On retracts, absolute retracts, and foldings in cographs

Ton Kloks\textsuperscript{1} · Yue-Li Wang\textsuperscript{2}

Abstract  A retract of a graph $G$ is an induced subgraph $H$ of $G$ such that there exists a homomorphism $\rho : G \rightarrow H$. When both $G$ and $H$ are cographs, we show that the problem to determine whether $H$ is a retract of $G$ is NP-complete; moreover, we show that this problem on cographs is fixed-parameter tractable when parameterized by the size of $H$. When restricted to the class of threshold graphs or to the class of trivially perfect graphs, the retract problem becomes tractable in polynomial time. The retract problem is also solvable in linear time when one cograph is given as an induced subgraph of the other. We characterize absolute retracts for the class of cographs. Foldings generalize retractions. We show that the problem to fold a trivially perfect graph onto a largest possible clique is NP-complete.

Keywords  Retracts · Absolute retracts · Foldings · Cographs

1 Introduction

For basic terminology on graph homomorphisms we refer to [17,21].
Definition 1.1 Let $G$ and $H$ be graphs. A homomorphism $\phi : G \to H$ is a map $\phi : V(G) \to V(H)$ which preserves edges, that is,

$$[x, y] \in E(G) \implies [\phi(x), \phi(y)] \in E(H).$$

(1)

We write $G \to H$ if there is a homomorphism $\phi : G \to H$.

Notice that $G \to K_k \iff \chi(G) \leq k$ and also that $K_k \to G \iff \omega(G) \geq k$, (2)

where $\chi(G)$ and $\omega(G)$ are the chromatic number and maximum clique number, respectively, of $G$.

Definition 1.2 Let $G$ and $H$ be graphs. The graph $H$ is a retract of $G$ if there exist homomorphisms $\rho : G \to H$ and $\gamma : H \to G$ such that $\rho \circ \gamma = \text{id}_H$, which is the identity map $V(H) \to V(H)$.

The functions $\rho$ and $\gamma$ are called the retraction and co-retraction, respectively.

When $H$ is a retract of $G$ then $H$ is isomorphic to an induced subgraph of $G$ [19]. Since there are homomorphisms in two directions, $G$ and $H$ have the same clique number, chromatic number and odd girth. Also, there is a retraction from $G$ to $K_k$ if and only if $\chi(G) = \omega(G) = k$.

There is a homomorphism $G \to H$ if and only if the union of $G$ and $H$ retracts to $H$. For any graph $H$, checking if there is a homomorphism $G \to H$ is polynomial when $H$ is bipartite and it is NP-complete otherwise [21]. It follows that, for any graph $H$, checking if a graph $G$ is a retract of a graph $g$ is NP-complete, unless $H$ is bipartite. The retract problem remains NP-complete, even when $H$ is an even cycle of length at least six, given as an induced subgraph of $G$ [9]. The question whether a graph $G$ has a homomorphism to itself which is not the identity is also NP-complete [20].

Graph homomorphisms have regained a lot of interest by the characterization of Grohe of the classes of graphs for which $\text{Hom}(G, -)$ is tractable [15]. To be precise, Grohe proves that, unless $\text{FPT} = \text{W}[1]$, deciding whether there is a homomorphism from a graph $G \in \mathcal{G}$ to some arbitrary graph $H$ is polynomial if and only if the graphs in $\mathcal{G}$ have bounded treewidth modulo homomorphic equivalence. The treewidth of a graph modulo homomorphic equivalence is defined as the treewidth of its core, i.e., a minimal retract. This makes it desirable to have algorithms that compute cores, or general retracts in graphs.

Definition 1.3 A graph is a cograph if it has no induced $P_4$, which is the path with four vertices.

Since the complement of a $P_4$ is a $P_4$, cographs are closed under complementation. Actually, a graph $G$ is a cograph if and only if one of the following holds.

1. $G$ has only one vertex, or
2. $G$ is disconnected and every component is a cograph, or
3. the complement of $G$, $\bar{G}$ is disconnected and every component of $\bar{G}$ is a cograph.

Springer
It follows that cographs have a decomposition tree, called a cotree, defined as follows. The decomposition tree is a rooted tree $T$. There is a bijection from the leaves of $T$ to the vertices of $G$. When $G$ has at least two vertices then each internal node of $T$, including the root, is labeled as $\otimes$ or $\oplus$. The $\oplus$ label at a node takes the union of the graphs that correspond with the children of the node. The $\otimes$ label takes the join of the graphs that correspond with the children.

A cocomponent of a graph $G$ is a subset of vertices which induces a component of the complement $\bar{G}$.

**Remark 1.4** When defined as above, we can assume that the labels of the internal nodes in any path from the root to a leaf alternate between $\oplus$ and $\otimes$. Alternatively, one frequently defines a cotree as a rooted binary tree, in which each internal node is labeled as $\oplus$ and $\otimes$. In this paper, when talking about cotrees, we always assume the first type of cotree. Thus, each child of the root corresponds with one component or with one cocomponent of the graph.

**Remark 1.5** It is well-known that cographs are recognizable in linear time [5]. A cotree has $O(n)$ nodes, where $n = |V(G)|$, and it can be obtained in linear time.

A subgraph $H$ of $G$ is a core of $G$ if there is a homomorphism $G \rightarrow H$ but no homomorphism $G \rightarrow H'$ for any proper subgraph $H'$ of $H$ [20]. For any graph $G$, all the cores of $G$ are isomorphic subgraphs of $G$. However, a fixed copy of the core in $G$ is not necessarily a retract. Therefore, when studying retracts or cores one usually assumes that the objective is given as an induced subgraph of $G$. In this paper we consider the retract problem for cographs. When restricted to cographs, it is easy to see that, when $H$ is given as an induced subgraph of $G$, it can be determined in linear time whether $H$ is a retract which will be described in Sect. 2. In the rest of the paper we do not assume that the graph $H$ is given as an induced subgraph of $G$. In that case the retract problem turns out to be NP-complete. We prove that in Sect. 2.

The related surjective graph homomorphism problem was recently studied in [11]. In this paper it was shown that the problem to decide whether there is a surjective homomorphism from one connected cograph to another connected cograph is NP-complete. The surjective homomorphism problem is also NP-complete if both graphs are unions of complete graphs [11]. Let us mention also the classic result of Damaschke, which is that the induced subgraph isomorphism problem is NP-complete for cographs [7].

The retract problem for cographs can be perceived as a pattern recognition problem for labeled trees. Many pattern recognition variants have been investigated and classified, see e.g., [4,7,14,23,24,28,30–33]. This last manuscript [33] contains references to a lot of the work done on motifs in graphs. However, the pattern recognition problem that corresponds with the retract problem on cographs seems to have eluded all these investigations [16].

This paper is organized as follows. In Sect. 2 we show that the retract problem is NP-complete for cographs. In Sects. 4 and 5 we show that the retract problem is polynomial when restricted to the classes of threshold and trivially perfect graphs which are subclasses of cographs. In Sect. 3 we show that the retract problem for cographs is fixed-parameter tractable. The characterization of absolute retracts for the
class of cographs is in Sect. 6. In Sect. 7 we show that computing the folding number is NP-complete for trivially perfect graphs. We conclude in Sect. 8.

2 NP-completeness of retracts in cographs

Recall that a graph $G$ is perfect when $\omega(G') = \chi(G')$ for every induced subgraph $G'$ of $G$. By the perfect graph theorem a graph is perfect if and only if it has no odd hole or odd antihole. This implies that cographs are perfect. Perfect graphs are recognizable in polynomial time. For a graph $G$, when $\omega(G) = \chi(G)$ one can compute this value in polynomial time via Lovász theta function.

The following lemma appears, e.g., in [10].

**Lemma 2.1** Assume that $\omega(H) = \chi(H)$. There is a homomorphism $G \to H$ if and only if $\chi(G) \leq \omega(H)$.

**Proof** Write $\omega = \omega(H) = \chi(H)$. First assume that there is a homomorphism $\phi : G \to H$. There is a homomorphism $f : H \to K_\omega$ since $H$ is $\omega$-colorable. Then $f \circ \phi : G \to K_\omega$ is a homomorphism, and so $G$ has an $\omega$-coloring. This implies that $\chi(G) \leq \omega$. Assume $\chi(G) \leq \omega$. There is a homomorphism $G \to K_k$, where $k = \chi(G)$. Since $K_k$ is an induced subgraph of $H$, there is also a homomorphism $K_k \to H$. This implies that $G$ is homomorphic to $H$, i.e., $G \to H$. $\Box$

**Corollary 2.2** When $G$ and $H$ are perfect one can check in polynomial time whether there is a homomorphism $G \to H$.

Throughout the remainder of this section it is assumed that $G$ and $H$ are cographs. Note that, using the cotree, $\omega(G)$ and $\chi(G)$ can be computed in linear time when $G$ is a cograph.

**Lemma 2.3** Assume $H$ is disconnected, with components $H_1, \ldots, H_t$. Assume that $H$ is a retract of a graph $G$. Then there is an ordering of the components of $G$, say $G_1, \ldots, G_s$ such that

(a) $s \geq t$, and
(b) $G_i$ retracts to $H_i$, for every $i \in \{1, \ldots, t\}$, and
(c) for every $j \in \{t + 1, \ldots, s\}$, there is a homomorphism $G_j \to H$.

**Proof** No connected graph has a disconnected retract since the homomorphic image of a connected graph is connected. To see that, notice that a homomorphism $\phi : G \to H$ is a vertex coloring of $G$, where the vertices of $H$ represent colors. By that we mean that, for each $v \in V(H)$, the pre-image $\phi^{-1}(v)$ is an independent set in $G$ or $\emptyset$. One obtains the image $\phi(G)$ by identifying vertices in $G$ that receive the same color. When $G$ is connected, this ‘quotient graph’ on the color classes is also connected, which is easy to prove by means of contradiction.

Assume that $G$ retracts to $H$. Then we may assume that $H_1, \ldots, H_t$ are induced subgraphs of components $G_1, \ldots, G_t$ of $G$ and that each $G_i$ retracts to $H_i$. For the remaining components $G_j$, where $j > t$, there is then a homomorphism $G_j \to H$. © Springer
Notice that, for \( j > t \), we can check if there is a homomorphism \( G_j \rightarrow H \) by checking if \( G_j \oplus H_k \) retracts to \( H_k \), for some \( 1 \leq k \leq t \) [35] or, equivalently (since cographs are perfect), if \( \omega(G_j) \leq \omega(H_k) \) for some \( 1 \leq k \leq t \).

**Remark 2.4** Assume that we are given, for each pair \( G_i \) and \( H_j \) whether \( G_i \) retracts to \( H_j \) or not. Then, to check if \( G \) retracts to \( H \), we may consider a bipartite graph \( B \) defined as follows. One color class of \( B \) has the components of \( G \) as vertices and the other color class has the components of \( H \) as vertices. There is an edge between \( G_i \) and \( H_j \) whenever \( G_i \) retracts to \( H_j \). To check if \( G \) retracts to \( H \), we can let an algorithm compute a maximum matching in \( B \). There is a retraction only if the matching exhausts all components of \( H \) and if \( \omega(G) = \omega(H) \).

**Lemma 2.5** Assume \( G \) is disconnected and assume that \( G \) retracts to \( H \). Let \( G_1, \ldots, G_t \) be the subgraphs of \( G \) induced by the cocomponents of \( G \). Then \( V(H) \) can be partitioned into \( t \) parts and the induced subgraphs of those parts can be ordered as \( H_1, \ldots, H_t \) such that \( G_i \) retracts to \( H_i \) for \( i \in \{1, \ldots, t\} \).

**Proof** Every subgraph \( G_i \) of \( G \), induced by a cocomponent, retracts to some induced subgraph of \( H \). By Lemma 2.3, these retracts are pairwise joined, so each \( G_i \) retracts to some subgraph induced by a cocomponent \( H_i \) of \( H \) for \( i \in \{1, \ldots, t\} \). Thus \( V(H_i) \) for \( i \in \{1, \ldots, t\} \) form a partition of \( H \).

**Theorem 2.6** Let \( G \) and \( H \) be cographs. The problem to decide whether \( H \) is a retract of \( G \) is NP-complete.

**Proof** Clearly, the problem to decide whether \( H \) is a retract of \( G \) is in NP. It remains to show that all the other problems in NP can be reduced to this problem. We reduce the 3-partition problem to the retract problem on cotrees.

The 3-partition problem is the following. Let \( m \) and \( B \) be integers. Let \( S \) be a multiset of \( 3m \) positive integers, \( a_1, \ldots, a_{3m} \). Determine if there is a partition of \( S \) into \( m \) subsets \( S_1, \ldots, S_m \), such that the sum of the numbers in each subset is \( B \). Without loss of generality we assume that each number is strictly between \( B/4 \) and \( B/2 \), which guarantees that in a solution each subset contains exactly three numbers that add up to \( B \).

The 3-partition problem is strongly NP-complete, that is, the 3-partition problem remains NP-complete when all the numbers in the input are represented in unary.

In our reduction, the cotree for the graph \( H \) has a root which is a join-node \( \otimes \) with label \( r_H \) (see Fig. 1a). The root has \( 3m \) children, one for each number \( a_i \). For simplicity we refer to the children as \( a_i, i \in \{1, \ldots, 3m\} \). Each child \( a_i \) has a union node \( \oplus \) as the root. The root of each \( a_i \)-child has two children, one is a single leaf and the other is a join-node \( \otimes \) with \( a_i \) leaves. This ends the description of \( H \). Let \( T_H \) be the cotree for the graph \( H \). Note that constructing \( T_H \) takes \( O(n + mB) \) time.

The cotree for the graph \( G \) has a join-node \( \otimes \) as a root with label \( r_G \) and \( r_G \) has \( m \) children (see Fig. 1b). The idea is that each child corresponds with one set of a 3-partition of \( S \). The subtrees for all the children are identical. Each subtree has a union-node \( \oplus \) as the root with label \( S_x \) for \( 1 \leq x \leq m \). Consider all triples \( \{i, j, k\} \) for which \( a_i + a_j + a_k = B \). For each such triple create one child, which is the join of three cotrees, one for \( a_i \), one for \( a_j \) and one for \( a_k \) in the triple. The subtree for \( a_i \) is a
union of two subtrees. As in the cotree for the pattern $H$, one subtree is a single leaf, and the other subtree is the join of $a_i$ leaves. The other two subtrees, for the numbers $a_j$ and $a_k$ in the triple are similar. Let $T_G$ be the cotree for the graph $G$. In constructing $T_G$, it takes $O(n^3)$ time to find out all possible triples $\{i, j, k\}$ with $a_i + a_j + a_k = B$. Thus the number of leaves in the subtree of $T_G$ with root $S_i$ for $1 \leq i \leq m$ is at most $O(n^3 B)$. Therefore, constructing $T_G$ takes at most $O(mn^3 B)$ time.

When the graph $H$ is a retract of $G$, then the $a_i$-children of $r_H$ are partitioned into triples, such that there is a bijection between the set of triples, say $\{a_i, a_j, a_k\}$, and the set of subtrees rooted at children of $r_G$. Each $\oplus$-node which is the root of a child of $r_G$ must have exactly one $\{a_i, a_j, a_k\}$-child that corresponds with the triple. Notice that, by the construction, all subgraphs induced by remaining components of this $\oplus$-node have maximal cliques of size $B$. Therefore, by Lemma 2.1, all other children of this $\oplus$-node are homomorphic to the one child which corresponds to the triple $\{a_i, a_j, a_k\}$.

It now follows from Lemma 2.5 that there is a 3-partition if and only if the graph $H$ is a retract of $G$. This completes the proof.

Notice that when $H$ is given as an induced subgraph of cograph $G$, it can be determined in linear time whether $H$ is a retract of $G$. We describe the algorithm as follows:
Algorithm A

**Input:** A cograph $G$ and an induced subgraph $H$ of $G$.

**Output:** Output ‘Yes’, if $H$ is a retract of $G$; otherwise, output ‘No’.

1. **Step 1.** /* Initialization. */
   
   Construct a cotree $T$ for $G$.

2. **Step 2.** /* Remove nodes. */
   
   For each $\oplus$-node $x$ from the bottom level to the top level, execute the following substeps:
   
   (a) **Step 2.1.** For each child $C_i$ of $x$ which has no leaves in $H$, check that whether there is another child $C_j$ of $x$ with children in $H$ and $\omega(C_i) \leq \omega(C_j)$. If there exists some $C_i$ which does not satisfy the condition above, then output ‘No’ and terminate; otherwise, mark $C_i$.

   (b) **Step 2.2.** Reconstruct $T$ as follows: Remove the subtree rooted at $x$ from $T$ and add the vertices in $C_j$ as children to the parent of $x$, if it exists, where $C_j$ is an unmarked child of $x$ with the maximum $\omega(C_j)$ among all unmarked children of $x$.

3. **Step 3.** /* Output. */
   
   Output ‘Yes’.

---

### 3 A fixed-parameter solution for retracts in cographs

In this section we look at a parameterized solution for the retract problem. Let $G$ and $H$ be cographs. We consider the parameterization by the number of vertices in $H$. Let

$$k = |V(H)|.$$  

**Proposition 3.1** When $H$ is a retract of $G$ then $\omega(G) = \omega(H) \leq k$. Let $T_G$ be a cotree for $G$. Then every join-node in $T_G$ has at most $k$ children, and the height of the cotree is $O(k)$.

**Theorem 3.2** The retract problem, which asks if a cograph $H$ is a retract of $G$, is fixed-parameter tractable. The retract problem can be solved in $O(k^2 \cdot |V(G)|^{5/2})$ time, where $k = |V(H)|$.

**Proof** Consider cotrees $T_G$ and $T_H$ for $G$ and $H$ and let $r_G$ and $r_H$ be the roots of the two cotrees. Assume both roots are $\boxtimes$-nodes. Then, by Proposition 3.1, both have at most $k$ children. According to Lemma 2.5, when $H$ is a retract of $G$ there is a partition $P$ of the branches incident with $r_H$ such that each child of $r_G$ represents a graph that retracts to the subgraph of $H$ induced by exactly one part of the partition. Let $p$ be the number of children of $r_G$ and let $q$ be the number of children of $r_H$. The number of partitions of a $q$-set into $p$ nonempty parts is given by the Stirling number of the second kind. A trivial upper bound for the number of different assignments of the children of $r_H$ to the children of $r_G$ is

$$p^q \leq k^k,$$

since, by Lemma 2.1, $p \leq k$ and $q \leq k$.

Our algorithm tries all possible partitions of the children of $r_H$. Consider a partition $P$, and let $P_i \in P$ be mapped to the $i$th child of $r_G$. Let $H_i$ be the subgraph of $H$ induced by $P_i$ and let $G_i$ be the cocomponent of $G$ induced by the $i$th child of $r_G$. We
proceed as in the proof of Theorem 5.3. Let \( C_1, \ldots, C_i \) be the components of the root of the \( i \)th child of \( r_G \). Let \( D_1, \ldots, D_i \) be the components of \( H_i \). Consider the bipartite graph with vertices the components of \( G_i \) and \( H_i \), where an edge \((C_1, D_1)\) indicates that \( C_1 \) retracts to \( D_1 \). The algorithm checks if there is a matching that exhausts all components of \( H_i \), and it checks if the remaining components of \( G_i \) are homomorphic to some component of \( H_i \).

Since the height of the cotree is bounded by \( k \), it follows that this algorithm can be implemented to run in \( O(k^{k^2} \cdot (k \cdot |V(G)|)^{5/2}) \). This proves the theorem. \( \square \)

Finding a retract to \( H \), when \( H \) is a fixed graph, is just an \( H \)-coloring problem (with a fixed number of colors). This can be formulated in monadic second-order logic [6]. The size of the formula depends only on \( |V(H)| \). Thus we have the following proposition.

**Proposition 3.3** For every \( H \), the \( H \)-retract problem can be formulated in monadic second-order logic (without quantification over subsets of edges).

By Courcelle’s theorem we may also conclude the following.

**Corollary 3.4** There exists a function \( f : \mathbb{N} \rightarrow \mathbb{N} \) such that, for every \( H \), say with \( k = |V(H)| \), there is an \( O(f(k) \cdot n) \) algorithm which checks if \( H \) is a retract of a cograph \( G \).

## 4 Retracts in threshold graphs

A subclass of the class of cographs is the class of threshold graphs. Threshold graphs are the graphs without induced \( 2K_2, C_4 \) and \( P_4 \). We use the following characterization of threshold graphs.

**Theorem 4.1** A graph is a threshold graph if and only if every induced subgraph has a universal vertex or an isolated vertex.

**Theorem 4.2** Let \( G \) and \( H \) be threshold graphs. There exists a linear-time algorithm to check if \( H \) is a retract of \( G \).

**Proof** Assume that \( H \) is a retract of \( G \) and let \( \rho \) and \( \gamma \) be the retraction and coretraction.

Assume that \( G \) has a universal vertex, say \( x_1 \). Then \( H \) must have a universal vertex as well, since a retract of a connected graph is connected. Let \( y_1 \) be a universal vertex of \( H \). Let \( y_i = \rho(x_1) \). Since \( \rho \) is a homomorphism it preserves edges, and since \( x_1 \) is universal in \( G \), \( \rho \) maps no other vertex of \( G \) to \( y_1 \). Notice also that \( \gamma(y_i) = x_1 \) since \( \rho \circ \gamma = id_H \) and \( \rho \) maps no other vertex to \( y_i \).

Assume that \( y_i \neq y_1 \). Let \( \gamma(y_1) = x_\ell \). Then \( x_\ell \neq x_1 \) since \( \gamma \) preserves edges and so

\[
\{y_1, y_i\} \in E(H) \implies \{\gamma(y_1), \gamma(y_i)\} = \{x_\ell, x_1\} \in E(G) \implies x_\ell \neq x_1.
\]

Furthermore, since \( y_1 \) is universal, \( \gamma \) maps no other vertex of \( H \) to \( x_\ell \). Of course, since \( \rho \circ \gamma = id_H \), \( \rho(x_\ell) = y_1 \).
We claim that $y_i$ is universal in $H$, and therefore exchangeable with $y_1$. Assume not and let $y_s \in V(H)$ be another vertex of $H$ not adjacent to $y_i$. Let $\gamma(y_s) = x_p$. Then $x_p \neq x_1$ since $\rho \circ \gamma = id_H$ and $\rho(x_1) = y_1 \neq y_s$. Now, since $\rho$ is a homomorphism,

$$\{x_1, x_p\} \in E(G) \Rightarrow \{\rho(x_1), \rho(x_p)\} = \{y_i, y_s\} \in E(H),$$

which is a contradiction. Therefore, we may assume that $y_i = y_1$. That is, from now on we assume that $\rho(x_1) = y_1$ and $\gamma(y_1) = x_1$.

This proves that, when $G$ is connected then $H$ is a retract of $G$ if and only if $H - y_1$ is a retract of $G - x_1$. By the way, notice that if $|V(H)| = 1$ then $H$ can be a retract of $G$ only if $G$ is an independent set, so this case is easy to check.

Finally, assume that $G$ is not connected. Since $G$ has no induced $2K_2$, all components, except possibly one, have only one vertex. The number of components of $H$ can be at most equal to the number of components of $G$, since $\rho$ maps components in $G$ to components of $H$, and $\rho \circ \gamma = id_H$, and so any two components of $H$ are mapped by $\gamma$ to different components of $G$.

First assume that $H$ is also disconnected. Let $x_1, \ldots, x_a$ be the isolated vertices of $G$ and let $y_1, \ldots, y_b$ be the isolated vertices of $H$. Let $\rho(x_i) = y_i$ and $\gamma(y_i) = x_i$ for $i \in \{1, \ldots, b\}$ and let $\rho(x_{b+1}) = \cdots = \rho(x_a) = y_b$. Now, $H$ is a retract of $G$ if and only if $H - \{x_1, \ldots, x_b\}$ is a retract of $G - \{x_1, \ldots, x_a\}$.

If $H$ is connected, with at least two vertices, then let $y_1$ be a universal vertex and let $\rho(x_1) = \cdots = \rho(x_a) = y_1$. If $H$ is a retract of $G$ then $G$ must have exactly one component with at least two vertices, since $G$ is a threshold graph and $\rho$ is a homomorphism. Let $x_a$ be the universal vertex of that component and define $\rho(x_a) = y_1$ and $\gamma(y_1) = x_a$. In this case, $H$ is a retract if and only if $H - y_1$ is a retract of $G - \{x_1, \ldots, x_a\}$.

An elimination ordering, which eliminates successive isolated and universal vertices in a threshold graph, can be obtained in linear time. This proves the theorem. \qed

## 5 Retracts in trivially perfect graphs

**Definition 5.1** [12, 34] A graph $G$ is trivially perfect if for all induced subgraphs $H$ of $G$, $\alpha(H)$ is equal to the number of maximal cliques in $H$, where $\alpha(H)$ is the cardinality of a maximum independent set.

Trivially perfect graphs are those graphs without induced $C_4$ and $P_4$.

**Theorem 5.2** [34] A graph is trivially perfect if and only if every connected induced subgraph has a universal vertex.

**Theorem 5.3** Let $G$ and $H$ be trivially perfect graphs. There exists an $O(N^{5/2})$ algorithm which checks if $H$ is a retract of $G$, where $N = |V(G)| + |V(H)|$.

**Proof** Assume that $H$ is a retract of $G$. Then, by Lemma 2.1, $\omega(G) = \Omega(H)$. This can be checked in linear time. Let $C_1, \ldots, C_t$ be the components of $G$ and let $D_1, \ldots, D_s$ be the components of $H$. Then $s \leq t$. Without loss of generality, let $D_i$ be a retract of $C_i$ for $i \in \{1, \ldots, s\}$. For the components $C_i$ with $i > s$, there must be a $j \leq s$ such
that there is a homomorphism from \(C_i\) to \(D_j\). Notice that, since \(\omega(G) = \omega(H)\), there is always a homomorphism from \(C_i\) to the component \(D_j\) of \(H\) with \(\omega(D_j) = \omega(H)\). Therefore, it suffices to find a matching from the components of \(G\) to the components of \(H\), as described below.

First assume that \(G\) and \(H\) are connected. Let \(g_1, \ldots, g_k\) be the universal vertices of \(G\) and let \(h_1, \ldots, h_\ell\) be the universal vertices of \(H\). As in the proof of Theorem 4.2 it follows that \(H\) is a retract of \(G\) if and only if

(i) \(\ell = k\), and

(ii) either \(H\) is a clique and \(\omega(G) = \omega(H)\) or \(H - \{h_1, \ldots, h_\ell\}\) is a retract of \(G - \{g_1, \ldots, g_k\}\).

For the general case, consider the following bipartite graph \(B\). The vertices of \(B\) are the components of \(G\) and \(H\). There is an edge \(\{C_i, D_j\} \in E(B)\) if and only if \(C_i\) retracts to \(D_j\). Then \(G\) retracts to \(H\) if and only if \(B\) has a matching that exhausts all components of \(H\).

Assume that \(G\) is disconnected. Recursively check if \(C_i\) retracts to \(D_j\). As described below, this is done in \(O((|C_i| + |D_j|)^{5/2})\) time. To check if a component \(G[C_i]\) retracts to some \(H[D_j]\) the algorithm greedily matches the universal vertices of \(G[C_i]\) and \(H[D_j]\) and checks if the remaining graph \(G'\), i.e., after removal of the matched universal vertices, retracts to the remaining graph \(H'\). Let \(C^1_i, \ldots, C^p_i\) and \(D^1_j, \ldots, D^q_j\) be the components of \(G'\) and \(H'\). The algorithm constructs the bipartite graph \(B_{ij}\) on the components \(C^k_i\) and \(D^\ell_j\), where \(k \in \{1, \ldots, p\}\) and \(\ell \in \{1, \ldots, q\}\). The algorithm checks if there is an edge \((C^k_i, D^\ell_j) \in E(B_{ij})\) recursively in \(O((|C^k_i| + |D^\ell_j|)^{5/2})\). And so, the bipartite graph \(B_{ij}\) is constructed in \(O(|C_i| + |D_j|)^{5/2})\) time. Edmond’s algorithm [8] computes a maximum matching in \(O((p+q)^{5/2}) = O(|C_i| + |D_j|)^{5/2})\) time.

Summing over the components \(C_i\) and \(D_j\), for \(i \in \{1, \ldots, t\}\) and \(j \in \{1, \ldots, s\}\), we obtain

\[
\sum_{i=1}^t \sum_{j=1}^s (|C_i| + |D_j|)^{5/2} = O(|V(G)| + |V(H)|)^{5/2}.
\]

This proves the claim.

\[ \square \]

6 Absolute retracts for cographs

Let \(G\) and \(H\) be connected graphs and \(\phi : V(G) \to V(H)\) a map. The map \(\phi\) is an isometric embedding if \(d_H(\phi(u), \phi(v)) = d_G(u, v)\) for any two vertices \(u\) and \(v\) of \(G\), where \(d_G(u, v)\) is the distance between \(u\) and \(v\) in \(G\) [17].

**Definition 6.1** Let \(\mathcal{G}\) be a class of graphs. A graph \(H\) is an absolute retract for \(\mathcal{G}\) if \(H\) is a retract of a graph \(G \in \mathcal{G}\) whenever \(G\) is an isometric embedding of \(H\) and \(\chi(H) = \chi(G)\).

Hell, in his PhD thesis, characterized absolute retracts for the class of bipartite graphs as the retracts of components of categorical products of paths [19]. Pesch and

\[ \copyright \ Springer \]
Poguntke characterized absolute retracts of $k$-chromatic graphs [29]. Their characterization can be strengthened for the case of bipartite graphs such that it leads to a polynomial recognition algorithm for absolute retracts of bipartite graphs [2]. Examples of absolute retracts of bipartite graphs are the chordal bipartite graphs [13]. Median graphs are exactly the absolute retracts of hypercubes [1]. For reasons of brevity we leave out the mention of all results on reflexive graphs.

**Theorem 6.2** Let $H$ be a connected cograph. Then $H$ is an absolute retract for the class of cographs if and only if every vertex of $H$ is in a maximal clique of cardinality $\omega(H)$.

**Proof** First notice that a cograph $G$ is an isometric embedding of a connected cograph $H$ if and only if $H$ is an induced subgraph of $G$. This follows since connected cographs have diameter two or, also, because they are distance hereditary. Thus, by the definition of distance-hereditary graphs [22], one distance-hereditary graph $G$ is an isometric embedding of another (connected) distance-hereditary graph $H$ if and only if $H$ is an induced subgraph of $G$.

Let $H$ be a connected cograph. Write $\omega = \omega(H)$ and assume that every vertex of $H$ is in a clique of cardinality $\omega$. Let $G$ be a cograph with $\omega(G) = \omega$ such that $H$ is an induced subgraph of $G$.

First assume that $G$ is disconnected. Then the vertices of $H$ are contained in one component of $G$ since $H$ is connected. If $W$ is any other component, then $G[W]$ has clique number at most $\omega$ and so there is a homomorphism from this component to the component that contains $H$. In other words, $H$ is a retract of $G$ if and only if $H$ is a retract of the component that contains $H$ as an induced subgraph. Henceforth, we may assume that $G$ is connected.

Consider a cotree for $G$. Since $G$ is connected the root is an $\otimes$-node. Let $C_1, \ldots, C_t$ be the cocomponents of $G$. Since $H$ is an induced subgraph with the same clique number as $G$, $H$ decomposes into the same number of cocomponents $D_1, \ldots, D_t$.

Notice that

$$\omega = \sum_{i=1}^{t} \omega(G[C_i]) = \sum_{i=1}^{t} \omega(H[D_i]).$$

Therefore, since $\omega(H[D_i]) \leq \omega(G[C_i])$, we have equality for each $i$, that is,

$$\omega(H[D_i]) = \omega(G[C_i]) \quad \text{for } i \in \{1, \ldots, t\}.$$ 

Now consider a $\oplus$-node. Let $C'_1, \ldots, C'_\ell$ be the sets of vertices of the subgraphs of $G$ induced by the children. Let

$$\omega' = \max\{\omega(G[C'_i])|i \in \{1, \ldots, \ell\}\}.$$ 

Write

$$D'_i = V(H) \cap C'_i \quad \text{for } i \in \{1, \ldots, \ell\}.$$
Since \( \omega(G) = \omega(H) \), we have that there is at least one component \( C'_i \) such that
\[
\omega' = \omega(H[D'_i])
\]

For \( G \) to retract to \( H \) we must have that, for every \( j \in \{1, \ldots, \ell\} \),
\[
D'_j \neq \emptyset \quad \text{implies} \quad \omega(H[D'_j]) = \omega(G[C'_j]).
\]

This condition is satisfied by virtue of the condition that every vertex of \( H \) is in a clique of cardinality \( \omega \). Namely, this implies that for every \( j \in \{1, \ldots, \ell\} \),
\[
D'_j \neq \emptyset \quad \text{implies} \quad \omega(H[D'_j]) = \omega'.
\]

Notice that this condition is necessary for \( H \) to be an absolute retract. This can be seen as follows. If there were a component \( D_j \) with
\[
\omega(H[D'_j]) < \max\{\omega(H[D'_i])|i \in \{1, \ldots, \ell\}\}
\]
then we could construct a cograph \( G \) such that \( H \) is an induced subgraph of \( G \) with \( \omega(G) = \omega(H) \) but such that \( G \) does not contract to \( H \). Namely, add one vertex to a component \( D'_j \) satisfying (3) as a true twin of a vertex which is in a maximum clique of \( H[D'_j] \).

This proves the theorem. \( \square \)

7 Foldings

**Definition 7.1** Let \( G = (V, E) \) be a graph and let \( x \) and \( y \) be two vertices in \( G \) that are at distance two. A simple fold with respect to \( x \) and \( y \) is the operation which identifies \( x \) and \( y \). A folding is a homomorphism which is a sequence of simple folds.

When \( G \to H \) is a folding then we say that \( G \) folds onto \( H \).

It is well-known that any retraction is a folding, see e.g., [17, Proposition 2.19].

**Definition 7.2** The folding number \( \Sigma_1(G) \) of a connected graph \( G \) is the largest number \( s \) such that \( G \) folds onto \( K_s \). When \( G \) is disconnected the folding number is the maximal folding number of the graphs induced by the components of \( G \).

The achromatic number \( \Psi(G) \) of a graph \( G \) is the largest number of colors with which one can properly color the vertices of \( G \) such that for any two colors there are two adjacent vertices that have those colors.

**Lemma 7.3** Assume that \( G \) has a universal vertex \( u \). Then
\[
\Sigma(G) = 1 + \Psi(G - u) = \Psi(G).
\]

**Proof** Any two nonadjacent vertices of \( G - u \) are at distance two in \( G \). Thus any achromatic coloring of \( G \) is a folding. The universal vertex must be in a color class
by itself. Harary and Hedetniemi [18] show that, when $G$ is the join of two graphs $G_1$ and $G_2$ then $\Psi(G) = \Psi(G_1) + \Psi(G_2)$. The proves the lemma. \hfill \Box

The achromatic number problem is NP-complete, even for trees [3]. However, the achromatic number problem is fixed-parameter tractable [27]. The image of a tree after a simple fold is a tree. Therefore, the folding number of a tree is at most two.

**Theorem 7.4** The problem to compute the folding number is NP-complete, even when restricted to trivially perfect graphs.

**Proof** Bodlaender shows in [3] that computing the achromatic number is NP-complete, even when restricted to trivially perfect graphs. Since the class of trivially perfect graphs is closed under adding universal vertices, by Lemma 7.3 computing the folding number is NP-complete for trivially perfect graphs. \hfill \Box

**Theorem 7.5** When $G$ is a threshold graph then

$$\chi(G) = \Sigma(G) = \Psi(G).$$

**Proof** When $G$ is the join of two graphs $G_1$ and $G_2$ then

$$\Psi(G) = \Psi(G_1) + \Psi(G_2).$$

Assume that $G$ has an isolated vertex $x$. In any achromatic coloring, the vertex must have a color that is used by another vertex also. Therefore,

$$\Psi(G) = \max\{1, \Psi(G - x)\}.$$

This proves the theorem. \hfill \Box

8 Concluding remarks

We proved that the retract problem for cographs is fixed-parameter tractable when parameterized by the number of vertices in the smaller of the two graphs. Perhaps more challenging and useful would be the cleaning-parameter, i.e., the difference between the number of vertices of the two graphs, recently introduced by Marx and Schlotter [25,26]. This parameterization was investigated for the induced subgraph isomorphism problem, when restricted to various classes of graphs, e.g., interval graphs, trees, planar graphs, and grids. As far as we know, whether cographs can be cleaned by a fixed-parameter algorithm is an open problem. Another interesting problem that we leave open is whether computing the folding number is fixed-parameter tractable.

**Acknowledgements** The authors would like to thank anonymous referees for their careful reading with corrections and useful comments which helped to improve the paper.
References

1. Bandelt, H.: Retracts of hypercubes. J. Graph Theory 8, 501–510 (1984)
2. Bandelt, H., Dählmann, A., Schütte, H.: Absolute retracts of bipartite graphs. Discrete Appl. Math. 16, 191–215 (1987)
3. Bodlaender, H.: Achromatic number is NP-complete for cographs and interval graphs. Inf. Process. Lett. 31, 135–138 (1989)
4. Chung, M.: \( O(n^{2.5}) \) time algorithms for the subgraph homomorphism problem on trees. J. Algorithms 8, 106–112 (1987)
5. Corneil, D., Perl, Y., Stewart, L.: A linear recognition algorithm for cographs. SIAM J. Comput. 14, 926–934 (1985)
6. Courcelle, B., Engelfriet, J.: Graph Structure and Monadic Second-Order Logic: A Language Theoretic Approach. In: Encyclopedia of mathematics and its applications, vol. 138. Cambridge University Press, Cambridge (2012)
7. Damaschke, P.: Induced subgraph isomorphism for cographs is NP-complete. In: Proceedings of 16th WG’90. Springer, LNCS 484, pp. 72–78 (1991)
8. Edmonds, J.: Paths, trees, and flowers. Can. J. Math. 17, 449–467 (1965)
9. Feder, T., Hell, P., Huang, J.: List homomorphisms and circular arc graphs. Combinatorica 19, 487–505 (1999)
10. Fomin, F., Heggernes, P., Kratsch, D.: Exact algorithms for graph homomorphisms. Theory Comput. Syst. 41, 381–393 (2007)
11. Golovach, P., Lidický, B., Martin, B., Paulusma, D.: Finding vertex-surjective graph homomorphisms. Acta Inf. 49, 381–394 (2012)
12. Golumbic, M.: Trivially perfect graphs. Discrete Math. 24, 105–107 (1978)
13. Golumbic, M., Goss, C.: Perfect elimination and chordal bipartite graphs. J. Graph Theory 2, 155–163 (1978)
14. Gótz, M., Koch, C., Martens, W.: Efficient algorithms for the tree homeomorphism problem. In: Proceedings of the 11th International Conference on Database Programming Languages, Springer, LNCS 4797, pp. 17–31 (2007)
15. Grohe, M.: The complexity of homomorphism and constraint satisfaction problems seen from the other side. J. ACM 54(1). Article no. 1 (2007). doi: 10.1145/1206035.1206036
16. Grohe, M.: Personal communication (2012)
17. Hahn, G., Tardif, C.: Graph homomorphisms: structure and symmetry. In: Hahn, G., Sabidussi, G. (eds.) Graph symmetry—Algebraic Methods and Applications, NATO ASI Series C: Mathematical and Physical Sciences, vol. 497, pp. 107–166, Kluwer, The Netherland (1997)
18. Harary, F., Hedetniemi, S.: The achronatic number of a graph. J. Comb. Theory 8, 154–161 (1970)
19. Hell, P.: Rétractions de graphes. Ph.D. Thesis, Université de Montréal (1972)
20. Hell, P., Nešetřil, J.: The core of a graph. Discrete Math. 109, 117–126 (1992)
21. Hell, P., Nešetřil, J.: Graphs Homomorph. Oxford University Press, Oxford (2004)
22. Howorka, E.: A characterization of distance-hereditary graphs. Q. J. Math. 28, 417–420 (1977)
23. Kilpeläinen, P., Mannila, H.: Quey primitives for tree-structured data. In: Proceedings of the 5th Annual Symposium on Combinatorial Pattern Matching, Springer, LNCS 807, pp. 213–225 (1994)
24. Kilpeläinen, P., Mannila, H.: Ordered and unordered tree inclusion. SIAM J. Comput. 24, 340–356 (1995)
25. Marx, D., Schlotter, I.: Parameterized graph cleaning problems. Discrete Appl. Math. 157, 3258–3267 (2009)
26. Marx, D., Schlotter, I.: Cleaning interval graphs. Algorithmica 65, 275–316 (2013)
27. Máté, A.: A lower estimate for the achronatic number of irreducible graphs. Discrete Math. 33, 171–183 (1981)
28. Matoušek, Thomas, R.: On the complexity of finding iso- and other morphisms for partial \( k \)-trees. Discrete Math. 108, 343–364 (1992)
29. Pesch, E., Poguntke, W.: A characterization of absolute retracts of \( n \)-chromatic graphs. Discrete Math. 57, 99–104 (1985)
30. Pinter, R., Rokhlenko, O., Tsur, D., Ziv-Ukelson, M.: Approximate labelled subtree homeomorphism. J. Discrete Algorithms 6, 480–496 (2008)
31. Reyner, S.: An analysis of good algorithms for the subtree problem. SIAM J. Comput. 6, 730–732 (1977)
32. Shamir, R., Tsur, D.: Faster subtree isomorphism. J. Algorithms 33, 267–280 (1999)
33. Sikora, F.: An (almost complete) state of the art around the graph motif problem. Universit’e Paris-Est, LIGM - UMR CNRS 8049, France (2012)
34. Wolk, E.: A note on “The comparability graph of a tree”. Proc. Am. Math. Soc. 16, 17–20 (1965)
35. Vikas, N.: A complete and equal computational complexity classification of compaction and retraction to all graphs with at most four vertices and some general results. J. Comput. Syst. Sci. 71, 406–439 (2005)