

Largest Chordal and Interval Subgraphs Faster than 2^n

Ivan Bliznets¹ · Fedor V. Fomin² ·
Michał Pilipczuk³ · Yngve Villanger²

© The Author(s) 2015. This article is published with open access at Springerlink.com

Abstract We prove that in a graph with n vertices, induced chordal and interval subgraphs with the maximum number of vertices can be found in time $\mathcal{O}(2^{\lambda n})$ for some $\lambda < 1$. These are the first algorithms breaking the trivial $2^n n^{\mathcal{O}(1)}$ bound of the brute-force search for these problems.

Keywords Exact exponential algorithms · Chordal graphs · Interval graphs · Maximum induced \mathcal{H} -subgraph

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007–2013)/ERC Grant Agreement No. 267959 and the Research Council of Norway (Yggdrasil mobility program 2013–2014, Project Number 227328/F11), as well as from the Government of the Russian Federation (Grant 14.Z50.31.0030) and Grant of the President of Russian Federation (MK-6550.2015.1). This research was done while M. Pilipczuk was affiliated with the University of Bergen and was supported by the aforementioned ERC Grant. A preliminary version of this paper appeared in the proceedings of ESA 2013.

✉ Michał Pilipczuk
michal.pilipczuk@mimuw.edu.pl

Ivan Bliznets
iabliznets@gmail.com

Fedor V. Fomin
fomin@ii.uib.no

Yngve Villanger
yngve.villanger@gmail.com

¹ St. Petersburg Department of Steklov Institute of Mathematics, 27 Fontanka, St. Petersburg, Russia

² Department of Informatics, University of Bergen, Bergen, Norway

³ Institute of Informatics, University of Warsaw, Warsaw, Poland

Table 1 Known results for MAXIMUM INDUCED Π -SUBGRAPH

Property	Time complexity	References
Edgeless	$\mathcal{O}(1.2109^n)$	Robson [22]
Acyclic	$\mathcal{O}(1.7548^n)$	Fomin et al. [7]
Bipartite	$\mathcal{O}(1.62^n)$	Raman et al. [21]
Planar	$\mathcal{O}(1.7347^n)$	Fomin et al. [9]
d -degenerate	$\mathcal{O}((2 - \epsilon_d)^n)$	Pilipczuk and Pilipczuk [20]
Cluster graph	$\mathcal{O}(1.6181^n)$	Fomin et al. [6]
Biclique	$\mathcal{O}(1.3642^n)$	Gaspers et al. [14]
$o(n / \log n)$ treewidth	$\mathcal{O}(1.7347^n)$	Fomin and Villanger [11]
r -regular	$\mathcal{O}((2 - \epsilon_r)^n)$	Gupta et al. [17]
Matching	$\mathcal{O}(1.6957^n)$	Gupta et al. [17]

1 Introduction

The area of exact exponential algorithms is about solving intractable problems faster than the trivial exhaustive search, though still in exponential time [8]. In this paper, we give algorithms computing maximum induced chordal and interval subgraphs in a graph faster than the trivial brute-force search. These problems are interesting cases of a more general meta-problem MAXIMUM INDUCED Π -SUBGRAPH, where for a given graph G and hereditary class of graphs Π one asks for an induced subgraph belonging Π with the maximum possible number of vertices. (Let us recall that a class of graphs is *hereditary* if it is closed under taking induced subgraphs).

By the result of Lewis and Yannakakis [19], the MAXIMUM INDUCED Π -SUBGRAPH problem is NP-hard for every non-trivial class of graphs Π . Different classes Π were studied in the literature; examples include classes of graphs that are edgeless, planar, outerplanar, bipartite, complete bipartite, acyclic, degree-constrained, chordal etc. From the point of view of exact algorithms, as far as membership in Π can be tested in polynomial time, a trivial brute-force search trying all possible vertex subsets of G solves MAXIMUM INDUCED Π -SUBGRAPH in time $\mathcal{O}^*(2^n)$ on a graph G with n -vertices.¹ However, many algorithms for MAXIMUM INDUCED Π -SUBGRAPH which are faster than $\mathcal{O}^*(2^n)$ can be found in the literature for explicit classes Π . Notable examples are Π being the class of graphs that are edgeless [22] (equivalent to MAXIMUM INDEPENDENT SET), acyclic [7] (equivalent to MAXIMUM INDUCED FOREST), bipartite [21], planar [9], degenerate [20], regular [17], cluster graphs [6], bounded treewidth [11], or bicliques [14], see Table 1. Very recently, Fomin et al. [10] have shown that for every hereditary class of graphs Π that have constant treewidth and are definable in counting monadic second-order logic (CMSO), the MAXIMUM INDUCED Π -SUBGRAPH problem can be solved in $\mathcal{O}(1.7347^n)$ time.

The listed examples suggest that existence of algorithms faster than 2^n for MAXIMUM INDUCED Π -SUBGRAPH can be a phenomenon of a much more general nature.

¹ In this paper we use the $\mathcal{O}^*(\cdot)$ notation suppressing polynomial terms in the input size.

In fact, so far we do not know any efficiently recognizable class Π for which a lower bound could be derived. Therefore, the following is a folklore open problem in the field; we could not find it explicitly stated in the literature, so we state it below.

Hypothesis 1 For every hereditary graph class Π that can be recognized in polynomial time, the MAXIMUM INDUCED Π -SUBGRAPH problem can be solved in time $\mathcal{O}^*(2^{\lambda n})$ for some $\lambda < 1$.

We expect that some additional weak conditions on Π may be needed to provide a positive answer to hypothesis we discuss propositions of such relaxations in Sect. 6. Thus, the aforementioned results [6, 7, 9, 10, 14, 20–22] can be viewed as partial progress on Hypothesis 1: the idea is to investigate how different features of the class Π can be used to design an algorithm breaking the 2^n barrier.

Since every hereditary class of graphs Π can be characterized by a (not necessarily finite) set of forbidden induced subgraphs, there is an equivalent formulation of the MAXIMUM INDUCED Π -SUBGRAPH problem. For a set of graphs \mathcal{F} , a graph G is called \mathcal{F} -free if it contains no graph from \mathcal{F} as an induced subgraph. The MAXIMUM \mathcal{F} -FREE SUBGRAPH problem is to find a maximum induced \mathcal{F} -free subgraph of G . Clearly, if \mathcal{F} is the set of forbidden induced subgraphs for Π , then the MAXIMUM INDUCED Π -SUBGRAPH problem and the MAXIMUM \mathcal{F} -FREE SUBGRAPH problem are equivalent.

It is well known that when the set \mathcal{F} is *finite*, then MAXIMUM \mathcal{F} -FREE SUBGRAPH can be solved in time $\mathcal{O}^*(2^{\lambda n})$, where $\lambda < 1$. This can be seen by applying a simple branching argument, see Proposition 2, or by reducing to the d -HITTING SET problem, which is solvable faster than $\mathcal{O}^*(2^n)$ for every fixed d [6, 13]. Examples of \mathcal{F} -free classes of graphs for some finite set \mathcal{F} are split graphs, cographs, line graphs or trivially perfect graphs; see the book [3] for more information on these graph classes.

It is however completely unclear if anything faster than the trivial brute-force is possible in the case when \mathcal{F} is an infinite set, even when \mathcal{F} consists of very simple graphs. One of the most known and well studied classes of \mathcal{F} -free graphs is the class of *chordal graphs*, where \mathcal{F} is the set of all cycles of length more than three. Chordal graphs form a fundamental class of graphs whose properties are well understood. Another fundamental class of graphs is the class of *interval graphs*. We refer to the book of Golumbic for an overview of properties and applications of chordal and interval graphs [16]. In spite of nice structural properties of these graphs, no exact algorithms for MAXIMUM INDUCED CHORDAL SUBGRAPH and MAXIMUM INDUCED INTERVAL SUBGRAPH problems better than the trivial $\mathcal{O}^*(2^n)$ were known prior to our work.

Our results We define four properties of a graph class and give an algorithm that, for every fixed graph class Π (not part of the input) satisfying these properties, and for a given n -vertex graph G , finds a maximum induced subgraph of G belonging to Π in time $\mathcal{O}^*(2^{\lambda n})$ for some $\lambda < 1$, where λ depends only on the class Π . Because classes of chordal and interval graphs satisfy the required properties, as an immediate corollary we obtain that MAXIMUM INDUCED CHORDAL SUBGRAPH and MAXIMUM INDUCED INTERVAL SUBGRAPH can be solved in time $\mathcal{O}^*(2^{\lambda n})$ for some $\lambda < 1$.

When pipelined with simple branching arguments, our algorithms can be used to obtain time $\mathcal{O}^*(2^{\lambda n})$ algorithms for some $\lambda < 1$ for a variety of other MAXIMUM

INDUCED Π -SUBGRAPH problems, where class Π comprises chordal/interval graphs that moreover contain no induced subgraph from a *finite* forbidden set of graphs. Examples of such classes are proper interval graphs, Ptolemaic graphs, block graphs; see [3] for definitions and discussions of these graph classes.

The main intention of our work was to break the trivial 2^n barrier, and thus to provide a new insight into Hypothesis 1 by analyzing chordal and chordal-like graph classes. For this reason, we did not try to optimize the constant λ in the exponent. There are several places where the running time of our algorithm can be improved at a cost of more involved arguments or intensive case analyses. However, we would like to stress again that the main motivation of our work is the theoretical study of Hypothesis 1, rather than pursuit of really efficient algorithms for the respective problems. Therefore, we refrain from giving these improvements and prefer keeping the arguments as simple as possible.

Organization In Sect. 2 we give notation and recall known tools that will be used later. In Sect. 3 we discuss the four properties of a graph class that are needed for our algorithm to be applicable. Section 4 contains the description of the algorithm. For the convenience of the reader, in Sect. 5 we summarize the order of choice of small constants used by the algorithm. Finally, in Sect. 6 we give some concluding remarks.

2 Preliminaries

Standard graph notation We denote by $G = (V, E)$ a finite, undirected and simple graph with vertex set $V(G) = V$ and edge set $E(G) = E$. We also use n to denote the number of vertices in G . For a nonempty subset of vertices $W \subseteq V$, a subgraph induced by W is defined as $G[W] = (W, E \cap (W \times W))$. An *induced subgraph* of a graph is a subgraph induced by some subset of vertices. A *clique* is a set of vertices inducing a complete subgraph. We say that a vertex set $W \subseteq V$ is *connected* if $G[W]$ is connected. The *open neighborhood* of a vertex v is $N(v) = \{u \in V : uv \in E\}$ and the *closed neighborhood* is $N[v] = N(v) \cup \{v\}$. For a vertex set $W \subseteq V$ we put $N(W) = \bigcup_{v \in W} N(v) \setminus W$ and $N[W] = N(W) \cup W$. Whenever the graph to which the neighbourhood operation is applied is not clear from the context, we put it in the subscript of N . For a vertex subset X of a graph G , we use $G \setminus X$ to denote the subgraph of G induced by $V(G) \setminus X$.

A *graph class* Π is simply a family of graphs. We often use terms Π -graph or Π -subgraph to express membership in Π . We say that a graph class is *hereditary* if Π is closed under taking induced subgraphs. Every hereditary graph class can be described by a (possibly infinite) list of minimal forbidden induced subgraphs \mathcal{F}_Π : graph G is in Π if and only if it does not contain any induced subgraph from \mathcal{F}_Π , and for each $H \in \mathcal{F}_\Pi$ every induced subgraph of H , apart from H itself, belongs to Π . The class of graphs not containing any induced subgraph from a list \mathcal{F} will be denoted by \mathcal{F} -free graphs.

Chordal graphs are graphs not containing any induced cycles of length more than three, that is, chordal graphs are \mathcal{F} -free graphs where the set \mathcal{F} consists of all cycles of length more than three. Chordal graphs are hereditary and polynomial-time recognizable [16, 23]. Chordal graphs admit many more characterizations, for example they

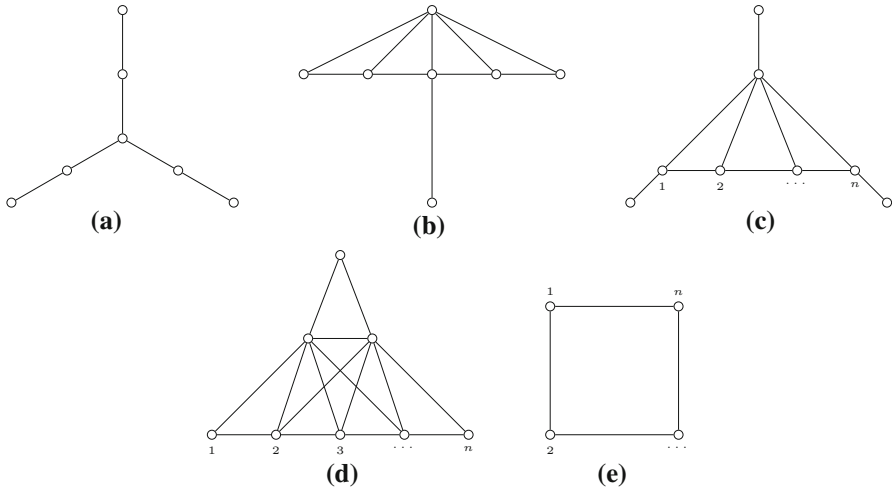


Fig. 1 Forbidden induced subgraphs for interval graphs. **a** bipartite claw, **b** umbrella, **c** n -net, $n \geq n2$, **d** n -tent, $n \geq 3$, **e** Cn , $n \geq 4$

are exactly graphs admitting a tree decomposition where every bag is a clique. By combining this with a well-known fact that every tree decomposition contains a bag whose removal splits the graph into connected components, each having at most half of the number of vertices, we obtain the following useful corollary.

Proposition 1 [15] *If H is a chordal graph, then there exists a clique S in H and a partition of $V(H) \setminus S$ into two subsets X_1, X_2 , such that*

- $|X_1|, |X_2| \leq \frac{2}{3}|V(H)|$, and
- there is no edge between X_1 and X_2 .

Such a set S is called a $\frac{2}{3}$ -balanced clique separator in H . Proposition 1 follows from a similar result on trees and tree decomposition of chordal graphs. Note that since $|X_2| \leq \frac{2}{3}|V(H)|$, then $|X_1| = |V(H)| - |S| - |X_2| \geq \frac{1}{3}|V(H)| - |S|$, and the same holds also for X_2 .

Interval graphs form a subclass of chordal graphs admitting a decomposition into a clique path instead of less restrictive clique tree. Interval graphs are also hereditary and polynomial-time recognizable [2, 16]. Their characterization in terms of minimal forbidden induced subgraphs was given by Lekkerkerker and Boland [18]; see Fig. 1 for reference. The book of Golombic [16] provides a thorough introduction to chordal and interval graphs.

We now describe the classical tools needed for the algorithm. The following result basically follows from the observation that branching on forbidden structures of constant size always leads to complexity better than 2^n .

Proposition 2 *Let \mathcal{F} be a finite set of graphs and let ℓ be the maximum number of vertices in a graph from \mathcal{F} . Let Π be a hereditary graph class that is polynomial-time recognizable. Assume that there exists an algorithm \mathcal{A} that for a given \mathcal{F} -free graph G on n vertices, in $\mathcal{O}^*(2^{\ell n})$ time finds a maximum induced Π -subgraph of G , for*

some $\epsilon < 1$. Then there exists an algorithm \mathcal{A}' that for a given graph G on n vertices, finds a maximum induced \mathcal{F} -free Π -graph in G in time $\mathcal{O}^*(2^{\epsilon n})$, where $\epsilon' < 1$ is a constant depending on ϵ and ℓ .

Proof Let Π' be the class of \mathcal{F} -free Π -graphs; since ℓ is a constant, Π' is polynomial-time recognizable. Algorithm \mathcal{A}' , given a graph $G = (V, E)$ with n vertices, tries to find a maximum induced Π' -subgraph using standard branching arguments. At each point the algorithm maintains two disjoint sets $A, D \subseteq V$; at the starting point $A = D = \emptyset$. Given A, D , the algorithm tries to find a maximum size set X inducing a Π' -graph such that $A \subseteq X$ and $D \cap X = \emptyset$. Whenever we arrive at a situation when $|A \cup D| > (1 - \epsilon)n$, we stop the branching procedure and perform a brute force check on the remaining vertices of $V \setminus (A \cup D)$. That is, we examine all subsets $A' \subseteq V \setminus (A \cup D)$ and test whether $G[A \cup A']$ induces a Π' -graph. This takes time $\mathcal{O}^*(2^{|V \setminus (A \cup D)|}) \leq \mathcal{O}^*(2^{\epsilon n})$.

At each step of the branching procedure we check in polynomial time whether $G \setminus D$ contains a subgraph isomorphic to any graph of \mathcal{F} . Assume first that we have found such a subgraph and let $S \subseteq V \setminus D$ be its vertex set. Clearly, for every induced Π' -subgraph, at least one of vertices of S is not contained in this subgraph. As vertices of $S \cap A$ have to be in the solution searched in this branch, we branch on set $S \setminus A$. More precisely, for every partition (A', D') of $S \setminus A$ where D' is nonempty, we produce a branch in which A' is incorporated into A and D' is incorporated into D . Note that this leads to $2^{\ell'} - 1$ branches produced and increasing $|A \cup D|$ by ℓ' , where $\ell' = |S \setminus A| \leq \ell$. Note moreover that since $\ell' \leq \ell$, then $2^{\ell'} - 1 \leq 2^{\epsilon \ell'}$ for some $\epsilon_\ell < 1$ depending on ℓ .

Assume now that $G \setminus D$ contains no induced subgraph from \mathcal{F} , hence it is \mathcal{F} -free. We apply the algorithm \mathcal{A} to $G \setminus D$ to compute the maximum induced Π -subgraph of $G \setminus D$. As $G \setminus D$ is \mathcal{F} -free, this subgraph is in fact in the class Π' . Note here that at this point we relax the condition that the set we are looking for has to contain A as a subset, however this does not affect correctness of the algorithm: the found subgraph is still an induced Π' -subgraph of G , so it can be only larger than the solution we are looking for in this branch. The running time of the application of \mathcal{A} is $\mathcal{O}^*(2^{\epsilon |V \setminus D|}) \leq \mathcal{O}^*(2^{\epsilon n})$.

Let us now discuss the running time of the algorithm. Note that at the point of applying brute-force check we have that $(1 - \epsilon)n + \ell \geq |A \cup D| > (1 - \epsilon)n$, as $|A \cup D|$ can increase by at most ℓ at each step. Each branching step increases $|A \cup D|$ by some ℓ' and introduces at most $2^{\epsilon \ell'}$ subbranches, hence the total number of instances where algorithm \mathcal{A} or a brute-force check is applied is at most $2^{\epsilon_\ell((1 - \epsilon)n + \ell)} = \mathcal{O}(2^{\epsilon_\ell(1 - \epsilon)n})$. Application of brute-force or algorithm \mathcal{A} takes $\mathcal{O}^*(2^{\epsilon n})$ time. Hence, the total running time is $\mathcal{O}^*(2^{\epsilon' n})$, where $\epsilon' = \epsilon_\ell(1 - \epsilon) + \epsilon < 1$. □

The following lemma of Fomin and Villanger [12] will be useful for us to guess connected sets of vertices with small running-time overhead.

Proposition 3 [12] *Let $G = (V, E)$ be a graph. For every $v \in V$, and $b, f \geq 0$, the number of connected vertex subsets $B \subseteq V$ such that*

- $v \in B$,
- $|B| = b + 1$, and
- $|N(B)| = f$,

is at most $\binom{b+f}{b}$. Moreover, all such subsets can be enumerated in time $\mathcal{O}^*\left(\binom{b+f}{b}\right)$.

Along with Proposition 3 we use the following standard bound on binomial coefficients in terms of entropy.

Proposition 4 $\binom{n}{k} \leq 2^{H(\frac{k}{n})n}$ for any $n \geq 1$ and $1 \leq k \leq n - 1$, where $H(t) = -t \log_2 t - (1 - t) \log_2(1 - t)$.

Note that from Proposition 4 it follows that for $k = \alpha n$ and $\alpha \neq \frac{1}{2}$ we have $\binom{n}{k} = \mathcal{O}(2^{\kappa n})$ for some $\kappa < 1$.

The last necessary ingredient is the classical idea used by Schroepel and Shamir [24] for solving SUBSET SUM by reducing it to an instance of 2-TABLE. In the 2-TABLE problem, we are given two $k \times m_i$ matrices $T_i, i = 1, 2$, and a vector $\mathbf{s} \in \mathbb{Q}^k$. Columns of each matrix are m_i vectors of \mathbb{Q}^k . The question is, if there is a column of the first matrix and a column of the second matrix such that the sum of these two columns is equal to \mathbf{s} . A trivial solution to the 2-TABLE problem would be to try all possible pairs of vectors; however, this problem can be solved more efficiently. We can sort columns of T_1 lexicographically in $\mathcal{O}(km_1 \log m_1)$ time, and for every column \mathbf{v} of T_2 check whether T_1 contains a column equal to $\mathbf{s} - \mathbf{v}$ in $\mathcal{O}(k \log m_1)$ time using binary search.

Proposition 5 [24] *The 2-TABLE problem can be solved in time $\mathcal{O}((m_1+m_2)k \log m_1)$.*

3 Properties of the graph class

In this section we gather the required properties of the graph class Π for our algorithm to be applicable. We consider only hereditary subclasses of chordal graphs, hence our first property is the following.

Property (1) Π is a hereditary subclass of chordal graphs.

As Π is hereditary, it may be described by a list of vertex-minimal forbidden induced subgraphs \mathcal{F}_Π . We need the following properties of \mathcal{F}_Π :

Property (2) All graphs in \mathcal{F}_Π are connected, and none of them contains a clique of size $\aleph + 1$ for some universal constant \aleph .

For chordal graphs \mathcal{F}_Π consists of cycles of length at least 4, hence $\aleph = 2$. For interval graphs, an inspection of the list of forbidden induced subgraphs, depicted on Fig. 1, shows that we may take $\aleph = 4$. In the following, we always treat \aleph as a universal constant for class Π on which all the later constants may depend; moreover, \aleph may influence the exponents of polynomial factors hidden in the \mathcal{O}^* notation. An example of a subclass of chordal graphs not satisfying this property, is the class of strongly chordal graphs. The reason for that is that minimal forbidden subgraphs of strongly chordal graphs can contain a clique of any size, see [3] and Sect. 6 for more information on this class of graphs. Note that other examples of classes not satisfying Property (2) are classes where some forbidden induced subgraphs are disconnected, e.g. split graphs. However, whenever the number of disconnected graphs in \mathcal{F}_Π is finite (as in the

case of split graphs), they can be always handled separately using Proposition 2. See also Theorem 2 in Sect. 6, in which we explicitly combine the forthcoming Theorem 1 with Proposition 2. Finally, let us remark that the connectedness of all the forbidden induced subgraphs is equivalent to requiring Π to be closed under taking disjoint union.

Thirdly, we need our graph class to be efficiently recognizable.

Property (3) Π is polynomial-time recognizable. Chordal graphs and interval graphs have polynomial time recognition algorithms, see e.g. [2, 16, 23]. For our arguments to work we need one more algorithmic property. The property that we need can be described intuitively as robustness with respect to clique separators. More precisely, we need the following statement.

Property (4) There exists a polynomial-time algorithm \mathcal{A} that takes as input a graph G together with a clique S in G . The algorithm answers YES or NO, such that the following conditions are satisfied:

- If \mathcal{A} answers YES on inputs (G_1, S_1) and (G_2, S_2) where $|S_1| = |S_2|$, then graph G' , obtained by taking disjoint union of G_1 and G_2 and identifying every vertex of S_1 with a different vertex of S_2 in any manner, belongs to Π .
- If $G \in \Pi$, then there exists a clique separator S in G such that $V(G) \setminus S$ may be partitioned into two sets X_1, X_2 such that (i) $|X_1|, |X_2| \leq \frac{2}{3}|V(G)|$, (ii) there is no edge between X_1 and X_2 , (iii) \mathcal{A} answers YES on $(G[X_1 \cup S], S)$ and on $(G[X_2 \cup S], S)$.

We remark that some of the proposed properties might be relaxed. For example, one might replace Property (3) by the property that the graph class is subexponential-time recognizable. Also, the property of being a subclass of chordal graphs might be relaxed in some way. However, such relaxations do not increase the number of natural graph classes to which our algorithm might be applied, and hence we state the properties to be as natural and simple as possible.

Observe that Property (1) and Proposition 1 already provides us with some $\frac{2}{3}$ -balanced clique separator S of G . Shortly speaking, Property (4) requires that in addition belonging to Π may be tested by looking at $G[X_1 \cup S]$ and $G[X_2 \cup S]$ independently. For chordal graphs, Property (4) follows from Proposition 1 and a folklore observation that if S is a clique separator in a graph G , with (X_1, X_2) being a partition of $V(G) \setminus S$ such that there is no edge between X_1 and X_2 , then G is chordal if and only if $G[X_1 \cup S]$ and $G[X_2 \cup S]$ are chordal. Hence, we may take chordality testing for the algorithm \mathcal{A} .

For interval graphs, let us take the clique path of G and examine a clique separator S such that there is at most half of vertices on the left of it and at most half on the right of it. Let X_1 be the vertices before S on the clique path, and X_2 be the vertices after S . Clearly, S is then even a $\frac{1}{2}$ -balanced clique separator, with partition (X_1, X_2) of $V(G) \setminus S$. Then it follows that $G[X_1 \cup S]$ and $G[X_2 \cup S]$ admit clique paths in which S is one of the end bags of the path. On the other hand, assume that we are given any two graphs G_1, G_2 with equally sized cliques S_1, S_2 , such that G_1, G_2 admit clique paths with S_1, S_2 as the end bags. Then we may create a clique path of the graph G'

obtained from the disjoint union of G_1 and G_2 and identification of S_1 and S_2 in any manner, by simply taking the clique paths for G_1 and G_2 and identifying the end bags containing S_1 and S_2 , respectively. Hence, as \mathcal{A} we may take an algorithm which for input (G, S) checks whether G is interval and admits a clique path with S as the end bag. Such a test may be easily done as follows: we add two vertices v, v' to G , where v is adjacent to v' and to every vertex of S , while v' is adjacent only to v . In this manner we force S to be the end bag, and run the intervality test. Hence, interval graphs also satisfy Property (4).

4 The algorithm

In this section we prove the main result of the paper, which is the following.

Theorem 1 *If Π satisfies Properties (1)–(4), then there exists an algorithm which, given a graph G with n vertices, returns a maximum induced subgraph of G belonging to Π in time $\mathcal{O}^*(2^{\lambda n})$ for some $\lambda < 1$, where λ depends only on Π .*

As we already observed, chordal and interval graphs satisfy Properties (1)–(4). Thus Theorem 1 implies immediately the results claimed in the introduction. Our approach is based on a thorough investigation of the structure of a maximum induced subgraph. In each of the cases, we deploy a different strategy to identify possible candidates for an optimal solution. The properties we strongly rely on are the balanced separation property (Property (4)), and conditions on minimal forbidden induced subgraphs for Π (Property (2)). The flowchart of the algorithm is given in Fig. 2. A shorter description of the algorithm may be found in conference version of the paper [1].

Let $G = (V, E)$. In the description of the algorithm we use several small positive constants: $\alpha, \beta, \gamma, \delta, \varepsilon$, and one large constant L (for reader's convenience a table with a short description of constants is given in Sect. 5). The final constant λ depends on the choice of $\alpha, \beta, L, \gamma, \delta, \varepsilon$; during the description we make sure that constants $(\alpha, \beta, L, \gamma, \delta, \varepsilon)$ can be chosen so that $\lambda < 1$. The choice of each constant depends on the later ones, e.g., having chosen $L, \gamma, \delta, \varepsilon$, we may find a positive upper bound on the value of β so that we may choose any positive β smaller than this upper bound. For reader's convenience, in Sect. 5 we give a summary of the procedure of choosing constants.

Firstly, we observe that by Proposition 2, we may assume that the input graph does not contain any forbidden induced subgraph from \mathcal{F}_Π of size at most ℓ for some constant ℓ , to be determined later. Indeed, if we are able to find an algorithm for maximum induced Π -subgraph running in $\mathcal{O}^*(2^{\lambda n})$ time for some $\lambda < 1$ and working in \mathcal{F}'_Π -free graphs, where \mathcal{F}'_Π consists of graphs of \mathcal{F}_Π of size at most ℓ , then by Proposition 2 we obtain an algorithm for maximum induced Π -subgraph working in general graphs and with running time $\mathcal{O}^*(2^{\lambda' n})$ for some $\lambda' < 1$. Hence, from now on we assume that the input graph G does not contain any forbidden induced subgraph from \mathcal{F}_Π of size at most ℓ .

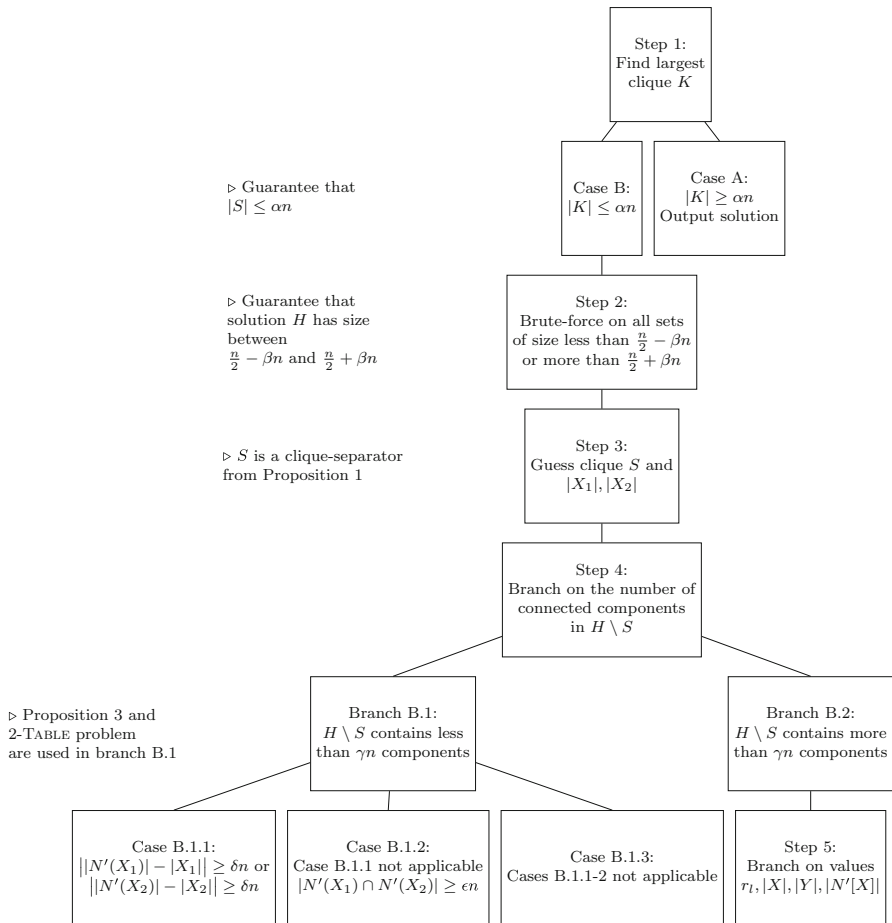


Fig. 2 Schematic representation of the algorithm

The algorithm performs a number of *steps*. After each step, depending on the result, the algorithm chooses one of the subcases (for reader convenience a schematic representation of algorithm is given in Fig. 2).

Step 1 Using the algorithm of Robson [22], find the largest clique K in G in $\mathcal{O}^*(2^{0.276n})$ time.

We consider two cases: either K is large enough to finish the search directly, or K is small and we have a guarantee that the maximum induced Π -graph we are looking for contains only small cliques. The threshold for small/large is αn for a constant α , $0 < \alpha < 1/48$, to be determined later.

Case A: $|K| \geq \alpha n$.

We show that in this case, the problem can be solved in $\mathcal{O}^*(2^{(1-(1-\kappa_0)\alpha)n})$ time for some $\kappa_0 < 1$ depending only on \aleph . We use the following auxiliary lemma.

Lemma 1 *Let P be a subset of the vertices of a graph G with n vertices that induces a graph belonging to Π , and let K be a clique in G such that $P \cap K = \emptyset$. Then in time $\mathcal{O}^*(2^{\kappa_0 \cdot |K|})$, for some $\kappa_0 < 1$ depending only on \aleph , it is possible to find an induced subgraph of G with the maximum number of vertices, where maximum is taken over all induced subgraphs H of G such that (i) $H \in \Pi$, (ii) $V(H) \setminus K = P$. In other words, the maximum is taken over all induced subgraphs belonging to Π which can be obtained by adding some vertices of K to P .*

Proof For every nonempty subset W of K of size at most \aleph , we colour W red if $G[W \cup P] \in \Pi$. Note that this construction may be performed using at most $\aleph \cdot |K|^\aleph$ tests of belonging to Π , hence in polynomial time for constant \aleph .

For every subset $X \subseteq K$, we observe that $G[P \cup X]$ belongs to Π if and only if all nonempty subsets of X of size at most \aleph are red. Indeed, if the latter is not the case, there is a subset $W \subseteq X$ such that $G[P \cup W] \notin \Pi$, so by Property (1) $G[P \cup X] \notin \Pi$ as well. For the opposite direction, let us assume that $G[P \cup X]$ contains some forbidden induced subgraph $F \in \mathcal{F}_\Pi$. Then $|F \cap X| > \aleph$ because otherwise, by the definition of the colouring, $F \cap X$ would not be coloured red. But since X is a clique, we conclude that F contains a clique on $\aleph + 1$ vertices, which is a contradiction with Property (2).

Hence, to obtain a maximum subgraph one has to find a maximum subset of X such that all its subsets of size at most \aleph are coloured red. This is equivalent to finding a maximum clique in a hypergraph with hyperedges of cardinality at most \aleph , which can be done using a branching algorithm in $\mathcal{O}^*(2^{\kappa_0 \cdot |K|})$ time for some $\kappa_0 < 1$, depending only on \aleph .

The branching algorithm maintains two disjoint sets of vertices A and D , at the beginning equal to \emptyset . Set A consists of vertices that are guessed to be in the solution, while D consists of vertices guessed to not be in the solution. The algorithm terminates the branch when $K \setminus D$ does not have any subset of size at most \aleph not coloured red. This makes $K \setminus D$ a candidate for the optimum X which is the largest set among the candidates. If the branch is not terminated, we infer that there must be a subset $W \subseteq K \setminus D$ of size at most \aleph which is not coloured red. Clearly, at least one of the vertices of W cannot be in the optimum X , hence we examine $W \setminus A$ and branch into $2^{|W \setminus A|} - 1$ cases, in each fixing a different choice which vertices of $W \setminus A$ should go to A and which should go to D ; the omitted case is when all the considered vertices go to A . As $|W \setminus A| \leq \aleph$, we have that $2^{|W \setminus A|} - 1 \leq 2^{\kappa_0 \cdot |W \setminus A|}$ for some $\kappa_0 < 1$ depending only on \aleph . Hence, we are able to fix alignment of $|W \setminus A|$ vertices by creating at most $2^{\kappa_0 \cdot |W \setminus A|}$ branches, and the total running time $2^{\kappa_0 \cdot |K|}$ follows. \square

To complete case A, let H be a maximum induced subgraph of G belonging to Π . We branch into at most $2^{|V \setminus K|}$ subcases, in each fixing a different subset P of $V \setminus K$ as $V(H) \setminus K$; we discard all the branches where the subgraph induced by P does not belong to Π . For each branch, we use Lemma 1 to find a maximum induced Π -subgraph which can be obtained from the guessed subset by adding vertices of K . This takes time $\mathcal{O}^*(2^{\kappa_0 \cdot |K|})$ for each branch. Thus the running time in this case is $\mathcal{O}^*(2^{|V \setminus K|} \cdot 2^{\kappa_0 \cdot |K|}) \leq \mathcal{O}^*(2^{(1-\alpha)n} \cdot 2^{\kappa_0 \cdot \alpha n}) = \mathcal{O}^*(2^{(1-(1-\kappa_0)\alpha)n})$. Note that $1 - (1 - \kappa_0)\alpha < 1$ for $\alpha > 0$ and $\kappa_0 < 1$.

Case B: *The graph G has no clique of size αn .*

Firstly, we search for solutions that have at most $n/2 - \beta n$ or at least $n/2 + \beta n$ vertices for some β , $0 < \beta < 1/16$, to be determined later. For this, we may apply a simple brute-force check that tries all vertex subsets of size at most $\lceil n/2 - \beta n \rceil$ or at least $\lfloor n/2 + \beta n \rfloor$ in time $\mathcal{O}^*\left(\binom{n}{\lceil n/2 - \beta n \rceil}\right)$; note that by Proposition 4 for $\beta > 0$ it holds that $\mathcal{O}^*\left(\binom{n}{\lceil n/2 - \beta n \rceil}\right) \leq \mathcal{O}^*(2^{\kappa_1 n})$ for some $\kappa_1 < 1$ depending on β .

Step 2 Test all subsets of size at most $n/2 - \beta n$ and at least $n/2 + \beta n$, whether they induce a graph from Π . Let H be a such induced subgraph of maximum size. If $|H| \geq n/2 + \beta n$, we output this solution and terminate the algorithm. If $|H| = \lfloor n/2 - \beta n \rfloor$, we proceed with the algorithm to test whether there is some induced subgraph of size between $n/2 - \beta n$ and $n/2 + \beta n$. If $|H| < \lfloor n/2 - \beta n \rfloor$, we again terminate the algorithm and output H ; the reason is that Π is hereditary, so if there would be some bigger H , the previous option would happen.

If execution of Step 2 did not terminate the algorithm, we know that the cardinality of the vertex set of a maximum induced subgraph belonging to Π is between $n/2 - \beta n$ and $n/2 + \beta n$. We proceed to further steps with this assumption.

Let H be a maximum induced Π -subgraph of G . We do not know how H looks and the only information about H we have so far is that:

- H has no clique of size αn , and
- that $n/2 - \beta n \leq |V(H)| \leq n/2 + \beta n$.

Let us note that the number of vertices of G not contained in H is also between $n/2 - \beta n$ and $n/2 + \beta n$.

We now use Property (4) to find a $\frac{2}{3}$ -balanced clique separator in H . More precisely, there is a clique S in H such that $V(H) \setminus S$ may be partitioned into sets X_1 and X_2 such that

- $\frac{1}{3}|V(H)| - |S| \leq |X_1|, |X_2| \leq \frac{2}{3}|V(H)|$, and
- there is no edge between X_1 and X_2 in G .

As S is also a clique in G , we have that $|S| \leq \alpha n$. Therefore, observe that $|X_1|, |X_2| \geq (\frac{1}{6} - \frac{\beta}{3} - \alpha)n > \frac{1}{8}n$, since $\beta < 1/16$ and $\alpha < 1/48$. Property (4) gives us more algorithmic properties of the partition (X_1, S, X_2) of $V(H)$; these properties will be useful later. As α is small, we may afford the following branching step.

Step 3 *Branch into at most $(1 + \alpha n) \binom{n}{\alpha n} \cdot (n + 1)^2$ subproblems, in each fixing a different subset of V of size at most αn as S , as well as the cardinalities of X_1, X_2 (we bound $\binom{n}{0} + \binom{n}{1} + \dots + \binom{n}{\alpha n}$ by $(1 + \alpha n) \cdot \binom{n}{\alpha n}$). Discard all the branches where S is not a clique.*

From now on we focus on one subproblem; hence, we assume that the clique S is fixed and the cardinalities of X_1, X_2 are known. Let $G' = G \setminus S$; to ease the notation, for $X \subseteq V(G')$ we denote $N'[X] = N_{G'}[X]$ and $N'(X) = N_{G'}(X)$. We consider two cases of how the structure of the optimal solution H may look like, depending on how many connected components $H \setminus S$ has. The threshold is γn for a small constant $\gamma > 0$ to be determined later.

Step 4 Branch into two subproblems: in the first branch assume that $H \setminus S$ has at most γn connected components, and in the second branch assume that $H \setminus S$ has more than γn connected components.

In the branches of Step 4 the algorithm checks several cases, and for every case proceeds with further branchings. To ease the description, we do not distinguish these branchings as separate Steps, but rather explain them in the text.

Branch B.1: Graph $H \setminus S$ has at most γn connected components.

We first branch into at most $(n + 1)^3$ subproblems, in each fixing the cardinalities of sets $N'(X_1)$, $N'(X_2)$ and $N'(X_1) \cap N'(X_2)$ such that $|N'(X_1) \cap N'(X_2)| \leq |N'(X_1)|, |N'(X_2)| \leq n - (|S| + |X_1| + |X_2|)$. From now on we assume that these cardinalities are fixed. We consider a few cases depending on the sizes of $N'(X_1)$, $N'(X_2)$ and $N'(X_1) \cap N'(X_2)$; in these cases we use small constants δ and ε , to be determined later.

Case B.1.1: $||N'(X_1)| - |X_1|| \geq \delta n$, or $||N'(X_2)| - |X_2|| \geq \delta n$.

We concentrate only on the subcase of $||N'(X_1)| - |X_1|| \geq \delta n$, as the second subcase is symmetric. As the number of components is small, their approximate location can be guessed at a cost of a small running time overhead as follows. Let P_1 be a set of vertices of size at most γn that is constructed by picking one vertex from every component of $G[X_1] = H[X_1]$. We branch into at most $(1 + \gamma n) \binom{n}{\gamma n}$ subproblems, in each fixing a different subset of size at most γn as P_1 .

We add an artificial vertex v_1 to G' , make it adjacent to all the vertices of P_1 , and enumerate all vertex sets of the new graph that (i) are connected, (ii) contain $P_1 \cup \{v_1\}$, (iii) are of size $|X_1| + 1$ and have neighbourhood of size $|N'(X_1)|$. By Proposition 3, the number of such sets is at most $\binom{|X_1| + |N'(X_1)|}{|X_1|}$ and they can be enumerated in time $\mathcal{O}^*(\binom{|X_1| + |N'(X_1)|}{|X_1|})$; note that here we enumerate candidates for such sets using Proposition 3 for vertex v_1 , and filter out all the subsets that do not contain P_1 . Clearly, $X_1 \cup \{v_1\}$ is among the enumerated sets.

We therefore branch into at most $\binom{|X_1| + |N'(X_1)|}{|X_1|}$ subproblems, in each fixing a different set out of the enumerated ones as X_1 (after excluding v_1). Moreover, in each subproblem we branch further into at most $\binom{n - |X_1| - |N'(X_1)|}{|X_2|}$ subproblems, in each fixing a different subset of $V \setminus (N'[X_1] \cup S)$ as X_2 . For each of these subproblems we check whether $G[X_1 \cup X_2 \cup S]$ belongs to Π in polynomial time.

Thus we obtain at most

$$(1 + \gamma n) \cdot \binom{n}{\gamma n} \cdot \binom{|X_1| + |N'(X_1)|}{|X_1|} \cdot \binom{n - |X_1| - |N'(X_1)|}{|X_2|}$$

subproblems. Since $|N'(X_1)|, |X_1| \leq n$ and $||N'(X_1)| - |X_1|| \geq \delta n$, by Proposition 4 we infer that $\binom{|X_1| + |N'(X_1)|}{|X_1|} \leq \mathcal{O}^*(2^{\kappa_2(|X_1| + |N'(X_1)|)})$ for some $\kappa_2 < 1$, depending on δ . On the other hand,

$$\binom{n - |X_1| - |N'(X_1)|}{|X_2|} \leq \mathcal{O}^*(2^{n-|X_1|-|N'(X_1)|}).$$

At the end of the discussion of Step 2 it was shown that $|X_1| \geq \frac{1}{8}n$, hence $|X_1| + |N'(X_1)| \geq \frac{1}{8}n$. So we have that in this case

$$\binom{|X_1| + |N'(X_1)|}{|X_1|} \cdot \binom{n - |X_1| - |N'(X_1)|}{|X_2|} = \mathcal{O}^*(2^{\kappa_3 n})$$

for some $\kappa_3 < 1$ depending on δ . Hence, the total number of branches produced by Case B.1.1, including the overheads from guessing S and cardinalities, is bounded by $\mathcal{O}^*\left(\binom{n}{\alpha n} \cdot \binom{n}{\gamma n} \cdot 2^{\kappa_3 n}\right)$. Given κ_3 , which depends on δ only, we may choose α and γ small enough so that this number is smaller than $\mathcal{O}^*(2^{\kappa_4 n})$ for some $\kappa_4 < 1$.

Case B.1.2: *Case B.1.1 does not apply, but $|N'(X_1) \cap N'(X_2)| \geq \varepsilon n$.*

We proceed similarly to Case B.1.1, but we change the strategy for guessing the set X_2 : instead of a brute-force guess in $V \setminus (N'[X_1] \cup S)$, we again make use of Proposition 3. Let P_1, P_2 be sets of vertices of size at most γn that are constructed by picking one vertex from every component of $G[X_1] = H[X_1]$ and $G[X_2] = H[X_2]$, respectively. Similarly as in the previous case, branch into at most $(1 + \gamma n)^2 \cdot \binom{n}{\gamma n}^2$ subproblems, in each fixing P_1 and P_2 . Again, we enumerate at most $\binom{|X_1| + |N'(X_1)|}{|X_1|}$ candidates for X_1 by adding an artificial vertex adjacent to all the vertices of P_1 , and then we branch into a number of subproblems, in each fixing one of these candidates as X_1 . We terminate all the branches where P_2 and X_1 are not disjoint, or there is an edge between them. Note that the total number of created subproblems is at most $\binom{|X_1| + |N'(X_1)|}{|X_1|} \leq \mathcal{O}^*(2^{2|X_1| + \delta n})$.

Now consider the graph $G'' = G \setminus (N'[X_1] \cup S)$. Note that $X_2 \subseteq V(G'')$ and the neighbourhood of X_2 in G'' is of size at most $|N'(X_2)| - \varepsilon n$, as at least εn vertices from the intersection with $N'(X_1)$ have been removed. Therefore, we can add an artificial vertex v_2 in G'' adjacent to all the vertices of P_2 , and apply Proposition 3 to it. Similarly as in the case of X_1 , we can enumerate at most

$$\binom{|X_2| + |N'(X_2)| - |N'(X_1) \cap N'(X_2)|}{|X_2|}$$

candidates for the set X_2 in time

$$\mathcal{O}^*\left(\binom{|X_2| + |N'(X_2)| - |N'(X_1) \cap N'(X_2)|}{|X_2|}\right).$$

Then we branch further into at most $\binom{|X_2| + |N'(X_2)| - |N'(X_1) \cap N'(X_2)|}{|X_2|}$ subproblems, in each fixing one of the candidates as X_2 . As $|N'(X_2)| \leq |X_2| + \delta n$ and $|N'(X_1) \cap N'(X_2)| \geq \varepsilon n$, we have that $|X_2| + |N'(X_2)| - |N'(X_1) \cap N'(X_2)| \leq 2|X_2| - (\varepsilon - \delta)n$.

When X_1 and X_2 are fixed, in polynomial time we check whether the graph $G[X_1 \cup X_2 \cup S]$ belongs to Π . Observe that $\mathcal{O}^*(2^{2|X_1| + \delta n}) \cdot \mathcal{O}^*(2^{2|X_2| - (\varepsilon - \delta)n}) =$

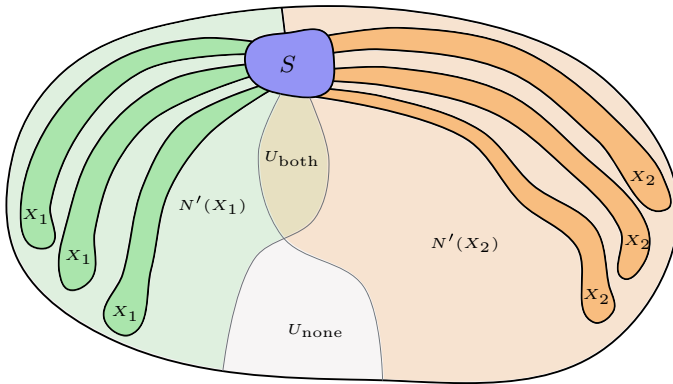


Fig. 3 Situation in Case B.1.3. Neighbourhoods $N'(X_1)$ and $N'(X_2)$ have almost equal sizes to X_1 and X_2 , respectively, while U_{both} and U_{none} contain only a tiny fraction of vertices

$\mathcal{O}^*(2^{2(|X_1|+|X_2|)-(\varepsilon-2\delta)n})$; moreover, $|X_1| + |X_2| \leq n/2 + \beta n$. Hence, given $\varepsilon > 0$ we may choose δ and β small enough so that $\mathcal{O}^*(2^{2|X_1|+\delta n}) \cdot \mathcal{O}^*(2^{2|X_2|-(\varepsilon-\delta)n}) \leq \mathcal{O}^*(2^{\kappa_5 n})$ for some $\kappa_5 < 1$ depending on ε . Now observe that the total number of branches produced in Case B.1.4, including overheads from guessing S , cardinalities, as well as P_1 and P_2 , is $\mathcal{O}^*\left(\binom{n}{\alpha n} \cdot \binom{n}{\gamma n}^2\right) \cdot \mathcal{O}^*(2^{\kappa_5 n})$, so given κ_5 we may choose α and γ small enough so that the total number of subbranches produced is at most $\mathcal{O}^*(2^{\kappa_6 n})$ for some $\kappa_6 < 1$. Every subbranch is then processed in polynomial time.

Case B.1.3: None of the cases B.1.1 or B.1.2 applies.

Summarizing, sets X_1 and X_2 have the following properties:

- $\frac{1}{6}n - \frac{\beta}{3}n - \alpha n \leq |X_1|, |X_2| \leq \frac{1}{3}n + \frac{2\beta}{3}n,$
- $\frac{1}{2}n - (\alpha + \beta)n \leq |X_1| + |X_2| \leq \frac{1}{2}n + \beta n,$
- $||N'(X_i)| - |X_i|| \leq \delta n$ for $i = 1, 2,$ and $|N'[X_1] \cap N'[X_2]| \leq \varepsilon n.$

Let $U_{\text{both}} = N'[X_1] \cap N'[X_2] = N'(X_1) \cap N'(X_2), U_{\text{none}} = V(G') \setminus (N'[X_1] \cup N'[X_2]),$ and $U = U_{\text{both}} \cup U_{\text{none}};$ see Fig. 3. We already know that $|U_{\text{both}}| \leq \varepsilon n.$ We now claim that $|U_{\text{none}}| \leq \zeta n,$ where $\zeta = 2\alpha + 2\beta + 2\delta + \varepsilon.$ Indeed, we have that

$$|U_{\text{none}}| = |V(G')| - |X_1| - |X_2| - |N'(X_1)| - |N'(X_2)| + |N'(X_1) \cap N'(X_2)| \leq n - 2(|X_1| + |X_2|) + 2\delta n + \varepsilon n \leq (2\alpha + 2\beta + 2\delta + \varepsilon)n.$$

Given that sets U_{both} and U_{none} are small, we may fix them with $\mathcal{O}^*\left(\binom{n}{\varepsilon n} \cdot \binom{n}{\zeta n}\right)$ overhead in the running time: we branch into $\mathcal{O}^*\left(\binom{n}{\varepsilon n} \cdot \binom{n}{\zeta n}\right)$ subproblems, in each fixing a pair of disjoint subsets U_{both} and U_{none} of $V \setminus S$ of cardinalities at most εn and $\zeta n,$ respectively. Note that then $V(G') \setminus U$ is the symmetric difference of $N'[X_1]$ and $N'[X_2];$ let $I = V(G') \setminus U.$ We are left with determining which part of I is in $X_1 \cup X_2,$ and which is outside.

Observe that every vertex of I is in exactly one of the two sets: $N[X_1]$ or $N[X_2].$ Hence, by Property (4) of $\Pi,$ we may look for subsets X_1, X_2 of $I,$ such that (i)

algorithm \mathcal{A} run on $G[X_1 \cup S]$ and $G[X_2 \cup S]$ with clique S distinguished provides a positive answer in both of the cases, and (ii) I is a disjoint union of $N[X_1]$ and $N[X_2]$. We model this situation as an instance of the 2-TABLE problem as follows. For $i = 1, 2$, enumerate all the subsets of I of size $|X_i|$ as candidates for X_i , and discard all the candidates for which the algorithm \mathcal{A} does not provide a positive answer when run on the subgraph induced by the candidate plus the clique S . For each remaining candidate subset create a binary vector of length $|I|$ indicating which vertices of I belong to its closed neighbourhood. Construct matrices T_1, T_2 by putting the vectors created for candidates for X_1, X_2 as columns of T_1, T_2 , respectively. Now, we need to check whether one can find a column of T_1 and a column of T_2 that sum up to a vector consisting only of ones.

As $|X_i| \leq \frac{1}{3}n + \frac{2\beta}{3}n$ for $i = 1, 2$, we have that tables T_1, T_2 have at most $\binom{n}{\frac{1}{3}n + \frac{2\beta}{3}n}$ columns, which is $\mathcal{O}^*(2^{\kappa_7 n})$ for some universal constant $\kappa_7 < 1$ (recall that $\beta < 1/16$, so $\frac{1}{3}n + \frac{2\beta}{3}n < \frac{3}{8}n$). Hence, by Proposition 5 we may solve the obtained instance of 2-TABLE in $\mathcal{O}^*(2^{\kappa_7 n})$ time. The total running time used by Case B.1.3, including the overheads for guessing clique S , set U and cardinalities, is $\mathcal{O}^*\left(\binom{n}{\alpha n} \cdot \binom{n}{\varepsilon n} \cdot \binom{n}{\zeta n} \cdot 2^{\kappa_7 n}\right)$; note that we may choose $\alpha, \beta, \delta, \varepsilon$ small enough so that this running time is $\mathcal{O}^*(2^{\kappa_8 n})$ for some $\kappa_8 < 1$.

Branch B.2: Graph $H \setminus S$ has more than γn connected components.

Consider connected components of $H \setminus S$ and fix a large constant $L > 2$ depending on γ , to be determined later. We say that a component containing at most $C = L/\gamma$ vertices is *small*, and otherwise it is *large*. Let r_ℓ and r_s be the numbers of large and small components of $H \setminus S$, respectively. The number of vertices contained in large components is hence at least $\frac{L \cdot r_\ell}{\gamma}$. Thus, $\frac{L \cdot r_\ell}{\gamma} \leq n, r_\ell \leq \frac{\gamma n}{L}$ and, consequently, $r_s \geq \gamma n - r_\ell \geq \gamma n(1 - \frac{1}{L}) \geq \frac{\gamma n}{2}$. Since small components are nonempty, they contain at least $\frac{\gamma n}{2}$ vertices in total.

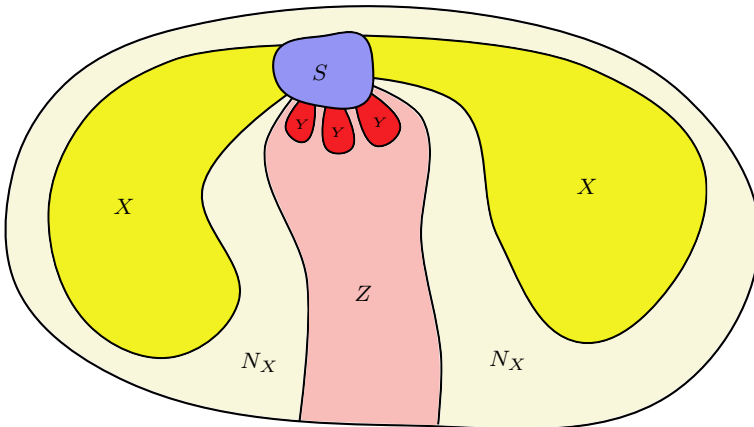


Fig. 4 Situation in Branch B.2. Note that even if we fix X and S , the remaining part $V \setminus (N'[X] \cup S)$ still needs to be partitioned between Y and Z

Let us summarize the situation; see Fig. 4 for reference. The vertices of V can be partitioned into disjoint sets $S, X, N_X, Y,$ and $Z,$ where

- (i) S is the clique guessed in Step 3;
- (ii) X are the vertices contained in large components of $H \setminus S;$
- (iii) $N_X = N'(X);$
- (iv) Y are the vertices contained in small components of $H \setminus S;$
- (v) Z consists of vertices not contained in H and not adjacent to $X.$

Note that $V(H) = S \cup X \cup Y.$ Unfortunately, even given X and $S,$ the algorithm still cannot deduce the solution: we still need to split the remaining part $V \setminus (N'[X] \cup S)$ into Y that will go into the solution, and Z that will be left out. However, as we know that $G[X]$ has a small number of components, we can proceed with a branching step that guesses X using Proposition 3. Let P be a set of vertices that contains one vertex from each connected component of $G[X];$ we have that $|P| = r_\ell \leq \frac{\gamma n}{L}.$

Step 5. Branch into at most $(n + 1)^4$ subbranches fixing $r_\ell, |X|, |Y|, |N'[X]|.$ Then branch into $\binom{n}{r_\ell} \leq \binom{n}{\frac{\gamma n}{L}}$ cases, in each fixing a different set of size r_ℓ as a candidate for $P.$ Add an artificial vertex v_1 adjacent to $P,$ and using Proposition 3 in $\mathcal{O}^*\left(\binom{|N'[X]|}{|X|}\right) \leq \mathcal{O}^*(2^{|N'[X]|})$ time enumerate at most $\binom{|N'[X]|}{|X|} \leq 2^{|N'[X]|}$ vertex sets that (i) are connected, (ii) contain $P \cup \{v_1\},$ (iii) are of size $|X| + 1$ and have neighbourhood of size $|N'(X)|.$ Note that we can do it by filtering out sets that do not contain P from the list given by Proposition 3. As $X \cup \{v\}$ is among enumerated candidates, branch into at most $2^{|N'[X]|}$ subcases, in each fixing a different candidate for $X.$

Let $R = G[V \setminus (N'[X] \cup S)].$ Note that we need to have $|V(R)| \geq |Y| \geq r_s \geq \frac{\gamma n}{2},$ so if $|V(R)| < \frac{\gamma n}{2}$ then we may safely terminate the branch. We will now use the fact that the input graph does not contain any forbidden induced subgraphs of size bounded by some bound $\ell;$ recall that this assumption was justified in the beginning of Sect. 4 by an application of Proposition 2. We set $\ell = 3C^2 + 1;$ hence, whenever we examine an induced subgraph of G of size at most $\ell,$ we know that it belongs to $\Pi.$ The later steps of the algorithm are encapsulated in the following lemma.

Lemma 2 Assuming $\alpha < \frac{\gamma}{104C^3}$ and $\ell = 3C^2 + 1,$ there exists a universal constant $\rho < 1$ and an algorithm working in $\mathcal{O}^*(2^{\rho|V(R)|})$ time that enumerates at most $\mathcal{O}(2^{\rho|V(R)|})$ candidate subsets of $V(R),$ such that Y is among the enumerated candidates.

Before we proceed to the proof, let us observe that application of Lemma 2 finishes the whole algorithm. Indeed, so far in the branching procedure we have an overhead of $\mathcal{O}^*\left(\binom{n}{\alpha n} \cdot \binom{n}{\frac{\gamma n}{L}} \cdot 2^{|N'[X]|}\right)$ for guessing S and $X.$ If we now enumerate and examine—by testing whether $G[X \cup S \cup Y] \in \Pi$ —all the candidates for Y given by Lemma 2, we arrive at running time

$$\mathcal{O}^*\left(\binom{n}{\alpha n} \cdot \binom{n}{\frac{\gamma n}{L}} \cdot 2^{|N'[X]|} \cdot 2^{\rho|V(R)|}\right).$$

Since $|N'[X]| + |V(R)| \leq n,$ $\rho < 1$ is a universal constant and $|V(R)| \geq \frac{\gamma n}{2},$ given $\gamma > 0$ we may choose L to be large enough and $\alpha > 0$ to be small enough

(and smaller than $\frac{\gamma}{104C^3}$) so that this running time is $\mathcal{O}^*(2^{\kappa_9 n})$ for some $\kappa_9 < 1$. Here we exploit the fact that ρ does not depend on α , γ or L . Intuitively, what is really happening at this point is that the threshold C for large components depends on γ and L , and thus the threshold ℓ for forbidden induced subgraphs on which we branch a priori using Proposition 2 depends on γ and L . This branching, however, is performed outside the current reasoning and we avoid a loop in the definitions of thresholds.

We proceed to the proof of Lemma 2.

Proof (of Lemma 2) The initial step is a classical branching algorithm whose goal is to reduce the degrees in R . The algorithm will produce a number of branches: pairs (A, D) , where A is the set of vertices assumed to be contained in the solution, and D is the set of vertices assumed to be excluded from it. During branching process we will be adding vertices to sets A and D , the algorithm starts with $A = D = \emptyset$. We say that a vertex $v \in V(R)$ is *heavy* if $\deg(v)_{R \setminus D} \geq 3C$, and is *light* otherwise. Our goal is to get rid of all the heavy vertices, that is, to achieve a situation where all the vertices in $R \setminus D$ are light (where the degrees are counted in $R \setminus D$). The following claim formally explains all the demanded properties.

Claim 1 There exists a universal constant $\sigma < 1$ and an algorithm running in time $\mathcal{O}^*(2^{\sigma|V(R)|})$, which outputs a set of pairs $\mathcal{L} = \{(A_1, D_1), (A_2, D_2), \dots, (A_p, D_p)\}$ of disjoint subsets of $V(R)$ with the following conditions satisfied:

- For every pair (A_i, D_i) , all vertices of $R \setminus D_i$ are light in $R \setminus D_i$.
- There is an index i_0 such that $A_{i_0} \subseteq Y$ and $D_{i_0} \cap Y = \emptyset$.
- $\sum_{i=1}^p \phi(A_i, D_i) \leq 2^{\sigma|V(R)|}$, where ϕ is a potential function defined as $\phi(A, D) = 2^{\sigma|V(R) \setminus (A \cup D)|}$.

Proof (of Claim 1) The algorithm maintains two disjoint sets A and D ; A is the set of vertices assumed to be contained in the constructed candidate set, while the vertices of D are assumed to be excluded from the constructed candidate set. Naturally, we begin with $A = D = \emptyset$. The algorithm stops branching when it finds out that $R \setminus D$ contains only light vertices. Thus, the output of the branching algorithm is a set of leaf branches (A_i, D_i) where $R \setminus D_i$ only contains light vertices. During branching we ensure the property that there is at least one output branch (A_i, D_i) such that $A_i \subseteq Y$ and $D_i \cap Y = \emptyset$; to express this property, we will also say that the branching is *correct*.

The progress of the algorithm is measured by the potential function $\phi(A, D) = 2^{\sigma|V(R) \setminus (A \cup D)|}$ for some universal constant $\sigma < 1$ to be determined later. At each branching step we will ensure that the sum of potentials in subbranches is at most the potential of the initial branch. As the potential is always at least 1, we will produce at most $2^{\sigma|V(R)|}$ leaf branches (A_i, D_i) in total, and their total sum of potentials will be at most $2^{\sigma|V(R)|}$. Each branching step will be performed in polynomial time, so the whole branching algorithm runs in $\mathcal{O}^*(2^{\sigma|V(R)|})$ time.

If $R \setminus D$ does not contain any heavy vertex, we terminate the branching procedure and output the current pair (A, D) . Otherwise, the graph $R \setminus D$ contains some heavy vertex v . At first consider the case $v \notin A$. As $R[Y]$ has all the connected components of size at most C , we infer that if $v \in Y$, then at most a third of the neighbours of v in $R \setminus D$ can belong to Y . Hence, we can afford the following branching step. We branch

into a number of subcases. In one subcase, v is assigned to D . In the other subcases, v is assigned to A and alignment of all the vertices of $N_{R \setminus D}(v) \setminus A$ is guessed in such a manner that v has less than C neighbours in A . As the neighbours of v in $R \setminus D$ can be only in A , or neither in A nor in D , at most a third of neighbours contained in $N_{R \setminus D}(v) \setminus A$ can go in this manner to A . Note here that if $N_{R \setminus D}(v) \setminus A$ is empty, this means that v has already at least C neighbours in A and we may safely terminate the branch. The correctness of the presented branching rules follows directly from the fact that all the connected components of $R[Y]$ are of size at most C .

From Proposition 4 follows combinatorial bound which will be useful when controlling the behaviour of the potential:

Fact 1 If $|M| = m$, then the number of subsets of M of size at most $m/3$ is bounded by $2^{\sigma' \cdot m}$ for some universal constant $\sigma' < 1$.

In fact we can choose $2^{\sigma'} = 1.89$. By Fact 1, in order to prove that the total potential of resulting instances is at most the initial potential, it suffices to check that for $n = |V(R) \setminus (A \cup D)|$ and $m = |N_{R \setminus D}(v) \setminus A| \geq 1$ it holds that

$$2^{\sigma n} \geq 2^{\sigma(n-1)} + 2^{\sigma' \cdot m} \cdot 2^{\sigma(n-1-m)}.$$

This is however equivalent to

$$2^\sigma \geq 1 + 2^{m(\sigma' - \sigma)}.$$

Let us choose $1 > \sigma > \sigma'$ so that $2^\sigma(2^\sigma - 1) \geq 2^{\sigma'}$. Observe that this can be done since function $f(t) = 2^t(2^t - 1)$ is continuous and strictly increasing in the neighbourhood of 1, and $f(1) = 2 > 2^{\sigma'}$. Then

$$2^\sigma \geq 1 + 2^{\sigma' - \sigma} \geq 1 + 2^{m(\sigma' - \sigma)},$$

since $m \geq 1$ and $\sigma' < \sigma$.

In the remaining case when $v \in A$, we simply omit the branch when v is assigned to D . Hence, to bound the total potential of obtained subbranches, we need to check that

$$2^{\sigma n} \geq 2^{\sigma' \cdot m} \cdot 2^{\sigma \cdot (n-m)},$$

which follows from the fact that $\sigma' < \sigma$. This completes the proof of Claim 1. □

We proceed with the proof of Lemma 2. Let

$$\mathcal{L} = \{(A_1, D_1), (A_2, D_2), \dots, (A_p, D_p)\}$$

be the set of pairs produced by Claim 1. We know that (i) $\sum_{i=1}^p \phi(A_i, D_i) \leq 2^{\sigma|V(R)|}$, (ii) for every i all the vertices in $R \setminus D_i$ are light, and (iii) there exists an index i_0 such that $A_{i_0} \subseteq Y$ and $D_{i_0} \cap Y = \emptyset$. Let $\mathcal{L}_{\text{small}}$ be the subset of \mathcal{L} consisting of pairs

(A_i, D_i) such that $|A_i| + |D_i| \geq \frac{|V(R)|}{4}$, and $\mathcal{L}_{\text{large}}$ be the subset of remaining instances from \mathcal{L} (here *small/large* in $\mathcal{L}_{\text{small}}, \mathcal{L}_{\text{large}}$ indicate that there is small or large number of remaining vertices for which we have to decide). Now, for every pair $(A, D) \in \mathcal{L}$ we produce a number of candidates for Y . We handle lists $\mathcal{L}_{\text{small}}$ and $\mathcal{L}_{\text{large}}$ differently.

For every pair $(A, D) \in \mathcal{L}_{\text{small}}$ we proceed by brute force. As the final candidates for Y , we output all the sets of form $A \cup Y'$, where Y' is a subset of $V(R) \setminus (A \cup D)$. Clearly, if $(A_{i_0}, D_{i_0}) \in \mathcal{L}_{\text{small}}$, then Y is among the output candidates. We now estimate how many candidates have been output.

For $(A, D) \in \mathcal{L}_{\text{small}}$, let $m = |V(R) \setminus (A \cup D)|$. Thus, for (A, D) we produce exactly 2^m candidates. Since $m \leq \frac{3}{4}|V(R)|$, we have that $2^m = 2^{\sigma m} \cdot 2^{(1-\sigma)m} \leq \phi(A, D) \cdot 2^{\frac{3}{4}|V(R)|(1-\sigma)}$. Hence, the total number of candidates produced for $\mathcal{L}_{\text{small}}$ is at most

$$\sum_{(A, D) \in \mathcal{L}_{\text{small}}} \phi(A, D) \cdot 2^{\frac{3}{4}|V(R)|(1-\sigma)} = 2^{\sigma|V(R)| + \frac{3}{4}|V(R)|(1-\sigma)} = 2^{\frac{3+\sigma}{4}|V(R)|}.$$

Note that $\frac{3+\sigma}{4} < 1$ for $\sigma < 1$.

We finally proceed to the pairs from $\mathcal{L}_{\text{large}}$. Let $(A, D) \in \mathcal{L}_{\text{large}}$, and let $Q = V(R) \setminus (A \cup D)$. The following claim is the crucial step in our reasoning:

Claim 2 If $(A, D) \in \mathcal{L}_{\text{large}}$ is such that $A \subseteq Y$ and $D \cap Y = \emptyset$, then we have that $|Q \cap Y| \geq \frac{2}{3}|Q|$.

In other words, we may safely assume that in the correct branch at least two thirds of the unresolved vertices must remain in the solution. Before we proceed to the proof of Claim 2, we present how it will be used to finish the whole algorithm of Lemma 2.

For every pair $(A, D) \in \mathcal{L}_{\text{large}}$ we again proceed by brute force, but we take Claim 2 into consideration as well. That is, we output as candidates all sets of form $A_i \cup Y'$, where Y' is a subset of Q of size at least $\frac{2}{3}|Q|$. By applying Fact 1 to the complement of Y' , we infer that the number of produced choices is at most $2^{\sigma' m} \leq 2^{\sigma m} = \phi(A, D)$, where again $m = |Q|$. Thus, the total number of candidates produced in this manner is at most

$$\sum_{(A, D) \in \mathcal{L}_{\text{large}}} \phi(A, D) \leq 2^{\sigma|V(R)|} \leq 2^{\frac{3+\sigma}{4}|V(R)|}.$$

Concluding, the algorithm will produce at most $2 \cdot 2^{\frac{3+\sigma}{4}|V(R)|}$ candidates for Y : $2^{\frac{3+\sigma}{4}|V(R)|}$ for $\mathcal{L}_{\text{small}}$ and $2^{\frac{3+\sigma}{4}|V(R)|}$ for $\mathcal{L}_{\text{large}}$. Hence we can take $\rho = \frac{3+\sigma}{4}$. Claim 2 ensures that Y will be among the candidates enumerated for $\mathcal{L}_{\text{large}}$ providing that $(A_{i_0}, D_{i_0}) \in \mathcal{L}_{\text{large}}$, while we have already argued that Y will be among the candidates enumerated for $\mathcal{L}_{\text{small}}$ providing that $(A_{i_0}, D_{i_0}) \in \mathcal{L}_{\text{small}}$.

We now proceed to the proof of Claim 2.

Proof (of Claim 2) Assume for the sake of contradiction that $|Q \cap Y| < \frac{2}{3}|Q|$. Then, since $|Q| \geq \frac{3}{4}|V(R)|$, we have that

$$|Q \setminus Y| > \frac{1}{3}|Q| \geq \frac{1}{4}|V(R)| \geq \frac{\gamma n}{8}.$$

We construct a set $T \subseteq Q \setminus Y$ with the following properties:

- T is independent in G ,
- no two vertices of T are adjacent to the same connected component of $R[Y]$,
- $|T| \geq \frac{\gamma n}{104C^3}$.

The construction of T is performed greedily. We iteratively pick to T an unused vertex v of $Q \setminus Y$ and mark the following vertices of $Q \setminus Y$ as used: (i) v itself, (ii) all the neighbours of v in $Q \setminus Y$, and (iii) all the vertices of $Q \setminus Y$ that are adjacent to any component of $R[Y]$ adjacent to v . Recall that the degrees in $R \setminus D$ are bounded by $3C$ and $Q \subseteq V(R) \setminus D$, so v can have at most $3C$ neighbours in $Q \setminus Y$. For the same reason, v can be adjacent only to at most $3C$ connected components of $R[Y]$. Each of these components is of size at most C , and each vertex contained in any such component can be adjacent to only $3C$ vertices of $Q \setminus Y$. In total, the number of vertices marked as used, including v itself, is at most $1 + 3C + 3C \cdot C \cdot 3C \leq 13C^3$. Hence, we can always find an unused vertex for at least $\frac{|Q \setminus Y|}{13C^3} \geq \frac{\gamma n}{104C^3}$ rounds. From the construction it trivially follows that the constructed T has the two remaining requested properties.

We now claim that $R[T \cup Y] \in \Pi$. Indeed, from the fact that the vertices of T have degree at most $3C$ in $R \setminus D$ we infer that the connected components of $R[T \cup Y]$ need to be of size at most $1 + 3C^2$. If $R[T \cup Y] \notin \Pi$, then there would be a forbidden induced subgraph from \mathcal{F}_Π in $R[T \cup Y]$. As all graphs in \mathcal{F}_Π are connected, this subgraph would need to be contained in one of the connected components of $R[T \cup Y]$, and hence would be of size at most $1 + 3C^2$. However, we assumed that G does not contain any graph from \mathcal{F}_Π of size at most $\ell = 1 + 3C^2$, a contradiction. Hence $R[T \cup Y] \in \Pi$.

It remains to show that the constructed T contradicts that H is maximum induced subgraph from Π . Consider the induced subgraph H' constructed as $H' = G[X \cup Y \cup T]$. In other words, we remove the clique S from the solution H , and insert the set T instead. Clearly, H' is a disjoint union of graphs $G[X]$ and $G[Y \cup T]$; as both of these graphs belong to Π , so does H' . Moreover, as $|S| = \alpha n < \frac{\gamma n}{104C^3} \leq |T|$, we have that $|V(H')| > |V(H)|$. This is a contradiction with optimality of H . □

As discussed before, Claim 2 finishes the proof of Lemma 2. □

5 Summary of the order of choice of constants

In this section we give a short summary of the order of choice of constants. The description of the constants is summarized in Table 2. In the following, by running time *faster than 2^n* we mean running time of form $\mathcal{O}^*(2^{\kappa n})$ for some $\kappa < 1$.

Table 2 Short description of important constants

Constant name	Short description	Defined in
α	Largest clique-size in input graph G	Step 1
β	Checking whether solution size is in $[\frac{n}{2} - \beta n, \frac{n}{2} + \beta n]$	Step 2
γ	Number of connected components in $H \setminus S$	Step 4
δ	$ N'(X_1) - X_1 \geq \delta n$	Case B.1.1
ε	$ N'(X_1) \cap N'(X_2) \geq \varepsilon n$	Case B.1.2
ζ	$2\alpha + 2\beta + 2\delta + \varepsilon$, upper bound on the size of U_{none}	Case B.1.3
L	Large constant > 2	Branch B.2
C	$C = \frac{L}{\gamma}$, size of small components in $H \setminus S$	Branch B.2
ℓ	The input graph does not contain any forbidden induced Subgraph from \mathcal{F}_Π of size at most ℓ	Before step 1
ℓ	Set to $3C^2 + 1$	End of algorithm

We first examine Case B.1.3. In this case, the running time is $\mathcal{O}^*(\binom{n}{\varepsilon n} \cdot \binom{n}{\zeta n}) \cdot \mathcal{O}^*(2^{\kappa_7 n})$ for some universal constant $\kappa_7 < 1$ such that $\binom{n}{\frac{n}{8}} = \mathcal{O}^*(2^{\kappa_7 n})$. Hence, we can find a positive upper bound $\epsilon_0 > 0$ on $\alpha, \beta, \delta, \varepsilon$, such that choosing these constants smaller than ϵ_0 results in Case B.1.3 running faster than 2^n .

We now proceed with Case B.1.2. We first fix any $\varepsilon > 0$ such that $\varepsilon < \epsilon_0$. As observed in this case, given $\varepsilon > 0$ we can find a positive upper bound $\epsilon_1 < \epsilon_0$ on $\alpha, \beta, \gamma, \delta$, such that for any choice of $\alpha, \beta, \gamma, \delta$ smaller than ϵ_1 we obtain running time faster than 2^n .

We proceed similarly with Case B.1.1. We fix any $\delta > 0$ such that $\delta < \epsilon_1$. Again, as observed in this case, given $\delta > 0$ we can find a positive upper bound $\epsilon_2 < \epsilon_1$ such that choosing α, γ to be smaller than ϵ_2 results in running time faster than 2^n .

Now we examine Branch B.2. Let us fix any $\gamma > 0$ such that $\gamma < \epsilon_2$. Recall that the running time in this branch was $\mathcal{O}^*(\binom{n}{\alpha n} \cdot \binom{n}{\frac{\gamma n}{L}} \cdot 2^{|N'[X]|} \cdot 2^{\rho|V(R)|})$, where $|N'[X]| + |V(R)| \leq n, |V(R)| \geq \frac{\gamma n}{2}$ and $\rho < 1$ is a universal constant. Hence, given $\gamma > 0$ we can find a positive lower bound $L_0 \geq 2$ on L and a positive upper bound $\epsilon_3 < \epsilon_2$ on α , such that taking any $L > L_0$ and positive $\alpha < \epsilon_3$ gives us running time faster than 2^n . We fix any $L > L_0$, and by lowering ϵ_3 if necessary we ensure that inequality $\epsilon_3 < \frac{\gamma}{104C^3}$ holds, where $C = L/\gamma$. Then we can fix the remaining two constants: we fix β to be any positive constant smaller than ϵ_1 so that Step 2 runs faster than 2^n , and α to be any positive constant smaller than ϵ_3 . Thus we make sure that in Branches B.1 and B.2 we obtain running time faster than 2^n .

Since Case A works faster than 2^n for any $\alpha > 0$, namely in $\mathcal{O}^*(2^{(1-(1-\kappa_0)\alpha)n})$ time for some $\kappa_0 < 1$ depending only on \aleph , we infer that the whole algorithm runs faster than 2^n . Note however, that we assumed that the algorithm runs on \mathcal{F}'_Π -free graphs, where \mathcal{F}'_Π consists of graphs of \mathcal{F}_Π of size at most ℓ , for $\ell = 3C^2 + 1 = 3\frac{L^2}{\gamma^2} + 1$. Since \mathcal{F}'_Π is a finite family of graphs, we can apply Proposition 2 as described in Sect. 4 before Step 1, and obtain running time faster than 2^n for the general problem.

Note now that ℓ is set to be quadratic in terms of C , which in turn depends on L and γ . Since L depends on γ , which in turn depends on δ and ε , even very rough estimates show that ℓ will be at least in the order of several thousands, or even far worse. Therefore, at the very beginning of the algorithm we branch on forbidden induced subgraphs of sizes up to at least several thousands; in this branching we omit only one in 2^ℓ branches, where no vertex of an obstacle is selected to the solution. This leads to the conclusion that if we spell out κ as $0.9999\dots 99$, then the number of nines will be at least two to the power of several thousands—much more than the number of atoms in the universe, not to mention the length of this paper. For this reason, we believe that the precise estimates of the obtained running time are pointless, especially given that the purpose of this paper is only to show that 2^n is not an unbreakable barrier for the MAXIMUM INDUCED Π -SUBGRAPH problem for chordal-like graph classes.

6 Conclusion

Theorem 1 shows that for any class of graphs Π satisfying Properties (1)–(4), a maximum induced subgraph from Π of a graph with n vertices can be found in time $O^*(2^{\lambda n})$ for some $\lambda < 1$. Pipelining Proposition 2 with Theorem 1 shows that we moreover may add any finite family of forbidden subgraphs on top of belonging to Π . More precisely, we have the following theorem.

Theorem 2 *Let \mathcal{F} be a finite set of graphs and Π be a class of graphs satisfying Properties (1)–(4). There exists an algorithm which for a given n -vertex graph G , finds a maximum induced \mathcal{F} -free Π -graph in G in time $O^*(2^{\lambda n})$ for some $\lambda < 1$, where λ depends only on \mathfrak{K} and \mathcal{F} .*

As mentioned in the introduction, Theorem 2 covers such graph classes as proper interval graphs (claw-free interval graphs), Ptolemaic graphs (chordal and gem-free), block graphs (chordal and diamond-free), see Fig. 5. We refer to [3] for the definitions and discussions on these graphs.

In this manner, we hope to provide a new insight into Hypothesis 1 by considering chordal-like graph classes. So far the research on breaking the 2^n barrier for the MAXIMUM INDUCED Π -SUBGRAPH problem concentrated mostly on exploiting sparsity of a graph class, like in [7, 9, 20], or thinness in terms of treewidth, like in the metaresult of Fomin et al. [10]. In this work we were dealing with graph classes which inherently allow existence of large cliques, and thus a new set of tools was needed. Shortly speaking, the crux of our approach is to use existence of balanced clique separators in chordal graphs to apply the 2-TABLE trick of Schroepel and Shamir [24]. However, this application needed to be preceded by a long and technical preparation of the instance at hand.

Clearly, the most important research direction stemming from our work is further investigation of Hypothesis 1. Since we believe that the fully general statement might turn out to be either false or very hard to prove, we propose some relaxations that can be more approachable.

Firstly, following the approach of Fomin et al. [10] one could require the graph class Π to be moreover definable in some logical formalism, for example in CMSO₂ (Counting Monadic Second-Order logic, that is, Monadic Second-Order logic with modular predicates and quantification over edge subsets) or CMSO₁ (the same as CMSO₂, but without quantification over edge subsets). It can be easily seen that both chordal and interval graphs are definable in CMSO₁ by testing existence of any of the forbidden induced subgraphs. We have two concrete examples of hereditary, polynomial-time recognizable, and CMSO₁-definable graph classes for which we do not know any algorithm faster than 2^n :

- *Perfect graphs* can be defined as graphs which contain neither an odd hole, nor an odd anti-hole; this result is known as the Strong Perfect Graph Theorem [5]. Perfect graphs are hereditary, polynomial-time recognizable [4], and containing an odd hole or an odd anti-hole can be easily expressed in CMSO₁.
- *Strongly chordal graphs* are chordal graphs that moreover exclude ℓ -suns (graphs with 2ℓ vertices partitioned in two sets $W = \{w_1, w_2, \dots, w_\ell\}$ and $U = \{u_1, u_2, \dots, u_\ell\}$ such that U is a clique and W is independent set, besides w_i is connected to u_j if and only if $i = j$ or $i \equiv j + 1 \pmod n$, see Fig. 5) for $\ell \geq 3$ as induced subgraphs; we refer to [3] for a broader discussion of this graph class. They are also hereditary, polynomial-time recognizable [3], and definable in CMSO₁. The reason why they do not fall under the regime of Theorem 2 is that ℓ -suns contain arbitrary large cliques and thus Property (2) is not satisfied.

Secondly, one could impose some structural properties on the set of forbidden induced subgraphs \mathcal{F}_Π . One obvious relaxation, already used in Property (2), is requiring that all the graphs from \mathcal{F}_Π are connected, or equivalently that Π is closed under

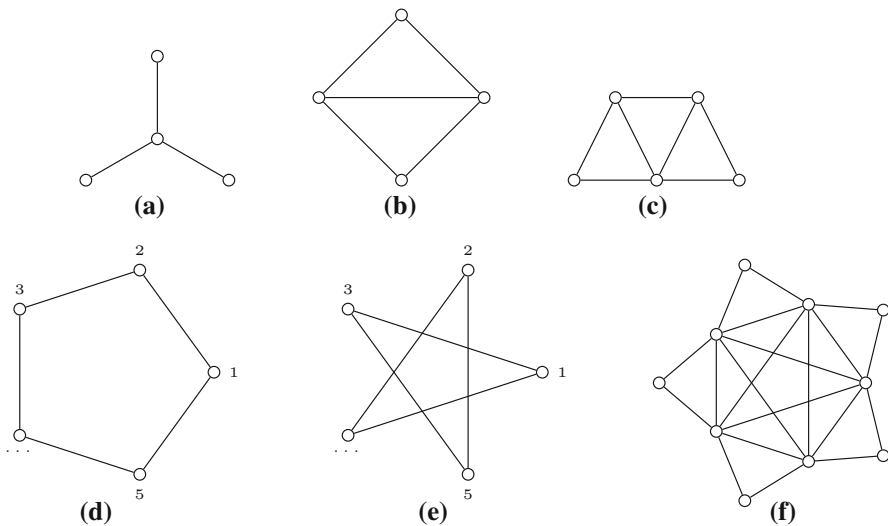


Fig. 5 Forbidden subgraphs used in characterizations of graph classes discussed in Sect. 6. **a** claw, **b** diamond, **c** gem, **d** hole (cycle), **e** anti-hole, **f** sun

taking disjoint union. More restrictions on the graphs from \mathcal{F}_Π can be further imposed. For instance, it would be interesting to see if requiring that all the graphs from \mathcal{F}_Π have treewidth bounded by some constant could help in breaking the 2^n barrier; note that this subsumes both the case of chordal and of interval graphs.

Finally, one could deviate from the precise statement of Hypothesis 1 and replace the condition of being hereditary with connectivity. That is, we would like to find a maximum induced *connected* graph belonging to Π . Of course, our approach fails since the connectivity requirements are not hereditary, and thus Property (1) is not satisfied. Say, can a maximum induced *connected* chordal subgraph be found faster than 2^n ?

Acknowledgments We thank the reviewers for careful reading of the manuscript and many helpful comments.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Bliznets, I., Fomin, F.V., Pilipczuk, M., Villanger, Y.: Largest chordal and interval subgraphs faster than 2^n . In: Proceedings of the 21st Annual European Symposium on Algorithms, ESA 2013, Lecture Notes in Computer Science, vol. 8125, pp. 193–204. Springer (2013)
2. Booth, K.S., Lueker, G.S.: Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *J. Comput. Syst. Sci.* **13**(3), 335–379 (1976)
3. Brandstädt, A., Le, V.B., Spinrad, J.P.: Graph Classes: A Survey. Society for Industrial and Applied Mathematics, Philadelphia (1999)
4. Chudnovsky, M., Cornuéjols, G., Liu, X., Seymour, P.D., Vuskovic, K.: Recognizing berge graphs. *Combinatorica* **25**(2), 143–186 (2005)
5. Chudnovsky, M., Robertson, N., Seymour, P., Thomas, R.: The strong perfect graph theorem. *Ann. Math.* **164**(1), 51–229 (2006)
6. Fomin, F.V., Gaspers, S., Kratsch, D., Liedloff, M., Saurabh, S.: Iterative compression and exact algorithms. *Theor. Comput. Sci.* **411**(7), 1045–1053 (2010)
7. Fomin, F.V., Gaspers, S., Pyatkin, A.V., Razgon, I.: On the minimum feedback vertex set problem: exact and enumeration algorithms. *Algorithmica* **52**(2), 293–307 (2008)
8. Fomin, F.V., Kratsch, D.: Exact Exponential Algorithms. Springer Science & Business Media, New York (2010)
9. Fomin, F.V., Todinca, I., Villanger, Y.: Exact algorithm for the Maximum Induced Planar Subgraph problem. In: Proceedings of the 19th Annual European Symposium on Algorithms, ESA 2011, Lecture Notes in Computer Science, vol. 6942, pp. 287–298. Springer (2011)
10. Fomin, F.V., Todinca, I., Villanger, Y.: Large induced subgraphs via triangulations and CMSO. *SIAM J. Comput.* **44**(1), 54–87 (2015)
11. Fomin, F.V., Villanger, Y.: Finding induced subgraphs via minimal triangulations. In: Proceedings of the 27th International Symposium on Theoretical Aspects of Computer Science, STACS 2010, LIPIcs, vol. 5, pp. 383–394. Schloss Dagstuhl-Leibniz-Zentrum für Informatik (2010)
12. Fomin, F.V., Villanger, Y.: Treewidth computation and extremal combinatorics. *Combinatorica* **32**(3), 289–308 (2012)
13. Gaspers, S.: Exponential time algorithms: structures, measures, and bounds. Ph.D. thesis, University of Bergen, Norway (2008)
14. Gaspers, S., Kratsch, D., Liedloff, M.: On independent sets and bicliques in graphs. *Algorithmica* **62**(3–4), 637–658 (2012)
15. Gilbert, J.R., Rose, D.J., Edenbrandt, A.: A separator theorem for chordal graphs. *SIAM J. Algebr. Discrete Methods* **5**(3), 306–313 (1984)

16. Golombic, M.C.: Algorithmic Graph Theory and Perfect Graphs. Academic Press, New York (1980)
17. Gupta, S., Raman, V., Saurabh, S.: Maximum r -regular induced subgraph problem: fast exponential algorithms and combinatorial bounds. *SIAM J. Discrete Math.* **26**(4), 1758–1780 (2012)
18. Lekkerkerker, C., Boland, J.: Representation of a finite graph by a set of intervals on the real line. *Fundam. Math.* **51**(1), 45–64 (1962)
19. Lewis, J.M., Yannakakis, M.: The node-deletion problem for hereditary properties is NP-complete. *J. Comput. Syst. Sci.* **20**(2), 219–230 (1980)
20. Pilipczuk, M., Pilipczuk, M.: Finding a maximum induced degenerate subgraph faster than 2^n . In: Proceedings of the 7th International Symposium on Parameterized and Exact Computation, IPEC 2012, Lecture Notes in Computer Science, vol. 7535, pp. 3–12. Springer (2012)
21. Raman, V., Saurabh, S., Sikdar, S.: Efficient exact algorithms through enumerating maximal independent sets and other techniques. *Theory Comput. Syst.* **41**(3), 563–587 (2007)
22. Robson, J.M.: Algorithms for maximum independent sets. *J. Algorithms* **7**(3), 425–440 (1986)
23. Rose, D.J., Tarjan, R.E., Lueker, G.S.: Algorithmic aspects of vertex elimination on graphs. *SIAM J. Comput.* **5**(2), 266–283 (1976)
24. Schroepel, R., Shamir, A.: A $t = O(2^{n/2})$, $s = O(2^{n/4})$ algorithm for certain NP-complete problems. *SIAM J. Comput.* **10**(3), 456–464 (1981)