New Width Parameters for Independent Set: One-Sided-Mim-Width and Neighbor-Depth

Benjamin Bergougnoux[1], Tuukka Korhonen[2](✉), and Igor Razgon[3]

[1] University of Warsaw, Warsaw, Poland
benjamin.bergougnoux@mimuw.edu.pl
[2] University of Bergen, Bergen, Norway
tuukka.korhonen@uib.no
[3] Birkbeck University of London, London, UK
i.razgon@bbk.ac.uk

Abstract. We study the tractability of the maximum independent set problem from the viewpoint of graph width parameters, with the goal of defining a width parameter that is as general as possible and allows to solve independent set in polynomial-time on graphs where the parameter is bounded. We introduce two new graph width parameters: one-sided maximum induced matching-width (o-mim-width) and neighbor-depth. O-mim-width is a graph parameter that is more general than the known parameters mim-width and tree-independence number, and we show that independent set and feedback vertex set can be solved in polynomial-time given a decomposition with bounded o-mim-width. O-mim-width is the first width parameter that gives a common generalization of chordal graphs and graphs of bounded clique-width in terms of tractability of these problems.

The parameter o-mim-width, as well as the related parameters mim-width and sim-width, have the limitation that no algorithms are known to compute bounded-width decompositions in polynomial-time. To partially resolve this limitation, we introduce the parameter neighbor-depth. We show that given a graph of neighbor-depth k, independent set can be solved in time $n^{O(k)}$ even without knowing a corresponding decomposition. We also show that neighbor-depth is bounded by a polylogarithmic function on the number of vertices on large classes of graphs, including graphs of bounded o-mim-width, and more generally graphs of bounded sim-width, giving a quasipolynomial-time algorithm for independent set on these graph classes. This resolves an open problem asked by Kang, Kwon, Strømme, and Telle [TCS 2017].

Keywords: Graph width parameters · Mim-width · Sim-width · Independent set

---

Due to space limits, most of technicals details are omitted or just sketched. The full version of the paper is available on arXiv [2]. Tuukka Korhonen was supported by the Research Council of Norway via the project BWCA (grant no. 314528).

D. Paulusma and B. Ries (Eds.): WG 2023, LNCS 14093, pp. 72–85, 2023.
https://doi.org/10.1007/978-3-031-43380-1_6

1 Introduction

Graph width parameters have been successful tools for dealing with the intractability of NP-hard problems over the last decades. While tree-width [25] is the most prominent width parameter due to its numerous algorithmic and structural properties, only sparse graphs can have bounded tree-width. To capture the tractability of many NP-hard problems on well-structured dense graphs, several graph width parameters, including clique-width [7], mim-width [26], Boolean-width [6], tree-independence number [9,27], minor-matching hypertree width [27], and sim-width [20] have been defined. A graph parameter can be considered to be more general than another parameter if it is bounded whenever the other parameter is bounded. For a particular graph problem, it is natural to look for the most general width parameter so that the problem is tractable on graphs where this parameter is bounded. In this paper, we focus on the maximum independent set problem (Independent Set).

Let us recall the standard definitions on branch decompositions. Let V be a finite set and $\mathtt{f} : 2^V \to \mathbb{Z}_{\geq 0}$ a symmetric set function, i.e., for all $X \subseteq V$ it holds that $\mathtt{f}(X) = \mathtt{f}(V \setminus X)$. A branch decomposition of $\mathtt{f}$ is a pair (T, δ), where T is a cubic tree and δ is a bijection mapping the elements of V to the leaves of T. Each edge e of T naturally induces a partition (X_e, Y_e) of the leaves of T into two non-empty sets, which gives a partition $(\delta^{-1}(X_e), \delta^{-1}(Y_e))$ of V. We say that the width of the edge e is $\mathtt{f}(e) = \mathtt{f}(\delta^{-1}(X_e)) = \mathtt{f}(\delta^{-1}(Y_e))$, the width of the branch decomposition (T, δ) is the maximum width of its edges, and the branchwidth of the function $\mathtt{f}$ is the minimum width of a branch decomposition of $\mathtt{f}$. When G is a graph and $\mathtt{f} : 2^{V(G)} \to \mathbb{Z}_{\geq 0}$ is a symmetric set function on $V(G)$, we say that the $\mathtt{f}$-*width* of G is the branchwidth of $\mathtt{f}$.

Vatshelle [26] defined the maximum induced matching-width (mim-width) of a graph to be the $\mathtt{mim}$-width where $\mathtt{mim}(A)$ for a set of vertices A is defined to be the size of a maximum induced matching in the bipartite graph $G[A, \overline{A}]$ given by edges between A and $\overline{A}$, where $\overline{A} = V(G) \setminus A$. He showed that given a graph together with a branch decomposition of mim-width k, any locally checkable vertex subset and vertex partitioning problem (LC-VSVP), including Independent Set, Dominating Set, and Graph Coloring with a constant number of colors, can be solved in time $n^{\mathcal{O}(k)}$. Mim-width has gained a lot of attention recently [1,4,5,17–19,22]. While mim-width is more general than clique-width and bounded mim-width captures many graph classes with unbounded clique-width (e.g. interval graphs), there are many interesting graph classes with unbounded mim-width where Independent Set is known to be solvable in polynomial-time. Most notably, chordal graphs, and even their subclass split graphs, have unbounded mim-width, but it is a classical result of Gavril [15] that Independent Set can be solved in polynomial-time on them. More generally, all width parameters in a general class of parameters that contains mim-width and was studied by Eiben, Ganian, Hamm, Jaffke, and Kwon [11] are unbounded on split graphs.

With the goal of providing a generalization of mim-width that is bounded on chordal graphs, Kang, Kwon, Strømme, and Telle [20] defined the parameter *special induced matching-width* (sim-width). Sim-width of a graph G is the $\mathtt{sim}$-width where $\mathtt{sim}(A)$ for a set of vertices A is defined to be the maximum size of an induced matching in G whose every edge has one endpoint in A and another in $\overline{A}$. The key difference of $\mathtt{mim}$ and $\mathtt{sim}$ is that $\mathtt{mim}$ ignores the edges in $G[A]$ and $G[\overline{A}]$ when determining if the matching is induced, while $\mathtt{sim}$ takes them into account, and therefore the sim-width of a graph is always at most its mim-width. Chordal graphs have sim-width at most one [20]. However, it is not known if Independent Set can be solved in polynomial-time on graphs of bounded sim-width, and indeed Kang, Kwon, Strømme, and Telle asked as an open question if Independent Set is NP-complete on graphs of bounded sim-width [20].

In this paper, we introduce a width parameter that for the Independent Set problem, captures the best of both worlds of mim-width and sim-width. Our parameter is inspired by a parameter introduced by Razgon [24] for classifying the OBDD size of monotone 2-CNFs. For a set of vertices A, let $E(A)$ denote the edges of the induced subgraph $G[A]$. For a set $A \subseteq V(G)$, we define the upper-induced matching number $\mathtt{umim}(A)$ of A to be the maximum size of an induced matching in $G - E(\overline{A})$ whose every edge has one endpoint in A and another in $\overline{A}$. Then, we define the *one-sided maximum induced matching-width* (o-mim-width) of a graph to be the $\mathtt{omim}$-width where $\mathtt{omim}(A) = \min(\mathtt{umim}(A), \mathtt{umim}(\overline{A}))$. In particular, o-mim-width is like sim-width, but we ignore the edges on one side of the cut when determining if a matching is induced. Clearly, the o-mim-width of a graph is between its mim-width and sim-width. Our first result is that the polynomial-time solvability of Independent Set on graphs of bounded mim-width generalizes to bounded o-mim-width. Moreover, we show that the interest of o-mim-width is not limited to Independent Set by proving that the Feedback Vertex Set problem is also solvable in polynomial time on graphs of bounded o-mim-width.

Theorem 1. *Given an n-vertex graph together with a branch decomposition of o-mim-width k,* Independent Set *and* Feedback Vertex Set *can be solved in time $n^{\mathcal{O}(k)}$.*

We also show that o-mim-width is bounded on chordal graphs. In fact, we show a stronger result that o-mim-width of any graph is at most its *tree-independence number* ($\mathsf{tree}\text{-}\alpha$), which is a graph width parameter defined by Dallard, Milanič, and Štorgel [9] and independently by Yolov [27], and is known to be at most one on chordal graphs.

Theorem 2. *Any graph with tree-independence number k has o-mim-width at most k.*

We do not know if there is a polynomial-time algorithm to compute a branch decomposition of bounded o-mim-width if one exists, and the corresponding question is notoriously open also for both mim-width and sim-width. Because of this, it is also open whether Independent Set can be solved in polynomial-time on graphs of bounded mim-width, and more generally on graphs of bounded o-mim-width.

In our second contribution we partially resolve the issue of not having algorithms for computing branch decompositions with bounded mim-width, o-mim-width, or sim-width. We introduce a graph parameter *neighbor-depth*.

Definition 3. *The neighbor-depth (*$\mathtt{nd}$*) of a graph* G *is defined recursively as follows:*

1. $\mathtt{nd}(G) = 0$ *if and only if* $V(G) = \emptyset$*,*
2. *if* G *is not connected, then* $\mathtt{nd}(G)$ *is the maximum value of* $\mathtt{nd}(G[C])$ *where* $C \subseteq V(G)$ *is a connected component of* G*,*
3. *if* $V(G)$ *is non-empty and* G *is connected, then* $\mathtt{nd}(G) \leq k$ *if and only if there exists a vertex* $v \in V(G)$ *such that* $\mathtt{nd}(G \setminus N[v]) \leq k - 1$ *and* $\mathtt{nd}(G \setminus \{v\}) \leq k$*.*

In the case (3) of Definition 3, we call the vertex v *the* pivot-vertex *witnessing* $\mathtt{nd}(G) \leq k$*.*

By induction, the neighbor-depth of all graphs is well-defined. We show that neighbor-depth can be computed in $n^{\mathcal{O}(k)}$ time and also Independent Set can be solved in time $n^{\mathcal{O}(k)}$ on graphs of neighbor-depth k.

Theorem 4. *There is an algorithm that given a graph* G *of neighbor-depth* k*, determines its neighbor-depth and solves* Independent Set *in time* $n^{\mathcal{O}(k)}$*.*

We show that graphs of bounded sim-width have neighbor-depth bounded by a polylogarithmic function on the number of vertices.

Theorem 5. *Any* n*-vertex graph of sim-width* k *has neighbor-depth* $\mathcal{O}(k \log^2 n)$*.*

Theorems 4 and 5 combined show that Independent Set can be solved in time $n^{\mathcal{O}(k \log^2 n)}$ on graphs of sim-width k, which in particular is quasipolynomial time for fixed k. This resolves, under the mild assumption that $\mathsf{NP} \not\subseteq \mathsf{QP}$, the question of Kang, Kwon, Strømme, and Telle, who asked if Independent Set is NP-complete on graphs of bounded sim-width [20, Question 2].

Neighbor-depth characterizes branching algorithms for Independent Set in the following sense. We say that an independent set branching tree of a graph G is a binary tree whose every node is labeled with an induced subgraph of G, so that (1) the root is labeled with G, (2) every leaf is labeled with the empty graph, and (3) if a non-leaf node is labeled with a graph $G[X]$, then either (a) its children are labeled with the graphs $G[L]$ and $G[R]$ where (L, R) is a partition of X with no edges between L and R, or (b) its children are labeled with the graphs $G[X \setminus N[v]]$ and $G[X \setminus \{v\}]$ for some vertex $v \in X$. Note that such a tree corresponds naturally to a branching approach for Independent Set, where we branch on a single vertex and solve connected components independently of each other. Let $\beta(G)$ denote the smallest number of nodes in an independent set branching tree of a graph G. Neighbor-depth gives both lower- and upper-bounds for $\beta(G)$.

Theorem 6. *For all graphs* G*, it holds that* $2^{\mathtt{nd}(G)} \leq \beta(G) \leq n^{\mathcal{O}(\mathtt{nd}(G))}$*.*

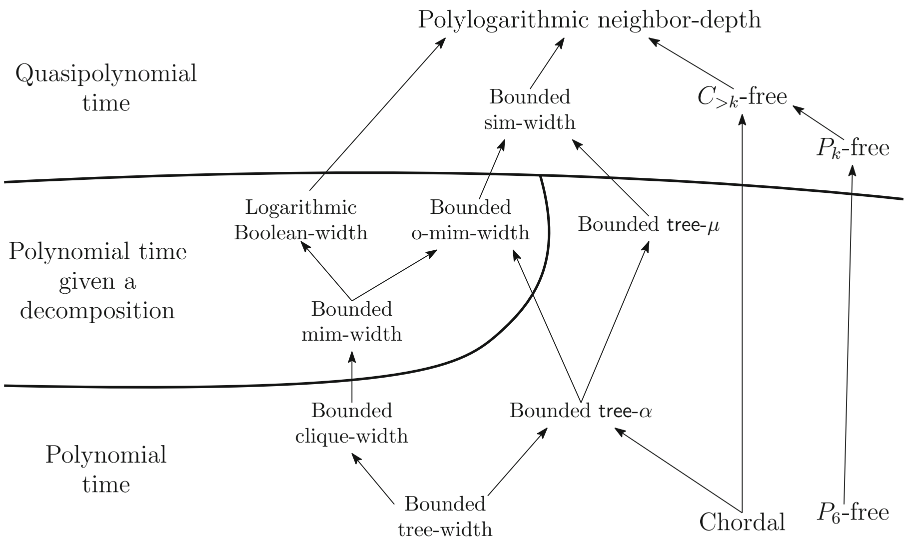

Fig. 1. Hierarchy of some graph classes with polylogarithmically bounded neighbor-depth, divided vertically on whether the best known algorithm for Independent Set on the class is polynomial time, polynomial time given a decomposition (and quasipolynomial without a decomposition), or quasipolynomial time.

By observing that some known algorithms for Independent Set in fact construct independent set branching trees implicitly, we obtain upper bounds for neighbor-depth on some graph classes purely by combining the running times of such algorithms with Theorem 6. In particular, for an integer k, we say that a graph is $C_{>k}$-free if it does not contain induced cycles of length more than k. Gartland, Lokshtanov, Pilipczuk, Pilipczuk and Rzazewski [14] showed that Independent Set can be solved in time $n^{\mathcal{O}(\log^3 n)}$ on $C_{>k}$-free graphs for any fixed k, generalizing a result of Gartland and Lokshtanov on P_k-free graphs [13]. By observing that their algorithm is a branching algorithm that (implicitly) constructs an independent set branching tree, it follows from Theorem 6 that the neighbor-depth of $C_{>k}$-free graphs is bounded by a polylogarithmic function on the number of vertices.

Proposition 7. *For every fixed integer k, $C_{>k}$-free graphs with n vertices have neighbor-depth at most $\mathcal{O}(\log^4 n)$.*

Along the same lines as Proposition 7, a polylogarithmic upper bound for neighbor-depth can be also given for graphs with bounded induced cycle packing number, using the quasipolynomial algorithm of Bonamy, Bonnet, Déprés, Esperet, Geniet, Hilaire, Thomassé, and Wesolek [3].

In Fig. 1 we show the hierarchy of inclusions between some of the graph classes discussed in this paper, and the known algorithmic results for Independent Set on those classes. All the inclusions shown are proper, and all the inclusions between these classes appear in the figure. Some of the inclusions are proven in Sects. 3 and 4, and some of the non-inclusions in the full version of the paper [2]. Note that bounded Boolean-width is equivalent to bounded clique-width [26]. The polynomial-time algorithm for Independent Set on P_6-free graphs is from [16], the definition of tree-μ and polynomial-time algorithm for Independent Set on graphs of bounded tree-μ is from [27], and the definition of Boolean-width and a polynomial-time algorithm for Independent Set on graphs of logarithmic Boolean-width is from [6]. The inclusion of logarithmic Boolean-width in polylogarithmic neighbor-depth follows from Theorem 5 and the fact the sim-width of a graph is at most its Boolean-width. Polynomial-time algorithm for Independent Set on graphs of bounded clique-width follows from [8,23].

Organization of this Paper. We prove the part of Theorem 1 on Independent Set and Theorem 2 in Sect. 3. Theorem 5 is proved in Sect. 4. Proofs omitted in this version of the paper due to space constraints are provided in the full version in [2].

2 Preliminaries

The size of a set V is denoted by $|V|$ and its power set is denoted by 2^V. We let $\max(\emptyset) := -\infty$. Our graph terminology is standard and we refer to [10].

The subgraph of G induced by a subset X of its vertex set is denoted by $G[X]$. We also use the notation $G \setminus X = G[V(G) \setminus X]$. For two disjoint subsets of vertices X and Y of $V(G)$, we denote by $G[X,Y]$ the bipartite graph with vertex set $X \cup Y$ and edge set $\{xy \in E(G) : x \in X \text{ and } y \in Y\}$. Given two disjoint set of vertices X, Y, we denote by $E(X)$ the set of edges of $G[X]$ and by $E(X,Y)$ the set of edges of $G[X,Y]$. For a set of edges E' of G, we denote by $G - E'$ the graph with vertex set $V(G)$ and edge set $E(G) \setminus E'$.

An *independent set* is a set of vertices that induces an edgeless graph. Given a graph G with a weight function $w : V(G) \to \mathbb{Z}_{\geq 0}$, the problem Independent Set asks for an independent set of maximum weight, where the weight of a set $X \subseteq V(G)$ is $\sum_{x \in X} w(x)$. A *feedback vertex set* is the complement of a set of vertices inducing a forest (i.e. acyclic graph). The problem Feedback Vertex Set asks for a feedback vertex set of minimum weight.

A *matching* in a graph G is a set $M \subseteq E(G)$ of edges having no common endpoint. We denote by $V(M)$ the set of vertices incident to M. An *induced matching* is a matching M such that $G[V(M)]$ does not contain any other edges than M. Given two disjoint subsets A, B of $V(G)$, we say that a matching M is a (A,B)-matching if every edge of M has one endpoint in A and the other in B.

Width Parameters. We refer to the introduction for the definitions of branch-decomposition and f-width, we recall below the definitions of mim-width, sim-width and o-mim-width.

– The maximum induced matching-width (mim-width) [26] of a graph G is the $\texttt{mim}$-width where $\texttt{mim}(A)$ is the size of a maximum induced matching of the graph $G[A, \overline{A}]$.
– The special induced matching-width (sim-width) [20] of a graph G is the $\texttt{sim}$-width where $\texttt{sim}(A)$ is the size of maximum induced $(A, \overline{A})$-matching in the graph G.
– Given a graph G and $A \subseteq V(G)$, the *upper-mim-width* $\texttt{umim}(A)$ of A is the size of maximum induced $(A, \overline{A})$-matching in the graph $G - E(\overline{A})$. The one-sided maximum induced matching-width (o-mim-width) of G is the $\texttt{omim}$-width where $\texttt{omim}(A) := \min(\texttt{umim}(A), \texttt{umim}(\overline{A}))$.

The following is a standard lemma that $\texttt{f}$-width at most k implies balanced cuts with $\texttt{f}$-width at most k.

Lemma 8. *Let G be a graph, $X \subseteq V(G)$ a set of vertices with $|X| \geq 2$, and $\texttt{f} : 2^{V(G)} \to \mathbb{Z}_{\geq 0}$ a symmetric set function. If the $\texttt{f}$-width of G is at most k, then there exists a bipartition $(A, \overline{A})$ of $V(G)$ with $\texttt{f}(A) \leq k$, $|X \cap A| \leq \frac{2}{3}|X|$, and $|X \cap \overline{A}| \leq \frac{2}{3}|X|$.*

A tree decomposition of a graph G is a pair $(T, \texttt{bag})$, where T is a tree and $\texttt{bag} : V(T) \to 2^{V(G)}$ is a function from the nodes of T to subsets of vertices of G called *bags*, satisfying that (1) for every edge $uv \in E(G)$ there exists a node $t \in V(T)$ so that $\{u, v\} \subseteq \texttt{bag}(t)$, and (2) for every vertex $v \in V(G)$, the set of nodes $\{t \in V(T) \colon v \in \texttt{bag}(t)\}$ induces a non-empty and connected subtree of T. The width of a tree decomposition is the maximum size of $\texttt{bag}(t)$ minus one, and the treewidth of a graph is the minimum width of a tree decomposition of the graph.

For a set of vertices $X \subseteq V(G)$, we denote by $\alpha(X)$ the maximum size of an independent set in $G[X]$. The independence number of a tree decomposition $(T, \texttt{bag})$ is the maximum of $\alpha(\texttt{bag}(t))$ over $t \in V(T)$ and it is denoted by $\alpha(T, \texttt{bag})$. The tree-independence number of a graph ($\texttt{tree-}\alpha$) is the minimum independence number of a tree decomposition of the graph [9,27].

For a set of vertices $X \subseteq V(G)$, we denote by $\mu(X)$ the maximum size of an induced matching in G so that for each edge of the matching, at least one of the endpoints of the edge is in X. For a tree decomposition $(T, \texttt{bag})$, we denote by $\mu(T, \texttt{bag})$ the maximum of $\mu(\texttt{bag}(t))$ over $t \in V(T)$. Yolov [27] defined the minor-matching hypertree width ($\texttt{tree-}\mu$) of a graph to be the minimum $\mu(T, \texttt{bag})$ of a tree decomposition $(T, \texttt{bag})$ of G.

3 O-Mim-Width

In this section, we prove the part of Theorem 1 on Independent Set and Theorem 2. We start with some intermediary results. The following reveals an important property of cuts of bounded upper-mim-width. Razgon proved a similar statement in [24]. To simplify the statements of this section, we fix an n-vertex graph G with a weight function $w : V(G) \to \mathbb{Z}_{\geq 0}$.

Lemma 9. *Let $A \subseteq V(G)$. For every $X \subseteq A$ that is the union of t independent sets, there exists $X' \subseteq X$ of size at most $t \cdot \mathtt{umim}(A)$ such that $N(X) \setminus A = N(X') \setminus A$. In particular, we have $|\{N(X) \setminus A : X \in \mathsf{IS}(A)\}| \leq n^{\mathtt{umim}(A)}$ where $\mathsf{IS}(A)$ is the set of independent sets of $G[A]$.*

Proof. It is sufficient to prove the lemma for $t = 1$, since if X is the union of t independent sets $X_1, \ldots, X_t$, then the case $t = 1$ implies that, for each $i \in [1, t]$, there exits $X_i' \subseteq X_i$ such that $N(X_i) \setminus A = N(X_i') \setminus A$ and $|X_i'| \leq \mathtt{umim}(A)$. It follows that $X' = X_1' \cup \cdots \cup X_t' \subseteq X$, $N(X) \setminus A = N(X') \setminus A$ and $|X'| \leq t \cdot \mathtt{umim}(A)$.

Let X be an independent set of $G[A]$. If for every vertex $x \in X$, there exists a vertex $y_x \in \overline{A}$ such that $N(y_x) \cap X = \{x\}$, then $\{xy_x : x \in X\}$ is an induced $(A, \overline{A})$-matching in $G - E(\overline{A})$. We deduce that either $|X| \leq \mathtt{umim}(A)$ or there exists a vertex $x \in X$ such that $N(X) \setminus A = N(X \setminus \{x\}) \setminus A$. Thus, we can recursively remove vertices from X to find a set $X' \subseteq X$ of size at most $\mathtt{umim}(A)$ and such that $N(X) \setminus A = N(X') \setminus A$. In particular, the latter implies that $\{N(X) \setminus A : X \in \mathsf{IS}(A)\} = \{N(X) \setminus A : X \in \mathsf{IS}(A) \wedge |X| \leq \mathtt{umim}(A)\}$. We conclude that $|\{N(X) \setminus A : X \in \mathsf{IS}(A)\}| \leq n^{\mathtt{umim}(A)}$. □

To solve Independent Set and Feedback Vertex Set, we use the general toolkit developed in [1] with a simplified notation adapted to our two problems. This general toolkit is based on the following notion of representativity between sets of partial solutions. In the following, the collection $\mathcal{S}$ represents the set of solutions, in our setting $\mathcal{S}$ consists of either all the independent sets or all the set of vertices inducing a forest.

Definition 10. *Given $\mathcal{S} \subseteq 2^{V(G)}$, for every $\mathcal{A} \subseteq 2^{V(G)}$ and $Y \subseteq V(G)$, we define* $\mathrm{best}_{\mathcal{S}}(\mathcal{A}, Y) := \max\{w(X) : X \in \mathcal{A} \wedge X \cup Y \in \mathcal{S}\}$. *Given $A \subseteq V(G)$ and $\mathcal{A}, \mathcal{B} \subseteq 2^A$, we say that $\mathcal{B}$ $(\mathcal{S}, A)$-represents $\mathcal{A}$ if for every $Y \subseteq \overline{A}$, we have* $\mathrm{best}_{\mathcal{S}}(\mathcal{A}, Y) = \mathrm{best}_{\mathcal{S}}(\mathcal{B}, Y)$.

Observe that if there is no $X \in \mathcal{B}$ such that $X \cup Y \in \mathcal{S}$, then $\mathrm{best}_{\mathcal{S}}(\mathcal{B}, Y) = \max(\emptyset) = -\infty$. It is easy to see that the relation "$(\mathcal{S}, A)$-represents" is an equivalence relation.

The following is an application of Theorem 4.1 from [1]. It proves that a routine for computing small representative sets can be used to design a dynamic programming algorithm.

Theorem 11 **([1]).** *Let $\mathcal{S} \subseteq 2^{V(G)}$. Assume that there exists a constant c and an algorithm that, given $A \subseteq V(G)$ and $\mathcal{A} \subseteq 2^A$, computes in time $|\mathcal{A}| n^{\mathcal{O}(\mathtt{omim}(A))}$ a subset $\mathcal{B}$ of $\mathcal{A}$ such that $|\mathcal{B}| \leq n^{c \cdot \mathtt{omim}(A)}$ and $\mathcal{B}$ $(\mathcal{S}, A)$-represents $\mathcal{A}$. Then, there exists an algorithm, that given a branch decomposition $\mathcal{L}$ of G, computes in time $n^{\mathcal{O}(\mathtt{omim}(\mathcal{L}))}$ a set of size at most $n^{c \cdot \mathtt{omim}(A)}$ that contains an element in $\mathcal{S}$ of maximum weight.*

The following lemma provides a routine to compute small representative sets for Independent Set. We denote by $\mathcal{I}$ the set of all independent sets of G.

Lemma 12. *Let $k = \mathtt{omim}(A)$. Given a collection $\mathcal{A} \subseteq 2^A$, we can compute in time $|\mathcal{A}| n^{\mathcal{O}(k)}$ a subset $\mathcal{B}$ of $\mathcal{A}$ such that $\mathcal{B}$ $(\mathcal{I}, A)$-represents $\mathcal{A}$ and $|\mathcal{B}| \leq n^k$.*

Proof. Let $\mathcal{A} \subseteq 2^A$. We compute $\mathcal{B}$ from the empty set as follows:

– If $\mathtt{umim}(A) = k$, then, for every $Y \in \{N(X) \setminus A : X$ is an independent in $\mathcal{A}\}$, we add to $\mathcal{B}$ an independent set $X \in \mathcal{A}$ of maximum weight such that $Y = N(X) \setminus A$.
– If $\mathtt{umim}(A) > k$, then, for each subset $Y \subseteq \overline{A}$ with $|Y| \leq k$, we add to $\mathcal{B}$ a set $X \in \mathcal{A}$ of maximum weight such that $X \cup Y$ is an independent set (if such X exists).

It remains to prove the runtime. First, we prove that $|\mathcal{B}| \leq n^k$. This is straightforward when $\mathtt{umim}(A) > k$. When $\mathtt{umim}(A) = k$, Lemma 9 implies that $|\{N(X) \setminus A : X$ is an independent in $\mathcal{A}\}| \leq n^k$ and thus, we have $|\mathcal{B}| \leq n^k$.

Next, we prove that $\mathcal{B}$ $(\mathcal{I}, A)$-represents $\mathcal{A}$, i.e. for every $Y \subseteq \overline{A}$, we have that $\mathrm{best}_{\mathcal{I}}(\mathcal{A}, Y) = \mathrm{best}_{\mathcal{I}}(\mathcal{B}, Y)$. Let $Y \subseteq \overline{A}$. As $\mathcal{B}$ is subset of $\mathcal{A}$, we have $\mathrm{best}_{\mathcal{I}}(\mathcal{B}, Y) \leq \mathrm{best}_{\mathcal{I}}(\mathcal{A}, Y)$. In particular, if there is no $X \in \mathcal{A}$ such that $X \cup Y$ is an independent set, then we have $\mathrm{best}_{\mathcal{I}}(\mathcal{A}, Y) = \mathrm{best}_{\mathcal{I}}(\mathcal{B}, Y) = -\infty$.

Suppose from now that $\mathrm{best}_{\mathcal{I}}(\mathcal{A}, Y) \neq -\infty$ and let $X \in \mathcal{A}$ such that $X \cup Y$ is an independent set and $w(X) = \mathrm{best}_{\mathcal{I}}(\mathcal{A}, Y)$. We distinguish the following cases:

– If $\mathtt{umim}(A) = k$, then, by construction, there exists an independent set $W \in \mathcal{B}$ such that $N(X) \setminus A = N(W) \setminus A$ and $w(X) \leq w(W)$. As $X \cup Y$ is an independent set, we deduce that $N(X) \cap Y = N(W) \cap Y = \emptyset$ and thus $W \cup Y$ is an independent set.
– If $\mathtt{umim}(A) > k$, then $\mathtt{umim}(\overline{A}) = k$ as $\mathtt{omim}(A) = \min(\mathtt{umim}(A), \mathtt{umim}(\overline{A})) = k$. By Lemma 9, there exists an independent set $Y' \subseteq Y$ of size at most k such that $N(Y) \setminus \overline{A} = N(Y') \setminus \overline{A}$. As $Y' \subseteq Y$, we know that $X \cup Y'$ is an independent set. Thus, by construction there exists a set $W \in \mathcal{B}$ such that $W \cup Y'$ is an independent set and $w(X) \leq w(W)$. Since $N(Y) \setminus A = N(Y') \setminus A$, we deduce that $W \cup Y$ is an independent set.

In both cases, there exists $W \in \mathcal{B}$ such that $W \cup Y$ is an independent set and $w(X) \leq w(W) \leq \mathrm{best}_{\mathcal{I}}(\mathcal{B}, Y)$. Since $\mathrm{best}_{\mathcal{I}}(\mathcal{B}, Y) \leq \mathrm{best}_{\mathcal{I}}(\mathcal{A}, Y) = w(X)$, it follows that $w(X) = \mathrm{best}_{\mathcal{I}}(\mathcal{A}, Y) = \mathrm{best}_{\mathcal{I}}(\mathcal{B}, Y)$. As this holds for every $Y \subseteq \overline{A}$, we conclude that $\mathcal{B}$ $(\mathcal{I}, A)$-represents $\mathcal{A}$.

It remains to prove the running time. Computing $\mathtt{omim}(A) = k$ and checking whether $\mathtt{umim}(A) = k$ can be done by looking at every set of $k+1$ edges and check whether one of these sets is an induced $(A, \overline{A})$-matching in $G - E(\overline{A})$ and in $G - E(A)$. This can be done in time $\mathcal{O}(\binom{n^2}{k+1} n^2) = n^{\mathcal{O}(k)}$ time. When $\mathtt{umim}(A) > k$, it is clear that computing $\mathcal{B}$ can be done in time $|\mathcal{A}| n^{\mathcal{O}(k)}$. This is also possible when $\mathtt{umim}(A) = k$ as Lemma 9 implies that $|\{N(X) \setminus A : X$ is an independent set in $\mathcal{A}\}| \leq n^k$. □

We obtain the following by using Theorem 11 with the routine of Lemma 12.

Theorem 13. *Given an n-vertex graph with a branch decomposition of o-mim-width k, we can solve* Independent Set *in time* $n^{\mathcal{O}(k)}$.

We show that the o-mim-width of a graph is upper bounded by its tree-independence number.

We say that a branch decomposition is *on a set* $V(G)$ if it is a branch decomposition of some function $\mathtt{f} : 2^{V(G)} \to \mathbb{Z}_{\geq 0}$. Next we give a general lemma for turning tree decompositions of G into branch decompositions on $V(G)$.

Lemma 14. *Let* $(T, \mathtt{bag})$ *be a tree decomposition of a graph* G*. There exists a branch decomposition* (T', δ) *on the set* $V(G)$ *so that for every bipartition* $(A, \overline{A})$ *of* $V(G)$ *given by an edge of* (T', δ)*, there exists a bag of* $(T, \mathtt{bag})$ *that contains either* $N(A)$ *or* $N(\overline{A})$*.*

Then we restate Theorem 2 and prove it using Lemma 14.

Theorem 2. *Any graph with tree-independence number* k *has o-mim-width at most* k*.*

Proof. Let G be a graph with tree-independence number k and $(T, \mathtt{bag})$ a tree decomposition of G with independence number $\alpha(T, \mathtt{bag}) = k$. By applying Lemma 14 we turn $(T, \mathtt{bag})$ into a branch decomposition on $V(G)$ so that for every partition $(A, \overline{A})$ of $V(G)$ given by the decomposition, either $N(A)$ or $N(\overline{A})$ has independence number at most k. Now, if $N(A)$ has independence number at most k, then $\mathtt{umim}(\overline{A}) \leq k$, and if $N(\overline{A})$ has independence number at most k, then $\mathtt{umim}(A) \leq k$, so we have that $\mathtt{omim}(A) \leq k$, and therefore the o-mim-width of the branch decomposition is at most k. □

With similar arguments, we also prove the following.

Theorem 15. *Any graph with minor-matching hypertreewidth* k *has sim-width at most* k*.*

4 Neighbor-Depth of Graphs of Bounded Sim-Width

In this section we show that graphs of bounded sim-width have poly-logarithmic neighbor-depth, i.e., Theorem 5. The idea of the proof will be that given a cut of bounded sim-width, we can delete a constant fraction of the edges going over the cut by deleting the closed neighborhood of a single vertex. This allows to first fix a balanced cut according to an optimal decomposition for sim-width, and then delete the edges going over the cut in logarithmic depth.

We say that a vertex $v \in V(G)$ *neighbor-controls* an edge $e \in E(G)$ if e is incident to a vertex in $N[v]$. In other words, v neighbor-controls e if $e \notin E(G \setminus N[v])$.

Lemma 16. *Let* G *be a graph and* $A \subseteq V(G)$ *so that* $\mathtt{sim}(A) \leq k$*. There exists a vertex* $v \in V(G)$ *that neighbor-controls at least* $|E(A, \overline{A})|/2k$ *edges in* $E(A, \overline{A})$*.*

Proof. Suppose the contradiction, i.e., that all vertices of G neighbor-control less than $|E(A, \overline{A})|/2k$ edges in $E(A, \overline{A})$. Let $M \subseteq E(A, \overline{A})$ be a maximum induced $(A, \overline{A})$-matching, having size at most $|M| \leq \mathtt{sim}(A) \leq k$, and let $V(M)$ denote the set of vertices incident to M. Now, an edge in $E(A, \overline{A})$ cannot be added to M if and only if one of its endpoints is in $N[V(M)]$. In particular, an edge in $E(A,\overline{A})$ cannot be added to M if and only if there is a vertex in $V(M)$ that neighbor-controls it. However, by our assumption, the vertices in $V(M)$ neighbor-control strictly less than

$$|V(M)| \cdot |E(A,\overline{A})|/2k = |E(A,\overline{A})|$$

edges of $E(A,\overline{A})$, so there exists an edge in $E(A,\overline{A})$ that is not neighbor-controlled by $V(M)$, and therefore we contradict the maximality of M. ◻

Now, the idea will be to argue that because sim-width is at most k, there exists a balanced cut $(A,\overline{A})$ with $\mathtt{sim}(A) \le k$, and then select the vertex v given by Lemma 16 as the pivot-vertex. Here, we need to be careful to persistently target the same cut until the graph is disconnected along it.

Theorem 5. *Any n-vertex graph of sim-width k has neighbor-depth* $\mathcal{O}(k \log^2 n)$

Proof. For integers $n \ge 2$ and $k,t \ge 0$, we denote by $\mathtt{nd}(n,k,t)$ the maximum neighbor-depth of a graph that

1. has at most n vertices,
2. has sim-width at most k, and
3. has a cut $(A,\overline{A})$ with $\mathtt{sim}(A) \le k$, $|E(A,\overline{A})| \le t$, $|A| \le 2n/3$, and $|\overline{A}| \le 2n/3$.

We observe that if a graph G satisfies all of the conditions 1–3, then any induced subgraph of G also satisfies the conditions. In particular, note that n can be larger than $|V(G)|$, and in the condition 3, the cut should be balanced with respect to n but not necessarily with respect to $|V(G)|$.

We will prove by induction that

$$\mathtt{nd}(n,k,t) \le 1 + 4k(\log_{3/2}(n) \cdot \log(n^2+1) + \log(t+1)). \tag{1}$$

This will then prove the statement, because by Lemma 8 any graph with n vertices and sim-width k satisfies the conditions with $t = n^2$.

First, when $n \le 2$ this holds because any graph with at most two vertices has neighbor-depth at most one. We then assume that $n \ge 3$ and that Eq. (1) holds for smaller values of n and first consider the case $t = 0$.

Let G be a graph that satisfies the conditions 1–3 with $t = 0$. Because $t = 0$, each connected component of G has at most $2n/3$ vertices, and therefore satisfies the conditions with $n' = 2n/3$, $k' = k$, and $t' = (2n/3)^2$. Therefore, by induction each component of G has neighbor-depth at most $\mathtt{nd}(2n/3, k, (2n/3)^2)$. Because the neighbor-depth of G is the maximum neighbor-depth over its components, we get that

$$\begin{aligned}
\mathtt{nd}(G) &\le \mathtt{nd}(2n/3, k, (2n/3)^2) \\
&\le 1 + 4k(\log_{3/2}(2n/3) \cdot \log((2n/3)^2+1) + \log((2n/3)^2+1)) \\
&\le 1 + 4k((\log_{3/2}(n)-1) \cdot \log((2n/3)^2+1) + \log((2n/3)^2+1)) \\
&\le 1 + 4k(\log_{3/2}(n) \cdot \log((2n/3)^2+1)) \le 1 + 4k(\log_{3/2}(n) \cdot \log(n^2+1)),
\end{aligned}$$

which proves that Eq. (1) holds when $t = 0$.

We then consider the case when $t \geq 1$. Assume that Eq. (1) does not hold and let G be a counterexample that is minimal under induced subgraphs. Note that this implies that G is connected, and every proper induced subgraph G' of G has neighbor-depth at most $1 + 4k(\log_{3/2}(n) \cdot \log(n^2+1) + \log(t+1))$. We can also assume that $t = |E(A, \overline{A})|$.

Now, by Lemma 16 there exists a vertex $v \in V(G)$ that neighbor-controls at least $t/2k$ edges in $E(A, \overline{A})$. We will select v as the pivot-vertex. By the minimality of G, we have that $\mathtt{nd}(G \setminus \{v\}) \leq 1 + 4k(\log_{3/2}(n) \cdot \log(n^2+1) + \log(t+1))$, so it suffices to prove that $\mathtt{nd}(G \setminus N[v]) \leq 1 + 4k(\log_{3/2}(n) \cdot \log(n^2 + 1) + \log(t+1)) - 1$. Because v neighbor-controls at least $t/2k$ edges in $E(A, \overline{A})$, the graph $G \setminus N[v]$ satisfies the conditions with $n' = n$, $k' = k$, and $t' = t - t/2k$. We denote

$$\alpha = \frac{t'+1}{t+1} = 1 - \frac{t/2k}{t+1} \leq 1 - \frac{t/2k}{2t} \leq 1 - \frac{1}{4k}.$$

Now we have that

$$\begin{aligned}
\mathtt{nd}(G) &\leq \mathtt{nd}(n, k, t - t/2k) + 1 \leq\ 2 + 4k(\log_{3/2}(n) \cdot \log(n^2+1) + \log(\alpha \cdot (t+1))) \\
&\leq 2 + 4k(\log_{3/2}(n) \cdot \log(n^2+1) + \log(\alpha) + \log(t+1)) \\
&\leq 2 + 4k\log(\alpha) + 4k(\log_{3/2}(n) \cdot \log(n^2+1) + \log(t+1)) \\
&\leq 2 - 4k \cdot \frac{1}{4k} + 4k(\log_{3/2}(n) \cdot \log(n^2+1) + \log(t+1)) \\
&\leq 1 + 4k(\log_{3/2}(n) \cdot \log(n^2+1) + \log(t+1)),
\end{aligned}$$

which proves that Eq. (1) holds when $t \geq 1$, and therefore completes the proof. □

5 Conclusion

We conclude with some open problems. First, as already discussed, it is still open if independent set can be solved in polynomial-time on graphs of bounded mim-width, because it is not known how to construct a decomposition of bounded mim-width if one exists. It would be very interesting to resolve this problem by either giving an algorithm for computing decompositions of bounded mim-width, or by defining an alternative width parameter that is more general than mim-width and allows to solve Independent Set in polynomial-time when the parameter is bounded.

The class of graphs of polylogarithmic neighbor-depth generalizes several classes where Independent Set can be solved in (quasi)polynomial time. Another interesting class where Independent Set can be solved in polynomial-time and which, to our knowledge, could have polylogarithmic neighbor-depth is the class of graphs with polynomial number of minimal separators [12]. It would be interesting to show that this class has polylogarithmic neighbor-depth. More generally, Korhonen [21] studied a specific model of dynamic programming algorithms for Independent Set, in particular, tropical circuits for independent set, and it appears plausible that all graphs with polynomial size tropical circuits for independent set could have polylogarithmic neighbor-depth.

References

1. Bergougnoux, B., Dreier, J., Jaffke, L.: A logic-based algorithmic meta-theorem for mim-width. In: Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 3282–3304. SIAM (2023). https://doi.org/10.1137/1.9781611977554.ch125
2. Bergougnoux, B., Korhonen, T., Razgon, I.: New width parameters for independent set: one-sided-mim-width and neighbor-depth. CoRR abs/2302.10643 (2023). https://doi.org/10.48550/arXiv.2302.10643
3. Bonamy, M., et al.: Sparse graphs with bounded induced cycle packing number have logarithmic treewidth. In: Bansal, N., Nagarajan, V. (eds.) Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, 22–25 January 2023, pp. 3006–3028. SIAM (2023). https://doi.org/10.1137/1.9781611977554.ch116
4. Brettell, N., Horsfield, J., Munaro, A., Paesani, G., Paulusma, D.: Bounding the mim-width of hereditary graph classes. J. Graph Theory **99**(1), 117–151 (2022). https://doi.org/10.1002/jgt.22730
5. Brettell, N., Horsfield, J., Munaro, A., Paulusma, D.: List k-colouring P_t-free graphs: a mim-width perspective. Inf. Process. Lett. **173**, 106168 (2022). https://doi.org/10.1016/j.ipl.2021.106168
6. Bui-Xuan, B., Telle, J.A., Vatshelle, M.: Boolean-width of graphs. Theor. Comput. Sci. **412**(39), 5187–5204 (2011). https://doi.org/10.1016/j.tcs.2011.05.022
7. Courcelle, B., Engelfriet, J., Rozenberg, G.: Handle-rewriting hypergraph grammars. J. Comput. Syst. Sci. **46**(2), 218–270 (1993). https://doi.org/10.1016/0022-0000(93)90004-G
8. Courcelle, B., Makowsky, J.A., Rotics, U.: Linear time solvable optimization problems on graphs of bounded clique-width. Theory Comput. Syst. **33**(2), 125–150 (2000). https://doi.org/10.1007/s002249910009
9. Dallard, C., Milanič, M., Štorgel, K.: Treewidth versus clique number. II. Tree-independence number. CoRR abs/2111.04543 (2022). https://doi.org/10.48550/arXiv.2111.04543
10. Diestel, R.: Graph Theory. Graduate Texts in Mathematics, vol. 173, 4th edn. Springer, Cham (2012)
11. Eiben, E., Ganian, R., Hamm, T., Jaffke, L., Kwon, O.: A unifying framework for characterizing and computing width measures. In: Braverman, M. (ed.) 13th Innovations in Theoretical Computer Science Conference, ITCS 2022, Berkeley, CA, USA, 31 January–3 February 2022. LIPIcs, vol. 215, pp. 63:1–63:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2022). https://doi.org/10.4230/LIPIcs.ITCS.2022.63
12. Fomin, F.V., Todinca, I., Villanger, Y.: Large induced subgraphs via triangulations and CMSO. SIAM J. Comput. **44**(1), 54–87 (2015). https://doi.org/10.1137/140964801
13. Gartland, P., Lokshtanov, D.: Independent set on P_k-free graphs in quasi-polynomial time. In: Irani, S. (ed.) 61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, 16–19 November 2020, pp. 613–624. IEEE (2020). https://doi.org/10.1109/FOCS46700.2020.00063

14. Gartland, P., Lokshtanov, D., Pilipczuk, M., Pilipczuk, M., Rzazewski, P.: Finding large induced sparse subgraphs in $C_{>t}$-free graphs in quasipolynomial time. In: Khuller, S., Williams, V.V. (eds.) STOC 2021: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, 21–25 June 2021, pp. 330–341. ACM (2021). https://doi.org/10.1145/3406325.3451034
15. Gavril, F.: Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph. SIAM J. Comput. **1**(2), 180–187 (1972). https://doi.org/10.1137/0201013
16. Grzesik, A., Klimosová, T., Pilipczuk, M., Pilipczuk, M.: Polynomial-time algorithm for maximum weight independent set on P_6-free graphs. ACM Trans. Algorithms **18**(1), 4:1–4:57 (2022). https://doi.org/10.1145/3414473
17. Jaffke, L., Kwon, O., Strømme, T.J.F., Telle, J.A.: Mim-width III. Graph powers and generalized distance domination problems. Theor. Comput. Sci. **796**, 216–236 (2019). https://doi.org/10.1016/j.tcs.2019.09.012
18. Jaffke, L., Kwon, O., Telle, J.A.: Mim-width I. Induced path problems. Discret. Appl. Math. **278**, 153–168 (2020). https://doi.org/10.1016/j.dam.2019.06.026
19. Jaffke, L., Kwon, O., Telle, J.A.: Mim-width II. The feedback vertex set problem. Algorithmica **82**(1), 118–145 (2020). https://doi.org/10.1007/s00453-019-00607-3
20. Kang, D.Y., Kwon, O., Strømme, T.J.F., Telle, J.A.: A width parameter useful for chordal and co-comparability graphs. Theor. Comput. Sci. **704**, 1–17 (2017). https://doi.org/10.1016/j.tcs.2017.09.006
21. Korhonen, T.: Lower bounds on dynamic programming for maximum weight independent set. In: 48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, Glasgow, Scotland, 12–16 July 2021 (Virtual Conference), pp. 87:1–87:14 (2021). https://doi.org/10.4230/LIPIcs.ICALP.2021.87
22. Munaro, A., Yang, S.: On algorithmic applications of sim-width and mim-width of (H_1, H_2)-free graphs. CoRR abs/2205.15160 (2022). https://doi.org/10.48550/arXiv.2205.15160
23. Oum, S., Seymour, P.D.: Approximating clique-width and branch-width. J. Comb. Theory Ser. B **96**(4), 514–528 (2006). https://doi.org/10.1016/j.jctb.2005.10.006
24. Razgon, I.: Classification of OBDD size for monotone 2-CNFs. In: Golovach, P.A., Zehavi, M. (eds.) 16th International Symposium on Parameterized and Exact Computation, IPEC 2021, Lisbon, Portugal, 8–10 September 2021. LIPIcs, vol. 214, pp. 25:1–25:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021). https://doi.org/10.4230/LIPIcs.IPEC.2021.25
25. Robertson, N., Seymour, P.D.: Graph minors. III. Planar tree-width. J. Comb. Theory Ser. B **36**(1), 49–64 (1984). https://doi.org/10.1016/0095-8956(84)90013-3
26. Vatshelle, M.: New width parameters of graphs. Ph.D. thesis, University of Bergen, Norway (2012). www.hdl.handle.net/1956/6166
27. Yolov, N.: Minor-matching hypertree width. In: Czumaj, A. (ed.) Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, 7–10 January 2018, pp. 219–233. SIAM (2018). https://doi.org/10.1137/1.9781611975031.16