Computability 0 (0) 1
IOS Press

Surjective H-Colouring: New Hardness Results[1]

Petr A. Golovach
Department of Informatics, University of Bergen, Norway
petr.golovach@ii.uib.no

Matthew Johnson
Department of Computer Science, Durham University, U.K.
matthew.johnson@durham.ac.uk

Barnaby Martin
Department of Computer Science, Durham University, U.K.
barnaby.d.martin@durham.ac.uk

Daniël Paulusma *
Department of Computer Science, Durham University, U.K.
daniel.paulusma@durham.ac.uk

Anthony Stewart
Department of Computer Science, Durham University, U.K.
a.g.stewart@durham.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 if 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.

Keywords: graph homomorphism, vertex surjectivity, computational complexity

1. Introduction

The well-known COLOURING problem is to decide if 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 H whenever there is an edge between u and v in G, that is, for every pair of vertices $u, v \in V_G$, if $uv \in E_G$, then $f(u)f(v) \in E_H$. We observe that k-COLOURING is equivalent to the problem of asking if 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 asks if a given 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 [14]. 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 $f(u) = x$.

[1]Supported by the Research Council of Norway via the project “CLASSIS” and the Leverhulme Trust (RPG-2016-258).

*Corresponding author. daniel.paulusma@durham.ac.uk.

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 [13] 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 [5] proved that LIST H-COLOURING is polynomial-time solvable if H is a bi-arc graph and NP-complete otherwise (we refer to [5] 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. [6] 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.

As mentioned, we impose a (vertex-)surjectivity condition on the graph homomorphism. Such a condition can be imposed locally or globally. If we require a homomorphism f from a graph G to a graph H to be surjective when restricted to the neighbourhood $\{v \mid uv \in E_G\}$ 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 [9]. We refer to the survey of Fiala and Kratochvíl [8] 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 [11]. 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 we are considering vertex-surjectivity and 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 [19, 21, 22], or when G is restricted to some special graph class [18]. Vikas also showed that whenever H-RETRACTION is polynomial-time solvable, then so is H-COMPACTION [21]. 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. [12]. 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], or when H is the 6-vertex cycle itself [20].

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

Figure 1. Relations between 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 if 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 [5] showed that the only reflexive bi-arc graphs are reflexive interval graphs). It is not known if the reverse direction holds for the two middle arrows.

Song [12] 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 [16] and independently by Vikas, as announced in [18]. 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 [21]. Hence, for instance, the aforementioned result of Feder, Hell and Huang [5] 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 [6]. 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 Figure 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 the well-known conjecture of Feder and Vardi [7], recently proved by Bulatov [2] and Zhuk [23], 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 [7].

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 and to a recent paper of Larose, Martin and Paulusma [15] for some initial results on Surjective H-Colouring for directed graphs.

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 Section 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 Section 3 the computational complexity of Surjective H-Colouring for every graph H on at most four vertices, just as Vikas [22] 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 [18]. As we will illustrate for one particular case, it is interesting to note that NP-hardness proofs for H-Compaction of [22] 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.

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 [12] 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 [12]. We will explain these two problems and prove NP-hardness for the former one in Section 2.1. Then in Section 2.2 we give our hardness reduction, and in Section 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, $i \leqslant j$. 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 $v \in V_1$, $d_M(v) \leqslant i$, and, for all $v \in V_2$, $d_M(v) \leqslant j$. 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 such a problem the input is "promised" to belong to some specific subset of inputs; we refer to the survey of Goldreich [10] for details. In the definition of (i, j)-FACTOR CUT WITH ROOTS, 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 our proof but we refer also to [12] 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 [12] (by making an observation about the proof of the result of Patrignani and Pizzonia [17] that deciding whether or not any given graph has a matching cut is NP-complete).

We will prove the NP-completeness of (i, j)-FACTOR CUT WITH ROOTS after first presenting a helpful lemma (a *clique* is a subset of vertices of G that are pairwise adjacent to each other).

Lemma 1. *Let i, j and k be positive integers where $i \leqslant j$ and $k > i + j$. Let G be a graph that contains a clique K on k vertices. Then, for every (i, j)-factor cut (V_1, V_2) of G, either $V_K \subseteq V_1$ or $V_K \subseteq V_2$.*

Proof. If the lemma is false, then for some (i, j)-factor cut (V_1, V_2), we can choose $v_1 \in V_1 \cap V_K$ and $v_2 \in V_2 \cap V_K$. Let $M = E_G(V_1, V_2)$. Since every vertex in $V_1 \cap V_K$ is linked by an edge of M to v_2 and every vertex in $V_2 \cap V_K$ is linked by an edge of M to v_1, we have $d_M(v_1) + d_M(v_2) \geqslant k > i + j$, contradicting the definition of an (i, j)-factor cut. □

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

Proof. If $i = j = 1$, then the problem is MATCHING CUT WITH ROOTS which, as we noted, is known to be NP-complete [12]. We split the remaining cases in two according to whether or not $i = 1$. In each case, we construct a polynomial time reduction from MATCHING CUT WITH ROOTS. In particular, we take an instance (G, s, t) of MATCHING CUT WITH ROOTS, and construct a graph G' that is a supergraph of $G = (V, E)$ and show that

(1) (G', s, t) is an instance of (i, j)-FACTOR CUT WITH ROOTS (that is, if G' has an (i, j)-factor cut (V_1', V_2'), then $s \in V_1$ and $t \in V_2$ or, possibly, vice versa if $i = j$),
(2) if G' has an (i, j)-factor cut, then G has a matching cut, and
(3) if G has a matching cut, then G' has an (i, j)-factor cut.

We note that (1) is an atypical feature of an NP-completeness proof. We need to prove (1), as (i, j)-FACTOR CUT WITH ROOTS is a promise problem. Hence it is not immediate to recognize an instance of it. We let $n = |V|$.

Case 1: $i = 1$.
Let $k = \max\{(n-1)(j-1), 1 + j\}$. Construct G' from G by first adding a complete graph K on k vertices and adding edges to G that go from s to every vertex of V_K. Then, for each $v \in V_G \setminus \{s\}$, add edges to G that go from v to $j - 1$ vertices of K in such a way that no vertex of V_K has more than one neighbour in $V_G \setminus \{s\}$.

Let (V_1', V_2') be a $(1, j)$-factor cut of G'. The vertices of $\{s\} \cup V_K$ induce a clique on $1 + k > 1 + j$ vertices. So, by Lemma 1, $\{s\} \cup V_K \subseteq V_1'$ or $\{s\} \cup V_K \subseteq V_2'$.

Suppose that $\{s\} \cup V_K \subseteq V_2'$. Then V_G must contain vertices of both V_1' (otherwise V_1' would be empty) and V_2' (at least s). Thus, as G is connected, we can find a vertex $v \in V_1' \cap V_G$ that has a neighbour in $V_2' \cap V_G$. But v also has $j - 1 \geqslant 1$ neighbours in V_K and so has at least 2 neighbours in V_2', contradicting the definition of a $(1, j)$-factor cut.

So we must have that $\{s\} \cup V_K \subseteq V_1'$. Let $V_1 = V_1' \cap V_G$ and $V_2 = V_2'$ be a partition of V_G, and let $M = E_G(V_1, V_2)$ and $M' = E_G(V_1', V_2')$ and notice that M' is the union of M and, for each $v \in V_2$, the $j - 1$ edges from v to V_K. For each $v \in V_1$, $d_M(v) = d_{M'}(v) \leqslant 1$. For each $v \in V_2$, $d_M(v) = d_{M'}(v) - (j - 1) \leqslant 1$. So (V_1, V_2) is a matching cut of G; this proves (2). And as $s \in V_1$, we have, by the definition of factor roots, $t \in V_2$; this proves (1).

To prove (3), we note that if (V_1, V_2) is a matching cut of G, then we can assume that $s \in V_1$ and $t \in V_2$ (else relabel them for the purpose of constructing G'), and then $(V_1 \cup V_K, V_2)$ is a $(1, j)$-factor cut of G'.

Case 2: $i \geqslant 2$.
Let $k = \max\{(n-1)(j-1), i + j\}$. Construct G' from G by first adding a complete graph K^s on k vertices and adding edges from s to every vertex of V_{K^s}, and then adding a complete graph K^t on k vertices and adding edges from t to every vertex of V_{K^t}. Then, for each $v \in V_G \setminus \{s\}$, add edges from v to $j - 1$ vertices of K^s in such a way that no vertex of V_{K^s} has more than one neighbour in $V_G \setminus \{s\}$. Afterwards, for each $v \in V_G \setminus \{t\}$, add edges from v to $i - 1$ vertices of K^t in such a way that no vertex of V_{K^t} has more than one neighbour in $V_G \setminus \{t\}$.

Let (V_1', V_2') be an (i, j)-factor cut of G'. The vertices of $\{s\} \cup V_{K^s}$ induce a clique on at least $1 + k > i + j$ vertices. So, by Lemma 1, $\{s\} \cup V_{K^s} \subseteq V_1'$ or $\{s\} \cup V_{K^s} \subseteq V_2'$. Similarly $\{t\} \cup V_{K^t} \subseteq V_1'$ or $\{t\} \cup V_{K^t} \subseteq V_2'$.

Suppose that $\{s\} \cup V_{K^s}$ and $\{t\} \cup V_{K^t}$ are both subsets of V_1'. Then V_G must contain vertices of both V_1' (at least s and t) and V_2' (else it would be empty). Thus, as G is connected, we can find a vertex $v \in V_2' \cap V_G$ that has a neighbour in $V_1' \cap V_G$. But v also has $j - 1$ neighbours in V_{K^s} and $i - 1$ neighbours in V_{K^t} and so has at least $1 + (i - 1) + (j - 1) = i + j - 1 > j \geqslant i$ neighbours in V_2', contradicting the definition of an (i, j)-factor. By an analogous argument $\{s\} \cup V_{K^s}$ and $\{t\} \cup V_{K^t}$ cannot both be subsets of V_2'.

Suppose that $i < j$ and $\{s\} \cup V_{K^s} \subseteq V_2'$. As G is connected and V_G contains vertices of both V_1' and V_2', we can find a vertex $v \in V_1' \cap V_G$ that has a neighbour in $V_2' \cap V_G$. But v also has $j - 1 > i - 1$ neighbours in V_{K^s} and so has more than i neighbours in V_2', contradicting the definition of a (i, j)-factor.

Thus we have that $\{s\} \cup V_{K^s}$ and $\{t\} \cup V_{K^t}$ are subsets of separate parts and, moreover, either $\{s\} \cup V_{K^s} \subseteq V_1'$ or $i = j$. Thus (1) is proved, and we have, in either case, that each vertex in $V_1' \cap V_G$ is joined by $i - 1$ edges to vertices in $V_2' \setminus V_G$, and each vertex in $V_2' \cap V_G$ is joined by $j - 1$ edges to vertices in $V_1' \setminus V_G$. Therefore each vertex in $V_1' \cap V_G$ is joined to at most one vertex in $V_2' \cap V_G$, and each vertex in $V_2' \cap V_G$ is joined to at most one vertex in $V_1' \cap V_G$. Thus $(V_1' \cap V_G, V_2' \cap V_G)$ is a matching cut of G. This proves (2).

To prove (3), we note that if (V_1, V_2) is a matching cut of G, then we can assume that $s \in V_1$ and $t \in V_2$ (else relabel them for the purpose of constructing G'), and then $(V_1 \cup V_{K^s}, V_2 \cup V_{K^t})$ is an (i, j)-factor cut of G'. □

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 $V_{H_2} = \{v \in V_H : \mathrm{dist}_H(v,q) \leqslant \mathrm{dist}_H(v,p)\}$. See Figure 2 for an example.

We need the following lemma.

Lemma 2. *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 $x \in V_{H_2}$.*

Proof. The statement of the lemma follows immediately from our assumption that H is connected and the definitions of H_1 and H_2. □

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}.

We note 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 Figure 2.

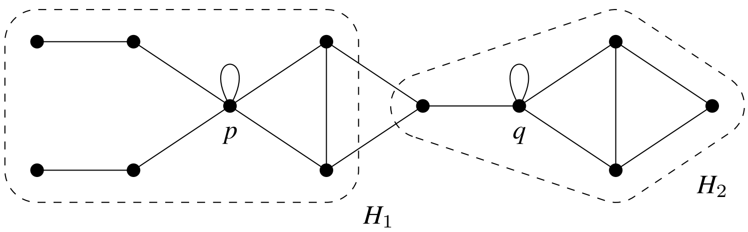


Figure 2. An example of the construction of graphs H_1 and H_2 from a connected 2-reflexive target graph H with $\omega = 3$.

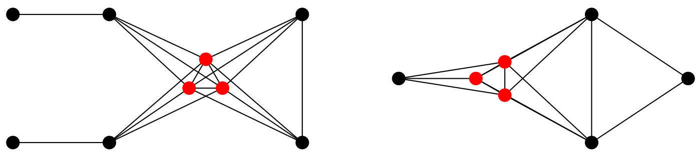


Figure 3. The graphs F_1 (left) and F_2 (right) resulting from the graph H in Figure 2.

Let $\ell = \mathrm{dist}_H(p,q) \geqslant 2$ 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.

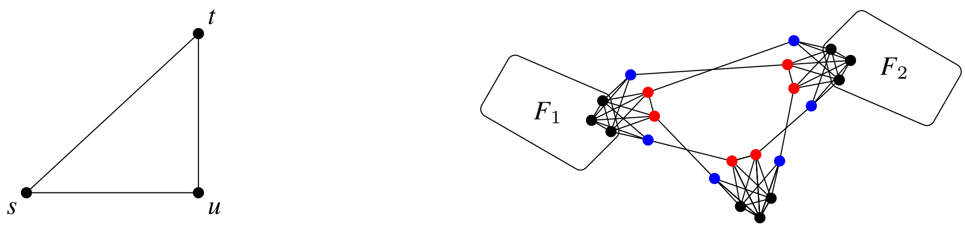
We say that we *identify* two vertices u and v of a graph when we remove them from the graph and replace them with a single vertex that we make adjacent to every vertex that was adjacent to u or v. From F_1, F_2, and G we construct a new graph G' as follows:

(1) For each edge $e = uv \in E_G$, we do as follows. We create four vertices, $g^{\mathrm{r}}_{u,e}$, $g^{\mathrm{b}}_{u,e}$, $g^{\mathrm{r}}_{v,e}$ and $g^{\mathrm{b}}_{v,e}$. We also create two paths P^1_e and P^2_e, each of length $\ell - 2$, between $g^{\mathrm{r}}_{u,e}$ and $g^{\mathrm{b}}_{v,e}$, and between $g^{\mathrm{r}}_{v,e}$ and $g^{\mathrm{b}}_{u,e}$, respectively. If $\ell = 2$ we identify $g^{\mathrm{r}}_{u,e}$ and $g^{\mathrm{b}}_{v,e}$ and $g^{\mathrm{r}}_{v,e}$ and $g^{\mathrm{b}}_{u,e}$ 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 $g^1_u, \ldots, g^{\omega}_u$. We then make every vertex in C_u adjacent to both $g^{\mathrm{r}}_{u,e}$ and $g^{\mathrm{b}}_{u,e}$ for every edge e incident to u; we call $g^{\mathrm{r}}_{u,e}$ and $g^{\mathrm{b}}_{u,e}$ a *red* and *blue* neighbour of C_u, respectively; if $\ell = 2$, then the vertex obtained by identifying two vertices $g^{\mathrm{r}}_{u,e}$ and $g^{\mathrm{b}}_{v,e}$, or $g^{\mathrm{r}}_{v,e}$ and $g^{\mathrm{b}}_{u,e}$ 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 $g^{\mathrm{r}}_{u,e}$ and $g^{\mathrm{r}}_{u,e'}$ adjacent, that is, the set of red neighbours of C_u form a clique, whereas the set of blue neighbours form an independent set.

(3) We add F_1 by identifying t^i_p and g^i_s for $i = 1, \ldots, \omega$, and we add F_2 by identifying t^i_q and g^i_t for $i = 1, \ldots, \omega$. We denote the vertices in F_1 and F_2 in G' by their label t^i_x in F_1 or F_2.

See Figure 4 for an example of a graph G'.



(a) An example of a graph G with a $(1, 2)$-factor cut $(\{s, u\}, \{t\})$.

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

Figure 4. An example of a graph G and the corresponding graph G'.

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

Lemma 3. *If there exists a homomorphism* $h : G' \to H$ *then* $\mathrm{dist}_{G'}(u, v) \geqslant \mathrm{dist}_H(h(u), h(v))$ *for every pair of vertices* $u, v \in V_{G'}$.

We now use the fact that the cliques C_u consist of ω vertices to prove the key property of our construction.

Lemma 4. *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. Since for each $u \in V_G$ and any edge e incident to u, every clique $C_u \cup \{g^{\mathrm{r}}_{u,e}\}$ 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 3, $\mathrm{dist}_{G'}(v, t_p) \geqslant \mathrm{dist}_H(h(v), p)$ and $\mathrm{dist}_{G'}(v, t_q) \geqslant \mathrm{dist}_H(h(v), p)$. In other words:

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

In fact by applying Lemma 3 we can generalize this further to any vertex mapped to p by h:

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

For every $v \in V_{G'}$ we define an upper bound $\mathcal{D}(v)$ on the distance of v to a vertex mapped to p as follows (see also Claim 1):

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

Claim 1. *$\mathcal{D}(v) \geqslant \min_{w \in h^{-1}(p)} \left(\mathrm{dist}_{G'}(v, w)\right) \geqslant \mathrm{dist}_H(h(v), p)$ for all $v \in V_{G'}$.*

We prove Claim 1 by showing that $\mathcal{D}(v) \geqslant \min_{w \in h^{-1}(p)} \left(\mathrm{dist}_{G'}(v, w)\right)$, which suffices due to (1). First suppose $v \in V_{F_1} \cup V_{F_2}$. We may assume, without loss of generality, that $v \in V_{F_2}$. So $\mathcal{D}(v) = \mathrm{dist}_{F_2}(v, t_q) = \mathrm{dist}_{G'}(v, t_q) \geqslant \min_{w \in h^{-1}(p)} \left(\mathrm{dist}_{G'}(v, w)\right)$, as $t_q \in h^{-1}(p)$.

Now suppose $v \notin V_{F_1} \cup V_{F_2}$. Then v either belongs to a clique C_u or is a vertex of a path P_e^1 or P_e^2 between two cliques. If v belongs to a clique or is an end-vertex of such a path, then v is either in $h^{-1}(p)$ or adjacent to a vertex in $h^{-1}(p)$ (since at least one vertex in C_u maps to p). Hence $\mathcal{D}(v) = \lfloor \ell/2 \rfloor \geqslant 1 \geqslant \min_{w \in h^{-1}(p)} \left(\mathrm{dist}_{G'}(v, w)\right)$. Finally, suppose v is an inner vertex of a path P_e^1 or P_e^2. By definition, such a path has length $\ell - 2$. Then v is at most distance $\lfloor (\ell - 2)/2 \rfloor$ from a vertex in a clique, which we know is either in $h^{-1}(p)$ or adjacent to a vertex in $h^{-1}(p)$. Hence $\mathcal{D}(v) = \lfloor \ell/2 \rfloor = \lfloor (\ell - 2)/2 \rfloor + 1 \geqslant \min_{w \in h^{-1}(p)} \left(\mathrm{dist}_{G'}(v, w)\right)$. This proves Claim 1.

We use Claim 1 to prove the following claim, which is crucial for obtaining a contradiction, as we will explain immediately after proving the claim.

Claim 2. *If there exists a surjective homomorphism from G' to H, then for any integer $d \geqslant \ell$:*

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

We prove Claim 2 as follows. Using the fact that with a surjective homomorphism every vertex must be mapped to, we see from Lemma 3 that if there are n vertices in H which are at a distance d from p, there must be at least n vertices in G' that are at distance at least d from every vertex that maps to p. This means we can say for any distance $d \geqslant 0$:

$$\left|\left\{v \in V_{G'} : \min_{w \in h^{-1}(p)} \left(\mathrm{dist}_{G'}(v, w)\right) \geqslant d\right\}\right| \geqslant \left|\left\{w \in V_H : \mathrm{dist}_H(w, p) \geqslant d\right\}\right|.$$

Combining this inequality with Claim 1 yields, for every distance $d \geqslant 0$:

$$\left|\left\{v \in V_{G'} : \mathcal{D}(v) \geqslant d\right\}\right| \geqslant \left|\left\{w \in V_H : \mathrm{dist}_H(w, p) \geqslant d\right\}\right|.$$

Now let $d \geqslant \ell$. Then we only have to consider vertices in $F_1 \cup F_2$. Hence, for every $d \geqslant \ell$:

$$\left|\left\{t_w^i \in V_{F_1} \cup V_{F_2} : \mathcal{D}(t_w^i) \geqslant d\right\}\right| \geqslant |\{w \in V_H : \mathrm{dist}_H(w,p) \geqslant d\}|.$$

By construction, for any t_w^i with $i > 1$ we have that $w \in \{s,t\}$ and thus $\mathcal{D}(t_w^i) \leqslant 1 < \ell \leqslant d$. Therefore, no vertex t_w^i with $i \neq 1$ is involved in the equation above, so we can write:

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

Hence Claim 2 is proven.

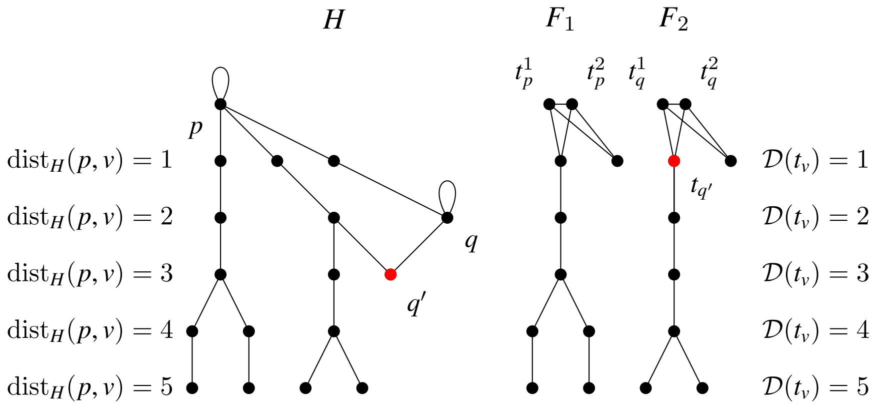


Figure 5. An example of a graph H with corresponding graphs F_1 and F_2. Vertices in H equidistant from p are plotted at the same vertical position and likewise vertices $t_v \in F_1$ and $t_w \in F_2$ with $\mathcal{D}(t_v) = \mathcal{D}(t_w)$ are plotted at the same vertical position. The vertices $q' \in H$ and corresponding $t_{q'} \in F_2$ are highlighted.

We first present the intuition behind the final part of the proof. Consider the graphs F_1, F_2 and H in the example shown in Figure 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, 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 Figure 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 in H (including q') with $\mathrm{dist}_H(p,v) \geqslant 3$ but only nine vertices (excluding q') in $F_1 \cup F_2$ with $\mathcal{D}(t_v) \geqslant 3$ which could be mapped to these vertices. This contradicts Claim 2.

We now formally prove that our initial assumption that $p \in h(C_u)$ for all $u \in V_G$ contradicts Claim 2. For every vertex x in H_1 there is a corresponding vertex t_x^1 such that $\mathcal{D}(t_x^1) = \mathrm{dist}_{F_1}(t_x^1, t_p) = \mathrm{dist}_{H_1}(x,p)$, where the latter equality follows from the construction of F_1. From Lemma 2 we find that $\mathrm{dist}_{H_1}(x,p) = \mathrm{dist}_H(x,p)$ for every $x \in V_{H_1}$. Hence $\mathcal{D}(t_x^1) = \mathrm{dist}_H(x,p)$, and for all $d \geqslant 0$:

$$\left|\left\{t_x^1 \in V_{F_1} : \mathcal{D}(t_x^1) \geqslant d\right\}\right| = |\{x \in V_{H_1} : \mathrm{dist}_H(x,p) \geqslant d\}|. \quad (2)$$

Now let $x \in V_{H_2}$. Using the same arguments, we see that $\mathcal{D}(t_x^1) = \mathrm{dist}_H(x,q)$, and thus $\mathcal{D}(t_x^1) = \mathrm{dist}_H(x,q) \leqslant \mathrm{dist}_H(x,p)$ by definition. Note that, had we instead supposed that it was q to which everything mapped, we would instead have a strict inequality. As it turns out, we only need the weaker inequality.

We now look for a vertex q' in H_2, such that q' is as far from p as possible, subject to the condition that $\mathrm{dist}_H(q', q) < \mathrm{dist}_H(q', p)$. Let $j = \mathrm{dist}_H(q', p)$. We see that for any vertex x in H_2 such that $\mathrm{dist}_H(x, p) > j$, it is the case that $\mathrm{dist}_H(x, q) = \mathrm{dist}_H(x, p)$. Note that there may be no vertices with $\mathrm{dist}_H(x, q) = \mathrm{dist}_H(x, p)$ in which case q' is simply the farthest vertex from p within H_2. We also observe that $q' = q$ is possible. So j is well defined and, in fact, we have that $j \geqslant \ell$.

We now consider the mapping of vertices in H_2 at a distance $d \geqslant \ell$ from p. We recall that $\mathcal{D}(t_x^1) = \mathrm{dist}_H(x, q)$ for every x in H_2 and that for a vertex $x \in H_2$ of distance at least $j + 1$ from q in H, it holds that $\mathrm{dist}_H(x, q) = \mathrm{dist}_H(x, p)$. Combining this with equation (2) yields that:

$$\left|\left\{t_x^1 \in V_{F_1} \cup V_{F_2} : \mathcal{D}(t_x^1) > j\right\}\right| = \left|\{x \in V_H : \mathrm{dist}_H(x, p) > j\}\right|. \tag{3}$$

However, for $d = j$ we find that, in addition to vertices in H_2 equidistant from p and q, there is at least one vertex that is closer to q than p, namely q', for which it holds that $\mathcal{D}(t_{q'}^1) = \mathrm{dist}_H(q', q) < \mathrm{dist}_H(q', p) = j$. It therefore follows that there are fewer vertices t_x^1 with $\mathcal{D}(t_x^1) = j$ than there are vertices x with $\mathrm{dist}_H(x, p) = j$ and hence we see that:

$$\left|\left\{t_x^1 \in V_{F_1} \cup V_{F_2} : \mathcal{D}(t_x^1) = j\right\}\right| < \left|\{x \in V_H : \mathrm{dist}_H(x, p) = j\}\right|. \tag{4}$$

By combining equations (3) and (4), we see that:

$$\left|\left\{t_x^1 \in V_{F_1} \cup V_{F_2} : \mathcal{D}(t_x^1) \geqslant j\right\}\right| < \left|\{x \in V_H : \mathrm{dist}_H(x, p) \geqslant j\}\right|.$$

As $j \geqslant \ell$, this contradicts Claim 2 and concludes the proof of Lemma 4. □

We are now ready to state 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 $\ell \geqslant 2$ 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, sets N_p an N_q, 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 $C'_u = C_u \cup \{g^{\mathrm{r}}_{u,e}\}$ 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 4 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^1_e and P^2_e 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 $i, j \geqslant 0$,* SURJECTIVE $H^{(i,j)}$-COLOURING *is* NP*-complete.*

3. Target Graphs Of At Most Four Vertices

In this section we classify the computational complexity of SURJECTIVE H-COLOURING for every target graph H with at most four vertices. We require a number of lemmas. The first lemma is proved for compaction and not vertex-surjection. However, the only property of compaction used is vertex-surjection and so it is easy to see it holds in this modified form. The second lemma is also displayed in Figure 1.

Lemma 5 ([22])**.** *Let H be a graph with connected components $H_1, \ldots, H_s$. If* SURJECTIVE H_i-COLOURING *is* NP*-complete for some i, then* SURJECTIVE H-COLOURING *is also* NP*-complete.*

Lemma 6 ([1])**.** *For every graph H, if H-*COMPACTION *is polynomial-time solvable, then* SURJECTIVE H-COLOURING *is polynomial-time solvable.*

We also need two results of Golovach, Paulusma and Song. Recall that in our context a tree is a connected graph with no cycles of length at least 3.

Lemma 7 ([12])**.** *Let H be an irreflexive non-bipartite graph. Then* SURJECTIVE H-COLOURING *is* NP*-complete.*

Lemma 8 ([12])**.** *Let H be a tree. Then* SURJECTIVE H-COLOURING *is solvable in polynomial time if H is loop-connected and* NP*-complete otherwise.*

Recall that C^*_4 denotes the reflexive cycle on four vertices (see also Figure 6).

Lemma 9 ([16])**.** *The* SURJECTIVE C^*_4-COLOURING *problem is* NP*-complete.*

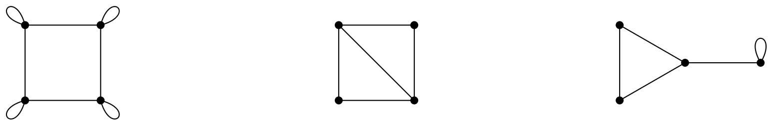
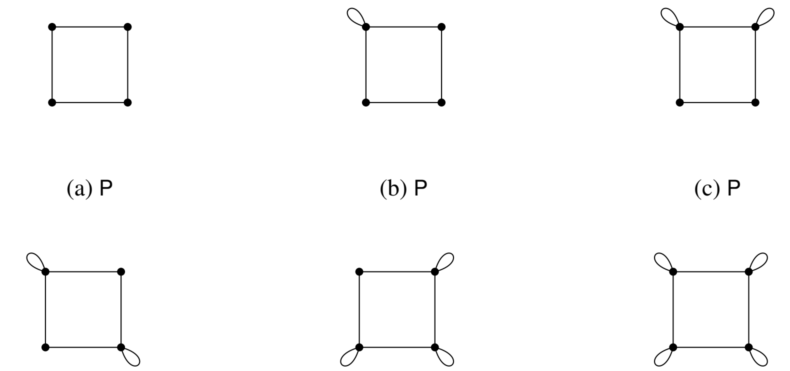


Figure 6. The graphs C_4^*, D and paw*.

We let D denote the irreflexive diamond, that is, the irreflexive complete graph on four vertices minus an edge. The (irreflexive) paw is the graph obtained from the triangle after attaching a pendant vertex to one of the vertices of the triangle, that is, the graph with vertices x_1, x_2, y, z and edges x_1x_2, x_1y, x_2y, yz. We let paw* denote the graph obtained from the paw after adding a loop to its vertex of degree 1 (that is, following the above notation, the loop zz). Both D and paw* are displayed in Figure 6 as well.



(a) P (b) P (c) P

(d) NP-complete (e) P (f) NP-complete

Figure 7. All cycles H on four vertices.

We are now ready to state our main result.

Theorem 4. *Let H be a graph with $|V_H| \leqslant 4$. 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.*

Proof. Let H be a graph on at most four vertices. If H is a loop-connected forest (that is, every component of H is loop-connected) or H has a dominating reflexive vertex, then Vikas [22] showed that H-COMPACTION is in P. Hence, SURJECTIVE H-COLOURING is in P by Lemma 6. If H contains a component that is a non-loop-connected tree, then SURJECTIVE H-COLOURING is NP-complete by Lemmas 5 and 8. If H is an irreflexive non-bipartite graph, then SURJECTIVE H-COLOURING is NP-complete by Lemma 7.

Note that the above cases cover all graphs H on at most three vertices, all disconnected graphs H on four vertices and all trees H on four vertices. The only two graphs H on at most three vertices for which SURJECTIVE H-COLOURING is NP-complete are the irreflexive cycle on three vertices and the 3-vertex path in which the two end-vertices are reflexive. The only disconnected graphs H on four vertices for which SURJECTIVE H-COLOURING is NP-complete are those that contain these two graphs as connected components. The only trees H on four vertices for which SURJECTIVE H-COLOURING is NP-complete are those that are not loop-connected. Hence the theorem

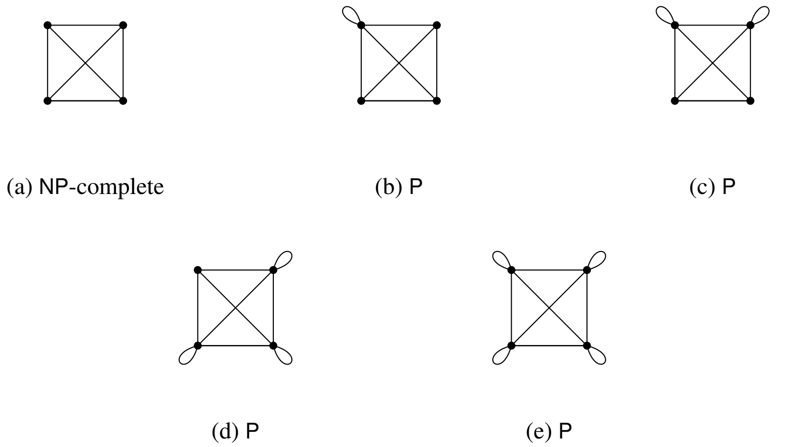


Figure 8. All complete graphs H on four vertices.

holds for every graph H on at most three vertices, for every disconnected graph H on four vertices and for every tree H on four vertices.

From now on we assume that H is a connected graph on four vertices that is not a tree. Then H is either the cycle on four vertices, the complete graph on four vertices, the diamond or the paw. We consider each of these cases separately.

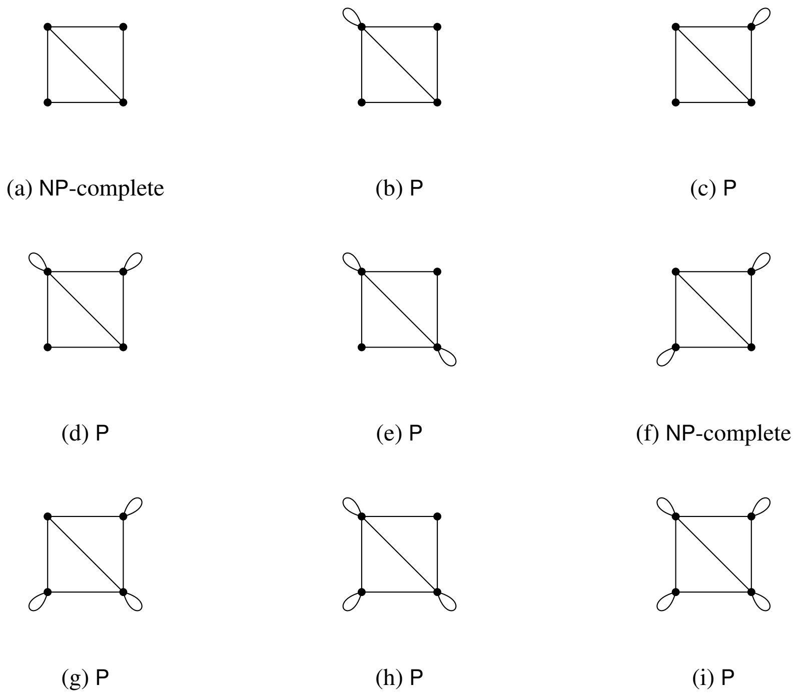
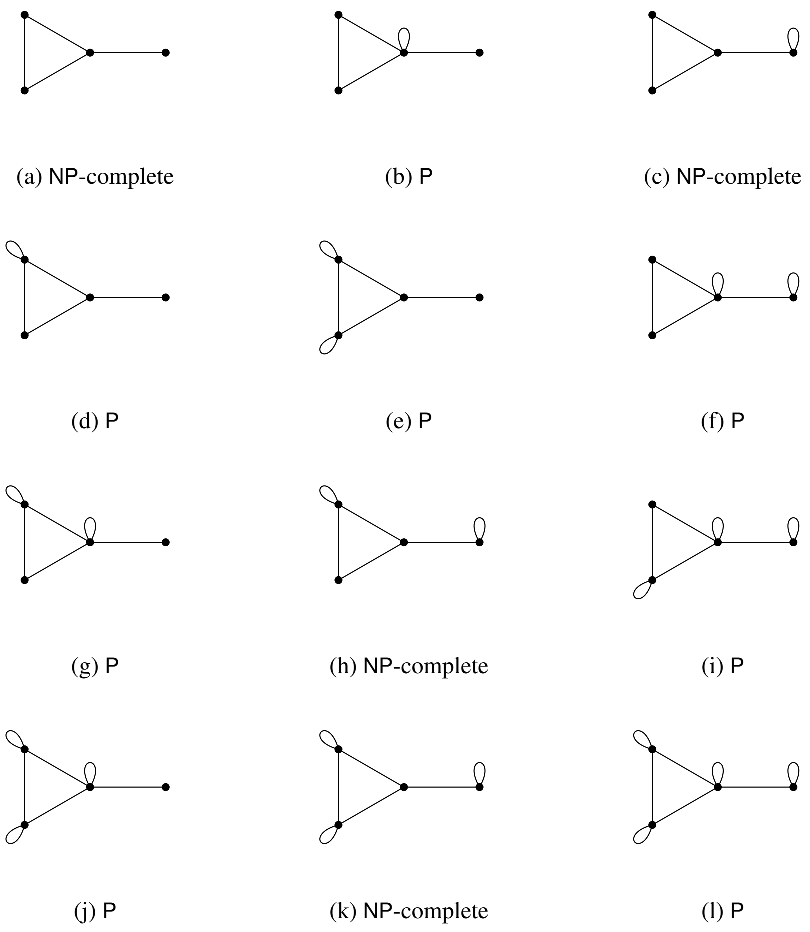


Figure 9. All diamonds H on four vertices.

Suppose H is the cycle on four vertices. There are six cases to consider (see also Figure 7). If H is reflexive, then SURJECTIVE H-COLOURING is NP-complete by Lemma 9. If H is not loop-connected, then H is 2-reflexive, and thus SURJECTIVE H-COLOURING is NP-complete by Theorem 2. In the remaining four cases H is loop-connected. For each of these target graphs, Vikas [22] showed that H-COMPACTION is in P. Hence, SURJECTIVE H-COLOURING is in P by Lemma 6. We find that the theorem holds when H is a cycle on four vertices.

Suppose H is the complete graph on four vertices. There are five cases to consider (see also Figure 8). If H is irreflexive, then SURJECTIVE H-COLOURING is NP-complete by Lemma 7 (as H is non-bipartite as well). For each of the other four target graphs, Vikas [22] showed that H-COMPACTION is in P. Hence, SURJECTIVE H-COLOURING is in P by Lemma 6. We find that the theorem holds when H is the complete graph on four vertices.



(a) NP-complete (b) P (c) NP-complete

(d) P (e) P (f) P

(g) P (h) NP-complete (i) P

(j) P (k) NP-complete (l) P

Figure 10. All paws H on four vertices.

Suppose H is the diamond. There are nine cases to consider (see also Figure 9). If H is irreflexive, then SURJECTIVE H-COLOURING is NP-complete by Lemma 7 (as H is non-bipartite as well). If H is not loop-connected, then H is 2-reflexive, and thus SURJECTIVE H-COLOURING is NP-complete by Theorem 2. For the remaining seven target graphs, Vikas [22] showed that H-COMPACTION is in P. Hence, SURJECTIVE H-COLOURING is in P by Lemma 6. We find that the theorem holds when H is the diamond.

Suppose H is the paw with vertices x_1, x_2, y, z and edges x_1x_2, x_1y, x_2y and yz and possibly one or more loops. There are twelve cases to consider (see also Figure 10). If H is irreflexive, then SURJECTIVE H-COLOURING is

NP-complete by Lemma 7 (as H is non-bipartite as well). If H is not loop-connected, then the set of reflexive vertices is formed by one or two vertices from $\{x_1, x_2\}$ and z. Then SURJECTIVE H-COLOURING is NP-complete by Theorem 3. We are left with nine cases. Vikas [22] showed that H-COMPACTION is in P for all of these cases except for the case where z is the only reflexive vertex. Hence, for eight of these nine cases, SURJECTIVE H-COLOURING is in P by Lemma 6.

We are left to consider the case in which z is the (only) reflexive vertex. Recall that we denote this target by paw*. Theorem 3.5 of [22] proves that paw*-COMPACTION is NP-complete using a reduction from C_3-RETRACTION (which is NP-complete), but we will argue the proof works also for SURJECTIVE paw*-COLOURING. It is shown that (i) a graph G retracts to C_3 if and only if a certain graph G' retracts to paw* if and only if (iii) G' compacts to paw*. The salient part of the proof is Lemma 3.5.2 of [22], in which it is argued that (ii) and (iii) are equivalent. We note that if a graph retracts to another graph, then there exists a surjective homomorphism from the first graph to the second graph. Hence, we need to verify only whether G' retracts to paw* should there exist a surjective homomorphism from G' to paw*. In the proof of Lemma 3.5.2 of [22], the properties of compaction are only used three times. The first two are paragraph 2, line 2 and paragraph 7, line 4 (in the proof of Lemma 3.5.2). The only property used of compaction on these two occasions is vertex surjection. Finally, compaction is alluded to in the final paragraph of the proof, but here any homomorphism would have the desired property. Thus, Vikas [22] has actually proved that G' retracts to paw* if and only if G' has a surjective homomorphism to paw*, and it follows that SURJECTIVE paw*-COLOURING is NP-complete.

From the above we conclude that the theorem holds in all cases when H is the paw. This completes the proof of Theorem 4. □

Theorem 4 corresponds to Vikas' complexity classification of H-COMPACTION for targets graphs H of at most four vertices. Vikas [22] 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.*

4. Conclusions

We proved that SURJECTIVE H-COLOURING 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. This enabled us to classify the computational complexity of SURJECTIVE H-COLOURING for every graph H on at most four vertices. 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. Completely achieving this goal also seems far from trivial. We note that our classification for target graphs H up to four vertices in Section 3 shows such an equivalence for these cases.

References

[1] M. Bodirsky, J. Kára and B. Martin, The complexity of surjective homomorphism problems – a survey, *Discrete Applied Mathematics* **160** (2012), 1680–1690.

[2] A.A. Bulatov, A dichotomy theorem for nonuniform CSPs, *Proc. FOCS 2017*, 319–330.

[3] T. Feder and P. Hell, List homomorphisms to reflexive graphs, *Journal of Combinatorial Theory, Series B* **72** (1998), 236-250.

[4] T. Feder, P. Hell and J. Huang, List homomorphisms and circular arc graphs, *Combinatorica* **19** (1999), 487-505.

[5] T. Feder, P. Hell and J. Huang, Bi-arc graphs and the complexity of list homomorphisms, *Journal of Graph Theory* **42** (2003), 61-80.

[6] T. Feder, P. Hell, P. Jonsson, A. Krokhin and G. Nordh, Retractions to pseudoforests, *SIAM Journal on Discrete Mathematics* **24** (2010), 101-112.
[7] T. Feder and M. Y. Vardi, The computational structure of monotone monadic SNP and constraint satisfaction: a study through datalog and group theory, *SIAM Journal on Computing* **28** (1998), 57–104.
[8] J. Fiala and J. Kratochvíl, Locally constrained graph homomorphisms – structure, complexity, and applications, *Computer Science Review* **2** (2008), 97-111.
[9] J. Fiala, and D. Paulusma, A complete complexity classification of the role assignment problem, *Theoretical Computer Science* **349** (2005), 67-81.
[10] O. Goldreich, On Promise Problems (a survey), *Lecture Notes in Computer Science* **3895** (2006), 254–290.
[11] P. A. Golovach, B. Lidický, B. Martin and D. Paulusma, Finding vertex-surjective graph homomorphisms, *Acta Informatica* **49** (2012), 381-394.
[12] P.A. Golovach, D. Paulusma and J. Song, Computing vertex-surjective homomorphisms to partially reflexive trees, *Theoretical Computer Science* **457** (2012), 86-100.
[13] P. Hell and J. Nešetřil, On the complexity of H-colouring, *Journal of Combinatorial Theory, Series B* **48** (1990), 92–110.
[14] P. Hell and J. Nešetřil, *Graphs and Homomorphisms*, Oxford University Press, 2004.
[15] B. Larose, B. Martin and D. Paulusma, Surjective H-Colouring over reflexive digraphs, Proc. STACS 2018, *Leibniz International Proceedings in Informatics*, to appear.
[16] B. Martin and D. Paulusma, The computational complexity of disconnected cut and $2K_2$-partition. *Journal of Combinatorial Theory. Series B* **111** (2015), 17-37.
[17] M. Patrignani and M. Pizzonia, The complexity of the matching-cut problem, Proc. WG 2001, *Lecture Notes in Computer Science* **2204** (2001), 284–295.
[18] N. Vikas, Algorithms for partition of some class of graphs under compaction and vertex-compaction, *Algorithmica* **67** (2013), 180–206.
[19] N. Vikas, Computational complexity of compaction to reflexive cycles, *SIAM Journal on Computing* **32** (2002), 253–280.
[20] N. Vikas, Computational complexity of graph partition under vertex-compaction to an irreflexive hexagon, Proc. MFCS 2017, *Leibniz International Proceedings in Informatics* **83** (2017), 69:1–69:14.
[21] N. Vikas, Compaction, Retraction, and Constraint Satisfaction, *SIAM Journal on Computing* **33** (2004), 761-782.
[22] N. Vikas, A complete and equal computational complexity classification of compaction and retraction to all graphs with at most four vertices and some general results, *Journal of Computer and System Sciences* **71** (2005), 406-439.
[23] D. Zhuk, A proof of CSP dichotomy conjecture, *FOCS 2017*, 331–342.