Largest Chordal and Interval Subgraphs Faster Than 2^n

Ivan Bliznets[1,⋆], Fedor V. Fomin[2,⋆⋆],
Michał Pilipczuk[2,⋆⋆], and Yngve Villanger[2]

[1] St. Petersburg Academic University of the Russian Academy of Sciences, Russia
ivanbliznets@tut.by
[2] Department of Informatics, University of Bergen, Norway
{fomin,michal.pilipczuk,yngvev}@ii.uib.no

Abstract. We prove that in an n-vertex graph, 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 $2^n n^{O(1)}$ bound of the brute-force search for these problems.

1 Introduction

The area of exact exponential algorithms is about solving intractable problems faster than the trivial exhaustive search, though still in exponential time [4]. 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 the Maximum Induced Subgraph with Property Π problem, where for a given graph G and hereditary property Π one asks for a maximum induced subgraph with property Π.

By the result of Lewis and Yannakakis [11], the problem is NP-complete for every non-trivial property Π. Different variants of property Π like being edgeless, planar, outerplanar, bipartite, complete bipartite, acyclic, degree-constrained, chordal etc., were studied in the literature. From the point of view of exact algorithms, as far as property Π can be tested in polynomial time, a trivial brute-force search trying all possible vertex subsets of G solves Maximum Induced Subgraph with Property Π in time $\mathcal{O}^*(2^n)$ on an n-vertex graph G.[1] However, many algorithms for Maximum Induced Subgraph with Property Π which are faster than $\mathcal{O}^*(2^n)$ can be found in the literature for explicit properties Π. Notable examples are Π being the property of being edgeless [14] (equivalent to Maximum Independent Set), acyclic [3] (equivalent to Maximum Induced Forest), regular [9], 2-colorable [13], planar [5], degenerate [12], cluster graph [2], or biclique [7]. A longstanding open question in the area is if

⋆ Supported by The Ministry of education and science of Russian Federation, project 8216 and Russian Foundation for Basic Research (12-01-31057-mol_a).

⋆⋆ Supported by the European Research Council (ERC) via grant Rigorous Theory of Preprocessing, reference 267959.

[1] The $O^*(\cdot)$ notation suppresses terms polynomial in the input size.

H.L. Bodlaender and G.F. Italiano (Eds.): ESA 2013, LNCS 8125, pp. 193–204, 2013.

Maximum Induced Subgraph with Property Π can be solved faster than the trivial $\mathcal{O}^*(2^n)$ for every hereditary property Π testable in polynomial time.

Since every hereditary class of graphs with property Π can be characterized by a (not necessarily finite) set of forbidden induced subgraphs, there is an equivalent formulation of the Maximum Induced Subgraph with Property Π 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 Subgraph with Property Π 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 arguments, 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 [2]. 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 [1] 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 which 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 [8]. 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 graph classes and give an algorithm that for every graph class Π satisfying these properties and for a given n-vertex graph G, 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 the class Π. Because classes of chordal and interval graphs satisfy the required properties, as an immediate corollary of our algorithm we obtain that Maximum Induced Chordal Subgraph and Maximum Induced Interval Subgraph can be solved in time $\mathcal{O}^*(2^{\lambda n})$ for some $\lambda < 1$. The main intention of our work was to break the trivial 2^n barrier and we did not try to optimize the constant in the exponent. There are several places where the running time of our algorithm can be improved by the cost of more involved arguments and intensive case analyses. We tried to keep the description of our algorithm as simple as possible, leaving only the ideas crucial for breaking the barrier, and postponing more complicated improvements till the full version of the paper. Moreover, 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 Maximum Induced Subgraph with Property Π problems, where property Π is to be chordal/interval graph containing no induced subgraph from a *finite* forbidden set of graphs. Examples of such graph classes are proper interval graphs, Ptolemaic graphs, block graphs, and proper circular-arc graphs; see [1] for definitions and discussions of these graph classes.

2 Preliminaries

In the paper we use standard graph notation. A graph class Π is simply a family of graphs. We sometimes use terms Π-graph or Π-subgraph to express membership in Π. An *induced subgraph* of a graph is a subset of vertices, with all the edges between those vertices that are present in the larger graph. 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 minimum forbidden induced subgraphs $\mathcal{F}_\Pi$: graph G is in Π if and only if it does not contain any induced subgraph from $\mathcal{F}_\Pi$, 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 is the class of graphs not containing any induced cycles of length more than three, that is, chordal graphs are $\mathcal{F}$-free graphs, were the set $\mathcal{F}$ consists of all cycles of length more than three. Chordal graphs are hereditary and polynomial-time recognizable. Chordal graphs admit many more characterizations, for example they are exactly graphs admitting a decomposition into a clique tree. A useful corollary of this fact is the following folklore lemma.

Proposition 1 (Folklore). *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 (i) $|X_1|, |X_2| \le \frac{2}{3}|V(H)|$, and (ii) 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. 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. Their characterization in terms of minimal forbidden induced subgraphs was given by Lekkerkerker and Boland [10]. The book of Golumbic [8] provides a thorough introduction to chordal and interval graphs.

We now describe the classical tools needed for the algorithm. The following folklore result basically follows from the observation that branching on forbidden structures of constant size always leads to complexity better than 2^n. Due to space constrains we omit its proof here.

Proposition 2 (Folklore). *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 $\mathcal{O}^*(2^{\epsilon n})$ 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.*

We need also the following proposition from [6].

Proposition 3 ([6]). *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* (i) *$v \in B$,* (ii) *$|B| = b + 1$, and* (iii) *$|N(B)| = f$, is at most $\binom{b+f}{b}$. Moreover, all such subsets can be enumerated in time $\mathcal{O}^*(\binom{b+f}{b})$.*

The last necessary ingredient is the classical idea used by Schroeppel and Shamir [15] 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 $\boldsymbol{s} \in \mathbb{Q}^k$. 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 vectors is $\boldsymbol{s}$. 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 $\boldsymbol{v}$ of T_2 check whether T_1 contains a column equal to $\boldsymbol{s} - \boldsymbol{v}$ in $\mathcal{O}(k \log m_1)$ time using binary search.

Proposition 4. *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 $\mathcal{F}_\Pi$. We need the following properties of $\mathcal{F}_\Pi$:

Property (2). All graphs in $\mathcal{F}_\Pi$ are connected, and all of them do not contain a clique of size $\aleph + 1$ for some universal constant $\aleph$.

For chordal graphs $\mathcal{F}_\Pi$ consists of cycles of length at least 4, hence $\aleph = 2$. For interval graphs, an inspection of the list of forbidden induced subgraphs [10], shows that we may take $\aleph = 4$. In the following, we always treat $\aleph$ as a universal constant on which all the later constants may depend; moreover, $\aleph$ may influence the exponents of polynomial factors hidden in the O^* notation. Let us remark that connectedness of all the forbidden induced subgraphs is equivalent to requiring Π to be closed under taking disjoint union. 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 [1] for more information on this class of graphs.

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 [8]. 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 an 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 $(G[X_2 \cup S], S)$.

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, we take clique path of the graph G and examine the clique separator S such that there is at most half of vertices before it and at most half after 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 P_4 to G and make one end of P_4 to be adjacent to every vertex of S, thus forcing S to be the end bag, and run 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 5. *If Π satisfies Properties (1)-(4), then there exists an algorithm which, given an n-vertex graph G, 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 $\aleph$.*

As we already observed, chordal and interval graphs satisfy Properties (1)-(4). Thus Theorem 5 implies immediately 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 suspects for an optimal solution. The properties we strongly rely on are the balanced separation property of chordal graphs (Property (4)), and conditions on minimal forbidden induced subgraphs for Π (Property (2)).

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. 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.

Firstly, we observe that by Proposition 2, we may assume that the input graph does not contain any forbidden induced subgraph from $\mathcal{F}_\Pi$ 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 $\mathcal{F}'_\Pi$-free graphs, where $\mathcal{F}'_\Pi$ consists of graphs of $\mathcal{F}_\Pi$ 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 $\mathcal{F}_\Pi$ of size at most ℓ.

The algorithm performs a number of *steps*. After each step, depending on the result, the algorithm chooses one of the subcases.

Step 1. *Using the algorithm of Robson [14], in $\mathcal{O}^*(2^{0.276n})$ time find the largest clique K in G.*

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 $\alpha > 0$, $\alpha < 1/48$, 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 claim.

Lemma 6. *Let P be a subset of vertices of an n-vertex graph G 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 $\aleph$.

We observe that for every subset $X \subseteq K$, $G[P \cup X]$ 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 induced 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 $\aleph$.

We now do the following. 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 6 to find a maximum induced chordal 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: *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 $\beta > 0$, $\beta < 1/16$ 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 $\mathcal{O}^*(\binom{n}{\lceil n/2 - \beta n \rceil})$, which is faster than $\mathcal{O}^*(2^n)$.

Step 2. *Iterate through all subsets of vertices of size at most $\lceil n/2 - \beta n \rceil$ or at least $\lfloor n/2 + \beta n \rfloor$, and for each of them check if it induces a graph belonging to Π. If some subset of size at least $\lfloor n/2 + \beta n \rfloor$ induces a Π-graph, output the subgraph induced by any of such subsets of maximum cardinality, and terminate the algorithm. If no subset of size exactly $\lceil n/2 - \beta n \rceil$ induces a Π-graph, output the subgraph induced by the maximum size subset inducing a Π-graph among those of size at most $\lceil n/2 - \beta n \rceil$, and terminate the algorithm.*

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

$|X_1|, |X_2| \geq (\frac{1}{6} - \frac{\beta}{3} - \alpha)n > \frac{1}{8}n$, as $\beta < 1/16$ and $\alpha < 1/48$. As S is also a clique in G, we have that $|S| \leq \alpha n$. Property (4) gives us more algorithmic claims about the partition (X_1, S, X_2) of $V(H)$; these claims 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 cardinalities of X_1, X_2. 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 cardinalities of X_1, X_2 are known. Let $G' = G \setminus S$; to ease the notation, for $X \subseteq V(G')$ let $N'[X] = N_{G'}[X]$ and $N'(X) = N_{G'}(X)$. We now 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 guessing the sizes 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, 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 is symmetric. Due to the space constraints, here we give only a brief outline. As $G[X_1]$ has only at most γn components, we can guess with $\mathcal{O}^*(\binom{n}{\gamma n})$ overhead a set that contains one element from each connected component of $G[X_1]$. Then, using Proposition 3 we can guess the whole set X_1 with $\binom{|X_1|+|N'(X_1)|}{|X_1|}$ overhead. For X_2 we perform a brute-force guess on the remaining part of $V(G')$, i.e., $V(G') \setminus N'[X_1]$, and at the end for each candidate set $X_1 \cup X_2 \cup S$ we test in polynomial time whether it induces a subgraph belonging to Π. As $||N'(X_1)| - |X_1|| \geq \delta n$, we have that $\binom{|X_1|+|N'(X_1)|}{|X_1|} = \mathcal{O}^*(2^{\kappa_2 |N[X_1]|})$ for some $\kappa_2 < 1$ depending only on δ. Since $|N'[X_1]| \geq \frac{1}{8}n$, given δ we can choose α, γ small enough so that the overhead $\mathcal{O}^*(\binom{n}{\alpha n} \cdot \binom{n}{\gamma n})$ is insignificant compared to the gain obtained when guessing X_1. Hence, we produce $\mathcal{O}^*(2^{\kappa_3 n})$ candidates in total, for some $\kappa_3 < 1$.

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

Again, due to the space constraints, we provide only a short description of this case. We perform a similar strategy as in Case B.1.1, but we guess both X_1 and

X_2 using Proposition 3. Observe that having guessed X_1, we can exclude $N'[X_1]$ from consideration when guessing X_2, thus removing at least εn neighbours of X_2. After the removal, the number of neighbours of X_2 differs much from $|X_2|$ (recall that δ is significantly smaller than ε), and we obtain a gain when guessing X_2. This gain depends on ε only, so we can choose α and γ small enough so that overhead $\mathcal{O}^*(\binom{n}{\alpha n} \cdot \binom{n}{\gamma n}^2)$ is insignificant compared to it.

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 $U = U_{\text{both}} \cup U_{\text{none}}$. 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

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

Given that sets U_{both} and U_{none} are small, we may fix them with $\mathcal{O}^*(\binom{n}{\varepsilon n} \cdot \binom{n}{\zeta n})$ overhead in the running time: we branch into $\mathcal{O}^*(\binom{n}{\varepsilon n} \cdot \binom{n}{\zeta n})$ subproblems, in each fixing disjoint subsets of $V \setminus S$ of sizes at most εn and ζn as U_{both}, U_{none}, 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 Π, 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, (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. Create matrices T_1, T_2 by putting the vectors of 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 $\mathcal{O}^*(2^{\kappa_6 n})$ for some universal constant $\kappa_6 < 1$ (recall that $\beta < 1/16$, so $\frac{1}{3}n + \frac{2\beta}{3}n < \frac{3}{8}n$). Hence, by Proposition 4 we may solve the obtained instance of 2-Table in $\mathcal{O}^*(2^{\kappa_6 n})$ time. The total running time used by Case B.1.3, including the overheads for guessing clique S, set U and cardinalities, is $\mathcal{O}^*(\binom{n}{\alpha n} \cdot \binom{n}{\varepsilon n} \cdot \binom{n}{\zeta n} \cdot 2^{\kappa_6 n})$; note that we may choose $\alpha, \beta, \delta, \varepsilon$ small enough so that this running time is $\mathcal{O}^*(2^{\kappa_7 n})$ for some $\kappa_7 < 1$.

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, $\frac{L \cdot r_\ell}{\gamma} \le n$, $r_\ell \le \frac{\gamma n}{L}$ and, consequently, $r_s \ge \gamma n - r_\ell \ge \gamma n(1 - \frac{1}{L}) \ge \frac{\gamma n}{2}$. Since small components are nonempty, they contain at least $\frac{\gamma n}{2}$ vertices in total.

Let us wrap up the situation. 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 component of $G[X]$; we have that $|P| = r_\ell \le \frac{\gamma n}{L}$.

Step 5. *Branch into at most $(n+1)^4$ subbranches fixing $r_\ell, |X|, |Y|, |N'[X]|$. Then branch into $\binom{n}{r_\ell} \le \binom{n}{\frac{\gamma n}{L}}$ 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 $\mathcal{O}^*(\binom{|N'[X]|}{|X|}) \le \mathcal{O}^*(2^{|N'[X]|})$ time enumerate at most $\binom{|N'[X]|}{|X|} \le 2^{|N'[X]|}$ 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 $|V(R)| \ge |Y| \ge r_s \ge \frac{\gamma n}{2}$, 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 ℓ; recall that this assumption was justified 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 ℓ, we know that it belongs to Π. The later steps of the algorithm are encapsulated in the following lemma.

Lemma 7. *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 $O(2^{\rho|V(R)|})$ candidate subsets of $V(R)$, such that Y is among the enumerated candidates.*

The full proof of Lemma 7 is omitted; here, we only sketch the intuition behind the proof. However, before we proceed to this sketch, let us observe that application of Lemma 7 finishes the whole algorithm. Indeed, so far in the branching procedure we have an overhead of $\mathcal{O}^*(\binom{n}{\alpha n} \cdot \binom{n}{\frac{\gamma n}{L}} \cdot 2^{|N'[X]|})$ 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 7, we arrive at running time $\mathcal{O}^*(\binom{n}{\alpha n} \cdot \binom{n}{\frac{\gamma n}{L}} \cdot 2^{|N'[X]|} \cdot 2^{\rho|V(R)|})$.

As $|N'[X]| + |V(R)| \le n$, $\rho < 1$ is a universal constant and $|V(R)| \ge \frac{\gamma n}{2}$, 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 $\mathcal{O}^*(2^{\kappa_8 n})$ for some $\kappa_8 < 1$. Here we exploit the fact that ρ does not depend on α, γ or L. What is really happening 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. Yet this branching is done outside the current reasoning and we avoid a loop in the definition of thresholds.

We now sketch the proof of Lemma 7. Firstly, as $G[Y]$ have connected components of size at most C, the degrees in $G[Y]$ are bounded by $C-1$. Hence, whenever we see a vertex v that has high degree in R, say at least $3C$, then we can infer that if it is in Y, then at most a third of its neighbours can be also in Y. This allows us to design a branching procedure with running time $2^{O(\sigma|V(R)|)}$ for some universal $\sigma < 1$ that gets rid of high-degree vertices in R. For simplicity, assume from now on that the degrees in R are bounded by $3C$.

The crucial observation now is that Y must in fact constitute almost the whole $V(R)$, hence we can guess Y in a much more efficient manner than via a $2^{|V(R)|}$ brute-force. For the sake of contradiction, assume that $V(R) \setminus Y$ constitutes a constant fraction of $V(R)$. The strategy is to show that the assumed maximum solution H is in fact not maximum, using the fact that α is very small compared to $|V(R)|$. Let us construct an alternative solution H' as follows: we remove S from H, thus losing at most αn vertices, and add vertices of $V(R) \setminus Y$ in a greedy manner so that no component larger than $3C^2 + 1$ is created in $V(R)$. As G does not contain any forbidden induced subgraph for Π of size at most $3C^2 + 1$, all these components belong to Π. As Π is closed under taking disjoint union, H' constructed in this manner also belongs to Π. The bound on the degrees in R ensures that the greedy procedure adds at least $\frac{|V(R) \setminus Y|}{O(C^3)}$ vertices; hence if we choose $\alpha < \frac{\gamma}{O(C^3)}$, then H' is larger than H, contradicting the maximality of H.

5 Conclusion

Theorem 5 shows that for any class of graphs Π satisfying Properties (1)–(4), a maximum induced subgraph from Π of an n-vertex graph can be found in time $\mathcal{O}^*(2^{\lambda n})$ for some $\lambda < 1$. Pipelining Proposition 2 with Theorem 5 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 8. *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 $\mathcal{O}^*(2^{\lambda n})$ for some $\lambda < 1$, where λ depends only on $\aleph$ and $\mathcal{F}$.*

As mentioned in introduction, Theorem 8 covers such graph classes as proper interval graphs, i.e. claw-free interval graphs, Ptolemaic graphs, which are chordal and gem-free, block graphs, which are chordal and diamond-free; proper circular-arc graphs which are chordal, claw-free, and $\bar{S}_3$-free. We refer to [1] for the definitions and discussions on these graphs.

We conclude with the following open questions. An interesting subclass of chordal graphs that cannot be handled by our approach is the class of strongly chordal graphs. The reason is that Property (2) does not hold here and we are not aware of any algorithm for finding a maximum induced strongly chordal subgraph faster than the trivial brute-force. Secondly, our approach fails when we require the induced subgraph to be additionally *connected*, since connectivity requirements are not hereditary, and thus is Property (1) is not satisfied. Say, can a maximum induced *connected* chordal subgraph be found faster than 2^n?

References

1. Brandstädt, A., Le, V., Spinrad, J.P.: Graph Classes. A Survey, SIAM Mon. on Discrete Mathematics and Applications. SIAM, Philadelphia (1999)
2. Fomin, F.V., Gaspers, S., Kratsch, D., Liedloff, M., Saurabh, S.: Iterative compression and exact algorithms. Theor. Comput. Sci. 411, 1045–1053 (2010)
3. Fomin, F.V., Gaspers, S., Pyatkin, A.V., Razgon, I.: On the minimum feedback vertex set problem: Exact and enumeration algorithms. Algorithmica 52, 293–307 (2008)
4. Fomin, F.V., Kratsch, D.: Exact Exponential Algorithms. Springer (2010)
5. Fomin, F.V., Todinca, I., Villanger, Y.: Exact algorithm for the maximum induced planar subgraph problem. In: Demetrescu, C., Halldórsson, M.M. (eds.) ESA 2011. LNCS, vol. 6942, pp. 287–298. Springer, Heidelberg (2011)
6. Fomin, F.V., Villanger, Y.: Treewidth computation and extremal combinatorics. Combinatorica 32, 289–308 (2012)
7. Gaspers, S., Kratsch, D., Liedloff, M.: On independent sets and bicliques in graphs. Algorithmica 62, 637–658 (2012)
8. Golumbic, M.C.: Algorithmic Graph Theory and Perfect Graphs. Academic Press, New York (1980)
9. Gupta, S., Raman, V., Saurabh, S.: Maximum r-regular induced subgraph problem: Fast exponential algorithms and combinatorial bounds. SIAM J. Discrete Math. 26, 1758–1780 (2012)
10. Lekkerkerker, C.G., Boland, J.C.: Representation of a finite graph by a set of intervals on the real line. Fund. Math. 51, 45–64 (1962)
11. Lewis, J.M., Yannakakis, M.: The node-deletion problem for hereditary properties is NP-complete. J. Comput. Syst. Sci. 20, 219–230 (1980)
12. Pilipczuk, M., Pilipczuk, M.: Finding a maximum induced degenerate subgraph faster than 2^n. In: Thilikos, D.M., Woeginger, G.J. (eds.) IPEC 2012. LNCS, vol. 7535, pp. 3–12. Springer, Heidelberg (2012)
13. Raman, V., Saurabh, S., Sikdar, S.: Efficient exact algorithms through enumerating maximal independent sets and other techniques. Theory Comput. Syst. 41, 563–587 (2007)
14. Robson, J.M.: Algorithms for maximum independent sets. J. Algorithms 7, 425–440 (1986)
15. Schroeppel, R., Shamir, A.: A $T = O(2^{n/2})$, $S = O(2^{n/4})$ algorithm for certain NP-complete problems. SIAM J. Comput. 10, 456–464 (1981)