Hadwiger Number of Graphs with Small Chordality

Petr A. Golovach[1(✉)], Pinar Heggernes[1], Pim van 't Hof[1], and Christophe Paul[2]

[1] Department of Informatics, University of Bergen, Bergen, Norway
{petr.golovach,pinar.heggernes,pim.vanthof}@ii.uib.no
[2] CNRS, LIRMM, Montpellier, France
paul@lirmm.fr

Abstract. The Hadwiger number of a graph G is the largest integer h such that G has the complete graph K_h as a minor. We show that the problem of determining the Hadwiger number of a graph is NP-hard on co-bipartite graphs, but can be solved in polynomial time on cographs and on bipartite permutation graphs. We also consider a natural generalization of this problem that asks for the largest integer h such that G has a minor with h vertices and diameter at most s. We show that this problem can be solved in polynomial time on AT-free graphs when $s \geq 2$, but is NP-hard on chordal graphs for every fixed $s \geq 2$.

1 Introduction

The Hadwiger number of a graph G, denoted by $h(G)$, is the largest integer h such that the complete graph K_h is a minor of G. The Hadwiger number has been the subject of intensive study, not in the least due to a famous conjecture by Hugo Hadwiger from 1943 [8] stating that the Hadwiger number of any graph is greater than or equal to its chromatic number. In a 1980 paper, Bollobás et al. [2] called Hadwiger's conjecture "one of the deepest unsolved problems in graph theory." Despite many partial results the conjecture remains wide open more than 70 years after it first appeared in the literature.

Given the vast amount of graph-theoretic results involving the Hadwiger number, it is natural to study the computational complexity of the Hadwiger Number problem, which is to decide, given an n-vertex graph G and an integer h, whether the Hadwiger number of G is greater than or equal to h (or, equivalently, whether G has K_h as a minor). Rather surprisingly, it was not until 2009 that this problem was shown to be NP-complete by Eppstein [6]. Two years earlier, Alon et al. [1] observed that the problem is fixed-parameter tractable when parameterized by h due to deep results by Robertson and Seymour [10]. This shows that the problem of determining the Hadwiger number of a graph is in

The research leading to these results has received funding from the Research Council of Norway and the European Research Council under the European Union's Seventh Framework Programme (FP/2007–2013)/ERC Grant Agreement n. 267959.

D. Kratsch and I. Todinca (Eds.): WG 2014, LNCS 8747, pp. 201–213, 2014.
DOI: 10.1007/978-3-319-12340-0_17

some sense easier than the closely related problem of determining the clique number of a graph, as the decision version of the latter problem is W[1]-hard when parameterized by the size of the clique. Alon et al. [1] showed that the same holds from an approximation point of view: they provided a polynomial-time approximation algorithm for the Hadwiger Number problem with approximation ratio $O(\sqrt{n})$, contrasting the fact that it is NP-hard to approximate the clique number of an n-vertex graph in polynomial time to within a factor better than $n^{1-\epsilon}$ for any $\epsilon > 0$ [13].

Bollobás et al. [2] referred to the Hadwiger number as the *contraction clique number*. This is motivated by the observation that for any integer h, a connected graph G has K_h as a minor if and only if G has K_h as a contraction. In this context, it is worth mentioning another problem that has recently attracted some attention from the parameterized complexity community. The Clique Contraction problem takes as input an n-vertex graph G and an integer k, and asks whether G can be modified into a complete graph by a sequence of at most k edge contractions. Since every edge contraction reduces the number of vertices by exactly 1, it holds that (G, k) is a yes-instance of the Clique Contraction problem if and only if G has the complete graph K_{n-k} as a contraction (or, equivalently, as a minor). Therefore, the Clique Contraction problem can be seen as the parametric dual of the Hadwiger Number problem, and is NP-complete on general graphs. When parameterized by k, the Clique Contraction problem was recently shown to be fixed-parameter tractable [4,9], but the problem does not admit a polynomial kernel unless NP $\subseteq$ coNP/poly [4].

In this paper, we study the computational complexity of the Hadwiger Number problem on several graph classes of bounded chordality. For chordal graphs, which form an important subclass of 4-chordal graphs, the Hadwiger Number problem is easily seen to be equivalent to the problem of finding a maximum clique, and can therefore be solved in linear time on this class [12]. In Sect. 3, we present polynomial-time algorithms for solving the Hadwiger Number problem on two other well-known subclasses of 4-chordal graphs: cographs and bipartite permutation graphs. We also prove that the problem remains NP-complete on co-bipartite graphs, and hence on 4-chordal graphs. The latter result implies that the problem is also NP-complete on AT-free graphs, a common superclass of cographs and bipartite permutation graphs.

In Sect. 4, we consider a natural generalization of the Hadwiger Number problem, and provide additional results about finding large minors of bounded diameter. We show that the problem of determining the largest integer h such that a graph G has a minor with h vertices and diameter at most s can be solved in polynomial time on AT-free graphs if $s \geq 2$. In contrast, we show that this problem is NP-hard on chordal graphs for every fixed $s \geq 2$, and remains NP-hard for $s = 2$ even when restricted to split graphs. Observe that when $s = 1$, the problem is equivalent to the Hadwiger Number problem and thus NP-hard on AT-free graphs and linear-time solvable on chordal graphs due to our aforementioned results.

Due to space restrictions, proofs are either omitted or just sketched in this extended abstract. The full version of the paper is available at [5].

2 Preliminaries

We consider finite undirected graphs without loops or multiple edges. For each of the graph problems considered in this paper, we let $n = |V(G)|$ and $m = |E(G)|$ denote the number of vertices and edges, respectively, of the input graph G. For a graph G and a subset $U \subseteq V(G)$ of vertices, we write $G[U]$ to denote the subgraph of G induced by U. We write $G-U$ to denote the subgraph of G induced by $V(G) \setminus U$, and $G - u$ if $U = \{u\}$. For a vertex v, we denote by $N_G(v)$ the set of vertices that are adjacent to v in G. The *distance* $\mathrm{dist}_G(u, v)$ between vertices u and v of G is the number of edges on a shortest path between them. The *diameter* $\mathrm{diam}(G)$ of G is $\max\{\mathrm{dist}_G(u, v) \mid u, v \in V(G)\}$. The *complement* of G is the graph $\overline{G}$ with vertex set $V(G)$, where two distinct vertices are adjacent in $\overline{G}$ if and only if they are not adjacent in G. For two disjoint vertex sets $X, Y \subseteq V(G)$, we say that X and Y are *adjacent* if there are $x \in X$ and $y \in Y$ that are adjacent in G.

We say that P is a (u, v)*-path* if P is a path that joins u and v. The vertices of P different from u and v are the *inner* vertices of P. We denote by P_n and C_n the path and the cycle on n vertices respectively. The *length* of a path is the number of edges in the path. A set of pairwise adjacent vertices is a *clique*. A *matching* is a set M of edges such that no two edges in M share an end-vertex. A vertex incident to an edge of a matching M is said to be *saturated* by M. We write K_n to denote the *complete* graph on n vertices, i.e., graph whose vertex set is a clique. For two integers $a \leq b$, the *(integer) interval* $[a, b]$ is defined as $[a, b] = \{i \in \mathbb{Z} \mid a \leq i \leq b\}$. If $a > b$, then $[a, b] = \emptyset$.

The *chordality* $\mathrm{chord}(G)$ of a graph G is the length of a longest induced cycle in G; if G has no cycles, then $\mathrm{chord}(G) = 0$. For a non-negative integer k, a graph G is k*-chordal* if $\mathrm{chord}(G) \leq k$. A graph is *chordal* if it is 3-chordal. A graph is *chordal bipartite* if it is both 4-chordal and bipartite. A graph is a *split* graph if its vertex set can be partitioned in an independent set and a clique. For a graph F, we say that a graph G is F*-free* if G does not contain F as an induced subgraph. A graph is a *cograph* if it is P_4-free. Let σ be a permutation of $\{1, \ldots, n\}$. A graph G is said to be a *permutation graph for* σ if G has vertex set $\{1, \ldots, n\}$ and two vertices i, j are adjacent if and only if i, j are reversed by the permutation. A graph G is a *permutation* graph if G is a permutation graph for some σ. A graph is a *bipartite permutation* graph if it is bipartite and permutation. An *asteroidal triple (AT)* is a set of three non-adjacent vertices such that between each pair of them there is a path that does not contain a neighbor of the third. A graph is *AT-free* if it contains no AT. Each of the above-mentioned graph classes can be recognized in polynomial (in most cases linear) time, and they are closed under taking induced subgraphs [3,7]. See the monographs by Brandstädt et al. [3] and Golumbic [7] for more properties and characterizations of these classes and their inclusion relationships.

Minors, Induced Minors, and Contractions. Let G be a graph and let $e \in E(G)$. The *contraction* of e removes both end-vertices of e and replaces them by a new vertex adjacent to precisely those vertices to which the two end-vertices were adjacent. We denote by G/e the graph obtained from G be the contraction of e. For a set of edges S, G/S is the graph obtained from G by the contraction of all edges of S. A graph H is a *contraction* of G if $H = G/S$ for some $S \subseteq E(G)$. We say that G is *k-contractible* to H if $H = G/S$ for some set $S \subseteq E(G)$ with $|S| \leq k$. A graph H is an *induced minor* of G if a H is a contraction of an induced subgraph of G. Equivalently, H is an induced minor of G if H can be obtained from G by a sequence of vertex deletions and edge contractions. A graph H is a *minor* of a graph G if H is a contraction of a subgraph of G. Equivalently, H is a minor of G if H can be obtained from G by a sequence of vertex deletions, edge deletions, and edge contractions.

Let G and H be two graphs. An *H-witness structure* $\mathcal{W}$ of G is a partition $\{W(x) \mid x \in V(H)\}$ of the vertex set of a (not necessarily proper) subgraph of G into $|V(H)|$ sets called *bags*, such that the following two conditions hold:

(i) each bag $W(x)$ induces a connected subgraph of G;
(ii) for all $x, y \in V(H)$ with $xy \in E(H)$, bags $W(x)$ and $W(y)$ are adjacent in G.

In addition, we may require an H-witness structure to satisfy one or both of the following additional conditions:

(iii) for all $x, y \in V(H)$ with $xy \notin E(H)$, bags $W(x)$ and $W(y)$ are not adjacent in G;
(iv) every vertex of G belongs to some bag.

By contracting each of the bags into a single vertex we observe that H is a contraction, an induced minor, or a minor of G if and only if G has an H-witness structure $\mathcal{W}$ that satisfies conditions (i)–(iv), (i)–(iii), or (i)–(ii), respectively. We will refer to such a structure $\mathcal{W}$ as an *H-contraction structure*, an *H-induced minor structure*, and an *H-minor structure*, respectively. Observe that, in general, such a structure $\mathcal{W}$ is not uniquely defined.

Let $\mathcal{W}$ be an H-witness structure of G, and let $W(x)$ be a bag of $\mathcal{W}$. We say that $W(x)$ is a *singleton* if $|W(x)| = 1$ and $W(x)$ is an *edge-bag* if $|W(x)| = 2$. We say that $W(x)$ is a *big bag* if $|W(x)| \geq 2$.

We conclude this section by presenting three structural lemmas that will be used in the polynomial-time algorithms presented in Sect. 3. The first lemma readily follows from the definitions of a minor, an induced minor, and a contraction.

Lemma 1. *For every connected graph G and non-negative integer p, the following statements are equivalent:*

- *G has K_p as a contraction;*
- *G has K_p as an induced minor;*
- *G has K_p as a minor.*

We say that an H-induced minor structure $\mathcal{W} = \{W(x) \mid x \in V(H)\}$ is *minimal* if there is no H-induced minor structure $\mathcal{W}' = \{W'(x) \mid x \in V(H)\}$ with $W'(x) \subseteq W(x)$ for every $x \in V(H)$ such that at least one inclusion is proper.

Lemma 2. *For any minimal K_p-induced minor structure of a graph G, each bag induces a subgraph of diameter at most* $\max\{\text{chord}(G) - 3, 0\}$.

Note that Lemma 2 immediately implies the aforementioned equivalence on chordal graphs between the Hadwiger Number problem and the problem of finding a maximum clique. Lemma 2 also implies the following result.

Corollary 1. *If G is a graph of chordality at most 4, then for any minimal K_p-induced minor structure in G, each bag is a clique.*

We say that a K_p-induced minor structure is *nice* if each bag is either a singleton or an edge-bag.

Lemma 3. *Let G be a $\overline{C}_6$-free graph of chordality at most 4. If K_p is an induced minor of G, then G has a nice K_p-induced minor structure.*

3 Computing the Hadwiger Number

First, we show that Hadwiger Number problem can be solved in polynomial time on bipartite permutation graphs.

Let us for a moment consider the class of chordal bipartite graphs. Recall that these are exactly the bipartite graphs that have chordality at most 4. It is well-known that chordal bipartite graphs form a proper superclass of the class of bipartite permutation graphs. Since chordal bipartite graphs have chordality at most 4 and are $\overline{C}_6$-free due to the absence of triangles, we can apply Lemma 3 to this class. Let us additionally observe that the number of singletons in any K_p-induced minor structure of a bipartite graph is at most 2.

The above observations allow us to reduce the Hadwiger Number problem on chordal bipartite graphs to a special matching problem as follows. We say that a matching M in a graph G is a *clique-matching* if for any two distinct edges $e_1, e_2 \in M$, there is an edge in G between an end-vertex of e_1 and an end-vertex of e_2. Now consider the following decision problem:

Clique-Matching
Instance: A graph G and a positive integer k.
Question: Is there a clique-matching of size at least k in G?

Lemma 4. *If the* Clique-Matching *problem can be solved in $f(n, m)$ time on chordal bipartite graphs, then the* Hadwiger Number *problem can be solved in $O((n + m) \cdot f(n, m))$ time on this graph class.*

We will use the following characterization of bipartite permutation graphs given by Spinrad et al. [11] (see also [3]). Let G be a bipartite graph and let V_1, V_2 be a bipartition of $V(G)$. An ordering of vertices of V_2 has the *adjacency property* if for every $u \in V_1$, $N_G(u)$ consists of vertices which are consecutive in the ordering of V_2. An ordering of vertices of V_2 has the *enclosure property* if for every pair of vertices $u, v \in V_1$ such that $N_G(u) \subseteq N_G(v)$, vertices in $N_G(v) \setminus N_G(u)$ occur consecutively in the ordering of V_2.

Lemma 5 [11]**.** *Let G be a bipartite graph with bipartition V_1, V_2. The graph G is a bipartite permutation graph if and only there is an ordering of V_2 that has the adjacency and enclosure properties. Moreover, bipartite permutation graphs can be recognized and the corresponding ordering of V_2 can be constructed in linear time.*

Theorem 1. *The* Clique-Matching *problem can be solved in $O(mn^4)$ time on bipartite permutation graphs.*

Proof. Let G be a bipartite permutation graph and let V_1, V_2 be a bipartition of the vertex set. We assume without loss of generality that G has no isolated vertices. Let $n_1 = |V_1|$ and $n_2 = |V_2|$. We present a dynamic programming algorithm for the problem. For simplicity, the algorithm we describe only finds the size of a maximum clique-matching M in G, but the algorithm can be modified to find a corresponding clique-matching as well.

Our algorithm starts by constructing an ordering σ_2 of V_2 that has the adjacency and enclosure properties, which can be done in linear time due to Lemma 5. From now on, we denote the vertices of V_2 by their respective rank in σ_2, that is $V_2 = \{1, \ldots, n_2\}$. Observe that for every vertex $u \in V_1$, $N_G(u)$ forms an interval of σ_2. The *rightmost* (resp. *leftmost*) neighbor of u in σ_2 is the vertex of $N_G(u)$ which is the largest (resp. smallest) in σ_2.

Let $uv \in E(G)$ with $u \in V_1$ and $v \in V_2$ be an edge in G such that uv belongs to some maximum clique-matching in G and there is no $v' \in V_2$ with $v' < v$ such that v' is saturated by a maximum clique-matching in G. Our algorithm guesses the edge uv by trying all different edges of G. For each guess of uv, it does as follows.

By the definition of uv, we can safely delete all vertices $v' \in V_2$ with $v' < v$. To simplify notation, we assume without loss of generality that $v = 1$, so $uv = u1$. Denote by r the rightmost neighbor of u. Then, by the adjacency property of σ_2, we have that $N_G(u) = [1, r]$.

The algorithm now performs the following preprocessing procedure.

– Find the vertices $v_1, \ldots, v_l \in V_1 \setminus \{u\}$ (decreasingly ordered with respect to their rightmost neighbor) such that $[1, r] \subseteq N_G(v_i)$. By consecutively checking the intervals $N_G(v_1), \ldots, N_G(v_l)$ and selecting the rightmost available (i.e., not selected before) vertex in the considered interval, find the maximum set $S = \{j_1, \ldots, j_h\}$ of integers such that $j_1 > \ldots > j_h > r$ and $j_i \in N_G(v_i)$ for $i \in \{1, \ldots, h\}$. Delete $v_1, \ldots, v_h$ from G.

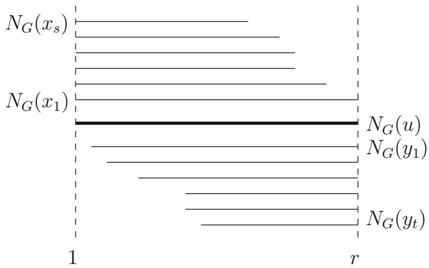

Fig. 1. Structure of the neighborhoods of u, $x_1, \ldots, x_s$ and $y_1, \ldots, y_t$ after the preprocessing procedure.

- Find the vertices $x_1, \ldots, x_s \in V_1 \setminus \{u\}$ (decreasingly ordered with respect to their rightmost neighbor) such that $[1, 2] \subseteq N_G(x_i)$.
- Find the vertices $y_1, \ldots, y_t \in V_1$ (increasingly ordered with respect to their leftmost neighbor) such that $1 \notin N_G(y_i)$ and $r \in N_G(y_i)$.
- Delete the vertices $r + 1, \ldots, n_2$ from V_2.

The structure of the neighborhoods of u, $x_1, \ldots, x_s$ and $y_1, \ldots, y_t$ after this preprocessing procedure is shown in Fig 1.

We prove that the preprocessing procedure is safe in the following claim.

Claim 1. *Let M be a clique-matching of maximum size in G such that $u1 \in M$. Then there is a clique-matching M' of maximum size such that $u1 \in M'$ and*

(i) $v_1 j_1, \ldots, v_h j_h \in M'$,
(ii) for any $vj \in M'$ such that $vj \neq u1$ and $v \notin \{v_1, \ldots v_h\}$, it holds that $v \in \{x_1, \ldots, x_s\} \cup \{y_1, \ldots, y_t\}$ and $j \in [2, r]$.

In the next stage of the algorithm we apply dynamic programming. For every $i \in \{0, \ldots, s\}$, $j \in \{0, \ldots, t\}$ and non-negative integer ℓ, let $c(i, j, \ell)$ denote the size of a maximum clique-matching M such that

(a) $u1 \in M$,
(b) for any $vp \in M$ such that $vp \neq u1$, it holds that $v \in \{x_1, \ldots, x_i\} \cup \{y_1, \ldots, y_j\}$, and
(c) there are at most ℓ vertices in $[a_{i,j}, b_{i,j}] = (\bigcap_{p=1}^{i} N_G(x_p)) \cap (\bigcap_{q=1}^{j} N_G(y_q))$ saturated by M.

Recall that the vertices of X and Y are ordered with respect to their rightmost and leftmost neighbors, respectively. Hence, for any $1 \leq p < q \leq i$, we have $1 \in N_G(x_q) \subseteq N_G(x_p) \subseteq [1, r]$, and for any $1 \leq p < q \leq j$, we have $1 \notin N_G(y_q) \subseteq N_G(y_p) \subseteq [2, r]$. In particular, $[a_{i,j}, b_{i,j}] = N_G(x_i) \cap N_G(y_j)$ for $i, j > 0$. In other words, if $[a_{i,j}, b_{i,j}] \neq \emptyset$, then $a_{i,j}$ is the left end-point of the interval $N_G(y_j)$ and $b_{i,j}$ is the right end-point of the interval $N_G(x_j)$. Observe that it can happen that $[a_{i,j}, b_{i,j}] = \emptyset$. Observe also that $c(i, j, \ell) = c(i, j, b_{i,j} - a_{i,j} + 1)$

if $[a_{i,j}, b_{i,j}] \neq \emptyset$ and $\ell > b_{i,j} - a_{i,j} + 1$. Hence, it is sufficient to compute $c(i, j, \ell)$ for $\ell \leq b_{i,j} - a_{i,j} + 1 \leq n_2$.

Because all the vertices in $[a_{i,j}, b_{i,j}]$ have the same neighbors in $\{x_1, \ldots, x_i\} \cup \{y_1, \ldots, y_j\}$, we can make the following observation.

Claim 2. *Let M be a clique-matching of maximum size such that M satisfies (a)–(c) and M has exactly f saturated vertices in $[a_{i,j}, b_{i,j}]$*, and let $W \subseteq [a_{i,j}, b_{i,j}]$ *be a set of size f. Then there is a clique-matching M' of maximum size that satisfies (a)–(c) such that W is the set of vertices of $[a_{i,j}, b_{i,j}]$ saturated by M'.*

If $i = j = 0$, then we set $c(i, j, \ell) = 1$ taking into account the matching with the unique edge $u1$. For other values of i, j, $c(i, j, \ell)$ is computed as follows. To simplify notation, we assume that $x_0 = y_0 = u$.

Computation of $c(i, j, \ell)$ for $i > 0, j = 0$. Because $1 \in N_G(x_q) \subseteq N_G(x_p) \subseteq [1, r]$ for every $1 \leq p < q \leq i$, any matching with edges incident to $x_1, \ldots, x_i$ is a clique-matching. This observation also implies that a maximum matching can be obtained in greedy way. Notice that $[a_{i,0}, b_{i,0}] = N_G(x_i)$. By consecutively checking the intervals $N_G(x_1), \ldots, N_G(x_i)$ and selecting the rightmost available (i.e., not selected before) vertex in the considered interval, we find the maximum set $\{p_1, \ldots, p_q\}$ of integers such that $t \geq p_1 > \ldots > p_q > 1$, $p_f \in N_G(x_f)$ for $f \in \{1, \ldots, q\}$, and $|\{p_1, \ldots, p_q\} \cap [a_{i,0}, b_{i,0}]| \leq \ell - 1$. Taking into account the edge $u1$, we observe that $M = \{u1, x_1p_1, \ldots, x_qp_q\}$ is a required matching, and we have that $c(i, j, \ell) = q + 1$.

Computation of $c(i, j, \ell)$ for $i = 0, j > 0$. Now we have that $r \in N_G(y_q) \subseteq N_G(y_p) \subseteq [2, r]$ for every $1 \leq p < q \leq j$. Hence, any matching with edges incident to $y_1, \ldots, y_j$ is a clique-matching and a maximum matching can be obtained in greedy way. Notice that $[a_{0,j}, b_{0,j}] = N_G(y_j)$. By consecutively checking the intervals $N_G(y_1), \ldots, N_G(y_j)$ and selecting the leftmost available (i.e., not selected before) vertex in the considered interval, we find the maximum set $\{p_1, \ldots, p_q\}$ of integers such that $1 < p_1 < \ldots < p_q \leq r$, $p_f \in N_G(y_f)$ for $f \in \{1, \ldots, q\}$, and $|\{p_1, \ldots, p_q\} \cap [a_{0,j}, b_{0,j}]| \leq \ell$. It is straightforward to see that $M = \{u1, y_1p_1, \ldots, y_qp_q\}$ is a required matching, and we have that $c(i, j, \ell) = q + 1$.

Computation of $c(i, j, \ell)$ for $i > 0, j > 0$. We compute $c(i, j, \ell)$ using the tables of already computed values $c(i - 1, j', \ell')$ for $j' \leq j$. We find the size of a maximum clique-matching M by considering all possible choices for the vertex x_i and then take the maximum among the obtained values. We distinguish three cases. Recall that $[a_{i,j}, b_{i,j}] = N_G(x_i) \cap N_G(y_j)$.

Case 1. The vertex x_i is not saturated by M. We have that $[a_{i-1,j}, b_{i-1,j}] = N_G(x_{i-1}) \cap N_G(y_j) \subseteq [a_{i,j}, b_{i,j}]$ and $|[a_{i,j}, b_{i,j}] \setminus [a_{i-1,j}, b_{i-1,j}]| = b_{i-1,j} - b_{i,j}$. By Claim 2 implies that for any maximum clique-matching M that satisfies (a)–(c) and has no edge incident to x_i, it holds that a clique-matching M' of maximum size that satisfies (a)–(b), has no edge incident to x_i, and has at most $\ell' = \ell + b_{i-1,j} - b_{i,j}$ saturated vertices in $[a_{i-1,j}, b_{i-1,j}]$ has the same size as M. Hence $c(i, j, \ell) = c(i - 1, j, \ell')$.

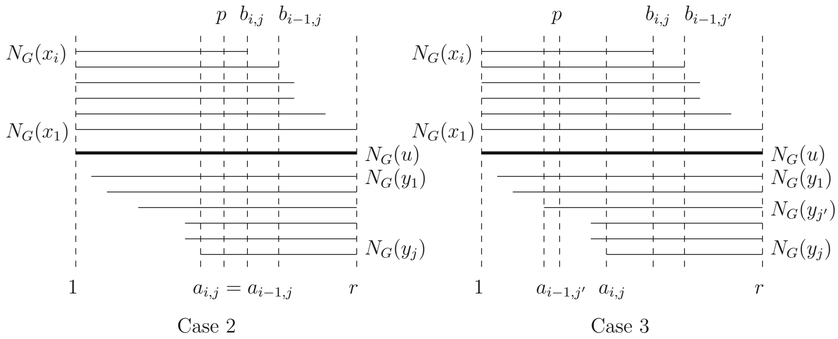

Fig. 2. Structure of the neighborhoods of u, $x_1, \ldots, x_i$ and $y_1, \ldots, y_j$ in Cases 2 and 3.

Now we consider the cases when x_i is saturated by M. Denote by $p \in N_G(x_i)$ the vertex such that $x_i p \in M$.

Case 2. Vertex $p \in [a_{i,j}, b_{i,j}]$ (see Fig. 2). Observe that p is adjacent to every vertex in $\{x_1, \ldots, x_{i-1}\} \cup \{y_1, \ldots, y_j\}$. Hence, for any edge vq such that $v \in \{u\} \cup \{x_1, \ldots, x_{i-1}\} \cup \{y_1, \ldots, y_j\}$ and $q \neq p$, $x_i p$ and vq have adjacent end-vertices, i.e., this choice of p does not influence the selection of other edges of M except that we can have at most $\ell - 1$ other saturated vertices in $[a_{i,j}, b_{i,j}]$. We have that $[a_{i-1,j}, b_{i-1,j}] = N_G(x_{i-1}) \cap N_G(y_j) \subseteq [a_{i,j}, b_{i,j}]$ and $|[a_{i,j}, b_{i,j}] \setminus [a_{i-1,j}, b_{i-1,j}]| = b_{i-1,j} - b_{i,j}$. By Claim 2, we obtain that for any maximum clique-matching M that satisfies (a)–(c) and $x_i p \in M$, a clique-matching M' of maximum size that satisfies (a)–(b), has no edge incident to x_i and has at most $\ell' = \ell + b_{i-1,j} - b_{i,j} - 1$ saturated vertices in $[a_{i-1,j}, b_{i-1,j}]$ has the same size as M. Hence $c(i, j, \ell) = c(i-1, j, \ell')$.

Case 3. Vertex $p \notin [a_{i,j}, b_{i,j}]$, i.e., $p < a_{i,j}$ (see Fig. 2). Let $j' = \max\{f \mid p \in N_G(y_f), 0 \leq f \leq j\}$. As $p < a_{i,j}$, it holds that $j' < j$.

Let $f \in \{j'+1, \ldots, j\}$, $g \in N_G(y_f)$ and $g > b_{i,j}$. Recall that $b_{i,j}$ is the right end-point of $N_G(x_i)$. Hence, $x_i g \notin E(G)$. Because $f > j'$, $x_f p \notin E(G)$. We conclude that such edges cannot be in M. Similarly, let $f \in \{j'+1, \ldots, j\}$, $g \in N_G(y_f)$ and $g \leq b_{i,j}$. Then for any $v \in \{x_1, \ldots, x_i\} \cup \{y_1, \ldots, y_{j'}\}$, it holds that $vg \in E(G)$. Also if $j'+1 \leq f < f' \leq j$, then for any $g \in N_G(x_{f'})$, $x_f g \in E(G)$. We have that it is safe to include in a clique-matching edges $x_f q$ for $f \in \{j'+1, \ldots, j\}$, $g \in N_G(y_f)$ and $g \leq b_{i,j}$. We select such edges in a greedy way. By consecutively checking the intervals $N_G(y_{j'+1}), \ldots, N_G(y_j)$ and selecting the leftmost available (i.e., not selected before) vertex in the considered interval, we find the maximum set $\{g_1, \ldots, g_q\}$ of integers such that $p < g_1 < \ldots < g_q \leq b_{i,j}$, $g_f \in N_G(y_{f+j'})$ for $f \in \{1, \ldots, q\}$ and $|\{g_1, \ldots, g_q\} \cap [a_{i,j}, b_{i,j}]| \leq \ell$.

Claim 3. *Let M be a clique-matching of maximum size that satisfies (a)–(c) and $x_i p \in M$. Then there is a clique-matching M' of maximum size that satisfies* *(a)–(c) and* $x_i p \in M'$ *such that* $y_{j'+1}g_1, \ldots, y_{j'+q}g_q \in M'$ *and for any* $vf \in M'$*, it holds that* $v \in \{y_{j'+1}, \ldots, y_{j'+q}\} \cup \{x_1, \ldots, x_i\} \cup \{y_1, \ldots, y_{j'}\}$*.*

Observe that the total number of saturated vertices in $[a_{i-1,j'}, b_{i-1,j'}]$ should be at most $(a_{i,j} - a_{i-1,j'}) + (b_{i-1,j'} - b_{i,j}) + \ell$. Using Claims 2 and 3 and taking into account that $x_i p \in M$, we obtain that $c(i,j,\ell) = c(i-1,j',\ell')$ for $\ell' = (a_{i,j} - a_{i-1,j'}) + (b_{i-1,j'} - b_{i,j}) + \ell - (q+1)$.

By our dynamic programming algorithm we eventually compute $c(s,t,\ell)$ for $\ell = 0$ if $[a_{i,j}, b_{i,j}] = \emptyset$ or $\ell = b_{i,j} - a_{i,j} + 1$ if $[a_{i,j}, b_{i,j}] \neq \emptyset$. Then $c(s,t,\ell)$ is the size of a maximum clique-matching M such that

(a) $u1 \in M$,
(b) for any $yp \in M$ such that $vp \neq u1$, it holds that $v \in \{x_1, \ldots, x_i\} \cup \{y_1, \ldots, y_j\}$.

By Claim 1, the size of a maximum clique-matching M in G such that $u1 \in M$ is $c(s,t,\ell) + |S|$, where S is the set of vertices constructed during the preprocessing procedure. Recall that the algorithm tries all possible choices for the edge uv, implying that our algorithm indeed computes the size of a maximum clique-matching in G.

It remains to evaluate the running time to complete the proof. Constructing the ordering σ_2 of V_2 can be done in $O(n+m)$ time by Lemma 5. The algorithm considers m choices for the edge uv. For each of these choices, the preprocessing procedure can be performed in $O(n)$ time given the orderings of V_1 and V_2 (notice that Lemma 5 is symmetric with respect to V_1, V_2, so we can obtain an ordering of V_1 with the adjacency and enclosure properties, too). Each step of the dynamic programming can be done in $O(n^2)$ time using the orderings of V_1, V_2. Observe that in this time we can compute $c(i,j,\ell)$ for all values of ℓ. Hence, the dynamic programming algorithm runs in time $O(n^4)$. We conclude that the total running time is $O(mn^4)$. □

Combining Lemma 4 and Theorem 1 yields the following result.

Corollary 2. *The* Hadwiger Number *problem can be solved in* $O((n+m) \cdot mn^4)$ *time on bipartite permutation graphs.*

We also show that the Hadwiger number of a cograph can be determined in polynomial time.

Theorem 2. *The* Hadwiger Number *problem can be solved in* $O(n^3)$ *time on cographs.*

We complement the aforementioned algorithmic results by showing that the Hadwiger Number problem is NP-complete on co-bipartite graphs, another well-known subclass of the class of 4-chordal graphs.

Theorem 3. *The* Hadwiger Number *problem is* NP*-complete on co-bipartite graphs.*

4 Minors of Bounded Diameter

In this section, we consider a generalization of the Hadwiger Number problem where the aim is to obtain a minor of bounded diameter. Let s be a positive integer. An *s-club* is a graph that has diameter at most s. We consider the following problem:

Maximum s-Club Minor
Instance: A graph G and a non-negative integer h.
Question: Does G have a minor with h vertices and diameter at most s?

When $s = 1$, the above problem is equivalent to the Hadwiger Number problem. Recall that, due to Lemma 1, the Hadwiger Number problem can be seen as the parametric dual of the Clique Contraction problem. The following straightforward lemma, which generalizes Lemma 1, will allow us to formulate the parametric dual of the Maximum s-Club Minor problem in a similar way.

Lemma 6. *For every connected graph G and non-negative integers p and s, the following statements are equivalent:*

- *G has a graph with p vertices and diameter at most s as a contraction;*
- *G has a graph with p vertices and diameter at most s as an induced minor;*
- *G has a graph with p vertices and diameter at most s as a minor.*

Lemma 6 implies that for any non-negative integer s, the parameteric dual of the Maximum s-Club Minor problem can be formulated as follows:

s-Club Contraction
Instance: A graph G and a positive integer k.
Question: Does there exist a graph H with diameter at most s such that G is k-contractible to H?

Observe that 1-Club Contraction is NP-complete on AT-free graphs as a result of Theorem 3. This is in stark contrast with our next result.

Theorem 4. *For any $s \geq 2$, the s-Club Contraction problem can be solved in $O(m^4 n^3)$ time on AT-free graphs, even if s is given as a part of the input.*

On chordal graphs, the situation turns out to be opposite. Recall that the Hadwiger Number problem, and hence the 1-Club Contraction problem, can be solved in linear time on chordal graphs.

Theorem 5. *For any $s \geq 2$, the s-Club Contraction problem on chordal graphs is NP-complete as well as W[2]-hard when parameterized by k. Moreover, 2-Club Contraction is NP-complete and W[2]-hard when parameterized by k even on split graphs.*

5 Concluding Remarks

In Sect. 3, we showed that the Hadwiger Number problem can be solved in polynomial time on cographs and on bipartite permutation graphs, respectively. A natural question is how far the results in those two sections can be extended to larger graph classes. An easy reduction from the Hadwiger Number problem on general graphs, involving subdividing every edge of the input graph exactly once, implies that the problem is NP-complete on bipartite graphs. Since bipartite permutation graphs form exactly the intersection of bipartite graphs and permutation graphs, and the class of permutation graphs properly contains the class of cographs, our results naturally raise the question whether the Hadwiger Number problem can be solved in polynomial time on permutation graphs. We leave this as an open question. We point out that the problem is NP-complete on co-comparability graphs, a well-known superclass permutation graphs, due to Theorem 3 and the fact that co-bipartite graphs form a subclass of co-comparability graphs.

In Sect. 4, we proved that the s-Club Contraction problem is polynomial on AT-free graphs for $s \geq 2$. An interesting direction for further research is to identify other non-trivial graph classes for which the s-Club Contraction problem is polynomial-time solvable (or fixed-parameter tractable when parameterized by k) for all values of $s \geq 2$.

References

1. Alon, N., Lingas, A., Wahlén, M.: Approximating the maximum clique minor and some subgraph homeomorphism problems. Theor. Comput. Sci. **374**, 149–158 (2007)
2. Bollobás, B., Catlin, P.A., Erdös, P.: Hadwiger's conjecture is true for almost every graph. Eur. J. Comb. **1**, 195–199 (1980)
3. Brandstädt, A., Le, V., Spinrad, J.: Graph Classes: A Survey. SIAM Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia (1999)
4. Cai, L., Guo, C.: Contracting few edges to remove forbidden induced subgraphs. In: Gutin, G., Szeider, S. (eds.) IPEC 2013. LNCS, vol. 8246, pp. 97–109. Springer, Heidelberg (2013)
5. Golovach, P.A., Heggernes, P., van 't Hof, P., Paul C.: Hadwiger number of graphs with small chordality. CoRR abs/1406.3812 (2014)
6. Eppstein, D.: Finding large clique minors is hard. J. Graph Algorithms Appl. **13**(2), 197–204 (2009)
7. Golumbic, M.C.: Algorithmic Graph Theory and Perfect Graphs. Annals of Discrete Mathematics, vol. 57, 2nd edn. Elsevier, Amsterdam (2004)
8. Hadwiger, H.: Über eine klassifikation der streckenkomplexe. Vierteljschr. Naturforsch. Ges. Zürich **88**, 133–143 (1943)
9. Lokshtanov, D., Misra, N., Saurabh, S.: On the hardness of eliminating small induced subgraphs by contracting edges. In: Gutin, G., Szeider, S. (eds.) IPEC 2013. LNCS, vol. 8246, pp. 243–254. Springer, Heidelberg (2013)
10. Robertson, N., Seymour, P.: Graph minors. XIII. The disjoint paths problem. J. Comb. Theory Ser. B **63**, 65–110 (1995)

11. Spinrad, J., Brandstädt, A., Stewart, L.: Bipartite permutation graphs. Discrete Appl. Math. **18**(3), 279–292 (1987)
12. Tarjan, R.E., Yannakakis, M.: Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. SIAM J. Comput. **13**(3), 566–579 (1984)
13. Zuckerman, D.: Linear degree extractors and the inapproximability of max clique and chromatic number. In: Proceedings of the STOC 2006, pp. 681–690 (2006)