Combinatorics, Probability and Computing (2009) 18, 53–81. © 2009 Cambridge University Press
doi:10.1017/S0963548309009663 Printed in the United Kingdom

Deciding Relaxed Two-Colourability: A Hardness Jump

R. BERKE[†] and T. SZABÓ[‡]

Institute of Theoretical Computer Science, ETH Zürich, 8092 Switzerland
(e-mail: {berker,szabo}@inf.ethz.ch)

Received 7 February 2007; revised 2 December 2008

We study relaxations of proper two-colourings, such that the order of the induced monochromatic components in one (or both) of the colour classes is bounded by a constant. A colouring of a graph G is called (C_1, C_2)-*relaxed* if every monochromatic component induced by vertices of the first (second) colour is of order at most C_1 (C_2, resp.). We prove that the decision problem 'Is there a $(1, C)$-relaxed colouring of a given graph G of maximum degree 3?' exhibits a hardness jump in the component order C. In other words, there exists an integer $f(3)$ such that the decision problem is NP-hard for every $2 \leqslant C < f(3)$, while every graph of maximum degree 3 is $(1, f(3))$-relaxed colourable. We also show $f(3) \leqslant 22$ by way of a quasilinear time algorithm, which finds a $(1, 22)$-relaxed colouring of any graph of maximum degree 3. Both the bound on $f(3)$ and the running time greatly improve earlier results. We also study the symmetric version, that is, when $C_1 = C_2$, of the relaxed colouring problem and make the first steps towards establishing a similar hardness jump.

1. Introduction

A function from the vertex set of a graph to a k-element set is called a k-*colouring*. The values of the function are referred to as *colours*. A colouring is called *proper* if the value of the function differs on any pair of adjacent vertices. Proper colouring and the *chromatic number* of graphs (the smallest number of colours which allow a proper colouring) are among the most important concepts of graph theory. Numerous problems of pure mathematics and theoretical computer science require the study of proper colourings and even more real-life problems require the calculation or at least an estimation of the chromatic number. Nevertheless, there is the discouraging fact that the calculation of the chromatic number of a graph or the task of finding an optimal proper colouring are both intractable problems, even fast approximation is probably not possible. This is one of our motivations for studying relaxations of proper colouring, because in some theoretical or

[†] Research partially supported by SNF Grant 200021-108099/1.
[‡] An extended abstract appeared in the *Proceedings of ESA 2006*.

practical situations a small deviation from proper is still acceptable, while the problem could become tractable.

In this paper we study various relaxations of proper colouring, which allow the presence of some small level of conflicts in the colour assignment. Namely, we will allow vertices of one or more colour classes to participate in one conflict or, more generally, let each conflicting connected component have at most C vertices, where C is a fixed integer, not depending on the order of the graph. Most of our results deal with the case of relaxed two-colourings.

To formalize our problem precisely we say that a two-colouring of a graph is (C_1, C_2)-*relaxed* if every monochromatic component induced by the vertices of the first colour is of order at most C_1, while every monochromatic component induced by the vertices of the second colour is of order at most C_2. Note that $(1,1)$-relaxed colouring corresponds to proper two-colouring.

In the present paper we deal with the two most natural cases of relaxed two-colourings. We say *symmetric relaxed colouring* when $C_1 = C_2$ and *asymmetric* relaxed colouring when $C_1 = 1$. Symmetric relaxed colourings were first studied by Alon, Ding, Oporowski and Vertigan [2] and implicitly, even earlier, by Thomassen [18], who resolved the problem for the line graph of 3-regular graphs initiated by Akiyama and Chvátal [1]. Asymmetric relaxed colourings were introduced in [5].

There are several other types of colouring concepts related to our relaxation of proper colouring. In *improper colourings*, introduced independently by Andrews and Jacobson [3], Harary and Jones [10, 11], and Cowen [7], the maximum conflicting degree is bounded. Linial and Saks [17] studied *low-diameter graph decompositions*, where the quality of the colouring is measured by the *diameter* of the monochromatic components. The concept of *fragmentability* of graphs was introduced by Edwards and Farr [9] and studied recently by Haxell, Pikhurko and Thomason [13] for bounded degree graphs; here one aims to break up the graph into small components by removing as small a fraction of the vertices as possible. It seems that the term *relaxed chromatic number* (sometimes also called *generalized chromatic number*) was coined by Weaver and West [20], who use the word 'relaxation' in a much more general sense than we do.

1.1. The problems

We study relaxed colourings from two points of view, extremal graph theory and complexity theory, and find that these coincide for asymmetric relaxed colourings. We also take the first steps towards a similar connection in the symmetric case. To demonstrate our problems, in the next few paragraphs we restrict our attention to asymmetric relaxed colourings; the corresponding questions are asked and partially answered for symmetric relaxed colourings, but there our knowledge is much less satisfactory.

On the one hand, there is the purely graph-theoretic question:

For a given maximum degree Δ, what is the smallest component order $f(\Delta) \in \mathbb{N} \cup \{\infty\}$ such that every graph of maximum degree Δ is $(1, f(\Delta))$-relaxed colourable?

On the other hand, for fixed Δ and C one can study the computational complexity question:

What is the complexity of the decision problem: Given a graph of maximum degree Δ, is there a $(1, C)$-relaxed colouring?

Obviously, for the critical component order $f(\Delta)$ which answers the extremal graph theory question, the answer is *trivial* for the complexity question: every instance is a YES-instance. Note also that for $C = 1$ the complexity question is polynomial time solvable, as it is equivalent to testing whether a graph is bipartite.

In this paper we investigate the complexity question in the range between 2 and the critical component order $f(\Delta)$. We establish the monotonicity of the hardness of the problem in the interval $C \geqslant 2$ and prove a very sharp 'hardness jump'. By this we mean that the problem is NP-hard for every component order $2 \leqslant C < f(\Delta)$, while, of course, the problem becomes trivial (*i.e.*, all instances are YES-instances) for component order $f(\Delta)$. It may be worthwhile to note that at the moment we do not see any *a priori* reason why the hardness of the decision problem should even be monotone in the component order C, *i.e.*, why the hardness of the problem for component order $C + 1$ should imply the hardness for component order C. In fact the problem is obviously polynomial time decidable for $C = 1$, while for $C = 2$ we show NP-completeness.

The other main contribution of the paper concerns the extremal graph theory question and obtains significant improvements over previously known bounds and algorithms. This result becomes particularly important in the light of our NP-hardness results, as the exact determination of the place of the jump from NP-hard to trivial comes within reach.

To formalize our theorems we need further definitions. Let us denote by (Δ, C)-AsymRelCol the decision problem whether a given graph G of maximum degree at most Δ allows a $(1, C)$-relaxed colouring. Analogously, let us denote by (Δ, C)-SymRelCol the decision problem of whether a given graph G of maximum degree at most Δ allows a (C, C)-relaxed colouring. Note here that both $(\Delta, 1)$-AsymRelCol and $(\Delta, 1)$-SymRelCol simply test whether a graph of maximum degree Δ is bipartite.

1.2. The asymmetric problem

For $\Delta = 2$, $(2, 2)$-AsymRelCol is already trivial. For $\Delta = 3$, it was shown in [5] that every cubic graph admits a $(1, 189)$-relaxed colouring, making $(3, 189)$-AsymRelCol trivial. In the proof, the vertex set of the graph was partitioned into a triangle-free and a triangle-full part (every vertex is contained in a triangle), then the parts were coloured separately, and finally the two colourings were assembled amid some technical difficulties. Here we present a completely different approach, which avoids the separation. While we still deal with our share of technical difficulties, we greatly improve on the previous bound on the component order and the running time of the algorithm involved.

A variant of the new method is first presented for 'triangle-full' graphs of maximum degree 3. One facet of our technique is much simpler to present in this scenario and gives an improved and optimal result.

Theorem 1.1. *Let G be a graph of maximum degree at most 3, in which every vertex is contained in a triangle. Then G has a $(1, 6)$-relaxed colouring.*

We prove the theorem in Section 2. An example in [5] shows that the component order 6 is best possible. We note that the existence of a 6-relaxed colouring for *triangle-free* graphs was already proved in [5].

The method is then enhanced to work for all graphs of maximum degree 3 in Section 3. It also implies a quasilinear time algorithm (as opposed to the $\Theta(n^7)$ algorithm implicitly contained in [5]).

Theorem 1.2. *Any graph G with maximum degree at most 3 is $(1,22)$-relaxed colourable, i.e.,*

$$f(3) \leqslant 22.$$

Moreover, there is an $O(n\log^4 n)$ algorithm which finds such a 22-relaxed colouring.

A lower bound of 6 on $f(3)$ was established in [5].

In our next theorem we show that $(3,C)$-AsymRelCol exhibits the promised hardness jump.

Theorem 1.3. *For the integer $f(3)$ we have that:*

(i) $(3,C)$-*AsymRelCol is NP-complete for every* $2 \leqslant C < f(3)$,

(ii) *any graph G of maximum degree at most 3 is* $(1,f(3))$-*relaxed colourable.*

In [5] it was shown that for any $\Delta \geqslant 4$ and positive C, (Δ,C)-AsymRelCol never becomes 'trivial', *i.e.*, for every finite C there is a NO-instance, so $f(4)=\infty$. We show here, however, that the monotonicity of the hardness of $(4,C)$-AsymRelCol still exists for $C \geqslant 2$.

Theorem 1.4. $(4,C)$-*AsymRelCol is NP-complete for every* $2 \leqslant C < f(4) = \infty$.

Obviously, this implies that (Δ,C)-AsymRelCol is NP-complete for every $\Delta > 4$ and $2 \leqslant C < f(\Delta) = \infty$. The proofs of Theorem 1.3 and Theorem 1.4 can be found in Section 4.2.

Remarks. **(1)** Let $f(\Delta,n)$ be the smallest integer f such that every n-vertex graph of maximum degree Δ is $(1,f)$-relaxed colourable. Then $f(\Delta) = \sup f(\Delta,n)$. While $f(3)$ is finite, our graph G_k in Figure 9 provides a simple example for $f(4)$ being non-finite in a strong sense: in any asymmetric relaxed colouring of G_k there is a monochromatic component whose order is linear in the number of vertices. This is in sharp contrast to the examples of [2, 5], where the monochromatic component order is only logarithmic in the number of vertices. It would be interesting to determine the exact asymptotics of the function $f(4,n)$; we only know of the trivial upper bound $f(4,n) \leqslant \frac{3}{4}n$ and the lower bound $f(4,n) \geqslant \frac{2}{3}n$ because of G_k.

(2) A similar hardness jump phenomenon occurs for the k-SAT problem with limited occurrences of each variable. Let k,s be positive integers. A Boolean formula in conjunctive normal form is called a (k,s)-formula if every clause contains *exactly* k distinct variables and every variable occurs in *at most* s clauses. Tovey showed that every $(3,3)$-formula is satisfiable, while the satisfiability problem restricted to $(3,4)$-formulas is NP-complete. Kratochvíl, Savický and Tuza [15] generalized this by establishing the existence of a function $f(k)$, such that every $(k, f(k))$-formula is satisfiable, while the satisfiability problem restricted to $(k, f(k)+1)$-CNF formulas is NP-complete. Observe that the monotonicity of the hardness of the satisfiability problem for (k, s)-formulas is given by definition.

1.3. The symmetric problem

Investigations about relaxed vertex colourings were originally initiated for the symmetric case by Alon, Ding, Oporowski and Vertigan [2]. They showed that any graph of maximum degree 4 has a two-colouring such that each monochromatic component is of order at most 57. This was improved by Haxell, Szabó and Tardos [12], who showed that a two-colouring is possible even with monochromatic component order 6, and such a $(6,6)$-relaxed colouring can be constructed in polynomial time (the algorithm of [2] is not obviously polynomial). In [12] it is also proved that the family of graphs of maximum degree 5 is $(17617, 17617)$-relaxed colourable. Alon, Ding, Oporowski and Vertigan [2] showed that a similar statement cannot be true for the family of graphs of maximum degree 6, as for every constant C there exists a 6-regular graph G_C such that in any two-colouring of $V(G_C)$ there is a monochromatic component of order larger than C.

For the problem (Δ, C)-SymRelCol we make progress in the direction of establishing a sudden jump in hardness. By taking a max-cut one can easily see that $(3, C)$-SymRelCol is already trivial for $C = 2$, so the first interesting maximum degree is $\Delta = 4$. From the result of [12] mentioned earlier it follows that $(4,6)$-SymRelCol is trivial. Cowen, Goddard and Jesurum [8] showed that $(4,2)$-SymRelCol is NP-complete even when the input graphs are restricted to being planar. We extend this result by showing that $(4,3)$-SymRelCol, and also $(6, C)$-SymRelCol, is NP-complete for $C \geqslant 2$. We do not know about the hardness of the problem $(4, C)$-SymRelCol for $C = 4$ and $C = 5$. Again, we do not know any *direct* reason for the monotonicity of the problem. At the moment it is in principle possible that $(4,4)$-SymRelCol is in P while $(4,5)$-SymRelCol is again NP-complete.

Theorem 1.5. *The problems $(4,2)$-SymRelCol, $(4,3)$-SymRelCol and $(6, C)$-SymRelCol, for $C \geqslant 2$, are NP-complete.*

Even though the rough outline of the proof follows that of Theorem 1.3, several sensitive modifications to the definition of the key concepts are necessary. The proof here is omitted: we refer the reader to the first author's PhD thesis [4].

1.4. Notation and terminology

The order of a graph G is defined to be the number of vertices of G. Similarly, the order of a connected component C of G is the number of vertices contained in C. The subgraph of a graph G induced by a vertex set $U \subseteq V(G)$ is denoted by $G[U]$. Vertices and edges in $G[U]$ are referred to as U-vertices and U-edges, respectively. Neighbours of a vertex $v \in V(G)$ in the induced subgraph $G[U]$ are called U-neighbours of v and connected components in an induced subgraph $G[U]$ are called U-components.

To simplify our notation we often say *C-relaxed colouring* instead of $(1, C)$-relaxed colouring. In our investigation of C-relaxed colourings we will encounter two colour classes I and B, where I denotes an independent set and B denotes the colour class that induces components of order at most C. We say that the colour classes B and I are *opposites* of each other. In one of the main auxiliary lemmas, we encounter a third colour class X. We will also use the term *opposite* in relation to X and say that B and X are *opposite*.

For a colour class R (which is a subset of the vertices of G), we often say that *we colour a vertex v with colour R* when in fact *we place v into R*.

2. 6-relaxed colouring of triangle-full graphs

Proof of Theorem 1.1. Let G be a graph of maximum degree 3 such that every vertex of G is contained in a triangle. We will subsequently call such graphs 'triangle-full'.

First we show that without loss of generality we can assume that G is diamond-free (a *diamond* is a graph on four vertices containing two triangles sharing an edge). We proceed by induction on the number of vertices in G. If G contains a diamond D, then by induction we give a 6-relaxed colouring to $G-D$. (Note that after the deletion of a diamond the graph is still triangle-full, since $\Delta(G) \leqslant 3$.) Then we extend this colouring to a 6-relaxed colouring of G. First, for any vertex v whose degree is 2 in D (there are two of these), we colour v with the colour opposite to that received by its unique neighbour in $G-D$ (if it exists). Then we extend this colouring to the whole D by colouring all uncoloured vertices with B. In this way the B-component containing a vertex of D is contained in D, and thus has at most four vertices.

Hence, from now on we can assume that every vertex is contained in exactly one triangle. Let M be the set of edges of G not contained in triangles of G. Obviously, M forms a matching. Further, $G-M$ consists of disjoint triangles covering all vertices of G. The algorithm `PA_TF`(G) (a pseudocode for `PA_TF` can be found in Algorithm 1) constructs a 6-relaxed colouring (I,B) of G by colouring the vertices triangle after triangle. It colours the currently processed vertex v with I if it can, *i.e.*, if v has no neighbour that is already coloured with I. The main point of the algorithm is how to select the next vertex to colour when all vertices in the current triangle are coloured. In particular we make sure that the first vertex we colour from each triangle gets a colour opposite to its partner in M.

Let us first introduce some notation used in Algorithm 1. For a vertex v and an oriented triangle C in $G-M$ containing v, we let v^- denote the predecessor of v in C, let v^+ be its successor in C, and let v^* be its unique neighbour in M (if it exists). We call v^* the *partner* of v.

We immediately see that I forms an independent set. Indeed, only in line 2 do we colour a vertex with I, where no neighbour of it is already coloured I.

Suppose that there is a B-component C larger than 6.

First observe that if a triangle T of G is completely contained in C then, according to line 1 in `PA_TF`(G), the partner of each vertex in T must be contained in I. Thus C consists of only the vertices from T, a contradiction.

Hence we assume that C does not contain any triangle from G completely. Such a component C intersects with at least four triangles T_1, T_2, T_3, T_4 in G. Suppose, without loss of generality, that T_i is incident to T_{i+1}, for $i \in \{1,2,3\}$, and that T_2 gets coloured before T_3 during the execution of `PA_TF`(G). We denote by $v_{i,j}$ the vertex contained in $T_i \cap C$ incident to triangle T_j.

Algorithm 1: PA_TF(G)

Input: Graph G; simple, $\Delta(G) \leqslant 3$, triangle-full and diamond-free.
Output: Vertex partition (I, B); I independent set, no component in $G[B]$ larger than 6.

$I \leftarrow \emptyset, B \leftarrow \emptyset$
give an arbitrary cyclic orientation to each triangle in G
choose arbitrary vertex v in G
while *not all vertices of* G *are coloured* **do**
  while *not all vertices of the triangle containing* v *are coloured* **do**
1   **if** $v^- \in I$ **or** $v^* \in I$ **or** $v^+ \in I$ **then** Add(v, B)
2   **else** Add(v, I)
    $v \leftarrow v^+$
  if *not all vertices of* G *are coloured* **then**
    $v \leftarrow v^-$ `// now` v `is the last vertex we coloured`
3   **if** v^* *is uncoloured* **then** $v \leftarrow v^*$
4   **else if** v^{-*} *is uncoloured* **then** $v \leftarrow v^{-*}$
5   **else** $v \leftarrow w$, where w is arbitrary uncoloured vertex with w^* coloured
return (I, B)

Which vertex of T_2 is coloured first? It can be neither $v_{2,1}$ nor $v_{2,3}$, since the first vertex of any triangle gets a colour opposite to its partner's. (In lines 3, 4, 5 we select the first vertex of the next triangle, such that its partner is coloured. This is true for the first coloured vertex of every triangle except the very first one. Then lines 1, 2 make sure that the first vertex receives a colour different from its partner. This is even true for the very first vertex, since it is coloured I in line 2 and its partner will receive colour B in line 1.)

So either $v_{2,1}$ or $v_{2,3}$ is the last vertex we colour in T_2. After all vertices of T_2 have been coloured, PA_TF(G) chooses either $v_{1,2}$ or $v_{3,2}$ to be coloured next, according to line 3 and line 4 (note that $v_{3,2}$ is not yet coloured according to our assumption). This is a contradiction since, again, the first vertex in any triangle has a colour opposite to its partner. □

Remark. Our proof is constructive and yields a C-relaxed colouring of triangle-full graphs. It is not hard to see that the running time of PA_TF(G) is linear in the number of vertices of G.

3. Trivial $(3, C)$-AsymRelCol; bounding $f(3)$

All graphs we consider in this section have maximum degree at most three.

Proof of Theorem 1.2. We prove the statement of Theorem 1.2 by induction on the number of vertices in G. A *generalized diamond* D is a subgraph of G induced by four vertices of G, such that $d_{V(G)-V(D)}(v) \leqslant 1$ for all $v \in V(D)$ and the vertices of D with degree 1 into $V(G) - V(D)$ form an independent set in G.

The core of the proof is the case when G is generalized diamond-free. Otherwise let D be a generalized diamond in G. By the induction hypothesis, $G - V(D)$ has an I/B-colouring such that the I-vertices form an independent set and the B-vertices induce monochromatic components of order at most 22. We extend this colouring to an I/B-colouring of G. We colour the vertices of D with B unless the vertex has a neighbour in $G - V(D)$, in which case we use the colour *opposite* to the colour of this neighbour. This is always possible since such vertices of D form an independent set in G. Hence all the B-components of $G - V(D)$ remain the same, while the vertices in D will be part of a B-component of order at most four.

It is now left to prove Theorem 1.2 when G is generalized diamond-free. One of the main ingredients of the proof is the following lemma.

Lemma 3.1. *Let G be a generalized diamond-free graph of maximum degree 3 on n vertices. Further, let $v_{\text{fix}} \in V(G)$ and $c \in \{I, B\}$. There exists a vertex partition (I, X, B) of G such that:*

(i) *$I \cup X$ induces a graph where each I-vertex has degree 0 and each X-vertex has degree 1,*
(ii) *no triangle contains two vertices from X,*
(iii) *every B-component is of order at most 6, and*
(iv) *if $d(v_{\text{fix}}) = 2$ then either v_{fix} is contained in c, or $c = I$ and v_{fix} is contained in X.*

Moreover, this vertex partition can be found in time $O(n \log^4 n)$.

First let us see how Lemma 3.1 implies Theorem 1.2. We note that property (iv) is only needed for the inductive proof of Lemma 3.1.

Let I, X and B be such as promised by Lemma 3.1. We do a postprocessing in two phases, during which we distribute the vertices of X between I and B: for each adjacent pair vw of vertices in X we put one of them into B and the other into I. When this happens we say that we have *distributed* the X-edge vw. We specify how we distribute an X-edge vw by the operation Distribute(v, c), where $c \in \{I, B\}$. Distribute(v, c) puts v into c while w is put into the opposite colour class. Note that if property (i) is valid at some point then it is still valid after the distribution of any X-edge. During the first phase some vertices contained in B will be moved to I, but once a vertex is in I, it stays there during the rest of the postprocessing.

For the first phase let us say that a vertex v is *ready for a change* if $v \in B$ and all the neighbours of v are in $B \cup X$. Once we find a vertex v ready for a change we move v to I, and distribute each X-edge which contains a neighbour u of v by Distribute(u, B). We iteratively make this change until we find no more vertices ready for a change, at which point the first phase ends. Note that if a vertex was ready for a change and then moved to I, it will not become ready for a change again, and thus the first phase terminates. Property (ii) ensures that the rules of our change are well defined: it is not possible for an X-neighbour of v to be instructed to be placed in B when it could also be the X-neighbour of another X-neighbour of v which would instruct it to be in I. Property (i) remains valid during the first phase, since besides X-edges being distributed (which preserves property (i)), only such B-vertices are moved to I whose neighbours will all be in B.

Let us now look at how property (iii) changes during the first phase. Crucially, at the end of the first phase every B-component is a path or a cycle, since any B-vertex with three B-neighbours is ready for a change. As a result of one change no two B-components are joined; possibly a vertex u from X which just changed its colour to B is now stuck to an old B-component. If this happens both of the other neighbours of u are in I (and stay there). Let C be a B-component after the first phase. We claim that all vertices adjacent to C are in I except possibly two: one at each endpoint of C (if C forms a path). Indeed, if an interior vertex of C had an X-neighbour, it would have been ready for a change. By (iii) there is a path C' in C containing at most 6 vertices which used to be part of the B-component before the first phase. So we can distinguish three cases in terms of how many X-neighbours C has besides its I-neighbours.

Observation 1. *After the first phase every B-component is one of the following:*

(a) *C is a path containing at most 6 vertices with one X-neighbour at each of its endpoints, or*
(b) *C is a path containing at most 7 vertices with one X-neighbour at one of its endpoints, or*
(c) *C is a path containing at most 8 vertices with no X-neighbours.*

In the second phase we distribute between I and B those vertices which are still in X. The vertices of colour I or B preserve their colour during this phase. Property (i) ensures that the set I we obtain at the end of the second phase is an independent set. We have to be very careful, though, that the connected components in $G[B]$ do not grow too much during the second phase. We guarantee this by finding a *matching transversal* in an auxiliary graph H. The graph H is defined on the vertices of X, $V(H) = X$. There is an edge between two vertices u and v of H if u and v are incident to the same component of $G[B]$.

Claim 1. $\Delta(H) \leqslant 2$.

Proof. Let us pick a vertex y from $V(H) = X$. We aim to show that each edge e incident to y which is not an X-edge (there are at most two of these) is 'responsible' for at most one neighbour of y in H. That is, the component of $G[B]$ adjacent to y via such an edge e is incident to at most one other vertex from X. Indeed, by Observation 1 above, each B-component is a path, possibly adjacent to X-vertices through its endpoints, but not more than to one at each. □

The following lemma guarantees a transversal inducing a matching.

Lemma 3.2 ([12], Corollary 4.3). *Let H be a graph with $\Delta(H) \leqslant 2$ together with a vertex partition $\mathcal{P} = \{P_1, \dots, P_m\}$ into 2-element subsets. Then there is a transversal T ($(T \cap P_i) \neq \emptyset$, for all $i \in \{1, \dots, m\}$) with $\Delta(H[T]) \leqslant 1$.*

We note that the proof of Lemma 3.2 in [12] involves a linear time algorithm which constructs the transversal.

We apply Lemma 3.2 for H with the partition defined by the edges of $G[X]$ (*i.e.*, $\mathcal{P} = E(G[X])$) and find a matching transversal T.

The second phase of our postprocessing consists of moving all vertices of T into B and moving $X \setminus T$ into I.

Since $\Delta(H[T]) \leqslant 1$ we connect at most three connected components Q_1, Q_2 and Q_3 of $G[B]$ by moving an edge $\{u, v\}$ of H into B, with u incident to Q_1 and Q_2 and v incident to Q_2 and Q_3. Obviously, Q_1 and Q_3 are incident to at least one vertex of H (u and v, respectively) and Q_2 is incident to at least two vertices from H (u and v) before moving the vertices of T. According to Observation 1, the largest B-component created in this way is of order at most $7 + 1 + 6 + 1 + 7 = 22$. Lemma 3.1(i) guarantees that I is independent, so the defined colouring is 22-relaxed.

We note that both phases of this proof could be turned into an algorithm whose running time is linear in the number of vertices of G. □

Proof of Lemma 3.1. We use induction on the number of vertices of G. By induction we can of course assume that G is connected. If G is not 2-connected then there is a cut-vertex u in G. Let $G_0 \subseteq G$ be a component of $G - u$ such that $d_{V(G_0)}(u) = 1$, and let u' be the unique neighbour of u in G_0. Define $G_1 = G - G_0$. Then $d_{V(G_1)}(u) \leqslant 2$. Suppose that $v_{\text{fix}} \in V(G_i)$ for $i = 0$ or 1. By induction, we can find a (I_i, X_i, B_i)-partition of G_i such that v_{fix} receives its prescribed colour. Depending on whether $u \in V(G_i)$, either u or u' has a colour assigned to it by the partition (I_i, X_i, B_i); say, u is part of the partition. Then we find a partition $(I_{1-i}, X_{1-i}, B_{1-i})$ of G_{1-i} by induction, such that the vertex u' receives the colour *opposite* to the colour of u. This implies that the partition of G defined by the partition $(I_0 \cup I_1, X_0 \cup X_1, B_0 \cup B_1)$ is as required by Lemma 3.1.

All these steps can be done quickly. Standard techniques involving a depth-first search tree of G enable us to find a cut-vertex of G in linear time in the number of edges plus the number of vertices of G (since we only consider graphs of maximum degree 3, this is certainly also linear in the number of vertices of G).

The essence of the proof of Lemma 3.1 is the case when G is 2-connected. We start proving this case by finding an appropriate matching in G.

Proposition 3.3. *Every n-vertex, 2-edge-connected graph G of maximum degree at most 3 contains a matching M such that:*

(i) $\Delta(G - M) \leqslant 2$,
(ii) $G - M$ *is triangle-free.*

Moreover, M can be found in time $O(n \log^4 n)$.

Proof. Let us first assume that G contains an even number of vertices of degree exactly two. We pair each vertex of degree 2 with another vertex of degree 2 and add one edge between the vertices of each such pair. We denote the new graph by H. Obviously H is a 3-regular, 2-edge connected multigraph.

Secondly, suppose that G contains an odd number of vertices of degree 2. We pick one vertex v with $d(v) = 2$ from G, remove v from G and connect its two neighbours via an edge e_v. The new graph contains an even number of vertices of degree 2. Then we proceed as above to obtain the graph H.

Assume first that H is triangle-free. By Petersen's theorem, H contains a perfect matching M_H. Moreover, if the number of vertices of degree 2 was odd, *i.e.*, if e_v is defined, then M_H can be chosen such that $e_v \notin M_H$. In [6] it is shown that such a matching M_H can be found in time $O(n \log^4 n)$. Let M consist of those edges of M_H which are also edges of G. Then the requirements of Proposition 3.3 are satisfied (if e_v is defined, then the neighbours of v have degree at most 2 in $G - M$, since $e_v \notin M_H$.)

Let us now consider the general case, when H might contain triangles. In order to obtain a perfect matching M such that $H - M$ is triangle-free, we iteratively contract all triangles of H into a vertex, yielding a new triangle-free graph H'. Then we apply the above procedure to H' instead of H and get a perfect matching M' of H'. We observe that this perfect matching M' can easily be extended to a perfect matching M_H of H where each triangle of H contains exactly one edge of M_H. Thus $H - M_H$ is triangle-free. Also, even if e_v is contained in a triangle T, we can ensure $e_v \notin M_H$ by simply forcing the unique edge incident to T, but not to e_v, to not be contained in M'. □

The algorithm that partitions the vertices of G will be denoted by $\mathtt{PA}(G, v_{\text{fix}}, c)$ (see Algorithm 2 for the pseudocode) with v_{fix} being the vertex of G that will be coloured c according to Lemma 3.1(iv).

Let us first discuss informally the main ideas of our algorithm. $\mathtt{PA}(G, v_{\text{fix}}, c)$ chooses a matching M of G as in Proposition 3.3. This is in fact the bottleneck of our algorithm: all other parts are done in linear time. The graph $G - M$ consists of path and cycle components. Algorithm $\mathtt{PA}(G, v_{\text{fix}}, c)$ colours the vertices of G, one component of $G - M$ after another, by traversing each component in a predefined orientation.

$\mathtt{PA}(G, v_{\text{fix}}, c)$ starts the colouring with the vertex v_{fix} and colour c. We will sometimes also refer to this vertex as the *very first vertex*.

For each component the algorithm chooses one of its two orientations. For the component of v_{fix} this is done according to a special rule. The orientation of other components is arbitrary. Recall that v^+ (v^-) denotes the vertex following (preceding) v according to the fixed orientation of its component. To simplify the description of our algorithm we introduce the following conventions. For the source v of a path component, we denote by v^- the sink of the path. Similarly, for the sink u of a path component we denote by u^+ the source of the path. If a vertex v is saturated by M, then the vertex v^* adjacent to v in M is called the *partner* of v.

As a default $\mathtt{PA}(G, v_{\text{fix}}, c)$ tries to colour the vertices of a component of $G - M$ with the colours I and B alternating. Its original goal is to create a proper two-colouring in this way. Of course, there are several reasons which will prevent $\mathtt{PA}(G, v_{\text{fix}}, c)$ from doing so. One main obstacle is when the partner (if it exists) of the currently processed vertex u is already coloured, and it is done so with the same colour we would assign to u. If the conflict is in colour I, then the algorithm resolves this by changing both u and its partner to X. The algorithm generally decides not to care if the conflict is in B. Of course, there is a complication with this rule when the partner is within the same triangle as u, since Lemma 3.1 does not allow two X-vertices in the same triangle. This and other anomalies

Algorithm 2: PA(G, v_{fix}, c)

Input: 2-edge-connected, generalized diamond-free graph G with $\Delta(G) \leqslant 3$; vertex $v_{\text{fix}} \in V(G)$; colour class $c \in \{I, B\}$.
Output: Vertex partition (I, X, B); according to Lemma 3.1(i)–(iv).

$I \leftarrow \emptyset$, $X \leftarrow \emptyset$, $B \leftarrow \emptyset$
choose matching M according to Proposition 3.3
while *not all vertices of G are coloured* **do**
  if $I \cup X \cup B = \emptyset$ **then**
    $v \leftarrow v_{\text{fix}}$
    Orient the component of v such that $\{v^{--}, v^{-}, v\}$ does not form a triangle and $\{v, v^{+}\} \in E(G)$
    if $d(v) = 3$ **then** Add(v, I)
    else Add(v, c) // rule 'very first'
  else
    $v \leftarrow$ FirstVertex(G, v, I, X, B)
    Orient the component of v arbitrarily
    if $v^{*} \in I \cup X$ **then** Add(v, B) // rule 'first'
    else Add(v, I)
  while *not all vertices of the component containing v are coloured* **do**
    $v \leftarrow v^{+}$
    if $v^{-} \in I \cup X$ **and** $\{v^{-}, v\} \in E(G)$ **then**
      Add(v, B) // rule 'standard'
    else // that is, $v^{-} \in B$ or $\{v^{-}, v\} \notin E(G)$
      if v^{+} *is not coloured* **or** $v^{+} \in B$ **or** $\{v, v^{+}\} \notin E(G)$ **then**
        if $v^{*} \in B$ **or** v^{*} *is not coloured* **or** v^{*} *does not exist* **then**
          Add(v, I)
        else // that is, $v^{*} \in I \cup X$
          if $\{v, v^{*}\}$ *in a triangle* **then** Add(v, B) // rule 'triangle'
          else if $v^{*} \in X$ **then** Distribute(v^{*}, B), Add(v, I) // rule 'special'
          else Add$(\{v, v^{*}\}, X)$ // move partners into X
      else // colour the last vertex of a cycle if the first is in $I \cup X$
        if $v^{*} \in I \cup X$ **or** v^{*} *does not exist* **or** $\{v, v^{*}\}$ *in a triangle* **then**
          Add(v, B) // rule 'last'
        else // that is, $v^{*} \in B$ or uncoloured, $\{v, v^{*}\}$ not in a triangle
          if $v^{+} \in X$ **then** Distribute(v^{+}, B), Add(v, I) // rule 'special'
          else Add$(\{v, v^{+}\}, X)$ // move non-partners into X

return (I, X, B)

Algorithm 3: `FirstVertex`(G, v, I, X, B)

Input: G, I, X, and B as defined in algorithm $\mathtt{PA}(G, v_{\text{fix}}, c)$, vertex $v \in V(G)$ coloured last.

Output: First vertex of an uncoloured component C to be coloured.

if v^* *is uncoloured* **then return** v^*
else
  $u \leftarrow v^-$
  while $u \neq v$ **and** ($u \notin B$ **or** u^* *is coloured*) **do**
    $u \leftarrow u^-$
  if $u \neq v$ **then return** u^*
  else return w, where w is arbitrary uncoloured vertex with w^* coloured.

(such as the colouring of the last vertex of a cycle when the first and next-to-last vertex have distinct colours) are handled by a well-designed set of exceptions in place. In fact the design of such a consistent set of exceptions poses a major challenge.

Subsequently a vertex which is coloured first in a component of $G - M$ is referred to as a *first vertex*. Similarly, a *last vertex* is just a vertex coloured last in a component of $G - M$.

After colouring the last vertex v of component C, the algorithm `FirstVertex`(G, v, I, X, B) chooses the partner v^* of v unless v^* is already coloured or v^* does not exist. In that case `FirstVertex`(G, v, I, X, B) looks for a vertex with colour B whose partner is uncoloured, by stepping backwards along the order in which the vertices of C have been coloured, and eventually starts to colour such a partner. If all of the B-coloured vertices of C have an already coloured partner or no partner, then `FirstVertex`(G, v, I, X, B) selects an arbitrary uncoloured vertex with an already coloured partner. The selection of first vertices according to `FirstVertex` coupled with `PA` makes sure that every first vertex has a colour opposite to its partner.

For some subset U of the vertices, the operation $\text{Add}(U, c)$, as used in `PA`, first uncolours those vertices of U which were coloured before and colours all vertices in U with c. $\text{Add}(v, c)$ will be written for $\text{Add}(\{v\}, c)$. If a vertex that has been referenced (for instance v^*) does not exist, then $\text{Add}(v^*, c)$ does not change anything. To simplify the description of the algorithm, by saying, for example, '$v^* \in I$', we mean 'v^* exists and $v^* \in I$'.

Analysis of $\mathtt{PA}(G, v_{\text{fix}}, c)$. In the following we make a couple of observations about first vertices. The proof of (ii) of Observation 2 depends on Corollary 3.4, whose proof only depends on part (i) of Observation 2.

Observation 2. *Let v be a first vertex (but not the very first vertex).*

(i) *The partner of v exists and v^* is coloured before v. In particular, v and v^* are contained in distinct components of $G - M$.*
(ii) *v and v^* receive opposite colours.*

Proof. **(i)** A new first vertex is chosen by `FirstVertex` when each component of $G - M$ has either all or none of its vertices coloured. If there are still uncoloured vertices in G, then there must be one which has a coloured partner (since G is connected) and `FirstVertex` will select such a first vertex. The last claim then follows since a first vertex by definition is coloured first within its component, so its partner cannot be in it.

(ii) When `FirstVertex` selects the next first vertex v, then we know that v^* exists and is coloured. Then line 4 or 5 of `PA` will colour v to the opposite colour, either I or B. If this colour changes later during the execution of `PA` then, according to parts (i) and (ii) of Corollary 3.4, this change must be from I to X, which does not affect the validity of (ii). By part (iii) of Corollary 3.4, an X-vertex can change its colour to B only if it is the very first vertex v_{fix}. □

Observation 3. *If algorithm `PA` recolours a previously coloured vertex, then one of the following three cases holds.*

(i) *A colour I is changed to X either in line 10 or 14. In line 10 we move partners to X, in line 14 we move the last and first vertex of a component into X.*
(ii) *In line 9 the previously uncoloured vertex v^*_{fix} receives colour I. Vertex v_{fix} changes its colour from X to B and v^-_{fix} changes its colour from X to I.*
(iii) *In line 13 the previously uncoloured vertex v^-_{fix} receives colour I. Vertex v_{fix} changes its colour from X to B and v^*_{fix} changes its colour from X to I.*

Proof. It is easy to check that `PA` always assigns colours to the currently processed vertex v, except in those lines stated in the observation.

Note that there are only two lines, line 10 and 14, when vertices are placed into X. Part (i) is then immediate.

Let v be the currently processed vertex which is eventually coloured I in line 9. Its partner, v^*, was coloured with X at a point when v was not yet coloured. Hence v^* was not coloured with X in line 10, where partners together are coloured with X, but it had to be coloured in line 14 where the first and last vertex of a component is coloured with X. Thus v^* is either a first or a last vertex. If v^* were a last vertex, then, since its partner, v, is then uncoloured, `FirstVertex` would select v as the next first vertex and `PA` would colour v in line 4 and not in line 9. So v^* must be a first vertex. Unless v^* is the very first vertex, according to Observation 2(i), its partner, v, should have been coloured already, which it is not, a contradiction. Hence v^* is the very first vertex and part (ii) follows.

For part (iii), suppose that v is the currently processed vertex which is eventually coloured I in line 13. We know that v^+ is a first vertex, which has colour X right before v is processed. v^+ had to receive its colour X in line 10 together with its partner. This is a contradiction unless v^+ is the very first vertex, since, according to `FirstVertex` and lines 4 or 5, a first vertex gets coloured right after its partner with the *opposite colour*. Hence v^+ is the very first vertex and part (iii) follows. □

Let us collect some direct implications of Observation 3.

Corollary 3.4.

(i) *A B-vertex is never recoloured.*
(ii) *An I-vertex can only change its colour to X. If so it had an uncoloured neighbour.*
(iii) *An X-vertex can be recoloured with B only if it is the very first vertex v_{fix} and $d(v_{\text{fix}}) = 3$.*
(iv) *An X-vertex can be recoloured with I only if its X-neighbour is v_{fix} and $d(v_{\text{fix}}) = 3$.*

After these preparations we are ready to start the actual proof of Lemma 3.1.

Property (i). The first property of Lemma 3.1 is certainly true at the initialization of `PA`; we must check that the algorithm maintains it. A vertex v can be added to I in lines 3, 5, 7, 9, or 13. In each of these cases it is easy to check that all the neighbours of v are in B or uncoloured. For lines 9 and 13 note that first we distribute an X-edge between B and I such that the neighbour of v in this X-edge gets colour B. (That is, we call Distribute(v^*, B) for the X-edge $\{v^*, v^{*-}\}$ in line 9 and Distribute(v^+, B) for the X-edge $\{v^+, v^{+*}\}$ in line 13.) Distributing an X-edge does not create any conflict with property (i), provided the property was true up to that point. Then we put v into I knowing that *all* its neighbours are in B or uncoloured.

Vertices are put into X in lines 10 and 14: always an uncoloured vertex v, together with one of its neighbours z. It is easy to check that in both of these lines all neighbours of v except z are in B or uncoloured. To maintain property (i) it is enough to verify that before processing v, z was in I. In line 10 we know that z is the partner of v and is coloured I or X; in fact line 9 excludes that $z \in X$. In line 14 we know that z is equal to v^+ and is coloured I or X, and line 13 excludes that $z \in X$.

In conclusion, property (i) is valid throughout the algorithm.

Property (ii). Why is property (ii) valid? The 'triangle rule' on line 8 ensures that the vertices we move to X in line 10 are not part of the same triangle. In line 14 we move the last and first vertices v and v^+, respectively, of a component of $G - M$ into X. We must check that neither $\{v, v^+, v^{++}\}$ nor $\{v^-, v, v^+\}$ induces a triangle in G. If $\{v, v^+, v^{++}\}$ was a triangle then, since no component of $G - M$ is a triangle, v^{++} has to be the partner of v. Then line 11 ensures that $v^* = v^{++}$ and v are not in the same triangle. Suppose now that $\{v^-, v, v^+\}$ induces a triangle. Again, since no component of $G - M$ is a triangle, v^+ has to be the partner of v^-. Unless v^+ is the very first vertex, v^- cannot be the partner of v^+, since, according to Observation 2(i), v^+ and its partner has to be in a different component of $G - M$. Finally, if v^+ is the very first vertex, then according to the orientation of v^+'s component (see line 1) $\{v^-, v, v^+\}$ does not form a triangle. Hence property (ii) is valid.

Property (iii). To derive the bound on the order of the B-components we list the six reasons a vertex u is coloured B. In the following we emphasize certain properties, which follow immediately from `PA` and Corollary 3.4. We will implicitly refer to these properties throughout the remainder of this section.

- 'very first'-B: given in line 3; u is the very first vertex v_{fix}, $u^+ \in I \cup X$.
- 'first'-B: given in line 4; u is the first vertex coloured in its cycle, $u^+, u^* \in I \cup X$.
- 'triangle'-B: given in line 8; u and u^* are in the same triangle and u^* is already coloured with an I (by the end u^* might change its colour to X).

- 'last'-B: given in line 12; u is the last vertex coloured in its cycle, whose colouring started with I or X, $u^+ \in I \cup X$.
- 'special'-B: given in lines 9 and 13; u is the very first vertex v_{fix}. $u^-, u^* \in I$, $u^+ \in B$, $u^{++} \in I \cup X$.
- 'standard'-B: given in line 6; $u^- \in I \cup X$ unless u^- is a 'special'-B and $u^+ \in I \cup X$.

Every B-coloured vertex has a exactly one of these six reasons why it is coloured B. Note that a B-coloured last vertex is not necessarily a 'last'-B, it could be a 'standard'- or 'triangle'-B. Also, a B-coloured very first vertex is not necessarily a 'very first'-B, but can also be a 'special'-B.

We call a B-component of a component C of $G - M$ a *segment*. Let $\tilde{C}$ be the component C together with the edges of G of the form $\{v, v^{++}\}$ for $v \in V(G)$ (we call such edges *extended edges*). Note that every triangle contains an extended edge. We call a B-component of $\tilde{C}$ an *extended segment*.

Proposition 3.5. *Any extended segment contains at most* 4 *vertices.*

Proof. First let us show the following facts.

Claim 2.

(i) *Suppose u^-, u, and u^+ are all coloured B for some $u \in V(G)$. Then u is a 'triangle'-B. In particular, its partner is in $I \cup X$.*

(ii) *Let v_1, v_2, v_3, v_4, v_5 be five distinct, consecutive vertices along some component C in $G - M$ which are coloured $B, B, I/X, B, B$, in this order. Then v_2 cannot be adjacent to v_4.*

Proof. **(i)** For a vertex v which is a 'standard'-B, 'first'-B, 'very first'-B, 'last'-B, or 'special'-B, either v^- or v^+ is in $I \cup X$.

(ii) Let us suppose that v_2 is adjacent to v_4 and the orientation of the cycle passes through these vertices from left to right (possibly starting/ending among them).

The vertex v_2 is not a 'triangle'-B since $v_2^* = v_4$ is not in $I \cup X$. If v_2 is a 'standard'-B, then v_1 has to be a 'special'-B, since $v_1 \notin I \cup X$. In any case, the first vertex coloured in C is either v_1, v_2 or v_3. This implies that v_5 is neither a 'first'-B nor a 'very first'-B nor a 'special'-B. If v_5 were a 'last'-B, then $v_5^+ \in I \cup X$. Also, v_5^+ is the first vertex of C, so $v_5^+ = v_1$, which has colour B, a contradiction. If v_5 were a 'standard'-B, then v_4 should be in $I \cup X$ or should be a 'special'-B, neither of which is the case. Hence v_5 is a 'triangle'-B. Its partner cannot be v_3, since then $\{v_2, v_3, v_4, v_5\}$ would induce a generalized diamond. So its partner is v_7 (the other vertex distance two away from v_5 along C) which then must have already been coloured when we arrive at v_5. Hence the first vertex coloured in C had to be either v_6 or v_7. Since v_7, as the partner of a 'triangle'-B, is in $I \cup X$, $v_7 \neq v_1, v_2, v_4$. Also, $v_7 \neq v_3$ since our assumption about the v_is being distinct. This contradicts the fact that the first vertex of C is among v_1, v_2, and v_3. □

Part (i) immediately implies that a segment of length 5 does not exist.

Let S be an extended segment and classify the cases according to a longest segment S' it contains.

If S' is of order one then obviously S is of order at most two.

If S' is of order two, then by part (ii) of Claim 2 S cannot contain more segments of order two, only possibly two more segment of order one. Hence its order is at most $1+2+1=4$.

If S' is of order three, then again by part (ii) it cannot be joined to a segment of order at least two. Moreover, it cannot be joined to segments of order one both ways, because, by part (i), at least one way it is closed by a triangle (no generalized diamonds!).

If S' is of order four, then by part (i) both endpoints participate in a triangle and they cannot extend the segment further, because G contains no generalized diamonds. □

A vertex v of an extended segment S is called a *potential connector* if its partner v^* exists, $\{v, v^*\}$ is not an extended edge, and v^* either has colour B or is uncoloured at the time when the colouring of the component of $G - M$ containing S is concluded. Observe that two extended segments can be connected only via their respective potential connectors.

Proposition 3.6.

(i) *If v is a potential connector of extended segment S which does not contain a 'special'-B, then $v^- \notin S$.*

Every extended segment contains at most one potential connector. In particular, every extended segment is adjacent to at most one other extended segment in G.

(ii) *No extended segment of order at least three is adjacent to another extended segment of order at least three.*

Proof. Let v be a potential connector of extended segment S, $|S| \geqslant 2$. We claim that v is a 'standard'-B.

If v was a 'first'-B, 'triangle'-B, or 'special'-B, then v^* is in $I \cup X$ immediately after we coloured v with B, so v is not a potential connector.

If v was a 'last'-B, then it is coloured in line 12. Since v^* exists and $\{v, v^*\}$ is not part of a triangle, we have that $v^* \in I \cup X$ at the time of the colouring. Hence v is not a potential connector.

If $v = v_{\text{fix}}$ was a 'very first'-B, then $v^+ \in I \cup X$. Since $\{v, v^+\} \in E(G)$ (see the orientation rule in line 1), v^*_{fix} exists, and $d(v_{\text{fix}}) = 2$ (see line 2), we have that $\{v^-, v\}$ is not an edge of G. Since $\{v, v^*\}$ is not an extended edge, S consists only of a single vertex.

Let us now show part (i) of Proposition 3.6. Let S be an extended segment not containing a 'special'-B with a potential connector v. Since v is a 'standard'-B and v^- is not a 'special'-B, $v^- \in I \cup X$ and, in particular, is not in S.

Suppose now that an arbitrary extended segment S contains two potential connectors u and w. In particular, $u^*, w^* \notin S$. Then either u^- or w^- has to be in S (otherwise u and w could not be in the same extended segment). Assume that, say, $u^- \in B$. In accordance with the above, u is a 'standard'-B. Hence u^- must be a 'special'-B and $u^+ \in I \cup X$. Moreover, u^{-*} and u^{--} are both contained in $I \cup X$. Thus $S = \{u, u^-\}$ and u^- is not a potential connector, a contradiction.

Let us now proceed with the proof of part (ii). Suppose there are two distinct extended segments S and S', each of order at least 3, contained in the same B-component C of G. If S contained a 'special'-B vertex v (which is the very first vertex) then v^+ is the only neighbour of v which is in B. Also, since $v^{++} \in I \cup X$ and $|S| \geqslant 3$, the partner of v^+ has to be v^{+++} and have colour B. It is easy to see that $v^{++++} \in I \cup X$, so C is equal to $S = \{v, v^+, v^{+++}\}$.

Hence we can assume that neither S nor S' contains a 'special'-B vertex. Suppose further that `PA` colours S prior to S'. According to (i), C does not contain any other vertex besides the vertices of S and S'. Let us denote the potential connectors of S and S' by w and w', respectively. Hence $w^* = w'$, $w'^* = w$ and $\{w, w'\} \in E(G)$.

We will derive a contradiction by showing that $w' \in I \cup X$.

Claim 3. *Let S be an extended segment of order at least three, which does not contain a 'special'-B vertex. Then S contains a last vertex v_l.*

We postpone the proof of Claim 3, and continue with the proof of (ii).

After having coloured the last vertex $v_l \in S$ of a component of $G - M$ containing the extended segment S, `FirstVertex`(G, v_l, I, X, B) searches for a vertex u with an uncoloured partner to continue the colouring with u^*. The potential connector w has an uncoloured partner, w', and we claim that `FirstVertex`(G, v_l, I, X, B) will reach w and will output $w^* = w'$ as the new first vertex. If v_l^* is uncoloured then v_l is the unique potential connector of S, $v_l = w$. Otherwise `FirstVertex`(G, v_l, I, X, B) starts stepping backwards on C looking for a vertex of colour B with an uncoloured partner (see line 1 of `FirstVertex`). We claim that the first such vertex is w. By Proposition 3.6(i) we have that $w^- \notin S$, and $\{w, w^{--}\} \notin E(G)$, since w is a potential connector, so $w^{--} \notin S$. Hence there is a $v_l w$-path $v_l = p_1 \cdots p_m = w$ in S such that $p_{i+1} = p_i^-$ or p_i^{--} for every $i = 1, \ldots, m-1$. `FirstVertex`(G, v_l, I, X, B) will consider all vertices of C in a backward direction from v_l to w. Vertices p_i with $i < m$ are not eligible, since they have a coloured partner. Other vertices between v_l and w are outside S and are thus contained in $I \cup X$. Eventually `FirstVertex`(G, v_l, I, X, B) reaches vertex w. According to our assumption $w' \in S'$ has not yet been coloured, thus `FirstVertex`(G, v_l, I, X, B) chooses w' to be coloured next. Then w' is coloured I according to line 5 of `PA`, a contradiction.

We have thus concluded the proof of Proposition 3.6. □

Proof of Claim 3. Suppose S with $|S| \geqslant 3$ contains neither a 'special'-B nor v_l. Then S certainly does not contain a 'last'-B vertex.

If S contained a 'very first'-B vertex v, then $v^- = v_l \notin S$ and $v^+ \in I \cup X$. Since $|S| \geqslant 3$, $v^* \in S$ and at least one of v^{*+} and v^{*-} is in S. First assume that $v^* = v^{++}$. It is easy to check that $v^{*+} \in I \cup X$, which is a contradiction since $v^{*-} = v^+ \in I \cup X$. Now assume that $v^* = v^{--}$. Obviously, v^{--} is not a 'very first'-B, nor a 'first'-B, nor a 'special'-B, nor a 'last'-B. Also, v^{--} is not a 'triangle'-B since its partner, v, is not in $I \cup X$. Therefore v^{--} has to be a 'standard'-B. Then v^{---} is in $I \cup X$ since it is certainly not a 'special'-B (it is not the very first vertex). This is then a contradiction to $|S| \geqslant 3$, since by our assumption $v^{--+} = v^- \in I \cup X$. We can thus conclude that S does not contain a 'very first'-B.

S does not contain a 'first'-B vertex v either, otherwise $S = \{v\}$. Indeed, $v^- = v_l$ and $v^+ \in I \cup X$ and, according to Observation 2(i), v^* is contained in a different component of $G - M$.

From now on we assume that every vertex of S is either a 'triangle'-B or a 'standard'-B. Suppose S contains a 'triangle'-B vertex u, such that $u^* = u^{++}$. Then $u^{++} \in I \cup X$ and u^+ has to be in B because properties (i) and (ii) of Lemma 3.1 hold. It follows that $u^+ \in S$, but u^+ can neither be a 'standard'-B, since its predecessor is not in $I \cup X$, nor can it be a 'triangle'-B, because $\{u, u^+, u^{++}, u^{+*}\}$ would form a generalized diamond. We conclude that S does not contain a 'triangle'-B vertex u, such that $u^* = u^{++}$. Suppose now that S contains a 'triangle'-B vertex v, such that $v^* = v^{--}$. Then $v^* \in I \cup X$. Vertex v^{-*} is not in S, otherwise $\{v, v^-, v^{--}, v^{-*}\}$ would be a generalized diamond. Since $|S| \geqslant 3$, vertex v^+ has to be in B. It cannot be a 'standard'-B because its predecessor is not in $I \cup X$. Vertex v^+ also cannot be a 'triangle'-B, since we have already seen that its partner cannot be v^{+++}, and if its partner were v^- then $\{v^{--}, v^-, v, v^+\}$ would form a generalized diamond.

Thus the vertices in S are all 'standard'-B vertices, each forming a (not extended) segment of order 1. Each such segment can connect to at most one other such segment via an extended edge. Thus $|S| \leqslant 2$, a contradiction. □

Propositions 3.5 and 3.6 immediately imply part (iii) of Lemma 3.1.

Property (iv). We can assume that $d(v_{\text{fix}}) = 2$. The vertex v_{fix} is contained in c after line 3. If $c = B$, then according to Corollary 3.4(i), v_{fix} is not recoloured at all. If $c = I$, then according to Corollary 3.4(ii) and (iii), v_{fix} can be recoloured with X, but not with B. □

4. Hardness results

4.1. 0/1-colourings

In this subsection we take the first step, which is common to all our hardness proofs. Our plan is to reduce our problems to 3-SAT. Given a 3-SAT formula F, we construct (in polynomial time) a graph G_F together with a constraint function $c = c_F$, such that (G_F, c) has a so-called 0/1-*colouring* if and only if the formula F is satisfiable.

Let G be a graph and $c : V(G) \to N \cup \{\infty\}$ be a constraint function. Then a mapping χ from $V(G)$ to $\{0, 1\}$ is called a 0/1-*colouring* of (G, c) if the vertices with χ-value 1 induce an independent set and the order of each connected component C induced by vertices of χ-value 0 is not larger than the constraint of any of its vertices, that is, $c(v) \geqslant |C|$ for all $v \in C$.

We will assemble G_F from various building blocks, pictured in Figures 1 and 2. In the following, if the constraint of a vertex is not specified then it is taken to be ∞.

The *not-gadget* NG is just a path $v\bar{v}$ of length one, where v has constraint 1.

The *copy gadget* $CG(i)$ consists of a path $P : v_1, v_2, \ldots, v_k$ with $k = 2i - 1$ and every vertex v_{2j-1}, $j \in \{1, \ldots, i-1\}$ is identified with an endpoint of a copy $P^{(j)}$ of a path of length two. The vertex v_1 is called the *root*; other vertices of degree one are called the *leaves* of the gadget. Thus $CG(i)$ contains exactly i leaves. Every vertex of degree two of $CG(i)$ is given constraint 1. For more insight see Figure 1. Let us collect some simple facts about these gadgets.

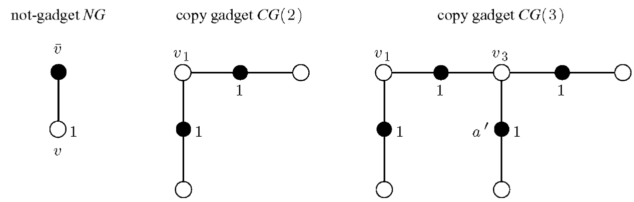

Figure 1. Basic building blocks of the graph G_F.

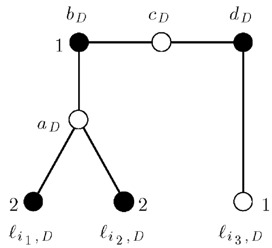

Figure 2. The clause gadget G_D^* for clause $D = (l_{i_1} \vee l_{i_2} \vee l_{i_3})$.

Proposition 4.1.

(i) *The not-gadget NG is $0/1$-colourable. Moreover, in any $0/1$-colouring of the not-gadget, the vertex $\bar{v}$ is coloured with a different colour from that of vertex v.*

(ii) *The copy gadget $CG(i)$ is $0/1$-colourable. Moreover, in any $0/1$-colouring of $CG(i)$, all i leaves have identical colours to the root of the gadget.*

Proof. For each gadget, a $0/1$-colouring is indicated in Figure 1 ('white' corresponds to colour 0 and 'black' corresponds to colour 1). All the statements are easily verified. $\square$

For every clause $D = (l_{i_1} \vee l_{i_2} \vee l_{i_3})$ in F we also construct a gadget. The clause gadget G_D^* as shown in Figure 2 contains vertices a_D, b_D, c_D, d_D and a vertex $l_{i,D}$ corresponding to each literal l_i appearing in the clause D. The constraints of $l_{i_1,D}$ and $l_{i_2,D}$ are 2 and the constraints of $l_{i_3,D}$ and b_D are 1.

Proposition 4.2. *A $0/1$-colouring χ of the vertices $l_{i_1,D}, l_{i_2,D}, l_{i_3,D}$ of the clause gadget G_D^* is extendable to a $0/1$-colouring of G_D^* if and only if at least one of $l_{i_1,D}, l_{i_2,D}, l_{i_3,D}$ received the colour* 1.

Proof. Let us first suppose that $\chi(l_{i_j,D}) = 0$, for all $j \in \{1,2,3\}$ and try to extend χ to a $0/1$-colouring of G_D^*. Then a_D must be coloured 1, since $l_{i_1,D}$ and $l_{i_2,D}$ have constraint 2. Since 1-vertices form an independent set, $\chi(b_D) = 0$. The constraint of b_D implies that $\chi(c_D) = 1$, which then implies that $\chi(d_D) = 0$. Hence $l_{i_3,D}$ is contained in a 0-component of order at least 2, which contradicts the fact that its constraint is 1. We conclude that an extension to a $0/1$-colouring of G_D^* is not possible.

Secondly, we show that an extension exists if some $l_{i_j,D}$ is coloured 1 in χ.

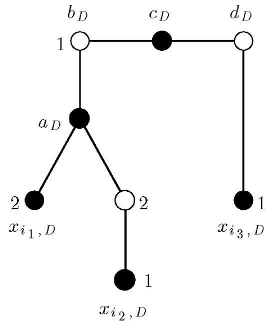

Figure 3. The extended clause gadget G_D for the clause $D = (x_{i_1} \vee \bar{x}_{i_2} \vee x_{i_3})$.

First suppose that $\chi(l_{i_1,D}) = \chi(l_{i_2,D}) = 0$ and $\chi(l_{i_3,D}) = 1$. Then $\chi(a_D) = \chi(c_D) = 1$, $\chi(b_D) = \chi(d_D) = 0$ is a $0/1$-colouring of G_D^*.

Now let $(\chi(l_{i_1,D}), \chi(l_{i_2,D})) \neq (0,0)$. Then $\chi(a_D) = 0$, $\chi(b_D) = 1$, $\chi(c_D) = 0$ and either $\chi(d_D) = 1$ if $\chi(l_{i_3,D}) = 0$ or $\chi(d_D) = 0$ if $\chi(l_{i_3,D}) = 1$ again results in a $0/1$-colouring of G_D^*. □

Now we are ready to define the graph G_F together with its constraint function c_F. First, for each clause D we construct the *extended clause gadget* G_D by taking the clause gadget G_D^* and identify each vertex $l_{i,D}$ corresponding to a negated variable $\bar{x}_i$ in the clause D with the leaf $x_{i,D}$ of a not-gadget. We call this the *extended clause gadget* of the clause D. See Figure 3 for an example.

Proposition 4.3. *An assignment α satisfies the clause D if and only if there is a $0/1$-colouring of the extended clause gadget G_D such that the vertices corresponding to the variables receive the colours the assignment α gives them.*

Proof. It is easy to verify, based on the properties of the not-gadget and the properties of the clause gadget discussed in the previous two propositions. □

The graph G_F is put together from these extended clause gadgets of the clauses of F with the help of one copy gadget for each variable of F. Formally, G_F is constructed as follows. We take the disjoint union of one extended clause gadget for each clause in F. Then we add one copy gadget C_x for each variable x. If the variable x occurs in i_x clauses then the leaves of the copy gadget $C_x \cong CG(i_x)$ are identified with the vertices corresponding to the same variable x in the extended clause gadgets.

Obviously, the graph G_F can be constructed in polynomial time in the number of clauses and variables of F.

The main theorem of the section is now a simple consequence of the above.

Theorem 4.4.

(i) *G_F is $0/1$-colourable if and only if F is satisfiable.*

(ii) *$\Delta(G_F) \leqslant 3$ and every vertex v of G_F with $c(v) < \infty$ has degree at most 2.*

Proof. Let α be a satisfying assignment of F. Then we start defining a 0/1-colouring of G_F by assigning colour $\alpha(x)$ to the root of the copy gadget C_x corresponding to the variable x. This can be extended to an 0/1-colouring of the copy gadgets by part (ii) of Proposition 4.1, where the leaves receive the same colour as their respective roots. All these leaves are identified with a vertex of an extended clause gadget. Since α satisfies all the clauses of F, these partial colourings of the extended clause gadgets can be extended to a 0/1-colouring of the whole gadget (see Proposition 4.3) and thus the whole graph G_F is 0/1-coloured.

Now let χ be a 0/1-colouring of G_F. We claim that the colours given to the roots of the copy gadgets corresponding to the variables give a satisfying assignment of F. By part (ii) of Proposition 4.1 all the leaves are the same colour as their roots in the copy gadget. By Proposition 4.3 every extended copy gadget has a satisfying assignment, so we are done.

Part (ii) is straightforward. □

4.2. Hard $(3, C)$-AsymRelCol

We will use the core graph G_F defined above to construct a graph $RelColGraph(F)$ in polynomial time, which is C-relaxed colourable if and only if the formula F is satisfiable.

For a C-relaxed colouring we denote the colour class forming an independent set by I and the colour class spanning components of order at most C by B.

Definition. Let $C \geqslant 2$ and $\Delta \geqslant 3$ be integers. A graph G is called (Δ, C)-*forcing* with forced vertex $f \in V(G)$ if:

(i) $\Delta(G) \leqslant \Delta$ and f has degree at most $\Delta - 1$,
(ii) G is C-relaxed colourable, and
(iii) f is contained in I for every C-relaxed two-colouring of G.

Lemma 4.5. *For any integers $\Delta \geqslant 3$ and $C \geqslant 2$ the decision problem (Δ, C)-AsymRelCol is NP-complete provided a (Δ, C)-forcing graph exists.*

Proof. We will show that there is a polynomial time algorithm which, given a 3-CNF formula F, produces a graph $RelColGraph(F)$ of maximum degree at most Δ such that F is satisfiable if and only $RelColGraph(F)$ has a C-relaxed colouring.

The *base gadget* BG_l contains l disjoint copies $H_1, \ldots, H_l$ of the (Δ, C)-forcing graph H, the forced vertex f_i of copy H_i is joined to a new vertex t_i for $i \in [l]$, and the vertices $t_1, t_2, \ldots, t_l$ form a path. The vertex t_1 (of degree two) is called the *sink* of the base gadget.

Proposition 4.6. *The base gadget BG_l is C-relaxed colourable for every $l \leqslant C$. Moreover, in any C-relaxed colouring of BG_l, $l \leqslant C$, the sink is contained in a B-component of order l.*

Proof. A C-relaxed colouring of the base gadget is indicated in Figure 4 with 'black' corresponding to colour class B and with 'white' corresponding to colour class I. In any C-relaxed colouring χ of the base gadget BG_l, $\chi(t_i) = B$, since f_i is forced to be contained in I. Thus the vertices t_i for $i \in \{1, \ldots, l\}$ form a B-component of order exactly l. □

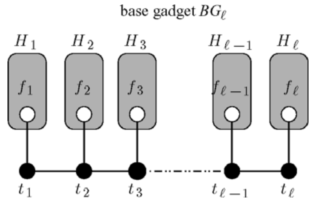

Figure 4. The base gadget BG_l.

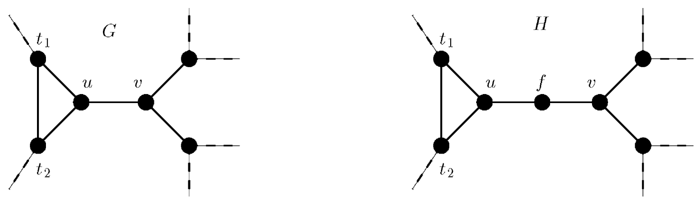

Figure 5. Splitting $e = \{u, v\}$ into $e_1 = \{u, f\}$ and $e_2 = \{f, v\}$.

Now *RelColGraph*(F) is obtained from G_F by connecting each vertex with constraint 1 to the sink of a base gadget BG_{C-1}, and connecting each vertex with constraint 2 to the sink of a base gadget BG_{C-2}. Note that the obtained graph has maximum degree Δ, according to part (ii) of Theorem 4.4. Note also that G_F is 0/1-colourable if and only if *RelColGraph*(F) has a C-relaxed colouring. A C-relaxed colouring of *RelColGraph*(F) restricted to $V(G_F)$ is a 0/1-colouring if we exchange the colour I to 1 and the colour B to 0. Conversely a 0/1-colouring of G_F can be extended to a C-relaxed colouring of *RelColGraph*(F) by identifying 1 with I, and 0 with B, and extending this colouring to the base gadgets appropriately (such a colouring exists by Proposition 4.6). □

4.2.1. $(3, C)$-forcing graphs. Let $\mathcal{G}_C$ denote the family of graphs of maximum degree at most three that are not C-relaxed two-colourable.

Lemma 4.7. *For all $C \geqslant 2$, if $\mathcal{G}_C \neq \emptyset$ then there is a $(3, C)$-forcing graph.*

Proof. Let us assume first that $C \geqslant 6$. By a lemma of [5] we can assume that any member of $\mathcal{G}_C$ contains a triangle.

Lemma 4.8 ([5]). *Any triangle-free graph of maximum degree at most 3 has a 6-relaxed colouring.*

Let us fix a graph $G \in \mathcal{G}_C$ which is minimal with respect to deletion of edges. Let T be a triangle in G (guaranteed by Lemma 4.8) with $V(T) = \{t_1, t_2, u\}$ and let $e = \{u, v\}$ be the unique edge incident to u not contained in T. We split e into e_1, e_2 with $e_1 = \{u, f\}$ and $e_2 = \{f, v\}$, and denote this new graph by H (see Figure 5). We claim that H is a $(3, C)$-forcing graph with forced vertex f. H is C-relaxed colourable since the minimality of G ensures that $G - e$ has a C-relaxed colouring, while the non-C-relaxed-colourability

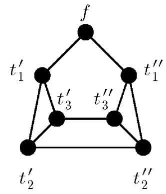

Figure 6. $(3,C)$-forcing graph for $C \in \{2,3\}$.

of G ensures that the colours of u and v are the same on *any* C-relaxed colouring of $G-e$. So *any* C-relaxed colouring χ of $G-e$ can be extended to a C-relaxed colouring of H by colouring f with the opposite of the colour of u and v. Moreover, any such extension is unique. If $\chi(u)=\chi(v)=I$, then obviously $\chi(f)=B$. If $\chi(u)=\chi(v)=B=\chi(f)$ and χ is a C-relaxed colouring of H, then χ restricted to $V(G)$ is a C-relaxed colouring of G, a contradiction.

Thus, in any C-relaxed two-colouring χ_H of H, $(\chi_H(u),\chi_H(f),\chi_H(v))$ is either (I,B,I) or (B,I,B).

We denote by v_1,v_2 the neighbours of t_1 and t_2, respectively, not contained in T (possibly $v_1=v_2$). Suppose the vertices (u,f,v) of H can be coloured with (I,B,I). But then $\chi_H(t_1)=\chi_H(t_2)=B$.

Case (i). If $\chi_H(v_1)=\chi_H(v_2)=I$ then we define a C-relaxed two-colouring χ_G for G as follows: $\chi_G(x)=\chi_H(x)$ for all $x\in V(G)\backslash\{u\}$ and $\chi_G(u)=B$.

Case (ii). Without loss of generality $\chi_H(v_1)=B$. We define a C-relaxed two-colouring χ_G for G as follows: $\chi_G(x)=\chi_H(x)$ for all $x\in V(G)\backslash\{t_1,u\}$, $\chi_G(t_1)=I$, and $\chi_G(u)=B$. Indeed, the B-component containing t_2 did not increase, since $\chi_G(t_1)=\chi_G(v)=I$ and in H $\chi_H(t_1)=B$.

In both cases G would be C-relaxed two-colourable, a contradiction. Thus, in any C-relaxed two-colouring of H the vertices (u,f,v) are coloured (B,I,B). The vertex f is contained in I and is of degree 2, hence H is a $(3,C)$-forcing graph with forced vertex f.

For $2\leqslant C\leqslant 5$ we explicitly construct $(3,C)$-forcing graphs. The graph G in Figure 6 is $(3,C)$-forcing for $C\in\{2,3\}$. First we observe that G is indeed 2-relaxed two-colourable: just take $I=\{f,t_2',t_3''\}$ and $B=V(G)\backslash I$. It is also not hard to check that there is no 3-relaxed two-colouring where vertex f is contained in B. Suppose there is a 3-relaxed two-colouring of G in which f is contained in B. If t_1',t_1'' are contained in I, then no other vertex is contained in I and we have a B-component of order four. On the other hand, if t_1',t_1'' are both contained in B then we have a B-component of order at least five. So without loss of generality t_1' is contained in I and t_1'' is contained in B. The B-components on both triangles are connected, thus we have a B-component of order five again.

Next we construct a graph H which is $(3,C)$-forcing for $C\in\{4,5\}$. First let us show that, for the graph H^* in Figure 7, (i) there is a 4-relaxed two-colouring, and (ii) there is no 5-relaxed colouring where u is contained in I.

(i) The vertex partition defined by $I=\{t_{1,2},t_{2,4},t_{3,1},t_{4,5},t_{5,3}\}$ and $B=V(H^*)\backslash I$ is a 4-relaxed two-colouring of H^*,

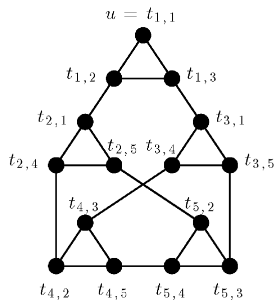

Figure 7. Graph H^*.

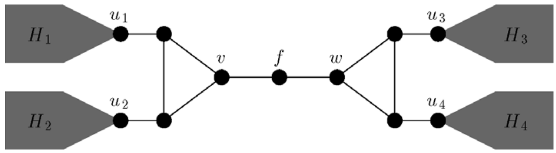

Figure 8. $(3, C)$-forcing graph for $C \in \{4, 5\}$.

Note that in this colouring $u = t_{1,1}$ is contained in a B-component of order two.

(ii) The key observation is that in any 5-relaxed colouring of H^* with $u \in I$, for a triangle T_i with $V(T_i) = \{t_{i,j}, t_{i,k}, t_{i,l}\}$, if $t_{i,j}$ is contained in I then at least one of $t_{k,i}, t_{l,i}$ is contained in I. Suppose not: then the at least six B-vertices of the three triangles T_i, T_k, and T_l are contained in the same B-component.

Thus, if $t_{1,1}$ is contained in I in a 5-relaxed colouring of H^*, then, without loss of generality $t_{3,1}$ is contained in I as well. This then implies that one of $t_{4,3}$ and $t_{5,3}$, say $t_{5,3}$, is in I. Hence $t_{1,2}, t_{5,2} \in B$ and $t_{3,4}, t_{5,4} \in B$. These, together with the key observation, imply that $t_{2,4} \in B$ and $t_{4,2} \in B$, respectively. Finally, all neighbours of triangle T_4 are in B, which together with the key observation imply that all vertices of T_4 are in B, so the B-component of T_4 has order at least six.

The graph H is pictured in Figure 8. The subgraphs H_i, $i \in \{1, \ldots, 4\}$, are copies of the graph H^*, with u_i corresponding to vertex u of H^*.

The colouring of part (i) can easily be extended to a 4-relaxed colouring of H.

As we have seen, in any 5-relaxed colouring of H all $u_i \in B$. Thus, as in the key observation above, v and w are contained in B. Hence, if f were in B, then its B-component would be of order at least seven, a contradiction. Thus, in any 5-relaxed colouring of H the vertex f is contained in I, so H is $(3, C)$-forcing for $C \in \{4, 5\}$. □

Note that $(3, C)$-AsymRelCol is obviously trivial for all C with $\mathcal{G}_C = \emptyset$, so Theorem 1.3 follows immediately from Lemma 4.7 and Lemma 4.5.

4.2.2. $(4, C)$-forcing graphs.

Lemma 4.9. *For all $\Delta \geqslant 4$ and all $C \geqslant 2$, there is a (Δ, C)-forcing graph.*

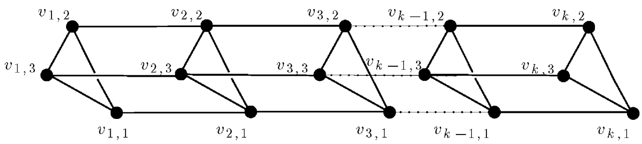

Figure 9. G_k with one B-component of order $2k$.

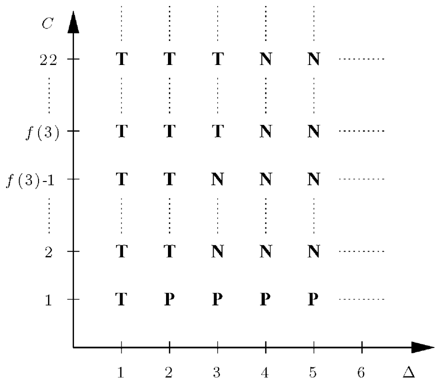

Figure 10. Hardness of (Δ, C)-AsymRelCol.

Proof. Suppose first that $C = 2k - 2$. Let us look at the graph G_k in Figure 9. This graph is not $(2k-1)$-relaxed two-colourable, since in any triangle $v_{i,1}, v_{i,2}, v_{i,3}$ at most one vertex is contained in the independent set I. The two other vertices are contained in B, and since there are three edges connecting this triangle to a neighbouring triangle the components in $G_k[B]$ of all triangles of G_k are connected and form one big component in $G_k[B]$. Removing the edge $e = \{v_{1,1}, v_{1,2}\}$ makes G_k $(2k-2)$-relaxed two-colourable, and in any such colouring χ, $\chi(v_{1,1}) = \chi(v_{1,2}) = I$. Thus $G_k - e$ is $(4, 2k-2)$-forcing, with forced vertex $v_{1,1}$ (or $v_{1,2}$).

Similarly, G_k with an additional vertex v adjacent to $v_{k,1}, v_{k,2}, v_{k,3}$ (denote this graph by H) is not $(2k)$-relaxed two-colourable, hence $H - e$ is $(v, 2k-1)$-forcing again with forcing vertex $v_{1,1}$ or $v_{1,2}$. □

Combining Lemmas 4.9 and 4.5 concludes the proof of Theorem 1.4.

5. Summary and open problems

It would be interesting to determine exactly the critical monochromatic component order $f(3)$ at which the problem $(3, C)$-AsymRelCol becomes trivial. In Figure 10 we summarize the results for the hardness of deciding (Δ, C)-AsymRelCol. We divide the results into three classes, depending on whether (Δ, C)-AsymRelCol is trivial (**T**), polynomial time decidable (**P**) or NP-complete (**N**).

We conjecture that there is a sudden jump in the hardness of the problem $(4, C)$-SymRelCol. Such a result would be particularly interesting, since here the determination

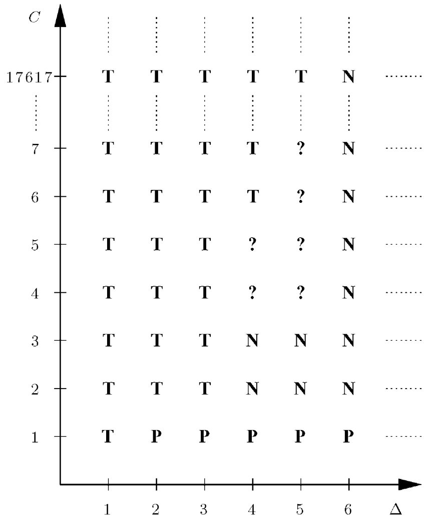

Figure 11. Hardness of (Δ, C)-SymRelCol.

of the critical component order is even more within reach (between 4 and 6). As a first step one could try to prove the monotonicity of the problem.

Conjecture 1. *Prove that there exists an integer* $g(4)$ *such that:*

- *for every* $C, 2 \leqslant C < g(4)$, *it is NP-hard to decide whether a given graph G of maximum degree* 4 *has a* (C, C)*-relaxed colouring, and*
- *every graph of maximum degree* 4 *is* $(g(4), g(4))$*-relaxed colourable.*

A similar problem is wide open for graphs with maximum degree 5: Does $(5, C)$-SymRelCol exhibit monotone behaviour for $C \geqslant 2$? Is there a 'jump in hardness'? Again we use a table to summarize the hardness results for deciding (Δ, C)-SymRelCol: see Figure 11.

For colourings with more than two colours we know much less. Even graph-theoretic questions concerning interesting maximum degrees are open. The following seems a challenging problem.

Open problem 1. *Determine asymptotically the largest* Δ_k *for which there exists a constant* C_k, *such that every graph of maximum degree* Δ_k *can be k-coloured such that every monochromatic component is of order at most* C_k.

The current bounds are $3 < \Delta_k/k \leqslant 4$ (see [12]).

The next two problems discuss the simplest special cases for three colours.

Open problem 2. *Is there a constant C such that every graph with maximum degree* 9 *can be three-coloured such that every monochromatic component is of order at most* C?

The answer is 'yes' for graphs with maximum degree 8 and 'no' for graphs of maximum degree 10 (see [12]).

Open problem 3. *Is there a constant C such that every graph of maximum degree 5 can be red/blue/green-coloured such that the set of red vertices and the set of blue vertices are both independent while every green monochromatic component is of order at most C?*

The answer is 'yes' for graphs with maximum degree 4 and 'no' for graphs of maximum degree 6 (see [5]).

The following problem came up in conversations with Nati Linial and Jirka Matoušek. Let $g(\Delta, n)$ be the smallest integer g such that every n-vertex graph of maximum degree Δ is (g, g)-relaxed colourable. Motivated by the fact that $g(n, 5) = O(1)$ [12] and their result [16] showing that $g(n, 7) = \Omega(n)$, we would be very curious to know the order of $g(n, 6)$. By a theorem of Hochberg, McDiarmid and Saks [14], any two-colouring of the graph T_n (which has maximum degree 6) contains a monochromatic component of order $\Omega(\sqrt{n})$.

References

[1] Akiyama, J. and Chvátal, V. (1981) A short proof of the linear arboricity for cubic graphs. *Bull. Liber. Arts Sci. NMS* **2** 1–3.
[2] Alon, N., Ding, G., Oporowski, B. and Vertigan, D. (2003) Partitioning into graphs with only small components. *J. Combin. Theory Ser. B* **87** 231–243.
[3] Andrews, J. A. and Jacobson, M. S. (1985) On a generalization of chromatic number. In *Proc. Sixteenth Southeastern International Conference on Combinatorics, Graph Theory and Computing* (Boca Raton 1985), Vol. 47, pp. 33–48.
[4] Berke, R. (2008) Transversals and colorings of graphs. PhD thesis, ETH Zürich.
[5] Berke, R. and Szabó, T. (2007) Relaxed coloring of cubic graphs. *J. Combin. Theory Ser. B* **97** 652–668. Extended abstract in *Proc. EuroComb* (2005) 341–344.
[6] Biedl, T. C., Bose, P., Demaine, E. D. and Lubiw, A. (2001) Efficient algorithms for Petersen's Matching Theorem. *J. Algorithms* **38** 110–134.
[7] Cowen, L. J., Cowen, R. H. and Woodall, D. R. (1986) Defective colorings of graphs in surfaces: Partitions into subgraphs of bounded valency. *J. Graph Theory* **10** 187–195.
[8] Cowen, L., Goddard, W. and Jesurum, E. (1997) Defective coloring revisited. *J. Graph Theory* **24** 205–219.
[9] Edwards, K. and Farr, G. (2001) Fragmentability of graphs. *J. Combin. Theory Ser. B* **82** 30–37.
[10] Harary, F. (1985) Conditional colorability in graphs. In *Graphs and Applications* (Boulder 1982), Wiley-Interscience, pp. 127–136.
[11] Harary, F. and Jones, K. F. (1985) Conditional colorability II: Bipartite variations. In *Proc. Sundance Conference on Combinatorics and Related Topics* (Sundance 1985), Vol. 50, pp. 205–218.
[12] Haxell, P., Szabó, T. and Tardos, G. (2003) Bounded size components: Partitions and transversals. *J. Combin. Theory Ser. B* **88** 281–297.
[13] Haxell, P., Pikhurko, O. and Thomason, A. (2008) Maximum acyclic and fragmented sets in regular graphs. *J. Graph Theory*. **57** (2) 149–156.
[14] Hochberg, R., McDiarmid, C. and Saks, M. (1995) On the bandwidth of triangulated triangles. *Discrete Math.* **138** 261–265.
[15] Kratochvíl, J., Savický, P. and Tuza, Z. (1993) One more occurrence of variables makes satisfiability jump from trivial to NP-complete. *SIAM J. Comput.* **22** 203–210.
[16] Linial, N. and Matoušek, J. Private communication.

[17] Linial, N. and Saks, M. (1993) Low diameter graph decompositions. *Combinatorica* **13** 441–454.
[18] Thomassen, C. (1999) Two-colouring the edges of a cubic graph such that each monochromatic component is a path of length at most 5. *J. Combin. Theory Ser. B* **75** 100–109.
[19] Tovey, C. (1984) A simplified NP-complete satisfiability problem. *Discrete Appl. Math.* **8** 85–89.
[20] Weaver, M. L. and West, D. B. (1994) Relaxed chromatic numbers of graphs. *Graphs Combin.* **10** 75–93.