Finding Matching Cuts in H-Free Graphs

Felicia Lucke
Department of Informatics, University of Fribourg, Fribourg, Switzerland

Daniël Paulusma
Department of Computer Science, Durham University, Durham, UK

Bernard Ries
Department of Informatics, University of Fribourg, Fribourg, Switzerland

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 (arXiv, 2020). For all three problems, we give state-of-the-art summaries of their computational complexity for H-free graphs.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases matching cut; perfect matching; H-free graph; computational complexity

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2022.25

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 Figure 1. Matching cuts are well studied due to their applications in number theory [16], graph drawing [27], graph homomorphisms [15], edge labelings [1] and ILFI networks [12]. 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 Figure 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 Figure 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.

33rd International Symposium on Algorithms and Computation (ISAAC 2022).
Editors: Sang Won Bae and Heejin Park; Article No. 25; pp. 25:1–25:17
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Figure 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.

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 [25], 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.

It is readily seen that a graph class is hereditary if and only if it can be characterized by a set of forbidden induced subgraphs. A well-known example of a family of hereditary graph classes is obtained when we forbid 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 [14, 28] or clique-width [10], but also from extensive studies on H-free graphs for specific graphs H, such as bull-free graphs [7] or claw-free graphs [8, 18]. 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 [9] 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 [3, 21]. In particular, we refer to a recent paper of Chen et al. [6] for a comprehensive overview.

On the positive side, Bonsma [2] proved that Matching Cut is polynomial-time solvable for $K_{1,3}$-free graphs and P_4-free graphs. Recently, Feghali [13] proved the same for P_5-free graphs, which we extended to P_6-free graphs in [24]. 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 Section 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 [9] (see also [2, 20]). Bonsma [2] 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 [22] 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 trick of Moshi [26], NP-completeness for H-free graphs also holds if H has an even cycle (see [24]). Feghali [13] 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. This problem was introduced by Bouquet and Picouleau [4], under a different name, but to avoid confusion with Perfect Matching Cut, Le and Telle [23] introduced the notion of disconnected perfect matchings, which we adapted. As observed in [4], it follows from a result of Diwan [11] that every planar cubic bridgeless graph, except the K_4, has a disconnected perfect matching. Bouquet and

Picouleau [4] 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 Perfect Matching Cut. Heggernes and Telle [17] proved that this problem is NP-complete. Le and Telle [23] 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 [23], 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.

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 [13], we prove in Section 3 that Matching Cut is NP-complete for $(4P_5, P_{19})$-free graphs. Using the aforementioned trick of Moshi [26], we also observe that Matching Cut is NP-complete for H^*-free graphs, where H^* is the 6-vertex graph that looks like the letter "H".

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 [4] 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 $(4P_7, P_{23})$-free graphs in Section 3 (via modifying our construction for Matching Cut for $(4P_5, P_{19})$-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 Section 4, we first prove that Perfect Matching Cut is polynomial-time solvable for graphs of radius at most 2, and 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 are obtained by combining a number of known propagation rules [21, 23] 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 [2, 4, 9, 22, 23, 24, 26] 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 Section 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_{19}, $4P_5$ 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_{23} or $4P_7$.*

▶ **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 Section 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 $\{H_1, \ldots, H_n\}$ be a set of graphs. Then G is $(H_1, \ldots, H_n)$*-free*, if G is H_i-free for every $i \in \{1, \ldots, n\}$. 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 [5]).

▶ **Theorem 4** ([29])**.** *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_3s.

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 Figure 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, [13, 24]).

▶ **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.

3 NP-Completeness Results

We prove three NP-completeness results in this section. Our first result is a straightforward observation. Recall that H^* is the six-vertex graph that looks like the letter "H". Let uv be an edge in a graph G. Replacing uv by new vertices w_1 and w_2 and edges uw_1, uw_2, vw_1, vw_2 is a *$K_{2,2}$-replacement*. Let G_{uv} be the new graph. Moshi [26] observed that G has a matching cut if and only if G_{uv} has a matching cut. Applying a $K_{2,2}$-replacement on every edge to ensure that no two degree-3 vertices are adjacent anymore leads to the following:

▶ **Theorem 6.** Matching Cut *is* NP*-complete for H^*-free graphs.*

Recall that Feghali [13] showed the existence of an integer r such that Matching Cut is NP-complete for P_r-free graphs. It can be shown that the gadget in Feghali's construction has an induced P_{26}, but no induced P_{27}, so one can take $r = 27$ but not $r \leq 26$. Our next result improves this value of r to $r = 19$. The proof of our result is based on Feghali's construction [13] after making some minor modifications to it. We omit the details. It can be verified that the gadget in the proof of Theorem 7 is not $(P_{18} + sP_4)$-free for any $s \geq 1$.

▶ **Theorem 7.** Matching Cut *is* NP*-complete for $(4P_5, P_{19})$-free graphs.*

We can 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 [4]. Again, we omit the proof details.

▶ **Theorem 8.** Disconnected Perfect Matching *is* NP*-complete for $(4P_7, P_{23})$-free graphs.*

4 Polynomial Results

In Section 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 Sections 4.1–4.4, but we give an outline of it below.

Outline We prove the above results via a common approach. First, in Section 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 Section 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 Section 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 Section 4.3 that exploit this property. Afterwards, more vertices will be coloured, and we show in Section 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, whose proofs we omit; they are similar to the proofs for valid but not necessarily perfect red-blue colourings; see, for example, [13] or [24].

▶ **Lemma 9.** *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.*

▶ **Lemma 10.** *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.*

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' and T' by finding new vertices whose colour must always be either red or blue. We place new red vertices in a set X and new blue vertices in a set Y. 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 $T \subseteq Y$. Initially, $S := S''$, $T := T''$, $X := S'$ and $Y := T'$, and 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 [21]. Rules R3-R7 are for finding perfect red-blue colourings; some of them are in a slightly different form in [23].

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. We omit the proof of the next lemma, in which we show that Rules R1–R7 are safe.

▶ **Lemma 11.** *Rules R1–R7 are safe.*

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. The first part of the next lemma follows from Lemma 11. The second part is straightforward.

▶ **Lemma 12.** *Let G be a 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.*

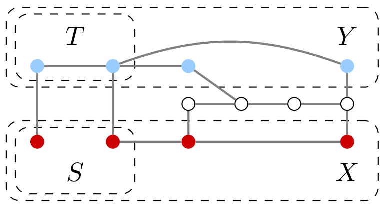

Figure 2 A red-blue (S, T, X, Y)-colouring of a graph with an intermediate 4-tuple (S, T, X, Y). In this example, $G[Z]$ consists of a single connected component isomorphic to P_4.

In the next lemma (proof omitted) we describe the structure of a graph with an intermediate 4-tuple (S, T, X, Y); see Figure 2 for an example.

▶ **Lemma 13.** *Let G be a 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 $Y \setminus T$.

4.3 Partial Red-Blue Colourings: Exploiting Monochromaticity

Let (S, T, X, Y) be an intermediate 4-tuple of a 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 [21, 23].

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.

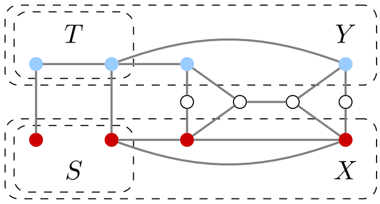

Figure 3 A red-blue (S, T, X, Y)-colouring of a graph with a final 4-tuple (S, T, X, Y). In this example, $G[Z]$ is isomorphic to $2P_1 + P_2$.

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. Our next lemma is not difficult to prove and we omit its proof.

▶ **Lemma 14.** *Rules R8–R11 are mono-safe.*

Suppose 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. The first part of Lemma 15 follows from Lemma 14. The second part is straightforward.

▶ **Lemma 15.** *Let G be a 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.*

In our next lemma (proof omitted) we describe the structure of a graph with a final 4-tuple (S, T, X, Y); see Figure 3 for an example.

▶ **Lemma 16.** *Let G be a 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 $Y \setminus T$.

4.4 Reduction to 2-SAT

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

▶ **Lemma 17.** *Let G be a 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 16-(v), we find that $|E^*| = 2|Z|$. By Lemma 16-(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 16-(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 [21], we now construct an instance ϕ of the 2-Satisfiability problem (2-SAT). Our 2-SAT formula differs from the one in [21] 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 known 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 16-(iii) and (iv), C and D are different connected components of $G[Z]$. First assume that $u \in X \setminus S$. By Lemma 16-(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 16-(i), every vertex in S has exactly one neighbour in Y. By Lemma 16-(ii), every vertex in T has exactly one neighbour in X. By Lemma 16-(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 16-(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 16-(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 16-(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 2 [24].

▶ **Theorem 18.** 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, say H has centre u and leaves $v_1, \ldots, v_s$ for some $s \geq 1$. 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 10 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 $v_1, \ldots, v_s$ 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 $i \in \{1, \ldots, s\}$. 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 $v_1, \ldots, v_s$ 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 12. If this exhaustive application leads to a no-answer, then by Lemma 12 we may discard the option. Suppose we obtain an intermediate 4-tuple (S, T, X, Y). By again applying Lemma 12, 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 $\{v_1, \ldots, v_s\} \setminus \{v_i\}$ 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 $w \in V(H) \setminus \{v_i\} = \{u, v_1, \ldots, v_s\} \setminus \{v_i\}$. As $u \in S' \subseteq S \subseteq X$ and $\{v_1, \ldots, v_s\} \setminus \{v_i\} \subseteq X$, 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 15. If this exhaustive application leads to a no-answer, then by Lemma 15 we may discard the option. Suppose we obtain a final 4-tuple (S^*, T^*, X^*, Y^*). By again applying Lemma 15, 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 17 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. ◀

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 9, a monochromatic dominating set, in which case we use Lemma 10, or it has radius 2, in which case we use Theorem 18.

▶ **Theorem 19.** 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 9 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 $\{u_1, \ldots, u_r\}$ and $\{v_1, \ldots, v_s\}$. We may assume without loss of generality that $r \leq s$.

If $r \geq 2$ and $s \geq 3$, then any starting pair $(\{u_i\}, \{v_j\})$ 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 10 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 2, and we apply Theorem 18. In the second case, $r \leq s \leq 2$, and thus G has domination number at most 4, and we apply Lemma 9. Hence, in both cases, we find in polynomial time whether or not G has a perfect red-blue colouring. ◀

For our last result we again apply our approach.

▶ **Theorem 20.** *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 Figure 4 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

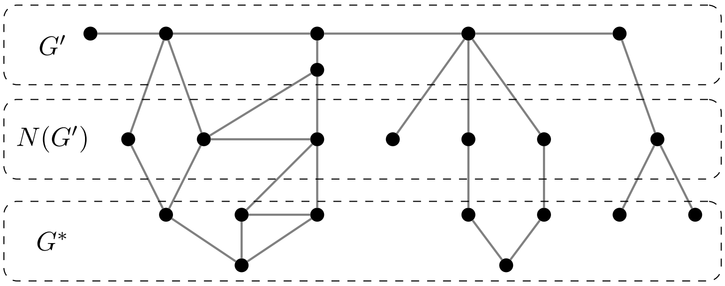

Figure 4 The decomposition of a graph G into the graphs G', which is isomorphic to H, the neighbourhood graph $N(G')$ of G', which is induced by all vertices not in G' but that have one or more neighbours in G', and the graph G^* induced by the remaining vertices of G.

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.

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 12. 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 a neighbour in both X and Y.

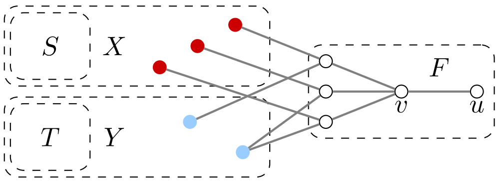

Figure 5 An example of a reducible connected component F of a graph G with an intermediate tuple (S, T, X, Y). For readability only edges with at least one end-vertex in F are drawn. Note that $F - \{u, v\}$ consists of three single-vertex components and that $F - \{u, v\}$ becomes a K_3 after our algorithm has processed F.

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 Figure 5 for an example.

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 a neighbour in both X and 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.

▷ 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 [19]) that every connected P_4-free graph has a spanning complete bipartite subgraph K. Say, K is isomorphic to $K_{k,\ell}$ 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 K contains a vertex u that has a neighbour in both X and Y, then the matching neighbour of u will be in $X\cup Y$, so not in F. Hence, the neighbours of u in F must receive the same colour as u. The latter means that the fourth vertex of F must also receive the same colour as u (if that vertex is not adjacent to u, then it will be adjacent to the two neighbours of u in F, as F contains a spanning C_4). 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 $w\in X$. 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 a neighbour in both X and 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 a neighbour in both X and 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. ◁

We now apply rules R1–R11 exhaustively. This takes polynomial time by Lemma 15. If this leads to a no-answer, then by Lemma 15 we may discard the option. Suppose we obtain a final 4-tuple (S^*,T^*,X^*,Y^*). By Lemma 15, 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 17 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. ◀

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 $r=27$ [13] to $r=19$ and for Disconnected Perfect Matching we showed that we can take $r=23$, addressing a question in [4]. We expect that these values of r might not be tight, but it does not seem straightforward to improve our current hardness constructions.

Second, is there a graph H for which the problems behave differently on H-free graphs? The graph $H=4P_5$ is a candidate graph should it be possible to generalize Theorem 20 from $(H+P_4)$-free graphs to $(H+P_5)$-free graphs.

References

1 Júlio Araújo, Nathann Cohen, Frédéric Giroire, and Frédéric Havet. Good edge-labelling of graphs. *Discrete Applied Mathematics*, 160:2502–2513, 2012.

2 Paul S. Bonsma. The complexity of the Matching-Cut problem for planar graphs and other graph classes. *Journal of Graph Theory*, 62:109–126, 2009.

3 Mieczyslaw Borowiecki and Katarzyna Jesse-Józefczyk. Matching cutsets in graphs of diameter 2. *Theoretical Computer Science*, 407:574–582, 2008.

4 Valentin Bouquet and Christophe Picouleau. The complexity of the Perfect Matching-Cut problem. *CoRR*, abs/2011.03318, 2020.

5 Eglantine Camby and Oliver Schaudt. A new characterization of P_k-free graphs. *Algorithmica*, 75:205–217, 2016.

6 Chi-Yeh Chen, Sun-Yuan Hsieh, Hoàng-Oanh Le, Van Bang Le, and Sheng-Lung Peng. Matching Cut in graphs with large minimum degree. *Algorithmica*, 83:1238–1255, 2021.

7 Maria Chudnovsky. The structure of bull-free graphs II and III - A summary. *Journal of Combinatorial Theory, Series B*, 102:252–282, 2012.

8 Maria Chudnovsky and Paul Seymour. The structure of claw-free graphs. *Surveys in Combinatorics, London Mathematical Society Lecture Note Series*, 327:153–171, 2005.

9 Vasek Chvátal. Recognizing decomposable graphs. *Journal of Graph Theory*, 8:51–53, 1984.

10 Konrad K. Dabrowski, Matthew Johnson, and Daniël Paulusma. Clique-width for hereditary graph classes. *Proc. BCC 2019, London Mathematical Society Lecture Note Series*, 456:1–56, 2019.

11 Ajit A. Diwan. Disconnected 2-factors in planar cubic bridgeless graphs. *Journal of Combinatorial Theory, Series B*, 84:249–259, 2002.

12 Arthur M. Farley and Andrzej Proskurowski. Networks immune to isolated line failures. *Networks*, 12:393–403, 1982.

13 Carl Feghali. A note on Matching-Cut in P_t-free graphs. *Information Processing Letters*, 179:106294, 2023.

14 Petr A. Golovach, Matthew Johnson, Daniël Paulusma, and Jian Song. A survey on the computational complexity of colouring graphs with forbidden subgraphs. *Journal of Graph Theory*, 84:331–363, 2017.

15 Petr A. Golovach, Daniël Paulusma, and Jian Song. Computing vertex-surjective homomorphisms to partially reflexive trees. *Theoretical Computer Science*, 457:86–100, 2012.

16 Ronald L. Graham. On primitive graphs and optimal vertex assignments. *Annals of the New York Academy of Sciences*, 175:170–186, 1970.

17 Pinar Heggernes and Jan Arne Telle. Partitioning graphs into generalized dominating sets. *Nordic Journal of Computing*, 5:128–142, 1998.

18 Danny Hermelin, Matthias Mnich, Erik Jan van Leeuwen, and Gerhard J. Woeginger. Domination when the stars are out. *ACM Transactions on Algorithms*, 15:25:1–25:90, 2019.

19 Walter Kern and Daniël Paulusma. Contracting to a longest path in H-free graphs. *Proc. ISAAC 2020, LIPIcs*, 181:22:1–22:18, 2020.

20 Dieter Kratsch and Van Bang Le. Algorithms solving the Matching Cut problem. *Theoretical Computer Science*, 609:328–335, 2016.

21 Hoàng-Oanh Le and Van Bang Le. A complexity dichotomy for Matching Cut in (bipartite) graphs of fixed diameter. *Theoretical Computer Science*, 770:69–78, 2019.

22 Van Bang Le and Bert Randerath. On stable cutsets in line graphs. *Theoretical Computer Science*, 301:463–475, 2003.

23 Van Bang Le and Jan Arne Telle. The Perfect Matching Cut problem revisited. *Proc. WG 2021, LNCS*, 12911:182–194, 2021.

24 Felicia Lucke, Daniël Paulusma, and Bernard Ries. On the complexity of Matching Cut for graphs of bounded radius and H-free graphs. *Theoretical Computer Science*, to appear.

25 Barnaby Martin, Daniël Paulusma, and Siani Smith. Hard problems that quickly become very easy. *Information Processing Letters*, 174:106213, 2022.

26 Augustine M. Moshi. Matching cutsets in graphs. *Journal of Graph Theory*, 13:527–536, 1989.

27 Maurizio Patrignani and Maurizio Pizzonia. The complexity of the Matching-Cut problem. *Proc. WG 2001, LNCS*, 2204:284–295, 2001.

28 Bert Randerath and Ingo Schiermeyer. Vertex colouring and forbidden subgraphs - A survey. *Graphs and Combinatorics*, 20:1–40, 2004.

29 Pim van 't Hof and Daniël Paulusma. A new characterization of P_6-free graphs. *Discrete Applied Mathematics*, 158:731–740, 2010.