is polynomially bounded in the input size of Π because $|\mathcal{U}(I)| \leq poly(|I|)$. Next, we encode the solution $S \in \mathcal{S}(I)$ to the nominal problem Π using the $\forall$-quantified y_2 within polynomial space. This is doable because the problem Π is in SSP-NP. At last, the verifier V has to verify the correctness of the given solution provided by the $\exists$-quantified y_1 and $\forall$-quantified y_2. Checking whether $|B| \leq t$ and $B \cap S \neq \emptyset$ is trivial and checking whether $S \in \mathcal{S}(I)$ is clearly in polynomial time because Π is in SSP-NP. It follows that Min Cardinality Interdiction-Π is in Σ_2^p. □

Next, we show the hardness of minimum cardinality interdiction problems under the condition that the nominal problem is NP-hard. For this, we introduce the concept of invulnerability reductions that helps us to grasp the problems in a unified approach. We describe this concept in the following subsection. Subsequently, we provide invulnerability reductions for a large set of problems in Section 5 with the goal to obtain the following main theorem of the paper. The definitions of the following problems can be found in Appendix A.

Theorem 2. *The problem Min Cardinality Interdiction-Π is Σ_2^p-complete for all the following problems: independent set, clique, subset sum, knapsack, Hamiltonian path/cycle (directed/undirected), TSP, k-directed vertex disjoint paths ($k \geq 2$), Steiner tree, dominating set, set cover, hitting set, feedback vertex set, feedback arc set, uncapacitated facility location, p-center, p-median.*

We remark that this list is missing a few of the SSP-NP-complete problems from the original paper [37]. We discuss in Section 6 the cases of vertex cover and satisfiability, where the above theorem does likely not apply, because the interdiction variant is likely not Σ_2^p-complete. In the case of satisfiability, an alternative variant of the interdiction problem can be proven to still be Σ_2^p-complete.

4.1. Invulnerability reduction

Our proof strategy for each of the problems listed in Theorem 2 is essentially the same. In fact, we show that Theorem 2 is actually a consequence of the following, more powerful *meta-theorem*. This meta-theorem catches the essence of an invulnerability reduction.

Theorem 3. *Consider an SSP-NP-complete problem Π. If there exists a polynomial-time reduction g which receives as input a tuple (I, C, k) of an instance I of Π, some set $C \subseteq \mathcal{U}(I)$ and some $k \in \mathbb{N}_0$, and returns instances $I' := g(I, C, k)$ of Π, such that the following holds:*

$$\exists B \subseteq C : |B| \leq k \text{ and } B \cap S \neq \emptyset\ \forall S \in \mathcal{S}(I) \Leftrightarrow \exists B' \subseteq \mathcal{U}(I') : |B'| \leq k \text{ and } B' \cap S' \neq \emptyset\ \forall S' \in \mathcal{S}(I').$$

Then Min Cardinality Interdiction-Π is Σ_2^p-complete.

It would be nice to have Theorem 3 for all problems in the class SSP-NPc, not only those who admit a function g with the properties as described above. However, we give a reasoning in Section 6 why such a generalization is not possible. The rest of this section is devoted to the proof of Theorem 3. For this proof, we start from the corresponding *combinatorial interdiction problem* as defined in

[37], which we restate in the following. A combinatorial interdiction problem is a more general version of interdiction, where there is a set $C \subseteq \mathcal{U}(I)$ of so-called vulnerable elements. One can also interpret the set of vulnerable elements C as the elements that have cost of interdiction of 1 while all other elements $\mathcal{U}(I) \setminus C$ have a cost of interdiction of ∞ and a blocker of small costs is sought. This problem is called the *combinatorial interdiction problem*.

Definition 5 (Comb. Interdiction Problem, from [37]). Let an SSP problem $\Pi = (\mathcal{I}, \mathcal{U}, \mathcal{S})$ be given. We define Comb. Interdiction-Π as follows: The input is an instance $I \in \mathcal{I}$, a number $k \in \mathbb{N}_0$, and a set $C \subseteq \mathcal{U}(I)$. The set C is called the set of vulnerable elements. The question is whether

$$\exists B \subseteq C,\ |B| \leq k : \forall S \in \mathcal{S}(I) : B \cap S \neq \emptyset.$$

It is proven in [37] that for every problem in SSP-NPc, the combinatorial interdiction problem is Σ_2^p-complete. Now, let Π be in SSP-NPc and g be a reduction such that

$$\exists B \subseteq C : |B| \leq k \text{ and } B \cap S \neq \emptyset\ \forall S \in \mathcal{S}(I) \ \Leftrightarrow\ \exists B' \subseteq \mathcal{U}(I') : |B'| \leq k \text{ and } B' \cap S' \neq \emptyset\ \forall S' \in \mathcal{S}(I'),$$

then g is a reduction from Comb. Interdiction-Π to Min Cardinality interdiction-Π. This is because the first line is equivalent to the statement that instance I is a yes-instance of Comb. Interdiction-Π, and the second line is equivalent to the statement that I' is a yes-instance of Min Cardinality interdiction-Π. It directly follows that Min Cardinality interdiction-Π is Σ_2^p-complete. This completes the proof of Theorem 3.

We remark that while in some sense the proof is rather trivial, we still see a lot of value in explicitly stating a set of easy-to-check sufficient conditions that render some minimum-cardinality interdiction problem Σ_2^p-complete.

How can one find a function g with properties as described above? Oftentimes it is possible by employing the following natural idea: Given an instance of the comb. interdiction problem, let the set $D := \mathcal{U}(I) \setminus C$ be called the *invulnerable* elements. For each problem we explain separately that a gadget for the invulnerable elements in D exists. Intuitively speaking this gadget guarantees that an attacker, no matter which k elements of the universe they attack, can never render the elements of D unusable. On the other hand, we make sure that the *invulnerability gadgets* do not meaningfully change the set of solutions. In Section 5 many examples of such gadgets are presented. We remark that we are not the first to come up with this natural idea. For example, Zenklusen [4] used the same idea in the context of matching interdiction.

4.2. Different variants of interdiction

In this section, we examine variants of interdiction problems that can be found in the literature. For this, we study the relation of our definition of a minimum cardinality interdiction problem and the existing variants. Additionally, we discuss the implications of the hardness of our minimum cardinality interdiction problems on the other variants.

1. Minimal Blocker Problem.

Input Instance I with universe U, blocker cost function c, solution cost function d, and solution threshold τ

Task Find the minimum-cost set $\min_{B \subseteq U} c(B)$ such that for all feasible solutions $F \in \mathcal{F}$ with $F \cap B = \emptyset$, we have $d(F) \leq \tau$.

2. Full Decision Variant of Interdiction.

Input Instance I with universe U, blocker cost function c and budget k, solution cost function d and threshold τ

Task Is there a set $B \subseteq U$ with $c(B) \leq k$ such that for all feasible solutions $F \in \mathcal{F}$ with $F \cap B = \emptyset$, we have $d(F) \leq \tau$?

3. Most Vital Elements Problem.

Input Instance I with universe U, blocker cost function c, blocker budget k, and solution cost function d

Task Find a set $B \subseteq U$ with $c(B) \leq k$ such that the costs of all feasible solutions $F \cap B = \emptyset$ are maximized, i.e. $\max_B \min_{F \in \mathcal{F}, F \cap B = \emptyset} d(F)$.

Note that all of these variants are based on LOP problems instead of SSP problems. An SSP problem, however, can always be trivially expressed as LOP problem by defining $d(u) = 0$ and $t = 0$ thus defining $\mathcal{S} = \mathcal{F}$. Our goal is to show that all of the variants from above are at least as hard as our formulation of *minimum cardinality interdiction* (Definition 4). This results in the following theorem.

Theorem 4. *Let $\Pi = (\mathcal{I}, \mathcal{U}, \mathcal{F}, d, t)$ be an LOP problem. Then the Most Vital Elements Problem of Π (for all problems Π in Theorem 2), the Minimal Blocker Problem of Π, and the Full Decision Variant of Interdiction of Π are at least as hard to compute as Min Cardinality Interdiction-Π.*

The rest of this section is devoted to the proof of this theorem. In our formulation of minimum cardinality interdiction, a set B is sought, which intersects every solution in the set $\mathcal{F}$ as given by the corresponding LOP problem. We now have to distinguish between problems, which are naturally formulated as SSP problems (e.g. Hamiltonian cycle), and naturally formulated LOP problems (e.g. clique). For natural SSP problems, the solution set $\mathcal{S}$ consists of all feasible solutions, i.e. there are no feasible solutions outside of $\mathcal{S}$ due to the missing cost function d on the solution elements. Thus all of the three variants from above are generalizations of minimum cardinality interdiction:

1. The *minimal blocker problem* is the optimization version of the corresponding minimum cardinality interdiction problem.
2. The *full decision version of interdiction* is a generalization of the corresponding minimum cardinality interdiction problem because the latter assumes to have unit costs in the cost function c for all elements from U.
3. The *most vital elements problem* behaves the same as the full decision version of interdiction.

For natural LOP problems, basically the same holds, however, with a modified and a technically more intricate argumentation. Here we are given a feasible solution set $\mathcal{F}$ and the solution set is defined by $\mathcal{S} = \{F \in \mathcal{F} : d(F) \leq t\}$ and we can find a reduction by generalization as follows:

1. For *minimal blocker problems*, we can set $\tau := t - 1$. Then, we again have that the minimal blocker problem on the SSP derived from the LOP problem is the optimization version of the corresponding minimum cardinality interdiction problem.
2. For the *full decision version of interdiction*, we can also set $\tau := t - 1$. Then, the full decision version is again a generalization of the corresponding minimum cardinality interdiction problem due to the fact that the latter has a unit cost function c.
3. For *most vital elements problems*, the situation is more complicated. We first observe that the blocker part of $B \subseteq U$ with $c(B) \leq k$ is a generalization of the blocker part in minimum cardinality interdiction. The inner part on the nominal problem deserves special attention, though, due to the fact that the most vital element problem maximizes the objective while minimum cardinality interdiction blocks all solutions from the solution set $\mathcal{S}$. We focus on this in the next paragraph.

Reducing Minimum Cardinality Interdiction to Most Vital Elements. The concepts of minimum cardinality interdiction and most vital elements coincide if and only of the set $\mathcal{S}$ contains exactly the optimal solutions, i.e. $\mathcal{S} = \{F \in \mathcal{F} : d(F) \leq t^\star\}$, where $t^\star$ is optimal (i.e. minimal). In order to assure that $\mathcal{S}$ captures exactly the optimal solutions, we need to include this condition into the reduction. In particular, the SSP reduction (g, f) needs to guarantee that all instances I are mapped to instances $g(I)$ such that all possible solutions are necessarily optimal. In other words, t is the optimal objective value of the LOP instance $g(I)$, since there are no feasible solutions, whose cost is even smaller than t. We call SSP reductions that fulfill this criterion *tight* and formally define them as follows.

Definition 6 (Tight SSP reduction)**.** Let Π_1 be an SSP problem and $\Pi_2 = (\mathcal{I}, \mathcal{U}, \mathcal{F}, d, t)$ be an LOP problem. Consider an SSP reduction $(g, (f_I)_{I \in \mathcal{I}})$ from Π_1 to (the SSP problem derived from) Π_2. The reduction is called tight if for all yes-instances I_1 of Π_1, the corresponding instance $I_2 = g(I_1)$ of Π_2 with the associated parameter $t := t^{(I_2)}$ and associated cost function $d := d^{(I_2)}$, the following holds:

$$\{F \in \mathcal{F}(I_2) : d(F) \leq t\} \neq \emptyset \text{ and } \{F \in \mathcal{F}(I_2) : d(F) \leq t - 1\} = \emptyset \tag{2}$$

All SSP reductions (to SSP problems derived from LOP problems) that can be found in [37] fulfill this definition and are thus tight. Therefore for all LOP problems (independent set, clique, knapsack, TSP, Steiner tree, dominating set, set cover, hitting set, feedback vertex set, feedback arc set), we obtain that the most vital element problem is at least as hard to compute as the minimum cardinality problem.

Vertex/Edge Deletion Problems. In this paper, we are concerned with finding a set B such that $B \cap S \neq \emptyset$ for every solution S. Note that this definition is meaningful even if the nominal problem is not graph-based. However, in the special case where the nominal problem is graph-based, one could also consider a very related notion which is usually called *vertex deletion problem* or *edge deletion problem*. Here, the question is how many vertices (edges) need to be deleted from the graph until some desired property is met. Element deletion problems are well-studied in classical complexity theory for hereditary graph properties [26] and in parameterized complexity theory for properties expressible by first order formulas [27]. In the general case, element deletion problems are not the same problem as our minimum cardinality interdiction problem. This is because for every set of deleted elements, the underlying instance is changed (vertices/edges are removed, which changes the graph). This is not the case for minimum cardinality interdiction problems as defined in this paper. Thus, it is not possible to transfer the results of minimum cardinality interdiction directly to element deletion problems. Albeit for the problems of clique and independent set, the Σ_2^p-completeness results hold for both minimum cardinality interdiction as well as for vertex deletion interdiction. This is because for these problems the deletion of a vertex coincides with not taking this vertex into the solution. An analogous statement holds for edge deletions for the problems of directed/undirected Hamiltonian cycle/path, k-vertex-disjoint path, and Steiner tree.

5. Invulnerability reductions for various problems

In this section, we show that a lot of well-known problems satisfy the assumptions of Theorem 3, i.e. it is possible to construct invulnerability reductions for them. Note that this proves the hardness part of Theorem 2. The containment is analogous to the one of minimum cost interdiction problems as provided in [37]. Let in the following $C \subseteq \mathcal{U}(I)$ denote the set of vulnerable elements, let $\mathcal{U}(I) \setminus C$ denote the set of invulnerable elements, and k denote the budget of the attacker. The following SSP problems are defined by using the natural choices of $(\mathcal{I}, \mathcal{U}, \mathcal{S})$ and are formally specified in Appendix A.

Clique. We have $\mathcal{U} = V$ in this case. For a given graph $G = (V, E)$, and a set $C \subseteq V$, we explain how to make $V \setminus C$ invulnerable. We obtain a graph G' from G by replacing every vertex $v \in V \setminus C$ with an independent set X_v of size $|X_v| = k + 1$. For a vertex $v \in C$, we define $X_v := \{v\}$. For all edges uv in G, the new graph G' contains the complete bipartite graph between X_u and X_v. Note that every clique of G' contains at most one vertex from every set X_v. Hence the size of a maximum clique is the same in G and G'. Since

for $v \in V \setminus C$, we have $|X_v| = k+1$ and all vertices in X_v have the same neighborhood, the attacker is not able to attack all vertices of X_v at once because its budget of k is too small. Hence v has been made "invulnerable". Furthermore, for every clique in G, we find a corresponding clique in G' that contains at most one vertex from each set X_v. Together, this implies that an attacker can find a set $B' \subseteq V(G')$ of size $|B'| \leq k$ interdicting all maximum cliques in G' if and only the attacker can find a set $B \subseteq C$ of size $|B| \leq k$ interdicting all maximum cliques of G, i.e. the assumptions of Theorem 3 are met.

Independent set. Analogous to clique in the complement graph.

Dominating set. We have $\mathcal{U} = V$ in this case. To make a vertex $v \in V \setminus C$ invulnerable, we use the same construction as for the clique problem, with the only difference that X_v is a clique instead of an independent set. Every optimal dominating set takes at most one vertex from each set X_v, but all $k+1$ vertices inside X_v are equivalent. More precisely, they have the same (closed) neighborhood. This means for an invulnerable vertex $v \in V \setminus C$, an attacker cannot attack all $k+1$ vertices of X_v simultaneously. Furthermore, it is easily seen that on the vulnerable vertices, the attacker interdicts all optimal dominating sets in the old graph if and only if the analogous attack interdicts all optimal dominating sets in the new graph.

Hitting set. In this case, we have some universe $\mathcal{U}$, sets $Y_1, \dots, Y_t \subseteq \mathcal{U}$, and the problem is to find a minimal hitting set $X \subseteq \mathcal{U}$ hitting all the sets Y_j, $j = 1, \dots, t$. To make an element $e \in \mathcal{U}$ invulnerable, simply delete it and replace it by $k+1$ copies. We modify the sets such that every set Y_j that contained e now contains the $k+1$ copies of e instead. It is clear that all the copies of e hit the same sets as e (i.e. taking multiple copies into the hitting set does not offer any advantage). Furthermore, it is not possible for the attacker to attack all $k+1$ copies simultaneously. By an argument analogous to the above paragraphs, we are done.

Set cover. We have a ground set E, and a family $\mathcal{F}$ of sets $S_1, \dots, S_n \subseteq E$ over the ground set. We let $\mathcal{U} := \{1, \dots, n\}$ and the goal is to pick a subset $I \subseteq \mathcal{U}$ of the indices such that $\bigcup_{i \in I} S_i = E$. The attacker can attack up to k of the indices $i \in I$ to forbid the corresponding sets from being picked. We can make some index $i \in \mathcal{U}$ invulnerable by simply duplicating the set S_i a total of $k+1$ times.

Note that this satisfies the assumptions of Theorem 3, but modifies the family $\mathcal{F}$ such that the same set could appear multiple times in the family. Alternatively, our construction can be adjusted such that this is avoided. For this, we introduce $k+1$ new elements $e_1, \dots, e_{k+1}$ and $k+2$ new elements $f_1, \dots, f_{k+2}$ to the ground set E. For each invulnerable index $i \in \{1, \dots, n\} \setminus C$, we substitute S_i by the $k+1$ sets $S_i^{(j)} = S_i \cup \{e_j\}$ for $j = 1, \dots, k+1$. Furthermore, we introduce $k+2$ new sets $S'_j := \{e_1, \dots, e_{k+1}\} \cup \{f_1, \dots, f_{k+2}\} \setminus \{f_j\}$ for $j = 1, \dots, k+2$. This completes the description of the instance. Note that the following holds: The elements $\{f_1, \dots, f_{k+2}\}$ are covered by a set cover, if and only if it contains at least two sets of the form S'_j. Under this assumption, all the elements $\{e_1, \dots, e_{k+1}\}$ are already covered. Hence all the different copies $S_i^{(j)}$ for $j = 1, \dots, k+1$ are essentially equivalent. Thus the attacker can not meaningfully attack all these copies simultaneously. Note that the attacker can also not meaningfully attack the sets S'_j, since no matter which k of them are attacked, 2 of them always remain.

Steiner tree. We have $\mathcal{U} = E$ in this case. To make an edge $uv \in E \setminus C$ invulnerable, we replace it with $k+1$ parallel subdivided edges, i.e. we introduce vertices $w_1, \dots, w_{k+1}$ and edges uw_i and $w_i v$ for $i = 1, \dots, k+1$. Every vulnerable edge uv is replaced with only a single subdivided edge, i.e. a vertex w and edges uw, wv. It is clear that the number of edges of a minimum Steiner tree in the new instance is exactly two times as big as before, and the edge uv has become effectively invulnerable.

Two vertex-disjoint path. We have $\mathcal{U} = A$ in this case. The gadget is the same as for Steiner tree, except that the construction is directed, i.e. the arc (u, v) is replaced either by the arcs $(u, w_i), (w_i, v)$ for $i = 1, \dots, k+1$ (invulnerable case) or by the two arcs $(u, w), (w, v)$ (vulnerable case). Since the paths in this problem have to be vertex disjoint, adding additional subdivided arcs between two existing vertices does not produce additional solutions because traveling from u to v renders all other paths from u to v unusable.

Feedback arc set. We have $\mathcal{U} = A$ in this case. Note that making some arc $a = (u, v) \in A \setminus C$ invulnerable means to ensure that it can be used in a minimal feedback arc set, no matter which k arcs the attacker chooses. This can be achieved the following way: Subdivide a into $k+1$ arcs. Clearly, the set of cycles in the new graph stays essentially the same. Furthermore, the attacker cannot block all $k+1$ arcs from being chosen for the solution. Choosing one of the subdivided pieces of a in the new instance has the same effect as choosing a in the old instance.

Feedback vertex set. We have $\mathcal{U} = V$ in this case. To make a vertex $v \in V \setminus C$ invulnerable, we split it into two vertices v_{in} and v_{out}, put all incoming arcs of the old vertex v to v_{in}, put all outgoing arcs of the old vertex v to v_{out}, and connect v_{in} to v_{out} with a directed path P_v of $k+1$ vertices. Note that in the new instance, a directed cycle uses one vertex of P_v if and only if the cycle uses all vertices of P_v if and only if a corresponding cycle in the old instance uses v. By an analogous to argument to the feedback arc set case, we are done.

Uncapacitated facility location. We have $\mathcal{U} = J$ in this case, where J is the set of sites for potential facilities. The attacker selects facility sites and forbids the decision maker to build a facility there. To make a facility site $j \in J \setminus C$ invulnerable, we can simply delete the site and replace it with $k+1$ identical sites, i.e. sites which have the same facility opening cost and service cost functions as the original facility j. Clearly, this way the attacker can not stop one of the equivalent facilities to be opened. On the other hand, since the facilities are identical (and uncapacitated), the decision maker has no advantage from opening two identical copies of the same facility. Hence the new instance is identical to the old instance, with the only difference that facility site j is invulnerable.

***p*-median, *p*-center.** The difference between the facility location problem and the p-center and p-median problem is that in the latter two, there are no facility opening costs, at most p facilities are allowed to be opened. The service costs in the p-center problem are calculated using a minimum, and in the p-median problem they are calculated using the sum. All of these differences do not affect the argument from above, i.e. one can still make a facility site invulnerable by creating $k+1$ identical facility sites. Hence the same argument holds.

Subset sum. We have $\mathcal{U} = \{1, \dots, n\}$ and are given numbers $a_1, \dots, a_n \in \mathbb{N}$ and a target value T. The question is whether there exists $S \subseteq \mathcal{U}$ with $\sum_{i \in S} a_i = T$. Consider some index $i \in \mathcal{U} \setminus C$. In order to make the index i invulnerable, the first idea is to copy the number a_i a total amount of $k+1$ times. But there is a problem with this construction – if we do this, then the same number a_i could be picked multiple times, which is not allowed in the original instance. We need an additional gadget to make sure that a_i gets used at most once for each i. This can be done in the following way: Choose some number $B > 2k+2$ as a basis. For each $i \in C$, it contains the single number $B^{n(k+1)}a_i$. For each $i \in \{1, \dots, n\} \setminus C$, it contains the $k+1$ distinct numbers $c_i^{(j)} := B^{n(k+1)}a_i + B^{(i-1)(k+1)+j}$ for $j = 0, \dots, k$ as well as the $k+1$ distinct numbers $d_i^{(j)} := \sum_{\ell=0, \ell \neq j}^{k} B^{(i-1)(k+1)+\ell}$ for $j = 0, \dots, k$ and the $k+1$ distinct numbers $e_i^{(j)} := B^{(i-1)(k+1)+j}$ for $j = 0, \dots, k$. We call $d_i^{(j)}$ and $e_i^{(j)}$ the auxiliary numbers. The new instance contains a total of $|C| + 3(k+1)(n - |C|)$ numbers. The new target value is

$$T' := B^{n(k+1)}T + \sum_{i \in \{1,\dots,n\} \setminus C} \sum_{\ell=0}^{k} B^{(i-1)(k+1)+\ell}.$$

Note that this has the following effect: Consider the representation of all involved numbers in base B. Let us call the digits 0 up to $n(k+1)-1$ the lower positions. Note that in the lower positions there can never be any carry, since for every lower position, all involved numbers have either a zero or one in that position and less than B numbers have a one in the same place. Due to that fact, in the lower positions the target T' is reached if and only if for every $i \in \{1, \dots, n\} \setminus C$, the corresponding "bitmask" is filled out (by this, we mean the positions $(i-1)(k+1)$ up to $i(k+1)-1$). This is achieved if and only if for some $j \in \{0, \dots, k\}$ both the values $c_i^{(j)}$ and $d_i^{(j)}$ or both the values $d_i^{(j)}$ and $e_i^{(j)}$ are picked. In particular, at most one of the $k+1$ values $c_i^{(j)}$ for $j = 0, \dots, k$ is picked. In the upper positions, the target T' is reached if and only if the corresponding choice in the old instance meets the target T.

Consider an attack of $k+1$ numbers by the attacker. For each $i \in \{1, \dots, n\} \setminus C$ it holds that there exists a j such that both $c_i^{(j)}$ and $d_i^{(j)}$ are not attacked. Likewise there exists a j' such that both $d_i^{(j')}$ and $e_i^{(j')}$ are not attacked. That means that if i is an invulnerable index, then no matter which $k+1$ values of $c_i^{(j)}$, $d_i^{(j)}$ and $e_i^{(j)}$ are attacked, a correct solution of subset sum will take for some j either both $c_i^{(j)}$ and $d_i^{(j)}$ (which corresponds to taking a_i in the original instance) or take both $d_i^{(j)}$ and $e_i^{(j)}$ (which corresponds to not taking a_i in the original instance). It follows that it is possible to block the new instance by attacking $k+1$ values if and only if it is possible to block the old instance by attacking $k+1$ of the vulnerable values. Finally, if the old numbers $a_1, \dots, a_n$ are pairwise distinct, the new numbers are distinct as well. Hence the interdiction problem for subset sum is Σ_2^p-complete, even if all involved numbers are distinct.

Knapsack. The knapsack problem can be seen as a more general version of the subset sum problem, by creating for each i from the subset sum instance a knapsack item with both profit $p_i = a_i$ and weight $w_i = a_i$, and setting both the weight and profit threshold to T. Hence the Σ_2^p-completeness of Min Cardinality Interdiction-Knapsack follows as a consequence of the Σ_2^p-completeness of Min Cardinality Interdiction-Subset Sum. This holds even if all the involved knapsack items are distinct.

5.1. An invulnerability reduction for Hamiltonian cycle

The invulnerability gadget for Hamiltonian cycle is the most involved of all our constructions, hence we devote a subsection to it. The main result in this section is that the minimum cardinality interdiction problem is Σ_2^p-complete for the nominal problems of both directed and undirected Hamiltonian cycle and path, as well as TSP.

We present our reduction for the case of undirected Hamiltonian cycle and then argue how it can be adapted to the other cases. The main idea is to consider 3-regular graphs $G = (V, E)$ as an intermediate step, and then for a subset $C \subseteq E$ show how $E \setminus C$ can be made invulnerable. To this end, consider the SSP problem 3Reg Ham:

3Reg Ham
Instances: Undirected, 3-regular Graph $G = (V, E)$
Universe: $\mathcal{U} := E$.
Solution set: The set of all Hamiltonian cycles in G.

In [37], it has been shown that Hamiltonian Cycle is SSP-NP-complete. We now require the following stronger statement.

Lemma 1. *3Reg Ham is SSP-NP-complete.*

Proof. Garey, Johnson & Tarjan [36] give a reduction from 3Sat to 3Reg Ham, such that for every variable x_i in the 3Sat instance the graph G has two distinct edges $e(x_i)$ and $e(\overline{x}_i)$ (compare Figure 7 in [36]). Let $E' := \bigcup_i \{e(x_i), e(\overline{x}_i)\}$ be the set of all these edges. For some assignment α of the 3Sat variables, we say that α corresponds to the edge set E_α defined by $\{e(x_i) : \alpha(x_i) = 1\} \cup \{e(\overline{x}_i) : \alpha(x_i) = 0\}$. Garey, Johnson & Tarjan show that there is a bijection between satisfying assignments and edge sets $E'' \subseteq E'$ that can be subset of a Hamiltonian cycle. More formally: 1.) For every satisfying assignment α, if one considers the set $E_\alpha \subseteq E'$ of edges corresponding to that assignment, there exists a Hamiltonian cycle H extending E_α, i.e. $H \cap E' = E_\alpha$. 2.) For every Hamiltonian cycle H, we have that $H \cap E'$ equals E_α for some satisfying assignment α. In total, 1.) and 2.) together show that the reduction in [36] is an SSP-reduction by defining $f(x_i) := e(x_i)$ and $f(\overline{x}_i) := e(\overline{x}_i)$. □

We remark that it follows from [36,38] by the same argument that the problem is even SSP-NP-complete if restricted to 3-regular, bipartite, planar, 2-connected graphs. However, for our arguments it suffices to consider 3-regular graphs.

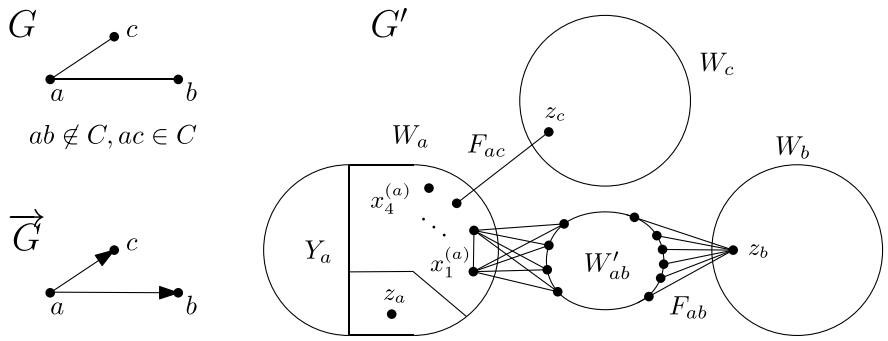


Fig. 1. Invulnerability gagdet for Hamiltonian cycle which makes the edge ab invulnerable while the edge ac remains vulnerable.

Consider now an instance of 3Reg Ham, i.e. a 3-regular undirected graph $G = (V, E)$. Let $C \subseteq E$ be a subset of the edges and $k \in \mathbb{N}_0$ the attacker's budget. We call C the vulnerable edges. Let $D := E \setminus C$. In the remainder of this section we describe and prove a construction how to make the edges in D invulnerable. We quickly sketch the main idea: To make an edge $e = ab$ invulnerable, we enlarge it by replacing it with a large clique W'_{ab} making sure that e can be traversed no matter which k edges inside W'_{ab} are attacked. We also blow up each vertex a of the original graph into a clique W_a. However, this introduces new vertices into the instance, and we need to make sure that a Hamiltonian cycle can always trivially visit all the new vertices. At the same time however, it should still hold that a Hamiltonian cycle in the new graph should be able to enter and exit these new objects W_a and W'_{ab} at most once, since otherwise a corresponding cycle in the old graph G would visit edges or vertices twice, which is of course forbidden. We achieve this by associating to each edge $e = ab$ a star F_{ab} and argue that a Hamiltonian cycle can use at most one edge of each star F_{ab}. Furthermore, we will show that the fact that G is 3-regular implies that each clique W_a can be traversed only once.

We are ready to begin with the construction. Let the directed graph $\vec{G}$ result from G by orienting its edges arbitrarily. We construct an undirected graph $G' = (V', E')$ from $\vec{G}$ as follows: Let $n := |V(G)|$. For each vertex $a \in V(\vec{G})$, let d_a be the out-degree of a, and let W_a be a set of $2d_a + 4k + 1$ vertices. For each invulnerable edge $ab \in D$ in the old graph G, let W'_{ab} be a set of $4k$ vertices. The vertex set $V(G')$ of the new graph G' is then defined by

$$V(G') = \bigcup_{a \in V} W_a \cup \bigcup_{ab \in D} W'_{ab}.$$

We further partition W_a into three disjoint parts $W_a = X_a \cup Y_a \cup \{z_a\}$ of size $|X_a| = 2d_a$ and $|Y_a| = 4k$ and $|\{z_a\}| = 1$. We denote the vertices of X_a by $x_1^{(a)}, \dots, x_{2d_a}^{(a)}$.

The edges of G' are defined as follows: First, we let W_a be a clique for all $v \in V$. Second, for each vertex $a \in V(\vec{G})$, let $e_1, \dots, e_{d_a}$ be its outgoing edges. For each $i = 1, \dots, d_a$, consider the i-th outgoing edge $e_i = (a, b)$ of a, where b is the corresponding neighbor. If $e_i \in C$, i.e. e_i is vulnerable, then G' contains simply the single edge $x_{2i-1}^{(a)} z_b$. In the other case, i.e. $e_i \in D$ is invulnerable, then G' contains an invulnerability gadget as depicted in Fig. 1 induced on the vertices $\{x_{2i-1}^{(a)}, x_{2i}^{(a)}\} \cup W'_{ab} \cup \{z_b\}$. The invulnerability gadget consists of a clique on the vertex set $\{x_{2i-1}^{(a)}, x_{2i}^{(a)}\} \cup W'_{ab}$, together with all edges from the set W'_{ab} to the vertex z_b, i.e. a star graph centered at z_b that has W'_{ab} as its leaves. Let F_{ab} denote this star graph. Finally, for all vulnerable edges $ab \in C$, we also define F_{ab} to be the single edge $x_{2i-1}^{(a)} z_b$ that connects W_a to W_b. This can be interpreted as a trivial star centered at z_b with only one leaf. This completes the description of G'.

The overall idea of this construction is that the cliques of W_a cannot be attacked because they have at least k vertices. Thus it is always possible to find a path visiting all vertices of W_a. Additionally, a star F_{ab} of size larger than k makes the edge $ab \in E$ invulnerable because at most k edges can be attacked. Thus there is always the possibility to travel over one edge of F_{ab} which corresponds to using edge ab in the original graph. On the other hand, since every edge of the star is connected to the same vertex z_b, we have that the star F_{ab} can be used (essentially) exactly once. Thus only the stars of size one (which correspond to the vulnerable edges) are attackable. We now have everything that we need to prove our main result of this section.

Theorem 5. *Minimum cardinality interdiction for Undirected Hamiltonian Cycle is Σ_2^p-complete.*

Proof. Due to [37], and Lemma 1, we have that Comb. Interdiction-3Reg Ham is Σ_2^p-complete. We claim that the construction of G' yields a correct reduction from Comb. Interdiction-3Reg Ham to Min Cardinality Interdiction-HamCycle. Indeed, the following two Lemmas 2 and 3 show that yes-instances of one problem get transformed into yes-instances of the other problem. □

We remark that the 3-regularity of the graph is not maintained by the reduction. Indeed, an argument similar to the arguments given later in Section 6 shows that the interdiction problem for Hamiltonian cycle restriced to only 3-regular graphs is likely not Σ_2^p-complete.

Lemma 2. *If there exists $B' \subseteq E'$ of size $|B'| \leq k$, such that $G' - B'$ has no Hamiltonian cycle, then there is $B \subseteq C$ of size $|B| \leq k$ such that $G - B$ has no Hamiltonian cycle.*

Proof. Proof by contraposition. Assume that for all $B \subseteq C$ with $|B| \leq k$ the graph $G - B$ has a Hamiltonian cycle H. Given some $B' \subseteq E'$ with $|B'| \leq k$, we have to show that the graph $G' - B'$ has a Hamiltonian cycle. Let $B \subseteq C$ be the set of vulnerable edges in G whose copies in G' are attacked by B' (i.e. $B = \{ab \in C : F_{ab} \in B'\}$). Since $B \subseteq C$ and $|B| \leq k$, by assumption $G - B$ has a Hamiltonian cycle H. We want to modify H to a Hamiltonian cycle of H' of $G' - B'$. The basic idea is to follow the same route in G' as H does in G. However, we have to pay attention, because we are not allowed to use edges from B'. We call a vertex in G' *attacked*, if at least one of its incident edges is attacked by B', and call it *free* otherwise. Note that since $|B'| \leq k$ and $|Y_a| = 4k$ and $|W'_{ab}| = 4k$ for $a \in V, ab \in E$, the vertex sets Y_a and W'_{ab} have at least $2k$ free vertices. Free vertices are good for the following reason: Whenever we plan to go from some vertex u to v in G', but we cannot because $uv \in B'$ was attacked, then we can instead choose any free vertex f and go the route u, f, v instead. Now the plan is that H' will roughly employ the following strategy: Follow the same path in G' like H does in G. That is whenever H' enters some new set W_a for the first time, then we first visit all the sets W'_{ab} for all out-neighbors b of a in $\overrightarrow{G}$ if and only if $ab \notin H$. Note that for such b, the set W'_{ab} has two adjacent vertices with W_a (we use these two vertices to enter and leave), and we collect all the vertices of W'_{ab}. Here, we prioritize to visit first the attacked vertices of W'_{ab} and then the remaining vertices of W'_{ab}. After that, we collect all remaining vertices of W_a (again prioritizing the attacked vertices first) before leaving W_a. (If the path on which we are leaving W_a corresponds to an invulnerable edge ab in G, we also collect all of W'_{ab} in the process of leaving W_a.)

Note that this plan might at first not be feasible, because it requires going over some edge $e' \in B'$. However note that, since H does not use any edge of B, for every such edge e' there are always at least $2k$ free vertices that are adjacent to both endpoints of e'. Hence it is possible to "repair" such an edge e' by rerouting over some free vertex instead (and later skip this free vertex). Since there are at most k defects, and there are at least $2k$ free vertices available at the end of traversing every set W_a or W'_{ab}, all defects can be repaired. Hence we can modify H' to be a Hamiltonian cycle of $G' - B'$, which was to show. □

Lemma 3. *If there exists $B \subseteq C$ of size $|B| \leq k$, such that $G - B$ has no Hamiltonian cycle, then there is $B' \subseteq E'$ of size $|B'| \leq k$ such that $G' - B'$ has no Hamiltonian cycle.*

Proof. Proof by contraposition. Assume that for all $B' \subseteq E'$ of size $|B'| \leq k$ the graph $G' - B'$ has a Hamiltonian cycle. Given some $B \subseteq C$ with $|B| \leq k$, we have to show that the graph $G - B$ has a Hamiltonian cycle. Let B' be the trivial stars in G' corresponding to the edges in B (i.e. $B' = \{F_{ab} : ab \in B\}$). Since $|B'| \leq k$, by assumption there is a Hamiltonian cycle H' in $G' - B'$. Consider the set $F := \bigcup_{ab \in E} E(F_{ab})$, i.e. the union of the edge sets of all the stars, trivial or not.

We claim that w.l.o.g. we can assume that $|H' \cap F_{ab}| \leq 1$ for all $ab \in E$. Indeed, the graph $G' - F$ consists out of multiple connected components. Each of these components contains exactly one set of the form W_a, for $a \in V(G)$, and is incident to exactly three sets of the form F_e in G' (where $e \in E(G)$ is an edge that is either incoming to or outgoing from a in $\overrightarrow{G}$). Suppose for some F_{ab} we have $|H' \cap F_{ab}| \geq 2$. Since F_{ab} is a star graph centered around the single vertex z_b, we have $|H' \cap F_{ab}| = 2$. Consider the edge $ab \in E(G)$ such that F_{ab} connects the vertex z_b with W'_{ab}. By the observation about $G' - F$, the following is true about H': It enters W'_{ab} in one of the two vertices attached to X_a, then traverses exactly all of $W'_{ab} \cup \{z_b\}$, then leaves through the other of the two vertices attached to X_a, and at a later point returns to collect all vertices of X_b. However, by the same observation as in Lemma 2, if we define a free vertex to be a vertex not adjacent to any edge in B', then both W'_{ab} and W_b have $2k$ free vertices. Hence we can modify H' such that $H' \cap F_{ab} = \emptyset$. We thus assume that $|H' \cap F_{ab}| \leq 1$ for all $ab \in E$.

Consider again the graph $G' - F$. Since each of its component is adjacent to three sets F_e and $|H' \cap F_e| \leq 1$, we conclude that H' uses exactly two of these three sets F_e. But this implies that H' enters and exits each of the components of $G' - F$ only once and collects all of its vertices in the process. This implies that H' globally follows the same path as some Hamiltonian cycle H of G. Since $H' \subseteq E' - B'$, we conclude $H \subseteq E - B$. This was to show. □

These two lemmas together prove Theorem 5. We would now like to prove Σ_2^p-completeness also for Hamiltonian cycle interdiction of directed graphs. Note that this does not follow from a trivial argument: Even though one can transform an undirected graph into a directed one, by substituting every undirected edge uv by two directed arcs $(u, v), (v, u)$, there is a problem: In the new setting the interdictor needs two attacks to separate u, v, while in the old setting the attacker only needs one.

Still, the above proof can be adapted to the case of directed Hamiltonian cycle the following way: We start with [39], which provides an SSP reduction to prove that the Hamiltonian cycle problem is NP-complete even in directed graphs G such that $\text{indegree}(v) + \text{outdegree}(v) \leq 3$ for every vertex v, and such that for all pairs u, v of G at most one of the two arcs (u, v) and (v, u) is present. Given a directed graph G, we then repeat the same construction as before, with the difference that we can start directly with the directed graph G instead of obtaining an orientation $\overrightarrow{G}$ first. This way, we can obtain an undirected graph G' in the same way as before. In a final step, we turn G' into a directed graph by substituting every undirected edge uv by a pair of two arcs $(u, v), (v, u)$. We perform this substitution for every edge of G' with the exception of the edges that are part of some star F_{ab}. Instead, for each star F_{ab}, we orient the edges of F_{ab} the same way as the original arc of G between a, b. It can be shown that all the arguments from the above construction still hold. Hence the minimum cardinality interdiction problem is Σ_2^p-complete also for directed graphs.

If one is interested in Hamiltonian paths instead of cycles, a similar modification is possible. Inspecting the proof of [36] (or [39], respectively) more closely, we find that in both constructions the graph G contains some edge $e = uv$ (some arc $e = (u, v)$, respectively) such that every Hamiltonian cycle uses e. We can delete e and identify the vertices s, t with the endpoints of e. Then a Hamiltonian s-t-path in the new graph corresponds to a Hamiltonian cycle in the old graph and vice versa. Note that this does not increase the degree of the graph. The rest of the proof proceeds in the same manner, both in the undirected and directed case. Finally, the proof can also easily be adapted to the TSP by a standard reduction of undirected Hamiltonian cycle to the TSP (a graph G is transformed into a TSP instance on the complete graph where the costs obey $c(uv) = 1$ if $ev \in E(G)$ and $c(uv) = n + 1$ if $uv \notin E(G)$). In conclusion, we

have proven that the minimum cardinality interdiction problem is Σ_2^p-complete for the directed/undirected Hamiltonian path/cycle problem and the TSP.

6. Cases where the meta-theorem does not apply

It would be nice to establish a meta-theorem providing Σ_2^p-completeness of the minimum cardinality interdiction version of all nominal problems, which are SSP-NP-complete, instead of only those problems that admit an additional function g with properties as stated in Theorem 3. However, we show in this section that this is not possible. In order to classify problems that do not admit this function g, we provide a lemma that guarantees that the minimum cardinality version of a problem in SSP-NP is in coNP. Therefore, under the usual complexity-theoretic assumption NP $\neq \Sigma_2^p$, the interdiction problem is not Σ_2^p-complete.

In order to provide an intuition under which circumstances a minimum cardinality interdiction problem resides in the class coNP, we examine the vertex cover problem. In a vertex cover, every edge uv needs to be covered by at least one of the two incident vertices u and v. This, however, gives the attacker the opportunity to attack both u and v such that the edge uv can never be covered. Therefore, an attacker budget of at least 2 results in a clear Yes-instance. On the other hand, if the attacker budget if at most 1, we can provide a certificate for No-instances. We can summarize this observation in the following lemma.

Lemma 4. *Let $\Pi = (\mathcal{I}, \mathcal{U}, \mathcal{S})$ be an SSP problem. If in each instance $I \in \mathcal{I}$ there is a subset $U' \subseteq \mathcal{U}(I)$ of constant size, i.e. $|U'| = O(1)$, such that for $U' \cap S \neq \emptyset$ for all $S \in \mathcal{S}(I)$, then* Min Cardinality Interdiction-Π *is contained in coNP.*

Proof. Let k be the interdiction budget. If $|U'| \leq k$, then the interdictor is able to block the whole set U'. By definition of U', there is no solution $S \in \mathcal{S}(I)$ such that $U' \cap S \neq \emptyset$ and thus the interdictor has a winning strategy. If on the other hand $k < |U'| = O(1)$, then there is a polynomially sized certificate encoding a winning strategy of the defender, i.e. a certificate for a No-instance of the problem. For this, we first encode the $\binom{|\mathcal{U}(I)|}{k} = |\mathcal{U}(I)|^{O(1)}$ possible blockers $B' \subseteq \mathcal{U}(I)$ and then the solution $S \in \mathcal{S}(I)$ such that $S \cap B' \neq \emptyset$ for all $B' \subseteq \mathcal{U}(I)$. It is possible to efficiently verify the solution by checking whether there is a solution $S \in \mathcal{S}(I)$ such that $S \cap B' \neq \emptyset$ for all $B' \subseteq \mathcal{U}(I)$ holds because the nominal problem Π is in NP. It follows that the problem lies in coNP. □

Consider the different variants of interdiction problems introduced in Section 4.2. Since they are more general, Lemma 4 does not immediately imply that those variants are contained in coNP. However, if for each instance the stronger condition $U' \cap F \neq \emptyset$ for all feasible solutions $F \in \mathcal{F}(I)$ and for some constant size set $U' \subseteq \mathcal{U}(I)$ holds, then the *full decision variant of interdiction* is contained in coNP. To prove this, we can employ the same strategy as in Lemma 4 while substituting the solutions $\mathcal{S}$ by the feasible solutions $\mathcal{F}$. This also has direct consequences on the *minimal blocker problem* and the *most vital elements problem*. On the one hand, the minimal blocker problem can be efficiently solved given an oracle for the full decision variant. On the other hand, the *most vital elements problem* can be solved by trying out all possible blockers $B' \subseteq \mathcal{U}(I)$ as in Lemma 4. If $k < |U'| = O(1)$, all blockers can be serached for the optimal blocker. If $k \geq |U'|$, then all feasible solutions can be blocked resulting in $\max_B \min_{F, F\cap B=\emptyset} d(F) = \infty$ because $U' \cap F \neq \emptyset$ holds for all feasible solutions $F \in \mathcal{F}(I)$. Besides the containment in coNP, we can also derive the following corollary pinpointing the complexity of minimum cardinality interdiction problems whose nominal problem is in SSP-NP.

Corollary 1. *Let $\Pi = (\mathcal{I}, \mathcal{U}, \mathcal{S})$ be an SSP-NP-complete problem. If in each instance $I \in \mathcal{I}$ there is a subset $U' \subseteq \mathcal{U}(I)$ of constant size, i.e. $|U'| = O(1)$, such that for $U' \cap S \neq \emptyset$ for all $S \in \mathcal{S}(I)$, then* Min Cardinality Interdiction-Π *is coNP-complete.*

Proof. There is a reduction by restriction: Setting the interdiction budget $k = 0$ results in the corresponding co-problem coΠ of the nominal problem Π. □

6.1. Applying the lemma to various problems

In this section, we apply Lemma 4 to the problems mentioned earlier in this paper. Some of the problems are affected in their original general form, e.g. vertex cover or satisfiability, while for others the lemma can be applied on a restricted version such as independent set on graphs with bounded minimum degree. For this, we shortly describe the problem and then give the argument on how the lemma is applicable.

Vertex cover. An instance of the vertex cover interdiction problem consists of a graph G and numbers $t, k \in \mathbb{N}_0$. The question is if the attacker can find a set $B \subseteq V(G)$ with $|B| \leq k$ such that $B \cap S \neq \emptyset$ for every vertex cover S of size at most t. Now, observe that if $k \geq 2$ (and the graph is non-empty), the attacker can easily find such a set B by selecting two adjacent vertices. Thus, Lemma 4 applies by defining $U' = \{u, v\}$ for some edge $uv \in E(G)$. Observe that this not only destroys the solutions $S \in \mathcal{S}(I)$ but also all feasible solutions $F \in \mathcal{F}(I)$. Thus the minimum cardinality interdiction version and the full decision variant of interdiction of vertex cover are coNP-complete.

Satisfiability. An instance of the satisfiability interdiction problem consists out of a formula in CNF over the variables $X = \{x_1, \dots, x_n\}$, with the literal set as universe, i.e. $\mathcal{U} = X \cup \overline{X}$, and interdiction budget k. A similar issue as in vertex cover interdiction arises here: If $k \geq 2$, the interdictor can just choose for some $i \in \{1, \dots, n\}$ to attack both literals $x_i, \overline{x}_i$. Every satisfying assignment (of non-trivial instances) contains either x_i or $\overline{x_i}$, hence this is a successful attack. Thus, Lemma 4 applies by defining $U' = \{x, \overline{x}\}$ for some literal pair $x, \overline{x} \in \mathcal{U}$. Again, this also destroys all feasible solutions $F \in \mathcal{F}(I)$. Thus the minimum cardinality interdiction version and the full decision variant of interdiction of satisfiability are coNP-complete.

Independent set on graphs with bounded minimum degree. An instance of the independent set interdiction problem consists of a graph $G = (V, E)$ with universe $U = V$, a threshold t and an interdiction budget k. The question of the independent set problem is

if there is a set $I \subseteq V$ such that all vertices in I do not share an edge. We now take the vertex d of bounded degree into consideration. If the attacker attacks the closed neighborhood $N[d]$ of d, all (optimal) solutions $S \in \mathcal{S}$ can be interdicted and thus Lemma 4 is applicable. Thus minimum cardinality interdiction independent set on graphs with bounded minimum degree is coNP-complete. In contrast to the other problems, this statement is not true for general feasible solutions $F \in \mathcal{F}(I)$. Hence we do not obtain a result for the variants from Section 4.2.

Dominating set on graphs with bounded minimum degree. An instance of the dominating set interdiction problem consists of a graph $G = (V, E)$ with universe $U = V$, a threshold t and an interdiction budget k. The question of the dominating set problem is if there is a set $D \subseteq V$ of size at most t such that D dominates all vertices of vertex set V. In other words, the union of the neighborhoods of the vertices in D is the vertex set V, i.e. $\bigcup_{v \in D} N[v] = V$. Again we consider a vertex d of bounded degree. Then, we can define the set of constant size to be $U' = N[d]$. All feasible solutions $F \in \mathcal{F}$ have to include some vertex from U' (otherwise d would not be dominated). Thus Lemma 4 is applicable to dominating set. Accordingly, the minimum cardinality interdiction version and the full decision variant of interdiction of dominating set on graphs with bounded minimum degree are coNP-complete.

Hitting set with bounded minimum set size. An instance of hitting set interdiction consists of a ground set $\{1, \dots, n\}$ and m sets $S_j \subseteq \{1, \dots, n\}$ as well as a threshold t and an interdiction budget k. The universe is defined by $\mathcal{U} = \{1, \dots, n\}$. The question of the hitting set problem is whether there is a hitting set $H \subseteq \{1, \dots, n\}$ of size at most t for the sets S_j, that is, $H \cap S_j \neq \emptyset$ for $1 \leq j \leq m$. We can apply Lemma 4 by defining U' to be the set of constant size $|S_c| = O(1)$. Then, the attacker is able to block all elements of the set S_c such that it is not hittable, which interdicts all feasible solutions $F \in \mathcal{F}$. Therefore the minimum cardinality interdiction version and the full decision variant of interdiction of hitting set with bounded minimum set size are coNP-complete.

Set cover with bounded minimum coverage. An instance of the set cover interdiction problem consists of sets $S_i \subseteq \{1, \dots, m\}$ for $1 \leq i \leq n$, a threshold t and an interdiction budget k. The universe is defined as the sets $\{S_i : 1 \leq i \leq n\}$. The question of the set cover problem is whether there is selection $S \subseteq \{S_1, \dots, S_n\}$ of size at most k such that $\bigcup_{s \in S} s = \{1, \dots, m\}$. If there is an element $e \in \{1, \dots, m\}$ of bounded coverage, i.e. there is a constant number of S_i, $1 \leq i \leq n$, with $e \in S_i$, then the attacker can attack all of these sets S_i. Thus, we can apply Lemma 4 by choosing $U' = \{S_i \mid e \in S_i\}$ and all feasible solutions $F \in \mathcal{F}$ are blockable. Accordingly, the minimum cardinality interdiction version and the full decision variant of interdiction of set cover with bounded minimum coverage are coNP-complete.

Steiner tree on graphs with bounded minimum degree of terminal vertices. An instance of the Steiner tree interdiction problem consists of a graph $G = (S \cup T, E)$ of Steiner vertices S and terminals T, edge weights $c : E \to \mathbb{N}$, a threshold t and a interdiction budget k. The universe is the edge set $\mathcal{U} = E$. The question of the Steiner tree problem is if there is a tree $E' \subseteq E$ of weight $c(E') \leq t$ such that all terminal vertices T are connected by E'. If there is a terminal vertex $d \in T$ of bounded degree, then all incident edges build up a set $U' = \{dv \in E\}$ on which we can apply Lemma 4. This blocks all feasible solutions $F \in \mathcal{F}$. Therefore, the minimum cardinality interdiction version and the full decision variant of interdiction of Steiner tree on graphs with bounded minimum degree of terminal vertices are coNP-complete.

Two vertex-disjoint path on graphs with bounded degree. An instance of the two vertex-disjoint path interdiction problem consists of a directed graph $G = (V, A)$, vertices $s_1, s_2, t_1, t_2 \in V$ and interdiction budget k. The universe is the arc set $\mathcal{U} = A$. The question is if there are two paths $P_1, P_2 \subseteq A$ such that P_i starts at s_i and ends at t_i and both paths P_1 and P_2 do not share a vertex. If the graph has bounded degree, we can choose any of the vertices that have to be included in on of the paths, e.g. s_1, and include all the incident arcs in $U' = \{(s_1, v) \in A\}$ such that we can apply Lemma 4. This blocks all feasible solutions $F \in \mathcal{F}$. Accordingly, the minimum cardinality interdiction version and the full decision variant of interdiction of two vertex-disjoint path on graphs with bounded degree are coNP-complete.

Feedback vertex set on graphs with bounded girth. Recall that the girth of the graph is the length of its shortest cycle. An instance of the feedback vertex set interdiction problem consists of a directed graph $G = (V, A)$, a threshold t and interdiction budget k. The universe is the vertex set $\mathcal{U} = V$. The question of feedback vertex set is if there is a set $V' \subseteq V$ such that the graph is cycle free. Accordingly, if the graph has bounded girth, there is a cycle of bounded length, which the attacker can attack or in other words, the cycle cannot be deleted by the defender by choosing a corresponding vertex to be in the feedback vertex set. Thus all feasible solutions $F \in \mathcal{F}$ are blockable by applying Lemma 4 with $U' = \{v \in V \mid v \text{ is part of the smallest cycle in } G\}$. Therefore, the minimum cardinality interdiction version and the full decision variant of interdiction of feedback vertex set on graphs with bounded girth are coNP-complete.

Feedback arc set on graphs with bounded girth. An instance of the feedback arc set interdiction problem consists of a directed graph $G = (V, A)$, a threshold t and interdiction budget k. The universe is the arc set $\mathcal{U} = A$. The question of feedback arc set is if there is an arc set $A' \subseteq A$ such that the graph is acyclic. We can use the same argument as in feedback vertex set. That is, the attacker can choose the arcs of the smallest cycle in G. Thus all feasible solutions $F \in \mathcal{F}$ are blockable by applying Lemma 4 with $U' = \{a \in A \mid a \text{ is part of the smallest cycle in } G\}$. Therefore, the minimum cardinality interdiction version and the full decision variant of interdiction of feedback arc set on graphs with bounded girth are coNP-complete.

Uncapacitated facility location, p-center, or p-median with bounded minimum customer coverage. An instance of the minimum cardinality interdiction version of these three problems consists of a set of potential facilities F and a set of clients C together with a cost function on the facilities $f : F \to \mathbb{N}$ and a service cost function $c : F \times C \to \mathbb{N}$ as well as a threshold t and an interdiction budget k. The universe is the potential facility set $\mathcal{U} = F$ and it is asked for a set of facilities $F' \subseteq F$ not exceeding the cost threshold t. If the coverage of one customer is bounded, i.e. there is a bounded number of potential facilities that are able to serve the customer, the attacker is able to block all of these. Thus we can define U' as the set of potential facilities that are able to serve the customer of bounded coverage such that all feasible solutions $F \in \mathcal{F}$ can be interdicted. Therefore, we can apply Lemma 4 and

the minimum cardinality interdiction version and the full decision variant of interdiction of these three facility locations problems with bounded minimum customer coverage are coNP-complete.

Hamiltonian path/cycle (directed/undirected), TSP on graphs with bounded minimum degree. An instance of the minimum cardinality interdiction version of these problems consists of a graph $G = (V, E)$ (respectively $G = (V, A)$ in the directed case) and an interdiction budget k. The universe is the set of edges $\mathcal{U} = E$ (respectively the set of arcs $\mathcal{U} = A$). The question is whether there is a Hamiltonian path or cycle in G, i.e. a path/cycle that visits every vertex exactly once. Because there is a vertex d of bounded degree which has to be visited, we can define the set of constant size $U' = \{dv \in E\}$ (respectively $U' = \{(d, v), (v, d) \in A\}$). If the set U' is blocked it is not possible to visit the vertex, thus all feasible solutions $F \in \mathcal{F}$ can be interdicted. Therefore, we can apply Lemma 4 and the minimum cardinality interdiction version and the full decision variant of interdiction of these five Hamiltonian problems on graphs with bounded minimum degree are coNP-complete.

6.2. Satisfiability with universe over the variables

In the previous subsection we explained why minimum cardinality interdiction-SAT is contained in coNP, hence likely not Σ_2^p-complete. Note that this is a consequence of our choice of definition of SATISFIABILITY, where we explicitly defined the universe to be the literal set $L = X \cup \overline{X}$. As a consequence, the interdictor may attack $X \cup \overline{X}$.

SATISFIABILITY ($\mathcal{U} = L$)
Instances: Literal Set $L = \{x_1, \dots, x_n\} \cup \{\overline{x}_1, \dots, \overline{x}_n\}$, Clauses $C \subseteq 2^L$
Universe: $L =: \mathcal{U}$.
Solution set: The set of all sets $L' \subseteq \mathcal{U}$ such that for all $i \in \{1, \dots, n\}$ we have $|L' \cap \{\ell_i, \overline{\ell}_i\}| = 1$, and such that $|L' \cap c_j| \geq 1$ for all $c_j \in C$.

An interesting behavior occurs, when we consider the following alternative version SATISFIABILITY ($\mathcal{U} = X$).

SATISFIABILITY ($\mathcal{U} = X$)
Instances: Variable Set $X = \{x_1, \dots, x_n\}$, Clauses $C \subseteq 2^{X \cup \overline{X}}$
Universe: $X =: \mathcal{U}$.
Solution set: The set of all sets $X' \subseteq \mathcal{U}$ such that the assignment $\alpha : X \to \{0, 1\}$ with $\alpha(x) = 1 \leftrightarrow x \in X'$ satisfies all clauses C.

Here the universe is only the variable set X, so in the interdiction version, the interdictor may only attack X, i.e. the interdictor may target individual variables and enforce that they must be set to *false*. We show now that in contrast to the variant, where the universe is the literal set, in this new variant the interdiction problem is Σ_2^p-complete again. Since the problem SATISFIABILITY ($\mathcal{U} = X$) is not part of the original problem set of [37], we perform this proof in two steps.

Lemma 5. *SATISFIABILITY ($\mathcal{U} = X$) is SSP-NP-complete, even when all clauses are restricted to length at most three.*

Proof. We provide an SSP reduction from the SSP-NP-complete problem SATISFIABILITY ($\mathcal{U} = L$) to SATISFIABILITY ($\mathcal{U} = X$). Consider an instance of SATISFIABILITY ($\mathcal{U} = L$) given by a formula φ with n variables $X = \{x_1, \dots, x_n\}$ and universe/literal set $\mathcal{U} = L = X \cup \overline{X}$. SATISFIABILITY ($\mathcal{U} = L$) is SSP-NP-complete even when all clauses are restricted to length three, so let us w.l.o.g. assume that property. We have to show how to embed this universe into the universe $\mathcal{U}'$ of some corresponding SATISFIABILITY ($\mathcal{U} = X$) instance φ', where only positive literals are allowed in $\mathcal{U}'$. This can be done the following way: We introduce $2n$ new variables $X' := \{x_1^t, \dots, x_n^t\} \cup \{x_1^f, \dots, x_n^f\}$. The universe $\mathcal{U}' := X'$ consists out of the $2n$ corresponding positive literals X'. The new formula φ' is defined from φ in two steps. First a substitution process takes place: For each $i = 1, \dots, n$, the positive literal x_i is replaced by the positive literal x_i^t and each negative literal $\overline{x}_i$ is replaced by the positive literal x_i^f. In a second step, the clauses $(x_i^t \vee \overline{x}_i^f) \wedge (\overline{x}_i^t \vee x_i^f)$ (note that these are equivalent to $x_i^t \oplus x_i^f$) are added to φ'. Formally,

$$\varphi' = \text{substitute}(\varphi) \wedge \bigwedge_{i=1}^{n} (x_i^t \vee x_i^f) \wedge (\overline{x}_i^t \vee \overline{x}_i^f).$$

The SSP reduction is completed by specifying the embedding function $f : \mathcal{U} \to \mathcal{U}'$ via $f(x_i) := x_i^t$ and $f(\overline{x}_i) := x_i^f$. Clearly all clauses of φ' have length at most three. Note that this reduction is a correct reduction, i.e. it transforms yes-instances into yes-instances and no-instances into no-instances, because the added constraints make sure that exactly one of x_i^t and x_i^f is true. Furthermore, it has the SSP property: For every solution $S \subseteq \mathcal{U}$ of SATISFIABILITY ($\mathcal{U} = L$), the "translated" set $f(S) \subseteq \mathcal{U}'$ is a solution of SATISFIABILITY ($\mathcal{U} = X$). Furthermore, for every solution $S \subseteq \mathcal{U}'$ of SATISFIABILITY ($\mathcal{U} = X$), the set $f^{-1}(S) \subseteq \mathcal{U}$ is a solution of SATISFIABILITY ($\mathcal{U} = L$). Accordingly, we have a correct SSP reduction (where the SSP mapping f is even bijective due to $f(\mathcal{U}) = \mathcal{U}'$). □

Theorem 6. *MIN CARDINALITY INTERDICTION-SATISFIABILITY ($\mathcal{U} = X$) is Σ_2^p-complete.*

Proof. By the previous lemma, SATISFIABILITY($\mathcal{U} = X$) is SSP-NP-complete, even if all clauses are restricted to length three. Due to [37], the problem COMB. INTERDICTION-SATISFIABILITY ($\mathcal{U} = X$) is Σ_2^p-complete, even if all clauses are restricted to length three. We provide a reduction from the latter problem in terms of an invulnerability gadget analogous to the gadgets presented in Section 5. For this, consider an instance of SATISFIABILITY ($\mathcal{U} = X$) with formula φ in CNF and every clause of length three, together with

the universe $\mathcal{U} = \{x_1, \dots, x_n\}$, a set $C \subseteq \mathcal{U}$ of vulnerable literals, and interdiction budget $k \in \mathbb{N}_0$. For every variable $x_i \in \mathcal{U} \setminus C$, we explain how to make x_i invulnerable. We introduce $k+1$ new variables $x_i^{(1)}, \dots x_i^{(k+1)}$. Our goal is to establish the equivalence

$$x_i \equiv x_i^{(1)} \vee \cdots \vee x_i^{(k+1)}.$$

We can achieve this through means of the following substitution process starting from formula φ: Every occurrence of x_i in the formula gets substituted by $x_i^{(1)} \vee \cdots \vee x_i^{(k+1)}$. Every occurrence of $\overline{x}_i$ gets substituted (by De Morgan's law) by $(\overline{x}_i^{(1)} \wedge \cdots \wedge \overline{x}_i^{(k+1)})$. Note that this has two effects: First, the length of a clause may now exceed 3. Secondly, the formula is not in CNF anymore. Note however that we can use the distributive law to expand every clause that is not in CNF. Since before each clause before had a length of at most three, this results in a blow-up of the instance size of a factor at most $(k+1)^3$, i.e. at most a polynomial factor. Let φ' be the resulting formula. We can see that there is an equivalence of the satisfying assignments of φ and φ', in the sense that x_i is true in φ if and only if $x_i^{(1)} \vee \cdots \vee x_i^{(k+1)}$ is true in φ' (for all invulnerable x_i). However, since the interdiction budget is only k, the interdictor can never enforce $x_i^{(1)} \vee \cdots \vee x_i^{(k+1)}$ to be false for all invulnerable variables. This shows that Comb. Interdiction-Satisfiability ($\mathcal{U} = X$) reduces to Min. Cardinality Interdiction-Satisfiability ($\mathcal{U} = X$), hence proving its Σ_2^p-completeness. □

Note that the reasoning presented in this proof was slightly different from Theorem 3, since we start with a formula where every clause has length three, but do not preserve this property during the proof. Hence Σ_2^p-completeness is only shown in the case where clauses can have unrestricted length.

We can use an argument similar to Lemma 4 to show the coNP-completeness of the minimum cardinality interdiction version, the full decision variant of interdiction and the most vital elements problem of 3-Satisfiability ($\mathcal{U} = X$), i.e. with clauses of length bounded by 3. Indeed, it is easy to see that the interdiction problem of Satisfiability ($\mathcal{U} = X$) where every clause has length 3 is coNP-complete: If $k \geq 3$ holds for the interdiction budget, the attacker distinguishes two cases: If there is a clause with three positive literals, the attacker blocks all of them and immediately wins. In the other case, every clause has at least one negative literal. Then the attacker can never win, since the defender can set every variable to false, which is a satisfying assignment that can never be blocked. By an analogous argument, we can see that for any constant $b \geq 3$, the interdiction problem of Satisfiability ($\mathcal{U} = X$) with clauses restricted to length b is coNP-complete.

Finally, we remark that slightly different variants of interdiction-3-Sat have been shown to be Σ_2^p-complete. In these variants, the interdictor does not have access to all variables (see [37, Sec. 4.2] or [40, Thm. 1]).

7. Conclusion

We have shown that for a large class of NP-complete problems, the corresponding minimum cardinality interdiction problem is Σ_2^p-complete. With that we have also shown the hardness of several different variants of interdiction that can be found in the literature including minimum blocker and most vital elements problems. For this, we introduced a new type of reduction, namely invulnerability reductions. This reduction uses the corresponding minimum cost interdiction problem as basis and ensures that non-blockable elements are effectively not attackable. The hardness of the minimum cost interdiction problem is provable via an SSP reduction. Additionally, we show that for some problems (e.g. vertex cover, satisfiability), the Σ_2^p-completeness cannot be derived despite the fact that the minimum cost interdiction problem is Σ_2^p-complete. Overall, we show for 23 minimum cardinality interdiction problems that they are either Σ_2^p-complete or coNP-complete with the ability to apply the framework to further problems.

The following natural questions arise. First, it is of interest to find more problems for which this framework is applicable. Second, it is relevant whether this framework is also extendable to problems that are in NP but not NP-complete. One might lose the Σ_p^2-completeness for these problems, however, a meta-theorem that proves NP-completeness for such problems and generalizes the existing results in the literature is important to obtain a deeper understanding on the structure of such problems. At last, the results of this paper are not always applicable (albeit sometimes) to the vertex deletion or in general element deletion interdiction problem. Thus, it is of interest to show a similar meta-theorem for element deletion problems.

CRediT authorship contribution statement

Christoph Grüne: Writing – review & editing, Writing – original draft, Investigation, Formal analysis; **Lasse Wulf:** Writing – review & editing, Writing – original draft, Investigation, Formal analysis.

Data availability

No data was used for the research described in the article.

Declaration of competing interests

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

[Christoph Grüne reports financial support was provided by German Research Foundation. Lasse Wulf reports financial support was provided by Carlsberg Foundation. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.]

Appendix A. Problem definitions

SATISFIABILITY
Instances: Literal Set $L = \{\ell_1, \dots, \ell_n\} \cup \{\overline{\ell}_1, \dots, \overline{\ell}_n\}$, Clauses $C \subseteq 2^L$.
Universe: $L =: \mathcal{U}$.
Solution set: The set of all sets $L' \subseteq \mathcal{U}$ such that for all $i \in \{1, \dots, n\}$ we have $|L' \cap \{\ell_i, \overline{\ell}_i\}| = 1$, and such that $|L' \cap c_j| \geq 1$ for all $c_j \in C$, $j \in \{1, \dots, |C|\}$.

3-SATISFIABILITY
Instances: Literal Set $L = \{\ell_1, \dots, \ell_n\} \cup \{\overline{\ell}_1, \dots, \overline{\ell}_n\}$, Clauses $C \subseteq 2^L$ s.t. $\forall c_j \in C : |c_j| = 3$.
Universe: $L =: \mathcal{U}$.
Solution set: The set of all sets $L' \subseteq \mathcal{U}$ such that for all $i \in \{1, \dots, n\}$ we have $|L' \cap \{\ell_i, \overline{\ell}_i\}| = 1$, and such that $|L' \cap c_j| \geq 1$ for all $c_j \in C$.

DOMINATING SET
Instances: Graph $G = (V, E)$, number $k \in \mathbb{N}$.
Universe: Vertex set $V =: \mathcal{U}$.
Feasible solution set: The set of all dominating sets.
Solution set: The set of all dominating sets of size at most k.

SET COVER
Instances: Sets $S_i \subseteq \{1, \dots, m\}$ for $i \in \{1, \dots, n\}$, number $k \in \mathbb{N}$.
Universe: $\{S_1 \dots, S_n\} =: \mathcal{U}$.
Feasible solution set: The set of all $S \subseteq \{S_1, \dots, S_n\}$ s.t. $\bigcup_{s \in S} s = \{1, \dots, m\}$.
Solution set: Set of all feasible solutions with $|S| \leq k$.

HITTING SET
Instances: Sets $S_j \subseteq \{1, \dots, n\}$ for $j \in \{1, \dots, m\}$, number $k \in \mathbb{N}$.
Universe: $\{1, \dots, n\} =: \mathcal{U}$.
Feasible solution set: The set of all $H \subseteq \{1, \dots, n\}$ such that $H \cap S_j \neq \emptyset$ for all $j \in \{1, \dots, m\}$.
Solution set: Set of all feasible solutions with $|H| \leq k$.

FEEDBACK VERTEX SET
Instances: Directed Graph $G = (V, A)$, number $k \in \mathbb{N}$.
Universe: Vertex set $V =: \mathcal{U}$.
Feasible solution set: The set of all vertex sets $V' \subseteq V$ such that after deleting V' from G, the resulting graph is cycle-free (i.e. a forest).
Solution set: The set of all feasible solutions V' of size at most k.

FEEDBACK ARC SET
Instances: Directed Graph $G = (V, A)$, number $k \in \mathbb{N}$.
Universe: Arc set $A =: \mathcal{U}$.
Feasible solution set: The set of all arc sets $A' \subseteq A$ such that after deleting A' from G, the resulting graph is cycle-free (i.e. a forest).
Solution set: The set of all feasible solutions A' of size at most k.

UNCAPACITATED FACILITY LOCATION
Instances: Set of potential facilities $F = \{1, \dots, n\}$, set of clients $C = \{1, \dots, m\}$, fixed cost of opening facility function $f : F \to \mathbb{Z}$, service cost function $c : F \times C \to \mathbb{Z}$, cost threshold $k \in \mathbb{Z}$
Universe: Facility set $F =: \mathcal{U}$.
Solution set: The set of sets $F' \subseteq F$ s.t. $\sum_{i \in F'} f(i) + \sum_{j \in C} \min_{i \in F'} c(i, j) \leq k$.

P-CENTER
Instances: Set of potential facilities $F = \{1, \dots, n\}$, set of clients $C = \{1, \dots, m\}$, service cost function $c : F \times C \to \mathbb{Z}$, facility threshold $p \in \mathbb{N}$, cost threshold $k \in \mathbb{Z}$
Universe: Facility set $F =: \mathcal{U}$.
Solution set: The set of sets $F' \subseteq F$ s.t. $|F'| \leq p$ and $\max_{j \in C} \min_{i \in F'} c(i, j) \leq k$.

P-MEDIAN
Instances: Set of potential facilities $F = \{1, \dots, n\}$, set of clients $C = \{1, \dots, m\}$, service cost function $c : F \times C \to \mathbb{Z}$, facility threshold $p \in \mathbb{N}$, cost threshold $k \in \mathbb{Z}$
Universe: Facility set $F =: \mathcal{U}$.
Solution set: The set of sets $F' \subseteq F$ s.t. $|F'| \leq p$ and $\sum_{j \in C} \min_{i \in F'} c(i, j) \leq k$.

INDEPENDENT SET
Instances: Graph $G = (V, E)$, number $k \in \mathbb{N}$.
Universe: Vertex set $V =: \mathcal{U}$.
Feasible solution set: The set of all independent sets.
Solution set: The set of all independent sets of size at least k.

CLIQUE
Instances: Graph $G = (V, E)$, number $k \in \mathbb{N}$.
Universe: Vertex set $V =: \mathcal{U}$.
Feasible solution set: The set of all cliques.
Solution set: The set of all cliques of size at least k.

Subset Sum
Instances: Numbers $\{a_1, \dots, a_n\} \subseteq \mathbb{N}$, and target value $M \in \mathbb{N}$.
Universe: $\{a_1, \dots, a_n\} =: \mathcal{U}$.
Solution set: The set of all sets $S \subseteq \mathcal{U}$ with $\sum_{a_i \in S} a_i = M$.

Knapsack
Instances: Objects with prices and weights $\{(p_1, w_1), \dots, (p_n, w_n)\} \subseteq \mathbb{N}^2$, and $W, P \in \mathbb{N}$.
Universe: $\{(p_1, w_1), \dots, (p_n, w_n)\} =: \mathcal{U}$.
Feasible solution set: The set of all $S \subseteq \mathcal{U}$ with $\sum_{(p_i, w_i) \in S} w_i \leq W$.
Solution set: The set of feasible S with $\sum_{(p_i, w_i) \in S} p_i \geq P$.

Directed Hamiltonian Path
Instances: Directed Graph $G = (V, A)$, Vertices $s, t \in V$.
Universe: Arc set $A =: \mathcal{U}$.
Solution set: The set of all sets $C \subseteq A$ forming a Hamiltonian path going from s to t.

Directed Hamiltonian Cycle
Instances: Directed Graph $G = (V, A)$.
Universe: Arc set $A =: \mathcal{U}$.
Solution set: The set of all sets $C \subseteq A$ forming a Hamiltonian cycle.

Undirected Hamiltonian Cycle
Instances: Graph $G = (V, E)$.
Universe: Edge set $E =: \mathcal{U}$.
Solution set: The set of all sets $C \subseteq E$ forming a Hamiltonian cycle.

Traveling Salesman Problem
Instances: Complete Graph $G = (V, E)$, weight function $w : E \to \mathbb{Z}$, number $k \in \mathbb{N}$.
Universe: Edge set $E =: \mathcal{U}$.
Feasible solution set: The set of all TSP tours $T \subseteq E$.
Solution set: The set of feasible T with $w(T) \leq k$.

Directed k-Vertex Disjoint Path
Instances: Directed graph $G = (V, A)$, $s_i, t_i \in V$ for $i \in \{1, \dots, k\}$.
Universe: Arc set $A =: \mathcal{U}$.
Solution set: The sets of all sets $A' \subseteq A$ such that $A' = \bigcup_{i=1}^{k} A(P_i)$, where all P_i are pairwise vertex-disjoint paths from s_i to t_i for $1 \leq i \leq k$.

STEINER TREE
Instances: Undirected graph $G = (S \cup T, E)$, set of Steiner vertices S, set of terminal vertices T, edge weights $c : E \to \mathbb{N}$, number $k \in \mathbb{N}$.
Universe: Edge set $E =: \mathcal{U}$.
Feasible solution set: The set of all sets $E' \subseteq E$ such that E' is a tree connecting all terminal vertices from T.
Solution set: The set of feasible solutions E' with $\sum_{e' \in E'} c(e') \leq k$.

References

[1] A. Bar-Noy, S. Khuller, B. Schieber, The complexity of finding most vital arcs and nodes (1998).
[2] L. Khachiyan, E. Boros, K. Borys, K.M. Elbassioni, V. Gurvich, G. Rudolf, J. Zhao, On short paths interdiction problems: total and node-wise limited interdiction, Theory Comput. Syst. 43 (2) (2008) 204–233. https://doi.org/10.1007/S00224-007-9025-6
[3] K. Malik, A.K. Mittal, S.K. Gupta, The k most vital arcs in the shortest path problem, Oper. Res. Lett. 8 (4) (1989) 223–227.
[4] R. Zenklusen, Matching interdiction, Discret. Appl. Math. 158 (15) (2010) 1676–1690. https://doi.org/10.1016/J.DAM.2010.06.006
[5] K. Lin, M. Chern, The most vital edges in the minimum spanning tree problem, Inf. Process. Lett. 45 (1) (1993) 25–31. https://doi.org/10.1016/0020-0190(93)90247-7
[6] R.K. Wood, Deterministic network interdiction, Math. Comput. Model. 17 (2) (1993) 1–18. https://doi.org/10.1016/0895-7177(93)90236-R
[7] C. Bazgan, S. Toubaline, Z. Tuza, Complexity of most vital nodes for independent set in graphs related to tree structures, in: C.S. Iliopoulos, W.F. Smyth (Eds.), Combinatorial Algorithms - 21st International Workshop, IWOCA 2010, London, UK, July 26-28, 2010, Revised Selected Papers, 6460 of *Lecture Notes in Computer Science*, Springer, 2010, pp. 154–166. https://doi.org/10.1007/978-3-642-19222-7_17
[8] C. Bazgan, S. Toubaline, Z. Tuza, The most vital nodes with respect to independent set and vertex cover, Discret. Appl. Math. 159 (17) (2011) 1933–1946. https://doi.org/10.1016/j.dam.2011.06.023
[9] C. Bazgan, C. Bentz, C. Picouleau, B. Ries, Blockers for the stability number and the chromatic number, Graphs Comb. 31 (1) (2015) 73–90. https://doi.org/10.1007/s00373-013-1380-2
[10] H.P. Hoang, S. Lendl, L. Wulf, Assistance and interdiction problems on interval graphs, Discret. Appl. Math. 340 (2023) 153–170. https://doi.org/10.1016/J.DAM.2023.06.046
[11] F. Lucke, B. Ries, On blockers and transversals of maximum independent sets in co-comparability graphs, Discret. Appl. Math. 356 (2024) 307–321. https://doi.org/10.1016/J.DAM.2024.06.020
[12] D. Paulusma, C. Picouleau, B. Ries, Blocking independent sets for H-Free graphs via edge contractions and vertex deletions, in: T.V. Gopal, G. Jäger, S. Steila (Eds.), Theory and Applications of Models of Computation - 14th Annual Conference, TAMC 2017, Bern, Switzerland, April 20-22, 2017, Proceedings, 10185 of *Lecture Notes in Computer Science*, 2017, pp. 470–483. https://doi.org/10.1007/978-3-319-55911-7_34
[13] D. Paulusma, C. Picouleau, B. Ries, Reducing the clique and chromatic number via edge contractions and vertex deletions, in: R. Cerulli, S. Fujishige, A.R. Mahjoub (Eds.), Combinatorial Optimization - 4th International Symposium, ISCO 2016, Vietri Sul Mare, Italy, May 16-18, 2016, Revised Selected Papers, 9849 of *Lecture Notes in Computer Science*, Springer, 2016, pp. 38–49. https://doi.org/10.1007/978-3-319-45587-7_4
[14] F. Furini, I. Ljubic, S. Martin, P.S. Segundo, The maximum clique interdiction problem, Eur. J. Oper. Res. 277 (1) (2019) 112–127. https://doi.org/10.1016/j.ejor.2019.02.028
[15] F.M. Pajouh, Minimum cost edge blocker clique problem, Ann. Oper. Res. 294 (1) (2020) 345–376. https://doi.org/10.1007/s10479-019-03315-x
[16] F.M. Pajouh, V. Boginski, E.L. Pasiliao, Minimum vertex blocker clique problem, Networks 64 (1) (2014) 48–64. https://doi.org/10.1002/net.21556
[17] A. Caprara, M. Carvalho, A. Lodi, G.J. Woeginger, A complexity and approximability study of the bilevel knapsack problem, in: M.X. Goemans, J.R. Correa (Eds.), Integer Programming and Combinatorial Optimization - 16th International Conference, IPCO 2013, Valparaíso, Chile, March 18-20, 2013. Proceedings, 7801 of *Lecture Notes in Computer Science*, Springer, 2013, pp. 98–109. https://doi.org/10.1007/978-3-642-36694-9_9
[18] N. Weninger, R. Fukasawa, A fast combinatorial algorithm for the bilevel knapsack problem with interdiction constraints, Math Program (2024) 1–33.
[19] E. Galby, P.T. Lima, B. Ries, Reducing the domination number of graphs via edge contractions and vertex deletions, Discret. Math. 344 (1) (2021) 112169. https://doi.org/10.1016/J.DISC.2020.112169
[20] F.M. Pajouh, J.L. Walteros, V. Boginski, E.L. Pasiliao, Minimum edge blocker dominating set problem, Eur. J. Oper. Res. 247 (1) (2015) 16–26. https://doi.org/10.1016/j.ejor.2015.05.037
[21] N. Fröhlich, S. Ruzika, On the hardness of covering-interdiction problems, Theor. Comput. Sci. 871 (2021) 1–15. https://doi.org/10.1016/J.TCS.2021.04.007
[22] C. Bazgan, S. Toubaline, D. Vanderpooten, Complexity of determining the most vital elements for the 1-median and 1-center location problems, in: W. Wu, O. Daescu (Eds.), Combinatorial Optimization and Applications - 4th International Conference, COCOA 2010, Kailua-Kona, HI, USA, December 18-20, 2010, Proceedings, Part I, 6508 of *Lecture Notes in Computer Science*, Springer, 2010, pp. 237–251. https://doi.org/10.1007/978-3-642-17458-2_20
[23] C. Bazgan, S. Toubaline, D. Vanderpooten, Complexity of determining the most vital elements for the p-median and p-center location problems, J. Comb. Optim. 25 (2) (2013) 191–207. https://doi.org/10.1007/s10878-012-9469-8
[24] J.C. Smith, M. Prince, J. Geunes, Modern network interdiction problems and algorithms, in: Handbook of Combinatorial Optimization, Springer New York, 2013, pp. 1949–1987.
[25] A.B. Tomasaz, M. Carvalho, R. Cordone, P. Hosteins, On the completeness of several fortification-interdiction games in the polynomial hierarchy, CoRR abs/2406.01756 (2024). https://doi.org/10.48550/arXiv.2406.01756
[26] J.M. Lewis, M. Yannakakis, The node-Deletion problem for hereditary properties is NP-Complete, J. Comput. Syst. Sci. 20 (2) (1980) 219–230. https://doi.org/10.1016/0022-0000(80)90060-4
[27] M. Bannach, F. Chudigiewitsch, T. Tantau, On the descriptive complexity of vertex deletion problems, in: R. Královic, A. Kucera (Eds.), 49th International Symposium on Mathematical Foundations of Computer Science, MFCS 2024, August 26-30, 2024, Bratislava, Slovakia, 306 of *LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024, pp. 17:1–17:14. https://doi.org/10.4230/LIPICS.MFCS.2024.17
[28] L.J. Stockmeyer, The polynomial-Time hierarchy, Theor. Comput. Sci. 3 (1) (1976) 1–22. https://doi.org/10.1016/0304-3975(76)90061-X
[29] G.J. Woeginger, The trouble with the second quantifier, 4OR 19 (2) (2021) 157–181. https://doi.org/10.1007/s10288-021-00477-y
[30] B. Johannes, New classes of complete problems for the second level of the polynomial hierarchy (2011).
[31] V. Rutenburg, Propositional truth maintenance systems: classification and complexity analysis, Ann. Math. Artif. Intell. 10 (3) (1993) 207–232. https://doi.org/10.1007/BF01530952
[32] S. Denegre, Interdiction and discrete bilevel linear programming, Ph.D. thesis, USA, 2011. AAI3456385.
[33] A. Nabli, M. Carvalho, P. Hosteins, Complexity of the multilevel critical node problem, J. Comput. Syst. Sci. 127 (2022) 122–145. https://doi.org/10.1016/J.JCSS.2022.02.004
[34] C.H. Papadimitriou, Computational Complexity, Addison-Wesley, 1994.
[35] R.G. Jeroslow, The polynomial hierarchy and a simple model for competitive analysis, Math. Program. 32 (2) (1985) 146–164. https://doi.org/10.1007/BF01586088

[36] M.R. Garey, D.S. Johnson, R.E. Tarjan, The planar Hamiltonian circuit problem is NP-Complete, SIAM J. Comput. 5 (4) (1976) 704–714. https://doi.org/10.1137/0205049
[37] C. Grüne & L. Wulf, Completeness in the polynomial hierarchy for many natural problems in bilevel and robust optimiza tion, N. Megow & A. BasuInteger Programming and Combinatorial Optimization- 26th International Conference, IPCO 2025, Baltimore, MD, USA 15620 of Springer (2025) 256–269. Lecture Notes in Computer Science. https://doi.org/10.1007/978-3-031-93112-3_19
[38] T. Akiyama, T. Nishizeki, N. Saito, NP-Completeness of the Hamiltonian cycle problem for bipartite graphs, J. Inf. processing 3 (2) (1980) 73–76.
[39] J. Plesník, The NP-Completeness of the Hamiltonian cycle problem in planar digraphs with degree bound two, Inf. Process. Lett. 8 (4) (1979) 199–201. https://doi.org/10.1016/0020-0190(79)90023-1
[40] M. Jackiewicz, A. Kasperski, P. Zieliński, Computational complexity of the recoverable robust shortest path problem with discrete recourse, arXiv preprint arXiv:2403.20000 (2024).