Discrete Mathematics and Theoretical Computer Science DMTCS vol. **23**:1, 2021, #14

Destroying Bicolored P_3s by Deleting Few Edges

Niels Grüttemeier Christian Komusiewicz
Jannik Schestag Frank Sommer[*]

Fachbereich Mathematik und Informatik, Philipps-Universität Marburg, Marburg, Germany

received 2020-02-17, *revised 2021-04-09*, *accepted 2021-05-07*.

We introduce and study the BICOLORED P_3 DELETION problem defined as follows. The input is a graph $G = (V, E)$ where the edge set E is partitioned into a set E_r of red edges and a set E_b of blue edges. The question is whether we can delete at most k edges such that G does not contain a bicolored P_3 as an induced subgraph. Here, a bicolored P_3 is a path on three vertices with one blue and one red edge. We show that BICOLORED P_3 DELETION is NP-hard and cannot be solved in $2^{o(|V|+|E|)}$ time on bounded-degree graphs if the ETH is true. Then, we show that BICOLORED P_3 DELETION is polynomial-time solvable when G does not contain a bicolored K_3, that is, a triangle with edges of both colors. We also provide a polynomial-time algorithm for the case that G contains no blue P_3, red P_3, blue K_3, and red K_3. Finally, we show that BICOLORED P_3 DELETION can be solved in $\mathcal{O}(1.84^k \cdot |V| \cdot |E|)$ time and that it admits a kernel with $\mathcal{O}(k\Delta \min(k, \Delta))$ vertices, where Δ is the maximum degree of G.

Keywords: NP-hard problem, graph modification, edge-colored graphs, parameterized complexity, graph classes

Acknowledgment

We would like to thank the reviewers of *Discrete Mathematics and Theoretical Computer Science* for their helpful comments and Michał Pilipczuk (University of Warsaw) for pointing out the connection to Gallai colorings and their characterization. A preliminary version of this work appeared in *Proceedings of the 15th Conference on Computability in Europe (CiE '19)*, volume 11558 of *Lecture Notes in Computer Science*, pages 193–204. The full version contains all missing proofs and an improved running time analysis of the fixed-parameter algorithm. Some of the results of this work are also contained in the third author's Bachelor thesis [25].

1 Introduction

Graph modification problems are a popular topic in computer science. In these problems, one is given a graph and wants to apply a minimum number of modifications, for example edge deletions, to obtain a graph that fulfills some graph property Π.

[*] FS was supported by the DFG, project MAGZ (KO 3669/4-1).

An important reason for the popularity of graph modification problems is their usefulness in graph-based data analysis. A classic problem in this context is CLUSTER EDITING where we may insert and delete edges and Π is the set of cluster graphs. These are exactly the graphs that are disjoint unions of cliques and it is well-known that a graph is a cluster graph if and only if it does not contain a P_3, a path on three vertices, as induced subgraph. CLUSTER EDITING has many applications [4], for example in clustering gene interaction networks [3] or protein sequences [30]. The variant where we may only delete edges is known as CLUSTER DELETION [26]. Further graph-based data analysis problems that lead to graph modification problems for some graph property Π defined by small forbidden induced subgraphs arise in the analysis of biological [8, 18] or social networks [6, 24].

Besides the application, there is a more theoretical reason why graph modification problems are very important in computer science: Often these problems are NP-hard [23, 31] and thus they represent interesting case studies for algorithmic approaches to NP-hard problems. For example, by systematically categorizing graph properties based on their forbidden subgraphs one may outline the border between tractable and hard graph modification problems [2, 22, 31].

In recent years, multilayer graphs have become an increasingly important tool for integrating and analyzing network data from different sources [21]. Formally, multilayer graphs can be viewed as edge-colored (multi-)graphs, where each edge color represents one layer of the input graph. With the advent of multilayer graphs in network analysis it can be expected that graph modification problems for edge-colored graphs will arise in many applications as it was the case in uncolored graphs.

One example for such a problem is MODULE MAP [27]. Here, the input is a simple graph with red and blue edges and the aim is to obtain by a minimum number of edge deletions and insertions a graph that contains no P_3 with two blue edges, no P_3 with a red and a blue edge, and no a triangle, called K_3, with two blue edges and one red edge. MODULE MAP arises in computational biology [1, 27]; the red layer represents genetic interactions and the blue layer represents physical protein interactions [1].

Motivated by the practical application of MODULE MAP, an edge deletion problem with bicolored forbidden induced subgraphs, we aim to study such problems from a more systematic and algorithmic point of view. Given the importance of P_3-free graphs in the uncolored case, we focus on the problem where we want to destroy all *bicolored P_3s*, that is, all P_3s with one blue and one red edge, by edge deletions.

BICOLORED P_3 DELETION (BPD)
Input: A two-colored graph $G = (V, E_r, E_b)$ and an integer $k \in \mathbb{N}$.
Question: Can we delete at most k edges from G such that the remaining graph contains no bicolored P_3 as induced subgraph?

We use $E := E_r \uplus E_b$ to denote the set of all edges of G, $n := |V|$ to denote the number of vertices in G, and $m := |E|$ to denote the number of edges in G.

Bicolored P_3s are closely connected to Gallai colorings of complete graphs [14, 17]. A Gallai coloring is an edge-coloring such that the edges of every triangle receive at most two different colors. When we view nonedges of G as edges with a third color, say green, then a bicolored P_3 is the same as a triangle that violates the property of Gallai colorings. Thus, BPD is essentially equivalent to the following problem: Given a complete graph with an edge-coloring with the colors red, blue, and green that is not a Gallai coloring, can we transform the coloring into a Gallai coloring by recoloring at most k blue or red edges with the color green?

Our Results. We show that BPD is NP-hard and that, assuming the Exponential-Time Hypothesis (ETH) [20], it cannot be solved in a running time that is subexponential in the instance size. We then study two different aspects of the computational complexity of the problem.

First, we consider special cases that can be solved in polynomial time, motivated by similar studies for problems on uncolored graphs [7]. We are in particular interested in whether or not we can exploit structural properties of input graphs that can be expressed in terms of *colored* forbidden subgraphs. We show that BPD can be solved in polynomial time on graphs that do not contain a certain type of bicolored K_3s as induced subgraphs, where bicolored K_3s are triangles with edges of both colors. Moreover, we show that BPD can be solved in polynomial time on graphs that contain no K_3s with one edge color and no P_3s with one edge color as induced subgraphs.

Second, we consider the parameterized complexity of BPD with respect to the natural parameter k. We show that BPD can be solved in $\mathcal{O}(1.84^k \cdot nm)$ time and that it admits a problem kernel with $\mathcal{O}(k\Delta \min(k, \Delta))$ vertices, where Δ is the maximum degree in G. As a side result, we show that BPD admits a trivial problem kernel with respect to $\ell := m - k$.

2 Preliminaries

We consider undirected simple graphs G with vertex set V and edge set E, where E is partitioned into a set E_b of *blue edges* and a set E_r of *red edges*, denoted by $G = (V, E_r, E_b)$. For a vertex v, $N_G(v) := \{u \mid \{u, v\} \in E\}$ denotes the *open neighborhood* of v and $N_G[v] := N_G(v) \cup \{v\}$ denotes the *closed neighborhood* of v. For a vertex set W, $N_G(W) := \bigcup_{w \in W} N(w) \setminus W$ denotes the *open neighborhood* of W and $N_G[W] := N_G(W) \cup W$ denotes the *closed neighborhood* of W. The *degree* $\deg(v) := |N_G(v)|$ of a vertex v is the size of its open neighborhood. We let $N_G^2(v) := N_G(N_G(v)) \setminus \{v\}$ denote the *second neighborhood* of v. For any two vertex sets $V_1, V_2 \subseteq V$, we denote by $E_G(V_1, V_2) := \{\{v_1, v_2\} \in E \mid v_1 \in V_1, v_2 \in V_2\}$ the set of edges between V_1 and V_2 in G and write $E_G(V') := E_G(V', V')$. In each context we may omit the subscript G if the graph is clear from the context.

For any $V' \subseteq V$, $G[V'] := (V', E(V') \cap E_r, E(V') \cap E_b)$ denotes the *subgraph induced by* V'. We say that some graph $H = (V^H, E_r^H, E_b^H)$ is an *induced subgraph* of G if there is a set $V' \subseteq V$, such that $G[V']$ is isomorphic to H, otherwise G is called *H-free*. Two vertices u and v are *connected* if there is a path from u to v in G. A *connected component* is a maximal vertex set S such that each two vertices are connected in $G[S]$. A *clique* in a graph G is a set $K \subseteq V$ of vertices such that in $G[K]$ each pair of vertices is adjacent. The graph $(\{u, v, w\}, \{\{u, v\}\}, \{\{v, w\}\})$ is called *bicolored* P_3. We say that a vertex $v \in V$ is *part of* a bicolored P_3 in G if there is a set $V' \subseteq V$ with $v \in V'$ such that $G[V']$ is a bicolored P_3. Furthermore, we say that two edges $\{u, v\}$ and $\{v, w\}$ *form* a bicolored P_3 if $G[\{u, v, w\}]$ is a bicolored P_3. An edge e is *part of a bicolored* P_3 if there exists some other edge e' such that e and e' form a bicolored P_3. For any edge set E' we denote by $G - E' := (V, E_r \setminus E', E_b \setminus E')$ the graph we obtain by deleting all edges in E'. As a shorthand, we write $G - e := G - \{e\}$ for an edge e. An edge deletion set S is a *solution* for an instance (G, k) of BPD if $G - S$ is bicolored-P_3-free and $|S| \leq k$.

A *branching rule* for some problem L is a computable function that maps an instance w of L to a tuple of instances $(w_1, \dots, w_t)$ of L. A branching rule is called *correct* if w is a yes-instance for L if and only if there is some $i \in \{1, \dots, t\}$ such that w_i is a yes-instance of L. The application of branching rules gives rise to a search tree whose size is analyzed using branching vectors; for more details refer to the textbook of Fomin and Kratsch [13]. A *reduction rule* for some problem L is a computable function that maps an instance w of L to an instance w' of L such that w is a yes-instance if and only if w' is a yes-instance.

Parameterized Complexity is the analysis of the complexity of problems depending on the input size n and a problem parameter k [9, 10]. A problem is called *fixed-parameter tractable* if there exists an algorithm with running time $f(k) \cdot n^{\mathcal{O}(1)}$ for some computable function f that solves the problem. An important tool in the development of parameterized algorithms is *problem kernelization*. Problem kernelization is a polynomial-time preprocessing by *reduction rules*: A problem L admits a problem kernel if, given any instance I of L with parameter k, one can compute an equivalent instance I' of L with parameter k' in polynomial time such that $k' \leq k$ and the size of I' is bounded by some computable function g only depending on k. The function g is called *kernel size*. The *Exponential Time Hypothesis (ETH)* is a standard complexity theoretical conjecture used to prove lower bounds. It implies that 3-SAT cannot be solved in $2^{o(|\phi|)}$ time where ϕ denotes the input formula [20].

3 Bicolored $\mathbf{P_3}$ Deletion is NP-hard

In this section we prove the NP-hardness of BPD. This motivates our study of polynomial-time solvable cases and the parameterized complexity in Sections 4 and 5, respectively.

Theorem 3.1. BPD *is NP-hard even if the maximum degree of G is* 8.

Proof: We present a polynomial-time reduction from the NP-hard (3,4)-SAT problem where one is given a 3-CNF formula ϕ where each variable occurs in at most four clauses, and the question is if there is a satisfying assignment for ϕ [29].

Let ϕ be a 3-CNF formula with variables $X = \{x_1, \dots, x_{|X|}\}$ and clauses $\mathcal{C} = \{C_1, \dots, C_{|\mathcal{C}|}\}$ with four occurrences per variable. For a given variable x_i that occurs in a clause C_j we define the *occurrence number* $\Psi(C_j, x_i)$ as the number of clauses in $\{C_1, C_2, \dots, C_j\}$ where x_i occurs. Intuitively, $\Psi(C_j, x_i) = r$ means that the rth occurrence of variable x_i is the occurrence in clause C_j. Since each variable occurs in at most four clauses, we have $\Psi(C_j, x_i) \in \{1, 2, 3, 4\}$.

Construction: We describe how to construct an equivalent instance $(G = (V, E_r, E_b), k)$ of BPD from ϕ.

For each variable $x_i \in X$ we define a *variable gadget* as follows. The variable gadget of x_i consists of a *central vertex* v_i and two vertex sets $T_i := \{t_i^1, t_i^2, t_i^3, t_i^4\}$ and $F_i := \{f_i^1, f_i^2, f_i^3, f_i^4\}$. We add a blue edge from v_i to every vertex in T_i and a red edge from v_i to every vertex in F_i.

For each clause $C_j \in \mathcal{C}$ we define a *clause gadget* as follows. The clause gadget of C_j consists of three vertex sets $A_j := \{a_j^1, a_j^2, a_j^3\}$, $B_j := \{b_j^1, b_j^2, b_j^3\}$, and $W_j := \{w_j^1, w_j^2, w_j^3, w_j^4\}$. We add blue edges such that the vertices in $B_j \cup W_j$ form a clique with only blue edges in G. Moreover, for each $p \in \{1, 2, 3\}$, we add a blue edge $\{a_j^p, b_j^p\}$ and a red edge $\{a_j^p, u\}$ for every $u \in W_j \cup B_j \setminus \{b_j^p\}$. Observe that there are no edges between a_j^1, a_j^2, and a_j^3; all other vertex pairs are connected either by a red edge or a blue edge.

We connect the variable gadgets with the clause gadgets by identifying vertices in $T_i \cup F_i$ with vertices in A_j as follows. Let C_j be a clause containing variables x_{i_1}, x_{i_2}, and x_{i_3}. For each $p \in \{1, 2, 3\}$ we set

$$a_j^p = \begin{cases} t_{i_p}^{\Psi(C_j, x_{i_p})} & \text{if } x_{i_p} \text{ occurs as a positive literal in } C_j \text{, and} \\ f_{i_p}^{\Psi(C_j, x_{i_p})} & \text{if } x_{i_p} \text{ occurs as a negative literal in } C_j. \end{cases}$$

Now, for every variable $x_i \in X$ each vertex in $T_i \cup F_i$ is identified with at most one vertex a_j^p. Figure 1 shows an example of a clause gadget and its connection with the variable gadgets. To complete the construction of the BPD instance (G, k) we set $k := 4 \cdot |X| + 14 \cdot |\mathcal{C}|$.

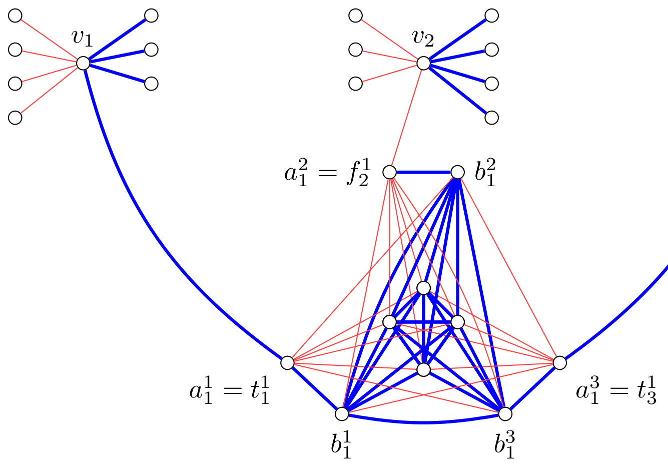

Figure 1: The lower part of the figure shows the clause gadget of a clause $C_1 = (x_1 \vee \overline{x_2} \vee x_3)$. The upper part of the figure shows variable gadgets representing variables x_1, x_2, and x_3. The vertices a_1^1, a_1^2, and a_1^3 from the clause gadget are identified with vertices from the variable gadgets. The bold lines represent blue edges and the thin lines represent red edges.

Intuition: Before showing the correctness of the reduction, we describe its idea. For each variable x_i we have to delete all blue edges in $E(\{v_i\}, T_i)$ or all red edges in $E(\{v_i\}, F_i)$ in the corresponding variable gadget. Deleting the edges in $E(\{v_i\}, T_i)$ assigns true to the variable x_i while deleting the edges in $E(\{v_i\}, F_i)$ assigns false to x_i. Since we identify vertices in $T_i \cup F_i$ with vertices in A_j the information of the truth assignment is transmitted to the clause gadgets. We will be able to make a clause-gadget bicolored-P_3-free with 14 edge deletions if and only if there is at least one vertex in A_j which is incident with a deleted edge of its variable gadget.

Correctness: We now show the correctness of the reduction by proving that there is a satisfying assignment for ϕ if and only if (G, k) is a yes-instance of BPD.

($\Rightarrow$) Let $\mathcal{A} : X \to \{\text{true}, \text{false}\}$ be a satisfying assignment for ϕ. In the following, we construct a solution S for (G, k).

For each variable x_i, we add $E(\{v_i\}, T_i)$ to S if $\mathcal{A}(x_i) = \text{true}$ and we add $E(\{v_i\}, F_i)$ to S if $\mathcal{A}(x_i) = \text{false}$. Note that for each variable we add exactly four edges to S. For each $C_j \in \mathcal{C}$ we add the following edges: Since $\mathcal{A}$ is satisfying, C_j contains a variable x_i such that $\mathcal{A}(x_i)$ satisfies C_j. By the construction of G there is exactly one $p \in \{1, 2, 3\}$ such that $a_j^p = t_i^{\Psi(C_j, x_i)}$ if x_i occurs as a positive literal in C_j or $a_j^p = f_i^{\Psi(C_j, x_i)}$ if x_i occurs as a negative literal in C_j. For both $q \in \{1, 2, 3\} \setminus \{p\}$ we add $E(\{a_j^q\}, W_j \cup B_j)$ to S. Note that since $|E(\{a_j^q\}, W_j \cup B_j)| = |W_j \cup B_j| = 7$ we add exactly 14 edges per clause. Thus, we have an overall number of $4 \cdot |X| + 14 \cdot |\mathcal{C}|$ edges in S.

Let $G' := G - S$. It remains to show that there is no bicolored P_3 in G'. Let $C_j \in \mathcal{C}$. We first show that no edge in $E_{G'}(A_j \cup B_j \cup W_j)$ is part of a bicolored P_3 in G'. For any two vertices $u_1, u_2 \in W_j \cup B_j$ it holds that $N_{G'}[u_1] = N_{G'}[u_2]$. Hence, no edge in $E_{G'}(W_j \cup B_j)$ is part of an induced P_3 in G' and therefore no edge in $E_{G'}(W_j \cup B_j)$ is part of a bicolored P_3 in G'. It remains to show that no edge in $E_{G'}(A_j, W_j \cup B_j)$ is part of a bicolored P_3 in G'. Without loss of generality assume that $E(\{a_j^1\} \cup \{a_j^2\}, W_j \cup B_j) \subseteq S$ and $E(\{a_j^3\}, W_j \cup B_j) \cap S = \emptyset$. Thus, the only possible edges in $E_{G'}(A_j \cup B_j \cup W_j)$ that might form a bicolored P_3 in G' are the edges in $E(\{a_j^3\}, W_j \cup B_j)$. We show for every $u \in W_j \cup B_j$ that $\{a_j^3, u\} \in E_{G'}(A_j, W_j \cup B_j)$ is not part of an induced bicolored P_3 in G'. Since $\{a_j^3\} \cup B_j \cup W_j$ is a clique in G', the edge $\{a_j^3, u\}$ may only form a bicolored P_3 with an edge $\{a_j^3, v_i\}$, where v_i is the central vertex of the variable gadget of a variable x_i occurring in C_j. By the construction of G it follows that $a_j^3 \in \{t_i^{\Psi(C_j,x_i)}, f_i^{\Psi(C_j,x_i)}\}$ for exactly one variable x_i. Since $E(\{a_j^3\}, W_j \cup B_j) \cap S = \emptyset$, the assignment $\mathcal{A}(x_i)$ satisfies clause C_j by the construction of S. If $\mathcal{A}(x_i) =$ true, then $\{a_j^3, v_i\} = \{t_i^{\Psi(C_j,x_i)}, v_i\} \in E_G(\{v_i\}, T_i) \subseteq S$ and therefore $\{a_j^3, v_i\}$ is not an edge of G'. Analogously, if $\mathcal{A}(x_i) =$ false, then $\{a_j^3, v_i\} = \{f_i^{\Psi(C_j,x_i)}, v_i\} \in E_G(\{v_i\}, F_i) \subseteq S$ is not an edge of G'. Hence, no edge in $E_{G'}(A_j \cup B_j \cup W_j)$ is part of a bicolored P_3 in G'.

Let $x_i \in X$. We show that no edge in $E_{G'}(\{v_i\} \cup T_i \cup F_i)$ is part of a bicolored P_3 in G'. Since either $E_G(\{v_i\}, T_i) \subseteq S$ or $E_G(\{v_i\}, F_i) \subseteq S$, all edges in $E_{G'}(\{v_i\} \cup T_i \cup F_i)$ have the same color. Hence, there is no bicolored P_3 in G' consisting of two edges from one variable gadget. Since there is no vertex in G' that is adjacent to two vertices of distinct variable gadgets, an edge $e \in E_{G'}(\{v_i\} \cup T_i \cup F_i)$ may only form a bicolored P_3 with an edge in $E_{G'}(A_j \cup B_j \cup W_j)$ for some clause C_j. However, since no edge in $E_{G'}(A_j \cup B_j \cup W_j)$ is part of a bicolored P_3 in G' as shown above, e does not form a bicolored P_3 with an edge from a clause gadget. Therefore, no edge in $E_{G'}(\{v_i\} \cup T_i \cup F_i)$ is part of a bicolored P_3. It follows that G' does not contain any bicolored P_3.

($\Leftarrow$) Conversely, let S be a solution for (G, k). For every variable $x_i \in X$ we have $|S \cap E_G(\{v_i\}, T_i \cup F_i)| \geq 4$ since every edge in $E_G(\{v_i\}, T_i)$ forms a bicolored P_3 with every edge in $E_G(\{v_i\}, F_i)$.

Before we define a satisfying assignment $\mathcal{A} : X \to \{\text{true}, \text{false}\}$ for ϕ, we take a more detailed look at the edges of the clause gadgets that need to be in S. Let $C_j \in \mathcal{C}$ be a clause and let $G_j := G[A_j \cup B_j \cup W_j]$ be the induced subgraph of the corresponding clause gadget. We show that 14 edge deletions are necessary and sufficient to transform G_j into a bicolored-P_3-free graph. Obviously, for pairwise distinct $p, q, r \in \{1, 2, 3\}$, deleting the 14 edges in $E_{G_j}(\{a_j^p, a_j^q\}, B_j \cup W_j)$ transforms G_j into a bicolored-P_3-free graph, since $\{a_j^r\} \cup B_j \cup W_j$ is a clique in G_j. Hence, deleting 14 edges is sufficient. It remains to show that when deleting less than 14 edges there are still bicolored P_3s in G_j. To this end, we show that either one of the vertices in A_j is not incident with an edge deletion in G_j or we need more than 14 edge deletions to transform G_j into a bicolored-P_3-free graph. We consider three vertices $u_1, u_2, u_3 \in B_j \cup W_j$ representing the endpoints of deleted edges incident with a_j^1, a_j^2, and a_j^3, respectively. Let $S_j := \{\{a_j^p, u_p\} \mid p \in \{1, 2, 3\}\}$. The following claim gives a lower bound on the number of edge deletions in G_j after deleting S_j.

Claim 1. *There are at least* 12 *edge-disjoint bicolored P_3s in $G_j - S_j$.*

Proof. We define three sets $\mathcal{P}^1$, $\mathcal{P}^2$, and $\mathcal{P}^3$ containing bicolored P_3s and show that the union $\mathcal{P}^1 \cup \mathcal{P}^2 \cup \mathcal{P}^3$ contains at least 12 edge-disjoint bicolored P_3s in $G_j - S_j$. Here, we represent bicolored P_3s by edge sets of size two. For each $p \in \{1, 2, 3\}$ we set

$$\mathcal{P}^p := \{\{\{a_j^p, w\}, \{w, u_p\}\} \mid w \in (B_j \cup W_j) \setminus \{b_j^p, u_p\}\}.$$

Let $p \in \{1, 2, 3\}$. Since $E_{G_j}(W_j \cup B_j) \subseteq E_b$, and $E_{G_j}(\{a_j^p\}, (B_j \cup W_j) \setminus \{b_j^p\}) \subseteq E_r$, it follows that every set $P \in \mathcal{P}^p$ is a bicolored P_3 in $G_j - S_j$. Obviously, the bicolored P_3s in $\mathcal{P}^p$ are edge-disjoint and $|\mathcal{P}^p| = |(B_j \cup W_j) \setminus \{b_j^1, u_1\}| = 5$.

We now show that the union $\mathcal{P}^1 \cup \mathcal{P}^2 \cup \mathcal{P}^3$ contains at least 12 edge-disjoint bicolored P_3s in $G_j - S_j$. To this end, consider the following subset

$$\mathbb{P} := (\mathcal{P}^1 \cup \mathcal{P}^2 \cup \mathcal{P}^3) \setminus \{Q_1, Q_2, Q_3\} \subseteq \mathcal{P}^1 \cup \mathcal{P}^2 \cup \mathcal{P}^3,$$

where

$$\begin{aligned} Q_1 &:= \{\{a_j^2, u_1\}, \{u_1, u_2\}\}, \\ Q_2 &:= \{\{a_j^3, u_1\}, \{u_1, u_3\}\}, \\ Q_3 &:= \{\{a_j^3, u_2\}, \{u_2, u_3\}\}. \end{aligned}$$

Obviously, $Q_1, Q_2, Q_3 \notin \mathbb{P}$ and $|\mathbb{P}| \geq 3 \cdot 5 - 3 = 12$. It remains to show that all bicolored P_3s in $\mathbb{P}$ are edge-disjoint. Assume towards a contradiction that there are $P, R \in \mathbb{P}$ with $P \neq R$ and $P \cap R \neq \emptyset$. Since every $\mathcal{P}^p$ contains edge-disjoint bicolored P_3s, it follows that $P \in \mathcal{P}^p$ and $R \in \mathcal{P}^r$ for some $p \neq r$. Without loss of generality assume $p < r$. Since for every $w \in B_j \cup W_j$, the edges $\{a_j^r, w\}$ are not part of any bicolored P_3 in $\mathcal{P}^p$ and, conversely, the edges $\{a_j^p, w\}$ are not part of any bicolored P_3 in $\mathcal{P}^r$ it follows that $P \cap R = \{\{u_p, u_r\}\}$. We conclude $R = \{\{a_j^r, u_p\}, \{u_p, u_r\}\}$.

Consider the case $p = 1$ and $r = 2$. Then, $R = Q_1 \notin \mathbb{P}$. Analogously, if $p = 1$ and $r = 3$, then $R = Q_2 \notin \mathbb{P}$, and if $p = 2$ and $r = 3$, then $R = Q_3 \notin \mathbb{P}$. In every case we have $R \notin \mathbb{P}$ which contradicts the assumption $P, R \in \mathbb{P}$. Hence, there are no bicolored P_3s in $\mathbb{P}$ that share an edge and therefore $\mathcal{P}^1 \cup \mathcal{P}^2 \cup \mathcal{P}^3$ contains at least 12 edge-disjoint bicolored P_3s as claimed. $\diamond$

Claim 1 implies that if every vertex in A_j is incident with an edge in $S \cap E_{G_j}(A_j, B_j \cup W_j)$, we have $|S \cap E_{G_j}(A_j \cup B_j \cup W_j)| \geq 3 + 12 = 15$. We now show that deleting $E_{G_j}(\{a_j^p, a_j^q\}, B_j \cup W_j)$ for distinct $p, q \in \{1, 2, 3\}$ are the only three possible ways to transform G_j into a bicolored-P_3-free graph with less than 15 edge deletions. By Claim 1, we can assume without loss of generality that $E_{G_j}(\{a_j^3\}, W_j \cup B_j) \cap S = \emptyset$. We show that this implies that all edges incident with a_j^1 and a_j^2 in G_j are deleted by S.

Claim 2. *If $E_{G_j}(\{a_j^3\}, W_j \cup B_j) \cap S = \emptyset$, then $E_{G_j}(\{a_j^p\}, B_j \cup W_j) \subseteq S$ for $p = 1$ and for $p = 2$.*

Proof. First, note that no edge $\{b_j^3, w\}$ with $w \in B_j \cup W_j$ is an element of S, since otherwise $\{a_j^3, b_j^3\} \in E_b$ and $\{a_j^3, w\} \in E_r$ form a bicolored P_3 in $G_j - S$ which contradicts the fact that $G - S$ is bicolored-P_3-free. Next, consider $\{a_j^p, b_j^3\}$. Clearly, $\{a_j^p, b_j^3\}$ is an element of S, since otherwise $\{a_j^p, b_j^3\} \in E_r$ and $\{a_j^3, b_j^3\} \in E_b$ form a bicolored P_3 in $G_j - S$ which contradicts the fact that $G - S$ is bicolored-P_3-free.

It remains to show that $E_{G_j}(\{a_j^p\}, (B_j \setminus \{b_j^3\}) \cup W_j) \subseteq S$. Assume towards a contradiction that there exists an edge $\{a_j^p, w\}$ in $G_j - S$ with $w \in (B_j \cup W_j) \setminus \{b_j^3\}$. If $w = b_j^p$, the edges $\{a_j^p, b_j^p\} \in E_b$ and $\{b_j^p, a_j^3\} \in E_r$ form a bicolored P_3 in $G_j - S$. Otherwise, if $w \neq b_j^p$ the edges $\{a_j^p, w\} \in E_r$ and $\{w, b_j^3\} \in E_b$ form a bicolored P_3 in $G_j - S$. Both cases contradict the fact that $G - S$ is bicolored-P_3-free and therefore $E_{G_j - S}(\{a_j^p\}, B_j \cup W_j) = \emptyset$ as claimed. $\diamond$

We conclude from Claim 2 that deleting the 14 edges in $E_{G_j}(\{a_j^p, a_j^q\}, B_j \cup W_j)$ for distinct $p, q \in \{1, 2, 3\}$ are the only three possible ways to destroy all bicolored P_3s in G_j with at most 14 edge deletions.

This fact combined with the fact that we need at least 4 edge deletions per variable gadget and $|S| \leq 4 \cdot |X| + 14 \cdot |\mathcal{C}|$ implies that $|E_G(A_j \cup B_j \cup W_j) \cap S| = 14$ for each clause C_j and $|E_G(\{v_i\} \cup T_i \cup F_i) \cap S| = 4$ for each variable x_i.

We now define a satisfying assignment $\mathcal{A} : X \to \{\text{true}, \text{false}\}$ for ϕ by

$$\mathcal{A}(x_i) := \begin{cases} \text{true} & \text{if } E(\{v_i\}, T_i) \subseteq S, \text{ and} \\ \text{false} & \text{if } E(\{v_i\}, F_i) \subseteq S. \end{cases}$$

The assignment $\mathcal{A}$ is well-defined since in each variable gadget either all red or all blue edges belong to S.

Let $C_j \in \mathcal{C}$ be some clause in ϕ. It remains to show that C_j is satisfied by $\mathcal{A}$. Since $|E_G(A_j \cup B_j \cup W_j) \cap S| = 14$ there are distinct $p, q \in \{1, 2, 3\}$ such that $S \cap E_G(A_j \cup B_j \cup W_j) = E_G(\{a_j^p, a_j^q\}, B_j \cup W_j)$. Without loss of generality assume $p = 1$ and $q = 2$. Therefore $E_G(\{a_j^3\}, W_j \cup B_j) \cap S = \emptyset$. By the construction of G we know that there is a variable $x_i \in X$ occurring in C_j such that either $a_j^3 = t_i^{\Psi(C_j, x_i)}$ or $a_j^3 = f_i^{\Psi(C_j, x_i)}$. We show that clause C_j is satisfied by the assignment $\mathcal{A}(x_i)$.

For all $w \in W_j$, $G - S$ contains the red edge $\{a_j^3, w\}$. Moreover, $G - S$ contains the blue edge $\{a_j^3, b_j^3\}$. Since no vertex in $W_j \cup B_j$ is adjacent to v_i we conclude that $\{a_j^3, v_i\} \in S$ since otherwise $\{a_j^3, v_i\}$ is part of a bicolored P_3 in G' which contradicts the fact that G' is bicolored-P_3-free. If $a_j^3 = t_i^{\Psi(C_j, x_i)}$, then variable x_i occurs as a positive literal in C_j by the construction of G. Then, $\{a_j^3, v_i\} \in E(\{v_i\}, T_i)$. We conclude from $\{a_j^3, v_i\} \in S$ that $E(\{v_i\}, T_i) \subseteq S$ and therefore $\mathcal{A}(x_i) = \text{true}$. Analogously, if $a_j^3 = f_i^{\Psi(C_j, x_i)}$, then $\{a_j^3, v_i\} \in E(\{v_i\}, F_i)$. From $\{a_j^3, v_i\} \in S$ we conclude $\mathcal{A}(x_i) = \text{false}$. In both cases, the assignment $\mathcal{A}$ satisfies C_j which completes the correctness proof. □

Note that for any instance ϕ of (3,4)-SAT it holds that $|\mathcal{C}| \leq \frac{4}{3}|X|$. Thus, in the proof of Theorem 3.1 we constructed a graph with $8 \cdot |X| + 42 \cdot |\mathcal{C}| \in \mathcal{O}(|X|)$ edges, $k = 4 \cdot |X| + 14 \cdot |\mathcal{C}| \in \mathcal{O}(|X|)$, and therefore $\ell = 4 \cdot |X| + 28 \cdot |\mathcal{C}| \in \mathcal{O}(|X|)$ for the dual parameter $\ell = m - k$. Considering the ETH [20] and the fact that there is a reduction from 3-SAT to (3,4)-SAT with a linear blow-up in the number of variables [29] this implies the following.

Corollary 3.2. *If the ETH is true, then* BPD *cannot be solved in* $2^{o(n+m+k+\ell)}$ *time even if the maximum degree in* G *is* 8.

4 Polynomial-Time Solvable Cases

Since BPD is NP-hard, there is little hope to find a polynomial-time algorithm that solves BPD on arbitrary instances. In this section we provide polynomial-time algorithms for two special cases of BPD that are characterized by colored forbidden induced subgraphs.

4.1 BPD *on Bicolored* $\mathbf{K_3}$*-free Graphs*

Our first result is a polynomial-time algorithm for BPD, when $G = (V, E_r, E_b)$ does not contain a certain type of bicolored K_3s.

Definition 4.1. Three vertices u, v, w form a *bicolored* K_3 if $G[\{u, v, w\}]$ contains exactly three edges such that exactly two of them have the same color. A bicolored K_3 is *endangered in* G if at least one of the two edges with the same color is part of a bicolored P_3 in G.

A bicolored K_3 on vertices u, v, w can be seen as an induced subgraph of G, such that after one edge deletion in $E_G(\{u, v, w\})$ one might end up with a new bicolored P_3 containing the vertices u, v, and w. This happens, if we delete one of the two edges with the same color. If the bicolored K_3 is endangered, it might be necessary to delete one of these two edges to transform G into a bicolored-P_3-free graph. Intuitively, a graph G that contains no (endangered) bicolored K_3 can be seen as a graph from which we can delete any edge that is part of a bicolored P_3 without producing a new one. Note that the following result also implies that BPD can be solved in polynomial time on triangle-free graphs and thus also on bipartite graphs.

Theorem 4.2. BPD *can be solved in $\mathcal{O}(nm^{\frac{3}{2}})$ time if G contains no endangered bicolored K_3.*

Proof: We prove the theorem by reducing BPD to VERTEX COVER on bipartite graphs which can be solved in polynomial time since it is equivalent to computing a maximum matching.

> VERTEX COVER
> **Input:** A graph $G = (V, E)$ and an integer $k \in \mathbb{N}$.
> **Question:** Is there a *vertex cover* of size at most k in G, that is, a set $S \subseteq V$ with $|S| \leq k$ such that every edge $e \in E$ has at least one endpoint in S?

Let $(G = (V, E_b, E_r), k)$ be an instance of BPD where G contains no endangered bicolored K_3. We define an instance (G', k') of VERTEX COVER as follows. Let $G' = (V', E')$ be the graph with vertex set $V' := E_r \cup E_b$ and edge set $E' := \{\{e_1, e_2\} \subseteq E_b \cup E_r \mid e_1 \text{ and } e_2 \text{ form a bicolored } P_3 \text{ in } G\}$. That is, G' contains a vertex for each edge of G and edges are adjacent if they form a P_3 in G. Moreover, let $k' = k$. The graph G' is obviously bipartite with partite sets E_b and E_r.

We now show that (G, k) is a yes-instance for BPD if and only if (G', k') is a yes-instance for VERTEX COVER.

$(\Rightarrow)$ Let S be a solution for (G, k). Note that the edges of G are vertices of G' by construction and therefore $S \subseteq V'$. We show that S is a vertex cover in G'. Assume towards a contradiction that there is an edge $\{x, y\} \in E'$ with $x, y \notin S$. By the definition of E', the edges x and y form a bicolored P_3 in G. This contradicts the fact that $G - S$ is bicolored-P_3-free. Hence, S is a vertex cover of size at most k in G'.

$(\Leftarrow)$ Let $C \subseteq V'$ with $|C| \leq k$ be a minimal vertex cover of G'. Note that the vertices of G' are edges of G by construction and therefore $C \subseteq E$. We show that $G - C$ is bicolored-P_3-free. Assume towards a contradiction that there are edges $x = \{u, v\} \in E_b \setminus C$ and $y = \{v, w\} \in E_r \setminus C$ forming a bicolored P_3 in $G - C$. Then, x and y do not form a bicolored P_3 in G since otherwise there is an edge $\{x, y\} \in E'$, which has no endpoint in the vertex cover C. It follows that there is an edge $\{u, w\}$ in G that is not present in $G - C$. Consequently, $\{u, w\} \in C$. Obviously, the vertices u, v, w form a bicolored K_3. Since x and y form a bicolored P_3 in $G - C$, one of these edges has the same color as $\{u, w\}$. Since $\{u, w\} \in C$ and C is minimal, it follows that $\{u, w\} \in V'$ is an endpoint of an edge in G' and thus $\{u, w\}$ is part of a bicolored P_3 in G. Therefore, $G[\{u, v, w\}]$ forms an endangered bicolored K_3 in G which contradicts the fact that G contains no endangered bicolored K_3. This proves the correctness of the reduction.

For a given instance (G, k) of BPD, the VERTEX COVER instance (G', k') can be computed in $\mathcal{O}(nm)$ time by computing all bicolored P_3s of G. Since VERTEX COVER can be solved in $\mathcal{O}(|E'| \cdot \sqrt{|V'|})$ time on bipartite graphs [19] and since $|V'| = m$ and $|E'| \leq nm$, we conclude that BPD can be solved in $\mathcal{O}(nm^{\frac{3}{2}})$ time on graphs without endangered K_3s. □

4.2 BPD on Graphs without Monochromatic K_3s and P_3s

We now show a second polynomial-time solvable special case that is characterized by four colored forbidden induced subgraphs: the two *monochromatic* K_3s, these are the K_3s where all three edges have the same color, and the two *monochromatic* P_3s, these are the P_3s where both edges have the same color. Observe that a graph that does not contain these forbidden induced subgraphs may still contain K_3s or P_3s.

We provide two reduction rules that lead to a polynomial-time algorithm for this special case. These rules can also be applied to general instances of BPD and thus their running time bound is given for general graphs. We will later show that on graphs without monochromatic K_3s and P_3s we can apply them exhaustively in $\mathcal{O}(n)$ time.

Reduction Rule 1. a) Remove all bicolored-P_3-free components from G.

b) If G contains a connected component C of size at most five, then compute the minimum number of edge deletions k_C to make $G[C]$ bicolored-P_3-free, remove C from G, and set $k \leftarrow k - k_C$.

Lemma 4.3. *Reduction Rule 1 is correct and can be exhaustively applied in $\mathcal{O}(nm)$ time.*

Proof: The correctness of Reduction Rule 1 is obvious. Part a) can be exhaustively applied in $\mathcal{O}(nm)$ time as follows: First compute the connected components of G in $\mathcal{O}(n + m)$ via breadth-first search. Then, enumerate all bicolored P_3s and label the vertices that are not part of any bicolored P_3 in $\mathcal{O}(nm)$ time. Finally, remove all connected components without labeled vertices in $\mathcal{O}(n + m)$ time. Part b) of Reduction Rule 1 can be applied exhaustively in $\mathcal{O}(n)$ time since we only need to find and remove connected components of constant size. □

The second reduction rule involves certain bridges that may be deleted greedily. An edge e is a *bridge* if the graph $G - e$ has more connected components than G.

Reduction Rule 2. a) Remove all bridges from G that are not contained in any bicolored P_3.

b) If G contains a bridge $\{u, v\}$ such that

- the connected component C containing v in $G - \{u, v\}$ is bicolored-P_3-free and
- $\{u, v\}$ forms a bicolored P_3 with some edge $\{v, w\}$ of C in G,

then remove C from G and set $k \leftarrow k - 1$.

Lemma 4.4. *Reduction Rule 2 is correct and can be applied exhaustively in $\mathcal{O}(nm)$ time.*

Proof: Let (G, k) be the original instance and (G', k') be the instance after the application of the rule. We first show the correctness of the two parts of the rule. For part a), observe that for every subgraph G^* of G, any bridge $\{u, v\}$ that is removed by the rule is not part of a bicolored P_3. Thus, any solution for G is a solution for $G - \{u, v\}$ and vice versa. For part b), observe first that if (G', k') has a solution S', then $S' \cup \{u, v\}$ is a solution for (G, k) and thus (G, k) is also a yes-instance. It remains to show that if (G, k) is a yes-instance, then so is (G', k'). Consider a solution S with $|S| \leq k$ for G. Observe that $\{u, v\} \in S$ or $\{v, w\} \in S$. This implies $S' := S \cap E(G[V \setminus C]) \leq k - 1$. Finally, observe that since $G[V \setminus C] = G'$, S' is a solution of size at most $k - 1$ for G'.

The running time can be seen as follows: We compute in $\mathcal{O}(n + m)$ time the bridges of G [28]. Given the bridges, one can compute in $\mathcal{O}(n + m)$ the block-cut-forest F of G. The vertices of F are maximal 2-edge-connected components of G and the edges correspond to the bridges of G. Then in $\mathcal{O}(nm)$ time, we enumerate all bicolored P_3s. Using the set of bicolored P_3s, we can compute for each 2-edge-connected component whether it contains a bicolored P_3. Moreover, for each bridge e and each incident 2-edge-connected component C, we can compute whether e forms a bicolored P_3 with some edge of C. Finally, we can compute for each bridge the set of edges with which it forms a conflict. This additional information can be computed in $\mathcal{O}(nm)$ time. We incorporate this information into the block-cut-forest F as follows: A vertex of F is colored black if the corresponding 2-edge-connected component contains a bicolored P_3, otherwise it is colored white.

The exhaustive application of the rule is now performed on the block-cut-forest F via the following algorithm. First, remove all white singletons from G and F. Then, remove all bridges of G from G and F that are not part of in any bicolored P_3 in G. Checking whether one of these two conditions is fulfilled can be performed in $\mathcal{O}(n + m)$ time per removed edge and vertex. In the following, we assume that these removals have been applied exhaustively.

To describe the final part of the reduction, we denote for each vertex v of G the 2-edge-connected component of G containing v by $[v]$. Note that $[v]$ is a vertex of F. Now we check for each vertex v of G whether it is incident with a bridge $\{u, v\}$ that fulfills the condition of part b) of the rule.

To do this efficiently, we characterize such bridges as follows.

Claim 3. *A bridge $\{u, v\}$ and a vertex v fulfill the condition of Reduction Rule 2 b) if and only if*

- $\{u, v\}$ *forms a bicolored P_3 with some edge of $[v]$ or some bridge that is incident with v,*
- $[v]$ *is white,*
- $N_F([v]) \setminus \{[u]\}$ *contains only white leaf-vertices, and*
- *all other bridges incident with some vertex of $[v]$ form a bicolored P_3 with $\{u, v\}$ and are not part of any further bicolored P_3.*

Proof. Let C denote the connected component containing v after the deletion of $\{u, v\}$. Clearly, the stated conditions are sufficient for $\{u, v\}$ and v to fulfill the requirements of the rule. For most conditions it is also clear that they are necessary for C to be bicolored P_3-free. The only non-obvious condition is that $N_F([v])\setminus\{[u]\}$ contains only leaf-vertices. This condition is necessary since, otherwise, C would contain some bridge e that is not incident with a vertex of $[v]$. This bridge is part of some bicolored P_3, since it has not been removed previously. This bicolored P_3 would also be contained in C since e and $\{u, v\}$ are not incident. $\diamond$

This characterization gives us now a way to check in $\mathcal{O}(n + m)$ time whether G contains some bridge that fulfills the condition of Reduction Rule 2 b). This check is done as follows. We consider each vertex x of F. At most two bridges incident with some vertex of x are candidates for fulfilling the conditions of the claim: all incident bridges must be incident with the same vertex v of x for the conditions to be fulfilled and an edge $\{u, v\}$ must be the only incident bridge of its color if the conditions are fulfilled. For each candidate edge, we check in $\mathcal{O}(|N_F([v])|)$ time whether the conditions of the claim are fulfilled using the precomputed information about the conflicts in G. Thus, the total running time for this final check is linear in the number of edges of F and thus in $\mathcal{O}(n + m)$. Note that after removing some edge $\{u, v\}$ in this way, the remaining vertices of C are removed via the previous two checks.

Altogether, applying the rules using F needs $\mathcal{O}(n + m)$ time per removed vertex and edge. Since each application removes some vertex or bridge of G, there are in total $\mathcal{O}(n)$ applications. Moreover, after removing all isolated vertices from G, we have $n \leq 2m$ and thus the overall running time of $\mathcal{O}(nm)$ for the application of the rule follows. □

As we will show later, any graph without monochromatic P_3s and monochromatic K_3s to which the above reduction rules do not apply has maximum degree two. These graphs can be solved in linear time as we see in the following lemma.

Lemma 4.5. *Let (G, k) be an instance of* BPD *such that G has maximum degree 2. Then, (G, k) can be solved in $\mathcal{O}(n)$ time.*

Proof: In the following, we construct a solution S for (G, k). If G has maximum degree 2, then each connected component of G is either a path or a cycle. The algorithm first deals with cycles and then considers the remaining paths. Observe that every connected component of size at most 3 can be solved within $\mathcal{O}(1)$ time. For the rest of the proof we assume that every connected component has size at least 4.

First, we consider each connected component C of G which is a cycle. We either transform C into one or two paths or solve C directly. First, assume that C contains three subsequent edges e_1, e_2, and e_3 of the same color, then edge e_2 is not part of any bicolored P_3. Hence e_2 can be removed without decreasing k. The remaining connected component is a path and will be solved in the second step. Second, assume that C contains two subsequent edges e_1 and e_2 with the same color. Recall that we may assume that $|C| \geq 4$. Further, let e_0 be the other edge that is incident with e_1 and let e_3 be the other edge that is incident with e_2. According to our assumption, e_0 and e_1 form a bicolored P_3 and e_2 and e_3 form a bicolored P_3. Hence, either $e_0 \in S$ or $e_1 \in S$ and either $e_2 \in S$ or $e_3 \in S$. Since e_1 and e_2 have the same color and no further edges are incident with e_1 and e_2, we may assume that $e_0, e_3 \in S$. The remaining connected components are paths and will be solved in the second step. Third, consider the case that C contains no two subsequent edges of the same color. Then C consists of 2ℓ edges $e_1, \ldots, e_{2\ell}$ and each two subsequent edges form a bicolored P_3. Thus, C contains a set of ℓ edge-disjoint bicolored P_3s: $\{e_1, e_2\}, \{e_3, e_4\}, \ldots, \{e_{2\ell-1}, e_{2\ell}\}$ and contains exactly ℓ blue edges. Thus, deleting the ℓ blue edges of C is optimal.

In a second step, we consider each connected component that is a path. Let P_n be a path consisting of n vertices $v_1, \ldots, v_n$. Visit the edges $\{v_i, v_{i+1}\}$ for increasing i starting at v_1. For each edge, check whether it is part of some bicolored P_3. Let $\{v_i, v_{i+1}\}$ be the first encountered edge that is in a bicolored P_3. Then, delete $\{v_{i+1}, v_{i+2}\}$, decrease k by one, and continue with $\{v_{i+2}, v_{i+3}\}$ if it exists. First, observe that $\{v_{i+1}, v_{i+2}\}$ exists since $\{v_i, v_{i+1}\}$ does not form a bicolored P_3 with $\{v_{i-1}, v_i\}$. Second, observe that the deletion of $\{v_{i+1}, v_{i+2}\}$ is simply an application of Reduction Rule 2 and therefore correct. Clearly, this greedy algorithm runs in $\mathcal{O}(n)$ time.

In altogether $\mathcal{O}(n)$ time, we can consider each cycle C and either solve C or delete one or two edges, which transforms C into one or two paths. The greedy algorithm for paths runs in $\mathcal{O}(n)$ time on all paths. Thus, the remaining instance can be solved in $\mathcal{O}(n)$ time. The overall running time follows. □

We have now all ingredients to present the polynomial-time algorithm for graphs without monochromatic K_3 and monochromatic P_3. In order to prove the correctness of the algorithm and the linear running time, we make the following observation about such graphs.

Lemma 4.6. *Let G be a graph that contains no monochromatic K_3 and no monochromatic P_3 as induced subgraphs. Then, the maximum blue degree and the maximum red degree in G are 2.*

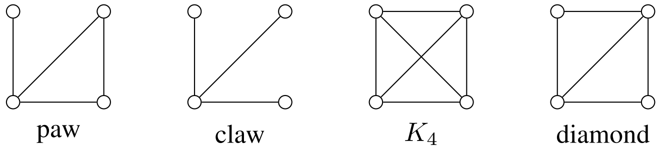

Figure 2: The (uncolored) small graphs used in this work.

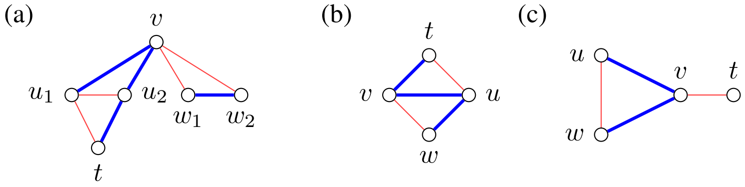

Figure 3: Subgraphs used in the proof of Theorem 4.7 (a) $|N(v)| = 4$, (b) $G[N[v]]$ is a diamond and (c) $G[N[v]]$ is a paw.

Proof: We show the proof only for the blue degree, the bound for the red degree can be shown symmetrically. Assume towards a contradiction that G contains a vertex t with at least three blue neighbors u, v, and w. Since G contains no blue P_3, the subgraph $G[\{u, v, w\}]$ has three edges. Moreover, since G contains no monochromatic K_3 not all of the three edges in $G[\{u, v, w\}]$ are red. Assume without loss of generality that $\{u, v\}$ is blue. Then $G[\{u, v, t\}]$ is a blue K_3, a contradiction. □

Theorem 4.7. BPD *can be solved in $\mathcal{O}(n)$ time if G contains no monochromatic K_3 and no monochromatic P_3.*

Proof: The algorithm first applies Reduction Rule 1 exhaustively. Afterwards, Reduction Rule 2 is applied on all bridges $\{u, v\}$ with $\deg(u) \geq 3$ or $\deg(v) \geq 3$. Thus, let G be the graph after the applications of Reduction Rules 1 and 2 as described above. We show that G has maximum degree at most 2. Afterwards, Lemma 4.5 applies and the remaining instance can be solved in $\mathcal{O}(n)$ time. Observe that by Lemma 4.6, the maximum degree in G is 4. For an illustration of the small uncolored graphs used in this proof see Figure 2.

First, assume that the maximum degree of G is 4 and let v be a vertex of degree 4. We show that $N[v]$ is a connected component of G. This implies that $N[v]$ is removed by Reduction Rule 1 in this case. By Lemma 4.6, the vertex v has exactly two blue neighbors u_1, u_2 and exactly two red neighbors w_1, w_2. Since G contains no monochromatic P_3 and no monochromatic K_3, $\{u_1, u_2\}$ is red and $\{w_1, w_2\}$ is blue. Now assume towards a contradiction, that one of these four vertices has a neighbor $t \notin N[v]$ in G.

Without loss of generality assume that this vertex is u_1. See Figure 3 (a) for an example. Then, $\{u_1, t\}$ is red because otherwise $G[\{u_1, v, t\}]$ is a blue P_3. This implies that $\{u_2, t\}$ is blue because otherwise $G[\{u_1, u_2, t\}]$ is a red P_3 or a red K_3. Then, however, $G[\{u_2, v, t\}]$ is a blue P_3, a contradiction. Altogether, this implies that $N[v]$ is a connected component of G. Hence, if G is reduced with respect to Reduction Rule 1, then G contains no vertices of degree 4.

Second, assume that the maximum degree of G is 3 and let v be a vertex of degree 3. If $G[N[v]]$ is a K_4, then $N[v]$ is a connected component of G and is removed by Reduction Rule 1. Next, assume $G[N[v]]$ is a diamond, and let t, u, and w denote the neighbors of v where u is the other vertex that has degree three in $G[N[v]]$. See Figure 3 (b) for an example. Assume without loss of generality that u is a blue neighbor of v. Then, by Lemma 4.6, one of t and w, say w, is a red neighbor of v. This implies that t is a blue neighbor of v, because otherwise $G[\{v, w, t\}]$ is a red P_3. Consequently, t is a red neighbor of u because otherwise t, v, and u form a blue K_3. Hence, w is a blue neighbor of u because otherwise $G[\{u, w, t\}]$ is a red P_3. Altogether, we have that t and w each have a red and a blue neighbor in $\{u, v\}$. Since $N[v] = N[u]$, we have that t and w cannot have a neighbor in $V \setminus N[v]$ as such a neighbor would form a monochromatic P_3 with one of u and v. Hence, $N[v]$ is a connected component of G in this case and consequently removed by Reduction Rule 1.

Thus $G[N[v]]$ is neither a K_4 nor a diamond if G is reduced with respect to Reduction Rule 1. Moreover, $G[N[v]]$ cannot be a claw since in this case G contains a monochromatic P_3. Hence, the only remaining case is that $G[N[v]]$ is a paw. Let t, u, and w be the neighbors of v where u and w are adjacent. For an example see Figure 3 (c). Assume furthermore without loss of generality that v is incident with two blue edges. This implies that t is a red neighbor of v as otherwise t, v, and u form a monochromatic P_3. Also, u and w are blue neighbors of v. Consequently, $\{u, w\}$ is red. As in the proof above for the case that $G[N[v]]$ is a diamond, u and w have no further neighbor in G. Thus, $\{v, t\}$ is a bridge with $\deg(v) = 3$ that fulfills the condition of Reduction Rule 2 b) and thus $\{v, t\}$ is removed by this rule. Altogether this implies that any instance to which Reduction Rules 1 and 2 have been applied as described above has maximum degree 2. By Lemma 4.5, we can thus solve the remaining instance in linear time.

Next, we consider the running times of Reduction Rules 1 and 2 in more detail since for both rules the running time analysis given above did *not* assume that G contains no monochromatic P_3 and no monochromatic K_3.

First, we apply Reduction Rule 1 exhaustively. Since G has maximum degree at most four, we can label all vertices that are part of some bicolored P_3 in $\mathcal{O}(n)$ time and thus Reduction Rule 1 can be applied exhaustively in $\mathcal{O}(n)$ time. Observe that in the resulting graph the maximum degree of G is three since vertices of degree four are in connected components of size five.

Next, we consider the running time of Reduction Rule 2, after Reduction Rule 1 was applied exhaustively. Recall that Reduction Rule 2 is only applied to bridges that have at least one endpoint with degree three. To apply the rule exhaustively, we first compute in $\mathcal{O}(n)$ time the set of all vertices of degree three. For each such vertex v, the graph $G[N[v]]$ is a paw because otherwise, $N[v]$ is a connected component of constant size as shown above. Hence, there exist two vertices $u, w \in N(v)$ such that $N(u) = \{v, w\}$ and $N(w) = \{u, v\}$. The vertices u and w can be determined in $\mathcal{O}(1)$ time. Let $t \in N(v) \setminus \{u, w\}$. Then, Reduction Rule 2 removes $\{v, t\}$ from G. Thus, in $\mathcal{O}(1)$ time, we may apply Reduction Rule 2 on the bridge containing v. Consequently, the rule can be applied exhaustively on all degree-three vertices $\mathcal{O}(n)$ time. Afterwards, the remaining instance has maximum degree two and can be solved in $\mathcal{O}(n)$ time. Hence, the overall running time is $\mathcal{O}(n)$. □

5 Parameterized Complexity

In this section we study the parameterized complexity of BPD parameterized by k, $\ell := m-k$, and (k, Δ), where Δ denotes the maximum degree of G. We first provide an $\mathcal{O}(1.84^k \cdot nm)$-time fixed-parameter algorithm for BPD. Afterwards, we study problem kernelizations for BPD parameterized by (k, Δ) and ℓ.

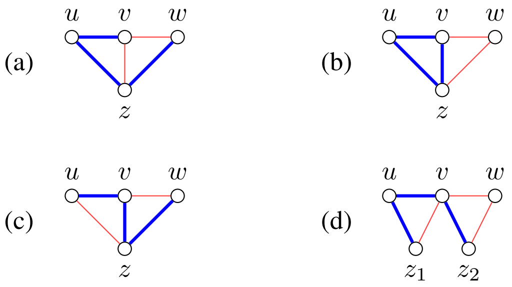

Figure 4: The graphs used in the proof of Theorem 5.6 (a) LC-Diamond, (b) LO-Diamond, (c) IIZ-Diamond and (d) CC-Hourglass. The names refer to the shapes of the connected components of both colors. Due to symmetry we use the same names if we swap the red and blue colors of the edges in each graph.

5.1 A Fixed-Parameter Algorithm for Bicolored $\mathbf{P_3}$ Deletion

We now provide a fixed-parameter algorithm that solves BPD parameterized by k. Note that there is a naive $\mathcal{O}(2^k \cdot nm)$ branching algorithm for BPD: For a given instance (G, k), check in $\mathcal{O}(nm)$ time if G contains a bicolored P_3. If this is not the case, then answer yes. Otherwise, answer no if $k < 1$. If $k \geq 1$, then compute a bicolored P_3 formed by the edges e_1 and e_2 and branch into the cases $(G - e_1, k - 1)$ and $(G - e_2, k - 1)$. We modify this simple algorithm by branching on slightly more complex structures, obtaining a running time of $\mathcal{O}(1.84^k \cdot nm)$. Note that by Corollary 3.2 a subexponential algorithm in k is not possible when assuming the ETH.

The basic idea of the algorithm is to branch on LC-Diamonds, LO-Diamonds, IIZ-Diamonds and CC-Hourglasses. For the definition of these structures see Figure 4. We say that a graph G is *nice* if G has none of the structures from Figure 4 as induced subgraph and every edge of G forms a bicolored P_3 with at most one other edge of G. We give a polynomial-time algorithm that solves BPD when the input graph is nice. To this end consider the following proposition.

Proposition 5.1. *Let $(G = (V, E_r, E_b), k)$ be an instance of* BPD *such that G is nice. Moreover, let p be the number of bicolored P_3s in G. Then, for every two edges e_1 and e_2 forming a bicolored P_3 in G there is an edge $e \in \{e_1, e_2\}$ such that*

a) $G - e$ contains $p - 1$ bicolored P_3s and every bicolored P_3 of $G - e$ is a bicolored P_3 in G, and

b) $G - e$ is nice.

Proof: For the proof of Statement *a)*, let u and v denote the endpoints of $e_1 \in E_b$ and let v and w denote the endpoints of $e_2 \in E_r$. Note that the number of bicolored P_3s in $G - e_1$ and $G - e_2$ is at least $p - 1$ since every edge of G is part of at most one bicolored P_3 since G is nice. It remains to show that there is an edge $e \in \{e_1, e_2\}$ such that the number of bicolored P_3s in $G - e$ is at most $p - 1$, and that every bicolored P_3 in $G - e$ is a bicolored P_3 in G. Assume towards a contradiction that there are at least p bicolored P_3s in $G - e_1$ and in $G - e_2$. Then, there exist vertices $z_1, z_2 \in V$, such that $\{u, z_1\}$ and $\{z_1, v\}$ form a bicolored P_3 in $G - e_1$ and also $\{v, z_2\}$ and $\{z_2, w\}$ form a bicolored P_3 in $G - e_2$.

First, assume $z_1 = z_2 =: z$. If $\{v, z\} \in E_r$, then $\{u, z\} \in E_b$, $\{w, z\} \in E_b$, and $G[\{u, v, w, z\}]$ is an LC-Diamond with two red and three blue edges. This contradicts the fact that G contains no induced

LC-Diamond. Analogously, if $\{v, z\} \in E_b$, then $G[\{u, v, w, z\}]$ is an LC-Diamond with two blue and three red edges. We conclude $z_1 \neq z_2$.

Second, assume $\{v, z_1\} \in E_b$. Then, $\{u, z_1\} \in E_r$. Since every edge of G is part of at most one bicolored P_3, there is an edge $\{z_1, w\} \in E$. If $\{z_1, w\} \in E_r$, then $G[\{u, v, w, z_1\}]$ is an LC-Diamond. Otherwise, if $\{z_1, w\} \in E_b$, then $G[\{u, v, w, z_1\}]$ is an IIZ-Diamond. This contradicts the fact that G contains no induced LC- or IIZ-Diamond. Therefore, $\{v, z_1\} \in E_r$. With the same arguments we can show that $\{v, z_2\} \in E_b$.

From $\{v, z_1\} \in E_r$ and $\{v, z_2\} \in E_b$ we conclude that $\{u, z_1\} \in E_b$ and $\{w, z_2\} \in E_r$. Next, if we have $E(\{u, v, w, z_1, z_2\}) = \{\{u, v\}, \{u, z_1\}, \{v, w\}, \{v, z_1\}, \{v, z_2\}, \{w, z_2\}\} =: A$, then the graph $G[\{u, v, w, z_1, z_2\}]$ is an induced CC-Hourglass which contradicts the fact that G does not contain induced CC-Hourglasses. Hence, $E(\{u, v, w, z_1, z_2\}) \supsetneq A$. Consider the following cases.

Case 1: $\{u, w\} \in E$. Then $\{u, v\}$ and $\{v, w\}$ do not form a bicolored P_3, which contradicts the choice of $\{u, v\}$ and $\{v, w\}$.

Case 2: $\{z_1, w\} \in E$. If $\{z_1, w\} \in E_b$, then $G[\{u, v, w, z_1\}]$ is an induced LC-Diamond. Otherwise, if $\{z_1, w\} \in E_r$, then $G[\{u, v, w, z_1\}]$ is an induced LO-Diamond. Both cases contradict the fact that G is nice.

Case 3: $\{z_2, u\} \in E$. If $\{z_2, u\} \in E_r$, then $G[\{u, v, w, z_2\}]$ is an induced LC-Diamond. Otherwise, if $\{z_1, w\} \in E_b$, then $G[\{u, v, w, z_2\}]$ is an induced LO-Diamond. Both cases contradict the fact that G is nice.

Case 4: $\{z_1, z_2\} \in E$. If $\{z_1, z_2\} \in E_b$, then $G[\{z_1, z_2, w, v\}]$ is an LC-Diamond. Otherwise, if $\{z_1, z_2\} \in E_r$, then $G[\{z_1, z_2, u, v\}]$ is an LC-Diamond. Both cases contradict the fact that G is nice.

All cases lead to a contradiction. Hence, there exists $e \in \{e_1, e_2\}$ such that $G - e$ contains $p - 1$ bicolored P_3s which proves Statement *a)*.

Next, we show Statement *b)*. To this end, let e_1 and e_2 be two edges forming a bicolored P_3 in G. Let $e \in \{e_1, e_2\}$ that satisfies *a)*. We show that $G - e$ is nice. From *a)* we know that every bicolored P_3 of $G - e$ is also a bicolored P_3 in G. Hence, the fact that every edge of G is part of at most one bicolored P_3 implies that every edge of $G - e$ is part of at most one bicolored P_3.

First, assume towards a contradiction that $G - e$ contains an induced LC-, LO- or IIZ-Diamond $(G - e)[\{u, v, w, z\}]$ as given in Figure 4. Since G contains no such structure, we conclude $e = \{u, w\}$ and $\{u, v, w, z\}$ is a clique in G. Then, deleting e from G produces a new bicolored P_3 on edges $\{u, v\}$ and $\{v, w\}$ in $G - e$ which contradicts Statement *a)*. Therefore, $G - e$ contains no induced LC-, LO- and IIZ-Diamonds.

Second, assume towards a contradiction that the graph $G - e$ contains some induced CC-Hourglass $(G - e)[\{u, v, w, z_1, z_2\}]$ as given in Figure 4. Then, since G does not contain an induced CC-Hourglass, both endpoints of e are elements of $\{u, v, w, z_1, z_2\}$ and $G[\{u, v, w, z_1, z_2\}]$ contains exactly seven edges.

Case 1: $e = \{u, w\}$ (or $e = \{z_1, z_2\}$). Then, $G - e$ contains the new bicolored P_3 formed by the edges $\{u, v\}$ and $\{u, w\}$ (by $\{z_1, v\}$ and $\{v, z_2\}$, respectively) which contradicts *a)*.

Case 2: $e = \{z_1, w\}$. Then, $G[\{u, v, w, z_1\}]$ is an induced LC-Diamond in G if $e \in E_b$ or an induced LO-Diamond in G if $e \in E_r$ which contradicts the fact that G has no induced LC-Diamonds and LO-Diamonds.

Case 3: $e = \{z_2, u\}$ Then, $G[\{u, v, w, z_2\}]$ is an induced LO-Diamond in G if $e \in E_b$ or an induced LC-Diamond in G if $e \in E_r$ which contradicts the fact that G has no induced LC-Diamonds and LO-Diamonds.

All cases lead to a contradiction and therefore $G - e$ contains no induced LC-, LO-, IIZ-Diamonds, CC-Hourglasses and every edge of $G - e$ is part of at most one bicolored P_3. □

Proposition 5.1 implies a simple algorithm for BPD on such graphs.

Corollary 5.2. *Let (G, k) be an instance of* BPD *where G is nice. Then, we can decide in $\mathcal{O}(nm)$ time whether (G, k) is a yes- or a no-instance of* BPD.

Proof: We solve BPD with the following algorithm: First, enumerate all bicolored P_3s in $\mathcal{O}(nm)$ time. Second, check if there are at most k bicolored P_3s. If yes, (G, k) is a yes-instance. Otherwise, (G, k) is a no-instance.

It remains to show that this algorithm is correct. Assume G contains p bicolored P_3s. Since every edge of G forms a bicolored P_3 with at most one other edge, all bicolored P_3s in G are edge-disjoint. Hence, p edge deletions are necessary. By Proposition 5.1 *a)* we can eliminate exactly one bicolored P_3 with one edge deletion without producing other bicolored P_3s. By Proposition 5.1 *b)* this can be done successively with every bicolored P_3, since after deleting one of its edges we do not produce LC-, LO-, IIZ-Diamonds, CC-Hourglasses or edges that form a bicolored P_3 with more than one other edge. Thus, p edge deletions are sufficient. Hence, the algorithm is correct. □

Next, we describe how to transform an arbitrary graph G into a nice graph G' by branching. To this end consider the following branching rules applied on an instance (G, k) of BPD.

Branching Rule 1. If there are three distinct edges $e_1, e_2, e_3 \in E_r \cup E_b$ such that e_1 forms a bicolored P_3 with e_2 and with e_3, then branch into the cases

- $I_1 := (G - e_1, k - 1)$, and
- $I_2 := (G - \{e_2, e_3\}, k - 2)$.

Lemma 5.3. *Branching Rule 1 is correct.*

Proof: We show that (G, k) is a yes-instance of BPD if and only if at least one of the instances I_1 or I_2 is a yes-instance of BPD.

($\Leftarrow$) Assume I_1 is a yes-instance or I_2 is a yes-instance. In each branching case I_i, the parameter k is decreased by the exact amount p_i of edges deleted from G. Therefore, if some I_i has a solution of size at most $k - p_i$, then (G, k) has a solution.

($\Rightarrow$) Let S be a solution for G. Since e_1 and e_2 form a bicolored P_3, at least one of these edges belongs to S. If $e_1 \in S$, then I_1 is a yes-instance since we can transform $G - e_1$ into a bicolored-P_3-free graph by deleting the at most $k - 1$ edges in $S \setminus \{e_1\}$. Otherwise, if $e_1 \notin S$, then $e_2, e_3 \in S$. Hence, I_2 is a yes-instance since we can transform $G - \{e_2, e_3\}$ into a bicolored-P_3-free graph by deleting the at most $k - 2$ edges in $S \setminus \{e_2, e_3\}$. □

Branching Rule 2. If there are vertices $u, v, w, z \in V$ such that $G[\{u, v, w, z\}]$ is an LC-Diamond (Figure 4 (a)) or an LO-Diamond (Figure 4 (b)) or an IIZ-Diamond (Figure 4 (c)), then branch into the cases

- $I_1 := (G - \{v, w\}, k - 1)$,
- $I_2 := (G - \{\{u, v\}, \{u, z\}\}, k - 2)$, and

- $I_3 := (G - \{\{u,v\},\{v,z\},\{w,z\}\}, k-3)$.

Lemma 5.4. *Branching Rule 2 is correct.*

Proof: We show that (G,k) is a yes-instance of BPD if and only if at least one of the instances I_1, I_2, or I_3 is a yes-instance of BPD.

$(\Leftarrow)$ This direction holds since in every instance I_i the parameter k is decreased by the exact amount of edges deleted from G.

$(\Rightarrow)$ Let S be a solution for G. In LO-Diamonds, LC-Diamonds, and in IIZ-Diamonds, the edges $\{u,v\}$ and $\{v,w\}$ form a bicolored P_3 in G and therefore $\{u,v\} \in S$ or $\{v,w\} \in S$. If $\{v,w\} \in S$, then I_1 is a yes-instance. Otherwise, if $\{v,w\} \notin S$ it follows that $\{u,v\} \in S$. If $\{u,z\} \in S$, we have $\{u,v\},\{u,z\} \in S$ and therefore I_2 is a yes-instance. So we assume $\{v,w\} \notin S$, $\{u,v\} \in S$, $\{u,z\} \notin S$ and consider the following cases.

Case 1: $G[\{u,v,w,z\}]$ is an LC-Diamond. Then, $\{u,z\}$ and $\{v,z\}$ form a bicolored P_3 in $G - \{u,v\}$. Since $\{u,z\} \notin S$, it follows that $\{v,z\} \in S$. The edges $\{v,w\}$ and $\{w,z\}$ form a bicolored P_3 in $G - \{\{u,v\},\{v,z\}\}$ which implies $\{v,w\} \in S$ or $\{w,z\} \in S$. Since $\{v,w\} \notin S$, we have $\{w,z\} \in S$. Thus, $\{u,v\},\{v,z\},\{w,z\} \in S$ and therefore I_3 is a yes-instance.

Case 2: $G[\{u,v,w,z\}]$ is an LO-Diamond. Then, $\{u,z\}$ and $\{w,z\}$ form a bicolored P_3 in $G - \{u,v\}$. Since $\{u,z\} \notin S$, it follows that $\{w,z\} \in S$. The edges $\{v,w\}$ and $\{v,z\}$ form a bicolored P_3 in $G - \{\{u,v\},\{w,z\}\}$ which implies $\{v,w\} \in S$ or $\{v,z\} \in S$. Since $\{v,w\} \notin S$, we have $\{v,z\} \in S$. Thus, $\{u,v\},\{v,z\},\{w,z\} \in S$ and therefore I_3 is a yes-instance.

Case 3: $G[\{u,v,w,z\}]$ is an IIZ-Diamond. Then, $\{u,z\}$ forms a bicolored P_3 with $\{w,z\}$ and with $\{v,z\}$ in $G - \{u,v\}$. Since $\{u,z\} \notin S$, it follows that $\{w,z\},\{v,z\} \in S$ and therefore I_3 is a yes-instance. □

Branching Rule 3. If there are vertices $u,v,w,z_1,z_2 \in V$ such that $G[\{u,v,w,z_1,z_2\}]$ is a CC-Hourglass as given in Figure 4 (d), then branch into the cases

- $I_1 := (G - \{v,w\}, k-1)$,
- $I_2 := (G - \{\{u,v\},\{v,z_1\}\}, k-2)$, and
- $I_3 := (G - \{\{u,v\},\{u,z_1\},\{v,z_2\}\}, k-3)$.

Lemma 5.5. *Branching Rule 3 is correct.*

Proof: We show that (G,k) is a yes-instance of BPD if and only if at least one of the instances I_1, I_2, or I_3 is a yes-instance of BPD.

$(\Leftarrow)$ This direction obviously holds since in every instance I_i the parameter k is decreased by the exact amount of edges deleted from G.

$(\Rightarrow)$ Let S be a solution for G. The edges $\{u,v\}$ and $\{v,w\}$ form a bicolored P_3 in G and therefore $\{u,v\} \in S$ or $\{v,w\} \in S$. If $\{v,w\} \in S$, then I_1 is a yes-instance. Otherwise, if $\{v,w\} \notin S$, then $\{u,v\} \in S$. The edge $\{v,z_1\}$ forms a bicolored P_3 with $\{u,z_1\}$ and $\{v,z_2\}$ in $G - \{u,v\}$. If $\{v,z_1\} \in S$, then I_2 is a yes-instance. Otherwise, if $\{v,z_1\} \notin S$, then $\{u,z_1\},\{v,z_2\} \in S$. Hence, I_3 is a yes-instance. □

We use the Branching Rules 1–3 to state the following theorem.

Theorem 5.6. BPD *can be solved in $\mathcal{O}(1.84^k \cdot nm)$ time.*

Proof: We solve BPD for an instance (G, k) as follows: Initially, we compute the adjacency matrix of G in $\mathcal{O}(n^2)$ time. We then compute one of the structures we branch on, which is an induced LC-Diamond, LO-Diamond, IIZ-Diamond, CC-Hourglass or some edge which forms a bicolored P_3 with two other edges. Next, we branch according to the Branching Rules 1, 2, and 3. If no further branching rule is applicable we check in $\mathcal{O}(nm)$ time whether the remaining instance is a yes-instance or not. This is possible by Corollary 5.2. The branching vectors are $(1, 2)$ for Branching Rule 1, and $(1, 2, 3)$ for Branching Rules 2 and 3. This delivers a branching factor smaller than 1.8393. We next describe in detail how we can find one of the structures we branch on, in such a way that the algorithm described above runs in $\mathcal{O}(1.84^k \cdot nm)$ time

Isolated vertices do not contribute to the solution of the instance. Thus, we delete all isolated vertices in $\mathcal{O}(n)$ time. Hence we can assume $n \leq 2m$ in the following. Afterwards, we compute a maximal packing $\mathcal{P}$ of edge-disjoint bicolored P_3s. Here, we represent a bicolored P_3 by an edge set of size two. We define $P := \bigcup_{p \in \mathcal{P}} p$ as the set of all edges of bicolored P_3s in $\mathcal{P}$. Note that $\mathcal{P}$ and thus P can be found in $\mathcal{O}(nm)$ time by enumerating all bicolored P_3s in G. If $|P| > 2k$, the graph G contains more than k edge-disjoint bicolored P_3s and (G, k) is a no instance. In this case we can stop and return no. Otherwise, we have $|P| \leq 2k$ and use P to compute the structures we apply Branching Rules 1–3 on as follows.

Note that in any LC-, LO- or IIZ- Diamond $G[\{u, v, w, z\}]$ the edges $\{u, v\}$ and $\{v, w\}$ form a bicolored P_3 in G and thus, $\{u, v\} \in P$ or $\{v, w\} \in P$. Therefore, we can find $G[\{u, v, w, z\}]$ in $\mathcal{O}(km)$ time by iterating over all possible tuples (e_1, e_2) with $e_1 \in P$ and $e_2 \in E$ and then check with the adjacency matrix in $\mathcal{O}(1)$ time if the induced subgraph $G[e_1 \cup e_2]$ is an LC-, LO- or IIZ- Diamond.

Furthermore, observe that any CC-Hourglass $G[\{u, v, w, z_1, z_2\}]$ contains the two edge-disjoint bicolored P_3s $\{\{u, v\}, \{v, w\}\}$ and $\{\{z_1, v\}, \{v, z_2\}\}$. Thus, at least two edges of $G[\{u, v, w, z_1, z_2\}]$ are elements of P. Therefore, we can find $G[\{u, v, w, z_1, z_2\}]$ in $\mathcal{O}(k^2 n^2)$ time by iterating over all possible tuples (e_1, e_2, x, y) with $e_1, e_2 \in P$ and $x, y \in V$ and then check with the adjacency matrix in $\mathcal{O}(1)$ time if the induced subgraph $G[e_1 \cup e_2 \cup \{x, y\}]$ is a CC-Hourglass.

Finally, let e_1 be an edge that forms two bicolored P_3s with other edges e_2 and e_3. Again, at least one of the edges e_1, e_2, or e_3 is an element of P. Thus, we can find e_1, e_2, and e_3 in $\mathcal{O}(kn^2)$ time iterating over all triples containing an edge from P and two vertices from V, which form the remaining two endpoints.

Since $n \leq 2m$ we can find one of the structures on which we apply Branching Rules 1–3 in $\mathcal{O}(k^2 nm)$ time. This gives us a total running time of $\mathcal{O}(1.8393^k \cdot k^2 nm) \subseteq \mathcal{O}(1.84^k \cdot nm)$ time as claimed. □

It is possible to improve the branching rules on LO-Diamonds, IIZ-Diamonds, and CC-Hourglasses to obtain a branching vector $(2, 2, 3, 3)$, but branching on LC-Diamonds still needs a branching vector of $(1, 2, 3)$, which is the bottleneck. To put the running time of Theorem 5.6 into perspective note that CLUSTER DELETION, which can be viewed as the uncolored version of BPD, can be solved in $\mathcal{O}(1.42^k + m)$ time [5]. Thus there is a large gap between the running time bounds of the problems. It would be interesting to know if this gap can be closed or if BPD is significantly harder than CLUSTER DELETION.

5.2 On Problem Kernelization

Finally, we consider problem kernelization for BPD parameterized by (k, Δ) and $\ell := m - k$. Recall that Δ denotes the maximum degree of the input graph. We show that BPD admits problem kernels with $\mathcal{O}(k\Delta \min(k, \Delta))$ vertices or at most 2ℓ edges, respectively.

In the following, we provide two reduction rules leading to an $\mathcal{O}(k\Delta\min(k,\Delta))$ vertex kernel for BPD. The first reduction rule deletes all edges which form more than k bicolored P_3s.

Reduction Rule 3. If G contains an edge $\{u,v\}$ such that there exist vertices $w_1,\dots,w_t$ with $t > k$ such that $G[\{u,v,w_i\}]$ is a bicolored P_3 for each i, then remove $\{u,v\}$ and decrease k by one.

Lemma 5.7. *Reduction Rule 3 is correct and can be applied exhaustively in $\mathcal{O}(nm)$ time.*

Proof: First, we prove the correctness of Reduction Rule 3. Let S be a solution for (G,k). Without loss of generality, consider an edge $\{u,v\}\in E_r$ such that there exist vertices $w_1,\dots,w_t$ such that for each i the graph $G[\{u,v,w_i\}]$ is a bicolored P_3. At least one edge of each bicolored P_3, $G[\{u,v,w_i\}]$ is an element of S. Assume towards a contradiction $\{u,v\}\notin S$. In each bicolored P_3, $G[\{u,v,w_i\}]$ the blue edge which is either $\{v,w_i\}$ or $\{u,w_i\}$ has to be removed. Note that for each w_i these are pairwise different edges. Thus, since $t > k$, $|S| > k$, a contradiction to $|S|\le k$. Hence, $\{u,v\}\in S$ and $S\setminus\{\{u,v\}\}$ is a solution for $(G-\{u,v\},k-1)$. For the opposite direction, if S is a solution for $(G-\{u,v\},k-1)$ then $S\cup\{\{u,v\}\}$ is is a solution for (G,k).

Second, we bound the running time of applying Reduction Rule 3 exhaustively. In a first step, for each edge $e\in E$ compute the number of bicolored P_3s containing e. This can be done in $\mathcal{O}(nm)$ time. In a second step, check if an edge $e=\{u,v\}$ is part of more than k bicolored P_3s and remove e if this is the case. After the removal of e, every new bicolored P_3 contains vertices u and v. Hence, for each remaining vertex $w\in V$, check if $G[\{u,v,w\}]$ is a new bicolored P_3 in $G-\{u,v\}$. If yes, then update the number of bicolored P_3s for edges $\{u,w\}$ and $\{v,w\}$. This can be done in $\mathcal{O}(n)$ time. Since $k < m$, the overall running time of Reduction Rule 3 is $\mathcal{O}(nm)$. □

Let $\mathcal{P}$ denote the set of all vertices of G which are part of bicolored P_3s. Then, the set $N[\mathcal{P}]$ contains all vertices which are either part of a bicolored P_3 or which are adjacent to a vertex in a bicolored P_3. In other words, a vertex v is contained in $V\setminus N[\mathcal{P}]$ if and only if each vertex $u\in N[v]$ is not part of any bicolored P_3. In the following, we present a reduction rule to remove all vertices in $V\setminus N[\mathcal{P}]$.

Reduction Rule 4. If G contains a vertex $v\in V$ such that each vertex $u\in N[v]$ is not part of any bicolored P_3, then, remove v from G.

To show that Rule 4 is correct, we provide two simple lemmas about edge deletion sets.

Lemma 5.8. *Let $G=(V,E)$ be a graph and let $S\subseteq E$ be an edge deletion set. If two edges $\{u,v\}\in E$ and $\{v,w\}\in E$ do not form a bicolored P_3 in G and form a bicolored P_3 in $G-S$, then $\{u,w\}\in S$.*

The proof of Lemma 5.8 is trivial and thus omitted.

Lemma 5.9. *Let (G,k) be an instance of* BPD *and let S be a solution for (G,k) of minimum size. Then, there exists an ordering $(e_1,\dots,e_{|S|})$ of S such that for each $i\in\{1,\dots,|S|\}$ the edge e_i is part of a bicolored P_3 in $G-\{e_1,\dots,e_{i-1}\}$.*

Proof: Assume towards a contradiction that such an ordering does not exist. Then, for every ordering of S, there exists a maximal index $1\le i<|S|$ such that there is a finite sequence $(e_1,\dots,e_i)$ where for each $1\le j\le i$ the edge e_j is part of a bicolored P_3 in $G-\{e_1,\dots,e_{j-1}\}$. According to our choice of i, there exists no edge of S which is part of a bicolored P_3 in $G-\{e_1,\dots,e_i\}$. If $G-\{e_1,\dots,e_i\}$ contains a bicolored P_3 formed by $\{x,y\}$ and $\{y,z\}$, then $\{x,y\}\notin S$ and $\{y,z\}\notin S$. This contradicts the fact that S is a solution for (G,k). Otherwise, $G-\{e_1,\dots,e_i\}$ is bicolored-P_3-free. Then, $\{e_1,\dots,e_i\}\subsetneq S$ is a solution for (G,k). This contradicts the fact that S is a solution of minimum size for (G,k). □

We next use the Lemmas 5.8 and 5.9 to show the correctness of Rule 4.

Lemma 5.10. *Reduction Rule 4 is correct and can be applied exhaustively in $\mathcal{O}(nm)$ time.*

Proof:

Let $H := G[V \setminus \{v\}]$. We prove that there exists a solution S for (G, k) if and only if S is also a solution for (H, k).

($\Rightarrow$) Let S be a solution for (G, k). Since H is an induced subgraph of G, $S \cap E(H)$ is a solution for (H, k).

($\Leftarrow$) Let S be a solution for (H, k) of minimum size. If $S = \emptyset$, S is also a solution for (G, k). Thus, in the following we assume that $S \neq \emptyset$. We show that $G - S$ is bicolored-P_3-free. To this end, we provide a claim to show that there are no edge-deletions in the neighborhood of v.

Claim 4. *In S, there are no edge deletions that are incident with some vertex in $N_G[v]$.*

Proof. Assume towards a contradiction that S contains such edge deletions. We consider an ordering $F = (e_1, \dots, e_{|S|})$ from Lemma 5.9 such that every e_i is part of a bicolored P_3 in $H - \{e_1, \dots, e_{i-1}\}$. Note that this implies that every e_i is in a bicolored P_3 in $G - \{e_1, \dots, e_{i-1}\}$. Let $t \in \{1, \dots, |S|\}$ be the minimum index such that $e_t =: \{w, z\}$ is incident with some $w \in N_G[v]$.

Consider the case $z \in N_G[v]$. Then, since $\{w, z\}$ is part of a bicolored P_3 in $G - \{e_1, \dots, e_{t-1}\}$ and $\{w, z\}$ is not part of a bicolored P_3 in G, Lemma 5.8 implies that there exists some index $j < t$ such that $e_j \in S$ is incident with w or with z. This contradicts the minimality of t. Thus, we may assume $z \in N_G^2(v)$ and therefore $w \in N_G(v)$ for the rest of this proof. Without loss of generality we may assume that $\{v, w\}$ is red. Observe that this implies that $\{w, z\}$ is red, since v is not part of any bicolored P_3 in G.

Since $\{w, z\}$ is part of a bicolored P_3 in $G - \{e_1, \dots, e_{t-1}\}$, there exists some vertex y such that $G - \{e_1, \dots, e_{t-1}\}[w, z, y]$ is a bicolored P_3. We show that the following cases are contradictory.

Case 1: $y \in N_G(v)$**.** Then, since $G - \{e_1, \dots, e_{t-1}\}[w, z, y]$ is a bicolored $P_3$and $G[w, z, y]$ is not a bicolored P_3, Lemma 5.8 implies that there exists some index $j < t$ and there is an edge e_j in the ordering F that is incident with y or with w. This contradicts the minimality of t since y and w are elements of $N_G(v)$.

Case 2: $y \in N_G^2(v)$**.** Then, $\{w, z\}$ and $\{w, y\}$ do not form a bicolored P_3 in $G - \{e_1, \dots, e_{t-1}\}$ since all edges in $E_G(\{w\}, N_G^2(v))$ are red since v is not part of any bicolored P_3 in G. Consequently, $\{w, z\}$ and $\{z, y\}$ form a bicolored P_3 in $G - \{e_1, \dots, e_{t-1}\}$. Then, analogous to Case 1, Lemma 5.8 implies that there exists some e_j with $j < t$ and $e_j = \{w, y\}$ which is a contradiction to the minimality of t.

Case 3: $y \in V \setminus (N_G(v) \cup N_G^2(v))$**.** Then, $\{w, z\}$ and $\{z, y\}$ form a bicolored P_3 in $G - \{e_1, \dots, e_{t-1}\}$. Moreover, $\{w, y\} \notin E$, since $y \notin N_G^2(v)$. Then, $\{w, z\}$ and $\{z, y\}$ also form a bicolored P_3 in G contradicting the fact that no vertex in $N_G[v]$ is part of a bicolored P_3. $\diamond$

We next use Claim 4 to show that $G - S$ is bicolored-P_3-free. Observe that it suffices to show that no edge incident with v is part of a bicolored P_3 in $G - S$. Consider an edge $\{u, v\} \in E$. Then, by Claim 4, u and v are incident with the same edges in G as in $G - S$. Therefore, since $\{u, v\}$ is not part of any bicolored P_3 in G, the edge $\{u, v\}$ is not part of any bicolored P_3 in $G - S$. Consequently, S is a solution for (G, k).

It remains to consider the running time of applying Reduction Rule 4 exhaustively. In a first step, determine all bicolored P_3s in G. Afterwards, determine for each vertex $v \in V$ if v is part of some bicolored P_3. This needs $\mathcal{O}(nm)$ time. Now, check for each vertex $v \in V$ if each vertex $u \in N[v]$ is not part of any bicolored P_3. This can be done in $\mathcal{O}(m)$ time. The claimed running time follows. □

Theorem 5.11. BPD *admits a* $\mathcal{O}(k\Delta \min(k, \Delta))$*-vertex kernel that can be computed in* $\mathcal{O}(nm)$ *time.*

Proof: First, apply Reduction Rule 3 exhaustively. Second, apply Reduction Rule 4 exhaustively. This needs $\mathcal{O}(nm)$ time altogether. We prove that G contains at most $12 \cdot k\Delta \min(k, \Delta)$ vertices if (G, k) is a yes-instance. Let $\mathcal{P}$ be the set of vertices which are contained in a bicolored P_3 in G, and let P be a maximal set of edge-disjoint bicolored P_3s in G. If (G, k) is a yes-instance for BPD, then $|P| \leq k$. Since Reduction Rule 3 was applied exhaustively, each edge is part of at most $\min(k, 2\Delta)$ bicolored P_3s. Hence, the total number of bicolored P_3s in G is at most $2k \min(k, 2\Delta)$. Consequently, $|\mathcal{P}| \leq 6k \min(k, 2\Delta)$. Since Reduction Rule 4 was applied exhaustively, $V \setminus N[\mathcal{P}] = \emptyset$. In other words, set $\mathcal{P}$ has no second neighborhood in G. Since each vertex has degree at most Δ we have $|N(\mathcal{P})| \leq 6k\Delta \min(k, 2\Delta)$. Hence, the overall number of vertices in G is $12 \cdot k\Delta \min(k, \Delta)$ if (G, k) is a yes-instance for BPD. □

By the above, BPD admits a linear problem kernel in k if G has constant maximum degree. Note that a kernelization by Δ alone is unlikely since BPD is NP-hard even if $\Delta = 8$ by Theorem 3.1. Since BPD is fixed-parameter tractable with respect to parameter k, we can trivially conclude that it admits an exponential-size problem kernel. It is open if there is a polynomial kernel depending only on k while CLUSTER DELETION has a relatively simple $4k$-vertex kernel [16]. Summarizing, BPD seems to be somewhat harder than CLUSTER DELETION if parameterized by k.

In contrast, BPD seems to be easier than CLUSTER DELETION if parameterized by the dual parameter $\ell := m - k$: there is little hope that CLUSTER DELETION admits a problem kernel of size $\ell^{\mathcal{O}(1)}$ [15] while BPD has a trivial linear-size problem kernel as we show below.

Theorem 5.12. BPD *admits a problem kernel with* 2ℓ *edges and vertices which can be computed in* $\mathcal{O}(n+m)$ *time.*

Proof: We show that instances with at least 2ℓ edges are trivial yes-instances. Let $(G = (V, E_r, E_b), k)$ with $|E| \geq 2\ell$ be an instance of BPD. Then, since E_r and E_b form a partition of E, we have $|E_r| \geq \ell$ or $|E_b| \geq \ell$. Without loss of generality let $|E_r| \geq \ell$. Since $|E_b| = m - |E_r| \leq m - \ell = k$, E_b is a solution for (G, k). □

6 Outlook

We have initiated the algorithmic study of a natural edge-deletion problem on edge-colored graphs. In companion work, we considered the problem of destroying paths of length at least 4 that fulfill certain coloring constraints [11, 12]. With this exception, however, the study of graph modification problems on edge-colored graphs has been neglected so far. Consequently, the complexity of many natural problems and a study of natural edge-colored graph classes remain open.

For the particular case of bicolored-P_3-free graphs, we have also left open many questions for future work. First, it would be interesting to further investigate the structure of bicolored-P_3-free graphs. Since each color class may induce an arbitrary graph it seems difficult to obtain a concise and non-trivial structural characterization of these graphs. One may, however, exploit the connection with Gallai colorings which are colorings where no triangle receives more than two colors. In particular, the following characterization of Gallai colorings is known [14, 17]: any Gallai coloring of a complete graph can be obtained by taking some complete 2-colored graph and substituting its vertices with a complete graph and some Gallai coloring of this complete graph. This characterization relies on the decomposition property that in any Gallai coloring of a complete graph G with at least three colors, there is at least one edge color that spans a disconnected graph H. Then, by the property of Gallai colorings, the edges in G that are between two different components H_1 and H_2 of H have the same color.

Second, there are many open questions concerning BICOLORED P_3 DELETION. Does BICOLORED P_3 DELETION admit a polynomial-size kernel for k? Can BICOLORED P_3 DELETION be solved in $2^{\mathcal{O}(n)}$ time? Can BICOLORED P_3 DELETION be solved in polynomial time on graphs that contain no monochromatic P_3? Can BICOLORED P_3 DELETION be solved in polynomial time on graphs that contain no cycle consisting only of blue edges? Even simpler is the following question: Can BICOLORED P_3 DELETION be solved in polynomial time if the subgraphs induced by the red edges and the subgraph induced by the blue edges are each a disjoint union of paths? Moreover, it would be interesting to perform a similar study on BICOLORED P_3 EDITING where we may also insert blue and red edges. Furthermore, it is open if BICOLORED-P_3-FREE COMPLETION where we only may insert red or blue edges is NP-hard. Observe in this context that the uncolored problem CLUSTER COMPLETION can easily be solved by adding all missing edges in each connected component.

Third, it would also be interesting to identify graph problems that are NP-hard on general two-edge colored graphs but polynomial-time solvable on bicolored-P_3-free graphs. Finally, we were not able to resolve the following question: Can we find bicolored P_3s in $\mathcal{O}(n + m)$ time?

Using the connection to Gallai colorings and the decomposition property of Gallai colorings seems to be a promising approach to address these open questions.

References

[1] David Amar and Ron Shamir. Constructing module maps for integrated analysis of heterogeneous biological networks. *Nucleic Acids Research*, 42(7):4208–4219, 2014.

[2] N. R. Aravind, R. B. Sandeep, and Naveen Sivadasan. Dichotomy results on the hardness of H-free edge modification problems. *SIAM Journal on Discrete Mathematics*, 31(1):542–561, 2017.

[3] Amir Ben-Dor, Ron Shamir, and Zohar Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3-4):281–297, 1999.

[4] Sebastian Böcker and Jan Baumbach. Cluster editing. In *Proceedings of the 9th Conference on Computability in Europe (CiE '13)*, volume 7921 of *LNCS*, pages 33–44. Springer, 2013.

[5] Sebastian Böcker and Peter Damaschke. Even faster parameterized cluster deletion and cluster editing. *Information Processing Letters*, 111(14):717–721, 2011.

[6] Ulrik Brandes, Michael Hamann, Ben Strasser, and Dorothea Wagner. Fast quasi-threshold editing. In *Proceedings of the 23rd Annual European Symposium on Algorithms (ESA '15)*, volume 9294 of *LNCS*, pages 251–262. Springer, 2015.

[7] Andreas Brandstädt, Van Bang Le, and Jeremy P. Spinrad. *Graph classes: a survey*. SIAM, Philadelphia, PA, USA, 1999.

[8] Sharon Bruckner, Falk Hüffner, and Christian Komusiewicz. A graph modification approach for finding core-periphery structures in protein interaction networks. *Algorithms for Molecular Biology*, 10:16, 2015.

[9] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

[10] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

[11] Nils Jakob Eckstein. Destroying multicolored paths and cycles in edge-colored graphs. Bachelorarbeit, Philipps-Universität Marburg, 2020.

[12] Nils Jakob Eckstein, Niels Grüttemeier, Christian Komusiewicz, and Frank Sommer. Destroying multicolored paths and cycles in edge-colored graphs. *CoRR*, abs/2104.03138, 2021.

[13] Fedor V. Fomin and Dieter Kratsch. *Exact Exponential Algorithms*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2010.

[14] Tibor Gallai. Transitiv orientierbare Graphen. *Acta Mathematica Hungarica*, 18(1–2):25–66, 1967.

[15] Niels Grüttemeier and Christian Komusiewicz. On the relation of strong triadic closure and cluster deletion. *Algorithmica*, 82(4):853–880, 2020.

[16] Jiong Guo. A more effective linear kernelization for cluster editing. *Theoretical Computer Science*, 410(8-10):718–726, 2009.

[17] András Gyárfás and Gábor Simonyi. Edge colorings of complete graphs without tricolored triangles. *Journal of Graph Theory*, 46(3):211–216, 2004.

[18] Marc Hellmuth, Nicolas Wieseke, Marcus Lechner, Hans-Peter Lenhof, Martin Middendorf, and Peter F. Stadler. Phylogenomics with paralogs. *Proceedings of the National Academy of Sciences*, 112(7):2058–2063, 2015.

[19] John E. Hopcroft and Richard M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, 1973.

[20] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.

[21] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, and Mason A. Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 2014.

[22] Christian Komusiewicz. Tight running time lower bounds for vertex deletion problems. *ACM Transactions on Computation Theory*, 10(2):6:1–6:18, 2018.

[23] John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980.

[24] James Nastos and Yong Gao. Familial groups in social networks. *Social Networks*, 35(3):439–450, 2013.

[25] Jannik Schestag. Liechtenstein-P_3s in two-colored graphs. Bachelorarbeit, Philipps-Universität Marburg, 2019.

[26] Ron Shamir, Roded Sharan, and Dekel Tsur. Cluster graph modification problems. *Discrete Applied Mathematics*, 144(1-2):173–182, 2004.

[27] Frank Sommer and Christian Komusiewicz. Parameterized algorithms for module map problems. *Discrete Applied Mathematics*, 283:396–416, 2020.

[28] Robert Endre Tarjan. A note on finding the bridges of a graph. *Information Processing Letters*, 2(6):160–161, 1974.

[29] Craig A. Tovey. A simplified NP-complete satisfiability problem. *Discrete Applied Mathematics*, 8(1):85–89, 1984.

[30] Tobias Wittkop, Jan Baumbach, Francisco P Lobo, and Sven Rahmann. Large scale clustering of protein sequences with force-a layout based heuristic for weighted cluster editing. *BMC Bioinformatics*, 8(1):396, 2007.

[31] Mihalis Yannakakis. Edge-deletion problems. *SIAM Journal on Computing*, 10(2):297–309, 1981.