Surjective H-Colouring: New Hardness Results

Petr A. Golovach[1], Matthew Johnson[2], Barnaby Martin[2], Daniël Paulusma[2], and Anthony Stewart[2(✉)]

[1] Department of Informatics, University of Bergen, Bergen, Norway
`petr.golovach@ii.uib.no`
[2] School of Engineering and Computing Sciences,
Durham University, South Road, Durham DH1 3LE, U.K.
{`matthew.johnson2,barnaby.d.martin,daniel.paulusma,a.g.stewart`}`@dur.ac.uk`

Abstract. A homomorphism from a graph G to a graph H is a vertex mapping f from the vertex set of G to the vertex set of H such that there is an edge between vertices $f(u)$ and $f(v)$ of H whenever there is an edge between vertices u and v of G. The H-COLOURING problem is to decide whether or not a graph G allows a homomorphism to a fixed graph H. We continue a study on a variant of this problem, namely the SURJECTIVE H-COLOURING problem, which imposes the homomorphism to be vertex-surjective. We build upon previous results and show that this problem is NP-complete for every connected graph H that has exactly two vertices with a self-loop as long as these two vertices are not adjacent. As a result, we can classify the computational complexity of SURJECTIVE H-COLOURING for every graph H on at most four vertices.

1 Introduction

The well-known COLOURING problem is to decide whether or not the vertices of a given graph can be properly coloured with at most k colours for some given integer k. If we exclude k from the input and assume it is fixed, we obtain the k-COLOURING problem. A *homomorphism* from a graph $G = (V_G, E_G)$ to a graph $H = (V_H, E_H)$ is a vertex mapping $f : V_G \to V_H$, such that there is an edge between $f(u)$ and $f(v)$ in E_H whenever there is an edge between u and v in E_G. We observe that k-COLOURING is equivalent to the problem of asking whether a graph allows a homomorphism to the complete graph K_k on k vertices. Hence, a natural generalization of the k-COLOURING problem is the H-COLOURING problem, which is to decide whether or not a graph allows a homomorphism to an arbitrary fixed graph H. We call this fixed graph H the *target graph*. Throughout the paper we consider undirected graphs with no multiple edges. We assume that an input graph G contains no vertices with self-loops (we call such vertices *reflexive*), whereas a target graph H may contain such vertices. We call H *reflexive* if all its vertices are reflexive, and *irreflexive* if all its vertices are irreflexive.

For a survey on graph homomorphisms we refer the reader to the textbook of Hell and Nešetřil [10]. Here, we will discuss the H-COLOURING problem, a number of its variants and their relations to each other. In particular, we will focus on the *surjective* variant: a homomorphism f from a graph G to a graph H is *(vertex-)surjective* if f is surjective, that is, if for every vertex $x \in V_H$ there exists at least one vertex $u \in V_G$ with $f(u) = x$.

The computational complexity of H-COLOURING has been determined completely. The problem is trivial if H contains a reflexive vertex u (we can map each vertex of the input graph to u). If H has no reflexive vertices, then the Hell-Nešetřil dichotomy theorem [9] tells us that H-Colouring is solvable in polynomial time if H is bipartite and that it is NP-complete otherwise.

The LIST H-COLOURING problem takes as input a graph G and a function L that assigns to each $u \in V_G$ a list $L(u) \subseteq V_H$. The question is whether G allows a homomorphism f to the target H with $f(u) \in L(u)$ for every $u \in V_G$. Feder, Hell and Huang [2] proved that LIST H-COLOURING is polynomial-time solvable if H is a bi-arc graph and NP-complete otherwise (we refer to [2] for the definition of a bi-arc graph). A homomorphism f from G to an induced subgraph H of G is a *retraction* if $f(x) = x$ for every $x \in V_H$, and we say that G *retracts to* H. A retraction from G to H can be viewed as a list-homomorphism: choose $L(u) = \{u\}$ if $u \in V_H$, and $L(u) = V_H$ if $u \in V_G \setminus V_H$. The corresponding decision problem is called H-RETRACTION. The computational complexity of H-RETRACTION has not yet been classified. Feder et al. [3] determined the complexity of the H-RETRACTION problem whenever H is a pseudo-forest (a graph in which every connected component has at most one cycle). They also showed that H-RETRACTION is NP-complete if H contains a connected component in which the reflexive vertices induce a disconnected graph.

We impose a surjective condition on the graph homomorphism. An important distinction is whether the surjectivity is with respect to vertices or edges. Furthermore, the condition can be imposed locally or globally. If we require a graph homomorphism f to be vertex-surjective when restricted to the open neighbourhood of every vertex u of G, we say that f is an H-*role assignment*. The corresponding decision problem is called H-ROLE ASSIGNMENT and its computational complexity has been fully classified [6]. We refer to the survey of Fiala and Kratochvíl [5] for further details on locally constrained homomorphisms and from here on only consider global surjectivity.

It has been shown that deciding whether a given graph G allows a surjective homomorphism to a given graph H is NP-complete even if G and H both belong to one of the following graph classes: disjoint unions of paths; disjoint unions of complete graphs; trees; connected cographs; connected proper interval graphs; and connected split graphs [7]. Hence it is natural, just as before, to fix H which yields the following problem:

SURJECTIVE H-COLOURING
    Instance: a graph G.
    Question: does there exist a surjective homomorphism from G to H?

We emphasize that being vertex-surjective is a different condition than being edge-surjective. A homomorphism from a graph G to a graph H is called *edge-surjective* or a *compaction* if for any edge $xy \in E_H$ with $x \neq y$ there exists an edge $uv \in E_G$ with $f(u) = x$ and $f(v) = y$. Note that the edge-surjectivity condition does not hold for any self-loops $xx \in E_H$. If f is a compaction from G to H, we say that G *compacts* to H. The corresponding decision problem is known as the H-COMPACTION problem. A full classification of this problem is still wide open. However partial results are known, for example when H is a reflexive cycle, an irreflexive cycle, or a graph on at most four vertices [13–15], or when G is restricted to some special graph class [16]. Vikas also showed that whenever H-RETRACTION is polynomial-time solvable, then so is H-COMPACTION [14]. Whether the reverse implication holds is not known. A complete complexity classification of SURJECTIVE H-COLOURING is also still open. Below we survey the known results.

We first consider irreflexive target graphs H. The SURJECTIVE H-COLOURING problem is NP-complete for every such graph H if H is non-bipartite, as observed by Golovach et al. [8]. The straightforward reduction is from the corresponding H-COLOURING problem, which is NP-complete due to the aforementioned Hell-Nešetřil dichotomy theorem. However, the complexity classifications of H-COLOURING and SURJECTIVE H-COLOURING do not coincide: there exist bipartite graphs H for which SURJECTIVE H-COLOURING is NP-complete, for instance when H is the graph obtained from a 6-vertex cycle to each of which vertices we add a path of length 3 [1].

We now consider target graphs with at least one reflexive vertex. Unlike the H-COLOURING problem, the presence of a reflexive vertex does not make the SURJECTIVE H-COLOURING problem trivial to solve. We call a connected graph *loop-connected* if all its reflexive vertices induce a connected subgraph. Golovach, Paulusma and Song [8] showed that if H is a tree (in this context, a connected graph with no cycles of length at least 3) then SURJECTIVE H-COLOURING is polynomial-time solvable if H is loop-connected and NP-complete otherwise. As such the following question is natural:

Is SURJECTIVE H-COLOURING NP-complete for every connected graph H that is not loop-connected?

The reverse statement is not true (if P$\neq$ NP): SURJECTIVE H-COLOURING is NP-complete when H is the 4-vertex cycle C_4^* with a self-loop in each of its vertices. This result has been shown by Martin and Paulusma [11] and independently by Vikas, as announced in [16]. Recall also that SURJECTIVE H-COLOURING is NP-complete if H is irreflexive (and thus loop-connected) and non-bipartite.

It is known that SURJECTIVE H-COLOURING is polynomial-time solvable whenever H-COMPACTION is [1]. Recall that H-COMPACTION is polynomial-time solvable whenever H-RETRACTION is [14]. Hence, for instance, the afore-mentioned result of Feder, Hell and Huang [2] implies that SURJECTIVE H-COLOURING is polynomial-time solvable if H is a bi-arc graph. We also recall that H-RETRACTION is NP-complete whenever H is a connected graph that is not loop-connected [3]. Hence, an affirmative answer to the above question

would mean that for these target graphs H the complexities of H-Retraction, H-Compaction and Surjective H-Colouring coincide.

In Fig. 1 we display the relationships between the different problems discussed. In particular, it is a major open problem whether the computational complexities of H-Compaction, H-Retraction and Surjective H-Colouring coincide for each target graph H. Even showing this for specific cases, such as the case $H = C_4^*$, has been proven to be non-trivial. If it is true, it would relate the Surjective H-Colouring problem to a well-known conjecture of Feder and Vardi [4], which states that the $\mathcal{H}$-Constraint Satisfaction problem has a dichotomy when $\mathcal{H}$ is some fixed finite target structure and which is equivalent to conjecturing that H-Retraction has a dichotomy [4]. We refer to the survey of Bodirsky, Kara and Martin [1] for more details on the Surjective H-Colouring problem from a constraint satisfaction point of view.

Fig. 1. Relations between Surjective H-Colouring and its variants. An arrow from one problem to another indicates that the latter problem is polynomial-time solvable for a target graph H whenever the former is polynomial-time solvable for H. Reverse arrows do not hold for the leftmost and rightmost arrows, as witnessed by the reflexive 4-vertex cycle for the rightmost arrow and by any reflexive tree that is not a reflexive interval graph for the leftmost arrow (Feder, Hell and Huang [2] showed that the only reflexive bi-arc graphs are reflexive interval graphs). It is not known whether the reverse direction holds for the two middle arrows.

1.1 Our Results

We present further progress on the research question of whether Surjective H-Colouring is NP-complete for every connected graph H that is not loop-connected. We first consider the case where the target graph H is a connected graph with exactly two reflexive vertices that are non-adjacent. In Sect. 2 we prove that Surjective H-Colouring is indeed NP-complete for every such target graph H. In the same section we slightly generalize this result by showing that it holds even if the reflexive vertices of H can be partitioned into two non-adjacent sets of twin vertices. This enables us to classify in Sect. 3 the computational complexity of Surjective H-Colouring for every graph H on at most four vertices, just as Vikas [15] did for the H-Compaction problem. A classification of Surjective H-Colouring for target graphs H on at most four vertices has also been announced by Vikas in [16], and it is interesting to note that NP-hardness proofs for H-Compaction of [15] may lift to NP-hardness for Surjective H-Colouring. However, this is not true for the reflexive cycle C_4^*, where a totally new proof was required.

1.2   Future Work

To conjecture a dichotomy of SURJECTIVE H-COLOURING between P and NP-complete seems still to be difficult. Our first goal is to prove that SURJECTIVE H-COLOURING is NP-complete for every connected graph H that is not loop-connected. However, doing this via using our current techniques does not seem straightforward and we may need new hardness reductions. Another way forward is to prove polynomial equivalence between the three problems SURJECTIVE H-COLOURING, H-COMPACTION and H-RETRACTION. However, completely achieving this goal also seems far from trivial. Our classification for target graphs H up to four vertices does show such an equivalence for these cases (see Sect. 3).

2   Two Non-adjacent Reflexive Vertices

We say that a graph is 2-*reflexive* if it contains exactly 2 reflexive vertices *that are non-adjacent*. In this section we will prove that SURJECTIVE H-COLOURING is NP-complete whenever H is connected and 2-reflexive. The problem is readily seen to be in NP. Our NP-hardness reduction uses similar ingredients as the reduction of Golovach, Paulusma and Song [8] for proving NP-hardness when H is a tree that is not loop-connected. There are, however, a number of differences. For instance, we will reduce from a factor cut problem instead of the less general matching cut problem used in [8]. We will explain these two problems and prove NP-hardness for the former one in Sect. 2.1. Then in Sect. 2.2 we give our hardness reduction, and in Sect. 2.3 we extend our result to be valid for target graphs H with more than two reflexive vertices as long as these reflexive vertices can be partitioned into two non-adjacent sets of twin vertices.

2.1   Factor Cuts

Let $G = (V_G, E_G)$ be a connected graph. For $v \in V_G$ and $E \subseteq E_G$, let $d_E(v)$ denote the number of edges of E incident with v. For a partition (V_1, V_2) of V_G, let $E_G(V_1, V_2)$ denote the set of edges between V_1 and V_2 in G.

Let i and j be positive integers, $i \leq j$. Let (V_1, V_2) be a partition of V_G and let $M = E_G(V_1, V_2)$. Then (V_1, V_2) is an (i, j)-*factor cut* of G if, for all $v \in V_1$, $d_M(v) \leq i$, and, for all $v \in V_2$, $d_M(v) \leq j$. Observe that if a vertex v exists with degree at most j, then there is a trivial (i, j)-factor cut $(V \setminus \{v\}, \{v\})$. Two distinct vertices s and t in V_G are (i, j)-*factor roots* of G if, for each (i, j)-factor cut (V_1, V_2) of G, s and t belong to different parts of the partition and, if $i < j$, $s \in V_1$ and $t \in V_2$ (of course, if $i = j$, we do not require the latter condition as (V_2, V_1) is also an (i, j)-factor cut). We note that when no (i, j)-factor cut exists, every pair of vertices is a pair of (i, j)-factor roots. We define the following decision problem.

(i, j)-FACTOR CUT WITH ROOTS
    Instance: a connected graph G with (i, j)-factor roots s and t.
    Question: does G have an (i, j)-factor cut?

We emphasize that the (i, j)-factor roots are given as part of the input. That is, the problem asks whether or not an (i, j)-factor cut (V_1, V_2) exists, but we know already that if it does, then s and t belong to different parts of the partition. That is, we actually define (i, j)-FACTOR CUT WITH ROOTS to be a promise problem in which we assume that if an (i, j)-factor cut exists then it has the property that s and t belong to different parts of the partition. The promise class may not itself be polynomially recognizable but one may readily find a subclass of it that is polynomially recognizable and includes all the instances we need for NP-hardness. In fact this will become clear when reading the proof of Theorem 1 but we refer also to [8] where such a subclass is given for the case $(i, j) = (1, 1)$. A $(1, 1)$-factor cut (V_1, V_2) of G is also known as a *matching cut*, as no two edges in $E_G(V_1, V_2)$ have a common end-vertex, that is, $E_G(V_1, V_2)$ is a *matching*. Similarly $(1, 1)$-FACTOR CUT WITH ROOTS is known as MATCHING CUT WITH ROOTS and was proved NP-complete by Golovach, Paulusma and Song [8] (by making an observation about the proof of the result of Patrignani and Pizzonia [12] that deciding whether or not any given graph has a matching cut is NP-complete). The proof of Theorem 1 has been omitted.

Theorem 1. *Let i and j be positive integers, $i \leq j$. Then (i, j)-FACTOR CUT WITH ROOTS is* NP-*complete.*

2.2   The Hardness Reduction

Let H be a connected 2-reflexive target graph. Let p and q be the two (non-adjacent) reflexive vertices of H. The *length* of a path is its number of edges. The *distance* between two vertices u and v in a graph G is the length of a shortest path between them and is denoted $\mathrm{dist}_G(u, v)$. We define two induced subgraphs H_1 and H_2 of H whose vertex sets partition V_H. First H_1 contains those vertices of H that are closer to p than to q; and H_2 contains those vertices that are at least as close to q as to p (so contains any vertex equidistant to p and q). That is, $V_{H_1} = \{v \in V_H : \mathrm{dist}_H(v, p) < \mathrm{dist}_H(v, q)\}$ and $V_{H_2} = \{v \in V_H : \mathrm{dist}_H(v, q) \leq \mathrm{dist}_H(v, p)\}$. See Fig. 2 for an example. The following lemma follows immediately from our assumption that H is connected.

Lemma 1. *Both H_1 and H_2 are connected. Moreover, $\mathrm{dist}_{H_1}(x, p) = \mathrm{dist}_H(x, p)$ for every $x \in V_{H_1}$ and $\mathrm{dist}_{H_2}(x, q) = \mathrm{dist}_H(x, q)$ for every $x \in V_{H_2}$.*

A *clique* is a subset of vertices of G that are pairwise adjacent to each other. Let ω denote the size of a largest clique in H.

From graphs H_1 and H_2 we construct graphs F_1 and F_2, respectively, in the following way:

1. for each $x \notin \{p, q\}$, create a vertex t_x^1;
2. for p, create ω vertices $t_p^1, \ldots, t_p^\omega$;
3. for q, create ω vertices $t_q^1, \ldots, t_q^\omega$;
4. for $i = 1, 2$, add an edge in F_i between any two vertices t_x^h and t_y^j if and only if xy is an edge of E_{H_i}.
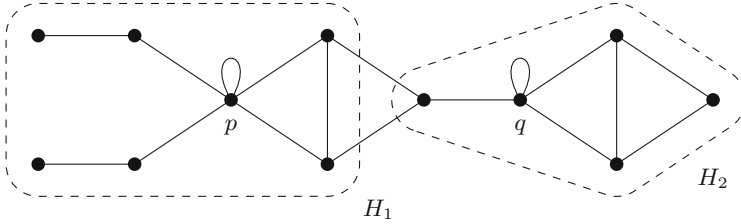
Fig. 2. An example of the construction of graphs H_1 and H_2 from a connected 2-reflexive target graph H with $\omega = 3$.

We observe that since p and q are reflexive, there are edges pp and qq, hence $t_p^1, \ldots, t_p^\omega$ and $t_q^1, \ldots, t_q^\omega$ form cliques of size ω. Note also that F_1 is the graph obtained by taking H_1 and replacing p by a clique of size ω. Similarly, F_2 is the graph obtained by taking H_2 and replacing q by a clique of size ω. We say that $t_p^1, \ldots, t_p^\omega$ are the roots of F_1 and that $t_q^1, \ldots, t_q^\omega$ are the roots of F_2. Figure 3 shows an example of the graphs F_1 and F_2 obtained from the graph H in Fig. 2.
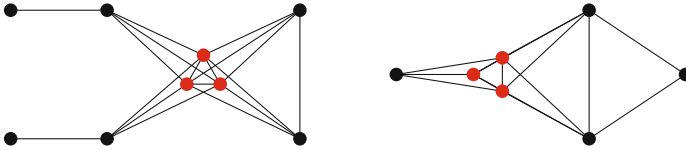
Fig. 3. The graphs F_1 (left) and F_2 (right) resulting from the graph H in Fig. 2.

Let $\ell = \operatorname{dist}_H(p, q) \geq 2$ denote the distance between p and q. Let N_p be the set of neighbours of p that are each on some shortest path (thus of length ℓ) from p to q in H. Let r_p be the size of a largest clique in N_p. We define N_q and r_q similarly. We will reduce from (r_p, r_q)-FACTOR CUT WITH ROOTS, which is NP-complete due to Theorem 1. Hence, consider an instance (G, s, t) of (r_p, r_q)-FACTOR CUT WITH ROOTS, where G is a connected graph and s and t form the (ordered) pair of (r_p, r_q)-factor roots of G. Recall that we assume that G is irreflexive.

We say that we *identify* two vertices u and v of a graph when we remove them from the graph and replace them with a single vertex that we make adjacent to every vertex that was adjacent to u or v. From F_1, F_2, and G we construct a new graph G' as follows:

1. For each edge $e = uv \in E_G$, we do as follows. We create four vertices, $g_{u,e}^r$, $g_{u,e}^b$, $g_{v,e}^r$ and $g_{v,e}^b$. We also create two paths P_e^1 and P_e^2, each of length $\ell - 2$, between $g_{u,e}^r$ and $g_{u,e}^b$, and between $g_{v,e}^r$ and $g_{u,e}^b$, respectively. If $\ell = 2$ we identify $g_{u,e}^r$ and $g_{v,e}^b$ and $g_{v,e}^r$ and $g_{u,e}^b$ to get paths of length 0.

2. For each vertex $u \in V_G$, we do as follows. First we construct a clique C_u on ω vertices. We denote these vertices by $g_u^1, \ldots, g_u^\omega$. We then make every vertex in C_u adjacent to both $g_{u,e}^r$ and $g_{u,e}^b$ for every edge e incident to u; we call $g_{u,e}^r$ and $g_{u,e}^b$ a *red* and *blue* neighbour of C_u, respectively; if $\ell = 2$, then the vertex obtained by identifying two vertices $g_{u,e}^r$ and $g_{v,e}^b$, or $g_{v,e}^r$ and $g_{u,e}^b$ is simultaneously a red neighbour of one clique and a blue neighbour of another one. Finally, for every two edges e and e' incident to u, we make $g_{u,e}^r$ and $g_{u,e'}^r$ adjacent, that is, the set of red neighbours of C_u form a clique, whereas the set of blue neighbours form an independent sets.

3. We add F_1 by identifying t_p^i and g_s^i for $i = 1, \ldots, \omega$, and we add F_2 by identifying t_q^i and g_t^i for $i = 1, \ldots, \omega$. We denote the vertices in F_1 and F_2 in G' by their label t_x^i in F_1 or F_2.

See Fig. 4 for an example of a graph G'.
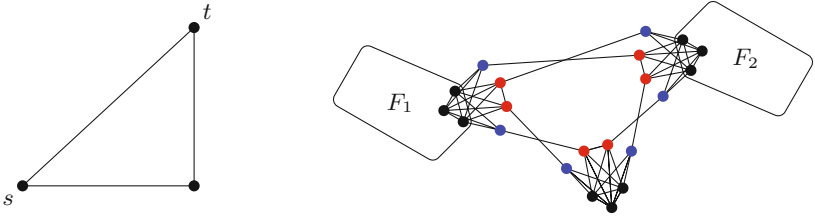
(a) An example of a graph G with a $(1,2)$-factor cut with $(1,2)$-factor roots s and t.

(b) The corresponding graph G' where H is a 2-reflexive target graph with $\ell = 3$ and $\omega = 3$.

Fig. 4. An example of a graph G and the corresponding graph G'. (Color figure online)

The next lemma describes a straightforward property of graph homomorphisms that will prove useful.

Lemma 2. *If there exists a homomorphism* $h : G' \to H$ *then* $\text{dist}_{G'}(u, v) \geq \text{dist}_H(h(u), h(v))$ *for every pair of vertices* $u, v \in V_{G'}$.

Here is the key property of our construction (proof omitted).

Lemma 3. *For every homomorphism* h *from* G' *to* H*, there exists at least one clique* C_a *with* $p \in h(C_a)$ *and at least one clique* C_b *with* $q \in h(C_b)$.

Proof Sketch. Since for each $u \in V_G$ and any edge e incident to u, every clique $C_u \cup \{g_{u,e}^r\}$ in G' is of size at least $\omega + 1$, we find that h must map at least two of its vertices to a reflexive vertex, so either to p or q. Hence, for every $u \in V_G$, we find that h maps at least one vertex of C_u to either p or q.

We prove the lemma by contradiction. We will assume that h does not map any vertex of any C_u to q, thus $p \in h(C_u)$ for all $u \in V_G$. We will note later that if instead $q \in h(C_u)$ for all $u \in V_G$ we can obtain a contradiction in the same way.

We consider two vertices $t_p^i \in F_1$ and $t_q^j \in F_2$ such that $h(t_p^i) = h(t_q^j) = p$. Without loss of generality let $i = j = 1$. We shall refer to these vertices as t_p and t_q respectively. We now consider a vertex $v \in V_{F_1} \cup V_{F_2}$. By Lemma 2, $\text{dist}_{G'}(v, t_p) \geq \text{dist}_H(h(v), p)$ and $\text{dist}_{G'}(v, t_q) \geq \text{dist}_H(h(v), p)$. In other words:

$$\min\left(\text{dist}_{G'}(v, t_p), \text{dist}_{G'}(v, t_q)\right) \geq \text{dist}_H(h(v), p).$$

In fact by applying Lemma 2 we can generalize this further to any vertex mapped to p by h:

$$\min_{w \in h^{-1}(p)}\left(\text{dist}_{G'}(v, w)\right) \geq \text{dist}_H(h(v), p). \tag{1}$$

For every $v \in V_{G'}$ we define a value $\mathcal{D}(v)$ as follows:

$$\mathcal{D}(v) = \begin{cases} \text{dist}_{F_1}(v, t_p) & \text{if } v \in F_1 \\ \text{dist}_{F_2}(v, t_q) & \text{if } v \in F_2 \\ \lfloor \ell/2 \rfloor & \text{otherwise} \end{cases}$$

As $h(t_p) = h(t_q) = p$ and any vertex not in $F_1 \cup F_2$ is either in a clique or on a path of length ℓ between two cliques, we can use inequality (1) to prove that the following inequality holds for any distance $d \geq \ell$:

$$\left|\{t_w^1 \in V_{F_1} \cup V_{F_2} : \mathcal{D}(t_w^1) \geq d\}\right| \geq \left|\{w \in V_H : \text{dist}_H(w, p) \geq d\}\right|. \tag{2}$$

In the remainder we only present the intuition behind the final part of the proof. Consider the graphs F_1, F_2 and H in the example shown in Fig. 5. We recall that every vertex v (other than p or q) has a single corresponding vertex t_v in F_1 or F_2. We may naturally want to map the vertices of F_1 onto the vertices of H_1,
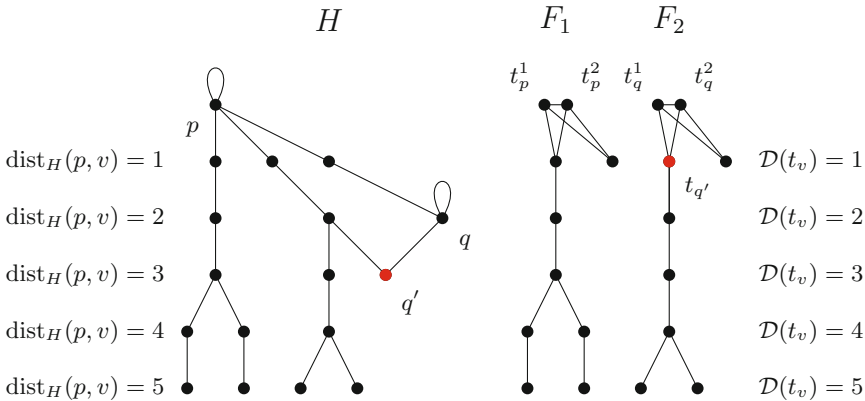
Fig. 5. An example of a graph H with corresponding graphs F_1 and F_2. Vertices in H equidistant from p are plotted at the same vertical position and likewise vertices $t_v \in F_1$ and $t_w \in F_2$ with $\mathcal{D}(t_v) = \mathcal{D}(t_w)$ are plotted at the same vertical position. The vertices $q' \in H$ and corresponding $t_{q'} \in F_2$ are highlighted.

which is possible by definition of F_1. However, when we try to map the vertices of F_2 onto the vertices of H_2, with $h(t_q^i) = p$ (for some i), we will prove that there is at least one vertex q' in H_2 which is further from p in H than it is from q and that cannot be mapped to and thus violates the surjectivity constraint. In Fig. 5 this vertex, which will play a special role in our proof, is shown in red. In the example of this figure, $\ell = 3$ and we observe that there are ten vertices (including q') in H with $\text{dist}_H(p, v) \geq 3$ but only nine vertices (excluding q') in $F_1 \cup F_2$ with $\mathcal{D}(t_v) \geq 3$ which could be mapped to these vertices. This contradicts inequality (2). $\qquad\square$

We are now ready to state and prove our main result.

Theorem 2. *For every connected 2-reflexive graph H, the* SURJECTIVE H-COLOURING *problem is* NP-*complete.*

Proof. Let H be a connected 2-reflexive graph with reflexive vertices p and q at distance $\ell \geq 2$ from each other. Let ω be the size of a largest clique in H. We define the graphs H_1, H_2, F_1 and F_2 and values r_p, r_q as above. Recall that the problem is readily seen to be in NP and that we reduce from (r_p, r_q)-FACTOR CUT WITH ROOTS. From F_1, F_2 and an instance (G, s, t) of the latter problem we construct the graph G'. We claim that G has an (r_p, r_q)-factor cut (V_1, V_2) if and only if there exists a surjective homomorphism h from G' to H.

First suppose that G has an (r_p, r_q)-factor cut (V_1, V_2). By definition, $s \in V_1$ and $t \in V_2$. We define a homomorphism h as follows. For every $x \in V_{F_1} \cup V_{F_2}$, we let h map t_x^1 to x. This shows that h is surjective. It remains to define h on the other vertices. For every $u \in V_G$, let h map all of C_u to p if u is in V_1 and let h map all of C_u to q if u is in V_2 (note that this is consistent with how we defined h so far). For each $uv \in E_G$ with $u, v \in V_1$, we map the vertices of the paths P_e^1 and P_e^2 to p. For each $uv \in E_G$ with $u, v \in V_2$, we map the vertices of the paths P_e^1 and P_e^2 to q. We are left to show that the vertices of the remaining paths P_e^1 and P_e^2 can be mapped to appropriate vertices of H.

Note that the red neighbours of each C_u form a clique (whereas all blue vertices of each C_u form an independent set and inner vertices of paths P_e^1 and P_e^2 have degree 2). However, as (V_1, V_2) is an (r_p, r_q)-factor cut of G, all but at most r_p vertices of these red cliques have been mapped to p already if $u \in V_1$ and all but at most r_q vertices have been mapped to q already if $u \in V_2$. By definition of r_p and r_q, this means that we can map the vertices of the paths P_e^1 and P_e^2 with $e = uv$ for $u \in V_1$ and $v \in V_2$ to vertices of appropriate shortest paths between p and q in H, so that h is a homomorphism from G' to H (recall that we already showed surjectivity). In particular, the clique formed by the red neighbours of each C_u is mapped to a clique in $N_p \cup \{p\}$ or $N_q \cup \{q\}$.

Now suppose that there exists a surjective homomorphism h from G' to H. For a clique C_u, we may choose any edge e incident to u, such that $C_u' = C_u \cup \{g_{u,e}^r\}$ is a clique of size $\omega + 1$. Since H contains no cliques larger than ω, we find that h maps each clique C_u' (which has size $\omega + 1$) to a clique in H that contains a reflexive vertex. Note that at least two vertices of C_u' are mapped to a reflexive vertex. Hence we can define the following partition of V_G. We let

$V_1 = \{v \in V_G : p \in h(C_v)\}$ and $V_2 = V_G \setminus V_1 = \{v \in V_G : q \in h(C_v)\}$. Lemma 3 tells us that $V_1 \neq \emptyset$ and $V_2 \neq \emptyset$. We define $M = \{uv \in E_G : u \in V_1, v \in V_2\}$.

Let $e = uv$ be an arbitrary edge in M. By definition, h maps all of C_u to a clique containing p and all of C_v to a clique containing q. Hence, the vertices of the two paths P_e^1 and P_e^2 must be mapped to the vertices of a shortest path between p and q. At most r_p red neighbours of every C_u with $u \in V_1$ can be mapped to a vertex other than p. This is because these red neighbours form a clique. As such they must be mapped onto vertices that form a clique in H. As such vertices lie on a shortest path from p to q, the clique in H has size at most r_p. Similarly, at most r_q red neighbours of every C_u with $u \in V_2$ can be mapped to a vertex other than q. As such, (V_1, V_2) is an (r_p, r_q)-factor cut in G.   □

2.3   A Small Extension

Two vertices u and v in a graph G are *true twins* if they are adjacent to each other and share the same neighbours in $V_G \setminus \{u, v\}$. Let $H^{(i,j)}$ be a graph obtained from a connected 2-reflexive graph H with reflexive vertices p and q after introducing i reflexive true twins of p and j reflexive true twins of q. In the graph G' we increase the cliques C_u to size $\omega + \max(i, j)$. We call the resulting graph G''. Then it is readily seen that there exists a surjective homomorphism from G' to H if and only if there exists a surjective homomorphism from G'' to $H^{(i,j)}$.

Theorem 3. *For every connected 2-reflexive graph H and integers $i, j \geq 0$,* SURJECTIVE $H^{(i,j)}$-COLOURING *is* NP-*complete.*

3   Target Graphs of at Most Four Vertices

For proving Theorem 4 we use Theorem 3 combined with known results [11,15]; we omit the details. In Fig. 6, the three graphs C_4^*, D and paw* are displayed.
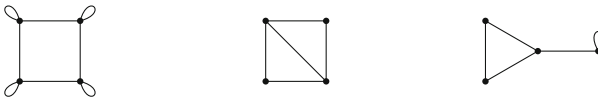
Fig. 6. The graphs C_4^*, D and paw*.

Theorem 4. *Let H be a graph with $|V_H| \leq 4$. Then* SURJECTIVE H-COLOURING *is* NP-*complete if some connected component of H is not loop-connected or is an irreflexive complete graph on at least three vertices, or $H \in \{C_4^*, D, \text{paw}^*\}$. Otherwise* SURJECTIVE H-COLOURING *is polynomial-time solvable.*

Theorem 4 corresponds to Vikas' complexity classification of H-Compaction for targets graphs H of at most four vertices. Vikas [15] showed that H-Compaction and H-Retraction are polynomially equivalent for target graphs H of at most four vertices. Thus, we obtain the following corollary.

Corollary 1. *Let H be a graph on at most four vertices. Then the three problems* Surjective H-Colouring*,* H-Compaction *and* H-Retraction *are polynomially equivalent.*

References

1. Bodirsky, M., Kára, J., Martin, B.: The complexity of surjective homomorphism problems - a survey. Discrete Appl. Math. **160**, 1680–1690 (2012)
2. Feder, T., Hell, P., Huang, J.: Bi-arc graphs and the complexity of list homomorphisms. J. Graph Theory **42**, 61–80 (2003)
3. Feder, T., Hell, P., Jonsson, P., Krokhin, A., Nordh, G.: Retractions to pseudo-forests. SIAM J. Discrete Math. **24**, 101–112 (2010)
4. Feder, T., Vardi, M.Y.: The computational structure of monotone monadic SNP and constraint satisfaction: a study through datalog and group theory. SIAM J. Comput. **28**, 57–104 (1998)
5. Fiala, J., Kratochvíl, J.: Locally constrained graph homomorphisms - structure, complexity, and applications. Comput. Sci. Rev. **2**, 97–111 (2008)
6. Fiala, J., Paulusma, D.: A complete complexity classification of the role assignment problem. Theoret. Comput. Sci. **349**, 67–81 (2005)
7. Golovach, P.A., Lidický, B., Martin, B., Paulusma, D.: Finding vertex-surjective graph homomorphisms. Acta Informatica **49**, 381–394 (2012)
8. Golovach, P.A., Paulusma, D., Song, J.: Computing vertex-surjective homomorphisms to partially reflexive trees. Theoret. Comput. Sci. **457**, 86–100 (2012)
9. Hell, P., Nešetřil, J.: On the complexity of H-colouring. J. Comb. Theory Ser. B **48**, 92–110 (1990)
10. Hell, P., Nešetřil, J.: Graphs and Homomorphisms. Oxford University Press, Oxford (2004)
11. Martin, B., Paulusma, D.: The computational complexity of disconnected cut and $2K_2$-partition. J. Comb. Theory Ser. B **111**, 17–37 (2015)
12. Patrignani, M., Pizzonia, M.: The complexity of the matching-cut problem. In: Brandstädt, A., Le, V.B. (eds.) WG 2001. LNCS, vol. 2204, pp. 284–295. Springer, Heidelberg (2001). doi:10.1007/3-540-45477-2_26
13. Vikas, N.: Computational complexity of compaction to reflexive cycles. SIAM J. Comput. **32**, 253–280 (2002)
14. Vikas, N.: Compaction, retraction, and constraint satisfaction. SIAM J. Comput. **33**, 761–782 (2004)
15. Vikas, N.: A complete and equal computational complexity classification of compaction and retraction to all graphs with at most four vertices and some general results. J. Comput. Syst. Sci. **71**, 406–439 (2005)
16. Vikas, N.: Algorithms for partition of some class of graphs under compaction and vertex-compaction. Algorithmica **67**, 180–206 (2013)