A Note on the Complexity of Defensive Domination

Steven Chaplick
Maastricht University, The Netherlands

Grzegorz Gutowski
Institute of Theoretical Computer Science, Faculty of Mathematics and Computer Science, Jagiellonian University, Kraków, Poland

Tomasz Krawczyk
Faculty of Mathematics and Information Science, Warsaw University of Technology, Poland

Abstract

In a graph G, a *k-attack* A is any set of at most k vertices and *ℓ-defense* D is a set of at most ℓ vertices. We say that defense D *counters* attack A if each $a \in A$ can be matched to a distinct *defender* $d \in D$ with a equal to d or a adjacent to d in G. In the *defensive domination problem*, we are interested in deciding, for a graph G and positive integers k and ℓ given on input, if there exists an ℓ-defense that counters every possible k-attack on G. Defensive domination is a natural resource allocation problem and can be used to model network robustness and security, disaster response strategies, and redundancy designs.

The defensive domination problem is naturally in the complexity class Σ_2^{P}. The problem was known to be NP-hard in general, and polynomial-time algorithms were found for some restricted graph classes. In this note, we prove that the defensive domination problem is Σ_2^{P}-complete.

We also introduce a natural variant of the defensive domination problem in which the defense is allowed to be a multiset of vertices. This variant is also Σ_2^{P}-complete, but we show that it admits a polynomial-time algorithm in the class of interval graphs. A similar result was known for the original setting in the class of proper interval graphs.

2012 ACM Subject Classification Theory of computation

Keywords and phrases graph domination, computational complexity

Digital Object Identifier 10.4230/LIPIcs.MFCS.2025.35

Funding *Grzegorz Gutowski*: Partially supported by grant no. 2023/49/B/ST6/01738 from National Science Centre, Poland.
Tomasz Krawczyk: Partially supported by grant no. 2024/53/B/ST6/02558 from National Science Centre, Poland.

1 Introduction

All graphs discussed in this paper are finite and simple. The vertex set and edge set of a graph G are denoted by $V(G)$ and $E(G)$. For a subset $U \subseteq V(G)$, $G[U]$ denotes the subgraph of G *induced by U*, and $G \setminus U$ denotes the subgraph $G[V(G) \setminus U]$, which is shortened to $G \setminus v$ when $U = \{v\}$. The *neighborhood* of a vertex v, denoted by $N_G(v)$, comprises of vertices adjacent to v and the *closed neighborhood* of v is $N_G[v] = N_G(v) \cup \{v\}$. The *closed neighborhood* and the *neighborhood* of a set $U \subseteq V(G)$ of vertices are defined as $N_G[U] = \bigcup_{v \in U} N_G[v]$ and $N_G(U) = N_G[U] \setminus U$, respectively. The subscript G can be dropped if the graph is clear from the context. A graph on t vertices with all $\binom{t}{2}$ edges is a *clique* K_t. When $G[U]$ is isomorphic to K_t for some set U of vertices, we say that G contains K_t as a subgraph. The clique problem asks for the maximum t such that K_t is contained as a subgraph in a given graph. A set $D \subseteq V(G)$ is a *dominating set* when $N_G[D] = V(G)$. The graph domination problem asks for the minimum size of a dominating set in a given graph and is a classical problem in graph theory and combinatorial optimization. We use the following specification of the problem.

▶ **Problem 1.** DominatingSet

Input: *A graph G and a positive integer ℓ*
Output: Yes *if and only if G admits a dominating set of size at most ℓ*

This problem has broad practical applications in resource allocation, network design, analysis and security. It is also of theoretical interest, as it is one of the first problems known to be NP-complete, see [7], and is used as a base for countless reductions. Consult [8, 9, 10] for various versions and applications of the domination problem.

Graph domination can be understood through the analogy in which the vertices of a graph are under threat of some attack and *defenders* need to be placed in the vertices so that each vertex either has a defender stationed directly in it or in an adjacent vertex. This concept is useful in network security, facility location problems (positioning service centers), and disaster response strategies (deploying rescue teams). Presented in this way, the DominatingSet problem looks for a minimum number of defenders that can counter any attack on a single vertex.

Farley and Proskurowski [6] proposed the following extension of the problem, called *defensive domination*, where we prepare for a simultaneous attack on at most k vertices. We say that any set A of at most k distinct vertices in G is a *k-attack*. An *ℓ-defense* D is a set of at most ℓ distinct vertices of G and corresponds to placing ℓ defenders, one in each vertex of D. We say that defense D *counters* attack A if there is a matching between A and D such that each $a \in A$ is matched to a distinct defender $d \in D$ with a equal to d or a adjacent to d in G. A defense D that counters every possible k-attack in G is called a *k-defensive dominating set*. This extension is natural and meets the redundancy requirements usual for all applications of the domination problem. We use the following formal specification of the problem.

▶ **Problem 2.** DefensiveDominatingSet

Input: *A graph G and positive integers k and ℓ*
Output: Yes *if and only if G admits an ℓ-defense that counters every k-attack in G*

The parametrized version of the problem, where the size of the attack is an external parameter and not a part of the input is also of interest.

▶ **Problem 3.** k-DefensiveDominatingSet

Input: *A graph G and a positive integer ℓ*
Output: Yes *if and only if G admits an ℓ-defense that counters every k-attack in G*

Observe that k-DefensiveDominatingSet for $k = 1$ is exactly DominatingSet. Dereniowski, Gavenčiak, and Kratochvíl [3] proposed a further extension of the problem, which seems a bit technical, but was successfully applied to study a variant of the cops and robbers game. The proposed extension of defensive domination allows the placement of multiple defenders in a single vertex of the graph and limits the possible attacks to the ones that are explicitly specified on the input. A *multiset ℓ-defense* D places ℓ defenders in total at the vertices of G, each vertex getting as many defenders as its multiplicity in D. Multiset defense D counters attack A if each $a \in A$ can be matched to a distinct defender stationed in a, or any vertex adjacent to a. The formal specification of the problem proposed by Dereniowski, Gavenčiak, and Kratochvíl [3] follows.

▶ **Problem 4.** $\mathcal{A}$-DefensiveDominatingMultiset

Input: *A graph G, a set of attacks $\mathcal{A} \subseteq 2^{V(G)}$, multisets D_1 and D_2 of vertices of G, and a positive integer ℓ*

Output: Yes *if and only if G admits a multiset ℓ-defense D with $D_1 \subseteq D \subseteq D_2$ that counters every attack $A \in \mathcal{A}$*

Observe, that we do not allow for multiset attacks, as it would lead to a different problem.

We believe that allowing for multiset defenses is an interesting extension and allows for various applications. We propose the following natural extension of the defensive domination problem, that allows for multiset defenses, but drops other technical conditions introdued by Dereniowski, Gavenčiak, and Kratochvíl.

▶ **Problem 5.** DefensiveDominatingMultiset

Input: *A graph G and positive integers k and ℓ*

Output: Yes *if and only if G admits a multiset ℓ-defense that counters every k-attack in G*

To illustrate the strength of this extension, consider the following example. Graph $K_{1,t}$ is a complete bipartite graph with bipartition classes of sizes 1 and t. Note that any 2-defensive dominating set on $K_{1,t}$ has at least t defenders, but it is enough to use 2 defenders in the multiset setting. In the proof of the main result of this paper, we focus on DefensiveDominatingSet, but the hardness also applies to DefensiveDominatingMultiset, which we believe should attract more attention.

As DominatingSet is NP-complete, it is straightforward that all mentioned domination problems are NP-hard. Both of the problems k-DefensiveDominatingSet and $\mathcal{A}$-DefensiveDominatingMultiset are in fact NP-complete, since one can guess a defense and check if it counters all possible attacks in polynomial time (for k-DefensiveDominatingSet there are $O\left(n^k\right)$ possible attacks to check which is polynomial in n when k is fixed). On the other hand, as DefensiveDominatingSet is naturally expressed as:

$$\exists_{D \subseteq V(G), |D| \leqslant \ell} : \quad \forall_{A \subseteq V(G), |A| \leqslant k} : \quad D \text{ counters } A \text{ in } G,$$

we easily get that DefensiveDominatingSet is in the second level of polynomial hierarchy class Σ_2^{P}. Consult the textbook by Arora and Barak [1, Chapter 5] for an introduction of the polynomial hierarchy. Schaefer and Umans [13, 14, 15] give an extensive list of complete problems for different classes in the polynomial hierarchy. For a very brief introduction, Σ_2^{P} is defined as $\mathsf{NP}^{\mathsf{NP}}$ – a class of languages decidable in polynomial time by nondeterministic Turing machines with access to NP-oracle, where NP-oracle allows to test any language in NP in a single step of execution. The canonical complete problem for Σ_2^{P} is the following.

▶ **Problem 6.** Existential-2-Level-SAT

Input: *Formula $\varphi(x_1, \ldots, x_a, y_1, \ldots, y_b)$ with variables in two disjoint sets $\{x_1, \ldots, x_a\}$ and $\{y_1, \ldots, y_b\}$*

Output: Yes *if and only if the following Boolean formula is true.*

$$\exists_{x_1, x_2, \ldots, x_a} : \quad \forall_{y_1, y_2, \ldots, y_b} : \quad \varphi(x_1, \ldots, x_a, y_1, \ldots, y_b)$$

It was indpendently proved by Stockmeyer [16] and Wrathall [17] that the class Σ_2^{P} is exactly the class of languages reducible to Existential-2-Level-SAT via polynomial-time many-one reductions. Clearly, the following nondeterministic algorithm using NP-oracle solves

DefensiveDominatingSet: algorithm first guesses a set of defenders D and then uses NP-oracle to test whether there exists an attack of size at most k not defended by D. This fact makes Σ_2^{P} a natural complexity class for DefensiveDominatingSet problem.

Ekim, Farley, and Proskurowski [4] showed that DefensiveDominatingSet is unlikely to be in NP. The reason is that the following problem that corresponds to checking if a given defense D counters every k-attack on G is already co-NP-complete.

▶ **Problem 7.** GoodDefense

Input: *A graph G, a subset D of vertices, and a positive integer k*
Output: Yes *if and only if defense D counters every k-attack in G*

For any defense set, or multiset, D, and any set of vertices $X \subseteq V(G)$, let $\mathrm{count}_D(X)$ denote the number of elements (counting multiplicities for multisets) in $D \cap X$. The following connection between defensive domination and Hall's condition was already observed in [4].

▶ **Observation 1** (Ekim, Farley, Proskurowski [4])**.** *The following conditions are equivalent:*

- *Defense D counters every k-attack in G.*
- *For every k-attack A we have $|A| \leqslant \mathrm{count}_D(N[A])$.*

This draws our attention to the complementary problem of GoodDefense and a very similar problem that is known to be NP-complete and W[1]-hard.

▶ **Problem 8.** BadDefense

Input: *A graph G, a subset D of vertices, and a positive integer k*
Output: Yes *if there exists k-attack A with $|A| > \mathrm{count}_D(N[A])$*

▶ **Problem 9.** HallSet

Input: *A bipartite graph G with bipartition classes U and W and a positive integer k*
Output: Yes *if and only if there exists $X \subseteq U$ with $|N_G(X)| < |X| \leqslant k$*

In [2, Exercise 13.28], a parametrized reduction is given from Clique to HallSet. HallSet has an easy parametrized reduction to BadDefense. All of these observations allow for the following conclusion.

▶ **Lemma 2** (Theorem 2.3 in Ekim, Farley, Proskurowski [4])**.** *GoodDefense is* co-NP-*complete. BadDefense is* NP-*complete and* W[1]-*hard when parametrized by k.*

When k is an external parameter of the problem, k-DefensiveDominatingSet is in NP, and it remains NP-complete even when the input graph is restricted to split graphs [4], or bipartite graphs [11]. On the positive side, DefensiveDominatingSet admits polynomial-time algorithms when the input graph is restricted to cliques, cycles, trees [6], co-chain graphs, threshold graphs [4], or proper interval graphs [5].

The main result of this paper is the following.

▶ **Theorem 3.** *DefensiveDominatingSet and DefensiveDominatingMultiset are Σ_2^{P}-complete.*

The introduced multiset setting not only may better fit some applications, but might also be more approachable algorithmically. For example, in Section 3 we investigate the multiset defensive domination problem on the class of interval graphs. A graph G is an *interval graph* when each vertex $v \in V(G)$ corresponds to a closed interval $I_v \subseteq \mathbb{R}$, and $\{u, v\} \in E(G)$ if and only if $I_u \cap I_v \neq \emptyset$. We prove the following.

▶ **Theorem 4.** *DefensiveDominatingMultiset is in* P *when the input graphs are restricted to the class of interval graphs.*

A similar result for DefensiveDominatingSet was shown for proper interval graphs by Ekim, Farley, Proskurowski, and Shalom [5]. The complexity of DefensiveDominatingSet for interval graphs remains unknown.

The proof of the main theorem is based on a reduction of the following problem, which was shown to be Σ_2^{P}-complete by Rutenburg [12].

▶ **Problem 10.** CliqueNodeDeletion

Input: *A graph G and positive integers s and t*

Output: Yes *if and only if G admits a set X of at most s vertices such that $G \setminus X$ does not contain K_t as a subgraph*

▶ **Theorem 5** (Theorem 6 in Rutenburg [12])**.** *CliqueNodeDeletion is Σ_2^{P}-complete.*
As the original paper includes only an idea of the proof that requires some minor alterations, we have decided to present a streamlined proof of Theorem 5 in Appendix A.

2 Main Result

We are ready for the proof of the main result.

▶ **Theorem 3.** *DefensiveDominatingSet and DefensiveDominatingMultiset are Σ_2^{P}-complete.*

Proof. We present a reduction from CliqueNodeDeletion. Assume that we are given an instance G, s, t, where s is the number of vertices to remove from G and t is the size of clique to avoid as a subgraph. For technical reasons, we assume that $t \geqslant 4$. Let $n = |V(G)|$ denote the number of vertices in G.

We construct an equivalent instance G', k, ℓ of DefensiveDominatingSet. We set the maximum size of an attack $k = n+s$, the maximum size of a defense $\ell = 4(n+s)+nt-(t-1)$, and construct the graph G' as depicted in Figure 1:

- For each vertex $v \in V(G)$, we introduce two vertices v' and v'' representing v in G'. Set W denotes vertices v', v'' introduced for all $v \in V(G)$.
- For each edge $e = (u, v)$ in $E(G)$, we introduce the vertex e' and add the edges joining e' with four vertices u', u'', v', v'' in W. Set F denotes vertices e' introduced for all $e \in E(G)$.
- We introduce four independent sets: I_1 of size $n + s$; I_2 of size $n + s - \binom{t}{2}$; I_3 of size $n + s$; and I_4 of size $n + s + \ell$.
- We introduce three cliques: Q_1 of size $n + s$; Q_2 of size $n + s - (t + 1)$; and Q_3 of size $n + s$.
- For each vertex $v \in V(G)$, we introduce a complete bipartite graph with one bipartition class I_v of size $\binom{t}{2}$, and the other class I'_v of size t. Set I_V denotes the union of sets I_v for all $v \in V(G)$ and I'_V denotes the union of sets I'_v for all $v \in V(G)$.
- We add edges of complete bipartite graphs given by the biparition classes: (Q_1, I_1); $(Q_2, Q_1 \cup I_2 \cup F \cup I_V)$; (Q_3, I_3); $(W, Q_3 \cup I_4)$; and $(\{v', v''\}, I_v)$ for each $v \in V(G)$.

We claim that s vertices can be removed from G so that the resulting graph does not include K_t as a subgraph if and only if there is an ℓ-defense that counters every k-attack in G'. Before presenting the proof, let us give some ideas on the role of different "gadgets" used

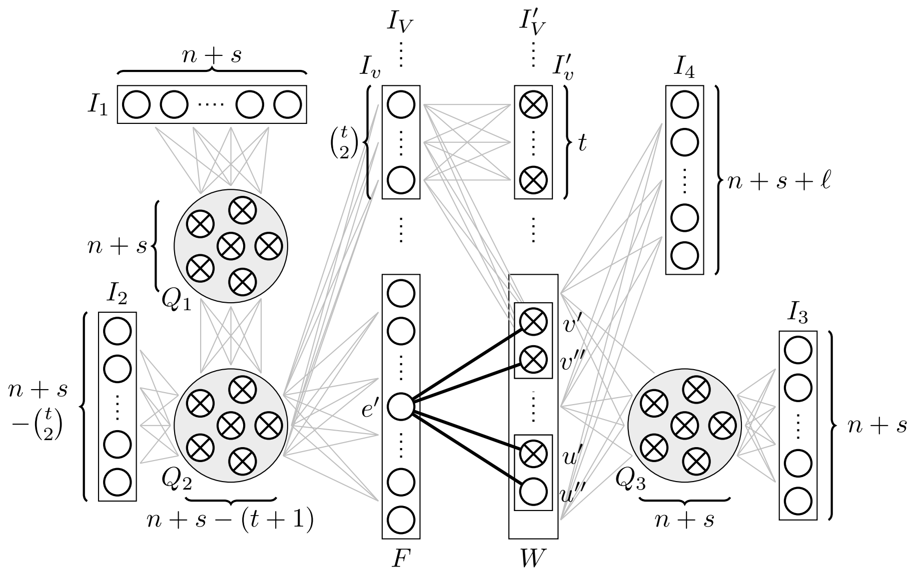

Figure 1 Reduction: The edge $e = (u, v)$ of G is represented by the edges between e' and v', v'', u', u'' in G'. The defenders are represented by crossed circles.

in the construction of G'. The sets I force any successful defense to position many defenders in their neighborhood. This allows for good control over the degrees of freedom in successful ℓ-defenses. The set W contains two copies of each vertex of G and the defense is forced to position at least one defender on them. We use the possibility of choosing some of the second copies in each pair to express the selection of a clique hitting set. We are able to limit the "dangerous" choices of k-attacks so that they express the choice of edges of a clique K_t in the set F. We independently prove the implications in both directions.

($\Rightarrow$) Let X be a solution to the instance G, s, t. We have $|X| = s$ and $G \setminus X$ does not include K_t as a subgraph. We construct the following defense D:

$$D = Q_1 \cup Q_2 \cup Q_3 \cup I'_V \cup \{v' : v \in V(G)\} \cup \{v'' : v \in X\}.$$

Note that D positions exactly $n + s$ defenders on W. Additionally, D has nt defenders on I'_V and $3(n + s) - (t + 1)$ defenders on $Q_1 \cup Q_2 \cup Q_3$, which gives ℓ defenders in total. We show that D counters every attack of size at most $n + s$. Suppose for a contradiction that A is an inclusion minimal attack with $\text{count}_D(N[A]) < |A| \leqslant n + s$. First, we observe that every vertex in the sets W, I_1, I_3, I_4, Q_1, Q_2, and Q_3 is adjacent to at least $n + s$ defenders (either from the set Q_1, Q_2, Q_3 or W). If any of these vertices is included in A, then $\text{count}_D(N[A]) \geqslant n + s \geqslant |A|$. We conclude that

$$A \subseteq I_2 \cup F \cup I_V \cup I'_V.$$

As every vertex in I'_V is in D, we do not have $A \subseteq I'_V$ and hence $A \cap (I_2 \cup F \cup I_V) \neq \emptyset$. In particular, $Q_2 \subseteq N[A]$, and hence $\text{count}_D(N[A]) \geqslant |Q_2| = n + s - (t + 1)$. Since every vertex in I_V is adjacent to additional $t + 1$ defenders in $W \cup I'_V$, we have $A \cap I_V = \emptyset$. Now, $N[I'_V] \cap A = I'_V \cap A$, and $I'_V \subseteq D$, so by the minimality of A we get $A \cap I'_V = \emptyset$. We conclude that

$$A \subseteq I_2 \cup F.$$

As $t+1 < \binom{t}{2}$ for $t \geqslant 4$ and $|A| > \mathrm{count}_D(N[A]) \geqslant n+s-(t+1) > n+s-\binom{t}{2} = |I_2|$, we also have $A \cap F \neq \emptyset$. As A includes at least one vertex in F, $N[A]$ includes at least two vertices in $W \cap D$ and $|A| > n+s-(t+1)+2$. We conclude that

$$|A \cap F| \geqslant |A| - |I_2| > n+s-(t+1)+2-\left(n+s-\binom{t}{2}\right) = \binom{t}{2} - (t-1) = \binom{t-1}{2}.$$

Now, as A includes more than $\binom{t-1}{2}$ vertices in F, $N[A]$ includes at least t vertices in $W \cap D$, and $|A| > n+s-(t+1)+t$. Hence, $|A| = n+s$ and $I_2 \subsetneq A$, as otherwise we would have $\mathrm{count}_D(N[A]) > n+s$. We call an attack A to be *serious* if

$$I_2 \subsetneq A \subseteq I_2 \cup F, \quad |A \cap F| = \binom{t}{2}.$$

We have shown that attack A is serious. Any $\binom{t}{2}$ edges in G span at least $t+1$ vertices or span t vertices that form a clique K_t in G. As we know that $G \setminus X$ does not contain K_t as a subgraph, we get that if the edges span only t vertices then at least one of them is in X. In either case, we get $\mathrm{count}_D(N[A]) \geqslant n+s-(t+1)+(t+1) = n+s$ and D counters A.

($\Leftarrow$) Let D be an ℓ-defense that counters every k-attack in G'. We make the following observations.

1. As D counters attack I_1, there are at least $n+s$ defenders in $I_1 \cup Q_1$.
2. As D counters attack I_3, there are at least $n+s$ defenders in $I_3 \cup Q_3$.
3. As D counters every possible $(n+s)$-attack in $I_4 \setminus D$, there are at least $n+s$ defenders in W.
4. For every $v \in V(G)$, as D counters attack I'_v, there are at least t defenders in $I_v \cup I'_v$. Thus, in total, there are at least nt defenders in $I_V \cup I'_V$.
5. By calculation, there are at most $n+s-(t+1)$ defenders in $I_2 \cup Q_2 \cup F$.
6. For every $v \in V(G)$, as D counters attack $I_2 \cup I_v$ and there are at most $n+s-(t+1)$ defenders in $I_2 \cup Q_2$, there are at least $t+1$ defenders in $I_v \cup I'_v \cup \{v', v''\}$. As $|I'_v| = t$, at least one of the defenders is in $I_v \cup \{v', v''\}$.

We now construct a modified defense D' and claim that D' counters every serious attack on G. As we focus on serious attacks, and the defenders in I_V are not used to counter any serious attack, we move some of them. For each $v \in V(G)$ if $v' \notin D$ and $v'' \notin D$, we move one defender from I_v to v' (guaranteed to be there by point 6). Observe that even after this move there are at least t defenders in $I_v \cup I'_v$. As there are still at least $nt + (n+s) + (n+s)$ defenders in $I_V \cup I'_V \cup I_1 \cup Q_1 \cup I_3 \cup Q_3$, there are at most $(n+s) + (n+s-(t+1)$ in $I_2 \cup Q_2 \cup F \cup W$. Second, since serious attacks include only vertices in $I_2 \cup F$, and vertices in Q_2 dominate $I_2 \cup F$, and there are at most $n+s-(t+1)$ defenders in $I_2 \cup Q_2 \cup F$, we move all defenders from I_2 and F to Q_2. Third, while $|D \cap W| > n+s$, we select any $v \in V(G)$ with defenders both in v' and v'' and move the defender from v'' to Q_2. The resulting defense D' has the property that it also counters every serious attack, as D did.

The resulting defense D' has exactly $n+s$ defenders in W, at most $n+s-(t+1)$ defenders in Q_2, at least one defender in each $\{v', v''\}$ for every $v \in V(G)$, and counters every serious attack. Let X be a set of vertices v in $V(G)$ for which both v' and v'' are in D'. There are exactly s such vertices. We claim that $G \setminus X$ does not contain K_t as a subgraph. Indeed, given a t-element set Q of vertices of G with $Q \cap X = \emptyset$ and $G[Q]$ isomorphic to K_t, we can create a k-attack composed of I_2 and the set of $\binom{t}{2}$ vertices e' representing edges of $G[Q]$. This attack has size $n+s$ and has at most $n+s-(t+1)$ neighboring defenders in Q_2 and exactly t neighboring defenders in W. Thus, this serious attack is not countered by D', which contradicts the construction of D'.

We leave it to the reader to verify that exactly the same reduction also shows that DefensiveDominatingMultiset is also a Σ_2^{P}-complete problem. ◀

3 Interval Graphs

Building on the work of Ekim, Farley, Proskurowski, and Shalom [5], who presented a greedy algorithm for DefensiveDominatingSet on proper interval graphs, we develop a similar greedy strategy for DefensiveDominatingMultiset on general interval graphs.

For the remainder of this section, let G be an interval graph given by its interval representation: each vertex $v \in V(G)$ corresponds to a closed interval $I_v \subseteq \mathbb{R}$, and $\{u, v\} \in E(G)$ if and only if $I_u \cap I_v \neq \emptyset$. We assume that this representation ensures that no two distinct intervals share an endpoint. For any bounded closed set $S \subseteq \mathbb{R}$, let $\mathrm{left}(S)$ and $\mathrm{right}(S)$ denote the minimum and the maximum element in S, respectively. For any set (or multiset) of intervals X, let $\mathrm{sum}(X) = \bigcup_{I \in X} I$ denote their union, and let $\mathrm{span}(X)$ be the minimum closed interval containing every interval in X, that is, the interval $[\mathrm{left}(X), \mathrm{right}(X)]$. For a set $Y \subseteq V(G)$, we define $\mathrm{sum}(Y) = \mathrm{sum}(\{I_v : v \in Y\})$ and $\mathrm{span}(Y) = \mathrm{span}(\{I_v : v \in Y\})$. A set (or multiset) of intervals X is *proper* if no interval in X is a proper subset of another interval in X. We say that a set $Y \subseteq V(G)$ is proper, when $\{I_v : v \in Y\}$ is proper. The algorithm presented by Ekim, Farley, Proskurowski, and Shalom [5] for DefensiveDominatingSet worked under the condition that $V(G)$ is proper.

Our first observation is that in the multiset setting we can focus on constructing proper defenses.

▶ **Observation 6.** *For any multiset defense D, there exists a proper multiset defense D' such that $|D'| = |D|$ and D' counters any attack that D counters.*

Proof. Consider a multiset defense D. If it is not proper, then there exist two vertices $u, v \in D$ such that $I_u \subsetneq I_v$. Let D' be the multiset defense obtained from D by replacing every copy of u with an additional copy of v. Clearly, $|D'| = |D|$. Since $I_u \subsetneq I_v$, we have that $N[u] \subseteq N[v]$ and that any attack A that is countered by D using defenders in u is countered by D' using added defenders in v. Therefore, D' counters every attack countered by D. Observe that this replacement increases the total length of the intervals that represent vertices in the defense. Thus, repeating this replacement procedure eventually stops and yields a proper defense that counters any attack that the original defense counters. ◀

Note that Observation 6 holds specifically for the multiset setting and does not have a direct analogue for DefensiveDominatingSet. The next observation is that sets with smaller union are more dangerous for any defense than those with larger union.

▶ **Observation 7.** *For any defense D and two sets A_1, A_2 with $\mathrm{count}_D(N[A_1]) < |A_1|$, $|A_1| \leq |A_2|$, and $\mathrm{sum}(A_2) \subseteq \mathrm{sum}(A_1)$ we have $\mathrm{count}_D(N[A_2]) < |A_2|$.*

Proof. We have that $\mathrm{count}_D(N[A_2])$ is the number of intervals in D that have a nonempty intersection with $\mathrm{sum}(A_2)$ which is a subset of $\mathrm{sum}(A_1)$. Thus,

$$\mathrm{count}_D(N[A_2]) \leq \mathrm{count}_D(N[A_1]) < |A_1| \leq |A_2| \,.$$ ◀

Let $x = \mathrm{right}(I_v)$ for some vertex $v \in V(G)$. Let $V_x = \{v \in V(G) : \mathrm{right}(I_v) \leq x\}$ denote the nonempty set of vertices that lie completely to the left of x in the representation. Let $c = |V_x|$ and for every integer $1 \leq i \leq c$ we define the *i-th block* at x, denoted $B_{x,i}$, in the following way. Let $v_1, v_2, \ldots, v_c$ be V_x arranged in a sequence sorted descending by the left endpoint of the representing interval, that is, $\mathrm{left}(I_u) > \mathrm{left}(I_w)$ if and only if u appears earlier than w in the sequence. We select $B_{x,i} = \{v_1, v_2, \ldots, v_i\}$ to be the first i elements in the sequence. Note that if defined, $B_{x,i}$ contains exactly i vertices, $B_{x,i+1}$ is a superset of $B_{x,i}$, and $B_{x,i}$ maximizes $\mathrm{left}(\mathrm{span}(X))$ among all subsets $X \subseteq V_x$ with $|X| = i$.

We say that D is a *k-block* defense if it counters every attack $B_{x,i}$ for every possible right endpoint x and every $i \leq k$.

▶ **Lemma 8.** *A proper k-block defense D counters every k-attack in G.*

Proof. Assume to the contrary that a proper k-block defense D does not counter some k-attack. By Observation 1 we have a set A with $\mathrm{count}_D(N[A]) < |A| = m \leq k$. Among such sets, we select A with the minimum size m.

Let $w \in A$ be the vertex that maximizes $\mathrm{right}(I_w)$ among the vertices in A. Let $x = \mathrm{right}(I_w)$, $c = |V_x|$, and $v_1, v_2, \ldots, v_c$ be V_x arranged in a sequence sorted descending by the left endpoint of the representing interval. All elements of A are represented to the left of x, so we have $m \leq c$. Recall that $B_{x,i} = \{v_1, v_2, \ldots, v_i\}$ for every $1 \leq i \leq c$. By the assumption of the lemma, we know that $A \neq B_{x,m}$. Since D counters $B_{x,m}$, we have $\mathrm{count}_D(N[A]) < \mathrm{count}_D(N[B_{x,m}])$ and there is $d \in D$ with $d \in N[B_{x,m}]$ and $d \notin N[A]$. Every interval representing a vertex in A is either completely to the left or completely to the right of I_d. Let

$$A_1 = \{v \in A : \mathrm{right}(I_v) < \mathrm{left}(I_d)\} \quad \text{and} \quad A_2 = \{v \in A : \mathrm{right}(I_d) < \mathrm{left}(I_v)\}.$$

Since $x \in I_w$ and $x \notin I_d$, we have $w \in A_2$. Since $\mathrm{left}(\mathrm{span}(A)) < \mathrm{left}(\mathrm{span}(B_{x,m}))$, we have $A_1 \neq \emptyset$. We have partitioned A into two nonempty subsets A_1 and A_2.

By the minimality of A we have $\mathrm{count}_D(N[A_1]) \geq |A_1|$ and $\mathrm{count}_D(N[A_2]) \geq |A_2|$. We get that there is at least one $d' \in D \cap N[A_1] \cap N[A_2]$, as otherwise we would have $\mathrm{count}_D(N[A]) = \mathrm{count}_D(N[A_1]) + \mathrm{count}_D(N[A_2]) \geq |A_1| + |A_2| = |A|$. Interval $I_{d'}$ intersects both $\mathrm{span}(A_1)$ and $\mathrm{span}(A_2)$. This allows us to write the following inequalities:

$$\mathrm{left}(I_{d'}) < \mathrm{right}(\mathrm{span}(A_1)) < \mathrm{left}(I_d) < \mathrm{right}(I_d) < \mathrm{left}(\mathrm{span}(A_2)) < \mathrm{right}(I_{d'}),$$

and conclude that I_d is a proper subset of $I_{d'}$ contradicting with D being a proper defense. ◀

We are ready for the proof of the main result of this section.

▶ **Theorem 4.** *DefensiveDominatingMultiset is in* P *when the input graphs are restricted to the class of interval graphs.*

Proof. The proposed algorithm, see Algorithm 1 for the pseudocode, works as follows. It assumes that G has n vertices and is given in the interval representation by the intervals $I_1, I_2, \ldots, I_n$. It sorts the intervals ascending by their right endpoints. Then, for every $1 \leq i \leq n$, let x be the right endpoint of the i-th interval in the list. The algorithm calculates the number *count* of defenders missing to counter every m-block attack $B_{x,m}$ for $m \leq k$. Then, it selects the interval d that maximizes the right endpoint among all intervals that include x. The algorithm adds *count* copies of the interval d to D.

Clearly, Algorithm 1 runs in polynomial time. The algorithm only adds inclusion maximal intervals to D, so the constructed multiset defense D is proper. The algorithm explicitly adds the required number of defenders to counter every possible attack $B_{x,m}$, so D is also k-block. Thus, by Lemma 8 we get that D counters every k-attack in G.

Algorithm 1 Greedy Multiset Defensive Domination in Interval Graphs.

procedure GreedyDefense($(I_1, I_2, \dots, I_n), k$)
    $D \leftarrow \emptyset$
    $I_1, I_2, \dots, I_n \leftarrow$ intervals $I_1, I_2, \dots, I_n$ sorted ascending by their right endpoints
    for $i = 1, 2, \dots, n$ **do**
        $x = \mathrm{right}(I_i)$
        for $m = 1, 2, \dots, \min(i, k)$ **do**
            $\mathit{count} \leftarrow \max(0, m - \mathrm{count}_D(N[B_{x,m}]))$
            $d \leftarrow$ interval with $\mathrm{left}(d) < x$ and maximum $\mathrm{right}(d)$
            Add *count* copies of d to D
        end for
    end for
    return D
end procedure

For the proof that the defense D is of minimum size, let $\ell = |D|$, $J_1, J_2, \dots, J_\ell$ be the multiset of intervals in D sorted ascending by their left endpoints. Assuming to the contrary, let D' be another proper defense that counters every k-attack with $\ell' = |D'| < |D|$ and $K_1, K_2, \dots, K_{\ell'}$ be the multiset of intervals in D' sorted ascending by their left endpoints. Among such defenses, select one that maximizes the number p such that $J_i = K_i$ for all $1 \le i \le p$. We have $p < \ell' < \ell$. Let $d = J_{p+1}$ and $d' = K_{p+1}$. We do not have $d' \subsetneq d$, as otherwise we could exchange d' for d in D' and get another defense that counters every k-attack of the same size, but with larger p. We also do not have $d \subsetneq d'$, as the algorithm only adds inclusion maximal intervals to D. Let x, m be such that the algorithm added d to D when considering attack $B_{x,m}$.

If $\mathrm{left}(d') > \mathrm{left}(d)$ then $\mathrm{right}(d') > \mathrm{right}(d)$ and as the algorithm did not add d' to D we have $\mathrm{left}(d') > x$. This means that D' has exactly p intervals $K_1, \dots, K_p = J_1, \dots, J_p$ with the left endpoint less than or equal to x. The algorithm calculated that these intervals do not counter $B_{x,m}$ and no other interval in D' intersects $\mathrm{span}(B_{x,m})$. We conclude that D' does not counter $B_{x,m}$, a contradiction.

If $\mathrm{left}(d') < \mathrm{left}(d)$ then $\mathrm{right}(d') < \mathrm{right}(d)$ and we can exchange d' for d in D' and get another defense D'' that counters every k-attack with $|D''| = |D'|$ but with bigger p. Indeed, we can show that the resulting defense D'' satisfies $\mathrm{count}_{D''}(N[B_{y,q}]) \ge q$ for every possible right endpoint y and every $q \le k$. For $y < x$, this follows from the fact that D'' agrees with D on the first p intervals that are enough to counter these attacks. For $y \ge x$, we have $d' \in N[B_{y,q}] \Rightarrow d \in N[B_{y,q}]$ and $\mathrm{count}_{D''}(N[B_{y,q}]) \ge \mathrm{count}_{D'}(N[B_{y,q}]) \ge q$. We get a contradiction with the choice of D'. ◀

We claim that Algorithm 1 can be implemented to run in $O(nk)$ time using standard techniques, and we omit the details of this implementation.

4 Summary

In this work, we have shown that both DefensiveDominatingSet and DefensiveDominatingMultiset are Σ_2^{P}-complete. For the multiset variant of the problem, in the class of interval graphs, we have indicated a polynomial-time algorithm. This algorithm does not work in the original setting where at most one defender can be located at a single vertex. Furthermore, we do not even know if GoodDefense admits a polynomial-time algorithm in the class of interval graphs. We would like to see the complexity status of DefensiveDominatingSet resolved in the class of interval graphs.

We also believe that potential applications in the facility location problem should justify the investigation of defensive domination problems in the class of planar graphs. Note that the reductions presented for Σ_2^{P}-completeness of DefensiveDominatingSet and DefensiveDominatingMultiset or the W[1]-hardness of BadDefense construct graphs with large cliques and cannot be applied to show the hardness in the class of planar graphs. There is a natural dynamic programming FPT-algorithm that checks BadDefense when parametrized by the tree-width of the input graph. When looking for dangerous k-attacks against a fixed defense, it is enough to consider attacks A such that $N[A]$ is connected. This means that in a planar graph, we can fix some outerplanar decomposition of the graph and only consider k-attacks that span at most $2k-1$ adjacent layers of the outerplanar decomposition. As the tree-width of such subgraphs is bounded, we obtain a simple FPT-algorithm that checks BadDefense when parametrized by k in the planar graphs. This shows that in the parametrized sense, defensive domination problems might be easier in planar graphs than they are in general graphs. This motivates the following questions. Does GoodDefense admit a polynomial-time algorithm in the class of planar graphs? What is the complexity of DefensiveDominatingSet and DefensiveDominatingMultiset in the class of planar graphs? Both problems are NP-hard, but we do not know if they are Σ_2^{P}-complete.

References

1 Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

2 Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

3 Dariusz Dereniowski, Tomáš Gavenčiak, and Jan Kratochvíl. Cops, a fast robber and defensive domination on interval graphs. *Theoretical Computer Science*, 794:47–58, 2019. Special Issue on Theory and Applications of Graph Searching. doi:10.1016/j.tcs.2018.09.031.

4 Tınaz Ekim, Arthur M. Farley, and Andrzej Proskurowski. The complexity of the defensive domination problem in special graph classes. *Discrete Mathematics*, 343(2):111665, 2020. doi:10.1016/j.disc.2019.111665.

5 Tınaz Ekim, Arthur M. Farley, Andrzej Proskurowski, and Mordechai Shalom. Defensive domination in proper interval graphs. *Discrete Applied Mathematics*, 331:59–69, 2023. doi:10.1016/j.dam.2023.01.016.

6 Arthur M. Farley and Andrzej Proskurowski. Defensive domination. In *SEICCGTC 2004: 35th Southeastern International Conference on Combinatorics, Graph Theory and Computing*, volume 168 of *Congressus Numerantium*, pages 97–107, 2004.

7 Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

8 Teresa W. Haynes, Stephen T. Hedetniemi, and Michael A. Henning. *Topics in Domination in Graphs*. Developments in Mathematics. Springer, 2020. doi:10.1007/978-3-030-51117-3.

9 Teresa W. Haynes, Stephen T. Hedetniemi, and Michael A. Henning. *Structures of Domination in Graphs*. Developments in Mathematics. Springer, 2021. doi:10.1007/978-3-030-58892-2.

10 Teresa W. Haynes, Stephen T. Hedetniemi, and Michael A. Henning. *Domination in Graphs: Core Concepts*. Springer Monographs in Mathematics. Springer, 2023. doi:10.1007/978-3-031-09496-5.

11 Michael A. Henning, Arti Pandey, and Vikash Tripathi. More on the complexity of defensive domination in graphs. *Discrete Applied Mathematics*, 362:167–179, 2025. doi:10.1016/j.dam.2024.11.023.

12 Vladislav Rutenburg. Complexity of generalized graph coloring. In *MFCS 1986: 12th Symposium on Mathematical Foundations of Computer Science*, volume 233 of *LNCS*, pages 573–581, 1986. doi:10.1007/BFb0016284.

13 Marcus Schaefer and Christopher Umans. Completeness in the polynomial-time hierarchy: Part I: A compendium. *SIGACT News*, 33(3):32–49, 2002. doi:10.1145/582475.582484.

14 Marcus Schaefer and Christopher Umans. Completeness in the polynomial-time hierarchy: Part II. *SIGACT News*, 33(4):22–36, 2002. doi:10.1145/601819.601829.

15 Marcus Schaefer and Christopher Umans. Completeness in the polynomial-time hierarchy: A compendium, 2008. URL: https://ovid.cs.depaul.edu/documents/phcom.pdf.

16 Larry J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, 1976. doi:10.1016/0304-3975(76)90061-X.

17 Celia Wrathall. Complete sets and the polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):23–33, 1976. doi:10.1016/0304-3975(76)90062-1.

A Clique Node Deletion

The main result of this paper is proved by a reduction from CliqueNodeDeletion to DefensiveDominatingSet. Clique Node Deletion Problem was first introduced by Rutenburg [12] in a more general setting called Generalized Node Deletion. Rutenburg gives an idea for a proof [12, Theorem 6] that CliqueNodeDeletion is Σ_2^{P}-complete. As CliqueNodeDeletion is an important intermediate problem for our result, we present a streamlined proof based on Rutenburg's idea.

The proof is based on a reduction from the following problem, which is a variation on the quantified boolean formula satisfaction problem. It is a natural Σ_2^{P}-complete problem with an easy reduction from Existential-2-Level-SAT [16].

▶ **Problem 11.** Existential-2-Level-3-CNF

Input: *3-CNF formula* $\varphi(x_1, \ldots, x_a, y_1, \ldots, y_b)$ *with variables in two disjoint sets* $\{x_1, \ldots, x_a\}$ *and* $\{y_1, \ldots, y_b\}$

Output: Yes *if and only if the following Boolean formula is true.*

$$\exists_{x_1,x_2,\ldots,x_a} : \quad \neg\exists_{y_1,y_2,\ldots,y_b} : \quad \varphi(x_1, \ldots, x_a, y_1, \ldots, y_b)$$

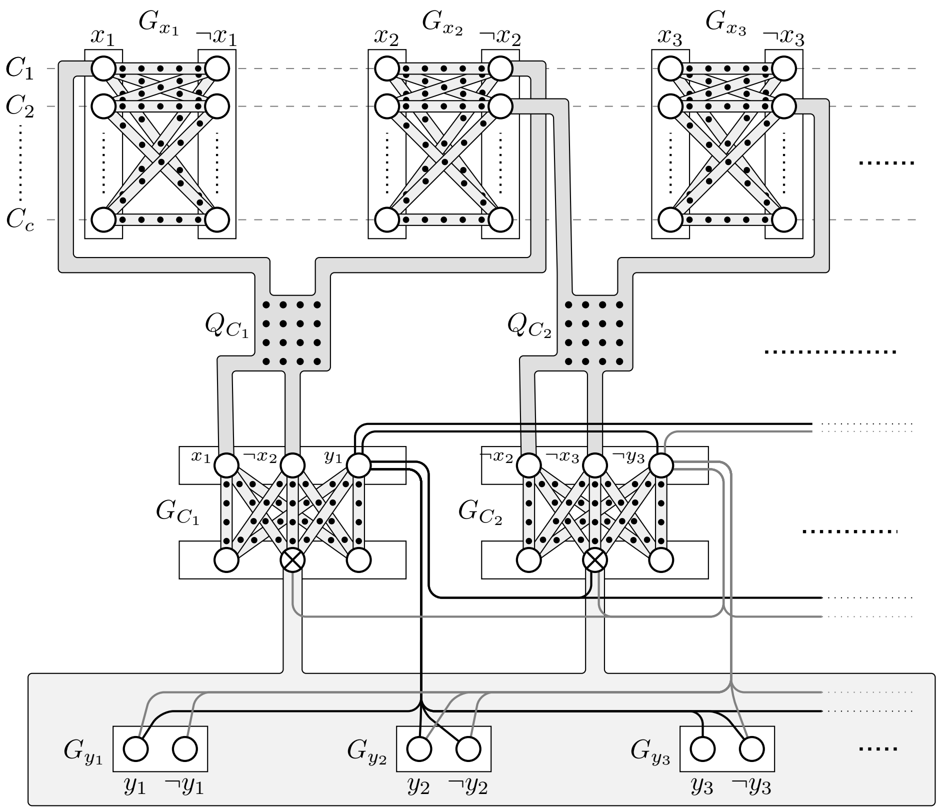

Figure 2 Rutenburg's reduction. The part of G corresponding to the clauses $C_1 \equiv (x_1 \vee \neg x_2 \vee y_1)$ and $C_2 \equiv (\neg x_2 \vee \neg x_3 \vee \neg y_3)$. Ugly vertices are represented by crossed circles.

We are now ready to present the proof of the following theorem.

▶ **Theorem 5** (Theorem 6 in Rutenburg [12]). *CliqueNodeDeletion is* Σ_2^{P}*-complete.*

Proof. We reduce from Existential-2-Level-3-CNF to CliqueNodeDeletion. Assume that we are given two disjoint sets of variables $X = \{x_1, \ldots, x_a\}$, $Y = \{y_1, \ldots, y_b\}$, and a set of clauses $\mathcal{C} = \{C_1, \ldots, C_c\}$ with each clause having exactly three occurrences of three distinct variables in $X \cup Y$. For technical reasons, we assume that $c > 6$, as otherwise there are at most 18 variables and we can simply check all the possibilities. We set $s = ac + 3c$, $t = b + c$, and construct a graph G such that G admits a subset X of vertices with $|X| \leq s$ such that $G \setminus X$ does not contain K_t as a subgraph if and only if the following formula

$$\exists_{x_1, x_2, \ldots, x_a} : \quad \neg\exists_{y_1, y_2, \ldots, y_b} : \quad C_1 \wedge C_2 \wedge \ldots \wedge C_c$$

is true. The graph G is constructed in the following steps. Consult Figure 2 for an example.

1. For each variable x_i $(1 \leqslant i \leqslant a)$ we introduce G_{x_i}: a copy of $K_{c,c}$ with one bipartition class called *positive* and the other class called *negative*. We number positive vertices from 1 to c and number negative vertices from 1 to c.
2. For each variable y_j $(1 \leqslant j \leqslant b)$ we introduce G_{y_j}: two independent vertices, one called *positive* and the other called *negative*.
3. For each clause C_k $(1 \leqslant k \leqslant c)$ we introduce G_{C_k}: a copy of $K_{3,3}$ with one bipartition class called *good* and the other class called *bad*. We select one of the bad vertices to be *ugly*. In C_k there are 3 occurrences of variables. For each such occurrence, we assign a

distinct good vertex in G_{C_k}. We call the good vertices assigned to variables in X (Y) to be *X-good* (*Y-good*).

4. For each edge (u, v) in every G_{x_i} and every G_{C_k} we add a copy of K_t on u, v and additional $t-2$ new vertices.
5. For each G_{C_k} with g X-good vertices we compose set Z of all X-good vertices in G_{C_k} and every positive (negative) vertex numbered k in G_{x_i} with x_i having a positive (negative) occurrence in C_k. Set Z has exactly $2g$ elements. We add a copy of K_{t-1+g} on vertices in Z and additional $t-1-g$ new vertices. We call this added clique Q_{C_k}.
6. For each positive (negative) occurrence of y_j in C_k, let u be the Y-good vertex in G_{C_k} assigned to this occurrence. We add an edge between u and the positive (negative) vertex in G_{y_j}. We also add edges between u and both vertices in every other $G_{y_{j'}}$ for $j' \neq j$.
7. For each clause C_k, let u be the ugly vertex in G_{C_k}. We add edges between u and both vertices in every G_{y_j}.
8. For every $1 \leqslant j < j' \leqslant b$, we add all four edges between any of the two vertices in G_{y_j} and any of the two vertices in $G_{y_{j'}}$.
9. Finally, for every $1 \leqslant k < k' \leqslant c$, every bad or Y-good vertex u in G_{C_k}, and every bad or Y-good vertex u' in $G_{C_{k'}}$ we add an edge between u and u'.

Observe that G contains many copies of K_t. We distinguish four types of such cliques:

(A) A clique that contains a positive and a negative vertex in a single G_{x_i}. The common neighborhood of such a pair of vertices is the set of additional vertices added in step 4 of the construction.

(B) A clique that contains a good and a bad vertex in a single G_{C_k}. The common neighborhood of such a pair of vertices is the set of additional vertices added in step 4 of the construction.

(C) A clique that contains an X-good vertex in some G_{C_k} and is not of type B. As this clique does not contain any bad vertex in G_{C_k}, it must be a subclique of Q_{C_k} constructed in step 5.

(D) Other. Observe that every other clique of size t can contain only Y-good or ugly vertices in G_{C_j} or positive or negative vertices in G_{y_i}.

We independently prove the implications in both directions.

($\Leftarrow$) Let $\nu_1, \nu_2, \ldots, \nu_a$ be a valuation of $x_1, x_2, \ldots, x_a$ such that formula

$$\exists_{y_1, y_2, \ldots, y_b} : \quad C_1 \wedge C_2 \wedge \ldots \wedge C_c$$

is false. We construct a set X of vertices in $V(G)$ in the following way. From each G_{x_i} we select all the c negative vertices if ν_i is true and all c positive vertices if ν_i is false. For each C_k if C_k is satisfied by any of the variables $x_i = \nu_i$ we select all 3 good vertices. Otherwise, we select all 3 bad vertices. The constructed set X has $ac + 3c$ vertices. We claim that $G \setminus X$ does not contain K_t as a subgraph. Each clique of type A or B is removed by the selection of vertices. Each clique Q_{C_k} of type C corresponding to a clause C_k with g occurrences of variables in X is removed, as we either remove g vertices corresponding to the false occurrences of variables in variable gadgets, or all good vertices in clause gadgets G_{C_k}.

Now, assuming to the contrary, let Q be some clique of type D that remains in $G \setminus X$. It is clear that Q can have at most one vertex in each G_{y_j} and at most one vertex in each G_{C_k}. As $t = b + c$, Q has exactly one vertex in each G_{y_j} that corresponds to a valuation $\mu_1, \mu_2, \ldots, \mu_b$ of variables $y_1, y_2, \ldots, y_b$. We claim that the combined valuation ($x_i = \nu_i$, $y_j = \mu_j$) satisfies all clauses $C_1, C_2, \ldots, C_c$, which gives a contradiction. As clique Q also has exactly one vertex in each G_{C_k}, let u be that vertex and observe that y is ugly or Y-good in G_{C_k}. If u is the ugly vertex in G_{C_k}, it means that the bad vertices are not removed, and C_k

is satisfied by one of the variables $x_1, x_2, \ldots, x_a$. If u is a Y-good vertex, then u corresponds to an occurrence of some variable y_j. The construction of G guarantees that there is only one edge between u and a vertex in G_{y_j} that corresponds to a valuation of y_j that satisfies clause C_k. Thus, every clause is satisfied by some variable. A contradiction.

($\Rightarrow$) Let X be a subset of vertices of G witch $|X| \leqslant ac + 3c$ and $G \setminus X$ does not contain K_t as a subgraph. We construct a valuation $\nu_1, \nu_2, \ldots, \nu_a$ of variables $x_1, x_2, \ldots, x_a$ such that formula

$$\exists_{y_1, y_2, \ldots, y_b} : \quad C_1 \wedge C_2 \wedge \ldots \wedge C_c$$

is false. To remove all cliques of type A, X must include at least a positive or a negative vertices in each G_{x_i}. To remove all cliques of type B, X must include at least 3 good or 3 bad vertices in each G_{C_k}. As $|X| \leqslant ac + 3c$, we get that X includes exactly a positive or exactly a negative vertices in each G_{x_i}. We set ν_i to be true if and only if X includes negative vertices in G_{x_i}. Observe that if the constructed valuation of $x_1, x_2, \ldots, x_a$ satisfies clause C_k, then X has to include 3 good vertices in G_{C_k}, and the ugly vertex in G_{C_k} remains in $G \setminus X$. Otherwise, there would remain a clique of type C in $G \setminus X$.

Now, assuming to the contrary, let $\mu_1, \mu_2, \ldots, \mu_b$ be a valuation of variables $y_1, y_2, \ldots, y_b$ that satisfies all the clauses. We construct a clique Q in $G \setminus X$ of size $t = b + c$ the following way. For each $1 \leqslant j \leqslant b$, we select a positive (negative) vertex from G_{y_j} when μ_j is true (false). For each $1 \leqslant k \leqslant c$, we select the ugly vertex from G_{C_k} if it is not removed. If the ugly vertex is removed, then we know that C_k is satisfied by one of the variables $y_1, y_2, \ldots, y_b$, and we select the Y-good vertex in G_{C_k} that was assigned to the satisfied occurrence.

We claim that the resulting set of vertices induces a clique with $b + c$ vertices. Indeed, vertices from different subgraphs G_{y_j}, $G_{y_{j'}}$ are always connected by an edge (step 8). All ugly vertices are connected to each other (step 9) and both vertices in every subgraph G_{y_j} (step 7). Let u be a Y-good vertex selected from some G_{C_k} to Q. Vertex u corresponds to the occurrence of variable y_j in C_k. Vertex u is connected to ugly and Y-good vertices in other $G_{C_{k'}}$ for $k \neq k'$ (step 9) and both vertices in every subgraph $G_{y_{j'}}$ for $j \neq j'$ (step 6). As u corresponds to a satisfied occurrence of y_j in C_k, we get that u is also connected to the vertex selected from G_{y_j}. Thus, we have found a clique K_t in $G \setminus X$. A contradiction. ◀