



Finding Matching Cuts in H -Free Graphs

Felicia Lucke¹ · Daniël Paulusma² · Bernard Ries¹

Received: 9 October 2022 / Accepted: 24 May 2023
© The Author(s) 2023

Abstract

The well-known NP-complete problem **MATCHING CUT** is to decide if a graph has a matching that is also an edge cut of the graph. We prove new complexity results for **MATCHING CUT** restricted to H -free graphs, that is, graphs that do not contain some fixed graph H as an induced subgraph. We also prove new complexity results for two recently studied variants of **MATCHING CUT**, on H -free graphs. The first variant requires that the matching cut must be extendable to a perfect matching of the graph. The second variant requires the matching cut to be a perfect matching. In particular, we prove that there exists a small constant $r > 0$ such that the first variant is NP-complete for P_r -free graphs. This addresses a question of Bouquet and Picouleau (The complexity of the Perfect Matching-Cut problem. CoRR, [arXiv:2011.03318](https://arxiv.org/abs/2011.03318), (2020)). For all three problems, we give state-of-the-art summaries of their computational complexity for H -free graphs.

Keywords Matching cut · Perfect matching · H -free graph · Computational complexity

Mathematics Subject Classification Mathematics of computing · Graph algorithms

An extended abstract of this paper has appeared in the proceedings of ISAAC 2022 [30].

✉ Daniël Paulusma
daniel.paulusma@durham.ac.uk
Felicia Lucke
felicia.lucke@unifr.ch
Bernard Ries
bernard.ries@unifr.ch

¹ Department of Informatics, University of Fribourg, Fribourg, Switzerland

² Department of Computer Science, Durham University, Durham, UK

1 Introduction

Cut sets and connectivity are central topics in algorithmic graph theory. We consider edge cuts in graphs that have some additional structure. The common property of these cuts is that the edges in them must form a matching. Formally, consider a connected graph $G = (V, E)$. A set $M \subseteq E$ is a *matching* if no two edges in M have a common end-vertex. A set $M \subseteq E$ is an *edge cut*, if V can be partitioned into sets B and R such that M consists of all the edges with one end-vertex in B and the other one in R . Now, M is a *matching cut* if M is a matching that is also an edge cut; see also Fig. 1.

In 1970, Graham [21] introduced graphs with matching cuts as *decomposable* graphs to solve a problem in number theory (on cube numbering). Matching cuts also have applications in ILFI networks [14] and graph drawing [34]. Moreover, they are used for proving hardness of surjective graph homomorphism problems [19], and for determining conflict graphs for WDM networks [1]. The corresponding decision problem, which asks whether a given connected graph has a matching cut, is known as MATCHING CUT.

We also consider two natural variants of MATCHING CUT. First, let G be a connected graph that has a *perfect* matching M , that is, every vertex of G is incident to an edge of M . If M contains a matching cut M' of G , then M is a *disconnected perfect matching* of G ; see again Fig. 1 for an example. The problem DISCONNECTED PERFECT MATCHING is to decide if a graph has a disconnected perfect matching. Every yes-instance of DISCONNECTED PERFECT MATCHING is a yes-instance of MATCHING CUT, but the reverse might not be true; for example, the 3-vertex path has a matching cut but no (disconnected) perfect matching.

Suppose now that we search for a matching cut with a maximum number of edges, or for a disconnected perfect matching with a matching cut that is as large as possible. In both settings, the extreme case is when the matching cut is a perfect matching itself. Such a matching cut is called *perfect*; see Fig. 1. By definition, a perfect matching cut is a disconnected perfect matching, but the reverse might not hold: take the cycle on six vertices which has several disconnected perfect matchings but no perfect matching cut. The problem PERFECT MATCHING CUT is to decide if a connected graph has a perfect matching cut.

All three problems are known to be NP-complete, as we will explain in more detail below. Hence, it is natural to restrict the input to some special graph class to obtain a better understanding of the computational hardness of some problem, or some set of problems. In particular, jumps in complexity can be large and unexpected. To give an extreme example [32], there exist problems that are PSPACE-complete in general but constant-time solvable for every other *hereditary* graph class, i.e., that is closed under vertex deletion.



Fig. 1 The graph P_6 with a matching cut that is not contained in a disconnected perfect matching (left), a matching cut that is properly contained in a disconnected perfect matching (middle) and a perfect matching cut (right). In each figure, thick edges denote matching cut edges

Hereditary graph classes enable a *systematic* study in the computational complexity of graph problems due to the following reasons. First of all, they capture many natural graph classes. Second, it is readily seen that a graph class \mathcal{G} is hereditary if and only if it can be characterized by a (unique) set \mathcal{H} of forbidden induced subgraphs. Hence, it is standard practice to first consider hereditary graph classes obtained by forbidding a single subgraph H . That is, a graph G is *H-free* if G does not contain H as induced subgraph, or equivalently, if G cannot be modified into H by a sequence of vertex deletions. Many classical graph problems and graph parameters have been studied for classes of H -free graphs, as can not only be seen from surveys for e.g. COLOURING [17, 35] or clique-width [12], but also from extensive studies on H -free graphs for specific graphs H , such as bull-free graphs [9] or claw-free graphs [10, 23]. We will also focus on H -free graphs in this paper. Before presenting our results we first discuss relevant known results.

1.1 Known Results

Out of the three problems, MATCHING CUT has been studied most extensively. Already in the eighties, Chvátal [11] proved that MATCHING CUT is NP-complete. Afterwards a large number of complexity results were proven for special graph classes. Here, we only discuss those results that are relevant for our context, whereas results for non-hereditary graph classes can, for example, be found in [5, 27]. In particular, we refer to a recent paper of Chen et al. [8] for a comprehensive overview. Similarly, we refer to [2, 18, 20, 25, 26] for parameterized complexity results and exact algorithms for MATCHING CUT. Moreover, we refer to [3, 20] for complexity results on a generalization of MATCHING CUT called d -CUT. In the latter problem, $d \geq 1$ is some fixed constant, and each vertex is allowed to have at most d neighbours across the cut instead of only one (so matching cuts are 1-cuts).

On the positive side, Bonsma [4] proved that MATCHING CUT is polynomial-time solvable for $K_{1,3}$ -free graphs and P_4 -free graphs. Recently, Feghali [15] proved the same for P_5 -free graphs, which we extended to P_6 -free graphs in [31]. In the latter paper, we also showed that if MATCHING CUT is polynomial-time solvable for H -free graphs, for some graph H , then it is so for $(H + P_3)$ -free graphs (see Sect. 2 for any unexplained notation and terminology).

On the negative side, MATCHING CUT is NP-complete even for $K_{1,4}$ -free graphs. This follows from the construction of Chvátal [11] (see also [4, 26]). Bonsma [4] proved that MATCHING CUT is NP-complete for planar graphs of girth 5, and thus for C_r -free graphs with $r \in \{3, 4\}$. Le and Randerath [28] proved that MATCHING CUT is NP-complete for $K_{1,5}$ -free bipartite graphs. Hence, it is NP-complete for H -free graphs if H has an odd cycle. Via a graph transformation of Moshi [33], NP-completeness for H -free graphs also holds if H has an even cycle, as shown in [31]. Moshi [33] used this graph transformation to show NP-completeness for bipartite graphs where the vertices in one set of the bipartition all have degree exactly 2. Consequently, MATCHING CUT is also NP-complete for H^* -free graphs, where H^* is the 6-vertex graph that looks like the letter “H”, that is, the graph with vertices $a_1, a_2, a_3, b_1, b_2, b_3$ and edges $a_1a_2, a_2a_3, b_1b_2, b_2b_3, a_2b_2$. Feghali [15] proved the existence of an unspecified

constant r such that **MATCHING CUT** is NP-complete for P_r -free graphs; we will show that $r = 27$ in his construction.

We now turn to **DISCONNECTED PERFECT MATCHING**. Even though disconnected perfect matchings have been (implicitly) studied for cubic graphs from a graph-structural point of view in several papers [13, 16], the decision problem itself was only introduced recently by Bouquet and Picouleau [6]. They used a different name, but to avoid confusion with **PERFECT MATCHING CUT**, Le and Telle [29] introduced the notion of disconnected perfect matchings, which we adapted. As observed in [6], it follows from a result of Diwan [13] that every planar cubic bridgeless graph, except the K_4 , has a disconnected perfect matching. Bouquet and Picouleau [6] proved that **DISCONNECTED PERFECT MATCHING** is, among others, polynomial-time solvable for claw-free graphs and P_5 -free graphs, but NP-complete for bipartite graphs (of diameter 4), for $K_{1,4}$ -free planar graphs (each vertex of which has either degree 3 or 4) and for planar graphs with girth 5.

Finally, we discuss the **PERFECT MATCHING CUT** problem. Heggernes and Telle [22] showed that it is a natural problem within their (σ, ρ) -vertex partitioning framework, and they proved that this problem is NP-complete. Le and Telle [29] proved that for every integer $g \geq 3$, **PERFECT MATCHING CUT** is NP-complete even for $K_{1,4}$ -free bipartite graphs of girth g . The same authors showed that the problem is polynomial-time solvable for the class of $S_{1,2,2}$ -free graphs (which contain the classes of $K_{1,3}$ -free graphs and P_5 -free graphs) and for chordal graphs. As explained in [29], the latter result generalizes a known result for interval graphs, for which a branch decomposition of constant mim-width can be computed in polynomial time. We refer to [29] for a brief discussion on the parameterized complexity of **PERFECT MATCHING CUT** and the first non-trivial exact algorithm for solving it.

1.2 New Results

For **MATCHING CUT** on H -free graphs, the remaining cases are when H is a P_{27} -free forest, each vertex of which has degree at most 3, such that H is not an induced subgraph of $P_6 + sP_3$ or $K_{1,3} + sP_3$ for some constant $s \geq 0$. By modifying the construction of Feghali [15], we prove in Sect. 3 that **MATCHING CUT** is NP-complete for $(3P_5, P_{15})$ -free graphs.

For **DISCONNECTED PERFECT MATCHING** on H -free graphs, the remaining cases are when H contains an even cycle of length at least 6, such that every vertex of H has degree at most 3 and H is not an induced subgraph of $K_{1,3}$ or P_5 . Bouquet and Picouleau [6] asked about the complexity of the problem for P_r -free graphs, with $r \geq 6$. We partially answer their question by proving NP-completeness for $(3P_7, P_{19})$ -free graphs in Sect. 3 (via modifying our construction for **MATCHING CUT** for $(3P_5, P_{15})$ -free graphs).

For **PERFECT MATCHING CUT** on H -free graphs, the remaining cases are when H is a forest of maximum degree 3, such that H is not an induced subgraph of $S_{1,2,2}$. In Sect. 4, we first prove that **PERFECT MATCHING CUT** is polynomial-time solvable for graphs of radius at most 2 (and thus also for graphs of diameter at most 2). We use this result to obtain a polynomial-time algorithm for P_6 -free graphs. We also prove

that if PERFECT MATCHING CUT is polynomial-time solvable for H -free graphs, for some graph H , then it is so for $(H + P_4)$ -free graphs.

All our results for PERFECT MATCHING CUT are obtained by combining a number of known propagation rules [27, 29] with new rules that we will introduce. After applying these rules exhaustively, we obtain a graph, parts of which have been allocated to the sides B and R of the edge cut that we are looking for. We will prove that the connected components of the remaining subgraph will be placed completely in B or R , and that this property suffices. By doing so, we extend a known approach with our new rules and show that in this way we widen its applicability.

The following three theorems present the state-of-art for H -free graphs. They are obtained by combining the aforementioned results from [4, 6, 11, 28, 29, 31, 33] with our new results. We write $G' \subseteq_i G$ to indicate that G' is an induced subgraph of G ; as mentioned, recall that all undefined notation can be found in Sect. 2.

Theorem 1 For a graph H , MATCHING CUT on H -free graphs is

- Polynomial-time solvable if $H \subseteq_i sP_3 + K_{1,3}$ or $sP_3 + P_6$ for some $s \geq 0$, and
- NP-complete if $H \supseteq_i C_r$ for some $r \geq 3$, $K_{1,4}$, P_{15} , $3P_5$ or H^* .

Theorem 2 For a graph H , DISCONNECTED PERFECT MATCHING on H -free graphs is

- Polynomial-time solvable if $H \subseteq_i K_{1,3}$ or P_5 , and
- NP-complete if $H \supseteq_i C_r$ for some odd $r \geq 3$, C_4 , $K_{1,4}$, P_{19} or $3P_7$.

Theorem 3 For a graph H , PERFECT MATCHING CUT on H -free graphs is

- Polynomial-time solvable if $H \subseteq_i sP_4 + S_{1,2,2}$ or $sP_4 + P_6$, for some $s \geq 0$, and
- NP-complete if $H \supseteq_i C_r$ for some $r \geq 3$ or $K_{1,4}$.

We state a number of open problems that originate from our systematic study in Sect. 5.

2 Preliminaries

We only consider finite undirected graphs without multiple edges and self-loops. Throughout this section, we let $G = (V, E)$ be a connected graph. Let $u \in V$. The set $N(u) = \{v \in V \mid uv \in E\}$ is the *neighbourhood* of u in G , where $|N(u)|$ is the *degree* of u . A graph F is a *spanning* subgraph of G if $V(F) = V(G)$ and $E(F) \subseteq E(G)$. Let $S \subseteq V$. The *neighbourhood* of S is the set $N(S) = \bigcup_{u \in S} N(u) \setminus S$. The graph $G[S]$ is the subgraph of G induced by $S \subseteq V$, that is, $G[S]$ is the graph obtained from G after deleting the vertices not in S . We write $G' \subseteq_i G$ if G' is an induced subgraph of G . We say that S is a *dominating* set of G , and that $G[S]$ *dominates* G , if every vertex of $V \setminus S$ has at least one neighbour in S . The *domination number* of G is the size of a smallest dominating set of G .

Let $u, v \in V$. The *distance* between u and v in G is the *length* (number of edges) of a shortest path between u and v in G . The *eccentricity* of u is the maximum distance between u and any other vertex of G . The *diameter* of G is the maximum eccentricity

over all vertices of G . The *radius* of G is the minimum eccentricity over all vertices of G . If G is not a tree, then the *girth* of G is the length of a shortest cycle in G .

Let H be a graph. Recall that G is H -free if G does not contain H as an induced subgraph. Let $\{H_1, \dots, H_n\}$ be a set of graphs. Then G is (H_1, \dots, H_n) -free, if G is H_i -free for every $i \in \{1, \dots, n\}$. The graph P_r is the path on r vertices. The graph C_r is the cycle on r vertices. A bipartite graph with non-empty partition classes V_1 and V_2 is *complete* if there is an edge between every vertex of V_1 and every vertex of V_2 . If $|V_1| = k$ and $|V_2| = \ell$, we write $K_{k,\ell}$. The graph $K_{1,\ell}$ is the *star* on $\ell + 1$ vertices. The graph $K_{1,3}$ is also known as the *claw*. For $1 \leq h \leq i \leq j$, the graph $S_{h,i,j}$ is the tree with one vertex of degree 3, whose (three) leaves are at distance h, i and j from the vertex of degree 3. Observe that $S_{1,1,1} = K_{1,3}$. We need the following known result (which has been strengthened in [7]).

Theorem 4 ([37]) *A graph G is P_6 -free if and only if each connected induced subgraph of G contains a dominating induced C_6 or a dominating (not necessarily induced) complete bipartite graph. Moreover, such a dominating subgraph of G can be found in polynomial time.*

Let G_1 and G_2 be two vertex disjoint graphs. The graph $G_1 + G_2 = (V(G_1) \cup V(G_2), E(G_1) \cup E(G_2))$ is the *disjoint union* of G_1 and G_2 . For a graph G , the graph sG is the disjoint union of s copies of G . Let H^* be the ‘‘H’’-graph, which is the graph on six vertices obtained from the $2P_3$ by adding an edge joining the middle vertices of the two P_3 s.

A *red–blue colouring* of G colours every vertex of G either red or blue. If every vertex of a set $S \subseteq V$ has the same colour (red or blue), then S (and also $G[S]$) are called *monochromatic*. A red–blue colouring is *valid*, if every blue vertex has at most one red neighbour; every red vertex has at most one blue neighbour; and both colours red and blue are used at least once. If a red vertex u has a blue vertex neighbour v , then u and v are *matched*. See also Fig. 1.

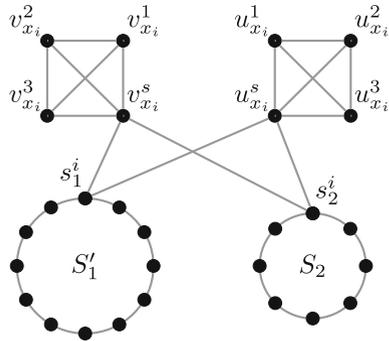
For a valid red–blue colouring of G , we let R be the *red* set consisting of all vertices coloured red and B be the *blue* set consisting of all vertices coloured blue (so $V = R \cup B$). Moreover, the *red interface* is the set $R' \subseteq R$ consisting of all vertices in R with a (unique) blue neighbour, and the *blue interface* is the set $B' \subseteq B$ consisting of all vertices in B with a (unique) red neighbour in R . A red–blue colouring of G is *perfect*, if it is valid and moreover $R' = R$ and $B' = B$. A red–blue colouring of a graph G is *perfect-extendable*, if it is valid and $G[R \setminus R']$ and $G[B \setminus B']$ both contain a perfect matching. In other words, the matching given by the valid red–blue colouring can be extended to a perfect matching in G or, equivalently, is contained in a perfect matching in G .

We can now make the following observation, which is easy to see (the notion of red–blue colourings has been used before; see, for example, [15, 31]).

Observation 5 *Let G be a connected graph. The following three statements hold:*

- (i) G has a matching cut if and only if G has a valid red–blue colouring;
- (ii) G has a disconnected perfect matching if and only if G has a perfect-extendable red–blue colouring;
- (iii) G has a perfect matching cut if and only if G has a perfect red–blue colouring.

Fig. 2 The cliques S'_1 and S_2 together with the variable gadget of x_i . For readability, the edges inside S'_1 and S_2 have been omitted. Note that S'_1 consists of all vertices of S_1 and all clause vertices



3 Our NP-Completeness Results

We prove two NP-completeness results in this section by reducing from EXACT POSITIVE 1-IN-3 SAT. This problem takes as input a pair (X, C) , where X is a set of variables and C is a set of clauses, each containing exactly three literals, all three of which are positive. Moreover, each variable of X appears in exactly three clauses of C . The question is whether there exists a truth assignment, such that each clause contains exactly one true literal.

Theorem 6 ([36]) EXACT POSITIVE 1-IN-3 SAT is NP-complete.

Theorem 7 is our first new result. Its proof follows from Feghali’s construction [15] after making some minor modifications to it. For completeness, and since we use the modified construction as a basis for the proof of Theorem 10, we added a detailed proof. Recall that Feghali [15] showed that MATCHING CUT is NP-complete for P_r -free graphs, for some unspecified constant r . We will show that $r = 27$ in [15]; see Remark 9 below.

Theorem 7 MATCHING CUT is NP-complete for $(3P_5, P_{15})$ -free graphs.

Proof MATCHING CUT is in NP, since it is possible to check in polynomial time if a given red–blue-colouring is valid or not. To prove NP-hardness, we will use a reduction from EXACT POSITIVE 1-IN-3 SAT, which is NP-complete by Theorem 6. Let \mathcal{I} be an instance of EXACT POSITIVE 1-IN-3 SAT with variable set X and clause set C . We will build a graph $G_{\mathcal{I}}$ (see also Figs. 2 and 3):

- For every $x_i \in X$, construct a variable gadget consisting of two disjoint cliques of size 4, U_{x_i} and V_{x_i} , with vertex set $\{u_{x_i}^s, u_{x_i}^1, u_{x_i}^2, u_{x_i}^3\}$ and $\{v_{x_i}^s, v_{x_i}^1, v_{x_i}^2, v_{x_i}^3\}$, respectively;
- Add two cliques S_1 and S_2 with vertex set $\{s_1^1, \dots, s_1^{|X|}\}$ and $\{s_2^1, \dots, s_2^{|X|}\}$, respectively;
- For every $i \in \{1, \dots, |X|\}$, add the edges $s_1^i u_{x_i}^s, s_1^i v_{x_i}^s, s_2^i u_{x_i}^s, s_2^i v_{x_i}^s$;
- For every $c_j \in C$, construct a clause gadget on clause vertices $v_{c_j}, u_{c_j}^1, u_{c_j}^2$, and auxiliary vertices $a_{c_j}^1, \dots, a_{c_j}^6$.

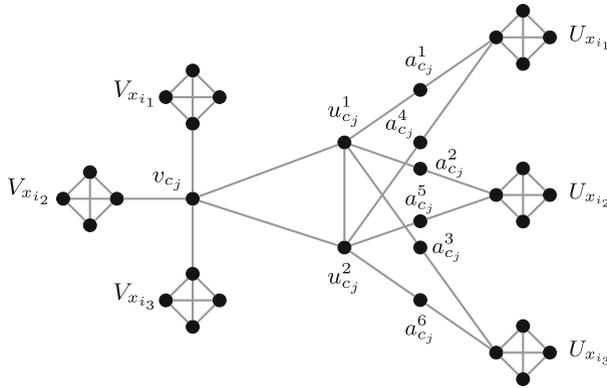


Fig. 3 The clause gadget for clause $c_j = x_{i_1} \vee x_{i_2} \vee x_{i_3}$

- Add edges between all clause vertices of all clause gadgets to obtain a clique and add all edges between S_1 and this clique. We call the resulting clique S'_1 .
- For every $j \in \{1, \dots, |C|\}$, add the edges $u_{c_j}^1 a_{c_j}^\ell$, for $\ell = 1, 2, 3$ and $u_{c_j}^2 a_{c_j}^\ell$, for $\ell = 4, 5, 6$;
- For every $x_i \in X$ occurring in clauses c_{j_1}, c_{j_2} and c_{j_3} , add the edges $v_{x_i}^k v_{c_{j_k}}$, for $k = 1, 2, 3$; and
- For every $c_j \in C$ such that $c_j = x_{i_1} \vee x_{i_2} \vee x_{i_3}$, add the edges $u_{x_{i_k}}^k a_{c_j}^k$ and $u_{x_{i_k}}^k a_{c_j}^{k+3}$, for $k = 1, 2, 3$.

We claim that \mathcal{I} admits a truth assignment such that each clause contains exactly one true literal if and only if $G_{\mathcal{I}}$ admits a valid red–blue–colouring.

First suppose that $G_{\mathcal{I}}$ admits a valid red–blue–colouring. We start with some useful claims. □

Claim 7.1 For any variable $x_i \in X$, $i = 1, \dots, |X|$, both V_{x_i} and U_{x_i} are monochromatic. Furthermore, S'_1 and S_2 are each monochromatic.

Proof This immediately follows from the fact these sets are cliques of size at least 3. □

We say that a monochromatic set has *colour* red or blue if all its vertices are coloured red or blue, respectively.

Claim 7.2 It holds that S'_1 and S_2 have different colours.

Proof Suppose for a contradiction that S'_1 and S_2 have the same colour. We may assume without loss of generality that S'_1 and S_2 are both coloured blue. Since for every variable $x_i \in X$, $i = 1, \dots, |X|$, there exist vertices $u_{x_i}^s$ and $v_{x_i}^s$ having each a neighbour in both S'_1 and S_2 , it follows that every variable gadget is coloured blue. Thus, both neighbours of each auxiliary vertex are blue, which forces the auxiliary vertices to be blue themselves. It follows that all vertices in $G_{\mathcal{I}}$ are coloured blue. Hence, the colouring is not valid, a contradiction. □

Claim 7.3 For every variable $x_i \in X$, $i \in \{1, \dots, |X|\}$, U_{x_i} and V_{x_i} have different colours.

Proof Suppose for a contradiction that for some variable $x_i \in X$, $i \in \{1, \dots, |X|\}$, U_{x_i} and V_{x_i} have the same colour. We may assume without loss of generality that they are both coloured blue. Since s_1^i and s_2^i are both adjacent to $u_{x_i}^s$ and to $v_{x_i}^s$, it follows that they are both coloured blue. Now it follows from Claim 7.1, that S'_1 and S_2 must both be coloured blue, a contradiction to Claim 7.2. \square

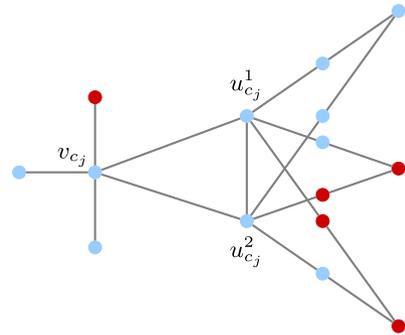
Claim 7.4 For every clause $c_j \in C$, exactly two neighbours of v_{c_j} outside of S'_1 have the same colour as v_{c_j} .

Proof We may assume without loss of generality that v_{c_j} is coloured blue. Let $c_j = (x_{i_1} \vee x_{i_2} \vee x_{i_3})$. Without loss of generality, let $v_{x_{i_1}}^1, v_{x_{i_2}}^2, v_{x_{i_3}}^3$ be the three neighbours of v_{c_j} outside of S'_1 and let $u_{x_{i_1}}^1, u_{x_{i_2}}^2$ and $u_{x_{i_3}}^3$ be the neighbours of the auxiliary vertices $a_{c_j}^k$, $k = 1, \dots, 6$. By definition of a valid red–blue–colouring, v_{c_j} has at least two neighbours outside of S'_1 that are coloured blue. Suppose for a contradiction that the vertices $v_{x_{i_1}}^1, v_{x_{i_2}}^2, v_{x_{i_3}}^3$ are all coloured blue. Then, it follows from Claim 7.1, that $V_{x_{i_1}}, V_{x_{i_2}}, V_{x_{i_3}}$ must be coloured blue. Since v_{c_j} is coloured blue, Claim 7.1 also implies that all vertices in S'_1 , and in particular $u_{c_j}^1$ and $u_{c_j}^2$, are coloured blue. Let A_1 (resp. A_2) be the set of auxiliary vertices which are adjacent to $u_{c_j}^1$ (resp. $u_{c_j}^2$). Then, at least two vertices in A_1 and two vertices in A_2 are coloured blue. Since $u_{x_{i_1}}^1, u_{x_{i_2}}^2$ and $u_{x_{i_3}}^3$ have each one neighbour in A_1 and one neighbour in A_2 , it follows that one of them has two blue neighbours in $A_1 \cup A_2$, and is therefore coloured blue. We may assume without loss of generality that this vertex is $u_{x_{i_1}}^1$. Using Claim 7.1 again, we get that $U_{x_{i_1}}$ is coloured blue, a contradiction to Claim 7.3. \square

We continue as follows. By Claim 7.1, we may assume without loss of generality that S'_1 is coloured blue. Then, we set every variable $x_i \in X$, $i \in \{1, \dots, |X|\}$, to true for which V_{x_i} has been coloured red. We set all other variables to false. By Claim 7.4, we know that for each clause $c_j \in C$, $j \in \{1, \dots, |C|\}$, there exists exactly one red neighbour of v_{c_j} . Hence, in every clause, exactly one literal is set to true. Since by Claim 7.1, every V_{x_i} , $i \in \{1, \dots, |X|\}$, is monochromatic, it follows that no variable gets both values true and false. Thus, \mathcal{I} admits a truth assignment such that each clause contains exactly one true literal.

Now suppose that \mathcal{I} admits a truth assignment such that each clause contains exactly one true literal. For every variable $x_i \in X$, $i \in \{1, \dots, |X|\}$, that is set to true, we colour V_{x_i} red and U_{x_i} blue; for every other variable $x_i \in X$, we colour V_{x_i} blue and U_{x_i} red. It follows that every vertex v_{c_j} , for $c_j \in C$ and $j \in \{1, \dots, |C|\}$, has exactly one red and two blue neighbours outside of S'_1 , since the truth assignment is such that each clause contains exactly one true literal. Thus, we colour S'_1 blue. If we consider a clause $c_j = (x_{i_1} \vee x_{i_2} \vee x_{i_3})$, we know that exactly one of $U_{x_{i_1}}, U_{x_{i_2}}, U_{x_{i_3}}$ is blue and the other ones are red. Assume without loss of generality that $U_{x_{i_1}}$ is blue and consider $a_{c_j}^1, a_{c_j}^4$, the neighbours of $u_{x_{i_1}}^1 \in U_{x_{i_1}}$. We then colour $a_{c_j}^1, a_{c_j}^4$ blue, since they have two blue neighbours ($u_{x_{i_1}}^1$ and $u_{c_j}^1$ resp. $u_{c_j}^2$). To obtain a valid red–blue colouring

Fig. 4 The given valid red–blue-colouring of the clause gadget for c_j (Color figure online)



of $G_{\mathcal{I}}$, we have to colour one of the vertices $a_{c_j}^2, a_{c_j}^5$ blue and the other one red, and similarly, one of the vertices $a_{c_j}^3, a_{c_j}^6$ blue and the other one red. Since $u_{c_j}^1, u_{c_j}^2$ are both coloured blue and each of them can have at most one red neighbour, we colour $a_{c_j}^2, a_{c_j}^6$ blue and $a_{c_j}^3, a_{c_j}^5$ red (see also Fig. 4). Finally, the only vertices that remain uncoloured are the vertices in S_2 . The cliques S'_1 and S_2 are coloured differently. Hence, as S'_1 is coloured blue, we must colour S_2 red. This gives us a valid red–blue-colouring of $G_{\mathcal{I}}$.

To complete the proof, it remains to show that $G_{\mathcal{I}}$ is $(3P_5, P_{15})$ -free. We first prove that $G_{\mathcal{I}}$ is P_{15} -free. Let P be a longest induced path in $G_{\mathcal{I}}$. Then P can contain at most two vertices from a same clique, since otherwise P would not be induced. Moreover, if P contains two vertices from a clique, then these two vertices must be consecutive in P . Let W_1 and W_2 be two disjoint cliques in $\{U_{x_i}, V_{x_i} \mid x_i \in X\}$. By construction, every path from a vertex in W_1 to a vertex in W_2 contains at least one vertex from one of the cliques S'_1 or S_2 . Hence, P can intersect at most three cliques belonging to some variable gadgets. Moreover, in the case where P intersects three cliques belonging to some variable gadgets, P intersects each of the cliques S'_1 and S_2 as well. Assume that P intersects W_1, S'_1 and W_2 in this order. Then P may contain at most one auxiliary vertex between intersecting W_1 and S'_1 and at most one between intersecting S'_1 and W_2 ; see Fig. 5 for an example. Hence, P contains at most 14 vertices. As P was a longest path in $G_{\mathcal{I}}$, we conclude that $G_{\mathcal{I}}$ is P_{15} -free.

We now prove that G is $3P_5$ -free. Let P be an induced P_5 in $G_{\mathcal{I}}$. First suppose that P does not contain any vertex of the cliques S'_1 and S_2 . Then P only contains vertices from variable gadgets and auxiliary vertices. By the same arguments as above, P can contain at most two vertices from a clique, and every path from a vertex in W_1 to a vertex in W_2 , where W_1 and W_2 are two distinct cliques in $\{U_{x_i}, V_{x_i} \mid x_i \in X\}$, contains at least one vertex from one of the cliques S'_1 or S_2 . Hence, P can contain at most four vertices, a contradiction. We conclude that every induced P_5 in $G_{\mathcal{I}}$ must intersect at least one of the cliques S'_1 or S_2 . Hence, $G_{\mathcal{I}}$ is $3P_5$ -free. \square

Remark 8 It can be verified that the $(3P_5, P_{15})$ -free graph $G_{\mathcal{I}}$ in the proof of Theorem 7 is not $(P_{14} + sP_4)$ -free for any $s \geq 1$; see also Fig. 5, which displays an induced $P_{14} + 4P_4$.

Remark 9 In the graph $G_{\mathcal{I}}$ from the proof of Theorem 7, no vertex of U_{x_i} is adjacent to a vertex of V_{x_i} , for any $x_i \in X$. In Feghali’s construction [15], there is an edge between

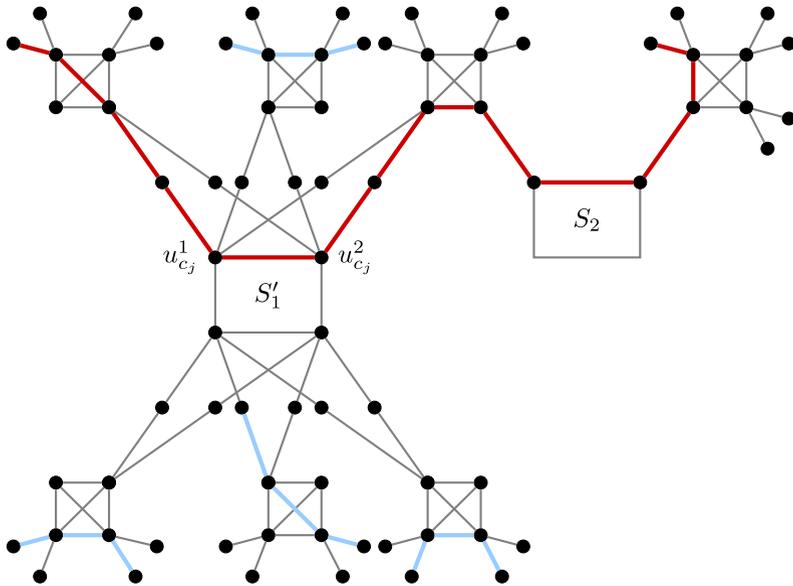


Fig. 5 An induced $P_{14} + 4P_4$, where the induced P_{14} is displayed in red and the four induced P_4 s are in blue (Color figure online)

any two such cliques. This implies that an induced path can use four consecutive vertices inside the same variable gadget. Via similar arguments as in our proof, one can show that Feighali’s construction has an induced P_{26} , but no induced P_{27} (and thus it is P_{27} -free).

We now modify the construction in the proof of Theorem 7 to obtain the following result for DISCONNECTED PERFECT MATCHING, which addresses a question of Bouquet and Picouleau [6].

Theorem 10 DISCONNECTED PERFECT MATCHING is NP-complete for $(3P_7, P_{19})$ -free graphs.

Proof We first note that DISCONNECTED PERFECT MATCHING belongs to NP, as we can verify in polynomial time if a given perfect-extendable red–blue-colouring is valid or not.

In order to prove NP-hardness, we reduce from EXACT POSITIVE 1-IN-3 SAT, which is NP-complete by Theorem 6. Let \mathcal{I} be an instance of EXACT POSITIVE 1-IN-3 SAT with variable set X and clause set C . We build a graph $G_{\mathcal{I}}$ similarly to the graph in Theorem 10 (see also Figs. 6 and 7):

- For every $x_i \in X$, construct a variable gadget consisting of two disjoint cliques of size 7, U_{x_i} and V_{x_i} , with vertex set $\{u_{x_i}^s, u_{x_i}^1, \dots, u_{x_i}^6\}$ and $\{v_{x_i}^s, v_{x_i}^1, \dots, v_{x_i}^6\}$, respectively;
- Add two cliques S_1 and S_2 with vertex set $\{s_1^1, \dots, s_1^{|X|}\}$ and $\{s_2^1, \dots, s_2^{|X|}\}$, respectively;

Fig. 6 The cliques S'_1 and S_2 , together with the variable gadget of x_i . For readability, the edges inside S'_1 and S_2 have been omitted. Note that S'_1 consists of all vertices of S_1 and all clause vertices

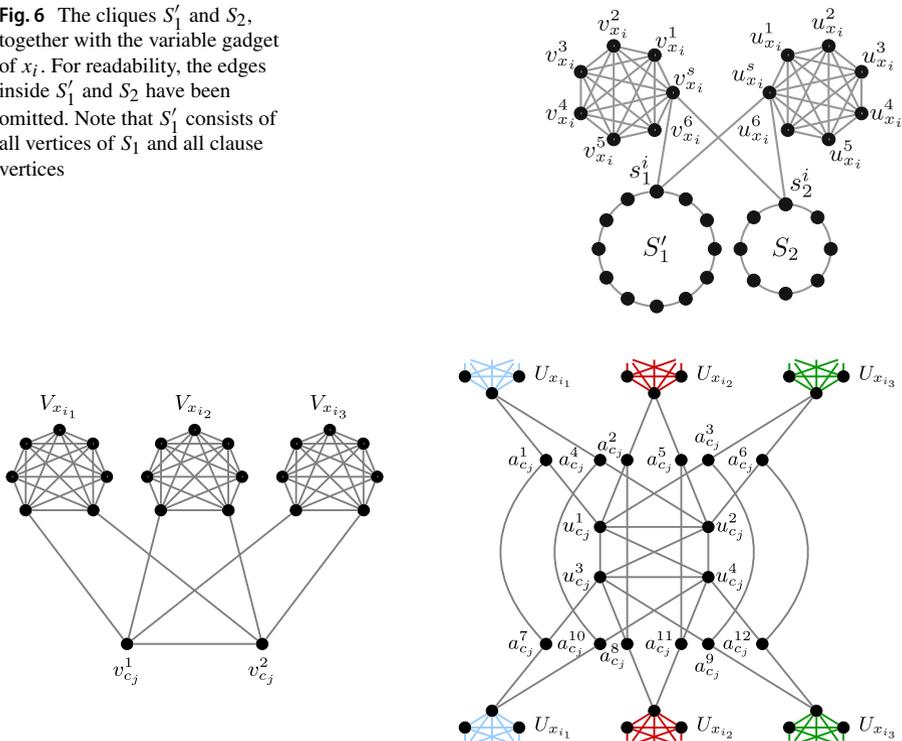


Fig. 7 The clause gadget for the clause $c_j = x_{i_1} \vee x_{i_2} \vee x_{i_3}$. The vertices $v_{c_j}^1, v_{c_j}^2, u_{c_j}^1, \dots, u_{c_j}^4$ belong to clique S'_1 ; for readability, we omitted the edges between $v_{c_j}^1, v_{c_j}^2$, and $u_{c_j}^1, \dots, u_{c_j}^4$. Moreover, coloured edges of the same colour belong to the same clique $U_{x_{i_j}}$, for $j = 1, 2, 3$ (again, for readability) (Color figure online)

- For every $i \in \{1, \dots, |X|\}$, add the edges $s_1^i u_{x_i}^s, s_1^i v_{x_i}^s, s_2^i u_{x_i}^s, s_2^i v_{x_i}^s$;
- For every $c_j \in C$, construct a clause gadget on clause vertices $v_{c_j}^1, v_{c_j}^2, u_{c_j}^1, u_{c_j}^2, u_{c_j}^3, u_{c_j}^4$, and auxiliary vertices $a_{c_j}^1, \dots, a_{c_j}^6$.
- Add edges between all clause vertices of all clause gadgets to obtain a clique and add all edges between this clique and S_1 . We call the resulting clique S'_1 .
- For every $j \in \{1, \dots, |C|\}$, add the edges $u_{c_j}^1 a_{c_j}^\ell$, for $\ell = 1, 2, 3, u_{c_j}^2 a_{c_j}^\ell$, for $\ell = 4, 5, 6, u_{c_j}^3 a_{c_j}^\ell$, for $\ell = 7, 8, 9$ and $u_{c_j}^4 a_{c_j}^\ell$, for $\ell = 10, 11, 12$. Add also the edges $a_{c_j}^\ell a_{c_j}^{\ell+6}$, for $\ell = 1, \dots, 6$;
- For every $x_i \in X$ occurring in clauses c_{j_1}, c_{j_2} and c_{j_3} , add the edges $v_{x_i}^k v_{c_{j_k}}^1$ and $v_{x_i}^{k+3} v_{c_{j_k}}^2$, for $k = 1, 2, 3$; and
- For every $c_j \in C$ such that $c_j = x_{i_1} \vee x_{i_2} \vee x_{i_3}$, add the edges $u_{x_{i_k}}^k a_{c_j}^k, u_{x_{i_k}}^k a_{c_j}^{k+3}, u_{x_{i_k}}^{k+3} a_{c_j}^{k+6}$ and $u_{x_{i_k}}^{k+3} a_{c_j}^{k+9}$, for $k = 1, 2, 3$.

We claim that \mathcal{I} has a truth assignment such that each clause contains exactly one true literal if and only if $G_{\mathcal{I}}$ has a perfect-extendable red–blue-colouring.

First suppose that $G_{\mathcal{I}}$ admits a valid perfect-extendable red–blue-colouring. We start with some useful claims, the first three have the same proof as Claims 7.1, 7.2, 7.3 in the proof of Theorem 7. □

Claim 10.1 *For any variable $x_i \in X$, $i = 1, \dots, |X|$, both V_{x_i} and U_{x_i} are monochromatic. Furthermore, S'_1 and S_2 are also monochromatic.*

Claim 10.2 *It holds that S'_1 and S_2 have different colours.*

Claim 10.3 *For every variable $x_i \in X$, $i \in \{1, \dots, |X|\}$, the cliques U_{x_i} and V_{x_i} have different colours.*

Claim 10.4 *For every clause $c_j \in C$, $j \in \{1, \dots, |C|\}$, exactly two neighbours of $v_{c_j}^1$ (resp. $v_{c_j}^2$) outside of S'_1 have the same colour as $v_{c_j}^1$ (resp. $v_{c_j}^2$).*

Proof We may assume without loss of generality that $v_{c_j}^1$ (resp. $v_{c_j}^2$) is coloured blue. By symmetry, it is enough to prove the claim for $v_{c_j}^1$. Let $c_j = (x_{i_1} \vee x_{i_2} \vee x_{i_3})$. Without loss of generality, let $v_{x_{i_1}}^1, v_{x_{i_2}}^2, v_{x_{i_3}}^3$ be the three neighbours of $v_{c_j}^1$ outside of S'_1 and let $u_{x_{i_1}}^1, u_{x_{i_2}}^2$ and $u_{x_{i_3}}^3$ be the neighbours of the auxiliary vertices $a_{c_j}^k$, $k = 1, \dots, 6$. By definition of a perfect-extendable red–blue-colouring, $v_{c_j}^1$ has at least two neighbours outside of S'_1 that are coloured blue. Suppose for a contradiction that the vertices $v_{x_{i_1}}^1, v_{x_{i_2}}^2, v_{x_{i_3}}^3$ are all coloured blue. Then, it follows from Claim 10.1 that $V_{x_{i_1}}, V_{x_{i_2}}, V_{x_{i_3}}$ must be coloured blue. Since $v_{c_j}^1$ is coloured blue, Claim 10.1 also implies that all vertices in S'_1 , in particular $u_{c_j}^1$ and $u_{c_j}^2$ are coloured blue. Let A_1 (resp. A_2) be the set of auxiliary vertices which are a neighbour of $u_{c_j}^1$ (resp. $u_{c_j}^2$). Then, at least two vertices in A_1 and two vertices in A_2 are coloured blue. Since $u_{x_{i_1}}^1, u_{x_{i_2}}^2$ and $u_{x_{i_3}}^3$ have each one neighbour in A_1 and one neighbour in A_2 , it follows that one of them has two blue neighbours in $A_1 \cup A_2$, and is thus coloured blue. We may assume without loss of generality that this vertex is $u_{x_{i_1}}^1$. Using Claim 10.1 again, we get that $U_{x_{i_1}}$ is coloured blue, a contradiction to Claim 10.3.

Since $v_{c_j}^1$ and $v_{c_j}^2$ are both in S'_1 , using Claim 10.1 we get that they are both blue. For each neighbour of $v_{c_j}^1$ there is a neighbour of $v_{c_j}^2$ in the same clique and vice versa. Hence, they have the same number of blue neighbours. □

We continue as follows. By Claim 10.1, we may assume without loss of generality that S'_1 is coloured blue. We set every variable $x_i \in X$ for which V_{x_i} has been coloured red to true. We set all other variables to false. By Claim 10.4, we know that for each clause $c_j \in C$, $j \in \{1, \dots, |C|\}$, there is exactly one red neighbour of $v_{c_j}^1$ (resp. $v_{c_j}^2$). Hence, in every clause, exactly one literal is set to true. Since by Claim 10.1, every V_{x_i} , $i \in \{1, \dots, |X|\}$, is monochromatic, it follows that no variable is both true and false. Thus, \mathcal{I} admits a truth assignment such that each clause contains exactly one true literal.

Conversely, assume now that \mathcal{I} admits a truth assignment such that each clause contains exactly one true literal. For every variable $x_i \in X$, $i \in \{1, \dots, |X|\}$, that

is set to true, we colour V_{x_i} red and U_{x_i} blue; for every other variable $x_i \in X$, we colour V_{x_i} blue and U_{x_i} red. It follows that every vertex $v_{c_j}^k, c_j \in C$ and $j \in \{1, \dots, |C|\}, k = 1, 2$, has exactly one red and two blue neighbours outside of S'_1 , since the truth assignment is such that each clause contains exactly one true literal. Thus, we colour S'_1 blue. If we consider a clause $c_j = (x_{i_1} \vee x_{i_2} \vee x_{i_3})$, we know that exactly one of $U_{x_{i_1}}, U_{x_{i_2}}, U_{x_{i_3}}$ is blue and the other ones are red. Assume, without loss of generality, that $U_{x_{i_1}}$ is blue and consider $a_{c_j}^1, a_{c_j}^4$, the neighbours of $u_{x_{i_1}}^1 \in U_{x_{i_1}}$ and $a_{c_j}^7, a_{c_j}^{10}$, the neighbours of $u_{x_{i_1}}^4 \in U_{x_{i_1}}$. We then colour $a_{c_j}^1, a_{c_j}^4, a_{c_j}^7, a_{c_j}^{10}$ blue, since they have two blue neighbours ($\{u_{x_{i_1}}^1, u_{c_j}^1\}, \{u_{x_{i_1}}^4, u_{c_j}^2\}, \{u_{x_{i_1}}^4, u_{c_j}^3\}$ resp. $\{u_{x_{i_1}}^4, u_{c_j}^4\}$). To obtain a perfect-extendable red–blue colouring of $G_{\mathcal{I}}$, we have to colour one of the vertices $a_{c_j}^2, a_{c_j}^5$ blue and the other one red, and similarly, one of the vertices $a_{c_j}^3, a_{c_j}^6$ blue and the other one red. Since $u_{c_j}^1, u_{c_j}^2$ are both coloured blue and each of them can have at most one red neighbour, we colour $a_{c_j}^2, a_{c_j}^6$ blue and $a_{c_j}^3, a_{c_j}^5$ red. With the same arguments we colour $a_{c_j}^8, a_{c_j}^{12}$ blue and $a_{c_j}^9, a_{c_j}^{11}$ red. Finally, the only vertices that remain uncoloured are the vertices in S_2 . The cliques S'_1 and S_2 are coloured differently. Hence, as S'_1 is coloured blue, we must colour S_2 red. This gives us a valid red–blue-colouring of $G_{\mathcal{I}}$.

We must still verify that this valid red–blue-colouring is perfect-extendable. First, consider the vertices s_j^i , for $i \in \{1, \dots, |X|\}$ and $j \in \{1, 2\}$. We know that each vertex s_j^i has two neighbours $v_{x_i}^s$ and $u_{x_i}^s$ outside of $S'_1 \cup S_2$. Since the red–blue-colouring is valid, it follows that exactly one of s_1^i, s_2^i and exactly one of $v_{x_i}^s$ and $u_{x_i}^s$ is blue. So every vertex s_j^i has a neighbour of the opposite colour and hence, can be matched with it. The same holds for the vertices $v_{x_i}^s, u_{x_i}^s$, for all $x_i \in X$.

Next, consider a clause $c_j \in C$, such that $c_j = x_{i_1} \vee x_{i_2} \vee x_{i_3}$. It follows from the construction of our valid red–blue-colouring that we may assume without loss of generality that $V_{x_{i_1}}, U_{x_{i_2}}, U_{x_{i_3}}$ are coloured red and $V_{x_{i_2}}, V_{x_{i_3}}, U_{x_{i_1}}$ are coloured blue. Since every clause vertex is coloured blue and has exactly one red neighbour, it follows that all clause vertices in the clause gadget of c_j can be matched and the same holds for the vertices in $V_{x_{i_1}} \setminus \{v_{x_{i_1}}^s\}$. The auxiliary vertices which are adjacent to vertices in $U_{x_{i_2}} \cup U_{x_{i_3}}$ have exactly one neighbour of the opposite colour and thus, they can be matched with either clause vertices or vertices in $U_{x_{i_2}} \cup U_{x_{i_3}}$ (see also Fig. 8). The auxiliary vertices $a_{c_j}^1, a_{c_j}^4, a_{c_j}^7$ and $a_{c_j}^{10}$, which are neighbours of $U_{x_{i_1}}$, are all coloured blue. In this case, we consider the edges $a_{c_j}^1 a_{c_j}^7$ and $a_{c_j}^4 a_{c_j}^{10}$ as matching edges.

The only vertices that remain unmatched are vertices in variable gadgets of variables $x_i \in X, i \in \{1, \dots, |X|\}$, which are set to false. Since all vertices $v_{x_i}^s$ and $u_{x_i}^s$ are already matched, for all $x_i \in X, i \in \{1, \dots, |X|\}$, there remains an even number of unmatched vertices in each clique of these variable gadgets, all coloured with the same colour. We can easily find matching edges inside these cliques. We conclude that our valid red–blue-colouring is indeed perfect-extendable.

To complete the proof, it remains to show that $G_{\mathcal{I}}$ is indeed $(3P_7, P_{19})$ -free. To show that it is P_{19} -free, we follow the same arguments as in the proof of Theorem 10, where we showed that the corresponding graph was P_{15} -free. Let P be a longest

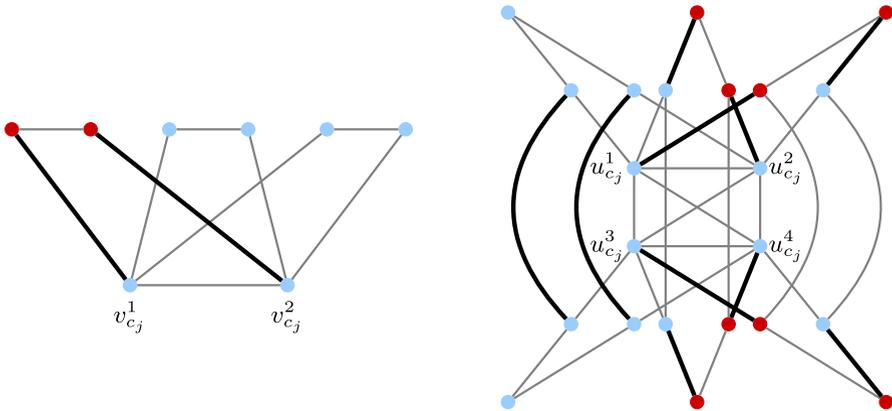
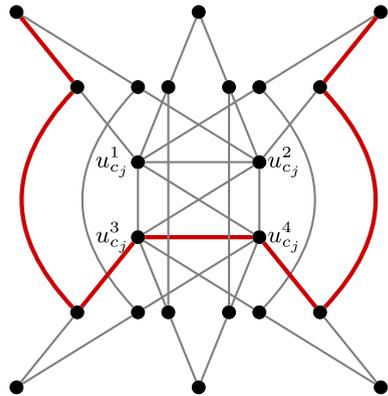


Fig. 8 A perfect-extendable red–blue-colouring of the clause gadget. Notice that vertices $v_{c_j}^1, v_{c_j}^2, u_{c_j}^1, \dots, u_{c_j}^4$ belong to the clique S'_1 , but for the sake of readability, we omitted the edges between vertices $v_{c_j}^1, v_{c_j}^2$ and vertices $u_{c_j}^1, \dots, u_{c_j}^4$ (Color figure online)

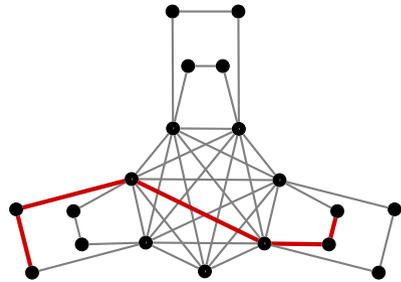
Fig. 9 A path intersecting a clause gadget and containing four auxiliary vertices



induced path in $G_{\mathcal{I}}$. It holds that P can contain at most two vertices from each clique, else P would not be induced. Also, if P contains two vertices from the same clique, then these two vertices are necessarily consecutive in P . Let W_1, W_2 be two disjoint cliques in $\{U_{x_i}, V_{x_i} \mid x_i \in X\}$. Every path from a vertex in W_1 to a vertex in W_2 contains at least one vertex from one of the cliques S'_1 and S_2 . Hence, P can intersect at most three cliques belonging to some variable gadgets. Moreover, in the case where P does intersect three cliques belonging to some variable gadgets, then P intersects each of the cliques S'_1 and S_2 as well. Directly before intersecting S'_1 the path P may contain up to two auxiliary vertices (see also Fig. 9). Similarly, directly after intersecting S'_1 , the path P may contain another two auxiliary vertices. Moreover, the first two, respectively the last two, vertices in P may correspond to auxiliary vertices. Hence, P contains at most 18 vertices, and thus, $G_{\mathcal{I}}$ is indeed P_{19} -free.

We now prove that $G_{\mathcal{I}}$ is $3P_7$ -free. Let $G' = G[V(G_{\mathcal{I}}) \setminus (V(S'_1) \cup V(S_2))]$. The graph G' consists of two types of connected components: (i) every V_{x_i} , for $x_i \in X$,

Fig. 10 A connected component of type (ii) obtained from the removal of S'_1 and S_2 . A longest path in this connected component is shown in red (Color figure online)



corresponds to a clique of size 7; (ii) every U_{x_i} , for $x_i \in X$, together with some auxiliary vertices corresponds to a connected component as shown in Fig. 10. Any induced path contains at most two vertices of the same clique. Also observe that any induced path contains at most six vertices from a connected component containing a clique U_{x_i} (see Fig. 10). Hence, every induced path with length at least 7 has to contain a vertex in one of the cliques S'_1 and S_2 . It immediately follows that $G_{\mathcal{I}}$ is $3P_7$ -free. \square

Remark 11 Just like we showed in Remark 8 that the gadget in the proof of Theorem 7 is not $(P_{14} + sP_4)$ -free for any integer $s \geq 0$, it can be verified that the $(3P_7, P_{19})$ -free graph $G_{\mathcal{I}}$ in the proof of Theorem 10 is not $(P_{18} + sP_6)$ -free for any $s \geq 1$.

4 Our Polynomial Results

In Sect. 4.5 we show that PERFECT MATCHING CUT is polynomial-time solvable for graphs of radius at most 2, for P_6 -free graphs and for $(H + P_4)$ -free graphs should PERFECT MATCHING CUT be polynomial-time solvable for H -free graphs. The proofs of these results are all based on a common approach. This approach is described in Sects. 4.1–4.4, but we give an outline of it below.

Outline. We prove the above results via a common approach. First, in Sect. 4.1, we deal with the case where the input graph G has a small dominating set. This case can be dealt with by using brute force. Now suppose that we find a dominating set D of G that is not small. In Sect. 4.1, we also show that we can test in polynomial time whether G has a perfect red–blue colouring in which D is monochromatic.

Due to the above it remains to check if G has a perfect red–blue colouring in which the dominating set D that we found is not monochromatic. We branch by essentially guessing an edge whose end-vertices are coloured with different colours. We then exhaustively apply, in Sect. 4.2, a number of rules due to which more vertices will be coloured either red or blue. So this leads to a partial red–blue colouring of G . We prove that the rules are safe, that is, each of the coloured vertices received their correct colour (assuming G has a perfect red–blue colouring that coincides with our original guess). We then show that the connected components of the subgraph of G induced by the uncoloured vertices must be monochromatic in every perfect red–blue colouring extension of the partial red–blue colouring of G . This allows us to apply a number of new rules given in Sect. 4.3 that exploit this property. Afterwards, more vertices will be coloured, and we show in Sect. 4.4 that we can check in polynomial time (by

a reduction to 2-SAT) whether the partial red–blue colouring can be extended to a perfect red–blue colouring of the whole graph G .

4.1 Small or Monochromatic Dominating Sets

We start with two lemmas. Note that the first lemma shows that PERFECT MATCHING CUT is in XP when parameterized by the domination number of a graph. The proofs of the two lemmas are similar to the proofs for valid but not necessarily perfect red–blue colourings; see, for example, [15] or [31].

Lemma 12 *For every integer g , it is possible to find in $O(2^g n^{g+2})$ -time a perfect red–blue colouring (if it exists) of a graph with n vertices and with domination number g .*

Proof Let $g \geq 1$ be an integer, and let G be a graph with domination number at most g . Hence, G has a dominating set D of size at most g . We consider all options of colouring the vertices of D red or blue; note that this number is $2^{|D|} \leq 2^g$. For every red vertex of D with no blue neighbour, we consider all $O(n)$ options of colouring exactly one of its neighbours blue (and thus, all of its other neighbours will be coloured red). Similarly, for every blue vertex of D with no red neighbour, we consider all $O(n)$ options of colouring exactly one of its neighbours red (and thus, all of its other neighbours will be coloured blue). Finally, for every red vertex in D with already one blue neighbour in D , we colour all its yet uncoloured neighbours red. Similarly, for every blue vertex in D with already one red neighbour in D , we colour all its yet uncoloured neighbours blue.

As D is a dominating set, the above means that we guessed a red–blue colouring of the whole graph G . We can check in $O(n^2)$ time if a red–blue colouring of a graph with n vertices is perfect. Moreover, the total number of red–blue colourings that we must consider is $O(2^g n^g)$. \square

Lemma 13 *Let D be a dominating set of a connected graph G . It is possible to check in polynomial time if G has a perfect red–blue colouring in which D is monochromatic.*

Proof Consider a perfect red–blue colouring c of G , in which D is monochromatic, say every vertex of D is coloured red. Let F_1, \dots, F_r , for some integer $r \geq 1$, be the connected components of $G - D$.

We claim that every F_i is monochromatic. This follows from the same argument as the one for valid red–blue colourings that are not necessarily perfect (see [31]) and we give it for completeness. For a contradiction, assume that, say, F_1 is not monochromatic. This means that F_1 contains an edge uv where u is coloured red and v is coloured blue. As D is a dominating set of G , we have that v also has a neighbour in D , which is coloured red. Hence, c is not valid and thus not perfect either, a contradiction.

Now suppose that some vertex u in F_i is coloured red. By the above claim, every vertex of F_i is coloured red. The neighbours of u outside F_i are all in D and thus, are coloured red as well. Hence, u has no blue neighbour, meaning that c is not perfect, a contradiction. We conclude that every vertex in $G - D$ must be coloured blue.

Hence, in order to check if G has a perfect red–blue colouring in which D is monochromatic, we can do as follows. Colour every vertex in D red and colour every vertex in $G - D$ blue. Then, check in polynomial time if the resulting red–blue colouring is perfect. \square

4.2 Partial Red–Blue Colourings: Applying General Rules

To handle “partial” red–blue colourings that we want to extend to perfect red–blue colourings, we introduce the following terminology. Let $G = (V, E)$ be a connected graph and $S, T, X, Y \subseteq V$ be four non-empty sets with $S \subseteq X, T \subseteq Y$ and $X \cap Y = \emptyset$. A *red–blue* (S, T, X, Y) -colouring of G is a red–blue colouring where

- Every vertex of X is coloured red and every vertex of Y is coloured blue;
- The blue neighbour of every vertex in S belongs to T and vice versa; and
- The blue neighbour of every vertex in $X \setminus S$ and the red neighbour of every vertex of Y belong to $V \setminus (X \cup Y)$.

For a connected graph $G = (V, E)$, let S' and T' be two disjoint subsets of V , such that (i) every vertex of S' is adjacent to at most one vertex of T' , and vice versa, and (ii) at least one vertex in S' is adjacent to a vertex in T' . Let S'' consist of all vertices of S' with a (unique) neighbour in T' , and let T'' consist of all vertices of T' with a (unique) neighbour in S' (so, every vertex in S'' has a unique neighbour in T'' , and vice versa). We call (S'', T'') the *core* of starting pair (S', T') ; note that $|S''| = |T''| \geq 1$.

We colour every vertex in S' red and every vertex in T' blue. Propagation rules will try to extend S' to a set X and T' to a set Y by finding new vertices whose colour must always be either red or blue. That is, we place new red vertices in the set X , which already contains S' and new blue vertices in the set Y , which already contains T' . If a red and blue vertex are matched to each other, then we add the red one to a set $S \subseteq X$ and the blue one to a set $T \subseteq Y$. So initially, $S := S'', T := T'', X := S'$ and $Y := T'$. We let $Z := V \setminus (X \cup Y)$.

We now present seven propagation rules for finding perfect red–blue (S, T, X, Y) -colourings. Rules R1 and R2 hold for finding red–blue colourings in general and correspond to the five rules from [27]. Rules R3–R7 are for finding perfect red–blue colourings; some of them are in a slightly different form in [29].

R1. Return no (i.e., G has no red–blue (S, T, X, Y) -colouring) if a vertex $v \in Z$ is

- (i) Adjacent to a vertex in S and to a vertex in T , or
- (ii) Adjacent to a vertex in S and to two vertices in $Y \setminus T$, or
- (iii) Adjacent to a vertex in T and to two vertices in $X \setminus S$, or
- (iv) Adjacent to two vertices in $X \setminus S$ and to two vertices in $Y \setminus T$.

R2. Let $v \in Z$.

- (i) If v is adjacent to a vertex in S or to two vertices of $X \setminus S$, then move v from Z to X . If moreover v is also adjacent to a vertex w in Y , then add v to S and w to T .

- (ii) If v is adjacent to a vertex in T or to two vertices of $Y \setminus T$, then move v from Z to Y . If moreover v is also adjacent to a vertex w in X , then add v to T and w to S .

R3. Let $v \in (X \cup Y) \setminus (S \cup T)$.

- (i) If $v \in X \setminus S$ and v is adjacent to a vertex w in Y , then add v to S and w to T .
- (ii) If $v \in Y \setminus T$ and v is adjacent to a vertex w in X , then add v to T and w to S .

R4. Return `no` if

- (i) a vertex $x \in X$ has no neighbours outside X or is adjacent to two vertices of Y , or
- (ii) a vertex $y \in Y$ has no neighbours outside Y , or is adjacent to two vertices of X .

R5. Let $v \in Z$, and let $w \in Z$ be a vertex with $N_G(w) = \{v\}$.

- (i) If v is adjacent to a vertex in X and to a vertex in Y , then return `no`.
- (ii) If v is adjacent to a vertex in X but not to a vertex in Y , then put v in X and w in Y , and also add v to S and w to T .
- (iii) If v is adjacent to a vertex in Y but not to a vertex in X , then put v in Y and w in X , and also add v to T and w to S .

R6. Let $v \in Z$ be in a connected component F of $G[Z]$ such that F contains C_4 as a spanning subgraph.

- (i) If v is adjacent to a vertex in X but not to a vertex in Y , and F contains a vertex not adjacent to a vertex in X , then move v from Z to X .
- (ii) If v is adjacent to a vertex in Y but not to a vertex in X , and F contains a vertex not adjacent to a vertex in Y , then move v from Z to Y .

R7. Let $v \in Z$ be in a connected component F of $G[Z]$ such that $\{v\}$ dominates F . Let $F - v$ have a vertex w with only one neighbour w' in $X \cup Y$.

- (i) If $w' \in X$, then put v in Y .
- (ii) If $w' \in Y$, then put v in X .

A propagation rule is *safe* if the input graph has a perfect red–blue (S, T, X, Y) -colouring before the application of the rule if and only if it has so after the application of the rule.

Lemma 14 *Rules R1–R7 are safe.*

Proof Let G be a connected graph with a perfect red–blue (S, T, X, Y) -colouring. First recall that, by definition, vertices in X will be coloured red by every red–blue (S, T, X, Y) -colouring, whilst vertices of Y will be coloured blue, and moreover that every (red) vertex in S has exactly one (blue) neighbour in T , and vice versa. The colour of the vertices in Z still has to be decided.

Rule R1-(i) is safe. A vertex adjacent to both a red vertex that already has a blue neighbour and to a blue vertex that already has a red neighbour can be coloured neither

red nor blue. Rule R1-(ii) is safe, as a vertex that is adjacent to a red vertex that already has a blue neighbour must be coloured red, so it cannot also be adjacent to two blue vertices. For the same reason, R1-(iii) is safe. Finally, R1-(iv) is safe, as a vertex that is adjacent to two red vertices must be coloured red, so it cannot also be adjacent to two blue vertices.

Rule R2-(i) is safe. Any vertex adjacent to a red vertex that already has a blue neighbour or to two red vertices must be coloured red. If such a vertex is adjacent to a vertex coloured blue already, it will have its blue neighbour and thus must be added to S , whilst its blue neighbour must be added to T . For the same reason, R2-(ii) is safe as well.

Rule R3-(i) is safe. Every red vertex must have a (unique) blue neighbour, and vice versa. For the same reason, R3-(ii) is safe.

Rule R4-(i) is safe. In the first case, x will only have red neighbours in G (as x is coloured red, x needs a blue neighbour as well). In the second case, x will have two blue neighbours, while x is coloured red. This is not possible. For the same reason, R4-(ii) is safe as well.

Rule R5-(i) is safe. As v will be adjacent to a blue and red neighbour, w cannot be its matching neighbour. As w has degree 1, we find that w cannot be matched. Rule R5-(ii) is safe as well. For a contradiction, suppose that we would put v in Y . Then the matched neighbour of v is the neighbour of v that belongs to x . Hence, again we find that w does not have a matched neighbour. So we must put v in X , and then v and w will be matched to each other. For the same reason, R5-(iii) is safe.

Rule R6-(i) is safe. Suppose that we put v in Y , that is, v will be coloured blue. Consequently, v has its matching neighbour in X . This means that the neighbours of v in F will be coloured blue. As F has a cycle on four vertices as a spanning subgraph, v has at most one non-neighbour in F , and this non-neighbour will get colour blue as well. By assumption, F contains a vertex that is not adjacent to a vertex in X . This vertex is coloured blue, but will not have a red neighbour. We conclude that v must be put in X . For the same reason, R6-(ii) is safe as well.

Rule R7-(i) is safe. Suppose that we put v in X , so v will be coloured red, just like w' . Hence, w is adjacent to two red vertices, and must be coloured red as well. As w' is the only neighbour of w in $X \cup Y$, we find that every other neighbour of w is in F . As $\{v\}$ dominates F , this means that such a neighbour of w is adjacent not only to w but also to v , and hence must be coloured red. This means that w will not have any blue neighbour. We conclude that v must be put in Y . For the same reason, R7-(ii) is safe as well. \square

Assume that exhaustively applying rules R1–R7 on a starting pair (S', T') did not lead to a no-answer but to a 4-tuple (S, T, X, Y) . Then we call (S, T, X, Y) an *intermediate* 4-tuple. We prove the following lemma.

Lemma 15 *Let G be a connected graph with a starting pair (S', T') with core (S'', T'') and a resulting intermediate 4-tuple (S, T, X, Y) . Then G has a perfect red–blue (S'', T'', S', T') -colouring if and only if G has a perfect red–blue (S, T, X, Y) -colouring. Moreover, (S, T, X, Y) can be obtained in polynomial time.*

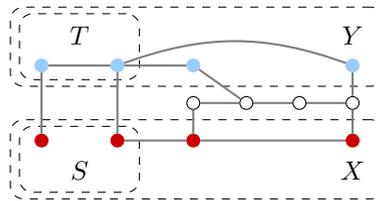


Fig. 11 A red–blue (S, T, X, Y) -colouring of a graph with an intermediate 4-tuple (S, T, X, Y) . In this example, $G[Z]$ consists of a single connected component isomorphic to P_4 (Color figure online)

Proof The first part of the lemma follows from Lemma 14 and our initialisation. To prove the running time statement, we first note that each application of R1–R7 takes polynomial time. For each rule we can also check in polynomial time if it can be applied. Moreover, after each application of a rule we either find a no-answer or reduce the size of at least one of the sets X, Y, Z . Hence, we obtained (S, T, X, Y) in polynomial time. \square

We now describe the structure of a graph with an intermediate 4-tuple (S, T, X, Y) ; see Fig. 11 for an example.

Lemma 16 *Let G be a connected graph with an intermediate 4-tuple (S, T, X, Y) . Then:*

- (i) *Every vertex in S has exactly one neighbour in Y and that neighbour belongs to T ;*
- (ii) *Every vertex in T has exactly one neighbour in X and that neighbour belongs to S ;*
- (iii) *Every vertex in $X \setminus S$ has no neighbour in Y ;*
- (iv) *Every vertex in $Y \setminus T$ has no neighbour in X ;*
- (v) *Every vertex in $V \setminus (X \cup Y)$ has no neighbour in $S \cup T$, at most one neighbour in $X \setminus S$ and at most one neighbour in $Y \setminus T$.*

Proof Let $Z = V \setminus (X \cup Y)$. We prove the five statements one by one.

Proof of (i). For a contradiction, first assume that some vertex u in S has no neighbour in T . Then u has no neighbour in $Y \setminus T$ either, else we would have applied R3. However, now we would have applied R4 (and returned a no-answer). Hence, every vertex in S has a neighbour in $T \subseteq Y$. If a vertex in S has more than one neighbour in Y , then we would have applied R4 as well.

Proof of (ii). Statement (ii) follows by symmetry: we can use the same arguments as in the proof of (i).

Proof of (iii). Let $u \in X \setminus S$. If u has a neighbour in Y , then we would have applied R3. Hence, u has no neighbours in Y .

Proof of (iv). Statement (iv) follows by symmetry: we can use the same arguments as in the proof of (iii).

Proof of (v). Let $u \in Z$. If u is adjacent to a vertex in $S \cup T$, then we would have applied R2. If u is adjacent to two vertices in $X \setminus S$ or to two vertices in $Y \setminus T$, then we would also have applied R2 as well. \square

4.3 Partial Red–Blue Colourings: Exploiting Monochromaticity

Let (S, T, X, Y) be an intermediate 4-tuple of a connected graph G . Let $Z = V \setminus (X \cup Y)$. A red–blue (S, T, X, Y) -colouring of G is *monochromatic* if all connected components of $G[Z]$ are monochromatic. Rules R8–R11 preserve this property; some of them were also used in [27, 29].

- R8.** Let $v \in Z$. If v is not adjacent to any vertex of $X \cup Y$, then return no .
- R9.** Let $v \in Z$ be a vertex in a connected component F of $G[Z]$ such that v has only one neighbour w in $X \cup Y$.
- (i) If $w \in X$, then put every vertex of F in Y and also add every vertex of F to T and every neighbour of every vertex of F in X to S .
 - (ii) If $w \in Y$, then put every vertex of F in X and also add every vertex of F to S and every neighbour of every vertex of F in Y to T .
- R10.** Let $v \in (X \cup Y) \setminus (S \cup T)$ and F be a connected component of $G[Z]$ such that v has two neighbours in F .
- (i) If $v \in X \setminus S$, then put every vertex of F in X , and also add every vertex of F to S and every vertex of every neighbour of F in $Y \setminus T$ to T .
 - (ii) If $v \in Y \setminus T$, then put every vertex of F in Y , and also add every vertex of F to T and every vertex of every neighbour of F in $X \setminus S$ to S .
- R11.** Let $v \in (X \cup Y) \setminus (S \cup T)$ and F be a connected component of $G[Z]$ such that v has one neighbour in F that is the only neighbour of v in Z .
- (i) If $v \in X \setminus S$ and v is not adjacent to Y , then put every vertex of F in Y , and also add every vertex of F to T and every vertex of every neighbour of F in $X \setminus S$ to S .
 - (ii) If $v \in Y \setminus T$ and v is not adjacent to X , then put every vertex of F in X , and also add every vertex of F to S and every vertex of every neighbour of F in $Y \setminus T$ to T .

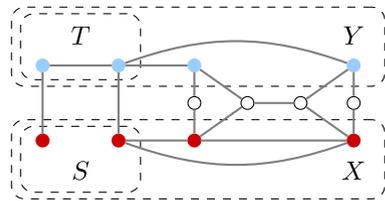
A propagation rule is *mono-safe* if the input graph has a (monochromatic) perfect red–blue (S, T, X, Y) -colouring before the application of the rule if and only if it has so after the application of the rule. We prove the following lemma.

Lemma 17 *Rules R8–R11 are mono-safe.*

Proof Let G be a connected graph with a monochromatic perfect red–blue (S, T, X, Y) -colouring. First recall that, by definition, vertices in X will be coloured red by every red–blue (S, T, X, Y) -colouring, whilst vertices of Y will be coloured blue, and moreover that every (red) vertex in S has exactly one (blue) neighbour in T , and vice versa. The colour of the vertices in Z still has to be decided.

Rule R8 is mono-safe. Let F be the connected component of $G[Z]$ that contains v . Then all neighbours of v belong to F , which must be monochromatic. Thus the neighbours of v are coloured either all red or all blue. Hence, v will not have the required neighbour with a different colour than itself.

Fig. 12 A red–blue (S, T, X, Y) -colouring of a graph with a final 4-tuple (S, T, X, Y) . In this example, $G[Z]$ is isomorphic to $2P_1 + P_2$ (Color figure online)



Rule R9-(i) is mono-safe. As all vertices in F will be coloured with the same colour, this means that w must receive a different colour than v . Hence, as w is coloured red, we find that v , and thus all other vertices of F , must be coloured blue. As every vertex of F only has neighbours in F and in $X \cup Y$, we find that all neighbours of every vertex in F are coloured. Hence, we can identify the unique red neighbours of the vertices of F , which in turn will be the unique blue neighbours of these vertices. For the same reason, Rule R9-(ii) is mono-safe as well.

Rule R10-(i) is mono-safe. All vertices in F will be coloured with the same colour and at least two of them are adjacent to v . Hence, the vertices in F must all get the same colour as the colour of v , which is red. Just as in the previous rule, we can now identify the unique blue neighbours of the vertices of F , which in turn will be the unique red neighbours of these vertices. For the same reason, Rule R10-(ii) is mono-safe as well.

Rule R11-(i) is mono-safe. Let w be the neighbour of v in F . Assume that $v \in X \setminus S$ and v is not adjacent to Y . As w is the only neighbour of v in Z , all other neighbours of v belong to X . As all vertices in X are coloured red, this means that w must be coloured blue. Hence, as every vertex of F will be coloured the same, every vertex of F will be coloured blue. Just as in the previous two rules, we can now identify the unique red neighbours of the vertices of F , which in turn will be the unique blue neighbours of these vertices. For the same reason, Rule R11-(ii) is mono-safe. \square

Suppose that exhaustively applying rules R1–R11 on an intermediate 4-tuple (S, T, X, Y) did not lead to a no-answer but to a 4-tuple (S^*, T^*, X^*, Y^*) . We call (S^*, T^*, X^*, Y^*) the *final* 4-tuple and prove the following lemma.

Lemma 18 *Let G be a connected graph with an intermediate 4-tuple (S, T, X, Y) and a resulting final 4-tuple (S^*, T^*, X^*, Y^*) . Then G has a monochromatic perfect red–blue (S, T, X, Y) -colouring if and only if G has a monochromatic perfect red–blue (S^*, T^*, X^*, Y^*) -colouring. Moreover, (S^*, T^*, X^*, Y^*) can be obtained in polynomial time.*

Proof The first part of the lemma follows from Lemma 17 and the fact that (S^*, T^*, X^*, Y^*) results from (S, T, X, Y) . To prove the running time statement, we first note that each application of R1–R11 takes polynomial time. For each rule we can also check in polynomial time if it can be applied. Moreover, after each application of a rule we either find a no-answer or reduce the size of at least one of the sets X, Y, Z . Hence, we obtained (S, T, X, Y) in polynomial time. \square

We now describe the structure of a connected graph with a final 4-tuple (S, T, X, Y) ; see Fig. 12 for an example.

Lemma 19 *Let G be a connected graph with a final 4-tuple (S, T, X, Y) . The following holds:*

- (i) *Every vertex in S has exactly one neighbour in Y , which belongs to T ;*
- (ii) *Every vertex in T has exactly one neighbour in X , which belongs to S ;*
- (iii) *Every vertex in $X \setminus S$ has no neighbour in Y , at least two neighbours in $V \setminus (X \cup Y)$ but no two neighbours in the same connected component of $G[V \setminus (X \cup Y)]$;*
- (iv) *Every vertex in $Y \setminus T$ has no neighbour in X , at least two neighbours in $V \setminus (X \cup Y)$ but no two neighbours in the same connected component of $G[V \setminus (X \cup Y)]$;*
- (v) *Every vertex of $V \setminus (X \cup Y)$ has no neighbour in $S \cup T$, exactly one neighbour in $X \setminus S$ and exactly one neighbour in $Y \setminus T$.*

Proof Let $Z = V \setminus (X \cup Y)$. We prove the five statements one by one.

Proof of (i). For a contradiction, first assume that some vertex u in S has no neighbour in T . Then u has no neighbour in $Y \setminus T$ either, else we would have applied R3. However, now we would have applied R4 (and returned a no-answer). Hence, every vertex in S has a neighbour in $T \subseteq Y$. If a vertex in S has more than one neighbour in Y , then we would have applied R4 as well.

Proof of (ii). Statement (ii) follows by symmetry: we can use the same arguments as in the proof of (i).

Proof of (iii). Let $u \in X \setminus S$. If u has a neighbour in Y , then we would have applied R3. Hence, u has no neighbours in Y . If u has no neighbours in $V \setminus (X \cup Y)$, then u would only have neighbours in X . In that case we would have applied R4 (and returned a no-answer). If u only has one neighbour in Z , then we would have applied R11. Hence, u has at least two neighbours in Z . If two neighbours of u in Z belong to the same connected component of Z , then we would have applied R10.

Proof of (iv). Statement (iv) follows by symmetry: we can use the same arguments as in the proof of (iii).

Proof of (v). Let $u \in Z$. If u is adjacent to a vertex in $S \cup T$, then we would have applied R2. Hence, we find that u is not adjacent to a vertex in $S \cup T$. If u is adjacent to two vertices in $X \setminus S$ or to two vertices in $Y \setminus T$, then we would have applied R2 as well. If u has exactly one neighbour in $X \cup Y$, then we would have applied R9. If u has no neighbour in $X \setminus S$ and no neighbour in $Y \setminus T$, then u has no neighbour in $X \cup Y$, as we already deduced that u has no neighbour in $S \cup T$. However, then we would have applied R8 (and returned a no-answer). □

4.4 Reduction to 2-SAT

We now prove a lemma that is the cornerstone for our polynomial-time results.

Lemma 20 *Let G be a connected graph with a final 4-tuple (S, T, X, Y) . Then it is possible to find in polynomial time a monochromatic perfect red–blue (S, T, X, Y) -colouring of G or conclude that such a colouring does not exist.*

Proof Let $Z = V \setminus (X \cup Y)$. Let $E^* \subseteq E$ be the set of edges consisting of all edges with one end-vertex in $(X \cup Y) \setminus (S \cup T)$ and the other end-vertex in Z . By Lemma 19-(v), we find that $|E^*| = 2|Z|$. By Lemma 19-(iii) and (iv), we find that $|E^*| \geq$

$2|(X \cup Y) \setminus (S \cup T)|$. Hence, $|Z| \geq |(X \cup Y) \setminus (S \cup T)|$, and $|Z| = |(X \cup Y) \setminus (S \cup T)|$ if and only if each vertex in $(X \cup Y) \setminus (S \cup T)$ has exactly two neighbours in Z .

Every vertex $u \in Z$ still needs their matching neighbour v . In order for G to have a monochromatic perfect red–blue (S, T, X, Y) -colouring, v must be outside $S \cup T$, so v belongs to $X \cup Y$. By Lemma 19-(v), we find that $v \in (X \cup Y) \setminus (S \cup T)$. As matching neighbours are “private”, $|Z| \leq |(X \cup Y) \setminus (S \cup T)|$. We conclude that $|(X \cup Y) \setminus (S \cup T)| = |Z|$. Our algorithm checks this in polynomial time and returns a no-answer if $|(X \cup Y) \setminus (S \cup T)| \neq |Z|$.

From now on, assume $|(X \cup Y) \setminus (S \cup T)| = |Z|$. Hence, each vertex in $(X \cup Y) \setminus (S \cup T)$ has exactly two neighbours in Z . Just like [27], we now construct an instance ϕ of the 2- SATISFIABILITY problem (2- SAT). Our 2-SAT formula differs from the one in [27] due to the perfectness requirement. For each connected component C of $G[Z]$, we do as follows. We define two variables x_C and y_C , and we add the clause $(x_C \vee y_C) \wedge (\neg x_C \vee \neg y_C)$ to ϕ . For each $u \in (X \cup Y) \setminus (S \cup T)$, we do as follows. From the above we know that u has exactly two neighbours v and w in Z . Let C be the connected component of $G[Z]$ that contains v and D be the connected component of $G[Z]$ that contains w . We add the clause $(x_C \vee x_D) \wedge (y_C \vee y_D)$ to ϕ . This finishes the construction of ϕ .

We claim that G has a monochromatic perfect red–blue (S, T, X, Y) -colouring if and only if ϕ has a satisfying truth assignment. It is readily seen and well known that 2- SAT is polynomial-time solvable, meaning we are done once we have proven this claim.

First suppose that G has a monochromatic perfect red–blue (S, T, X, Y) -colouring c . By definition, the vertices in each connected component C of $G[Z]$ are coloured alike. We define a truth assignment τ as follows. We let x_C be true if and only if the vertices of C are coloured red. We let y_C be true if and only if the vertices of C are coloured blue. As exactly one of these options holds, the clause $(x_C \vee y_C) \wedge (\neg x_C \vee \neg y_C)$ is satisfied.

Now consider a clause $(x_C \vee x_D) \wedge (y_C \vee y_D)$ corresponding to a vertex $u \in (X \cup Y) \setminus (S \cup T)$ that has a neighbour in each of the connected components C and D of $G[Z]$. Then, by Lemma 19-(iii) and (iv), C and D are different connected components of $G[Z]$. First assume that $u \in X \setminus S$. By Lemma 19-(iii), we find that u has no neighbour in Y and thus its blue neighbour must either be in C or in D . If it is in C , then the neighbour of u in D is coloured blue, and vice versa. As c is monochromatic, this means that either all vertices of C are coloured red and all vertices of D are coloured blue, or the other way around. Hence, the clause $(x_C \vee x_D) \wedge (y_C \vee y_D)$ is satisfied. If $u \in Y \setminus T$, we can use exactly the same arguments. We conclude that τ is a satisfying truth assignment.

Now suppose that ϕ has a satisfying truth assignment τ . For every connected component C of $G[Z]$, we colour the vertices of C red if x_C is true and we colour the vertices of C blue if y_C is true. As τ satisfies $(x_C \vee y_C) \wedge (\neg x_C \vee \neg y_C)$, exactly one of x_C or y_C is true. Hence, the colouring of the vertices of Z is well defined.

We also colour all vertices of X red and all vertices of Y blue. We let c be the resulting colouring. By construction, it is monochromatic. Hence, it remains to show that c is a perfect red–blue (S, T, X, Y) -colouring. We will do this below.

First, it follows from the definition of a core (S'', T'') that S'' and T'' are non-empty. Moreover, before applying the reduction rules, we first do an initiation, from which it follows that $S'' \subseteq S$ and $T'' \subseteq T$. Hence, at least one vertex of G is coloured red and at least one vertex of G is coloured blue.

By Lemma 19-(i), every vertex in S has exactly one neighbour in Y . By Lemma 19-(ii), every vertex in T has exactly one neighbour in X . By Lemma 19-(v), no vertex of $S \cup T$ is adjacent to a vertex of Z . Hence, the vertices in $S \cup T$ have exactly one neighbour of opposite colour.

By Lemma 19-(v), every vertex $z \in Z$ has exactly one neighbour in $X \setminus S$, which is coloured red, and exactly one neighbour in $Y \setminus T$, which is coloured blue; moreover, z is not adjacent to any vertex in $S \cup T$. Let C be the connected component of $G[Z]$ that contains z . As c is monochromatic, all vertices of C receive the same colour. Hence, the vertices in Z have each exactly one neighbour of opposite colour.

Finally, we must verify the vertices in $(X \cup Y) \setminus (S \cup T)$. Let $u \in (X \cup Y) \setminus (S \cup T)$. First assume that $u \in X \setminus S$, so u is coloured red. We recall that u has exactly two neighbours v and w in Z . Let C be the connected component of $G[Z]$ that contains u , and let D be the connected component of $G[Z]$ that contains w . Hence, τ contains the clause $(x_C \vee x_D) \wedge (y_C \vee y_D)$. By Lemma 19-(iii), we find that C and D are two distinct connected components of $G[Z]$. As τ satisfies $(x_C \vee x_D) \wedge (y_C \vee y_D)$, the vertices of one of C, D are coloured red, while the vertices of the other one are coloured blue. By Lemma 19-(iii), we find that u has no (blue) neighbour in Y . Hence, u has exactly one blue neighbour. If $u \in Y \setminus T$, we can apply the same arguments. We conclude that also the vertices in $(X \cup Y) \setminus (S \cup T)$ have exactly one neighbour of the opposite colour.

From the above we conclude that c is monochromatic and perfect. □

4.5 Applications of Our Approach

We first apply the approach described in the previous subsections to graphs of radius at most 2. Our proof is similar but more involved than the one for MATCHING CUT on graphs of radius at most 2 [31].

Theorem 21 PERFECT MATCHING CUT is polynomial-time solvable for graphs of radius at most 2.

Proof Let G be a graph of radius r at most 2. If $r = 1$, then G has a vertex that is adjacent to all other vertices. In this case G has a perfect matching cut if and only if G consists of two vertices with an edge between them. From now on, assume that $r = 2$. Then G has a dominating star H as a subgraph, say H has centre u and leaves v_1, \dots, v_s for some $s \geq 1$. We can find H in polynomial time, as finding a dominating star in a graph is equivalent to finding a vertex u with the property that every non-neighbour of u is adjacent to a neighbour of u . By Observation 5 it suffices to check if G has a perfect red–blue colouring.

We first check if G has a perfect red–blue colouring in which $V(H)$ is monochromatic. By Lemma 13 this can be done in polynomial time. Suppose we find no such red–blue colouring. Then we may assume without loss of generality that a perfect

red–blue colouring of G (if it exists) colours u red and exactly one of v_1, \dots, v_s blue. That is, G has a perfect red–blue colouring if and only if G has a perfect red–blue $(\{u\}, \{v_i\}, \{u\}, \{v_i\})$ -colouring for some $i \in \{1, \dots, s\}$. We consider all $O(n)$ options of choosing which v_i is coloured blue.

For each option we do as follows. Let v_i be the vertex of v_1, \dots, v_s that we coloured blue. We define the starting pair (S', T') with core (S', T') , where $S' = \{u\}$ and $T' = \{v_i\}$. We now apply rules R1–R7 exhaustively. The latter takes polynomial time by Lemma 15. If this exhaustive application leads to a no-answer, then by Lemma 15 we may discard the option. Suppose we obtain an intermediate 4-tuple (S, T, X, Y) . By again applying Lemma 15, we find that G has a perfect red–blue $(\{u\}, \{v_i\}, \{u\}, \{v_i\})$ -colouring if and only if G has a perfect red–blue (S, T, X, Y) -colouring. By R2-(i) and the fact that $u \in S' \subseteq S$ we find that $\{v_1, \dots, v_s\} \setminus \{v_i\}$ belongs to X .

Suppose that G has a perfect red–blue (S, T, X, Y) -colouring c such that $G[V(G) \setminus (X \cup Y)]$ has a connected component D that is not monochromatic. Then D must contain an edge uv , where u is coloured red and v is coloured blue. Note that v cannot be adjacent to v_i , as otherwise v would have been in Y by R3 (since $v_i \in T' \subseteq T$). As H is dominating, this means that v must be adjacent to a vertex $w \in V(H) \setminus \{v_i\} = \{u, v_1, \dots, v_s\} \setminus \{v_i\}$. As $u \in S' \subseteq S \subseteq X$ and $\{v_1, \dots, v_s\} \setminus \{v_i\} \subseteq X$, we find that $w \in X$ by R2-(i) and thus will be coloured red. However, now v being coloured blue is adjacent to two red vertices (namely u and w), contradicting the validity of c .

From the above we conclude that every perfect red–blue (S, T, X, Y) -colouring of G is monochromatic. We now apply rules R1–R11 exhaustively. The latter takes polynomial time by Lemma 18. If this exhaustive application leads to a no-answer, then by Lemma 18 we may discard the option. Suppose we obtain a final 4-tuple (S^*, T^*, X^*, Y^*) . By again applying Lemma 18, we find that G has a monochromatic perfect red–blue (S, T, X, Y) -colouring if and only if G has a monochromatic perfect red–blue (S^*, T^*, X^*, Y^*) -colouring. We can now apply Lemma 20 to find in polynomial time whether or not G has a monochromatic perfect red–blue (S^*, T^*, X^*, Y^*) -colouring. The correctness of our algorithm follows from the above arguments. As we branch $O(n)$ times and each branch takes polynomial time to process, the total running time of our algorithm is polynomial. \square

We now consider P_6 -free graphs. As a consequence of Theorem 4, a P_6 -free graph either has a small domination number, in which case we use Lemma 12, a monochromatic dominating set, in which case we use Lemma 13, or it has radius at most 2, in which case we use Theorem 21.

Theorem 22 PERFECT MATCHING CUT *is polynomial-time solvable for P_6 -free graphs.*

Proof Let G be a connected P_6 -free graph. By Theorem 4, we find that G has a dominating induced C_6 or a dominating (not necessarily induced) complete bipartite graph $K_{r,s}$. By Observation 5 it suffices to check if G has a perfect red–blue colouring.

If G has a dominating induced C_6 , then G has domination number at most 6. In that case we apply Lemma 12 to find in polynomial time if G has a perfect red–blue colouring. Suppose that G has a dominating complete bipartite graph D with partition classes $\{u_1, \dots, u_r\}$ and $\{v_1, \dots, v_s\}$. We may assume without loss of generality that $r \leq s$.

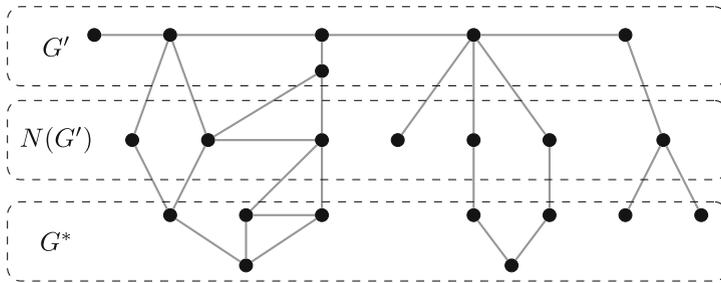


Fig. 13 The decomposition of a graph G into the graphs G' , which is isomorphic to H , the neighbourhood graph $N(G')$ of G' , which is induced by all vertices not in G' but that have one or more neighbours in G' , and the graph G^* induced by the remaining vertices of G

If $r \geq 2$ and $s \geq 3$, then any starting pair of the form $(\{u_i\}, \{v_j\})$, $(\{u_i\}, \{u_j\})$ (if u_i and u_j are adjacent) or $(\{v_i\}, \{v_j\})$ (if v_i and v_j are adjacent) yields a no-answer. Hence, $V(D)$ is monochromatic for any perfect red–blue colouring of G . This means that we can check in polynomial time by Lemma 13 if G has a perfect red–blue colouring.

Now assume that $r = 1$ or $s \leq 2$. In the first case, G has a (not necessarily induced) dominating star and thus G has radius at most 2, and we apply Theorem 21. In the second case, $r \leq s \leq 2$, and thus G has domination number at most 4, and we apply Lemma 12. Hence, in both cases, we find in polynomial time whether or not G has a perfect red–blue colouring. \square

For our last result we again apply our approach.

Theorem 23 *Let H be a graph. If PERFECT MATCHING CUT is polynomial-time solvable for H -free graphs, then it is so for $(H + P_4)$ -free graphs.*

Proof Assume that PERFECT MATCHING CUT can be solved in polynomial time for H -free graphs. Let G be a connected $(H + P_4)$ -free graph. Say, G has an induced subgraph G' that is isomorphic to H ; else we are done by our assumption. Let G^* be the graph obtained from G after removing every vertex that belongs to G' or that has a neighbour in G' . As G' is isomorphic to H and G is $(H + P_4)$ -free, G^* is P_4 -free. See Fig. 13 for an example of this decomposition of G , where we have chosen $H = S_{1,2,2}$.

We use Observation 5-(iii) and search for a perfect red–blue colouring. We define $n = |V(G)|$ and $m = |E(G)|$. Following our approach, we need a starting pair (S', T') with core (S'', T'') . By definition, $|S''| = |T''| \geq 1$. Hence, we consider all $O(m)$ options of choosing an edge uv from $E(G)$, one of whose end-vertices we colour red, say u (so $u \in S''$) and the other one, v , blue (so $v \in T''$). Afterwards, for each (uncoloured) vertex in G' we consider all options of colouring it either red or blue. As G' is isomorphic to H , the number of distinct options is a constant, namely $2^{|V(H)|}$. Now, for every red (blue) vertex of G' with no blue (red) neighbour, we consider all $O(n)$ options of colouring exactly one of its neighbours blue (red). Hence, afterwards each vertex of $V(G') \cup N(V(G'))$ is either coloured red or blue. This leads to $O(m2^{|V(H)|}n^{|V(H)|})$ options (branches), which we handle one by one.

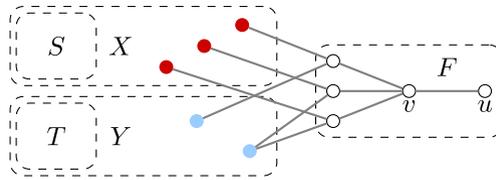


Fig. 14 An example of a reducible connected component F of a graph G with an intermediate tuple (S, T, X, Y) . For readability only edges with at least one end-vertex in F are drawn. Note that $F - \{u, v\}$ consists of three single-vertex components and that $F - \{u, v\}$ becomes a K_3 after our algorithm has processed F

Consider an option as described above. Let S' consist of u and all red vertices of $V(G') \cup N(V(G'))$, and let T' consist of v and all blue vertices of $V(G') \cup N(V(G'))$. In this way we obtain a starting pair (S', T') with core (S'', T'') . We apply rules R1-R7 exhaustively. If we find a no-answer, then we can discard the option by Lemma 15. Else we found in polynomial time an intermediate 4-tuple (S, T, X, Y) , such that G has a perfect red–blue (S'', T'', S', T') -colouring if and only if G has a perfect red–blue (S, T, X, Y) -colouring.

Consider a connected component F of $G - (X \cup Y)$, for which the following holds:

1. F contains two adjacent vertices u and v , each with no neighbours in $X \cup Y$ and moreover, v is dominating F ; and
2. Every vertex in $F - \{u, v\}$ has both a neighbour in X and a neighbour in Y .

We say that F is a *reducible* connected component of $G - (X \cup Y)$, as after processing F in the way described below we either found that G has no perfect red–blue (S, T, X, Y) -colouring, or we have reduced the size of G . See Fig. 14 for an example.

We will now describe how we process a reducible component F . As G is connected, the fact that u and v have no neighbours in $X \cup Y$ implies that $F - \{u, v\}$ is non-empty. As every vertex in $F - \{u, v\}$ has both a neighbour in X and a neighbour in Y , their matching neighbour is not in F in every perfect red–blue (S, T, X, Y) -colouring of G . As v dominates F , all vertices of $F - \{u, v\}$ are adjacent to v . Hence, we find that in every perfect red–blue (S, T, X, Y) -colouring of G , all vertices of $F - \{u, v\}$ have the same colour as v , that is, all vertices of $F - \{u\}$ are coloured alike. If u is adjacent to a vertex in $F - \{u, v\}$, this means that u will receive the same colour as every other vertex in F including v and hence, u will not have a matching neighbour. So, in this case, G has no perfect red–blue (S, T, X, Y) -colouring.

Now suppose that u is not adjacent to any vertex of $F - \{u, v\}$. Then u is only adjacent to v in G . We now remove u and v and we add any missing edge between two vertices of $F - \{u, v\}$ such that in the end $F - \{u, v\}$ has become a complete graph.

The above operation is safe to do, as in any perfect red–blue (S, T, X, Y) -colouring of the new graph (if it exists) the vertices of $F - \{u, v\}$ will all be coloured alike. This is because the matching neighbour of every vertex of $F - \{u, v\}$ belongs to $X \cup Y$ and we have modified $F - \{u, v\}$ into a complete graph. We can now give v the same colour as the vertices of $F - \{u, v\}$ and u the opposite colour. In this way we obtain a perfect red–blue (S, T, X, Y) -colouring of G . Similarly, as we argued above, a perfect

red–blue (S, T, X, Y) -colouring of G gives all the vertices of $F - \{u\}$ the same colour and makes u the matching neighbour of v . Hence, such a colouring (if it exists) will correspond to a perfect red–blue (S, T, X, Y) -colouring of the new graph.

We will also process any other reducible connected components of $G - (X \cup Y)$ in the same way. Then either we found that the original graph G has no perfect red–blue (S, T, X, Y) -colouring and we discard the option, or we found a new graph that has a perfect red–blue (S, T, X, Y) -colouring if and only if G has a perfect red–blue (S, T, X, Y) -colouring. Assume that we are in the latter situation. We continue with the new graph and denote it by G again (note that the new graph is the same graph as G if G had no reducible connected components). We now prove the following claim. \square

Claim *Every connected component of $G - (X \cup Y)$ in every perfect red–blue (S, T, X, Y) -colouring of G is monochromatic.*

Proof In order to see this claim, let F be a connected component of $G - (X \cup Y)$. If F corresponds to some reducible connected component in the original graph then, as we argued above, F will be monochromatic in any perfect red–blue (S, T, X, Y) -colouring of G .

Now suppose that F was not obtained from some reducible connected component. By construction, F is not reducible. If $|V(F)| = 1$, then F will be monochromatic. Assume $|V(F)| \geq 2$. As $V(G') \cup N(V(G')) \subseteq S' \cup T'$ and $S' \subseteq X$ and $T' \subseteq Y$, we find that $V(F)$ belongs to G^* . Since G^* is P_4 -free, F is P_4 -free. It is well-known (see e.g. Lemma 2 in [24]) that every connected P_4 -free graph has a spanning complete bipartite subgraph $K_{k,\ell}$ for some integers $1 \leq k \leq \ell$.

If $k \geq 2$ and $\ell \geq 3$, then F must be monochromatic. Now suppose that $k = \ell = 2$, so F contains a C_4 as a spanning subgraph. If F contains a vertex u that has both a neighbour in X and a neighbour in Y , then the matching neighbour of u will be in $X \cup Y$, so it is not in F . Hence, the two neighbours u' and u'' of u on the C_4 in F must receive the same colour as u . The latter means that the fourth vertex u^* of F must also receive the same colour as u (if u^* is not adjacent to u , then u^* will be adjacent to u' and u'' , as the vertices u, u', u'', u^* form a spanning C_4 of F). So F is monochromatic.

We conclude that every vertex of F is adjacent to at most one vertex of $X \cup Y$. As G is connected, F has at least one vertex v with a neighbour w in $X \cup Y$, say without loss of generality that $w \in X$. Then the other three vertices of F must also have a neighbour in X (and thus no neighbour in Y), else we would have applied R6. The only way we can extend the red–blue (S, T, X, Y) -colouring to a perfect red–blue colouring of G is by colouring each vertex of F blue, so F is monochromatic.

It remains to consider the case where $k = 1$ and $\ell \geq 1$. In this case F contains a vertex v such that $\{v\}$ dominates F . Then every vertex in $F - v$ has either no neighbours in $X \cup Y$, or it has both a neighbour in X and a neighbour in Y ; else we would have applied R7. Let U be the set of vertices in $F - v$ with no neighbour in $X \cup Y$. As $\{v\}$ dominates F , every connected component of $F - v$ is monochromatic. So, v is the matching neighbour of every vertex of U . Hence, if $|U| \geq 2$, then G has no perfect red–blue (S, T, X, Y) -colouring so the claim is true. If $|U| = 0$, then the vertices in $F - v$ all have a neighbour both in X and Y . So, they do not have their matching neighbour in F and thus will receive the same colour as v . Hence, F is monochromatic.

Assume that $|U| = 1$, say $U = \{u\}$ for some vertex u of G . As v is the matching neighbour of u , we find that v is adjacent to at most one vertex of $X \cup Y$.

We now have that F contains two adjacent vertices u and v , where u has no neighbours in $X \cup Y$ and moreover, v is dominating F , and every vertex in $F - \{u, v\}$ has both a neighbour in X and a neighbour in Y . Recall that F is not reducible. Hence, v is adjacent to exactly one vertex w of $X \cup Y$. Then u has at least one neighbour in $F - v$; else we would have applied R5. Let u' be an arbitrary neighbour of u in $F - v$. As both u and u' are adjacent to v , it follows that u, u', v are coloured alike. Hence, u has no matching neighbour. This means that G has no perfect red–blue (S, T, X, Y) -colouring and the claim is true. \square

We now apply rules R1–R11 exhaustively. This takes polynomial time by Lemma 18. If this leads to a no-answer, then by Lemma 18 we may discard the option. Suppose we obtain a final 4-tuple (S^*, T^*, X^*, Y^*) . By Lemma 18, G has a monochromatic perfect red–blue (S, T, X, Y) -colouring if and only if G has a monochromatic perfect red–blue (S^*, T^*, X^*, Y^*) -colouring. We apply Lemma 20 to find in polynomial time if the latter holds. If so, we are done by the Claim, else we discard the option.

The correctness of our algorithm follows from its description. As the total number of branches is $O(m2^{|V(H)|}n^{|V(H)|})$ and we can process each branch in polynomial time, the total running time of our algorithm is polynomial. Hence, we have proven the theorem. \square

5 Conclusions

We found new results on H -free graphs for three closely related edge cut problems: the classical MATCHING CUT problem and its variants, DISCONNECTED PERFECT MATCHING and PERFECT MATCHING CUT. We summarized all known and new results for H -free graphs in Theorems 1–3. We finish our paper with two open questions.

First, as can be noticed from Theorems 1–3, our knowledge on the complexity of the three problems is different. In particular, does there exist a constant r such that PERFECT MATCHING CUT is NP-complete for P_r -free graphs? For the other two problems such a constant exists. For MATCHING CUT we improved the previous value $r = 27$ [15] to $r = 15$ and for DISCONNECTED PERFECT MATCHING we showed that we can take $r = 19$, addressing a question in [6].

Second, is there a graph H for which the problems behave differently on H -free graphs? The graph $H = 3P_5$ is a candidate graph should it be possible to generalize Theorem 23 from $(H + P_4)$ -free graphs to $(H + P_5)$ -free graphs.

Acknowledgements We thank three anonymous reviewers for their helpful comments. In particular, we thank one referee for suggesting a change in our hardness gadgets that made it possible to prove Theorem 7 for $(3P_5, P_{15})$ -free graphs instead of $(4P_5, P_{19})$ -free graphs and Theorem 10 for $(3P_7, P_{19})$ -free graphs instead of $(4P_7, P_{23})$ -free graphs.

Author Contributions All authors contributed to the paper.

Data Availability Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors have no competing interests as defined by Springer, or other interests that might be perceived to influence the results and/or discussion reported in this paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Araújo, J., Cohen, N., Giroire, F., Havet, F.: Good edge-labelling of graphs. *Discrete Appl. Math.* **160**, 2502–2513 (2012)
2. Aravind, N.R., Kalyanasundaram, S., Kare, A.S.: Vertex partitioning problems on graphs with bounded tree width. *Discrete Appl. Math.* **319**, 254–270 (2022)
3. Aravind, N.R., Saxena, R.: An FPT algorithm for Matching Cut and d -Cut. *Proc. IWOCA 2021, LNCS 12757*, 531–543 (2021)
4. Bonsma, P.S.: The complexity of the Matching-Cut problem for planar graphs and other graph classes. *J. Graph Theory* **62**, 109–126 (2009)
5. Borowiecki, M., Jesse-Józefczyk, K.: Matching cutsets in graphs of diameter 2. *Theor. Comput. Sci.* **407**, 574–582 (2008)
6. Bouquet, V., Picouleau, C.: The complexity of the Perfect Matching-Cut problem. *CoRR, arXiv:2011.03318*, (2020)
7. Camby, E., Schaudt, O.: A new characterization of P_k -free graphs. *Algorithmica* **75**, 205–217 (2016)
8. Chen, C.-Y., Hsieh, S.-Y., Le, H.-O., Van Le, B., Peng, S.-L.: Matching Cut in graphs with large minimum degree. *Algorithmica* **83**, 1238–1255 (2021)
9. Chudnovsky, M.: The structure of bull-free graphs II and III—A summary. *J. Comb. Theory Series B* **102**, 252–282 (2012)
10. Chudnovsky, M., Seymour, P.: The structure of claw-free graphs. *Surveys in Combinatorics, London Mathematical Society Lecture Note Series* **327**, 153–171 (2005)
11. Chvátal, V.: Recognizing decomposable graphs. *J. Graph Theory* **8**, 51–53 (1984)
12. Dabrowski, K.K., Johnson, M., Paulusma, D.: Clique-width for hereditary graph classes. *Proc. BCC 2019, London Mathematical Society Lecture Note Series* **456**, 1–56 (2019)
13. Diwan, A.A.: Disconnected 2-factors in planar cubic bridgeless graphs. *J. Comb. Theory Series B* **84**, 249–259 (2002)
14. Farley, A.M., Proskurowski, A.: Networks immune to isolated line failures. *Networks* **12**, 393–403 (1982)
15. Feghali, C.: A note on Matching-Cut in P_t -free graphs. *Inf. Process. Lett.* **179**, 106294 (2023)
16. Funk, M., Jackson, B., Labbate, D., Sheehan, J.: 2-Factor hamiltonian graphs. *J. Comb. Theory Series B* **87**, 138–144 (2003)
17. Golovach, P.A., Johnson, M., Paulusma, D., Song, J.: A survey on the computational complexity of colouring graphs with forbidden subgraphs. *J. Graph Theory* **84**, 331–363 (2017)
18. Golovach, P.A., Komusiewicz, C., Kratsch, D., Van Le, B.: Refined notions of parameterized enumeration kernels with applications to matching cut enumeration. *J. Comput. Syst. Sci.* **123**, 76–102 (2022)
19. Golovach, P.A., Paulusma, D., Song, J.: Computing vertex-surjective homomorphisms to partially reflexive trees. *Theor. Comput. Sci.* **457**, 86–100 (2012)

20. Gomes, G., Sau, I.: Finding cuts of bounded degree: complexity, FPT and exact algorithms, and kernelization. *Algorithmica* **83**, 1677–1706 (2021)
21. Graham, R.L.: On primitive graphs and optimal vertex assignments. *Ann. New York Acad. Sci.* **175**, 170–186 (1970)
22. Heggenes, P., Telle, J.A.: Partitioning graphs into generalized dominating sets. *Nord. J. Comput.* **5**, 128–142 (1998)
23. Hermelin, D., Mnich, M., van Leeuwen, E.J., Woeginger, G.J.: Domination when the stars are out. *ACM Trans. Algorithms* **15**, 25:1–25:90 (2019)
24. Kern, W., Paulusma, D.: Contracting to a longest path in H -free graphs. *Proc. ISAAC 2020, LIPIcs* **181**, 22:1–22:18 (2020)
25. Komusiewicz, C., Kratsch, D., Van Le, B.: Matching Cut: kernelization, single-exponential time FPT, and exact exponential algorithms. *Discrete Appl. Math.* **283**, 44–58 (2020)
26. Kratsch, D., Van Le, B.: Algorithms solving the Matching Cut problem. *Theor. Comput. Sci.* **609**, 328–335 (2016)
27. Le, H., Van Le, B.: A complexity dichotomy for Matching Cut in (bipartite) graphs of fixed diameter. *Theor. Comput. Sci.* **770**, 69–78 (2019)
28. Le, V.B., Randerath, B.: On stable cutsets in line graphs. *Theor. Comput. Sci.* **301**, 463–475 (2003)
29. Le, V.B., Telle, J.A.: The perfect Matching Cut problem revisited. *Theor. Comput. Sci.* **931**, 117–130 (2022)
30. Lucke, F., Paulusma, D., Ries, B.: Finding Matching Cuts in H -free graphs. *Proc. ISAAC 2022, LIPIcs* **248**, 22:1–22:16 (2022)
31. Lucke, F., Paulusma, D., Ries, B.: On the complexity of Matching Cut for graphs of bounded radius and H -free graphs. *Theor. Comput. Sci.* **936**, 33–42 (2022)
32. Martin, B., Paulusma, D., Smith, S.: Hard problems that quickly become very easy. *Inf. Process. Lett.* **174**, 106213 (2022)
33. Moshi, A.M.: Matching Cutsets in graphs. *J. Graph Theory* **13**, 527–536 (1989)
34. Patrignani, M., Pizzonia, M.: The complexity of the Matching–Cut problem. *Proc. WG 2001, LNCS* **2204**, 284–295 (2001)
35. Randerath, B., Schiermeyer, I.: Vertex colouring and forbidden subgraphs—A survey. *Graphs Comb.* **20**, 1–40 (2004)
36. Schmidt, T.: Computational complexity of SAT, XSAT and NAE-SAT for linear and mixed Horn CNF formulas. PhD thesis, Universität zu Köln, (2010)
37. Van 't Hof, P.: A new characterization of P_6 -free graphs. *Discrete Appl. Math.* **158**, 731–740 (2010)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.