A complete and equal computational complexity classification of compaction and retraction to all graphs with at most four vertices and some general results

Narayan Vikas
School of Computing Science, Simon Fraser University, Burnaby, British Columbia, Canada V5A 1S6

Received 15 December 2003
Available online 23 May 2005

Abstract

A very close relationship between the compaction, retraction, and constraint satisfaction problems has been established earlier providing evidence that it is likely to be difficult to give a complete computational complexity classification of the compaction and retraction problems for reflexive or bipartite graphs. In this paper, we give a complete computational complexity classification of the compaction and retraction problems for all graphs (including partially reflexive graphs) with four or fewer vertices. The complexity classification of both the compaction and retraction problems is found to be the same for each of these graphs. This relates to a long-standing open problem concerning the equivalence of the compaction and retraction problems. The study of the compaction and retraction problems for graphs with at most four vertices has a special interest as it covers a popular open problem in relation to the general open problem. We also give complexity results for some general graphs. The compaction and retraction problems are special graph colouring problems, and can also be viewed as partition problems with certain properties. We describe some practical applications also.

© 2004 Elsevier Inc. All rights reserved.

Keywords: Computational complexity; Graph; Colouring; Homomorphism; Retraction; Compaction; Partition

A version presenting the results of this paper appears in Proceedings of the Tenth Annual International Computing and Combinatorics Conference (COCOON), 2004.

E-mail address: vikas@cs.sfu.ca.
Thus reflexive and irreflexive graphs are special partially reflexive graphs. A bipartite graph $G$ in a graph if $G_A$ containing $v$ (and $d_G(v, u)$ other vertex $u$ in the graph, for all $i$; we may write such a cycle as $v_0v_1v_2...v_k$, where the vertex $v_0$ is called the origin or the first vertex of the path, the vertex $v_{k-1}$ is called the terminus or the last vertex of the path, and the vertices $v_1, v_2, ..., v_{k-1}$ are called the internal vertices of the path. A cycle of length $k$, called a $k$-cycle, is a graph containing $k$ distinct vertices, say $v_0, v_1, v_2, ..., v_k$, such that $v_0v_1, v_1v_2, ..., v_{k-2}v_{k-1}$ are all the non-loop edges of the graph, $k \geq 3$; we may write such a cycle as $v_0v_1v_2...v_{k-1}v_0$. A square will be used as a synonym for a 4-cycle. A triangle will be used as a synonym for a 3-cycle. A walk of length $n$ in a graph is a sequence of vertices $v_0v_1v_2...v_n$ not necessarily distinct such that $v_i v_{i+1}$ is an edge of the graph, for all $i = 0, 1, 2, ... , n-1, n \geq 0$; we say that such a walk is from $v_0$ to $v_n$. For a graph $G$, we use $V(G)$ and $E(G)$ to denote the vertex set and the edge set of $G$ respectively. The size of a graph is the number of vertices plus the number of edges in the graph. We define min $S$ and max $S$ to give the minimum and the maximum element respectively in a set $S$. When a set $S$ is an argument of a mapping $f$, we define $f(S) = \{ f(s) | s \in S \}$. If a set has only one vertex, we may just write the vertex instead of the set.

Let $G$ be a graph. A vertex $v$ of $G$ is said to be an isolated vertex of $G$, if $v$ is not adjacent to any other vertex $v'$ of $G$, $v \neq v'$ (note that an isolated vertex may have a loop). Two vertices $u$ and $v$ of $G$ are said to be connected in $G$, if there exists a path from $u$ to $v$ in $G$; otherwise $u$ and $v$ are said to be disconnected in $G$. The distance between a pair of vertices $u$ and $v$ in $G$, denoted as $d_G(u, v)$ or $d_G(v, u)$, is the length of a shortest path from $u$ to $v$ in $G$, if $u$ and $v$ are connected in $G$; we define $d_G(u, v)$ (and $d_G(v, u)$) to be infinite, if $u$ and $v$ are disconnected in $G$. The diameter of $G$ is the maximum
distance between any pair of vertices in \( G \). The distance between two sets \( X \) and \( Y \) of vertices in \( G \), denoted as \( d_G(X, Y) \) or \( d_G(Y, X) \), is the minimum distance between any vertex of \( X \) and any vertex of \( Y \) in \( G \), that is, \( d_G(X, Y) = \min \{ d_G(x, y) \mid x \in X, y \in Y \} \). A graph \( H \) is said to be a subgraph of \( G \) if \( V(H) \subseteq V(G) \) and \( E(H) \subseteq E(G) \). We say that \( G \) is connected, if every pair of vertices in \( G \) is connected; otherwise we say that \( G \) is disconnected. A component of \( G \) is a maximal connected subgraph of \( G \). Clearly, \( G \) is a disjoint union of its components; in particular, if \( G \) has only one component then \( G \) is connected. A tree is a connected graph containing no cycles. A forest is a graph each of whose component is a tree. If \( H \) is a subgraph of \( G \) such that \( H \) contains all the edges of \( G \) whose both endpoints are in \( V(H) \) then \( H \) is called the subgraph of \( G \) induced by \( V(H) \), and we say that \( H \) is an induced subgraph of \( G \). Given an induced subgraph \( H \) of \( G \), we denote by \( G - H \), the subgraph obtained by deleting from \( G \) the vertices of \( H \) together with the edges incident with them; thus \( G - H \) is a subgraph of \( G \) induced by \( V(G) - V(H) \). For a vertex \( v \) of \( G \), we define \( G - v \) similarly (in the above, we have a single vertex \( v \) instead of the graph \( H \)).

The vertices in a set \( I \subseteq V(G) \) are said to be independent if there is no edge in the subgraph of \( G \) induced by \( I \). A chordal graph is a graph which does not contain any induced cycle of length greater than three. A chordal bipartite graph is a bipartite graph which does not contain any induced cycle of length greater than four. In the following, let \( G \) and \( H \) be graphs.

A homomorphism \( f : G \to H \), of \( G \) to \( H \), is a mapping \( f \) of the vertices of \( G \) to the vertices of \( H \), such that \( f(g) \) and \( f(g') \) are adjacent vertices of \( H \) whenever \( g \) and \( g' \) are adjacent vertices of \( G \). If there exists a homomorphism of \( G \) to \( H \), then \( G \) is said to be homomorphic to \( H \). Note that for any homomorphism \( f : G \to H \), if a vertex \( v \) of \( G \) has a loop then the vertex \( f(v) \) of \( H \) necessarily also has a loop. If \( G \) is irreflexive then clearly \( G \) is \( k \)-colourable if and only if \( G \) is homomorphic to \( K_k \). Thus the concept of a homomorphism generalises the concept of a \( k \)-colourability.

A compaction \( c : G \to H \), of \( G \) to \( H \), is a homomorphism of \( G \) to \( H \), such that for every vertex \( x \) of \( H \), there exists a vertex \( v \) of \( G \) with \( c(v) = x \), and for every edge \( hh' \) of \( H \), \( h \neq h' \), there exists an edge \( gg' \) of \( G \) with \( c(g) = h \) and \( c(g') = h' \). Notice that the first part of the definition for compaction (the requirement for every vertex \( x \) of \( H \)) is relevant only if \( H \) has isolated vertices. If there exists a compaction of \( G \) to \( H \), then \( G \) is said to be compact to \( H \). Given a compaction \( c : G \to H \), if for a vertex \( v \) of \( G \), we have \( c(v) = x \), where \( x \) is a vertex of \( H \), then we say that the vertex \( v \) of \( G \) covers the vertex \( x \) of \( H \) under \( c \); and if for an edge \( gg' \) of \( G \), we have \( c([g, g']) = [h, h'] \), where \( hh' \) is an edge of \( H \), then we say that the edge \( gg' \) of \( G \) covers the edge \( hh' \) of \( H \) under \( c \) (note that in the definition of compaction, it is not necessary that a loop of \( H \) be covered by any edge of \( G \) under \( c \)).

We note that the notion of a homomorphic image used by Harary [12] (also cf. Hell and Miller [15]) coincides with the notion of a compaction in the case of irreflexive graphs (i.e., when \( G \) and \( H \) are irreflexive in the above definition for compaction).

A retraction \( r : G \to H \), of \( G \) to \( H \), with \( H \) as an induced subgraph of \( G \), is a homomorphism of \( G \) to \( H \), such that \( r(h) = h \), for every vertex \( h \) of \( H \). If there exists a retraction of \( G \) to \( H \) then \( G \) is said to retract to \( H \), and \( H \) is said to be a retract of \( G \). Note that every retraction \( r : G \to H \) is necessarily also a compaction but not vice versa.

An identification of two distinct vertices \( u \) and \( v \) of \( G \) is an execution of the following steps (1), (2), and (3), resulting in a new graph: (1) For every non-loop edge \( uu' \) of \( G \), if \( uu' \) is not an edge of \( G \) then we add the edge \( vu' \) to \( G \) (note that if \( uv \) is an edge of \( G \) then \( uu' = v \) and we will have the loop \( vv \)). (2) If \( u \) has a loop then \( v \) is also made to have a loop if it does not already have one. (3) Delete the vertex \( u \) together with the edges incident with \( u \) from \( G \).
1.2. Homomorphism, compaction, and retraction problems

The problem of deciding the existence of a homomorphism to a fixed graph $H$, called the homomorphism problem for $H$, also known as the $H$-colouring problem, and denoted as $H$-$\text{COL}$, asks whether or not an input graph $G$ is homomorphic to $H$. If $H$ is a graph with a loop then every graph is trivially homomorphic to $H$. Also, if $H$ is a bipartite graph then we note that a graph $G$ is homomorphic to $H$ if and only if $G$ is also bipartite and $H$ has an edge if $G$ has an edge. Thus the problem $H$-$\text{COL}$ is interesting only if $H$ is an irreflexive non-bipartite graph. A complete complexity classification of $H$-$\text{COL}$ is given by Hell and Nesetril [16]. It is shown by them that $H$-$\text{COL}$ is NP-complete for any fixed irreflexive non-bipartite graph $H$. As pointed above, $H$-$\text{COL}$ is polynomial time solvable otherwise. Note that the classic $k$-colourability problem is a special case of the problem $H$-$\text{COL}$ when $H$ is $K_k$ and the input graph $G$ is irreflexive.

The problem of deciding the existence of a compaction to a fixed graph $H$, called the compaction problem for $H$, and denoted as $\text{COMP-H}$, asks whether or not an input graph $G$ compacts to $H$. The problem $\text{COMP-H}$, in general, where $H$ is a fixed partially reflexive graph, can be viewed as the problem to decide whether or not it is possible to partition the vertices of a graph into certain fixed number of distinct non-empty sets such that there is at least one edge between some pair of distinct sets, and there is no edge between all other pair of distinct sets, and certain sets may be required to be independent (an independent set has no edge), where the sets and edges correspond to the vertices and edges of $H$, and an independent set in particular correspond to a vertex without a loop in $H$.

When both $G$ and $H$ are input graphs (i.e., $H$ is not fixed), and $H$ is reflexive, the problem of deciding whether or not $G$ compacts to $H$ has been studied by Karabeg and Karabeg [18,19]. Some related work has recently been studied by Feder, Hell, Klein, and Motwani [6,7]. Note that unlike the problem $H$-$\text{COL}$, the problem $\text{COMP-H}$ is still interesting if $H$ has a loop or $H$ is bipartite.

The problem of deciding the existence of a retraction to a fixed graph $H$, called the retraction problem for $H$, and denoted as $\text{RET-H}$, asks whether or not an input graph $G$, containing $H$ as an induced subgraph, retracts to $H$. Note that $H$ is a particular copy in $G$. It is possible that $G$ contains another induced subgraph $H'$ that is isomorphic to $H$ but distinct from $H$, and it may be the case that $G$ retracts to $H'$ but not to $H$, and vice versa.

The problem $\text{RET-H}$ can again be viewed as the partition problem, as described above, with the restriction that each vertex of $H$ is in a distinct set of partition. Retraction problems have been of continuing interest in graph theory for a long time and have been studied in various literature including [1,2,4,5,10,13,14,17,22–24].

Note that the graph $H$ for the problems $H$-$\text{COL}$, $\text{COMP-H}$, and $\text{RET-H}$ is assumed to be fixed by default even if not explicitly mentioned.

1.3. Motivation and results

It is not difficult to show that for every fixed graph $H$, if $\text{RET-H}$ is solvable in polynomial time then $\text{COMP-H}$ is also solvable in polynomial time (a polynomial transformation from $\text{COMP-H}$ to $\text{RET-H}$ under Turing reduction is shown by Vikas [28]). Is the converse true? This was also asked, in the context of reflexive graphs, by Peter Winkler in 1988 (personal communication, cf. [10]). Thus the question is whether $\text{RET-H}$ and $\text{COMP-H}$ are polynomially equivalent for every fixed graph $H$. The answer to this is not known even when $H$ is reflexive or bipartite. However, it is shown by Vikas [28] that for every
fixed reflexive (bipartite) graph $H$ there exists a fixed reflexive (bipartite) graph $H'$ such that $RET-H$ and $COMP-H'$ are polynomially equivalent.

Using the above result of Vikas [28], and results of Feder and Hell [4] and Feder and Vardi [9], it is established by Vikas [28] that for every constraint satisfaction problem $II$ (with fixed templates), there exists a fixed reflexive (bipartite) graph $H$ such that the constraint satisfaction problem $II$ and the compaction problem $COMP-H$ are polynomially equivalent. Since it is thought to be likely difficult to determine whether every constraint satisfaction problem (with fixed templates) is polynomial time solvable or NP-complete, thus evidence is provided by Vikas [28] that it is likely to be difficult to determine whether for every fixed reflexive (bipartite) graph $H$, the problem $COMP-H$ is polynomial time solvable or NP-complete. Similar evidence has been shown for $RET-H$ by Feder and Hell [4] in the case of fixed reflexive graphs $H$, and by Feder and Vardi [9] in the case of fixed bipartite graphs $H$. Issues related to the constraint satisfaction problem have also been considered by Feder and Vardi [8,9].

We however give in this paper, a complete complexity classification of $COMP-H$ and $RET-H$ when $H$ has four or fewer vertices, i.e., for every graph $H$ with at most four vertices (including when $H$ is partially reflexive), we determine whether $COMP-H$ is polynomial time solvable or NP-complete, and whether $RET-H$ is polynomial time solvable or NP-complete. We find that the complexity classification of $COMP-H$ and $RET-H$ do not differ for such graphs $H$. Studying the complexity classification of the compaction and retraction problems for graphs with at most four vertices has an additional significance as it includes a widely publicised open problem posed by Peter Winkler in 1988 to determine the complexity of $COMP-H$ when $H$ is a reflexive 4-cycle. This has been shown to be NP-complete by Vikas [25,26]. The problem was asked in relation to the general problem mentioned earlier concerning the equivalence of the compaction and retraction problems, as the unique smallest reflexive graph $H$ for which $RET-H$ is NP-complete turned out to be a reflexive 4-cycle.

Thus our study in this paper is motivated by two issues. One issue is concerned with the complete complexity classification of the compaction and retraction problems, and the other issue is concerned with the equivalence of the compaction and retraction problems. We present results in this paper resolving fully the two issues for graphs up to four vertices. We hope that the techniques and constructions developed in this paper and the papers [25–28] would be helpful in resolving the bigger problem whether the compaction and retraction problems are equivalent for all graphs.

We have more results showing that for several graphs $H$, the problems $RET-H$ and $COMP-H$ are polynomially equivalent. We mention below a few classes of such graphs. We do not know of any graph $H$ for which the complexity classification of $RET-H$ and $COMP-H$ differ.

It is known that $RET-H$ is NP-complete when $H$ is a reflexive $k$-cycle, for all $k \geq 4$, cf. Feder and Hell [4], G. MacGillivray, 1988 (personal communication), and for $k = 4$, also Feder and Winkler [10]. It is shown by Vikas [25,26] that $COMP-H$ is NP-complete when $H$ is a reflexive $k$-cycle, for all $k \geq 4$. In particular, as mentioned above, for $k = 4$, this result of Vikas [25,26] solves a widely publicised open problem posed by Peter Winkler in 1988. When $H$ is a reflexive chordal graph (which includes a reflexive 3-cycle), the problem $RET-H$ is shown to be polynomial time solvable by Feder and Hell [4], and hence $COMP-H$ is also polynomial time solvable.

It is also known that $RET-H$ is NP-complete when $H$ is an irreflexive even $k$-cycle, for all even $k \geq 6$, cf. Feder, Hell, and Huang [5], G. MacGillivray, 1988 (personal communication). It is shown by Vikas [25,27] that $COMP-H$ is NP-complete when $H$ is an irreflexive even $k$-cycle, for all even $k \geq 6$. This result of Vikas [25,27] also solves a long-standing problem that has been of interest since about 1988 to various people including Pavol Hell and Jaroslav Nesetril (personal communications). When $H$ is a chordal
bipartite graph (which includes an irreflexive 4-cycle), the problem $RET-H$ is shown to be polynomial time solvable by Bandelt, Dahlmann, and Schutte [1], and hence $COMP-H$ is also polynomial time solvable.

The case of irreflexive odd cycles is a special case of a more general result whereby $RET-H$ and $COMP-H$ are polynomially equivalent for every non-bipartite irreflexive graph $H$. Note that a graph $G$ is homomorphic to a graph $H$ if and only if the disjoint union $G \cup H$ retracts/compacts to $H$. Thus we have a polynomial transformation from $H-COL$ to $RET-H$ and $COMP-H$. As mentioned earlier, the problem $H-COL$ is NP-complete for any non-bipartite irreflexive graph $H$ due to Hell and Nesetril [16]. It follows that $RET-H$ and $COMP-H$ are also NP-complete for any non-bipartite irreflexive graph $H$; in particular if $H$ is an irreflexive odd $k$-cycle then $RET-H$ and $COMP-H$ are NP-complete, for all odd $k \geq 3$.

Thus we conclude that when $H$ is an irreflexive $k$-cycle, $RET-H$ and $COMP-H$ both are NP-complete, for all $k \geq 3$, $k \neq 4$, and polynomial time solvable, for $k = 4$.

The problem $RET-H$ for some partially reflexive graphs $H$ has been studied by Feder, Hell, and Huang, cf. [5]. Let $H$ be a graph and let $V_L$ denote its set of vertices which have loops. It is shown by Feder, Hell, and Huang (personal communication, cf. [5]) that if $H$ is connected but $V_L$ is not then $RET-H$ is NP-complete. It is also shown by them (personal communication, cf. [5]) that if $V_L$ is connected and $H$ is a tree then $RET-H$ is polynomial time solvable.

With regards to more general graphs (not necessarily with at most four vertices), we show in this paper that if $H$ is a path of length $k \geq 2$, with loops on the first (origin) and the last (terminus) vertices only, then $COMP-H$ is NP-complete ($RET-H$ is also NP-complete as noted above). Using this result, we prove in this paper NP-completeness of $COMP-H$ for some other general partially reflexive graphs $H$ also (again $RET-H$ is NP-complete for these graphs $H$ also due to the above result of Feder, Hell, and Huang, cf. [5]).

We also prove in this paper various results for the compaction and retraction problems in relation to connected and disconnected general graphs. These results are useful in determining our complexity classification.

In the figures in this paper, we shall not be depicting any edge $vh$ of $G$, with $v \in V(G - H)$ and $h \in V(H)$, where $G$ is any graph containing $H$ as an induced subgraph, i.e., $G$ is an instance of $RET-H$.

We first give in Section 2, some general results for the compaction and retraction problems for connected and disconnected graphs. We need these results in Section 3, where we give a complete complexity classification of $COMP-H$ as well as $RET-H$ when $H$ has four or fewer vertices. In Section 4, we present our results on partially reflexive paths of length $k \geq 2$, with loops only on the origin and terminus vertices, and we also present results on some other general graphs. In Section 5, we describe some applications of compaction.

## 2. Compaction and retraction to connected and disconnected graphs

In this section, we show the relationship between compaction to connected and disconnected graphs, and the relationship between retraction to connected and disconnected graphs. These results will show that if $H$ is disconnected then the complexity of $COMP-H$ and $RET-H$ can be deduced directly from the complexity of the compaction and retraction problems, respectively, for the connected components of $H$. Hence it would be sufficient to consider only connected graphs $H$ when determining the complexity of $COMP-H$ or $RET-H$ in the next section.
Theorem 2.1. Let $H$ be a graph with components $H_1, H_2, \ldots, H_s$. Suppose that RET-$H_i$ is polynomial time solvable, for all $i = 1, 2, \ldots, s$. Then RET-$H$ is also polynomial time solvable.

Proof. Let a graph $G$ containing $H$ as an induced subgraph be an instance of RET-$H$. Clearly, if two components of $H$ are subgraphs of the same component of $G$ then $G$ does not retract to $H$. Now suppose that no two components of $H$ are subgraphs of the same component of $G$. Hence $G$ has at least $s$ components. Let $G_1, G_2, \ldots, G_t$, with $t \geq s$, be the components of $G$ with $G_i$ containing $H_i$ as an induced subgraph, for all $i = 1, 2, \ldots, s$. Clearly, $G$ retracts to $H$ if and only if $G_i$ retracts to $H_i$, for all $i = 1, 2, \ldots, s$, and $G_j$ is homomorphic to $H$, for all $j = s + 1, s + 2, \ldots, t$. Note that we can test whether $G_j$ is homomorphic to $H$, $s + 1 \leq j \leq t$, by testing whether the disjoint union $G_j \cup H_k$ retracts to $H_k$, for some $k$, $1 \leq k \leq s$.

Theorem 2.2. Let $H$ be a fixed graph with components $H_1, H_2, \ldots, H_s$. Suppose that COMP-$H_i$ is polynomial time solvable, for all $i = 1, 2, \ldots, s$. Then COMP-$H$ is also polynomial time solvable.

Proof. Let a graph $G$ be an instance of COMP-$H$. We construct in time polynomial in the size of $G$, graphs $G_1, G_2, \ldots, G_{\beta}$, each containing a copy of $H$ as an induced subgraph, such that $G$ compacts to $H$ if and only if there exists a graph $G_i, 1 \leq i \leq \beta$, which retracts to $H$ (note that we can only remark that such a construction exists and are constructed when giving a polynomial transformation from COMP-$H$ to RET-$H$ (under Turing reduction), see Vikas [28] for the construction.

In Vikas [28], each graph $G_i$ is constructed such that no two components of $H$ are subgraphs of the same component of $G_i$, as a necessary condition for $G_i$ to retract to $H$ (and hence $G_i$ has at least $s$ components), $1 \leq i \leq \beta$. Let $G_{i,1}, G_{i,2}, \ldots, G_{i,\eta_i}, \eta_i \geq s$, be the components of $G_i$, with $G_{i,j}$ containing a copy of $H_j$ as an induced subgraph, for all $j = 1, 2, \ldots, s, i = 1, 2, \ldots, \beta$. We have that $G_i$ retracts to $H$ if and only if $G_{i,j}$ retracts to $H_j$, for all $j = 1, 2, \ldots, s, i = 1, 2, \ldots, \beta$. Thus we have that $G$ compacts to $H$ if and only if there exists an $i, 1 \leq i \leq \beta$, such that $G_{i,j}$ retracts to $H_j$, for all $j = 1, 2, \ldots, s, \beta$. We have that $G_i$ retracts to $H$ if and only if $G_{i,k}$ is homomorphic to $H$, for all $k = s + 1, s + 2, \ldots, t_i, i = 1, 2, \ldots, \beta$. Thus we have that $G$ compacts to $H$ if and only if there exists an $i, 1 \leq i \leq \beta$, such that $G_{i,j}$ retracts to $H_j$, for all $j = 1, 2, \ldots, s, \beta$. We have that $G_i$ retracts to $H$ if and only if $G_{i,k}$ is homomorphic to $H$, for all $k = s + 1, s + 2, \ldots, t_i$. Recall that every retraction is also a compaction. Also, in Vikas [28], each graph $G_i$ is obtained as a result of identification of certain vertices of $G$, and hence $G$ compacts to $G_i$, $1 \leq i \leq \beta$. It follows that $G$ compacts to $H$ if and only if there exists an $i, 1 \leq i \leq \beta$, such that $G_{i,j}$ compacts to $H_j$, for all $j = 1, 2, \ldots, s, \beta$. It follows that $G$ compacts to $H$ if and only if there exists an $i, 1 \leq i \leq \beta$, such that $G_{i,j}$ compacts to $H_j$, for all $j = 1, 2, \ldots, s, \beta$. It follows that $G$ retracts to $H$ if and only if there exists an $i, 1 \leq i \leq \beta$, such that $G_{i,j}$ retracts to $H_j$, for all $j = 1, 2, \ldots, s, \beta$. It follows that $G$ retracts to $H$ if and only if there exists an $i, 1 \leq i \leq \beta$, such that $G_{i,j}$ retracts to $H_j$, for all $j = 1, 2, \ldots, s, \beta$.

Theorem 2.3. Let $H$ be a fixed connected graph. Let RETC-$H$ denote the problem RET-$H$ with instances restricted to connected graphs. Then the problems RETC-$H$ and RET-$H$ are polynomially equivalent.

Proof. Trivially, the problem RETC-$H$ polynomially transforms to the problem RET-$H$, as RETC-$H$ is just a restriction of RET-$H$.

Now we give a polynomial transformation from RET-$H$ to RETC-$H$. Let a graph $G$ containing $H$ as an induced subgraph be an instance of RET-$H$. Let $G_1, G_2, \ldots, G_t$ be the components of $G$, with $G_1$
Theorem 2.4. Let $H$ be a fixed connected graph. Let $\text{COMPC-H}$ denote the problem $\text{COMP-H}$ with either $e(x,h)$ or $G_i$ to $H$, for all $i = 2, 3, \ldots, t$. Select any one vertex $v_i$ from $G_i$, for all $i = 2, 3, \ldots, t$.

Suppose first that $H$ is a non-bipartite graph. Thus $H$ is either an irreflexive graph with an odd cycle, or a graph with a loop on at least one vertex. If $H$ has only one vertex then it must be the case that $H$ is reflexive, as $H$ is non-bipartite, and the theorem readily follows, as every input graph containing $H$ retracts to $H$. Now suppose that $H$ has more than one vertex. There exists a walk of even length as well as a walk of odd length between every pair of vertices (not necessarily distinct) in $H$, due to the presence of an odd cycle or a loop in $H$. Select any one vertex $x$ from $H$. If $G$ retracts to $H$ then we know that the vertex $v_i$ of $G_i$ maps (under a retraction of $G$ to $H$) to some vertex $y_i$ of $H$ such that there exists a walk of even length as well as a walk of odd length from $x$ to $y_i$ in $H$, for all $i = 2, 3, \ldots, t$. With this in mind, we construct in time polynomial in the size of $G$, a connected graph $G' \prime$ (containing $G$ as an induced subgraph) such that $G$ retracts to $H$ if and only if $G' \prime$ retracts to $H$.

Let $e(x,h)$ denote the length of a shortest walk of even length from $x$ to $h$ in $H$, with $h \in V(H)$. Let $ow(x,h)$ denote the length of a shortest walk of odd length from $x$ to $h$ in $H$, with $h \in V(H)$. Note that either $e(x,h)$ or $ow(x,h)$ is the distance between $x$ and $h$, with $h \in V(H)$. Since $H$ is fixed, we can compute such walks in a fixed time. Let $p$ be the maximum length of such even and odd walks from $x$ to any vertex of $H$, i.e., let $p = \max\{e(x,h), ow(x,h) | h \in V(H)\}$. Clearly, $p > 2$. For each $G_i$, $2 \leq i \leq t$, we add to $G$ a path $U_i = u_1^i u_2^i \ldots u_{p-1}^i$ containing $p - 1$ new vertices and we add the edges $xu_1^i$ and $v_i u_{p-1}^i$; thus we have the path $xU_i v_i$ of length $p$. This completes the construction of $G'$ which as seen is connected and $G$ is an induced subgraph of $G'$. Note that for any vertex $y$ of $H$ there exists a walk from $x$ to $y$ in $H$ of length $p$. It is not difficult to see that $G$ retracts to $H$ if and only if $G' \prime$ retracts to $H$. Thus we have a polynomial transformation from RET-$H$ to RETC-$H$.

Now suppose that $H$ is a bipartite graph. Since only a bipartite graph may possibly be homomorphic to $H$, we may assume that $G$ is also bipartite. If $H$ has only one vertex $h$ then we obtain a connected graph $G''$ as a result of identifying the vertices $h, v_2, v_3, \ldots, v_t$; trivially, $G$ retracts to $H$ if and only if $G''$ retracts to $H$. If $H$ has more than one vertex then clearly $G_2, G_3, \ldots, G_t$ are homomorphic to $H$, implying that $G$ retracts to $H$ if and only if $G_1$ retracts to $H$. Thus we have a polynomial transformation from RET-$H$ to RETC-$H$.

Theorem 2.4. Let $H$ be a fixed connected graph. Let $\text{COMPC-H}$ denote the problem $\text{COMP-H}$ with instances restricted to only connected graphs. The problems $\text{COMPC-H}$ and $\text{COMP-H}$ are polynomially equivalent under Turing reduction.

Proof. Trivially, the problem $\text{COMPC-H}$ polynomially transforms to the problem $\text{COMP-H}$, as $\text{COMPC-H}$ is a restriction of $\text{COMP-H}$.

We now give a polynomial transformation from $\text{COMP-H}$ to $\text{COMPC-H}$. Let a graph $G$ be an instance of $\text{COMP-H}$. As in the proof of Theorem 2.2, we construct in time polynomial in the size of $G$, graphs $G_1, G_2, \ldots, G_\beta$, each containing a copy of $H$ as an induced subgraph, such that $G$ compacts to $H$ if and only if there exists a graph $G_i$, $1 \leq i \leq \beta$, which retracts to $H$ ($\beta$ is a polynomial in the size of $G$).

Let $G_{i,1}, G_{i,2}, \ldots, G_{i,\beta}$ be the components of $G_i$, with $G_{i,1}$ containing a copy of $H$ as an induced subgraph, for all $i = 1, 2, \ldots, \beta$. As in the proof of Theorem 2.2, we have that $G$ compacts to $H$ if and only if there exists an $i$, $1 \leq i \leq \beta$, such that $G_{i,1}$ compacts to $H$, and $G_{i,j}$ is homomorphic to $H$, for all $j = 2, 3, \ldots, \beta$. 

Suppose first that $H$ is a non-bipartite graph. Thus $H$ is either an irreflexive graph with an odd cycle, or a graph with a loop on at least one vertex. As in the proof of Theorem 2.3, the case when $H$ has only one vertex is trivial (in this case $H$ must be reflexive, and every input graph compacts to $H$). Now suppose that $H$ has more than one vertex. Let $G_i$ denote the graph with components $G_{i,1}, G_{i,2}, G_{i,3}, \ldots, G_{i,t_i}$ (i.e., $G_i^*$ is the graph $G_i$ without $G_{i,1}$), for all $i = 1, 2, \ldots, \beta$. We construct in time polynomial in the size of $G_i^*$, a connected graph $G_i'$ (containing the components $G_{i,1}, G_{i,2}, G_{i,3}, \ldots, G_{i,t_i}$, and a copy of $H$ as induced subgraphs) such that $G_{i,2}, G_{i,3}, \ldots, G_{i,t_i}$ are homomorphic to $H$ if and only if $G_i'$ compacts to $H$, for all $i = 1, 2, \ldots, \beta$. The construction of $G_i'$, $1 \leq i \leq \beta$, is as follows. Let $H'$ be a copy of $H$. We first add $H'$ to $G_i^*$ as a component. Let $p = \max\{\text{ew}(h, h'), \text{ow}(h, h')|h, h' \in V(H)\}$, where $\text{ew}(h, h')$ and $\text{ow}(h, h')$ denote the length of a shortest walk of even length and odd length respectively from $h$ to $h'$, with $h, h' \in V(H)$ (cf. proof of Theorem 2.3). Let $x$ be a fixed vertex of $H'$, and $v_i, j$ be a fixed vertex of $G_{i,j}$, for all $j = 2, 3, \ldots, t_i$. For each $G_{i,j}$, $2 \leq j \leq t_i$, we add a path $x U_{i,j} v_{i,j}$ of length $p$, where $U_{i,j}$ is a path containing $p - 1$ new vertices. This completes the construction of $G_i'$. It is not difficult to see that $G_{i,2}, G_{i,3}, \ldots, G_{i,t_i}$ are homomorphic to $H'$ if and only if $G_i'$ compacts to $H'$, for all $i = 1, 2, \ldots, \beta$ (cf. proof of Theorem 2.3). It follows that $G$ compacts to $H$ if and only if there exists an $i, 1 \leq i \leq \beta$, such that $G_{i,1}$ compacts to $H$, and $G_i'$ compacts to $H$. Both $G_{i,1}$ and $G_i'$ are connected, for all $i = 1, 2, \ldots, \beta$. Thus we have a polynomial transformation from $\text{COMP-H}$ to $\text{COMPC-H}$ under Turing reduction.

Now suppose that $H$ is a bipartite graph. As in the proof of Theorem 2.3, we may again assume that $G$ is also bipartite. If $H$ has only one vertex then we choose a vertex from each component of $G$, and identify them resulting in a connected graph $G''$; trivially, $G$ compacts to $H$ if and only if $G''$ compacts to $H$. Now suppose that $H$ has more than one vertex. Then it is easy to see that $G_{i,2}, G_{i,3}, \ldots, G_{i,t_i}$ are homomorphic to $H$, for all $i = 1, 2, \ldots, \beta$. Hence $G$ compacts to $H$ if and only if there exists an $i, 1 \leq i \leq \beta$, such that $G_{i,1}$ compacts to $H$. Thus we have a polynomial transformation from $\text{COMP-H}$ to $\text{COMPC-H}$ under Turing reduction. \qed

**Theorem 2.5.** Let $H$ be a graph with components $H_1, H_2, \ldots, H_s$. Suppose that $\text{RET-H}_1$ is NP-complete for some $i, 1 \leq i \leq s$. Then $\text{RET-H}$ is also NP-complete.

**Proof.** Clearly, the problem $\text{RET-H}$ is in NP. Without loss of generality, suppose that $\text{RET-H}_1$ is NP-complete. We give a polynomial transformation from $\text{RET-H}_1$ to $\text{RET-H}$. Let a graph $G$ containing $H_1$ as an induced subgraph be an instance of $\text{RET-H}_1$. It follows from Theorem 2.3 that the problem $\text{RET-H}_1$ remains NP-complete with instances restricted to connected graphs. Hence we assume that $G$ is connected. Let $G'$ be a graph with components $G, H_2, H_3, \ldots, H_s$. Thus $H$ is an induced subgraph of $G'$. Clearly, $G$ retracts to $H_1$ if and only if $G'$ retracts to $H$. Thus $\text{RET-H}$ is NP-complete. \qed

**Theorem 2.6.** Let $H$ be a graph with components $H_1, H_2, \ldots, H_s$. Suppose that $\text{COMP-H}_i$ is NP-complete for some $i, 1 \leq i \leq s$. Then $\text{COMP-H}$ is also NP-complete.

**Proof.** Clearly, the problem $\text{COMP-H}$ is in NP. Without loss of generality, suppose that $\text{COMP-H}_1$ is NP-complete. We give a polynomial transformation from $\text{COMP-H}_1$ to $\text{COMP-H}$. Let a graph $G$ be an instance of $\text{COMP-H}_1$. We may assume that $G$ is connected, as it follows from Theorem 2.4 that the problem $\text{COMP-H}_1$ remains NP-complete with instances restricted to connected graphs. Let $G'$ be a
3. A complete complexity classification of compaction and retraction to all graphs with four or fewer vertices

In this section, we give a complete complexity classification of COMP-H and RET-H when H has four or fewer vertices. We shall see that the complexity classification of COMP-H and RET-H do not differ for such graphs H. We note here that for every graph H, the problems COMP-H and RET-H are in NP, and we will not be mentioning this explicitly in our proofs. First we point out below those general results (not restricted to graphs with four or fewer vertices) which we shall be using as part of our complete complexity classification, as they provide the complexity of COMP-H and RET-H for some of the graphs H that we need to consider.

Let H be a forest, with trees $H_1, H_2, \ldots, H_s$ such that the set $V_{L_i}$ of vertices of $H_i$ with loops is connected, for all $i = 1, 2, \ldots, s$. It is shown by Feder, Hell, and Huang (personal communication, cf. [5]) that RET-H is polynomial time solvable. Since COMP-H is polynomially transformable to RET-H (under Turing reduction), it follows that COMP-H is also solvable in polynomial time.

Now let H be a graph containing a vertex z adjacent to every vertex of H (including itself). This includes the case of a reflexive complete graph, and a reflexive 3-cycle in particular. Then every graph G containing H as an induced subgraph retracts to H, as the mapping $r : G \to H$, with $r(h) = h$, for all $h \in V(H)$, and $r(v) = z$, for all $v \in V(G) - V(H)$, is a retraction. Thus RET-H, and hence COMP-H, are solvable in polynomial time.

Let H be a bipartite graph with a non-empty edge set and bipartition $(H_A, H_B)$ such that there exists a vertex, say y, in $H_A$ adjacent to every vertex in $H_B$, and there exists a vertex, say z, in $H_B$ adjacent to every vertex in $H_A$. This includes the case of a complete bipartite graph with non-empty edge set, and an irreflexive 4-cycle in particular. Let G be any bipartite graph with bipartition $(G_A, G_B)$ containing H as an induced subgraph, with $H_A \subseteq G_A$ and $H_B \subseteq G_B$. Then G necessarily retracts to H, as the mapping $r : G \to H$, with $r(h) = h$, for all $h \in V(H)$, $r(a) = y$, and $r(b) = z$, for all $a \in G_A - H_A$, $b \in G_B - H_B$, is a retraction. Since only a bipartite graph may possibly be homomorphic to a bipartite graph, it follows that RET-H, and hence COMP-H, are polynomial time solvable. If H is a graph with an empty edge set, i.e., H has only isolated vertices with no loops, then we know that a graph G containing H as an induced subgraph retracts to H if and only if the edge set of G is also empty, and trivially RET-H, and hence COMP-H, are polynomial time solvable.
If \( H \) is an irreflexive non-bipartite graph (which includes an irreflexive odd \( k \)-cycle, for all odd \( k \geq 3 \)) then as seen in Section 1.3, it follows from the result of Hell and Nesetril [16] that \( RET-H \) and \( COMP-H \) both are NP-complete.

If \( H \) is a chordal bipartite graph (which includes an irreflexive 4-cycle), the problem \( RET-H \) is shown to be polynomial time solvable by Bandelt, Dahlmann, and Schutte [1], and hence \( COMP-H \) is also polynomial time solvable.

Let \( H \) be a reflexive \( k \)-cycle, for any \( k \geq 4 \). It is known that \( RET-H \) is NP-complete, cf. Feder and Hell [4], G. MacGillivray, 1988 (personal communication), and for \( k = 4 \), also Feder and Winkler [10]. It is shown by Vikas [25,26] that \( COMP-H \) is NP-complete. Thus, in particular, when \( H \) is a reflexive square, \( RET-H \) and \( COMP-H \) both are NP-complete.

If \( H \) is a reflexive chordal graph (which includes a reflexive 3-cycle), the problem \( RET-H \) is shown to be polynomial time solvable by Feder and Hell [4], and hence \( COMP-H \) is also polynomial time solvable.

When considering graphs \( H \) with four or fewer vertices, we will not be separately considering graphs \( H \) that fall in the above categories. Also, due to Theorems 2.1, 2.2, 2.5, and 2.6, we do not need to separately consider graphs \( H \) that are disconnected. For graphs \( H \) with four or fewer vertices, this leaves us to consider the graphs \( H \) in Fig. 1. We consider each of the graphs \( H \) in Fig. 1, and present the complexity results for \( COMP-H \) and \( RET-H \).

For the graphs \( H \) in Figs. 1(a)–(f), (j), (k), (m), and (p), the set of vertices of \( H \) with loops is disconnected (and \( H \) is connected), and hence it follows that \( RET-H \) is NP-complete, cf. Feder, Hell, and Huang [5]; we however need to determine the complexity of \( COMP-H \).

We now first prove the following theorem which is a special case of Theorem 4.1. We show in Theorem 4.1 that \( COMP-H \) is NP-complete for any path \( H \) of length greater than or equal to two, whose first and last vertices have loops, and none of its other vertices have loops. Although the following theorem is a special case of Theorem 4.1, we first present its proof, as it is easier to understand and will be helpful in understanding the proof for Theorem 4.1 which we present later.

When proving NP-completeness of \( COMP-H \) for any graph \( H \), we note the genericness of the technique described by Vikas [25,26] which involves proving equivalence of three different statements. The technique was described by Vikas [25,26] as a generic technique, which we have used throughout supporting its genericness. The actual construction of the gadget graphs, needed to form the three statements for proving NP-completeness of \( COMP-H \), is however a separate issue and depends on the graph \( H \).

**Theorem 3.1.** \( COMP-H \) is NP-complete for the graph \( H \) in Fig. 2 (Fig. 1(a)).

**Proof.** We prove NP-completeness of \( COMP-H \) by giving a polynomial transformation from \( RET-H \) to \( COMP-H \). As mentioned above, the problem \( RET-H \) is NP-complete, cf. Feder, Hell, and Huang [5]. Let \( G \) be a graph containing \( H \) as an induced subgraph, i.e., let \( G \) be an instance of \( RET-H \). We construct in time polynomial in the size of \( G \), a graph \( G' \) (containing \( G \) as an induced subgraph) such that the following statements (i), (ii), and (iii) are equivalent:

(i) \( G \) retracts to \( H \).
(ii) \( G' \) retracts to \( H \).
(iii) \( G' \) compacts to \( H \).

We prove that (i) is equivalent to (ii), and (ii) is equivalent to (iii), in Lemma 3.1.1 and Lemma 3.1.2 respectively. Since \( RET-H \) is NP-complete, this shows that \( COMP-H \) is NP-complete.
Fig. 1. List of graphs $H$ to be considered.
The construction of $G'$ is as follows. For each vertex $v$ in $V(G - H)$, we add to $G$ three distinct new vertices: $u_v$ adjacent to $v$, $h_0$; $w_v$ adjacent to $v$, $u_v$; and $y_v$ adjacent to $u_v$, $w_v$, $h_2$. Thus $u_v$, $w_v$, and $y_v$ form a triangle, and $u_v$, $w_v$, and $v$ form a triangle. See Fig. 3. Note that there could be edges in $G$ from $v$ to some vertices of $H$ but as mentioned earlier, in Fig. 3 and all subsequent figures in this paper, we are not depicting these edges. This completes the construction of $G'$.

We now prove the following two lemmas in order to prove the theorem.

**Lemma 3.1.1.** $G$ retracts to $H$ if and only if $G'$ retracts to $H$.

**Proof.** If $G'$ retracts to $H$ then it is clear that $G$ retracts to $H$ since $G$ is a subgraph of $G'$. Now suppose that $r : G \to H$ is a retraction. We define a retraction $r' : G' \to H$ as follows.

For each vertex $v$ of the graph $G$, we define

$$r'(v) = r(v).$$

For the vertices $u_v$, $w_v$, and $y_v$ of $G'$, with $v \in V(G - H)$, we define

$$r'(u_v) = h_0, \quad r'(w_v) = h_0, \quad \text{and} \quad r'(y_v) = h_1, \quad \text{if } r(v) = h_0 \text{ or } h_1,$$

$$r'(u_v) = h_1, \quad r'(w_v) = h_2, \quad \text{and} \quad r'(y_v) = h_2, \quad \text{if } r(v) = h_2.$$
It is not difficult to see that for every edge \( ab \) of \( G' \), \( r'(a)r'(b) \) is an edge of \( H \), i.e., \( r' : G' \rightarrow H \) is a homomorphism. Recall that the edges of \( G' \) are: \( gg', uv_{v}, vw_{v}, h_{0}u_{v}, h_{2}y_{v}, u_{v}w_{v}, u_{v}y_{v}, \) and \( w_{v}y_{v} \), with \( gg' \in E(G), v \in V(G - H) \). Since \( r'(h) = r(h) = h \), for all \( h \in V(H) \), the homomorphism \( r' : G' \rightarrow H \) is a retraction, and the lemma is proved. ∎

**Lemma 3.1.2.** \( G' \) retracts to \( H \) if and only if \( G' \) compacts to \( H \).

**Proof.** If \( G' \) retracts to \( H \) then by definition \( G' \) compacts to \( H \). Now suppose that \( c : G' \rightarrow H \) is a compaction. We first prove that \( c(h_{0}) \neq c(h_{2}) \). Suppose that \( c(h_{0}) = c(h_{2}) \). Since \( h_{0} \) and \( h_{2} \) both have loops, it must be that \( c(h_{0}) = c(h_{2}) = h_{0} \) or \( c(h_{0}) = c(h_{2}) = h_{2} \). Without loss of generality, let \( c(h_{0}) = c(h_{2}) = h_{0} \). We can assume this, as due to symmetry of vertices in \( H \), we can always redefine the compaction \( c \) so that \( c(h_{0}) = c(h_{2}) = h_{0} \). Since \( u_{v}, y_{v}, \) and \( h_{1} \) are adjacent to \( h_{0} \) or \( h_{2} \), \( c(u_{v}) \neq h_{2}, \), \( c(y_{v}) \neq h_{2}, \) and \( c(h_{1}) \neq h_{2} \), with \( v \in V(G - H) \). This implies that if \( c(w_{v}) = h_{2}, \) for some vertex \( v \in V(G - H) \), then \( c(u_{v}) = c(y_{v}) = h_{1}, \) as \( u_{v} \) and \( y_{v} \) are adjacent to \( w_{v} \). But this is impossible, as \( u_{v} \) is adjacent to \( y_{v} \), and \( h_{1} \) does not have a loop. Hence \( c(w_{v}) \neq h_{2}, \) for any \( v \in V(G - H) \). Similarly, if \( c(v) = h_{2}, \) for some vertex \( v \in V(G - H) \), then \( c(u_{v}) = c(w_{v}) = h_{1}, \) and \( h_{1} \) does not have a loop. Hence \( c(v) \neq h_{2}, \) for any \( v \in V(G - H) \).

Thus we have shown that \( c(a) \neq h_{2}, \) for all \( a \in V(G'), \) contradicting our assumption that \( c : G' \rightarrow H \) is a compaction. Hence, it must be that \( c(h_{0}) \neq c(h_{2}) \). Since \( h_{0} \) and \( h_{2} \) are the only vertices in \( H \) with a loop, \( c([h_{0}, h_{2}]) = [h_{0}, h_{2}] \). Without loss of generality, let \( c(h_{0}) = h_{0} \) and \( c(h_{2}) = h_{2} \) (due to symmetry). Since \( h_{1} \) is adjacent to \( h_{0} \) and \( h_{2} \), we have \( c(h_{1}) = h_{1} \). Thus \( c : G' \rightarrow H \) is a retraction, and the lemma is proved. ∎

This completes the proof of Theorem 3.1. □

For the graph \( H \) in Fig. 1(b), the fact that \( COMP-H \) is NP-complete follows from our general result in Theorem 4.1 which we present later. We now consider the graph in Fig. 1(p).

**Theorem 3.2.** \( COMP-H \) is NP-complete for the graph \( H \) in Fig. 4 (Fig. 1(p)).

![Fig. 4. H.](image-url)

**Proof.** Let \( S \) be the graph in Fig. 5. We give a polynomial transformation from \( RET-S \) to \( COMP-H \). Let a graph \( G \) containing \( S \) as an induced subgraph be an instance of \( RET-S \). We construct in time polynomial in the size of \( G \), a graph \( G' \) (containing \( G \) and \( H \) as induced subgraphs) such that the following statements (i),
We prove that (i) is equivalent to (ii), and (ii) is equivalent to (iii), in Lemma 3.2.1 and Lemma 3.2.2.

(ii), and (iii) are equivalent:

(i) $G$ retracts to $S$.

(ii) $G'$ retracts to $H$.

(iii) $G'$ compacts to $H$.

We prove that (i) is equivalent to (ii), and (ii) is equivalent to (iii), in Lemma 3.2.1 and Lemma 3.2.2 respectively. Since $RET-S$ is NP-complete, cf. Feder, Hell, and Huang [5], this shows that $COMP-H$ is NP-complete. We note that the equivalence of (i) and (ii) shows that $RET-H$ is NP-complete. As mentioned earlier, the fact that $RET-H$ is NP-complete is already known, cf. Feder, Hell, and Huang [5].

The construction of $G'$ is as follows. The graph $G'$ is the graph $G$ constructed in the proof of Theorem 3.1 (where $H$ is replaced by $S$, see Fig. 3) together with a new vertex $h_3$ adjacent to $h_0$ and $h_2$ only. Note that the resulting graph $G'$ has $H$ as an induced subgraph. We now prove the following two lemmas in order to prove the theorem.

**Lemma 3.2.1.** $G$ retracts to $S$ if and only if $G'$ retracts to $H$.

**Proof.** If $G'$ retracts to $H$ then $G'$ retracts to $S$, since $H$ retracts to $S$. Hence $G$ retracts to $S$, since $G$ is a subgraph of $G'$.

If $G$ retracts to $S$ then it follows from Lemma 3.1.1, that $G' - h_3$ retracts to $S$. Since $h_3$ is adjacent to only $h_0$ and $h_2$ in $G'$, it follows that $G'$ retracts to $H$ (if $r : G' - h_3 \rightarrow S$ is a retraction then we can define a retraction $r' : G' \rightarrow H$ as follows: $r'(x) = r(x)$, for all $x \in V(G' - h_3)$, and $r'(h_3) = h_3$). □

**Lemma 3.2.2.** $G'$ retracts to $H$ if and only if $G'$ compacts to $H$.

**Proof.** If $G'$ retracts to $H$ then by definition $G'$ compacts to $H$. Now suppose that $c : G' \rightarrow H$ is a compaction. We first prove that $c(h_0) \neq c(h_2)$. Suppose that $c(h_0) = c(h_2)$. Since $h_0$ and $h_2$ both have loops, it must be that $c(h_0) = c(h_2) = h_0$ or $c(h_0) = c(h_2) = h_2$. Without loss of generality, let $c(h_0) = c(h_2) = h_0$ (due to symmetry). From the proof of Lemma 3.1.2 and the fact that $h_3$ neither has a loop nor is adjacent to $h_1$, it follows that $c(a) \neq h_2$, for all $a \in V(G' - h_3)$. Also, $c(h_3) \neq h_2$, as $h_3$ is adjacent to $h_0$ in $G'$. Thus we have that $c(a) \neq h_2$, for all $a \in V(G')$, which contradicts the assumption that $c : G' \rightarrow H$ is a compaction. Hence, it must be that $c(h_0) \neq c(h_2)$. Since $h_0$ and $h_2$ are the only vertices in $H$ with a loop, $c([h_0, h_2]) = \{h_0, h_2\}$. Without loss of generality, let $c(h_0) = h_0$ and $c(h_2) = h_2$ (due to symmetry). Since $h_1$ and $h_3$ are both adjacent to $h_0$ and $h_2$, we have $c(h_1), c(h_3) \in \{h_1, h_3\}$. Without loss of generality (due to symmetry), let $c(h_1) = h_1$ and $c(h_3) = h_3$ (in fact, $h_3$ is adjacent to only $h_0$ and $h_2$ in $G'$). Thus $c : G' \rightarrow H$ is a retraction, and the lemma is proved. □

We have thus proved Theorem 3.2. □
Now consider the graphs $H$ in Figs. 1(c)–(f), (j), and (k). Let $S$ be the graph in Fig. 5. Each of these graphs $H$ have a copy of $S$ as an induced subgraph to which it retracts. Similar to the proof for Theorem 3.2 (with some minor modifications in arguments in the proof of Lemma 3.2.2 when appropriate), we can show that $COMP-H$ is NP-complete for each of these graphs $H$. The construction of $G'$ is analogous to the construction in Theorem 3.2. Let $H$ be the graph in Fig. 1(e) or (f), so as to see some minor changes in arguments. There are two vertices at distance two in $H$ from both the loop vertices of $S$. Let $V(H) - V(S) = \{h_3\}$, and $c : G' \rightarrow H$ be a compaction. In comparison to the proof of Lemma 3.2.2, if $c(h_0) = c(h_2) = h_0$ then from the proof of Lemma 3.1.2, it follows that $c(a) \neq h_2$ and $c(a) \neq h_3$, for all $a \in V(G' - h_3)$, but it may be possible to have $c(h_3) = h_2$ or $h_3$, implying that either $h_2$ or $h_3$ will remain uncovered under $c$, which is a contradiction, and hence it must be that $c(h_0) \neq c(h_2)$.

**Theorem 3.3.** $COMP-H$ is NP-complete for the graph $H$ in Fig. 6 (Fig. 1(m)).

![Fig. 6. $H$.](image)

**Proof.** The proof is again similar to that of Theorem 3.2. We include the proof here as there is some deviation. Let $S$ be the graph in Fig. 5. We give a polynomial transformation from $RET-S$ to $COMP-H$. Let $G$ be a graph with $S$ as an induced subgraph of $G$, i.e., let $G$ be instance of $RET-S$. We again construct in time polynomial in the size of $G$, a graph $G'$ (containing $G$ and $H$ as induced subgraphs) such that the statements (i), (ii), and (iii), as mentioned in the proof of Theorem 3.2, are equivalent. We prove the equivalence of these statements in Lemma 3.3.1 and Lemma 3.3.2. We note that the equivalence of (i) and (ii) shows that $RET-H$ is also NP-complete. The fact that $RET-H$ is NP-complete is also known earlier, cf. Feder, Hell, and Huang [5].

The construction of $G'$ is as follows. The graph $G'$ is the graph $G'$ constructed in the proof of Theorem 3.1 (where $H$ is replaced by $S$, see Fig. 3) together with a new vertex $h_3$ adjacent to every other vertex of $G'$. Thus the resulting graph $G'$ has $H$ as an induced subgraph. We now prove the following two lemmas in order to prove the theorem.

**Lemma 3.3.1.** $G$ retracts to $S$ if and only if $G'$ retracts to $H$.

**Proof.** Suppose that $G'$ retracts to $H$, and let $r' : G' \rightarrow H$ be a retraction. Since $h_3$ is adjacent to every vertex of $G' - h_3$, and $h_3$ does not have a loop, $r'(x) \neq h_3$, for all $x \in V(G' - h_3)$. This implies that $G' - h_3$ retracts to $S$. Since $G$ is a subgraph of $G' - h_3$, $G$ retracts to $S$. 
Now suppose that $G$ retracts to $S$. Then it follows, from Lemma 3.1.1, that $G' - h_3$ retracts to $S$. Since $h_3$ is adjacent to every vertex of $S$ in $H$, it follows that $G'$ retracts to $H$. □

**Lemma 3.3.2.** $G'$ retracts to $H$ if and only if $G'$ compacts to $H$.

**Proof.** If $G'$ retracts to $H$ then by definition $G'$ compacts to $H$. Now suppose that $c : G' \rightarrow H$ is a compaction. We first show that $G' - h_3$ compacts to $S$. Since $h_3$ is adjacent to every vertex of $G' - h_3$, $c(h_3) = h_1$ or $h_3$. Without loss of generality, suppose that $c(h_3) = h_3$ (due to symmetry). Since $h_3$ does not have a loop, $r'(x) \neq h_3$, for all $x \in V(G' - h_3)$. This implies that $G' - h_3$ compacts to $S$. It follows, from Lemma 3.1.2, that $G' - h_3$ retracts to $S$. Since $h_3$ is adjacent to every vertex of $S$ in $H$, it follows that $G'$ retracts to $H$. □

This completes the proof of Theorem 3.3. □

We now consider the graph $H$ in Fig. 1(g). We first prove that $RET-H$ is NP-complete and then prove that $COMP-H$ is NP-complete. Although NP-completeness of $RET-H$ follows as a byproduct when proving NP-completeness of $COMP-H$, we prove it separately, to illustrate the fact that a simpler construction will suffice for proving NP-completeness of $RET-H$, as compared to the construction used for proving NP-completeness of $COMP-H$.

**Theorem 3.4.** $RET-H$ is NP-complete for the graph $H$ in Fig. 7 (Fig. 1(g)).

![Fig. 7. H.](image)

**Proof.** Let $S$ be the irreflexive triangle in Fig. 8.

![Fig. 8. S.](image)
We give a polynomial transformation from RET-S to RET-H. Recall that it follows from Hell and Nesetril [16] that RET-S is NP-complete. Let a graph $G$ containing $S$ as an induced subgraph be an instance of RET-S. We construct a graph $G'$ (containing $G$ and $H$ as induced subgraphs), in time polynomial in the size of $G$, such that $G$ retracts to $S$ if and only if $G'$ retracts to $H$.

The construction of $G'$ is as follows. For each vertex $v$ of $G - S$, we add to $G$ two distinct new vertices: $u_v$ adjacent to $v$ and $h_0$; and $w_v$ adjacent to $v$, $h_1$, and $u_v$. We also add to $G$ a new vertex $h_3$ adjacent to $h_2$ and itself only (thus $H$ is an induced subgraph of the resulting graph). See Fig. 9. This completes the construction of $G'$.

We now prove the following lemma in order to prove the theorem.

**Lemma 3.4.1.** $G$ retracts to $S$ if and only if $G'$ retracts to $H$.

**Proof.** Suppose first that $r' : G' \rightarrow H$ is a retraction. We show that $r'(v) \neq h_3$, for all $v \in V(G - S)$. Suppose that $r'(v) = h_3$, for some $v \in V(G - S)$. This implies that $r'(u_v) = r'(w_v) = h_2$, as $r'(u_v)$ must be adjacent to $r'(h_0) = h_0$ and $r'(v) = h_3$, and $r'(w_v)$ must be adjacent to $r'(h_1) = h_1$ and $r'(v) = h_3$. But this is impossible, as $u_v$ is adjacent to $w_v$, and $h_2$ has no loop. Thus $r'(v) \neq h_3$. Hence $G$ retracts to $S$ under $r'$.

Now suppose that $r : G \rightarrow S$ is a retraction. We define a retraction $r' : G' \rightarrow H$ as follows. We define $r'(v) = r(v)$, for each vertex $v$ of the graph $G$, and $r'(h_3) = h_3$. For the vertices $u_v$ and $w_v$ of $G'$, with $v \in V(G - S)$, we define

- $r'(u_v) = h_1$ and $r'(w_v) = h_2$, if $r(v) = h_0$,
- $r'(u_v) = h_2$ and $r'(w_v) = h_0$, if $r(v) = h_1$,
- $r'(u_v) = h_1$ and $r'(w_v) = h_0$, if $r(v) = h_2$.

Recall that the edges of $G'$ are: $gg'$, $vuv$, $vwv$, $h_0u_v$, $h_1w_v$, $u_vw_v$, $h_2h_3$, $h_3h_3$, with $gg' \in E(G)$, $v \in V(G - S)$. It is not difficult to see that $r' : G' \rightarrow H$ is a retraction. □

Thus we have now proved Theorem 3.4. □

**Theorem 3.5.** COMP-H is NP-complete for the graph $H$ in Fig. 7 (Fig. 1(g)).

**Proof.** Let $S$ be the irreflexive triangle in Fig. 8. We give a polynomial transformation from RET-S to COMP-H. Let a graph $G$ containing $S$ as an induced subgraph be an instance of RET-S. Once more, we
We define $r$. This completes the definition of $r$. We define a retraction $uv$ of the graph $G$ retracts to $S$ if and only if $G$ retracts to $S$. For the vertices $v$ of the resulting graph. See Fig. 10. This completes the construction of $G$.\nThe construction of $G'$ is as follows. For each vertex $v$ of $G - S$, we add to $G$ six distinct new vertices: $u_v$ adjacent to $v$ and $h_0$; $w_v$ adjacent to $v$, $h_1$, and $u_v$ (so far, apart from $h_3$, this gives us Fig. 9); $x_v$ adjacent to $v$ and $h_0$; $y_v$ adjacent to $x_v$, $w_v$, and $h_0$; $s_v$ adjacent to $v$ and $h_1$; and $z_v$ adjacent to $s_v$, $u_v$, and $h_1$. We also add to $G$ a new vertex $h_3$ adjacent to $h_2$ and itself only (thus $H$ is an induced subgraph of the resulting graph). See Fig. 10. This completes the construction of $G'$.

We now prove the following two lemmas in order to prove the theorem.

**Lemma 3.5.1.** $G$ retracts to $S$ if and only if $G'$ retracts to $H$.

**Proof.** Suppose first that $r' : G' \rightarrow H$ is a retraction. As in the proof of Lemma 3.4.1, we conclude that $r'(v) \neq h_3$, for all $v \in V(G - S)$. Hence $G$ retracts to $S$ under $r'$. Now suppose that $r : G \rightarrow S$ is a retraction. We define a retraction $r' : G' \rightarrow H$ as follows.

We define $r'(v) = r(v)$, for each vertex $v$ of the graph $G$, and $r'(h_3) = h_3$. For the vertices $u_v, w_v, x_v, y_v, z_v$, and $s_v$ of $G'$, with $v \in V(G - S)$, we define $r'$ as follows.

If $r(v) = h_0$ then

$$r'(u_v) = h_1, \quad r'(w_v) = h_2, \quad r'(x_v) = h_2, \quad r'(y_v) = h_1, \quad r'(s_v) = h_2, \quad \text{and} \quad r'(z_v) = h_0.$$ 

If $r(v) = h_1$ then

$$r'(u_v) = h_2, \quad r'(w_v) = h_0, \quad r'(x_v) = h_2, \quad r'(y_v) = h_1, \quad r'(s_v) = h_2, \quad \text{and} \quad r'(z_v) = h_0.$$ 

If $r(v) = h_2$ then

$$r'(u_v) = h_1, \quad r'(w_v) = h_0, \quad r'(x_v) = h_1, \quad r'(y_v) = h_2, \quad r'(s_v) = h_0, \quad \text{and} \quad r'(z_v) = h_2.$$ 

This completes the definition of $r'$. It is not difficult to see that $r' : G' \rightarrow H$ is a retraction. Recall that the edges of $G'$ are: $gg', uu_v, vv_v, xv_v, vs_v, h_0u_v, h_0x_v, h_0y_v, h_1w_v, h_1s_v, h_1z_v, u_vw_v, u_vz_v, w_vy_v, x_vy_v, s_vz_v, h_2h_3, h_3h_3$, with $gg' \in E(G)$, $v \in V(G - S)$. \qed

Fig. 10. Construction of $G'$ for a vertex $v$ in $G - H$. 

construct in time polynomial in the size of $G$, a graph $G'$ (containing $G$ and $H$ as induced subgraphs) such that the statements (i), (ii), and (iii), as mentioned in the proof of Theorem 3.2, are equivalent.

We prove that (i) is equivalent to (ii), and (ii) is equivalent to (iii), in Lemma 3.5.1 and Lemma 3.5.2 respectively. Since $RET-S$ is NP-complete, this shows that $COMP-H$ is NP-complete. We note that the equivalence of (i) and (ii) shows that $RET-H$ is also NP-complete. The fact that $RET-H$ is NP-complete is also shown using a simpler construction in the proof of Theorem 3.4.
Lemma 3.5.2. $G'$ retracts to $H$ if and only if $G'$ compacts to $H$.

Proof. If $G'$ retracts to $H$ then by definition $G'$ compacts to $H$. Now suppose that $c : G' \to H$ is a compaction. We prove that $c(h_0) \neq h_3$ or $h_2$. We have $c(h_3) = h_3$, as $h_3$ has a loop and there is no other vertex with a loop in $H$.

Suppose first that $c(h_0) = h_3$. Since $c(h_1)$ and $c(h_2)$ must both be adjacent to $c(h_0) = h_3$, we have $c(h_1), c(h_2) \in \{h_2, h_3\}$. Thus $c(H) \subseteq \{h_2, h_3\}$. Since $c : G' \to H$ is a compaction, there exists a vertex $a$ of $G' - H$ with $c(a) = h_0$, and hence $a$ must not be adjacent to $h_0$ in $G'$ as $c(h_0) = h_3$. The only possible vertices of $G' - H$ not adjacent to $h_0$ are $w_v, z_v,$ and $s_v$, with $v \in V(G - S)$. We also need to include $v \in V(G - S)$ if $v$ is not adjacent to $h_0$. We shall consider each of these possible vertices $a$ and show that $c(a) \neq h_0$.

Let $c(v) = h_0$, with $v \in V(G - S)$. Since $c(u_v)$ and $c(x_v)$ must both be adjacent to $c(h_0) = h_3$ and $c(v) = h_0$, this implies that $c(u_v) = h_2$ and $c(x_v) = h_2$. Since $w_v$ is adjacent to $v$ and $u_v$, we have $c(w_v) = h_1$. Now, $c(y_v) = h_2$, as $c(y_v)$ must be adjacent to $c(h_0) = h_3$ and $c(w_v) = h_1$. Thus we have $c(x_v) = c(y_v) = h_2$ which is impossible as $x_v$ is adjacent to $y_v$ in $G'$, and $h_2$ does not have a loop.

Now let $c(w_v) = h_0$, with $v \in V(G - S)$. Since $u_v$ and $y_v$ are both adjacent to $w_v$ and $h_0$, we have $c(u_v) = h_2$ and $c(y_v) = h_2$. Since $v$ is adjacent to $u_v$ and $w_v$, this implies that $c(v) = h_1$. Now, since $c(x_v)$ must be adjacent to $c(h_0) = h_3$ and $c(v) = h_1$, we have $c(x_v) = h_2$. Thus we again have $c(x_v) = c(y_v) = h_2$ which is impossible.

Now suppose that $c(z_v) = h_0$, with $v \in V(G - S)$. Since $u_v$ and $h_1$ are both adjacent to $z_v$ and $h_0$, we have $c(u_v) = h_2$ and $c(h_1) = h_2$. Since $s_v$ is adjacent to $z_v$ and $h_1$, this implies that $c(s_v) = h_1$. Since $c(v)$ must be adjacent to $c(u_v) = h_2$ and $c(s_v) = h_1$, this implies that $c(v) = h_0$, which we have already proved is impossible.

Now assume that $c(s_v) = h_0$, with $v \in V(G - S)$. Since $h_1$ is adjacent to $s_v$ and $h_0$, we have $c(h_1) = h_2$. Since $z_v$ is adjacent to $s_v$ and $h_1$, this implies that $c(z_v) = h_1$. Since $u_v$ is adjacent to $z_v$ and $h_0$, we have $c(u_v) = h_2$. Since $v$ is adjacent to $u_v$ and $s_v$, this implies that $c(v) = h_1$. Now, since $c(w_v)$ must be adjacent to $c(u_v) = h_2$ and $c(v) = h_1$, we have $c(w_v) = h_0$, which we have already proved is impossible.

We have thus shown that $c(h_0) \neq h_3$. Now suppose that $c(h_0) = h_2$. Since $c(h_2)$ must be adjacent to $c(h_3) = h_3$ and $c(h_0) = h_2$, this implies that $c(h_2) = h_3$. Now $c(h_1)$ must be adjacent to $c(h_0) = h_2$ and $c(h_2) = h_3$. Hence $c(h_1) = h_3$. Thus $c(H) \subseteq \{h_2, h_3\}$. As above, there exists a vertex $a$ of $G' - H$ with $c(a) = h_0$, and hence $a$ must not be adjacent to $h_1$ in $G'$ as $c(h_1) = h_3$. The only possible vertices of $G' - H$ not adjacent to $h_1$ are $u_v, x_v,$ and $y_v$, with $v \in V(G - S)$. We also need to include $v \in V(G - S)$ if $v$ is not adjacent to $h_1$. We consider each of these possible vertices $a$ and show that $c(a) \neq h_0$.

First let $c(v) = h_0$, with $v \in V(G - S)$. Since $c(w_v)$ and $c(s_v)$ must both be adjacent to $c(h_1) = h_3$ and $c(v) = h_0$, this implies that $c(w_v) = h_2$ and $c(s_v) = h_2$. Since $u_v$ is adjacent to $v$ and $w_v$, we have $c(u_v) = h_1$. Now, $c(z_v) = h_2$, as $c(z_v)$ must be adjacent to $c(h_1) = h_3$ and $c(u_v) = h_1$. Thus we have $c(s_v) = c(z_v) = h_2$ which is impossible as $s_v$ is adjacent to $z_v$ in $G'$, and $h_2$ does not have a loop.

Next let $c(u_v) = h_0$, with $v \in V(G - S)$. Since $c(w_v)$ and $c(z_v)$ must both be adjacent to $c(h_1) = h_3$ and $c(u_v) = h_0$, we have $c(w_v) = h_2$ and $c(z_v) = h_2$. Since $v$ is adjacent to $u_v$ and $w_v$, this implies that
Now, since \( c(s_v) \) must be adjacent to \( c(h_1) = h_3 \) and \( c(v) = h_1 \), we have \( c(s_v) = h_2 \). Thus we again have \( c(s_v) = c(z_v) = h_2 \) which is impossible.

Now let \( c(x_v) = h_0 \), with \( v \in V(G - S) \). Since \( c(y_v) \) must be adjacent to \( c(h_0) = h_2 \) and \( c(x_v) = h_0 \), this implies that \( c(y_v) = h_1 \). Since \( c(w_v) \) must be adjacent to \( c(h_1) = h_3 \) and \( c(y_v) = h_1 \), we have \( c(w_v) = h_2 \). Since \( v \) is adjacent to \( w_v \) and \( x_v \), this implies that \( c(v) = h_1 \). Now, since \( c(u_v) \) must be adjacent to \( c(w_v) = h_2 \) and \( c(v) = h_1 \), we have \( c(u_v) = h_0 \), which we have already proved is impossible.

Finally, let \( c(y_v) = h_0 \), with \( v \in V(G - S) \). Since \( c(x_v) \) must be adjacent to \( c(h_0) = h_2 \) and \( c(y_v) = h_0 \), we have \( c(x_v) = h_1 \). Since \( c(w_v) \) must be adjacent to \( c(h_1) = h_3 \) and \( c(y_v) = h_0 \), this implies that \( c(w_v) = h_2 \). Now, since \( c(v) \) must be adjacent to \( c(x_v) = h_1 \) and \( c(w_v) = h_2 \), this implies that \( c(v) = h_0 \), which we have already proved is impossible. This completes the proof that \( c(h_0) \neq h_2 \).

We have thus shown that \( c(h_0) \neq h_2 \) and \( c(h_0) \neq h_3 \). Hence \( c(h_0) = h_0 \) or \( h_1 \). Due to symmetry of the vertices \( h_0 \) and \( h_1 \) in \( H \), we can choose \( c \) to be the compaction with \( c(h_0) = h_0 \). Recall that \( c(h_3) = h_3 \). Since \( c(h_2) \) must be adjacent to \( c(h_0) = h_0 \) and \( c(h_3) = h_3 \), this implies that \( c(h_2) = h_2 \). Now, \( c(h_1) \) must be adjacent to \( c(h_0) = h_0 \) and \( c(h_2) = h_2 \). Hence \( c(h_1) = h_1 \). Thus we have \( c(h_i) = h_i \), for all \( i = 0, 1, 2, 3 \). Hence \( c : G' \to H \) is a retraction, and the lemma is proved. \( \square \)

This completes the proof of Theorem 3.5. \( \square \)

**Theorem 3.6.** COMP-H and RET-H are polynomial time solvable for the graphs \( H \) in Figs. 11 (a) and (b) (Figs. 1(h) and (i)).

![Fig. 11. H.](image)

**Proof.** Let \( H \) be the graph (a) or (b) in Fig. 11. We first show that RET-H is polynomial time solvable by giving a polynomial time algorithm for it. Since COMP-H polynomially transforms to RET-H (under Turing reduction), this shows that COMP-H is also polynomial time solvable. Let a graph \( G \) containing \( H \) as an induced subgraph be an instance of RET-H. The algorithm for RET-H is as follows.

1. If \( h_3 \) has neighbours \( u \) and \( w \) in \( G \), not necessarily distinct, such that \( u \) and \( w \) are adjacent, then \( G \) does not retract to \( H \).
2. Else $G$ retracts to $H$ with a retraction $r : G \to H$ defined as follows:

\[
\begin{align*}
    r(h) &= h, \text{ for all } h \in V(H), \\
    r(v) &= h_1, \text{ for all } v \in Nbr(h_3), \text{ and} \\
    r(v) &= h_0, \text{ for all } v \in V(G) - (Nbr(h_2) \cup V(H)).
\end{align*}
\]

□

**Theorem 3.7.** COMP-$H$ and RET-$H$ are polynomial time solvable for the graph $H$ in Fig. 12 (Fig. 1(a)).

![Fig. 12. $H$.](image)

**Proof.** It again suffices to give a polynomial time algorithm for RET-$H$, as COMP-$H$ polynomially transforms to RET-$H$. Let $G$ be a graph containing $H$ as an induced subgraph, i.e., let $G$ be an instance of RET-$H$. The algorithm for RET-$H$ is given below.

1. If $h_2$ has neighbours $u$ and $w$ in $G$, not necessarily distinct, such that $u$ and $w$ are adjacent, then $G$ does not retract to $H$.
2. Else $G$ retracts to $H$ with a retraction $r : G \to H$ defined as follows:

\[
\begin{align*}
    r(h) &= h, \text{ for all } h \in V(H), \\
    r(v) &= h_1, \text{ for all } v \in Nbr(h_2) - \{h_3\}, \text{ and} \\
    r(v) &= h_0, \text{ for all } v \in V(G) - (Nbr(h_2) \cup V(H)).
\end{align*}
\]

□

**Theorem 3.8.** COMP-$H$ and RET-$H$ are polynomial time solvable for the graph $H$ in Fig. 13 (Fig. 1(o)).

![Fig. 13. $H$.](image)
Proof. We give below a polynomial time algorithm for RET-$H$. Let $G$ be a graph with $H$ as its induced subgraph, i.e., let $G$ be an instance of RET-$H$.

1. If $h_2$ and $h_3$ have a common neighbour in $G$ then $G$ does not retract to $H$.
2. Else $G$ retracts to $H$ with a retraction $r : G \rightarrow H$ defined just like in step 2 in the proof of Theorem 3.7. □

Theorem 3.9. COMP-$H$ and RET-$H$ are polynomial time solvable for the graph $H$ in Fig. 14 (Fig. 1(q)).

![Diagram of H](image1)

Fig. 14. $H$.

Proof. We again give a polynomial time algorithm for RET-$H$. Let a graph $G$ containing $H$ as an induced subgraph be an instance of RET-$H$. Let $S$ be a subgraph of $G$ induced by $Nbr(h_3)$ (which includes $h_0$ and $h_2$). The algorithm for RET-$H$ is described below.

1. If there exists a path from $h_0$ to $h_2$ in $S$ then $G$ does not retract to $H$.
2. Else $G$ retracts to $H$ with a retraction $r : G \rightarrow H$ defined as follows. We have that $h_0$ and $h_2$ are disconnected in $S$. Let $Z$ be a component of $S$ containing $h_2$. Thus $h_0 \notin Z$. We define
   
   $$r(h) = h, \text{ for all } h \in V(H),$$
   $$r(v) = h_0, \text{ for all } v \in Nbr(h_3) - V(Z),$$
   $$r(v) = h_2, \text{ for all } v \in V(Z), \text{ and}$$
   $$r(v) = h_1, \text{ for all } v \in V(G) - (Nbr(h_3) \cup V(H)).$$

   □

Theorem 3.10. COMP-$H$ and RET-$H$ are polynomial time solvable for the graph $H$ in Fig. 15 (Fig. 1(l)).

![Diagram of H](image2)

Fig. 15. $H$. 

Proof. We again give a polynomial time algorithm for RET-$H$. Let a graph $G$ containing $H$ as an induced subgraph be an instance of RET-$H$. Let $S$ be a subgraph of $G$ induced by $Nbr(h_3)$ (which includes $h_0$ and $h_2$). The algorithm for RET-$H$ is described below.

1. If there exists a path from $h_0$ to $h_2$ in $S$ then $G$ does not retract to $H$.
2. Else $G$ retracts to $H$ with a retraction $r : G \rightarrow H$ defined as follows. We have that $h_0$ and $h_2$ are disconnected in $S$. Let $Z$ be a component of $S$ containing $h_2$. Thus $h_0 \notin Z$. We define
   
   $$r(h) = h, \text{ for all } h \in V(H),$$
   $$r(v) = h_0, \text{ for all } v \in Nbr(h_3) - V(Z),$$
   $$r(v) = h_2, \text{ for all } v \in V(Z), \text{ and}$$
   $$r(v) = h_1, \text{ for all } v \in V(G) - (Nbr(h_3) \cup V(H)).$$

   □
Proof. It will again suffice to give a polynomial time algorithm for \( \text{RET-H} \). Let a graph \( G \) with \( H \) as its induced subgraph be an instance of \( \text{RET-H} \). Let \( S \) be a subgraph of \( G \) induced by \( Nbr(h_2) \) (which includes \( h_1 \) and \( h_3 \)). We outline below the algorithm for \( \text{RET-H} \).

1. If \( S \) is not bipartite then \( G \) does not retract to \( H \).
2. Else \( G \) retracts to \( H \) with a retraction \( r : G \rightarrow H \) defined as follows. Let \((S_A, S_B)\) be a bipartition of \( S \), with \( h_1 \in S_A \) (and hence \( h_3 \in S_B \)). We define
   \[
   r(h) = h, \text{ for all } h \in V(H),
   \]
   \[
   r(v) = h_1, \text{ for all } v \in S_A,
   \]
   \[
   r(v) = h_3, \text{ for all } v \in S_B, \text{ and}
   \]
   \[
   r(v) = h_0, \text{ for all } v \in V(G) - (Nbr(h_2) \cup V(H)).
   \]

We have now considered all the graphs \( H \) in Fig. 1 and shown that \( \text{COMP-H} \) and \( \text{RET-H} \) are NP-complete or polynomial time solvable, and that the complexity classification of \( \text{COMP-H} \) and \( \text{RET-H} \) do not differ. For the graphs \( H \) in Figs. 1(a)–(g), (j), (k), (m), and (p), both \( \text{COMP-H} \) and \( \text{RET-H} \) are polynomial time solvable (it is not difficult to see that none of these graphs are absolute retracts which would imply polynomiality anyway; we do not discuss absolute retracts here but can be referred in literature on retraction such as [1,2]). As discussed before, for the graphs \( H \) with at most four vertices not listed in Fig. 1, either we know from the results mentioned in the beginning of this section, or we can infer from Theorems 2.1, 2.2, 2.5, and 2.6, that \( \text{COMP-H} \) and \( \text{RET-H} \) are NP-complete or polynomial time solvable, and notice again that the complexity classification of \( \text{COMP-H} \) and \( \text{RET-H} \) do not differ.

A list of all graphs \( H \) with at most four vertices for which \( \text{COMP-H} \) and \( \text{RET-H} \) are NP-complete is given in Fig. 16. For all other graphs \( H \) with at most four vertices, not in this list, both \( \text{COMP-H} \) and \( \text{RET-H} \) are polynomial time solvable.

We note the following theorem for retraction and then remark on some of our findings.

Theorem 3.11. Let \( H \) be a given graph, and \( H' \) be a graph containing \( H \) as an induced subgraph, such that \( H' \) retracts to \( H \). Then \( \text{RET-H} \) polynomially transforms to \( \text{RET-H}' \).

Proof. Let \( G \) be a graph with \( H \) as an induced subgraph, i.e., let \( G \) be an instance of \( \text{RET-H} \). We construct in time polynomial in the size of \( G \), a graph \( G' \) (containing \( G \) and \( H' \) as induced subgraphs) such that the following statements (i) and (ii) are equivalent:

(i) \( G \) retracts to \( H \).
(ii) \( G' \) retracts to \( H' \).

This would imply that \( \text{RET-H} \) polynomially transforms to \( \text{RET-H}' \).

The construction of \( G' \) is as follows. We add to the subgraph \( H \) of \( G \), the vertices of \( V(H') - V(H) \) and the edges of \( E(H') - E(H) \) such that \( H \) is expanded to the graph \( H' \) (the vertices of \( H' - H \) are not adjacent to any vertex of \( G - H \)). Clearly, \( G \) retracts to \( H \) if and only if \( G' \) retracts to \( H' \). We see the mappings as follows.
Fig. 16. List of all graphs $H$ with at most four vertices for which $COMP-H$ and $RET-H$ are NP-complete.
If \( r : G \to H \) is a retraction then we can define a retraction \( r' : G' \to H' \) as follows:

\[
\begin{align*}
  r'(v) &= r(v), \text{ for all } v \in V(G), \\
  r'(h') &= h', \text{ for all } h' \in V(H') - V(H).
\end{align*}
\]

Now suppose that \( r' : G' \to H' \) is a retraction. Let \( s : H' \to H \) be a retraction. Note that \( s(r'(v)) = r'(v) \), if \( r'(v) \in V(H) \), with \( v \in V(G) \). We can define a retraction \( r : G \to H \) as follows:

\[
  r(v) = s(r'(v)), \text{ for all } v \in V(G). \quad \square
\]

It follows from Theorem 3.11 that if \( RET-H \) is NP-complete for a graph \( H \) then for any graph \( H' \) that retracts to \( H \), \( RET-H' \) is also NP-complete. There are however graphs \( H' \) with \( RET-H' \) NP-complete although \( H' \) does not retract to its induced subgraph \( H \) with \( RET-H \) NP-complete. Note that for the graph \( H \) in Fig. 6 (Fig. 1(m)), \( H \) does not retract to its induced subgraph \( S \), where \( S = h_0h_1h_2 \) with loops on \( h_0 \) and \( h_2 \) only (we know that \( RET-S \) is NP-complete) but \( RET-H \) is NP-complete (compare with Theorem 3.11). Similarly, for the graph \( H \) in Fig. 7 (Fig. 1(g)) containing an irreflexive triangle \( S = h_0h_1h_2h_0 \) as an induced subgraph, \( H \) does not retract to \( S \) (we know that \( RET-S \) is NP-complete) but \( RET-H \) is NP-complete (compare with Theorem 3.11).

4. Compaction to a partially reflexive path and other graphs

**Theorem 4.1.** Let \( H \) be a path of length \( k \geq 2 \), whose first and last vertices have loops, and no other vertex of \( H \) has a loop. Then \( COMP-H \) is NP-complete.

**Proof.** Let \( H \) be the graph in Fig. 17.

![Fig. 17. H.](image)

It is clear that \( COMP-H \) is in NP. We give a polynomial transformation from \( RET-H \) to \( COMP-H \). Recall that \( RET-H \) is NP-complete, cf. Feder, Hell, and Huang [5]. Let a graph \( G \) containing \( H \) as an induced subgraph be an instance of \( RET-H \). We construct in time polynomial in the size of \( G \), a graph \( G' \) (containing \( G \) as an induced subgraph) such that the statements (i), (ii), and (iii), as mentioned in the proof of Theorem 3.1, are equivalent. We prove that (i) is equivalent to (ii), and (ii) is equivalent to (iii), in Lemma 4.1.1 and Lemma 4.1.2 respectively.

In our discussion, we let \( k = 2n \) or \( 2n + 1 \), for some integer \( n \geq 1 \) (thus \( n = \lfloor k/2 \rfloor \)). The construction of \( G' \) is as follows. For each vertex \( v \) in \( V(G - H) \), we add to \( G \) a new vertex \( w_v \) and three vertex disjoint paths \( U_v, Y_v, \) and \( X_v \), where \( U_v \) and \( Y_v \) both contain \( k - 1 \) new vertices, and \( X_v \) contains \( n - 1 \) or \( n \) new vertices depending on whether \( k \) is even or odd respectively (note that the path \( X_v \) exists only when \( k > 2 \)). Let \( U_v = u^v_1u^v_2 \ldots u^v_{k-1}, \) \( Y_v = y^v_1y^v_2 \ldots y^v_{k-1}, \) and \( X_v = x^v_1x^v_2 \ldots x^v_p, \) where \( p = n - 1, \) if \( k \) is even, and \( p = n, \) if \( k \) is odd, with \( v \in V(G - H) \). For \( k > 2 \), we add the edges \( u^v_1w_v, u^v_1y^v_1, w_vy^v_1, u^v_1x^v_1, w_vx^v_1, h_0u^v_{k-1}, h_ky^v_{k-1}, \) and \( v^v_p, \) with \( v \in V(G - H) \). See Fig. 18. For \( k = 2, \) we add the same edges
as for $k > 2$ except that now the path $Xvv$ collapses just to the vertex $v$, and instead of the edges $u_1^v x_1^v$, $w_v x_1^v$, and $v x_p^v$, we just add the edges $u_1^v v$ and $w_v v$, with $v \in V(G - H)$. Note that $u_1^v$, $x_1^v$, and $w_v$ form a triangle, for $k > 2$; $u_1^v$, $v$, and $w_v$ form a triangle, for $k = 2$; and $u_1^v$, $w_v$, and $y_1^v$ form a triangle, for all $k \geq 2$, with $v \in V(G - H)$. This completes the construction of $G'$.

We now prove the following two lemmas in order to prove the theorem.

**Lemma 4.1.1.** $G$ retracts to $H$ if and only if $G'$ retracts to $H$.

**Proof.** If $G'$ retracts to $H$ then it is clear that $G$ retracts to $H$ since $G$ is a subgraph of $G'$. Now suppose that $r : G \rightarrow H$ is a retraction. Below, we define a retraction $r' : G' \rightarrow H$. As we go along the definition of $r'$, we shall be considering the edges $ab$ of $G'$, showing that $r'(a)r'(b)$ is an edge of $H$ (as required for $r' : G' \rightarrow H$ to be a homomorphism). Recall that the edges of $G'$ are: $ab, u_i^v u_{i+1}^v, y_i^v y_{i+1}^v, u_1^v w_v, u_1^v y_1^v, w_v y_1^v, h_0 u_{k-1}^v$, and $h_k y_{k-1}^v$, with $ab \in E(G), v \in V(G - H), i = 1, 2, \ldots, k - 2$. Further, if $k > 2$ then $G'$ also has the edges $x_i^v x_{i+1}^v, u_i^v x_1^v, w_v x_1^v$, and $v x_p^v$, with $v \in V(G - H), i = 1, 2, \ldots, p - 1$. If $k = 2$ then $G'$ also has the edges $u_1^v v$ and $w_v v$, with $v \in V(G - H)$.

For each vertex $v$ of the graph $G$, we define

$$r'(v) = r(v).$$

Thus for an edge $vv'$ of $G'$, $r'(v)r'(v') = r(v)r(v')$ is an edge of $H$, with $vv' \in E(G)$.
We now consider all the remaining edges $ab$ and $wv$. Let $r(v) = h_j$. We shall define $r'$ for the said vertices when $0 \leq j \leq n$, and when $n + 1 \leq j \leq k$.

**First assume that $0 \leq j \leq n$.**

For the vertices of $U_v$, $Y_v$, and $X_v$, and for the vertex $w_v$, we define $r'$ as follows.

$$r'(u^v_i) = h_0, \quad \text{for all } i = 1, 2, \ldots, k - 1,$$
$$r'(w_v) = h_0,$$
$$r'(y^v_i) = h_i, \quad \text{for all } i = 1, 2, \ldots, k - 1,$$

if $j = 0$ then
$$r'(x^v_i) = h_0, \quad \text{for all } i = 1, 2, \ldots, p,$$

if $j > 0$ and $k$ is even then
$$r'(x^v_i) = h_0, \quad \text{for all } i = 1, 2, \ldots, n - j,$$
$$r'(x^v_i) = h_{i-n+j}, \quad \text{for all } i = n - j + 1, n - j + 2, \ldots, n - 1,$$

if $j > 0$ and $k$ is odd then
$$r'(x^v_i) = h_0, \quad \text{for all } i = 1, 2, \ldots, n - j + 1,$$
$$r'(x^v_i) = h_{i-n+j-1}, \quad \text{for all } i = n - j + 2, n - j + 3, \ldots, n.$$

We now consider all the remaining edges $ab$ of $G'$, i.e., $ab \in E(G') - E(G)$ and prove that $r'(a)r'(b)$ is indeed an edge of $H$.

Clearly, $r'(a)r'(b)$ is an edge of $H$ where $ab$ is an edge of $G'$ among $u^v_i u^v_{i+1}$, $y^v_i y^v_{i+1}$, $x^v_s x^v_{s+1}$, for all $i = 1, 2, \ldots, k - 2, s = 1, 2, \ldots, p - 1$.

For the edges $u^v_i w_v, u^v_i y^v_i$, and $w_v y^v_i$ of $G'$, $r'(u^v_i)r'(w_v) = h_0 h_0, r'(u^v_i)r'(y^v_i) = h_0 h_1$, and $r'(w_v)r'(y^v_i) = h_0 h_2$ are respectively the edges of $H$.

For the edges $h_0 u^v_{k-1}$ and $h_k y^v_{k-1}$ of $G'$, $r'(h_0) r'(u^v_{k-1}) = h_0 h_0$ and $r'(h_k) r'(y^v_{k-1}) = h_k h_{k-1}$ are respectively the edges of $H$.

We now consider an edge $uv^v_p$ of $G'$, for $k > 2$. If $j = 0$, we have that $r'(v)r'(x^v_p) = h_0 h_0$ is an edge of $H$. If $j > 0$, we have that $r'(v)r'(x^v_p) = h_j h_{j-1}$ is an edge of $H$.

Next consider the edges $u^v_i x^v_p$ and $w_v x^v_i$ of $G'$, for $k > 2$. For even $k$, we have $r'(x^v_i) = h_1$, if $j = n$, and $r'(x^v_i) = h_0$, otherwise. For odd $k$, we have $r'(x^v_i) = h_0$ always. Hence, $r'(u^v_i)r'(x^v_i) = h_0 r'(x^v_i)$ and $r'(w_v)r'(x^v_i) = h_0 r'(x^v_i)$ are the edges of $H$.

Finally, consider the edges $u^v_i v$ and $w_v v$, for $k = 2$. We have $r(v) = h_0$ or $h_1$. Hence, $r'(u^v_i)r'(v) = h_0 r(v)$ and $r'(w_v)r'(v) = h_0 r(v)$ are the edges of $H$.

Thus we have proved that, when $0 \leq j \leq n$, $r'(a)r'(b)$ is an edge of $H$, for every edge $ab$ of $G'$.

**Now assume that $n + 1 \leq j \leq k$.**

For the vertices of $U_v$, $Y_v$, and $X_v$, and for the vertex $w_v$, we define $r'$ as follows.

$$r'(u^v_i) = h_{k-i}, \quad \text{for all } i = 1, 2, \ldots, k - 1,$$
$$r'(w_v) = h_k,$$
$$r'(y^v_i) = h_k, \quad \text{for all } i = 1, 2, \ldots, k - 1,$$
if \( j = k \) then
\[
r'(x_i^k) = h_k, \quad \text{for all } i = 1, 2, \ldots, p,
\]
if \( j < k \) and \( k \) is even then
\[
r'(x_i^k) = h_k, \quad \text{for all } i = 1, 2, \ldots, j - n,
\]
\[
r'(x_i^k) = h_{n-i+j}, \quad \text{for all } i = j - n + 1, j - n + 2, \ldots, n - 1,
\]
if \( j < k \) and \( k \) is odd then
\[
r'(x_i^k) = h_k, \quad \text{for all } i = 1, 2, \ldots, j - n,
\]
\[
r'(x_i^k) = h_{n-i+j+1}, \quad \text{for all } i = j - n + 1, j - n + 2, \ldots, n.
\]
Note that, for even \( k \), \( r'(x_i^k) = h_k \); for odd \( k \), \( r'(x_i^k) = h_{k-1} \), if \( j = n + 1 \), and \( r'(x_i^k) = h_k \), otherwise; for both even and odd \( k \), \( r'(x_i^{k-1}) = h_k \), if \( j = k \), and \( r'(x_i^{k-1}) = h_{j+1} \), if \( j < k \). Similar to the previous case when \( 0 \leq j \leq n \), it can be verified that \( r'(a)r'(b) \) is an edge of \( H \), for all \( ab \in E(G') - E(G) \).

Thus we have proved that \( r' : G' \to H \) is a homomorphism. Since \( r'(h) = r(h) = h \), for all \( h \in V(H) \), we have that \( r' : G' \to H \) is a retraction. \( \square \)

**Lemma 4.1.2.** \( G' \) retracts to \( H \) if and only if \( G' \) compact to \( H \).

**Proof.** If \( G' \) retracts to \( H \) then by definition \( G' \) compact to \( H \). Now suppose that \( c : G' \to H \) is a compactation. We first prove that \( c(h_0) \neq c(h_k) \). Suppose that \( c(h_0) = c(h_k) \). Since \( h_0 \) and \( h_k \) both have loops, it must be that \( c(h_0) = c(h_k) = h_0 \) or \( c(h_0) = c(h_k) = h_k \). Without loss of generality, let \( c(h_0) = c(h_k) = h_0 \) (due to symmetry). If \( c(a) = h_k \), for any vertex \( a \) of \( G' \), then it must be that \( d_{G'}([h_0, h_k], a) \leq d_H(c([h_0, h_k]), c(a)) = d_H(h_0, h_k) = k \). We show that there is no vertex \( a \) in \( G' \) with \( c(a) = h_k \). We have, in \( G' \), the paths \( u_1^0u_2^0\ldots u_{k-1}^0h_0, y_1^0y_2^0\ldots y_{k-1}^nh_k \), and \( h_0h_1\ldots h_{k-1} \) of length \( k - 1 \) each, which we use in our distance calculations below, with \( v \in V(G - H) \). Since \( d_{G'}(h_0, u_i^0) = k - i < d_H(c(h_0) = h_0, h_k) = k, c(u_i^0) \neq h_k \), for all \( i = 1, 2, \ldots, k - 1 \), with \( v \in V(G - H) \). Since \( d_{G'}(h_k, y_i^0) = k - i < d_H(c(h_k) = h_0, h_k) = k, c(y_i^0) \neq h_k \), for all \( i = 1, 2, \ldots, k - 1 \), with \( v \in V(G - H) \). Since \( d_{G'}(h_0, h_i) = i < d_H(c(h_0) = h_0, h_k) = k, c(h_i) \neq h_k \), for all \( i = 1, 2, \ldots, k - 1 \).

If \( c(w_v) = h_k \), for some \( v \in V(G - H) \), then \( c(u_i^0) = c(y_i^0) = h_{k-1}, as u_i^0 \) and \( y_i^0 \) are adjacent to \( w_v \), and we have shown that \( c(u_1^0) \neq h_k \) and \( c(y_1^0) \neq h_k \). But this is impossible, as also \( u_i^0 \) is adjacent to \( y_1^0 \) in \( G' \), and \( h_{k-1} \) does not have a loop. Hence \( c(w_v) \neq h_k \).

Now, for \( k = 2 \), if \( c(v) = h_k \), for some \( v \in V(G - H) \), then \( c(u_i^1) = c(w_v) = h_{k-1}, as u_i^1 \) and \( w_v \) are adjacent to \( v \) in \( G' \), and we have shown that \( c(u_1^1) \neq h_k \) and \( c(w_v) \neq h_k \). But this is impossible, as also \( u_i^1 \) is adjacent to \( w_v \) in \( G' \), and \( h_{k-1} \) does not have a loop. Hence \( c(v) \neq h_k \), if \( k = 2 \). Similarly, we show that, for \( k > 2 \), \( c(x_i^1) \neq h_k \) (recall that \( u_1^1, x_1^1, \) and \( w_v \) form a triangle in \( G' \), for \( k > 2 \), and \( u_i^1, v, \) and \( w_v \) form a triangle in \( G' \), for \( k = 2 \); in the above arguments, we just replace \( v \) by \( x_1^1 \), with \( v \in V(G - H) \). It only remains to show that, for \( k > 2 \), \( c(v) \neq h_k \) and \( c(x_i^1) \neq h_k \), for all \( i = 2, 3, \ldots, p, with v \in V(G - H) \). Since \( u_i^1, x_1^1, \) and \( w_v \) form a triangle in \( G' \), and we showed that none of these vertices map to \( h_k \) under \( c \), it must be that \( c([u_1^1, x_1^1, w_v]) = [h_0, h_1] \) or \([h_0] \), as \( h_0 \) is the only other vertex in \( H \) with a loop, with \( v \in V(G - H) \). Thus we have \( c(x_i^1) = h_0 \) or \( h_1 \), with \( v \in V(G - H) \). Since \( d_{G'}(x_i^1, x_1^1) = i - 1 < d_H(c(x_i^1) = h_0 or h_1, h_k) = k - 1 \), we have \( c(x_i^1) \neq h_k \), for all \( i = 2, 3, \ldots, p, \)
with \( v \in V(G - H) \). Since \( d_{G'}(x_1^v, v) = p < d_H(c(x_1^v) = h_0 \text{ or } h_1, h_k) \geq k - 1 \), we have \( c(v) \neq h_k \), with \( v \in V(G - H) \).

Thus, for all \( k \), we have shown that \( c(a) \neq h_k \), for all \( a \in V(G') \). Hence, it must be that \( c(h_0) \neq c(h_k) \). Since \( h_0 \) and \( h_k \) are the only vertices in \( H \) with a loop, \( c([h_0, h_k]) = \{h_0, h_k\} \). Without loss of generality, let \( c(h_0) = h_0 \) and \( c(h_k) = h_k \) (due to symmetry). This implies that \( c(h_i) = h_i \), for all \( i = 2, 3, \ldots, k - 1 \).

Thus \( c : G' \rightarrow H \) is a retraction and the lemma is proved. \( \Box \)

This completes the proof of Theorem 4.1. \( \Box \)

Now, we consider some other general graphs \( H \) for \( \text{COMP-H} \) whose complexity results follow using Theorem 4.1. Let \( H \) be a graph with several paths of the same length \( k \geq 2 \) from \( h_0 \) to \( h_k \) such that the paths are internally disjoint, where \( h_0 \) and \( h_k \) are the only vertices of \( H \) with loops, and \( H \) consists of nothing else other than these paths. See Fig. 19.

Let \( S \) be the graph in Fig. 20. The graph \( H \) has \( S \) as an induced subgraph, and clearly, \( H \) retracts to \( S \). Similar to the proof for Theorem 3.2, we can show that \( \text{COMP-H} \) is NP-complete using Theorem 4.1. Note that \( \text{RET-H} \) is also NP-complete, cf. Feder, Hell, and Huang [5].

Now, let \( H \) be a graph consisting of a path \( h_0h_1\ldots h_k \) of length \( k \geq 2 \), and a vertex \( h_{k+1} \) adjacent to every vertex of this path, where \( h_0 \) and \( h_k \) are the only vertices of \( H \) with loops, and \( H \) consists of nothing else other than the stated description. See Fig. 21. We note again that \( \text{RET-H} \) is NP-complete, cf. Feder, Hell, and Huang [5]. Similar to the proof for Theorem 3.3, we can show using Theorem 4.1 and the graph \( S \) in Fig. 20 that \( \text{COMP-H} \) is NP-complete.
5. Applications of compaction

In this section, we address three categories of applications of compaction. These are with regards to the constraint satisfaction problem, the colouring problem, and parallel computation in a multiprocessor system.

A very close relationship between the compaction problem and the constraint satisfaction problem is shown by Vikas [28]. The constraint satisfaction problem is well known to have an important role in artificial intelligence with vast applications. It is shown by Vikas [28] that the constraint satisfaction problem can be viewed as the compaction problem for reflexive as well as bipartite graphs. Similar results have been shown with respect to the retraction problem for bipartite graphs by Feder and Vardi [9] and for reflexive graphs by Feder and Hell [4]. Thus a constraint satisfaction problem for a practical application may be solved as a corresponding compaction problem.

The compaction problem is a special graph colouring problem. The colouring problem is a classic problem in graph theory, and has several applications, such as in printed circuit testing [11], storage [3], scheduling [20], etc. The graph homomorphism problem, i.e., the $H$-colouring problem, is a generalisation of the colouring problem, and known to be related to grammars and interpretations [21]. The compaction problem is the graph homomorphism problem with additional constraints. Thus applications of the colouring problem and the $H$-colouring problem with additional appropriate requirements may be formalised as the compaction problem.

We now mention an application of compaction in a multiprocessor system for parallel computation. Suppose that there are a fixed number of processors with communication links between some pairs of processors. Suppose that there are several jobs to be executed on the processors, and that some pairs of jobs need to communicate with each other during execution. It is desired that the execution of the jobs be completed in minimum possible time, and that each pair of jobs which need to communicate are allocated either the same processor or a pair of processors which have a communication link between them. Completion time may be affected due to (i) job loads on each processor, and (ii) communication delays. Note that each job may contribute a different load on the processors, and the communication delay due to a processor communicating with another processor through a com-
munication link can be expected to be more than the time taken for a processor to communicate with itself.

We find an allocation of the processors to the jobs meeting the above requirements through compaction. Define $H$ to be the graph whose vertices are processors, and there is an edge between a pair of vertices of $H$ if there is a communication link between them. Since a processor can communicate with itself, we assume that there is a loop on each vertex of $H$, i.e., $H$ is reflexive. Define $G$ to be the graph whose vertices are jobs, and there is an edge between a pair of vertices of $G$ if they need to communicate. Let $S$ be a subgraph of $H$, and $c : G \rightarrow S$ be a compaction. The graph $S$ together with the allocation of the processors to the jobs given by the mapping $c$ provides a measure of the completion time for the jobs with respect to $S$ and $c$, as governed by (i) and (ii) mentioned above; we say that the pair $(S, c)$ has the corresponding completion time (such measurements of completion time are normally only an estimate). There may be several different compactions of $G$ to $S$. Thus we meet the requirements by finding a subgraph $M$ of $H$ together with a compaction $c : G \rightarrow M$ such that $(M, c)$ has the minimum completion time. Note that we are guaranteed to find an allocation of the processors to the jobs, and hence be able to execute the jobs, as clearly $G$ compacts to a subgraph of $H$ containing only one vertex since the vertices of $H$ have loops.

Now consider the scenario related to the utilization of the multiprocessor network system as a heuristic to contribute towards the completion time of the jobs. Suppose the criteria is just to execute the jobs utilizing maximum resources provided by the multiprocessor network system. The resources in the system are the processors and the communication links. Thus it is desired to execute the jobs on as many processors as possible (which could be a heuristic to speed up execution of the jobs) using as many communication links as possible (which could be a heuristic to reduce congestion on the communication links, and hence again contribute towards the completion time of the jobs). Further, suppose it is also desired that each pair of jobs which need to communicate are allocated either the same processor or a pair of processors which have a communication link between them. This may be desired to facilitate fast communication between the jobs needing to communicate with each other, assuming that the cost of individual communication links is negligible (but the cost of more than one communication link together may not be negligible). Hence it again contributes towards the completion time of the jobs. We meet the above requirements by finding a subgraph $S$ of $H$ with maximum number of vertices and edges such that $G$ compacts to $S$, and defining a compaction $c : G \rightarrow S$ for the actual allocation of the processors to the jobs; the graph $S$ provides information on the maximum utilization of the system by the execution of the jobs meeting the requirements. As explained above, we are guaranteed to find an allocation of the processors to the jobs, and be able to execute the jobs. If we wish to see whether the system can be fully utilized by the execution of the jobs, meeting the requirements, then we simply need to see whether $G$ compacts to $H$.

Now consider the following situation. Suppose that some processors are not very efficient. Hence in addition to the above requirements, it is desired that if a processor $p$ is inefficient then not both the jobs which need to communicate with each other are allocated $p$. This may be desirable from the following point of view. Let $J_1$ and $J_2$ be two jobs which need to communicate with each other. Then $J_1$ may have to wait until $J_2$ has produced some value, and vice versa. In order that such values be already calculated and ready when needed, it may be desirable in certain situations that $J_1$ and $J_2$ not both run on the same processor to avoid congestion on the processor resource to produce those values, and hence avoid longer waiting time for jobs, especially when the processor is inefficient. Hence it is desired that $J_1$ and $J_2$ run either on the same processor that is efficient or on different processors (which have a
communication link between them); this could be a heuristic to speed up execution of the jobs. It is more useful to assume here that the cost of individual communication links is negligible (but the cost of more than one communication link together may not be negligible). With this requirement, we can assume for our purpose that if a processor $p$ is inefficient then the vertex of $H$ corresponding to $p$ does not have a loop, whereas the vertices of $H$ corresponding to efficient processors continue to have loops. Thus $H$ is a partially reflexive graph. Note that assuming that the graph $G$ is irreflexive does not affect any of the requirements. We need to assume here that $G$ is irreflexive in order to allow a job to run on an inefficient processor. Note however, if we want a particular job $J$ to run only on an efficient processor for suitability then we assume that the vertex of $G$ corresponding to $J$ has a loop and otherwise it does not have a loop. Thus $G$ may also be a partially reflexive graph. The entire requirement is again met by finding a compaction of $G$ to a subgraph $S$ of $H$, where $S$ is as described above. Note that if $H$ is irreflexive then there may exist no compaction of $G$ to any subgraph of $H$, and we may accordingly decide to have loops on certain vertices of $H$.

References