Deciding Relaxed Two-Colorability — a Hardness Jump *

Robert Berke Tibor Szabó †

March 20, 2007

Abstract

We study relaxations of proper two-colorings, such that the order of the induced monochromatic components in one (or both) of the color classes is bounded by a constant. A coloring of a graph G is called (C_1, C_2)-*relaxed* if every monochromatic component induced by vertices of the first (second) color is of order at most C_1 (C_2, resp.). We prove that the decision problem: "Is there a $(1, C)$-relaxed coloring 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 $2 \leq C < f(3)$, while every graph of maximum degree 3 is $(1, f(3))$-relaxed colorable. We also show $f(3) \leq 22$ by way of a quasilinear time algorithm which finds a $(1, 22)$-relaxed coloring of any graph of maximum degree 3. Both the bound on $f(3)$ and the running time improve greatly earlier results. We also study the symmetric version, that is when $C_1 = C_2$, of the relaxed coloring 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-*coloring*. The values of the function are referred to as *color*s. A coloring is called *proper* if the value of the function differs on any pair of adjacent vertices. Proper coloring and the *chromatic number* of graphs (the smallest number of colors which allow a proper coloring) are among the most important concepts of graph theory. Numerous problems of pure mathematics and theoretical computer science require the study of proper colorings 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 coloring are both intractable problems, even fast approximation is probably not possible. This is one of our motivations to study relaxations of proper coloring, because in some theoretical or practical situations a small deviation from proper is still acceptable, while the problem could become tractable. Another reason for the introduction of relaxed colorings is that in certain problems the use of the full strength of proper coloring is an "overkill". Often a weaker concept suffices and provides better overall results.

*An extended abstract appeared in the Proceedings of ESA 2006

†Institute of Theoretical Computer Science, ETH Zürich, 8092 Switzerland.

Email address: {berker, szabo}@inf.ethz.ch

Research is supported in part by the SNF Grant 200021-108099/1.

In this paper we study various relaxations of proper coloring, which allow the presence of some small level of conflicts in the color assignment. Namely, we will allow vertices of one or more color 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-colorings.

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

In the present paper we deal with the two most natural cases of relaxed two-colorings. We say *symmetric relaxed coloring* when $C_1 = C_2$ and *asymmetric* relaxed coloring when $C_1 = 1$. Symmetric relaxed colorings were first studied by Alon, Ding, Oporowski and Vertigan [3] and implicitly, even earlier, by Thomassen [22] who resolved the problem for the line graph of 3-regular graphs initiated by Akiyama and Chvátal [1]. Asymmetric relaxed colorings were introduced in [5].

Related relaxations of proper colorings. There are several other types of coloring concepts related to our relaxation of proper coloring.

Independently Andrews and Jacobson [4], Harary and Jones [10, 11], and Cowen [7] introduced and investigated the concept of *improper colorings* over various families of graphs. A coloring is called (k, l)-improper if none of the at most k colors induces a monochromatic component containing vertices of degree larger than l. Hence in an improper coloring the amount of error is measured in terms of the *maximum degree* of monochromatic components rather than in terms of their order. Several papers on the topic have since appeared; in particular, two papers, by Eaton and Hull [8] and Škrekovski[21], have extended the work of Cowen et al to a list colouring variant of improper colorings.

Linial and Saks [19] studied low diameter graph decompositions, where the quality of the coloring is measured by the *diameter* of the monochromatic components. Their goal was to color graphs with as few colors as possible such that each monochromatic connected component has a small diameter.

Haxell, Pikhurko and Thomason [13] study the *fragmentability* of graphs introduced by Edwards and Farr [9], in particular for bounded degree graphs. A graph is called (α, f)-fragmentable if one can remove α fraction of the vertices and end up with components of order at most f. For comparison, in a $(1, C)$-relaxed coloring one must remove an independent set and end up with small components.

It seems that the term *relaxed chromatic number* (sometimes also called *generalized chromatic number*) was coined by Weaver and West [24]. They used "relaxation" in a much more general sense than us, requiring that each color class is the member of a given family $\mathcal{P}$ of graphs. Naturally, our version also fits into this model.

The problems. We study relaxed colorings from two points of view, extremal graph theory and complexity theory, and find that these points eventually meet for asymmetric relaxed colorings. We also make the first steps for a similar connection in the symmetric case. To demonstrate our problems, in the next few paragraphs we restrict our attention to asymmetric relaxed colorings; the corresponding questions are asked and partially answered for symmetric relaxed colorings, 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 colorable?

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 coloring?

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 $C \geq 2$ and prove a very sharp "hardness jump". By this we mean that the problem is NP-hard for every component order $2 \leq C < f(\Delta)$, while, of course, the problem becomes trivial (i.e. all instances are "YES"-instances) for component order $f(\Delta)$. It is maybe 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 light of our NP-hardness results, as the exact determination of the place of the jump from NP-hard to trivial gets 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 coloring. Analogously, let us denote by (Δ, C)-SymRelCol the decision problem whether a given graph G of maximum degree at most Δ allows a (C, C)-relaxed coloring. Note here that both $(\Delta, 1)$-AsymRelCol and $(\Delta, 1)$-SymRelCol is simply testing whether a graph of maximum degree Δ is bipartite.

The asymmetric problem. For $\Delta = 2$, already $(2, 2)$-AsymRelCol is trivial. For $\Delta = 3$, it was shown in [5] that every cubic graph admits a $(1, 189)$-relaxed coloring, 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 colored separately, finally the two colorings 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. *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 coloring.*

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 coloring 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 2. *Any graph G with maximum degree at most 3 is $(1,22)$-relaxed colorable, i.e.*

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

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

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 3. *For the integer $f(3)$ we have that*
(i) $(3,C)$-AsymRelCol is NP-complete for every $2 \leq C < f(3)$;
(ii) any graph G of maximum degree at most 3 is $(1,f(3))$-relaxed colorable.

In [5] it was shown that for any $\Delta \geq 4$ 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 $C \geq 2$.

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

Obviously, this implies that (Δ,C)-AsymRelCol is NP-complete for every $\Delta > 4$ and $2 \leq C < f(\Delta) = \infty$. The proofs of Theorem 3 and Theorem 4 can be found in Subsection 4.2.

Remark. Let $f(\Delta,n)$ be the smallest integer f such that every n-vertex graph of maximum degree Δ is $(1,f)$-relaxed colorable. Then $f(\Delta) = \sup f(\Delta,n)$. While $f(3)$ is finite, our graph G_k on Figure 9 provides a simple example for $f(4)$ being non-finite in a strong sense: in any asymmetric relaxed coloring of G_k there is a monochromatic component whose order is linear in the number of vertices. This is in sharp contrast with the examples of [3, 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 $f(4,n) \leq \frac{3}{4}n$ and the lower bound $f(4,n) \geq \frac{2}{3}n$ because of G_k.

The symmetric problem. Investigations about relaxed vertex colorings were originally initiated for the symmetric case by Alon, Ding, Oporowski and Vertigan [3]. They showed that any graph of maximum degree 4 has a two-coloring such that each monochromatic component is of order at most 57. This was by improved Haxell, Szabó and Tardos [12], who showed that a two-coloring is possible even with monochromatic component order 6, and such a $(6, 6)$-relaxed coloring can be constructed in polynomial time (the algorithm of [3] is not obviously polynomial). In [12] it is also proved that the family of graphs of maximum degree 5 is $(17617, 17617)$-relaxed colorable.

Alon et al. [3] 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-coloring 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 trivial already 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. Here we show that $(4, C)$-SymRelCol is NP-complete for $C = 2$ and $C = 3$, and that $(6, C)$-SymRelCol is NP-complete for $C > 2$. 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. I.e., at the moment it is in principle possible that $(4, 4)$-SymRelCol is in P while $(4, 5)$-SymRelCol is again NP-complete.

Theorem 5. *The problems* $(4, 2)$*-SymRelCol,* $(4, 3)$*-SymRelCol and* $(6, C)$*-SymRelCol, for* $C \geq 2$ *are* NP*-complete.*

The proof of the theorem appears in Section 4.3.

Related work. Similar hardness jumps of the k-SAT problem with limited occurrences of each variable was shown by Tovey [23] for $k = 3$ and Kratochvíl, Savický and Tuza [17] for arbitrary k. 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 [17] 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. By a standard application of the Local Lemma they obtained $f(k) \geq \left\lfloor \frac{2^k}{ek} \right\rfloor$. After some development [17, 20] the most recent upper estimate on $f(k)$ is only a log-factor away from the lower bound and is due to Hoory and Szeider [15]. Recently new bounds were also obtained on small values of the function $f(k)$ [16]. Observe that the monotonicity of the hardness of the satisfiability problem for (k, s)-formulas is given by definition.

Notation, 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. A graph G is r-regular if all its vertices have degree r. A graph G is called k-edge-connected

(k-vertex-conntected) if there is no edge-cut (vertex-cut, resp.) (a subset of the edges (vertices, resp.) of G that disconnects G) of size at most $k-1$.

The subgraph of a graph G induced by a vertex set $U \subseteq V(G)$ is denoted throughout by $G[U]$. Vertices and edges in $G[U]$ are refered to as U-vertices and U-edges, respectively. Neighbors of a vertex $v \in V(G)$ in the induced subgraph $G[U]$ are called U-neighbors of v and connected components in an induced subgraph $G[U]$ are called U-components.

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

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

2 6-relaxed coloring of triangle-full graphs

Proof of Theorem 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 coloring to $G-D$. (Note that after the deletion of a diamond the graph is still triangle-full, since $\Delta(G) \leq 3$.) Then we extend this coloring to 6-relaxed coloring of G. First for any vertex v whose degree is 2 in D (there are two of these), we color v with the color opposite to what its unique neighbor in $G-D$ (if it exists) has received. Then we extend this coloring to the whole D by coloring all uncolored vertices with a B. 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 coloring (I, B) of G by coloring the vertices triangle after triangle. It colors the currently processed vertex v with I if it can, i.e., if v has no neighbor which is colored with I already. The main point of the algorithm is how to select the next vertex to color when all vertices in the current triangle are colored. In particular we make sure that the first vertex we color from each triangle gets a color opposite to its partner.

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

Algorithm 1: `PA_TF`(G)

Input: Graph G; simple, $\Delta(G) \leq 3$, 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 colored* **do**

    while *not all vertices of the triangle containing v are colored* **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 colored* **then**

        $v \leftarrow v^-$ `// now` v `is the last vertex we colored`

3   **if** *v^* is uncolored* **then** $v \leftarrow v^*$

4   **else if** *v^{-*} is uncolored* **then** $v \leftarrow v^{-*}$

5   **else** $v \leftarrow w$, where w is arbitrary uncolored vertex with w^* colored

return (I, B)

We immediately see that I forms an independent set. Indeed, only in Line 2 color we a vertex with I, where no neighbor of it is colored I already.

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) 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 colored 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.

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

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

Remark. Our proof is constructive and yields a C-relaxed coloring of triangle-full graphs. It is not hard to see that the running time of $\texttt{PA\_TF}(G)$ 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 2. We prove the statement of Theorem 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 $d_{V(G)-V(D)}(v) \leq 1$ 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-coloring such that the I-vertices form an independent set and the B-vertices induce monochromatic components of order at most 22. We extend this coloring to an I/B-coloring of G. We color the vertices of D with B unless the vertex has a neighbor in $G - V(D)$, in which case we use the color *opposite* to the color of this neighbor. 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 2 when G is generalized diamond-free. One of the main ingredients of the proof is the following lemma:

Lemma 1. *Let G be a generalized diamond-free graph of maximum degree 3 on n vertices. Further let $v_{\mathit{fix}} \in V(G)$ 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 $d(v_{\mathit{fix}}) = 2$ 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 1 implies Theorem 2. We note that property (iv) is only needed for the inductive proof of Lemma 1.

Let I, X and B be such as promised by Lemma 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 to B and the other into I. When this happens we say that we *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 color-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 neighbors 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 neighbor u of v by *Distribute*(u, B). We iteratively make this change until we find no more vertex ready for a change, at which point the first phase ends. Property (ii) ensures that the rules of our change are well-defined: It is not possible that an X-neighbor of v is instructed to be placed in B, while it could also be the X-neighbor of another X-neighbor 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 neighbors 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, since any B-vertex with three B-neighbors 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 color to B is now stuck to an old B-component. In case this happens both of the other neighbors 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-one at each endpoint of C. Indeed, if an interior vertex of C had an X-neighbor, 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 a B-component before the first phase. So we can distinguish three cases in terms of how many X-neighbors C has besides its I-neighbors.

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-neighbor at each of its endpoints or

(b) C is a path containing at most 7 vertices with one X-neighbor at one of its endpoints or

(c) C is a path containing at most 8 vertices with no X-neighbors.

In the second phase we distribute between I and B those vertices which are still in X. The vertices of color I or B preserve their color 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 $G[B]$ don't grow too much during the second phase. We guarantee this via 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. $\Delta(H) \leq 2$.

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 neighbor of y in H. That is, the component of $G[B]$ adjacent to y via such 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 2 ([12], Corollary 4.3). *Let H be a graph with $\Delta(H) \leq 2$ together with a vertex partition $\mathcal{P} = \{P_1, \ldots, P_m\}$ into 2-element subsets. Then there is a transversal T ($(T \cap P_i) \neq \emptyset$, for all $i \in \{1, \ldots, m\}$) with $\Delta(H[T]) \leq 1$.*

We note that the proof of Lemma 2 in [12] involves a linear time algorithm which constructs the transversal.

We apply Lemma 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 $\Delta(H[T]) \leq 1$ 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 this way is of order at most $7+1+6+1+7 = 22$. Lemma 1(i) guarantees that I is independent so the defined coloring 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 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 neighbor of u in G_0. Define $G_1 = G - G_0$. Then $d_{V(G_1)}(u) \leq 2$. 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 color. Depending on whether $u \in V(G_i)$, either u or u' has a color 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 color *opposite* to the color of u. This implies that the parition 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 1.

All these steps can be done quickly. Standard techniques involving a depth first search tree of G enable to find a cut-vertex of G in linear time in the number of edges plus 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 1 is the case when G is 2-connected. We start proving this case by finding an appropriate matching in G.

Proposition 1. *Every n-vertex, 2-edge-connected graph G of maximum degree at most 3 contains a matching M such that*

(i) $\Delta(G - M) \leq 2$,

(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 neighbors 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 1 are satisfied (if e_v is defined, then the neighbors 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 force $e_v \notin M_H$ by simply forcing that the unique edge incident to T, but not to e_v, is not 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 colored c according to Lemma 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 1. 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 $\mathtt{PA}(G, v_{\text{fix}}, c)$ colors 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 coloring with the vertex v_{fix} and color 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 color the vertices of a component of $G - M$ with the colors I and B alternatingly. Its original goal is to create a proper two-coloring 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 colored, and it is done so with the same color we would just want to give to u. If the conflict would be in color 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 1 does not allow two X-vertices in the same triangle. This and other anomalies (like the coloring of the last vertex of a cycle when the first and next-to-last vertex have distinct colors) 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 colored first in a component of $G-M$ is referred to as a *first vertex*. Similarly, a *last vertex* is just a vertex colored last in a component of $G-M$.

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

For some subset U of the vertices, the operation $\mathit{Add}(U, c)$, as used in `PA`, first uncolores those vertices of U which were colored before and colors all vertices in U with c. $\mathit{Add}(v, c)$ will be written for $\mathit{Add}(\{v\}, c)$. In case a vertex that has been referenced (for instance v^*) does not exist, then $\mathit{Add}(v^*, c)$ 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 $\texttt{PA}(G, v_{\mathbf{fix}}, c)$ In the following we make a couple of observations about first vertices. The proof of (ii) of Observation 2 does depend on Corollary 1 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 colored before v. In particular, v and v^* are contained in distinct components of $G-M$.*

(ii) *v and v^* receive opposite colors.*

Proof. (i) A new first vertex is chosen by `FirstVertex` when each component of $G-M$ has either all or none of its vertices colored. If there are still uncolored vertices in G, then there must be one which has a colored 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 colored 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 colored. Then Line 4 or 5 of `PA` will color v to the opposite color, either I or B. If this color changes later during the execution of `PA` then, according to part (i) and (ii) of Corollary 1, this change must be from I to X, which does not effect the validity of (ii). By part (iii) of Corollary 1, an X-vertex can change its color to B only if it is the very first vertex v_{fix}. $\square$

Algorithm 2: $\texttt{PA}(G, v_{\text{fix}}, c)$

Input: 2-edge-connected, generalized diamond-free graph G with $\Delta(G) \leq 3$;
vertex $v_{\text{fix}} \in V(G)$; color-class $c \in \{I, B\}$;
Output: Vertex partition (I, X, B); according to Lemma 1(i)-(iv).

$I \leftarrow \emptyset$, $X \leftarrow \emptyset$, $B \leftarrow \emptyset$
choose matching M according to Proposition 1
while *not all vertices of* G *are colored* **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
    $v \leftarrow \texttt{FirstVertex}(G, v, I, X, B)$
    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 colored* **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 colored* **or** $v^{+} \in B$ **or** $\{v, v^{+}\} \notin E(G)$ **then**
        if $v^{*} \in B$ **or** v^{*} *is not colored* **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 // color 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 uncolored, $\{v, v^{*}\}$ 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 `PA`(G, v_{fix}, c), vertex $v \in V(G)$ colored last.
Output: First vertex of an uncolored component C to be colored.

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

Observation 3. *If Algorithm `PA` recolors a previously colored vertex then one of the following three cases hold.*

(i) A color 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 uncolored vertex v^*_{fix} receives color I. Vertex v_{fix} changes its color from X to B and v^-_{fix} changes its color from X to I.

(iii) In Line 13 the previously uncolored vertex v^-_{fix} receives color I. Vertex v_{fix} changes its color from X to B and v^*_{fix} changes its color from X to I.

Proof. It is easy to check that `PA` always assigns colors 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 colored I in Line 9. Its partner, v^* was colored to X at a point when v was not yet colored. Hence v^* was not colored with X in Line 10, where partners together are colored with X, but it had to be colored in Line 14 where the first and last vertex of a component is colored with X. Thus v^* is either a first or a last vertex. If v^* was a last vertex, then, since its partner, v, is uncolored at the time, `FirstVertex` would select v as the next first vertex and `PA` would color 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 colored 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 colored I in Line 13. We know that v^+ is a first vertex, which has color X right before v is processed. v^+ had to receive its color 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 colored right after its partner with the *opposite color*. Hence v^+ is the very first vertex and part (iii) follows. □

Let us collect some direct implications of Observation 3.

Corollary 1. *(i) A B-vertex is never recolored.*

(ii) An I-vertex can only change its color to X. In this case it had an uncolored neighbor.

(iii) An X-vertex can be recolored to B only if it is the very first vertex v_{fix} and $d(v_{fix}) = 3$.

(iv) An X-vertex can be recolored to I only if its X-neighbor is v_{fix} and $d(v_{fix}) = 3$.

After these preparations we are ready to start the actual proof of Lemma 1.

Property (i) The first property of Lemma 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 neighbors of v are in B or uncolored. For Lines 9 and 13 note that first we distribute an X-edge between B and I such that the neighbor of v in this X-edge gets color 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 neighbors are in B or uncolored.

Vertices are put into X in Lines 10 and 14; always an uncolored vertex v, together with one of its neighbors z. It is easy to check that in both of these lines all neighbors of v except z are in B or uncolored. 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 colored 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 colored 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 colored B. In the following we emphasize some property of each, which follow immediately from `PA` and Corollary 1. We will implicitly refer to these properties throughout the remainder of this section.

- "very first"-B: it is given in Line 3; u is the very first vertex v_{fix}, $u^+ \in I \cup X$.

- "first"-B: it is given in Line 4; u is the first vertex colored in its cycle, $u^+, u^* \in I \cup X$
- "triangle"-B: it is given in Line 8; u and u^* are in the same triangle and u^* is already colored with an I (by the end u^* might change its color to X).
- "last"-B: it is given in Line 12; u is the last vertex colored in its cycle, whose coloring started with I or X, $u^+ \in I \cup X$.
- "special"-B: it is 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: it is given in Line 6; $u^- \in I \cup X$ unless u^- is a "special"-B and $u^+ \in I \cup X$.

Every B-colored vertex has a exactly one of these six reasons why it is colored a B. Note that a B-colored last vertex is not necessarily a "last"-B, it could be a "standard"- or "triangle"-B. Also, a B-colored 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 $v \in V(G)$ (such edges we call *extended edges*). Note that every triangle contains an extended edge. We call a B-component of $\tilde{C}$ an *extended segment*.

Proposition 2. *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 colored 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 colored $B, B, I/X, B, B$, 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 is passing through these vertices from left to right (with 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 colored 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 was 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 color B, a contradiction. If v_5 was 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 been colored already when we arrive to v_5. Hence the first vertex colored 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_i's being distinct. This contradicts 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 1 then obviously S is of order at most 2.

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 3, 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 4 then by part (i) both endpoints participate in a triangle and they cannot extend the segment further, because G contains no generalized diamonds. $\square$

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 color B or is uncolored at the time when the coloring 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. *(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, $|S| \geq 2$. We claim that v is a "standard"-B.

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

If v was a "last"-B, then it is colored 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 coloring. 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. 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 neighbor of v which is in B. Also, since $v^{++} \in I \cup X$ and $|S| \geq 3$, the partner of v^+ has to be v^{+++} and have color 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` colors 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 this Claim 3 a little and continue with the proof of (ii).

After having colored 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 uncolored partner to continue the coloring with u^*. The potential connector w has an uncolored partner, w', and we claim that `FirstVertex`(G, v_l, I, X, B) will arrive to w and will output $w^* = w'$ as the new first vertex. If v_l^* is uncolored 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 color B with an uncolored partner (c.f. Line 1 of `FirstVertex`). We claim that the first such vertex is w. By Proposition 3(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 colored partner. Other vertices between v_l and w are outside of S and thus are 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 colored, thus `FirstVertex`(G, v_l, I, X, B) chooses w' to be colored next. Then w' is colored I according to Line 5 of `PA`, a contradiction.

We thus concluded the proof of Proposition 3. □

Proof of Claim 3. Suppose S with $|S| \geq 3$ does neither contain 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 $|S| \geq 3$, $v^* \in S$ and at least one of v^{*+} and v^{*-} is in S. First assume that $v^* = v^{++}$. It is easy to check, that then $v^{*+} \in I \cup X$, which is a contradiction since $v^{*-} = v^+ \in I \cup X$. Now assume that $v^* = v^{--}$. Obviously, v^{--} is not a "very-first"-B, not a "first"-B, not a "special"-B and not 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 $|S| \geq 3$ 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 property (i) and (ii) of Lemma 1 hold. It follows that $u^+ \in S$, but u^+ neither can be a "standard"-B since its predecessor is not in $I \cup X$ nor can 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 $|S| \geq 3$, 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 already saw that its partner cannot be v^{+++} and if its partner was 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 $|S| \leq 2$, a contradiction. $\square$

Proposition 2 and Proposition 3 immediately imply part (iii) of Lemma 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 1(i), v_{fix} is not recolored at all. If $c = I$, then according to Corollary 1(ii) and (iii), v_{fix} can be recolored to X, but not to B.

$\square$

4 Hardness results

4.1 0/1-colorings

In this subsection we take the first step, which is common in 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-*coloring* 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-coloring* 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 $c(v) \geq |C|$ for all $v \in C$.

We will assemble G_F from various building blocks, pictured in Figure 1 and Figure 2. In the following, if the constraint of a vertex is not specified than 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* $CG(1)$ consists of just one vertex v_1, which is called both the *root* and the *leaf* of the gadget. Let P be a path of length two, where the interior vertex is given constraint 1. For $i \geq 2$, a copy-gadget $CG(i)$ is constructed from $CG(i-1)$ by identifying an arbitrary leaf

v_{i-1} of $CG(i-1)$ with one endpoint of each of two copies of P. Note that v_{i-1} is no longer a leaf and we gained two new leaves - the other endpoints of the two copies of P. Thus $CG(i)$ contains exactly i leaves. The root of $CG(i)$ is the vertex v_1 for every i.

For more insight see Figure 1. Let's collect some simple facts about these gadgets.

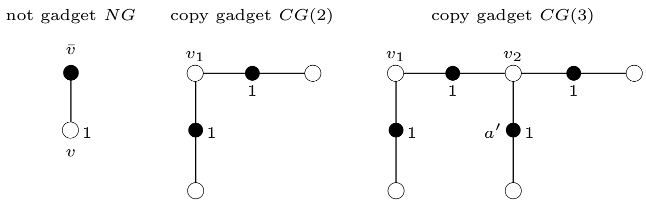

Figure 1: Basic building blocks of the graph G_F.

Proposition 4. *(i) The not-gadget NG is 0/1-colorable. Moreover in any 0/1-coloring of the not-gadget the vertex $\bar{v}$ is colored with a different color than vertex v.*

(ii) The copy gadget $CG(i)$ is 0/1-colorable. Moreover in any 0/1-coloring of $CG(i)$, all i leaves have identical colors with the root of the gadget.

Proof. For each gadget a 0/1-coloring is indicated on Figure 1. All the statements are easily verified. □

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 constraint 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.

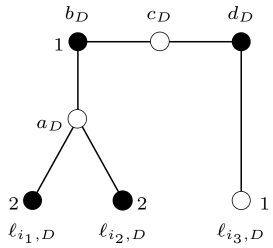

Figure 2: The clause-gadget G^*_D for clause $D = (l_{i_1} \vee l_{i_2} \vee l_{i_3})$.

Proposition 5. *An* 0/1*-coloring* 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-coloring 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 color* 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-coloring of G^*_D. Then a_D must be colored 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 that its constraint is 1. We conclude that an extension to a 0/1-coloring of G^*_D is not possible.

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

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-coloring 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-coloring 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.

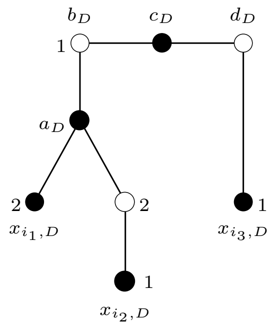

Figure 3: The extended clause-gadget G_D for the clause $D = (x_{i_1} \vee \bar{x}_{i_2} \vee x_{i_3})$.

Proposition 6. *An assignment α satisfies the clause D if and only if there is a 0/1-coloring of the extended clause-gadget G_D such that the vertices corresponding to the variables receive the colors 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 proposition. □

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 than 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 6. *(i) G_F is 0/1-colorable if and only if F is satisfiable.*

(ii) $\Delta(G_F) \leq 3$ 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-coloring of G_F by assigning color $\alpha(x)$ to the root of the copy-gadget C_x corresponding to the variable x. This can be extended to an 0/1-coloring of the copy-gadgets by part *(ii)* of Proposition 4 where the leaves receive the same color 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 colorings of the extended clause-gadgets can be extended to a 0/1-coloring of the whole gadget (cf. Proposition 6) and thus the whole graph G_F is 0/1-colored.

Let now χ be a 0/1-coloring of G_F. We claim that the colors given to the roots of the copy-gadgets corresponding to the variable is a satisfying assignment of F. By part *(ii)* of Proposition 4 all the leaves are the same color as their roots in the copy-gadget. By Proposition 6 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 in polynomial time a graph $RelColGraph(F)$ which is C-relaxed colorable if and only if the formula F is satisfiable.

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

Definition 1. *Let $C \geq 2$ and $\Delta \geq 1$ be integers. A graph G is called* (Δ, C)-forcing *with forced vertex $f \in V(G)$ if*

(i) $\Delta(G) \leq \Delta$ and f has degree at most $\Delta - 1$,

(ii) G is C-relaxed colorable, and

(iii) f is contained in I for every C-relaxed two-coloring of G.

Lemma 3. *For any integer $\Delta \geq 1$ and integer $C \geq 2$ the decision problem (Δ, C)-AsymRelCol is* NP*-complete provided a (Δ, C)-forcing graph exists.*

Proof. We assume the existence of a (Δ, C)-forcing graph H, hence $\Delta \geq 3$. 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 coloring.

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 7. *The base gadget BG_l is C-relaxed colorable for every $l \leq C$. Moreover in any C-relaxed coloring of BG_l, $l \leq C$, the sink is contained in a B-component of order l.*

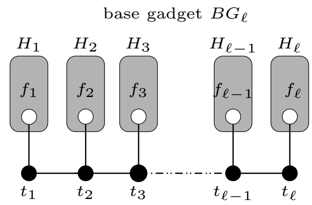

Figure 4: The base gadget BG_l

Proof. A C-relaxed coloring of the base-gadget is indicated on Figure 4. In any C-relaxed coloring 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 $i \in \{1, \dots, l\}$ form a B-component of order exactly l. □

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 connect 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 6. Note also that G_F is 0/1-colorable if and only if $RelColGraph(F)$ has a C-relaxed coloring. A C-relaxed coloring of $RelColGraph(F)$ restricted to $V(G_F)$ is a 0/1-coloring if we exchange the color I to 1 and the color B to 0. Conversely a 0/1-coloring of G_F can be extended to a C-relaxed coloring of $RelColGraph(F)$ by identifying 1 with I, and 0 with B, and extending this coloring to the base-gadgets appropriately (such coloring exists by Proposition 7). □

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-colorable.

Lemma 4. *For all $C \geq 2$, if $\mathcal{G}_C \neq \emptyset$ then there is a $(3, C)$-forcing graph.*

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

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

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 5) with $V(T) = \{t_1, t_2, u\}$ and $e = \{u, v\}$ 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 (cf. Figure 5). We claim that H is $(3, C)$-forcing graph with forced vertex f. H is C-relaxed colorable since the minimality of G ensures that $G - e$ has a C-relaxed coloring while the non-C-relaxed-colorability of G ensures that the colors of u and v are the same on *any* C-relaxed coloring of $G - e$. So *any* C-relaxed coloring χ of $G - e$ can be extended to a C-relaxed coloring of H by coloring f to the opposite of the color 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 coloring of H, then χ restricted to $V(G)$ is a C-relaxed coloring of G, a contradiction.

Thus in any C-relaxed two-coloring 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 neighbors of t_1 and t_2, respectively, not contained in T (might be $v_1 = v_2$). Suppose the vertices (u, f, v) of H can be colored with (I, B, I). But then $\chi_H(t_1) = \chi_H(t_2) = B$.

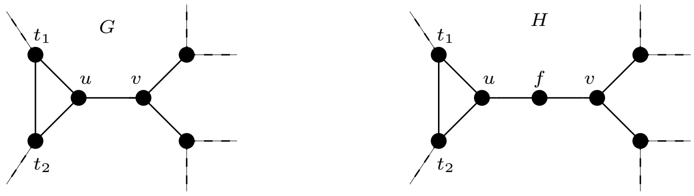

Figure 5: Splitting $e = \{u, v\}$ into $e_1 = \{u, f\}$ and $e_2 = \{f, v\}$

Case (i)**:** If $\chi_H(v_1) = \chi_H(v_2) = I$ then we define a C-relaxed two-coloring χ_G for G as follows: $\chi_G(x) = \chi_H(x)$ for all $x \in V(G) \setminus \{u\}$ and $\chi_G(u) = B$.
Case (ii)**:** Without loss of generality $\chi_H(v_1) = B$. We define a C-relaxed two-coloring χ_G for G as follows:
$\chi_G(x) = \chi_H(x)$ for all $x \in V(G) \setminus \{t_1, u\}$, $\chi_G(t_1) = I$, 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-colorable, a contradiction. Thus in any C-relaxed two-coloring of H the vertices (u, f, v) are colored (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 $2 \leq C \leq 5$ 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-colorable: just take $I = \{f, t_2', t_3''\}$ and $B = V(G) \setminus I$. It is also not hard to check that there is no 3-relaxed two-coloring where vertex f is contained in B. Suppose there is a 3-relaxed two-coloring 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.

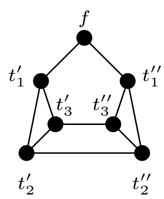

Figure 6: $(3, C)$-forcing graph for $C \in \{2, 3\}$

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-coloring and (ii) there is no 5-relaxed coloring 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-coloring of H^*,

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

(ii) The key observation is that in any 5-relaxed coloring of H^*, 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 coloring 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 neighbors 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.

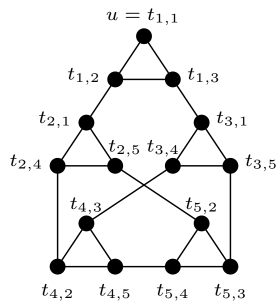

Figure 7: Graph H^*

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

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

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

Note that $(3, C)$-AsymRelCol is obviously trivial for all C with $\mathcal{G}_C = \emptyset$, so Theorem 3 follows immediately from Lemma 4 and Lemma 3.

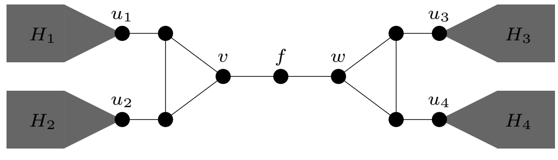

Figure 8: $(3, C)$-forcing graph for $C \in \{4, 5\}$

4.2.2 $(4, C)$-forcing graphs

Lemma 6. *For all $\Delta \geq 4$ and all $C \geq 2$ there is a (Δ, C)-forcing graph.*

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-colorable, 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 neighboring 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-colorable and in any such coloring 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 $v_{1,2}$).

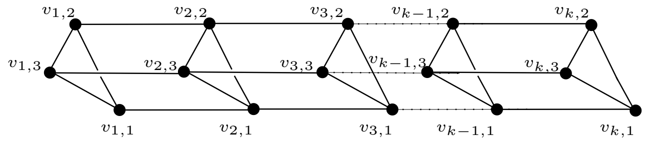

Figure 9: G_k with one B-component of order $2k$

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-colorable, hence $H - e$ is $(v, 2k-1)$-forcing again with forcing vertex $v_{1,1}$ or $v_{1,2}$.

□

Combining Lemma 6 and Lemma 3 concludes the proof of Theorem 4.

4.3 Hard (Δ, C)-SymRelCol

In this subsection we prove Theorem 5 by constructing the appropriate base gadgets and defining the graph *SymRelColGraph*(F) which can be (C, C)-relaxed colored if and only if the formula F is satisfiable. We denote the two color classes of a (C, C)-relaxed two-coloring by B_1 and B_2.

Definition 2. *Let $C \geq 2$ and $\Delta \geq 4$ be integers. A graph G is called* (Δ, C)-sym-forcing *with a set $F \subseteq V(G)$ of at most two forced vertices if*

(i) $\Delta(G) \leq \Delta$ and $\sum_{f \in F}(\Delta - d(f)) \geq 2$,
(ii) G is (C, C)-relaxed two-colorable, and
(iii) for every (C, C)-relaxed two-coloring of G there is a color-class c such that every $f \in F$ is contained in a c-component of order at least C.

Lemma 7. *For any two integers $\Delta \geq 4$ and $C \geq 2$ the decision problem (Δ, C)-SymRelCol is* NP*-complete provided a (Δ, C)-sym-forcing graph exists.*

Proof. Suppose a (Δ, C)-sym-forcing graph H exists. We will reduce our problem to 3-SAT. As in the asymmetric problem, the graph we construct will be an extension of the core graph G_F.

But the base-gadgets will be different from the ones in the previous subsection and some of them will be connected to each other unlike in the asymmetric problem.

We construct our base-gadget BG_l by taking l copies $H_1, \ldots, H_l$ of the (Δ, C)-sym-forcing graph H and l vertices $s_1, \ldots, s_l$ and connecting them in a path-like fashion as depicted in Figure 10.

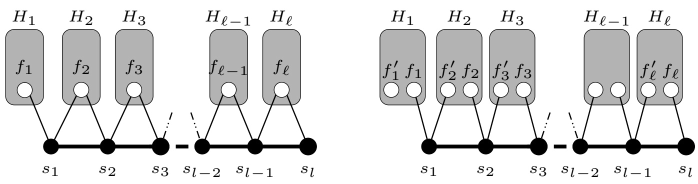

Figure 10: Base gadget BG_l using sym-forcing graphs with either one or two forced vertices.

By property (i) of Definition 2, $\Delta(B_l) \leq \Delta$. By property (iii), in any (C, C)-relaxed coloring of the base gadget BG_l all the forced vertices f_i have the same color, which is different from the color of the vertices s_i. Thus the vertices s_i form a monochromatic component of order l. We call f_1 and s_l the *source* and *sink* of the base gadget, respectively.

We can conclude the following.

Proposition 8. *The base gadget BG_l is (C, C)-relaxed colorable for every $l \leq C$. Moreover, in any (C, C)-relaxed coloring of BG_l the sink is contained in a monochromatic component of order exactly l whose color is different from the color of the source.*

Suppose now that we are given a 3-SAT formula F. We construct the graph $SymRelColGraph(F)$ by connecting various base gadgets to vertices of G_F via an edge. First, we connect the sink of a base gadget $B_w^{(1)} \cong BG_{C-2}$ to each vertex w of G_F which has constraint 2 and the sink of a base gadget $B_w^{(1)} \cong BG_{C-1}$ to each vertex w with a constraint 1. These we call the *base-gadgets of the first-type*. Further, we connect the sink of a base-gadget $B_w^{(2)} \cong BG_{C-1}$ to every vertex w of G_F. These we call the *base gadgets of the second-type*. Note that by part (ii) of Theorem 6, after adding these new edges the degree of each vertex of G_F is at most four. Also, the sink of each base-gadget now has degree three, and the source has degree at most $\Delta - 1$.

Then, by adding an edge between some sink and source vertices, we connect all base gadgets of the first-type in a path-like fashion, pictured on Figure 11. We act similarly for base gadgets of the second-type. Finally, we add an edge between the source of the first base gadget of the first-type and the source of the first base gadget of the second-type. Let us denote this new graph by $SymRelColGraph(F)$. By the above, the maximum degree of $SymRelColGraph(F)$ is clearly at most Δ. For an insight about the connections between the base-gadgets, see Figure 11.

The following is an immediate corollary of Proposition 8

Proposition 9. *In any (C, C)-relaxed coloring of $SymRelColGraph(F)$ the sinks of base-gadgets of the first-type are all colored with the same color, say B_1. On the other hand the sinks of base-gadgets of the second-type are all colored with the other color, B_2.*

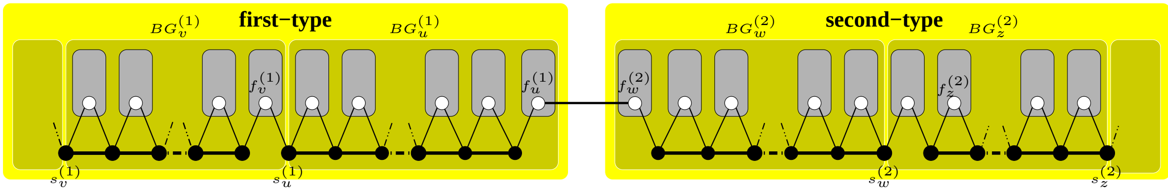

Figure 11: Connecting base gadgets of the first- and second-type

We claim that there is a (C, C)-relaxed two-coloring of $SymRelColGraph(F)$ if and only if the core G_F has a 0/1-coloring. Via Theorem 6, this will conclude the proof of Lemma 7.

Suppose χ is a (C, C)-relaxed coloring of $SymRelColGraph(F)$. Suppose the sinks of the base-gadgets of the first-type all received color B_1. (We know they are all the same from Proposition 9) Then all sinks of the base-gadgets of the second-type must receive color B_2. Changing color B_1 to 0 and color B_2 to 1 gives us the the 0/1-coloring of G_F which observes all the constraints.

Conversely, suppose we are given a 0/1-coloring of G_F. We can extend this to a (C, C)-coloring of $SymRelColGraph(F)$ by arbitrarily selecting either 0 or 1 to color the forced vertices of the base-gadgets of the first-type and then extending this coloring to all vertices of all base-gadgets. □

4.3.1 (Δ, C)-sym-forcing graphs

We are able to show the existence of $(4, C)$-sym-forcing graphs with $C \in \{2, 3\}$ and $(6, C)$-sym-forcing graphs, for $C \geq 2$. This will conclude the proof of Theorem 5.

Proposition 10. *The graph G in Figure 12 is $(4, 2)$-sym-forcing with a forced set $\{f', f''\}$.*

Proof. Adding the edge $\{f', v\}$ to G yields a graph that is not $(2, 2)$-relaxed colorable. In order to not contradict this fact, f' and v must have the same color in any $(2, 2)$-relaxed coloring G, whereas the two common neighbors of f' and v are contained in the other color-class. Hence also f'' is contained in the same color-class as f' and v. □

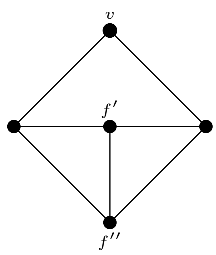

Figure 12: $(4, 2)$-sym-forcing graph

To construct a $(4, 3)$-sym-forcing graph we introduce a weakened concept of Definition 2 which makes it easier to construct (Δ, C)-sym-forcing graphs, and thus is interesting in its own right.

Definition 3. *Let $C \geq D \geq 2$ and $\Delta \geq 4$ be integers. A graph G is called* (Δ, C, D)-sym-forcing *with a set $F \subseteq V(G)$ of at most two forced vertices if*

(i) $\Delta(G) \leq \Delta$ and $\sum_{f \in F}(\Delta - d(f)) \geq 2$,

(ii) G is (C, C)-relaxed two-colorable, and

(iii) for every (C, C)-relaxed two-coloring of G there is a color class c such that every $f \in F$ is contained in a c-monochromatic component of order at least D.

Observe that (Δ, C, C)-sym-forcing is the same as (Δ, C)-sym-forcing.

Proposition 11. *The existence of a $(\Delta, C, \lceil \frac{C+1}{2} \rceil)$-sym-forcing graph implies the existence of a (Δ, C, D)-sym-forcing graph for every D, $\lceil \frac{C+1}{2} \rceil \leq D \leq C$.*

Proof. Let G_1 and G_2 be two copies of an (Δ, C, i)-sym-forcing graph, $\lceil \frac{C+1}{2} \rceil \leq i \leq C-1$. First assume that we have one forcing vertex in G_i. We connect the forcing vertex f_1 of G_1 to the forcing vertex f_2 of G_2. Also we add a new vertex v to the new graph, denote it by H, and connect it to f_1 and f_2, see Figure 13. Suppose f_1 and f_2 are contained in the same color-class in a (C, C)-relaxed coloring of H, then the two adjacent vertices f_1 and f_2 are contained in one monochromatic component of order at least $2i \geq C+1$, a contradiction. Thus without loss of generality $f_1 \in B_1$ and $f_2 \in B_2$. We conclude that v is contained in a monochromatic component of order $i+1$. The construction for the case when the G_i's have two forcing f_i' and f_i'' vertices is depicted in Figure 13 as well. The proof is very similar to the former case. □

Figure 13: Weakly forcing graphs

Proposition 12. *The graph in Figure 14 is $(4, 3, 2)$-sym-forcing with forced vertex f.*

Proof. The graph is $(3, 3)$-relaxed colorable. It is then sufficient to observe that in a $(3, 3)$-relaxed coloring the neighbors v_1, v_2 of f cannot have the same color, so f participates in a monochromatic component of order at least two. Let us assume to the contrary that v_1 and v_2 are both contained in color-class B_2. Obviously at least three out of the four neighbors of v_1 and v_2 (not considering f) have to be contained in B_1 in order to not span a B_2-component of order 4. On the other hand not all of the four vertices can be contained in B_1. Hence the unique common neighbor v of those four vertices is incident to a B_1 component and a B_2-component, each of order 3. We conclude that v cannot be colored. Hence one of v_1 and v_2 has a color identical to that of f, that is, f is a forced vertex. □

The previous two propositions imply the existence of a $(4, 3)$-sym-forcing graph.

Definition 4. *For a positive integer C, let T_C be the graph whose vertices are the triples (x, y, z) of nonnegative integers summing to C, with an edge connecting two triples if they agree in one*

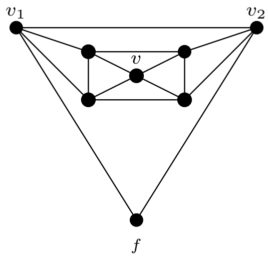

Figure 14: $(4,3,2)$-forcing

coordinate and differ by one in the remaining two coordinates. We denote by v, w and f the following three vertices, respectively: $v = (0, C, 0), w = (0, C-1, 1)$ and $f = (1, C-1, 0)$.

The graph T_4 is shown in Figure 15. Let H_{C-1} denote the graph T_C with the edge $\{v, f\}$ removed.

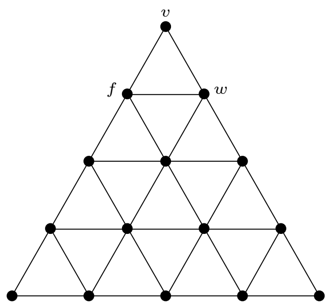

Figure 15: The graph T_4

Proposition 13. *H_{C-1} is $(6, C)$-sym-forcing for $2 \leq C$ with forced vertex f.*

Proof. It is not hard to check that H_{C-1} is (C, C)-relaxed colorable. We will use the following lemma.

Lemma 8 ([14]). *T_C is not (C, C)-relaxed colorable.*

The following three properties of (C, C)-relaxed colorings of H_{C-1} are immediate consequences of the lemma.

(i) v and f are contained in the same color-class.

(ii) w is contained in the other color-class than v and f.

(iii) The order of the union of the monochromatic component containing v and containing f is at least $C + 1$.

According to (ii) and the fact that v has a unique neighbor w, v is contained in a monochromatic component of order exactly 1. We conclude due to (iii) that f is contained in a monochromatic component of order C always. □

5 Summarizing Overview and Open Problems.

It would be interesting to determine exactly the critical monochromatic component order $f(3)$ from where the problem $(3, C)$-AsymRelCol becomes trivial. In Figure 16 we overview the results about 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**).

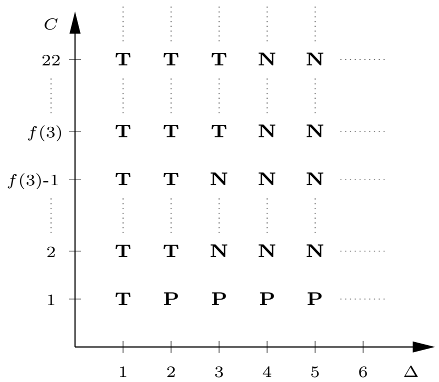

Figure 16: Hardness of (Δ, C)-AsymRelCol

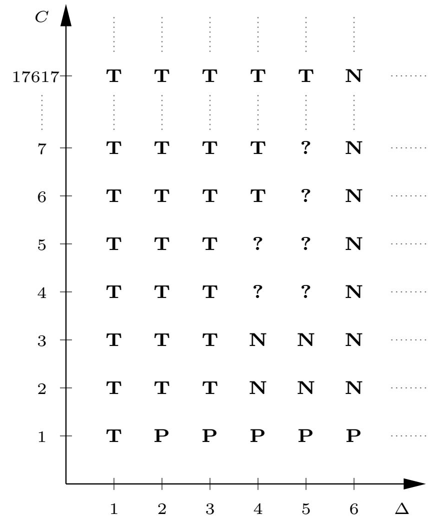

Figure 17: Hardness of (Δ, C)-SymRelCol

We conjecture that there is a sudden jump in the hardness of the problem $(4, C)$-SymRelCol. Such a result would particularly be interesting, since here the determination 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 $C, 2 \leq C < g(4)$, it is NP-hard to decide whether a given graph G of maximum degree 4 has a (C, C)-relaxed coloring and*
- *every graph of maximum degree 4 is $(g(4), g(4))$-relaxed colorable.*

The similar problem is wide open for graphs with maximum degree 5: Does $(5, C)$-SymRelCol exhibit a monotone behavior for $C \geq 2$? Is there a "jump in hardness"? Again we overview the hardness results about deciding (Δ, C)-SymRelCol in a table, see Figure 17.

For colorings with more than two colors we know much less. Even the graph theoretic questions about 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-colored such that every monochromatic component is of order at most C_k.*

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

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

Open Problem 2. *Is there a constant C such that every graph with maximum degree 9 can be three-colored 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-colored 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 colorable. Motivated by the fact that $g(n, 5) = O(1)$ [12] and their result [18] 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], for any two-coloring of the graph T_n (which has maximum degree 6) does contain a monochromatic component of order $\Omega(\sqrt{n})$.

References

[1] J. Akiyama, V. Chvátal, A short proof of the linear arboricity for cubic graphs, *Bull. Liber. Arts Sci. NMS* 2 (1981).

[2] J. Akiyama, G. Exoo, F. Harary, Coverings and packings in graphs II. Cyclic and acyclic invariants. *Math. Slovaca* **30** (1980), 405-417.

[3] N. Alon, G. Ding, B. Oporowski, D. Vertigan, Partitioning into graphs with only small components, *Journal of Combinatorial Theory, (Series B)*, **87**(2003) 231-243.

[4] J. A. Andrews and M. S. Jacobson. On a generalization of chromatic number. In Proceedings of the sixteenth Southeastern international conference on combinatorics, graph theory and computing (Boca Raton, Fla., 1985), volume 47, pages 33?48, 1985.

[5] R. Berke, T. Szabó, Relaxed Coloring of Cubic Graphs, *submitted*; extended abstract in *Proceedings EuroComb* (2005), 341-344.

[6] T.C. Biedl, P. Bose, E.D. Demaine, A. Lubiw, Efficient Algorithms for Petersen's Matching Theorem, *Journal of Algorithms*, **38**, (2001), 110-134.

[7] L. J. Cowen, R. H. Cowen, and D. R. Woodall. Defective colorings of graphs in surfaces: partitions into subgraphs of bounded valency. J. Graph Theory, 10(2):187?195, 1986.

[8] N. Eaton and T. Hull. Defective list colorings of planar graphs. Bull. Inst. Combin. Appl., 25:79?87, 1999.

[9] K. Edwards, G. Farr, Fragmentability of graphs, *J. Comb. Th. Series B*, **82**, no. 1, (2001), 30-37.

[10] F. Harary. Conditional colorability in graphs. In Graphs and applications (Boulder, Colo., 1982), Wiley-Intersci. Publ., pages 127?136. Wiley, New York, 1985.

[11] F. Harary and K. F. Jones. Conditional colorability. II. Bipartite variations. In Proceedings of the Sundance conference on combinatorics and related topics (Sundance, Utah, 1985), volume 50, pages 205?218, 1985.

[12] P. Haxell, T. Szabó, G. Tardos, Bounded size components — partitions and transversals, *Journal of Combinatorial Theory (Series B)*, **88** no.2. (2003) 281-297.

[13] P. Haxell, O. Pikhurko, A. Thomason, Maximum Acyclic and Fragmented Sets in Regular Graphs, *submitted*.

[14] R. Hochberg, C. McDiarmid, M. Saks, On the bandwidth of triangulated triangles. *Discrete Mathematics* **138** (1995), no. 1-3, 261–265.

[15] S. Hoory, S. Szeider, Families of unsatisfiable k-CNF formulas with few occurrences per variable, *arXiv.org e-Print archive*, math.CO/0411167, (2004).

[16] S. Hoory, S. Szeider.Computing Unsatisfiable k-SAT Instances with Few Occurrences per Variable, *Theoretical Computer Science*, **337**, no. 1-3, (2005) 347-359.

[17] J. Kratochvíl, P. Savický, Zs. Tuza, One more occurrence of variables makes satisfiability jump from trivial to NP-complete, *SIAM J. Comput.* **22** (1) (1993) 203-210.

[18] N. Linial, J. Matoušek*private communication.*

[19] N. Linial, M. Saks, Low diameter graph decompositions, *Combinatorica*, **13** no.4. (1993) 441-454.

[20] P. Savický, J. Sgall, DNF tautologies with a limited number of of occurrences of every variable, *Theoret. Comput. Sci.*, **238** (1-2) (2000) 495-498.

[21] R. Škrekovski, List improper colorings of planar graphs, *Comb. Prob. Comp.*, **8** (1999), 293-299.

[22] C. Thomassen, Two-colouring the edges of a cubic graph such that each monochromatic component is a path of length at most 5, *J. Comb. Th. Series B* **75**, (1999), 100-109.

[23] C. Tovey, A simplified NP-complete satisfiability problem, *Discrete Appl. Math.*, **8** no.1 (1984) 85-89.

[24] M. L. Weaver, D. B. West, Relaxed Chromatic Numbers of Graphs, *Graphs and Combinatorics*, **10** (1994), 75-93.