The complexity of partially dominating graphs

David C. Kutner[1 [0000−0003−2979−4513]] and Iain A. Stewart[1 [0000−0002−0752−1971]]

Department of Computer Science, Durham University,
Upper Mountjoy Campus, Stockton Road, Durham DH1 3LE, UK
{david.c.kutner, i.a.stewart}@durham.ac.uk

Abstract. What is the complexity of dominating part of a graph? We formalize this question as Partial Domination: given a graph H and a subset T of its vertices, determine its domination number $\gamma(H[T])$. We study the complexity landscape of this problem when the "host graph" H is restricted to some graph class $\mathcal{G}$. Partial Domination restricted to $\mathcal{G}$ is at least as hard as Dominating Set is restricted the same, and at most as hard as Dominating Set is restricted to the class of induced subgraphs of members of $\mathcal{G}$ (its hereditary closure). We establish that this is the most that can be said for an arbitrary $\mathcal{G}$ by exhibiting graph classes with every computational hardness combination allowed by the previous sentence. For example, we construct a class $\mathcal{F}$ such that: Dominating Set is in P restricted to $\mathcal{F}$, Partial Domination is NP-intermediate restricted to $\mathcal{F}$, and Dominating Set is NP-complete restricted to the hereditary closure of $\mathcal{F}$.

Keywords: computational complexity · graph classes · hypercubes · graph domination.

This paper under preparation is made available for ease of access; it will likely be removed (and replaced with a redirect link) once a (more complete) preprint is on the arXiv.

1 Introduction

Our questions stem from an offshoot of a recently-submitted journal paper, an extended abstract for which was presented at ALGOWIN 2024 [2]. In that work, it was theoretically interesting and practically relevant to determine the computational complexity of Dominating Set restricted to induced subgraphs of hypercubes, but with the subtlety that we required the full hypercube (i.e. the induced supergraph) to also be in the input. It was possible, but nontrivial, to prove NP-completeness of this case. This raised the question: is there a class of graphs such that including the supergraph as a witness makes finding a dominating set strictly easier? (In the formalism introduced below: is there some $\mathcal{G}$ such that Partial Domination($\mathcal{G}$) is strictly easier than Dominating Set($\mathcal{G}_I$)?)

2 Prerequisites and definitions

Throughout this work, we assume P≠NP (or our questions become meaningless) and, more strongly, the Exponential Time Hypothesis (ETH) - though we expect this second requirement may be unnecessary to obtain our results.

We begin with some basic definitions from graph theory. Let $G = (V, E)$ be any simple undirected graph. We define the *open neighborhood* of a vertex v to be $N(v) := \{u : (u, v) \in E\}$, and its *closed neighborhood* $N[v] := N(v) \cup \{v\}$. Likewise, for any set of vertices S, we define $N[S] := \cup_{v \in S} N[v]$ and $N(S) := N[V] \setminus S$. A *dominating set* is a set of vertices $S \subseteq V$ such that $N[v] = V$. The *domination number* $\gamma(G)$ is the least number of vertices in any dominating set of G. The decision problem Dominating Set takes as input a graph G and integer k and asks whether $\gamma(G_{\leq} k)$. Where $S \subseteq V$ is a set of vertices in the graph, we denote $G[S]$ the subgraph of G induced by S. That is, $G[S]$ has S as its set of vertices and as edges exactly those edges of G with both endpoints incident to a vertex in S. We shall also make use of the following:

Definition 1 (Induced subgraph/supergraph, hereditary closure). *Where $T \subseteq V$ is a set of vertices in the graph, we denote $G[T]$ the subgraph of G induced by T. That is, $G[T]$ has T as its set of vertices and as edges exactly those edges of G with both endpoints incident to a vertex in T. We say G is an* induced subgraph *of H, and that H is an* induced supergraph *of G, if there exists some T such that $G = H[T]$.*

Let $\mathcal{G}$ be a class of graphs (the object $\mathcal{G}$ is formally an infinite set of graphs). Then we denote $\mathcal{G}_I$ the family $\{H : H$ is an induced subgraph of some graph $G \in \mathcal{G}\}$. We say $\mathcal{G}_I$ is the hereditary closure *of $\mathcal{G}$. If $\mathcal{G} = \mathcal{G}_I$, then we say the $\mathcal{G}$ is* hereditary.

Definition 2 ($\Pi(\mathcal{G})$). *Let Π be a graph problem with inputs $G, X_1, \ldots, X_\ell$, with G an undirected graph (possibly with $\ell = 0$, in which case Π takes just a graph as input). We denote $\Pi(\mathcal{G})$ the restriction of Π to the graph class $\mathcal{G}$, that is, $\Pi(\mathcal{G})$ has as instances exactly those instances of Π satisfying $G \in \mathcal{G}$.*

We introduce a new graph problem, which is subtly different from the restriction of Dominating Set, as we shall see.

> **Partial Domination**
> *Input:* Simple undirected graph $G = (V, E)$; set $T \subseteq V$; integer k.
> *Question:* Is $(G[T], k)$ a yes-instance of Dominating Set? Equivalently, is there some set $S \subseteq T$ with $|S| \leq k$ such that $T \subseteq N[S]$?

3 Known and immediate results

It is obvious from our definitions above that, for any class of graphs $\mathcal{G}$, if Dominating Set($\mathcal{G}$) is NP-complete, then Partial Domination($\mathcal{G}$) is NP-complete.

Further, it is natural and immediate that, if Partial Domination($\mathcal{G}$) is NP-complete, then Dominating Set($\mathcal{G}_I$) is NP-complete. Polynomial solvability results propagate in the opposite direction. As implications, we may write:

$$\begin{aligned}
&\forall \mathcal{G} : \textsc{Dominating Set}(\mathcal{G}) \text{ is NP-c} \implies \textsc{Partial Domination}(\mathcal{G}) \text{ is NP-c}\\
&\forall \mathcal{G} : \textsc{Partial Domination}(\mathcal{G}) \text{ is NP-c} \implies \textsc{Dominating Set}(\mathcal{G}_I) \text{ is NP-c}\\
&\forall \mathcal{G} : \textsc{Dominating Set}(\mathcal{G}) \text{ is in P} \impliedby \textsc{Partial Domination}(\mathcal{G}) \text{ is in P}\\
&\forall \mathcal{G} : \textsc{Partial Domination}(\mathcal{G}) \text{ is in P} \impliedby \textsc{Dominating Set}(\mathcal{G}_I) \text{ is in P}
\end{aligned}$$

On the other hand, it is not immediately clear whether the reverse implications hold. We shall exhibit classes of graphs which prove they do not.

4 Contribution

We define, in Table 1, *families of classes of graphs* $\mathfrak{A}, \ldots, \mathfrak{J}$. Every graph class $\mathcal{G}$ belongs to exactly one of these families of classes depending on the computational hardness (following directly from the implications above). For example, $\mathfrak{A}_\Pi$ includes planar graphs, $\mathfrak{E}_\Pi$ includes grids, and $\mathfrak{J}_\Pi$ includes trees. (The class NP-intermediate (NPI) contains those problems in NP which are neither in P nor NP-hard, unless P=NP.) Observe that any hereditary graph class necessarily belongs to one of $\mathfrak{A} \cup \mathfrak{D} \cup \mathfrak{J}$ (since then $\mathcal{G} = \mathcal{G}_I$). Our principal contributions are constructive proofs that each of thes families of classes is nonempty. The intention is that these preliminary findings may help to motivate further research (see Section 6).

$\mathcal{G}$ is in ...	DomSet($\mathcal{G}$) is ...	ParDom($\mathcal{G}$) is ...	DomSet($\mathcal{G}_I$) is ...	Example member
$\mathfrak{A}$	NPc	NPc	NPc	planar graphs
$\mathfrak{B}$	NPI	NPc	NPc	$\mathcal{B}$ (Sec. 5.3)
$\mathfrak{C}$	NPI	NPI	NPc	$\mathcal{C}$ (Sec. 5.3)
$\mathfrak{D}$	NPI	NPI	NPI	$\mathcal{D}$ (Sec. 5.2)
$\mathfrak{E}$	P	NPc	NPc	grids [1]
$\mathfrak{F}$	P	NPI	NPc	$\mathcal{F}$ (Sec. 5.3)
$\mathfrak{G}$	P	NPI	NPI	$\mathcal{G}$ (Sec. 5.3)
$\mathfrak{H}$	P	P	NPc	$\mathcal{H}$ (Sec. 5.2)
$\mathfrak{I}$	P	P	NPI	$\mathcal{I}$ (Sec. 5.2)
$\mathfrak{J}$	P	P	P	trees

Table 1. The definition of various families of graph classes. We prove that all families are nonempty by constructing a member class for each.

5 Results

5.1 Tools

For ease, we shall denote $\mathcal{A}$ the class of general graphs, $\mathcal{E}$ the class of grids, and $\mathcal{J}$ the class of trees (as these belong to $\mathfrak{A}, \mathfrak{E}, \mathfrak{J}$ respectively).

Definition 3 (Leafing of a graph, graph class). *Where G is a graph class, the* leafing *of G is the graph $L(G)$ obtained by creating, for each vertex v in G, a new vertex $l(v)$ adjacent only to v. (This graph may also be defined as the Corona product of G and K_1.) Where $\mathcal{G}$ is a graph class, the leafing of $\mathcal{G}$ is the class $L(\mathcal{G}) = \{L(G) : G \in \mathcal{G}\}$.*

Note that the set of neighbors of leaves in a leafed graph is a dominating set of minimum cardinality, yielding the following:

Lemma 1. *For each $\mathcal{G}$, Dominating Set($L(\mathcal{G})$) is in P. Also, for each $\mathcal{G}$, the complexity of Partial Domination($L(\mathcal{G})$) (resp. Dominating Set($L(\mathcal{G})_I$) is the same as the complexity of Partial Domination($\mathcal{G}$) (resp. Dominating Set($\mathcal{G}_I$)).*

Lemma 2. *There is no $2^{o(n)}$ algorithm for Dominating Set or for Vertex Cover unless ETH fails.*

Lemma 3. *If G is a graph on n vertices and m edges, and H is the graph obtained by subdividing each edge of G ℓ times.*

Then H has a vertex cover of cardinality $m \cdot \frac{l}{3} + k$ if and only if G has a vertex cover of size k.

5.2 $\mathfrak{D}, \mathfrak{H}, \mathfrak{I}$ are nonempty.

Let $\mathcal{D} := \left\{ G \text{ subdivided } 2^{3\lceil\sqrt{|V(G)|}\rceil} \text{ times } : G \in \mathcal{A} \right\}_I$. (Note the subscript I, meaning we take the hereditary closure of the set of graphs obtained by subdividing n-vertex graphs $3\lceil\sqrt{n}\rceil$ times.)

Theorem 1. $\mathcal{D} \in \mathfrak{D}$.

Proof (Sketch). First observe that $\mathcal{D}$ is hereditary. Consequently, to show that Dominating Set($\mathcal{D}$) is NP-intermediate to show $\mathcal{D} \in \mathfrak{D}$.

First, we show that a polynomial-time algorithm for Dominating Set($\mathcal{D}$) would entail a subexponential time algorithm for Vertex Cover - impossible unless ETH fails (applying Lemma 2).

Next, we show that Dominating Set($\mathcal{D}$) is not NP-hard (unless ETH fails). Suppose Dominating Set($\mathcal{D}$) is NP-hard. This is done by showing that instances of Dominating Set($\mathcal{D}$) can be reduced in polynomial time to a kernel instance of the same of size polylogarithmic in the size of the original, which may then be solved by brute force in time $2^{\mathrm{polylog}(n)}$; applying Lemma 2 again, we obtain the desired contradiction.

An important aspect of this second part of the proof is the fact that, for any instance (G, k) of Dominating Set($\mathcal{D}$), G either contains an induced path of length $\Omega(2^{\sqrt{k}})$, or G consists entirely of disjoint subdivided stars and paths.

Let $\mathcal{H} := \left\{G \oplus \overline{K_{2^{|V(G)|}}} : G \in \mathcal{A}\right\}$. That is, $\mathcal{H}$ is the family of graphs containing, for each graph G on n vertices (recall $\mathcal{A}$ is the class of general graphs), the disjoint union of G and 2^n isolated vertices.

Theorem 2. $\mathcal{H} \in \mathfrak{H}$.

Proof (Sketch). First observe that $\mathcal{H}_I = \mathcal{A}$, immediately yielding that Dominating Set($\mathcal{H}_I$) is NP-complete. We show that any instance of Partial Domination($\mathcal{H}$) may be reduced to a "kernel" instance of size logarithmic in that of the original, and subsequently may apply a brute-force algorithm to solving the kernel instance in time polynomial in the size of the original input. Applying our implications from Section 3 the result also holds for Dominating Set($\mathcal{H}$).

We now combine both ideas above: let $\mathcal{I} := \left\{D \oplus \overline{K_{2^{|V(G)|}}} : D \in \mathcal{D}\right\}$. That is, $\mathcal{I}$ is the family of graphs containing, for each graph G on n vertices from $\mathcal{D}$, the disjoint union of G and 2^n isolated vertices.

Theorem 3. $\mathcal{I} \in \mathfrak{I}$.

Proof (Sketch). First observe that $\mathcal{H}_I = \mathcal{D}$, so Dominating Set($\mathcal{H}_I$) is NP-intermediate (applying Theorem 1). Applying the same logic as in our proof of Theorem 2, any instance of Partial Domination($\mathcal{H}$) may be reduced to a "kernel" instance of size logarithmic in that of the original, and subsequently we may apply a brute-force algorithm to solving the kernel instance in time polynomial in the size of the original input. Applying our implications from Section 3 the result also holds for Dominating Set($\mathcal{H}$).

5.3 $\mathfrak{B}, \mathfrak{C}, \mathfrak{F}, \mathfrak{G}$ are each nonempty.

Using the results above, it becomes relatively straightforward to prove the following:

- $\mathfrak{B} \ni \mathcal{B} := \mathcal{E} \cup \mathcal{D}$,
- $\mathfrak{C} \ni \mathcal{D} \cup \mathcal{H}$,
- $\mathfrak{F} \ni \mathcal{F} := L(\mathcal{C})$ (applying Lemma 1), and
- $\mathfrak{G} \ni \mathcal{G} := L(\mathcal{D})$ (again applying Lemma 1)

6 Further questions

Without assuming P≠NP our problems become vacuous; nonetheless, it seems likely that many of our results could be proven without assuming ETH.

Question 1. Can we remove our reliance on ETH?

Also, we note that, to our knowledge, none of the classes we construct were previously known.

Question 2. Which (if any) *natural* classes belong to $\mathfrak{B}, \mathfrak{C}, \mathfrak{D}, \mathfrak{F}, \mathfrak{G}, \mathfrak{H}, \mathfrak{I}$?

One particularly interesting candidate is the family of hypercubes $\mathcal{Q}$.

Question 3. Is there a polynomial-time algorithm for Dominating Set($\mathcal{Q}$)?

It is unlikely there is a practical polynomial-time algorithm for Dominating Set($\mathcal{H}$); even $\gamma(Q_{10})$ is unkown. See `https://oeis.org/A000983` and the discussion therein. On the other hand, it is provable (by leveraging Mahaney's Theorem [3]) that Dominating Set($\mathcal{H}$) is not NP-c unless P=NP. If the answer to Question 3 is negative, this would provide a first natural class belonging to $\mathfrak{B}$ (if the answer is positive, then $\mathcal{Q} \in \mathfrak{E}$, the same as grids).

References

1. David C. Kutner and Iain A. Stewart. Reconfigurable routing in data center networks, 2024.
2. David C. Kutner and Iain A. Stewart. Reconfigurable routing in data center networks. In Quentin Bramas, Arnaud Casteigts, and Kitty Meeks, editors, *Algorithmics of Wireless Networks*, pages 117–130, Cham, 2025. Springer Nature Switzerland.
3. Stephen R. Mahaney. Sparse complete sets for np: Solution of a conjecture of berman and hartmanis. *Journal of Computer and System Sciences*, 25(2):130–143, 1982.