Dichotomies for Maximum Matching Cut: H-Freeness, Bounded Diameter, Bounded Radius

Felicia Lucke
Department of Informatics, University of Fribourg, Switzerland

Daniël Paulusma
Department of Computer Science, Durham University, UK

Bernard Ries
Department of Informatics, University of Fribourg, Switzerland

Abstract

The (Perfect) Matching Cut problem is to decide if a graph G has a (perfect) matching cut, i.e., a (perfect) matching that is also an edge cut of G. Both Matching Cut and Perfect Matching Cut are known to be NP-complete, leading to many complexity results for both problems on special graph classes. A perfect matching cut is also a matching cut with maximum number of edges. To increase our understanding of the relationship between the two problems, we introduce the Maximum Matching Cut problem. This problem is to determine a largest matching cut in a graph. We generalize and unify known polynomial-time algorithms for Matching Cut and Perfect Matching Cut restricted to graphs of diameter at most 2 and to $(P_6 + sP_2)$-free graphs. We also show that the complexity of Maximum Matching Cut differs from the complexities of Matching Cut and Perfect Matching Cut by proving NP-hardness of Maximum Matching Cut for $2P_3$-free quadrangulated graphs of diameter 3 and radius 2 and for subcubic line graphs of triangle-free graphs. In this way, we obtain full dichotomies of Maximum Matching Cut for graphs of bounded diameter, bounded radius and H-free graphs.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases matching cut, perfect matching, H-free graph, diameter, radius, dichotomy

Digital Object Identifier 10.4230/LIPIcs.MFCS.2023.64

1 Introduction

A *matching* M (i.e., a set of pairwise disjoint edges) of a connected graph $G = (V, E)$ is a *matching cut* if V can be partitioned into a set of blue vertices B and a set of red vertices R, such that M consists of all the edges with one end-vertex in B and the other one in R. Graphs with matching cuts were introduced in 1970 by Graham [20] (as *decomposable* graphs) to solve a problem on cube numbering. Other relevant applications include ILFI networks [13], WDM networks [1], graph drawing [33] and surjective graph homomorphisms [18].

The decision problem is called Matching Cut: does a given connected graph have a matching cut? In 1984, Chvátal [11] proved that it is NP-complete even for graphs of maximum degree at most 4. Afterwards, parameterized and exact algorithms were given [2, 8, 17, 19, 24, 25]. A variant called Disconnected Perfect Matching "does a connected graph have a perfect matching that contains a matching cut?" has also been studied [7, 15, 31], and the problem was generalized, for every $d \geq 1$, to d-Cut "does a connected graph have an edge cut where each vertex has at most d neighbours across the cut?" [3, 19]. But, in particular, many results have appeared where the input for Matching Cut was restricted to some special graph class, and this is what we do in our paper as well. We first discuss related work, restricting ourselves mainly to those classes relevant to our paper (see, for example, [8] for a more comprehensive overview):

48th International Symposium on Mathematical Foundations of Computer Science (MFCS 2023).
Editors: Jérôme Leroux, Sylvain Lombardy, and David Peleg; Article No. 64; pp. 64:1–64:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

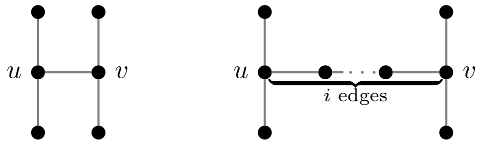

Figure 1 The graphs H_1^* (left) and H_i^* (right).

- graphs of bounded diameter;
- graphs of bounded radius;
- hereditary graph classes; in particular H-free graphs.

The *distance* between two vertices u and v in a connected graph G is the *length* (number of edges) of a shortest path between u and v in G. The *eccentricity* of a vertex u is the maximum distance between u and any other vertex of G. The *diameter*, denoted by $\mathsf{diameter}(G)$, and *radius*, denoted by $\mathsf{radius}(G)$, are the maximum and minimum eccentricity, respectively, over all vertices of G; note that $\mathsf{radius}(G) \leq \mathsf{diameter}(G) \leq 2 \cdot \mathsf{radius}(G)$ for every graph G.

The Matching Cut problem is polynomial-time solvable for graphs of diameter at most 2 [6, 26]. This result was extended to graphs of radius at most 2 [30]. In contrast, the problem is NP-complete for graphs of diameter at most 3 [26], yielding two dichotomies:

▶ **Theorem 1** ([26, 30]). *For an integer $d \geq 1$, Matching Cut for graphs of diameter d and for graphs of radius d is polynomial-time solvable if $d \leq 2$ and NP-complete if $d \geq 3$.*

A class of graphs is *hereditary* if it is closed under vertex deletion. Hereditary graph classes include many well-known classes, such as those that are H-free for some graph H. A graph G is *H-free* if G does not contain H as an *induced* subgraph, that is, G cannot be modified into H by a sequence of vertex deletions. For a set of graphs $\mathcal{H}$, a graph G is *$\mathcal{H}$-free* if G is H-free for every $H \in \mathcal{H}$. If $\mathcal{H} = \{H_1, \ldots, H_p\}$ for some $p \geq 1$, we also say that G is *$(H_1, \ldots, H_p)$-free*. Note that a class of graphs $\mathcal{G}$ is hereditary if and only if there is a set of graphs $\mathcal{H}$, such that every graph in $\mathcal{G}$ is $\mathcal{H}$-free. Hence, for a *systematic* complexity study, it is natural to first focus on the case where $\mathcal{H}$ has size 1; see, e.g., [9, 10, 12, 16, 22, 35].

For an integer $r \geq 1$, let P_r denote the path on r vertices, $K_{1,r}$ the star on $r+1$ vertices, and $K_{1,r}+e$ the graph obtained from $K_{1,r}$ by adding one edge (between two leaves). The graph $K_{1,3}$ is also known as the *claw*. For $s \geq 3$, let C_s denote the cycle on s vertices. Let H_1^* be the graph that looks like the letter "H", and for $i \geq 2$, let H_i^* be the graph obtained from H_1^* by subdividing the middle edge of H_1^* exactly $i-1$ times; see also Figure 1. We denote the *disjoint union* of two graphs G_1 and G_2 by $G_1 + G_2 = (V(G_1) \cup V(G_2), E(G_1) \cup E(G_2))$. We denote by sG the disjoint union of s copies of G, for $s \geq 1$.

Polynomial-time algorithms for Matching Cut exist for *subcubic* graphs (graphs of maximum degree at most 3) [11], $K_{1,3}$-free graphs [5], P_6-free graphs [30], $(K_{1,4}, K_{1,4}+e)$-free graphs [25] and *quadrangulated* graphs, i.e., $(C_5, C_6, \ldots)$-free graphs [32]; the latter class contains the class of *chordal* graphs, i.e., $(C_4, C_5, C_6, \ldots)$-free graphs. Moreover, if Matching Cut is polynomial-time solvable for H-free graphs, then it is so for $(H+P_3)$-free graphs [30]. The problem is NP-complete even for graphs of maximum degree at most 4 [11]; $K_{1,4}$-free graphs [11] (see [5, 25]); planar graphs of girth 5 [5]; $K_{1,5}$-free bipartite graphs [28]; graphs of girth at least g, for every $g \geq 3$ [15]; $(3P_5, P_{15})$-free graphs [31] (improving a result of [14]); bipartite graphs where the vertices in one bipartition class all have degree exactly 2 [32] and thus for H_i^*-free graphs for every odd $i \geq 1$; and for H_i^*-free graphs for every even $i \geq 2$ [15]. Even more recently, Le and Le [27] proved that Matching Cut is NP-complete even for $(3P_6, 2P_7, P_{14})$-free graphs.

The above results imply the following partial complexity classification, which leaves open only a number of cases where H is a *linear forest*, that is, the disjoint union of one or more paths. For two graphs H and H', we write $H \subseteq_i H'$ if H is an induced subgraph of H'.

▶ **Theorem 2** ([5, 11, 15, 27, 31, 30, 32]). *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 $H \supseteq_i K_{1,4}$, P_{14}, $3P_5$, $2P_7$, C_r for some $r \geq 3$ or H_j^* for some $j \geq 1$.*

Figure 2 The graph P_6 with a matching cut of size 2 (left), another matching cut of size 2 (middle) and a perfect matching cut (right). In each figure, thick edges denote matching cut edges.

1.1 Our Focus

We already mentioned the known generalization of Matching Cut (i.e. 1-Cut) to d-Cut. In our paper, we consider a different kind of generalization, namely Maximum Matching Cut, which is to determine a *largest* matching cut of a connected graph (if a matching cut exists). So far, this problem has only been studied for the extreme case, where the task is to decide if a connected graph has a *perfect* matching cut, that is, a matching cut that saturates every vertex; see also Figure 2. This variant was introduced as Perfect Matching Cut by Heggernes and Telle [21], who proved it was NP-complete. We briefly discuss some very recent results for Perfect Matching Cut on special graph classes below.

It is readily seen that the gadget in the NP-hardness reduction of Heggernes and Telle [21] has diameter 6 and radius 3. More recently Le and Le [27] gave a reduction with a graph of diameter 4. It is also known that Perfect Matching Cut is polynomial-time solvable for graphs of radius (and thus also diameter) at most 2 [31]. Hence, we only obtain a partial complexity classification for graphs of bounded diameter in this case.

▶ **Theorem 3** ([21, 31]). *For integers d and r,* Perfect Matching Cut *for graphs of diameter d and for graphs of radius r is polynomial-time solvable if $d \leq 2$ or $r \leq 2$, respectively, and* NP*-complete if $d \geq 4$ or $r \geq 3$, respectively.*

For $1 \leq h \leq i \leq j$, the graph $S_{h,i,j}$ is the tree of maximum degree 3 with exactly one vertex u of degree 3, whose leaves are at distance h, i and j, respectively, from u; note $S_{1,1,1} = K_{1,3}$.

It is known that Perfect Matching Cut is polynomial-time solvable for $S_{1,2,2}$-free graphs (and thus for $K_{1,3}$-free graphs) [29]; P_6-free graphs [31]; and for pseudo-chordal graphs [29] (and thus for chordal graphs, i.e., $(C_4, C_5, \ldots)$-free graphs). Moreover, Perfect Matching Cut is polynomial-time solvable for $(H + P_4)$-free graphs if it is polynomial-time solvable for H-free graphs [31]. It is also known that Perfect Matching Cut is NP-complete even for 3-connected cubic planar bipartite graphs [4], $(3P_6, 2P_7, P_{14})$-free graphs [27], $K_{1,4}$-free bipartite graphs of girth g for every $g \geq 3$ [29] and for H_i^*-free graphs for every $i \geq 1$ [15]. This gives us a partial complexity classification:

▶ **Theorem 4** ([15, 27, 29, 31]). *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 $H \supseteq_i K_{1,4}$, P_{14}, $3P_6$, $2P_7$, C_r for some $r \geq 3$ or H_j^* for some $j \geq 1$.*

From Theorem 4 it can be seen that again only cases where H is a linear forest remain open. However, the number of open cases is smaller than for Matching Cut. So far, all known complexities for Matching Cut and Perfect Matching Cut on special graph classes coincide except for (sub)cubic graphs. We note that whenever Maximum Matching Cut is polynomial-time solvable for some graph class, then so are Matching Cut and Perfect Matching Cut. Similarly, if one of the latter two problems is NP-complete, then Maximum Matching Cut is NP-hard. For instance, this immediately yields a complexity dichotomy for graphs of maximum degree at most Δ. Namely, as Maximum Matching Cut is trivial if $\Delta = 2$ and Perfect Matching Cut is NP-complete if $\Delta = 3$, we have a complexity jump from $\Delta = 2$ to $\Delta = 3$, just like Perfect Matching Cut; recall that for Matching Cut this jump appears from $\Delta = 3$ to $\Delta = 4$. We consider the following research question:

For which graph classes is Maximum Matching Cut *harder than* Matching Cut *and* Perfect Matching Cut *and for which graph classes do the complexities coincide?*

1.2 Our Results

In Section 4 we show that Maximum Matching Cut is NP-hard for $2P_3$-free quadrangulated graphs of diameter 3 and radius 2. We note that the restrictions to radius 2 and diameter 3 are not redundant: consider, for example, the P_6, which is $2P_3$-free but which has radius 3 and diameter 5. In the same section, we also show NP-hardness for subcubic line graphs of triangle-free graphs, or equivalently, subcubic ($K_{1,3}$, diamond)-free graphs (the diamond is obtained from the K_4 after removing an edge). These NP-hardness results are in stark contrast to the situation for Matching Cut and Perfect Matching Cut, as evidenced by Theorems 1–4. Recall also that Matching Cut is polynomial-time solvable for quadrangulated graphs [32].

Before proving these results, we first show in Section 3 that Maximum Matching Cut is polynomial-time solvable for graphs of diameter 2, generalizing the known polynomial-time algorithms for Matching Cut and Perfect Matching Cut for graphs of diameter at most 2. Hence, all three problems have the same dichotomies for graphs of bounded diameter.

We also prove in Section 3 that Maximum Matching Cut is polynomial-time solvable for P_6-free graphs, generalizing the previous polynomial-time results for Matching Cut and Perfect Matching Cut for P_6-free graphs. Due to the hardness result for $2P_3$-free graphs, we cannot show polynomial-time solvability for "$+P_4$" (as for Perfect Matching Cut) or "$+P_3$" (as for Matching Cut). However, we can prove that if Maximum Matching Cut is polynomial-time solvable for H-free graphs, then it is so for $(H + P_2)$-free graphs; again, see Section 3. The common proof technique for our polynomial-time results is as follows:

1. Translate the problem into a colouring problem. We pre-colour some vertices either red or blue, and try to extend the pre-colouring to a red-blue colouring of the whole graph via reduction rules. This technique has been used for Matching Cut and Perfect Matching Cut, but our analysis is different. In particular, the algorithms for Matching Cut and Perfect Matching Cut on P_6-free graphs use an algorithm for graphs of radius at most 2 as a subroutine (shortcut). We cannot do this for Maximum Matching Cut, as we will show NP-hardness for radius 2.
2. Reduce the set of uncoloured vertices, via a number of branching steps, to an independent set, and then translate the problem into a matching problem. This is a new proof ingredient. The matching problem is to find a largest matching that saturates every vertex of the independent set of uncoloured vertices. Plesník [34] gave a polynomial time algorithm for this, which we will use as subroutine.[1]

[1] The polynomial-time algorithm of Plesník [34] solves a more general problem. It takes as input a

The above polynomial-time and NP-hardness results yield the following three dichotomies for Maximum Matching Cut proven in Section 5; in particular we have obtained a complete complexity classification of Maximum Matching Cut for H-free graphs (whereas such a classification is only partial for the other two problems, as shown in Theorems 2 and 4).

▶ **Theorem 5.** *For an integer d,* Maximum Matching Cut *on graphs of diameter d is*
- *polynomial-time solvable if $d \leq 2$, and*
- NP-*hard if $d \geq 3$.*

▶ **Theorem 6.** *For an integer r,* Maximum Matching Cut *on graphs of radius r is*
- *polynomial-time solvable if $r \leq 1$, and*
- NP-*hard if $r \geq 2$.*

▶ **Theorem 7.** *For a graph H,* Maximum Matching Cut *on H-free graphs is*
- *polynomial-time solvable if $H \subseteq_i sP_2 + P_6$ for some $s \geq 0$, and*
- NP-*hard if $H \supseteq_i K_{1,3}$, $2P_3$ or $H \supseteq_i C_r$ for some $r \geq 3$.*

Finally, in Section 6 we pose a number of open problems.

2 Preliminaries

We consider finite, undirected graphs without multiple edges and self-loops. Let $G = (V, E)$ be a connected graph. For $u \in V$, the set $N(u) = \{v \in V(G) \mid uv \in E(G)\}$ is the *neighbourhood* of u in G, where $|N(u)|$ is the *degree* of u. For $S \subseteq V$, 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 $S \subseteq V(G)$, that is, $G[S]$ is the graph obtained from G after deleting the vertices not in S. We say that S is a *dominating* set of G, and that $G[S]$ *dominates* G if every vertex of $V(G) \setminus S$ has at least one neighbour in S. The *domination number* of G is the size of a smallest dominating set of G. The set S is an *independent set* if no two vertices in S are adjacent and S is a *clique* if every two vertices in S are adjacent. A matching M is *S-saturating* if every vertex in S is an end-vertex of an edge in M. An S-saturating matching is *maximum* if there is no S-saturating matching of G with more edges. We will use the following result.

▶ **Theorem 8** ([34]). *For a graph G and set $S \subseteq V(G)$, it is possible in polynomial time to find a maximum S-saturating matching or conclude that G has no S-saturating matching.*

The *line graph* of G is the graph $L(G)$ whose vertices are the edges of G, such that for every two vertices e and f, there exists an edge between e and f in $L(G)$ if and only if e and f share an end-vertex in G. 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 $|V_2| = \ell$, then we denote the complete bipartite graph by $K_{k,\ell}$. We will need the following theorem.

▶ **Theorem 9** ([36]). *A graph G on n vertices 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. We can find such a dominating subgraph of G in $O(n^3)$ time.*

A *red-blue colouring* of a connected graph 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*. An edge with a blue and a red end-vertex is *bichromatic*. A red-blue

graph G with an edge weighting w, a vertex subset S and two integers a and b. It then finds a maximum weight matching over all matchings that saturate S and whose cardinality is between a and b.

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. 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 $V(G) = R \cup B$). 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. The *value* of a valid red-blue colouring is its number of bichromatic edges, or equivalently, the size of its red (or blue) interface. A valid red-blue colouring is *maximum* if there is no valid red-blue colouring of the graph with larger value. The notion of a red-blue colouring has been used before (see e.g. [14, 30]), and the next observations are easy to see.

▶ **Observation 10.** *For every connected graph G and integer k, it holds that G has a matching cut with at least k edges if and only if G has a valid red-blue colouring of value at least k.*

▶ **Observation 11.** *Every complete graph K_r with $r \geq 3$ and every complete bipartite graph $K_{r,s}$ with $\min\{r, s\} \geq 2$ and $\max\{r, s\} \geq 3$ is monochromatic.*

We omitted the proof of our next lemma; it is very similar to the proofs of corresponding lemmas for Matching Cut [14] and Perfect Matching Cut [31]. On an aside, the lemma implies that Maximum Matching Cut is in XP when parameterized by the domination number of a graph.

▶ **Lemma 12.** *For a connected n-vertex graph G with domination number g, it is possible to find a maximum red-blue colouring (if a red-blue colouring exists) in $O(2^g n^{g+2})$ time.*

To handle "partial" red-blue colourings that we want to extend to maximum valid red-blue colourings, we slightly modify some terminology from [31] to work for maximum matching cuts as well.

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 of the vertices in $X \cup Y$, with a red set containing X; a blue set containing Y; a red interface containing S and a blue interface containing T. To obtain a red-blue (S, T, X, Y)-colouring, we start with two disjoint subsets S'' and T'' of V, called a *starting pair*, 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 (S'', T''). Note that $|S^*| = |T^*| \geq 1$.

We now 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 adjacent, 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 $Z := V \setminus (X \cup Y)$.

Our task is to try to extend the partial red-blue colouring on $X \cup Y$ to a *maximum* valid red-blue (S, T, X, Y)-colouring of G, that is, a valid red-blue (S, T, X, Y)-colouring that has largest value over all valid red-blue (S, T, X, Y)-colourings of G. In order to do this, we present three propagation rules, which indicate necessary implications of previous choices.

We start with rules R1 and R2, which together correspond to the five rules from [26]. Rule R1 detects cases where we cannot extend the partial red-blue colouring defined on $X \cup Y$. Rule R2 tries to extend the sets S, T, X, Y as much as possible. While the sets S, T, X, Y grow, Rule R2 ensures that we keep constructing a (maximum) valid red-blue (S, T, X, Y)-colouring (assuming G has a valid red-blue (S, T, X, Y)-colouring).

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 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 v is also adjacent to a vertex w in X, then add v to T and w to S.

Assume that exhaustively applying rules R1 and R2 on a starting pair (S'', T'') does not lead to a no-answer but to a tuple (S', T', X', Y'). Then, we call (S', T', X', Y') an *intermediate tuple*; see also Figure 3. A propagation rule is *safe* if for every integer ν the following holds: G has a valid red-blue (S,T,X,Y)-colouring of value ν before the application of the rule if and only if G has a valid red-blue (S,T,X,Y)-colouring of value ν after the application of the rule. Le and Le [26] proved the following lemma, which shows that R1 and R2 can be used safely and which is not difficult to verify. The fact that the value ν is preserved in Lemma 13 (ii) below is implicit in their proof.

▶ **Lemma 13** ([26]). *Let G be a connected graph with a starting pair (S'', T'') with core (S^*, T^*), and with an intermediate tuple (S', T', X', Y'). The following holds:*
(i) *$S^* \subseteq S'$, $S'' \subseteq X'$ and $T^* \subseteq T'$, $T'' \subseteq Y'$ and $X' \cap Y' = \emptyset$,*
(ii) *For every integer ν, G has a valid red-blue (S^*, T^*, S'', T'')-colouring of value ν if and only if G has a valid red-blue (S', T', X', Y')-colouring of value ν (note that the backward implication holds by definition), and*
(iii) *every vertex in S' (resp. T') has exactly one neighbour in Y' (resp. in X'), which belongs to T' (resp. S'); every vertex in $X' \setminus S'$ (resp. $Y' \setminus T'$) has no neighbour in Y' (resp. X'); and every vertex of $V \setminus (X' \cup Y')$ has no neighbour in $S' \cup T'$, at most one neighbour in $X' \setminus S'$, and at most one neighbour in $Y' \setminus T'$.*
Moreover, (S', T', X', Y') is obtained in polynomial time.

Let (S', T', X', Y') be an intermediate 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. We say that an intermediate tuple (S', T', X', Y') is *monochromatic* if every connected component of $G[V \setminus (X' \cup Y')]$ is monochromatic in every valid red-blue (S', T', X', Y')-colouring of G. A propagation rule is *mono-safe* if for every integer ν the following holds: G has a valid monochromatic red-blue (S,T,X,Y)-colouring of value ν before the application of the rule if and only if G has a valid monochromatic red-blue (S,T,X,Y)-colouring of value ν after the application of the rule.

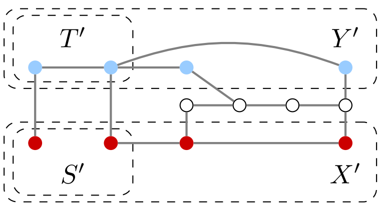

Figure 3 An example (from [31]) of a red-blue (S', T', X', Y')-colouring of a graph with an intermediate 4-tuple (S', T', X', Y').

We now present Rule R3 (used implicitly in [26]) and prove that R3 is mono-safe.

R3. If there are two distinct vertices u and v in a connected component D of $G[Z]$ with a common neighbour $w \in X \cup Y$, then colour every vertex of D with the colour of w.

▶ **Lemma 14.** *Rule R3 is mono-safe.*

Proof. Say $w \in X \cup Y$ is in X, so w is red. Then, at least one of x and y must be coloured red. Hence, as D must be monochromatic, every vertex of D must be coloured red. Note that the value of a maximum monochromatic red-blue (S,T,X,Y)-colouring (if it exists) is not affected. ◀

Suppose that exhaustively applying rules R1–R3 on an intermediate tuple (S',T',X',Y') does not lead to a no-answer but to a tuple (S,T,X,Y). We call (S,T,X,Y) the *final* tuple. The following lemma can be proved by a straightforward combination of the arguments of the proof of Lemma 13 with Lemma 14 and the observation that an application of R3 takes polynomial time, just as a check to see if R3 can be applied.

▶ **Lemma 15.** *Let G be a connected graph with a monochromatic intermediate tuple (S',T',X',Y') and a resulting final tuple (S,T,X,Y). The following three statements hold:*

(i) *$S' \subseteq S$, $X' \subseteq X$, $T' \subseteq T$, $Y' \subseteq Y$, and $X \cap Y = \emptyset$,*

(ii) *For every integer ν, G has a valid (monochromatic) red-blue (S',T',X',Y')-colouring of value ν if and only if G has a valid monochromatic red-blue (S,T,X,Y)-colouring of value ν (note that the backward implication holds by definition), and*

(iii) *every vertex in S (resp. T) has exactly one neighbour in Y (resp. X), which belongs to T (resp. S); every vertex in $X \setminus S$ (resp. $Y \setminus T$) has no neighbour in Y (resp. X) and no two neighbours in the same connected component of $G[V \setminus (X \cup Y)]$; and every vertex of $V \setminus (X \cup Y)$ has no neighbour in $S \cup T$, at most one neighbour in $X \setminus S$, and at most one neighbour in $Y \setminus T$.*

Moreover, (S,T,X,Y) is obtained in polynomial time.

3 Polynomial-Time Results

The following lemma uses Theorem 8 and is the final step in *all* our polynomial-time results.

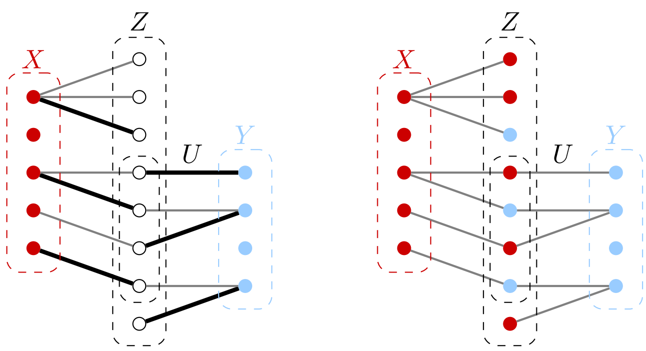

Figure 4 A U-saturating matching (left) and the corresponding valid red-blue colouring (right). Note that not every vertex in $X \cup Y$ belongs to W.

▶ **Lemma 16.** *Let $G = (V, E)$ be a connected graph with a monochromatic intermediate tuple (S', T', X', Y') and a final tuple (S, T, X, Y). If $V \setminus (X \cup Y)$ is an independent set, then it is possible to find in polynomial time either a maximum valid red-blue (S, T, X, Y)-colouring of G or conclude that G has no such colouring.*

Proof. Let $Z = V \setminus (X \cup Y)$ and $W = N(Z)$. As Z is independent, every vertex of W belongs to $(X \setminus S) \cup (Y \setminus T)$ due to Lemma 15-(iii). Let $U \subseteq Z$ consist of all vertices of Z with a neighbour in both $X \setminus S$ and $Y \setminus T$. We claim that the set of bichromatic edges of every valid red-blue (S, T, X, Y)-colouring is the union of a U-saturating matching in $G[W \cup Z]$ (if it exists) and the set of edges with one end-vertex in S and the other one in T.

First suppose that $G[W \cup Z]$ has a U-saturating matching M. We colour every vertex in X red and every vertex in Y blue. Let $z \in Z$. First assume that z is incident to an edge $zw \in M$. If $w \in X \setminus S$, then colour z blue. If $w \in Y \setminus T$, then colour z red. Now suppose z is not incident to an edge in M. Then $z \notin U$, as M is U-saturating. Hence, either every neighbour of z belongs to $X \setminus S$ and is coloured red, in which case we colour z red, or every neighbour of z belongs to $Y \setminus T$ and is coloured blue, in which case we colour z blue. This gives us a valid red-blue (S, T, X, Y)-colouring of G. See also Figure 4.

Now suppose that G has a valid red-blue (S, T, X, Y)-colouring. By definition, every vertex of X is coloured red, and every vertex of Y is coloured blue. By Lemma 15-(iii), every edge with an end-vertex in S and the other one in T is bichromatic, and there are no other bichromatic edges in $G[X \cup Y]$. Let M be the set of other bichromatic edges. Then, every vertex of M has one vertex in Z and the other one in W. Moreover, if $z \in U$, then z has a red neighbour (its neighbour in $X \setminus S$) and a blue neighbour (its neighbour in $Y \setminus T$). Hence, no matter what colour z has itself, z is incident to a bichromatic edge of M. We conclude that M is U-saturating, and the claim is proven.

From the above claim, it follows that all we have to do is to find a maximum U-saturating matching in $G[W \cup Z]$. By Theorem 8, this takes polynomial time. ◀

We are now ready to present our first result.

▶ **Theorem 17.** Maximum Matching Cut *is solvable in polynomial time for P_6-free graphs.*

Proof. Let $G = (V, E)$ be a connected P_6-free graph. By Observation 10 it suffices to find a maximum valid red-blue colouring of G. We know from Theorem 9 that G has a dominating induced C_6 or a dominating (not necessarily induced) complete bipartite graph $K_{r,s}$, which can be found in polynomial time. If G has a dominating induced C_6, then G has domination number at most 6, and we apply Lemma 12. Suppose G has a dominating complete bipartite graph F with partition classes $\{u_1, \ldots, u_r\}$ and $\{v_1, \ldots, v_s\}$, where $r \leq s$. If $s \leq 2$, then G has domination number at most 4, and we apply Lemma 12 again. So we assume that $s \geq 3$.

If $r \geq 2$, then $V(F)$ must be monochromatic in any valid red-blue colouring of G by Observation 11. In this case we colour every vertex of $V(F)$ blue. If $r = 1$, then we may assume without loss of generality that $N(u_1) = \{v_1, \ldots, v_s\}$. In this case we colour u_1 blue, and we branch over all $O(n)$ options of colouring at most one vertex of $N(u_1)$ red.

So, now we consider a red-blue colouring of F. It might be that F is monochromatic (in particular, this will be the case if $r \geq 2$). If F is monochromatic, then every vertex of F is blue. In order to get a starting pair with a non-empty core, we branch over all $O(n^2)$ options of choosing a bichromatic edge (one end-vertex of which may belong to F). Let D be the set of all coloured vertices, that is, D contains $V(F)$ and possibly one or two other vertices. By construction, exactly one vertex of D is coloured red, and all other vertices of D are blue.

Let $S^* = S''$ be the set containing the red vertex of D. Let T^* be the singleton set containing the blue neighbour of the vertex in S^*. Let T'' be the set of blue vertices, so $T^* \subseteq T''$. We exhaustively apply rules R1 and R2 on the starting pair (S'', T''). By Lemma 13

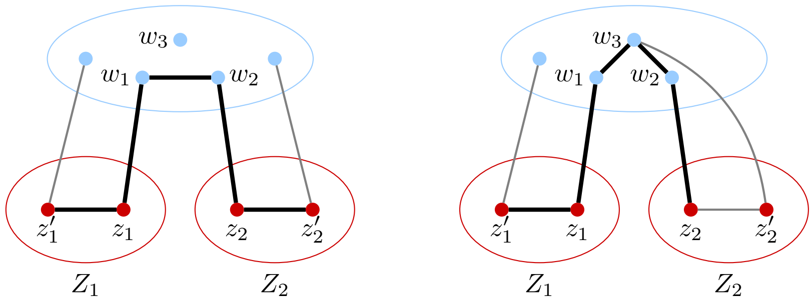

Figure 5 The situation in Claim 17.3 where two connected components Z_1, Z_2 of $G[Z]$, each with at least two vertices, are both coloured red. This will always yield an induced path on at least six vertices, even if w_1 and w_2 are not adjacent, as at most one of z_1', z_2' is adjacent to w_3.

we either find in polynomial time that G has no valid red-blue (S^*, T^*, S'', T'')-colouring, and we discard the branch, or we obtain an intermediate tuple (S', T', X', Y') of G. Suppose the latter case holds. Let $Z' = V \setminus (X' \cup Y')$ be the set of uncoloured vertices.

▷ Claim 17.1. Every vertex $z \in Z'$ has a neighbour in $Y' \setminus T'$ that belongs to F.

Proof. As F is dominating, z has a neighbour in F. Since $D \supseteq V(F)$ contains exactly one red vertex x, which has a blue neighbour in D, all neighbours of x in $G - D$ are coloured red, that is, belong to X. As $z \in G - D$ belongs to Z', this means that x and z are non-adjacent. So, the neighbour of z in F must belong to $Y' \setminus T'$ (as else we could have applied R2). ◁

▷ Claim 17.2. The intermediate tuple (S', T', X', Y') is monochromatic.

Proof. Suppose for a contradiction that there is an edge $uv \in E(G[Z'])$ such that u is blue and v is red. Then v has two blue neighbours by Claim 17.1, a contradiction. ◁

Since Claim 17.2 holds, we may now exhaustively apply R1–R3 to the intermediate tuple (S', T', X', Y'). By Lemma 15 we either find in polynomial time that G has no valid red-blue (S', T', X', Y')-colouring, and thus no valid red-blue (S^*, T^*, S', T')-colouring, and we discard the branch, or we obtain a final tuple (S, T, X, Y) of G. Again, we let $Z = V \setminus (X \cup Y)$. By the same lemma and Claim 17.1, the following holds for every (uncoloured) vertex $w \in Z$:

- w has at most one neighbour in $X \setminus S$,
- w has exactly one neighbour in $Y \setminus T$, which belongs to F, and
- if w' is in the same connected component of $G[Z]$ as w, then w and w' do not share a neighbour in $G - Z$.

▷ Claim 17.3. In any valid red-blue (S, T, X, Y)-colouring at most one red component may have more than one vertex.

Proof. For a contradiction, assume that Z_1 and Z_2 are connected components of size at least 2 that are both coloured red. For $i = 1, 2$, let z_i and z_i' be two adjacent vertices in Z_i, and let w_i be the blue neighbour of z_i in F (which exists due to Claim 17.1). Note that w_1 is not adjacent to any vertex of $\{z_1', z_2, z_2'\}$, and w_2 is not adjacent to any vertex of $\{z_1, z_1', z_2'\}$. Moreover, w_1 and w_2 are distinct vertices, and do not have any other neighbours in $Z_1 \cup Z_2$. If w_1 and w_2 are adjacent, then $z_1' z_1 w_1 w_2 z_2 z_2$ is an induced P_6. As G is P_6-free, this is not possible. Hence, w_1 and w_2 are not adjacent.

We now use the fact that w_1 and w_2 both belong to F and that F is complete bipartite. As $w_1 w_2 \notin E$, the latter means there is a vertex $w_3 \in V(F)$ adjacent to both w_1 and w_2, so w_3 is blue as well. As z_1' and z_2' are both coloured red, at most one of z_1', z_2' can be adjacent to w_3. Hence, we may assume without loss of generality that w_3 is not adjacent to z'_1. As z_1 and z_2 have w_1 and w_2, respectively, as their matching partner, w_3 is adjacent neither to z_1 nor to z_2. Now, $z'_1 z_1 w_1 w_3 w_2 z_2$ is an induced P_6, a contradiction. See also Figure 5. ◁

We then exhaustively apply rules R1-R3 again. This takes polynomial time. In essence, we merely pre-coloured some more vertices red. So, in the end we either find a new tuple of G with the same properties as those listed in Lemma 15, or we find that G has not such a tuple, in which case we discard the branch. Suppose we have not discarded the branch. Now the set of uncoloured vertices form an independent set. Hence, we can apply Lemma 16 to find in polynomial time a red-blue colouring of G that is a maximum red-blue (S^*, T^*, S'', T'')-colouring due to Lemmas 13-(ii) and 15-(ii).

If somewhere in the above process we discarded a branch, that is, if G has no valid red-blue (S^*, T^*, S'', T'')-colouring, we consider the next one. Else remember the value of the maximum red-blue (S^*, T^*, S'', T'')-colouring that we found. Afterwards, we pick one with the largest value to obtain a maximum valid red-blue colouring of G.

The correctness of our algorithm follows from its description. The running time is polynomial: each of the in total $O(n^3)$ branches takes polynomial time to process. ◀

The proof of our next result combines Lemma 16 with arguments from the proof that Matching Cut is polynomial-time solvable for $(H + P_3)$-free graphs if it is so for H-free graphs [30]. We omit the proof.

▶ **Theorem 18.** *Let H be a graph. If* Maximum Matching Cut *is polynomial-time solvable for H-free graphs, then it is so for $(H + P_2)$-free graphs.*

We now show our third polynomial-time result; we will again apply Lemma 16.

▶ **Theorem 19.** Maximum Matching Cut *is solvable in polynomial time for graphs with diameter at most* 2.

Proof sketch. Let $G = (V, E)$ be a graph of diameter at most 2. If G has diameter 1, then the problem is trivial to solve. Assume that G has diameter 2. By Observation 10 it suffices to find a maximum valid red-blue colouring of G. By definition, such a colouring has at least one bichromatic edge (has value at least 1). We branch over all $O(n^2)$ options of choosing the bichromatic edge.

Consider a branch, where $e = uv$ is the bichromatic edge, say u is blue and v is red. All other neighbours of u must be coloured blue. Let $D = \{u\} \cup N(u)$. Then D dominates G, as G has diameter 2. Set $S^* = \{u\}$, $T^* = \{v\}$, $S'' = \{u\}$ and $T'' = N(u)$. This gives us a starting pair (S'', T'') with core (S^*, T^*). We exhaustively apply R1 and R2 on (S'', T''). By Lemma 13 we either find in polynomial time that G has no valid red-blue (S^*, T^*, S'', T'')-colouring, and we discard the branch, or we obtain an intermediate tuple (S', T', X', Y'). Say the latter holds. Let $Z' = V \setminus (X' \cup Y')$ be the set of uncoloured vertices. We show the following claim (proof omitted).

▷ Claim 19.1. The intermediate tuple (S', T', X', Y') is monochromatic.

By Claim 19.1, we may exhaustively apply R1–R3 to the intermediate tuple (S', T', X', Y'). By Lemma 15 we either find in polynomial time that G has no valid red-blue (S', T', X', Y')-colouring, and thus no valid red-blue (S^*, T^*, S', T')-colouring, and we discard the branch, or we obtain a final tuple (S, T, X, Y) of G. We let $Z = V \setminus (X \cup Y)$. We show the following claims (proofs omitted).

▷ Claim 19.2. Every vertex $w \in Z$ has exactly one neighbour in $X \setminus S$.

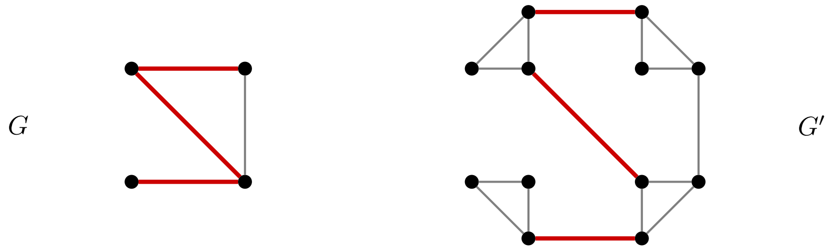

Figure 6 A graph G (left) where the tick red edges form a maximum edge cut, and the graph G' (right) from the proof of Theorem 20, where the thick red edges form a maximum matching cut.

▷ Claim 19.3. If $G[Z]$ contains two connected components F_1 and F_2 of size at least 2, then $G[Z] = F_1 + F_2$.

We use Claim 19.2 to prove Claim 19.3, from which it follows that $G[Z]$ has at most two components with more than one vertex, which are both monochromatic in every valid red-blue (S, T, X, Y)-colouring of G (if such a colouring exists) due to Claim 19.1. Hence, we can branch over all possible colourings of these connected components (there are at most four branches).

For each branch, we propagate the obtained partial red-blue colouring by exhaustively applying rules R1–R3. This takes polynomial time. In essence, we merely pre-coloured some more vertices red or blue. So, in the end we either find a new tuple of G with the same properties as those listed in Lemma 15, or we find that G has not such a tuple, in which case we discard the branch. Suppose not. Now the set of uncoloured vertices form an independent set. Hence, we can apply Lemma 16 to find in polynomial time a red-blue colouring of G that is a maximum red-blue (S^*, T^*, S'', T'')-colouring due to Lemmas 13-(ii) and 15-(ii).

If we discarded a branch, that is, if G has no valid red-blue (S^*, T^*, S'', T'')-colouring, we consider the next one. If we did not discard the branch, then we remember the value of the maximum red-blue (S^*, T^*, S'', T'')-colouring that we found. Afterwards, we pick one with the largest value to obtain a maximum valid red-blue colouring of G.

The correctness of our branching algorithm follows from its description. The running time is polynomial: each of the in total $O(n^2)$ branches takes polynomial time to process. ◀

4 Hardness Results

We sketch the two hardness proofs. For the first one we reduce from Maximum Cut, which is NP-complete even for subcubic graphs [37]: does a subcubic graph G have an edge cut of size at least k for some integer k?

▶ **Theorem 20.** Maximum Matching Cut *is* NP*-hard for subcubic line graphs of triangle-free graphs.*

Proof sketch. Let (G, k) be an instance of Maximum Cut, where G is subcubic. Build a graph G' as follows (see also Figure 6). Replace every vertex $v \in V(G)$ by a triangle C_v. For every edge $uv \in E(G)$, add an edge between a vertex in C_v and a vertex in C_u, such that every vertex in C_v has at most one neighbour outside C_v. This is possible, as G is subcubic. The graph G' is also subcubic, as every vertex in G' has two neighbours inside a triangle and at most one neighbour outside. Moreover, G' is $(K_{1,3}, \text{diamond})$-free, so the line graph of a triangle-free graph. We can show that G has an edge cut of size at least k if and only if G' has a maximum matching cut of size at least k. ◀

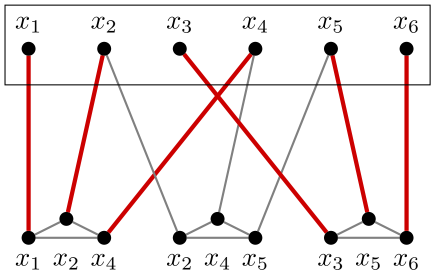

Figure 7 The graph G for $X = \{x_1, \ldots, x_6\}$ and $\mathcal{S} = \{\{x_1, x_2, x_4\}, \{x_2, x_4, x_5\}, \{x_3, x_5, x_6\}\}$. The vertices in the rectangle form a clique. The set $\mathcal{S}' = \{\{x_1, x_2, x_4\}, \{x_3, x_5, x_6\}\}$ is an exact 3-cover of X. The thick red edges in the graph show the corresponding maximum matching cut.

An *exact* 3*-cover* of a set X is a collection $\mathcal{C}$ of 3-element subsets of X, such that every $x \in X$ is in exactly one 3-element subset of $\mathcal{C}$. We now reduce from Exact 3-Cover, which is to decide if a collection $\mathcal{S}$ of 3-element subsets of a set X with q elements has an *exact* 3*-cover* of X (which will be of size q). This problem is NP-complete (see [23]).

▶ **Theorem 21.** Maximum Matching Cut *is* NP*-hard for* $2P_3$*-free quadrangulated graphs of radius at most* 2 *and diameter at most* 3*.*

Proof sketch. Let $(X, \mathcal{S})$ be an instance of Exact 3-Cover where $X = \{x_1, \ldots, x_{3q}\}$ and $\mathcal{S} = \{S_1, \ldots, S_k\}$, such that each S_i contains exactly three elements of X. From $(X, \mathcal{S})$ we construct a graph G as follows (see also Figure 7). We first define a clique $K_X = \{x_1, \ldots, x_{3q}\}$. For each $S \in \mathcal{S}$, we do as follows. Let $S = \{x_h, x_i, x_j\}$. We add a a triangle K_S on vertices x_h^S, x_i^S and x_j^S. We add an edge between a vertex $x_i \in K_X$ and a vertex $u \notin K_X$ if and only if $u = x_i^S$ for some $S \in \mathcal{S}$. This completes the construction of G. Note that G is $2P_3$-free, quadrangulated as well as that the radius of G is at most 2 and the diameter of G is at most 3. We can show that $\mathcal{S}$ contains an exact 3-cover of X if and only if G has a matching cut of size $3q$. ◀

5 The Proofs of Theorems 5–7

We first note that Theorem 5 immediately follows from Theorems 19 and 21.

The first part of Theorem 6 follows from the fact that a graph of radius 1 has a dominating vertex, and thus it has a matching cut if and only if it has a vertex of degree 1, which can be checked in polynomial time. The second part of Theorem 6 follows from Theorem 21.

To prove Theorem 7, let H be a graph. If H contains a cycle, then Matching Cut, and thus Maximum Matching Cut, is NP-hard due to Theorem 2. Now suppose H has no cycle, so H is a forest. If H contains a vertex of degree at least 3, then the class of H-free graphs contains the class of $K_{1,3}$-free graphs. The latter class contains the class of line graphs, and thus we apply Theorem 20. Now suppose H is a forest of maximum degree at most 2, that is, H is a linear forest. If $H \subseteq_i sP_2 + P_6$ for some $s \geq 0$, then we apply Theorem 17. Else H has an induced $2P_3$. We apply Theorem 21. This completes the proof.

6 Conclusions

We considered the optimization version Maximum Matching Cut of the classical Matching Cut problem after first observing that the Perfect Matching Cut problem is a special case of the former problem. We generalized known algorithms for graphs of diameter at most 2 and P_6-free graphs from Matching Cut and Perfect Matching Cut to Maximum Matching Cut. We also showed that the latter problem is computationally harder (assuming $\mathsf{P} \neq \mathsf{NP}$) than Matching Cut and Perfect Matching Cut for various graph classes. Our results led to three new dichotomy results, including a complete computational complexity classification of Maximum Matching Cut for H-free graphs. The latter classification is still unsettled for the other two problems, as can be observed from Theorems 2 and 4. Below we discuss some other open problems.

We first recall that the complexity of Perfect Matching Cut has not been fully classified for graphs of diameter at most d. What is the complexity of Perfect Matching Cut in the remaining open case where $d = 3$? We showed that Maximum Matching Cut is NP-hard for $2P_3$-free quadrangulated graphs of diameter 3 and radius 2, whereas Matching Cut is polynomial-time solvable for quadrangulated graphs [32]. We recall an interesting open problem of Le and Telle [29] who asked, after proving polynomial-time solvability for chordal graphs: what is the complexity of Perfect Matching Cut for quadrangulated graphs, or more general, k-chordal graphs for $k \geq 4$? Here, a graph is *k-chordal* for some $k \geq 3$ if it is $(C_{k+1}, C_{k+2}, \ldots)$-free, so 3-chordal graphs are the chordal 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 N. R. Aravind, Subrahmanyam Kalyanasundaram, and Anjeneya Swami Kare. Vertex partitioning problems on graphs with bounded tree width. *Discrete Applied Mathematics*, 319:254–270, 2022.

3 N. R. Aravind and Roopam Saxena. An FPT algorithm for Matching Cut and d-Cut. *Proc. IWOCA 2021, LNCS*, 12757:531–543, 2021.

4 Edouard Bonnet, Dibyayan Chakraborty, and Julien Duron. Cutting barnette graphs perfectly is hard. *Proc. WG 2023, LNCS*, to appear.

5 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 (conference version: WG 2003).

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

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

8 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.

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

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

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

12 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.

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

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

15 Carl Feghali, Felicia Lucke, Daniël Paulusma, and Bernard Ries. Matching cuts in graphs of high girth and H-free graphs. *CoRR*, abs/2212.12317, 2022.

16 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.

17 Petr A. Golovach, Christian Komusiewicz, Dieter Kratsch, and Van Bang Le. Refined notions of parameterized enumeration kernels with applications to matching cut enumeration. *Journal of Computer and System Sciences*, 123:76–102, 2022.

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

19 Guilherme Gomes and Ignasi Sau. Finding cuts of bounded degree: complexity, FPT and exact algorithms, and kernelization. *Algorithmica*, 83:1677–1706, 2021.

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

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

22 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.

23 Richard M. Karp. Reducibility among Combinatorial Problems. *Complexity of Computer Computations*, pages 85–103, 1972.

24 Christian Komusiewicz, Dieter Kratsch, and Van Bang Le. Matching Cut: Kernelization, single-exponential time FPT, and exact exponential algorithms. *Discrete Applied Mathematics*, 283:44–58, 2020.

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

26 Hoang-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.

27 Hoàng-Oanh Le and Van Bang Le. Complexity results for matching cut problems in graphs without long induced paths. *Proc. WG 2023, LNCS*, to appear.

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

29 Van Bang Le and Jan Arne Telle. The Perfect Matching Cut problem revisited. *Theoretical Computer Science*, 931:117–130, 2022.

30 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*, 936, 2022.

31 Felicia Lucke, Daniël Paulusma, and Bernard Ries. Finding matching cuts in H-free graphs. *Algorithmica*, to appear.

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

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

34 Ján Plesník. Constrained weighted matchings and edge coverings in graphs. *Discrete Applied Mathematics*, 92:229–241, 1999.

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

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

37 Mihalis Yannakakis. Node-and edge-deletion NP-complete problems. *Proc. STOC 1978*, pages 253–264, 1978.