UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG

Felicia Lucke

Matching Cut and Variants in Graphs of Bounded Radius, Bounded Diameter and H-free Graphs

Faculty of Science and Medicine

DOI : https://doi.org/10.51363/unifr.sth.2026.070

Felicia Lucke https://orcid.org/0000-0002-9860-2928

Department of Informatics
University of Fribourg (Switzerland)

Matching Cut and Variants in Graphs of Bounded Radius, Bounded Diameter and H-free Graphs

THESIS

presented to the Faculty of Science and Medicine of the University of Fribourg (Switzerland) in consideration for the award of the academic grade of

Doctor of Philosophy in Computer Science

by
Felicia Lucke
from
Germany

Thesis No: 8038
Uniprint
2024

Accepted by the Faculty of Science and Medicine of the University of Fribourg (Switzerland) upon the recommendation of Prof. Ulrich Ultes-Nitsche (President of the Jury), Prof. Philippe Cudré-Mauroux, Prof. Matthew Johnson and Prof. Van Bang Le.

Fribourg, the 25.07.2024

Thesis Supervisor
Prof. Bernard Ries

Dean
Prof. Ulrich Ultes-Nitsche

Acknowledgements

Doing this PhD would not have been possible without the support and advice of different people. First of all, I would like to express my gratitude to my supervisor Prof. Bernard Ries for enabling me to do a PhD in Fribourg and for all the further opportunities to travel and to learn. Thank you, Bernard, for always being available to discuss my questions, for your helpful guidance, and your encouragement for the projects during the PhD and in the future.

I am thankful to Prof. Daniël Paulusma, who suggested to work on the Matching Cut problem. Thank you, Daniel, not only for suggesting this wonderful problem but also for this collaboration, for your nice writing style, for hosting me at Durham University and for all your very helpful comments during the last years.

I also thank Prof. Philippe Cudré-Mauroux, Prof. Matthew Johnson and Prof. Van Bang Le for accepting to be members of the jury, for coming to Fribourg and for their valuable feedback.

Further, I wish to thank my coauthors Felix Mann, Dr. Carl Feghali, Prof. Van Bang Le, Dr. Siani Smith and Ali Momeni for our collaborations, and Dr. Carl Feghali and Prof. Christophe Picouleau for hosting me at the ENS Lyon and the CNAM in Paris, respectively.

I want to thank all former and current members of the Decision Support and Operations Research Team for the very nice working atmosphere. Special thanks go to Felix for all our important and not so important discussions and to Vera for the great time we had together in our office.

I am grateful to my friends and family for their support and company over the years. Insbesondere danke ich meinen Eltern und meiner Schwester Amelie für ihr Vertrauen in mich und für ihre Unterstützung unter anderem in der Logistik. Danke, Amelie, für deine fortwährende moralische Unterstützung während der letzten Jahre. Besonders danken möchte ich auch Jannik, der in den vier Jahren meines Doktorats viele Stunden im Zug (und während Covid auch in Quarantäne) verbracht hat. Danke für dein offenes Ohr, deine Unterstützung, deine Ruhe und dass du da bist.

Abstract

A *matching* in a graph $G = (V, E)$ is a set $M \subseteq E$, such that no two edges in M share an endvertex. An *edge cut* in G is a set of edges $C \subseteq E$, such that we can partition V into two non-empty sets R and B, where C is the set of edges with one endvertex in R and one in B.

A *matching cut* is a set of edges $M \subseteq E$ which is both a matching and an edge cut. A matching M is a *perfect matching* if $|M| = |V|/2$. A matching cut is a *perfect matching cut*, if it is a perfect matching. Further, a perfect matching cut containing a matching cut is a *disconnected perfect matching*.

In this thesis, we consider the decision problems MATCHING CUT, DISCONNECTED PERFECT MATCHING and PERFECT MATCHING CUT, asking if there exists a matching cut, a disconnected perfect matching, respectively a perfect matching cut in a given graph G. We show that MATCHING CUT and PERFECT MATCHING CUT are solvable in polynomial time for graphs of radius at most 2 and P_6-free graphs. Further, if they are solvable in polynomial time for H-free graphs, we get that this also holds for $(H + P_3)$-free graph, respectively $(H + P_4)$-free graphs. In addition, we show that PERFECT MATCHING CUT is solvable in polynomial time for bipartite graphs of diameter 3. Moreover, we show the NP-completeness of MATCHING CUT and DISCONNECTED PERFECT MATCHING for graphs of high girth, H_i^*-free graphs and P_r-free graphs, where $r = 15$ for MATCHING CUT, and $r = 19$ for DISCONNECTED PERFECT MATCHING.

We also consider the maximization versions MAXIMUM MATCHING CUT and MAXIMUM DISCONNECTED PERFECT MATCHING, where we ask for a matching cut, respectively a disconnected perfect matching of maximum size, that is with the maximum number of edges in the matching cut. For these two variants we give a complexity dichotomy for graphs of bounded radius, bounded diameter, H-free graphs and bipartite graphs of bounded radius and diameter.

We then focus on d-CUT, a generalization of MATCHING CUT. In the problem d-CUT, for some integer $d \geq 1$, we are looking for a partition of the vertex set into two non-empty sets such that every vertex is adjacent to at most d vertices in the other partition set. We give a complexity dichotomy for (bipartite) graphs of bounded radius and bounded diameter. Further, we show that d-CUT is solvable in polynomial time for $(P_3 + P_4)$-free graphs and P_5-free graphs. On the other hand, we show that the problem is NP-complete for graphs of high girth, $K_{1,4}$-free graphs, $3P_2$-free graphs (for $d \geq 3$) and $3P_3$-free graphs (for $d = 2$).

We conclude with an overview over all problems considered in this thesis. This overview allows us to compare known and new results and to identify interesting open problems.

Zusammenfassung

Eine *Paarung* eines Graphen $G = (V, E)$ ist eine Kantenmenge $M \subseteq E$, sodass es keine Kanten in M gibt, die einen gemeinsamen Knoten haben. Ein *Kantenschnitt* in G ist eine Kantenmenge $C \subseteq E$, sodass wir die Knoten V in zwei nichtleere Mengen R und B partitionieren können und C genau die Kanten enthält, die einen Endknoten in R und einen Endknoten in B haben.

Ein *Paarungsschnitt* ist eine Kantenmenge $M \subseteq E$, die sowohl eine Paarung als auch ein Kantenschnitt ist. Eine Paarung M nennen wir eine *perfekte Paarung*, wenn $|M| = |V|/2$. Ein Paarungsschnitt, der zusätzlich eine perfekte Paarung ist, ist ein *perfekter Paarungsschnitt*. Desweiteren bezeichnen wir als *vervollständigbaren Paarungsschnitt* einen Paarungsschnitt, der zu einer perfekten Paarung ergänzt werden kann.

In dieser Dissertation betrachten wir die drei Entscheidungsprobleme Paarungsschnitt, Vervollständigbarer Paarungsschnitt und Perfekter Paarungsschnitt. Das Problem besteht darin, zu entscheiden, ob es in einem gegebenen Graphen G einen Paarungsschnitt, einen vervollständigbaren Paarungsschnitt beziehungsweise einen perfekten Paarungsschnitt gibt.

Wir zeigen, dass Paarungsschnitt (beziehungsweise Perfekter Paarungsschnitt) für Graphen mit Radius maximal 2 und für P_6-freie Graphen in polynomieller Zeit gelöst werden können. Desweiteren zeigen wir, dass, falls diese Probleme in H-freien Graphen in polynomieller Zeit gelöst werden können, dies auch in $(H + P_3)$-freien (beziehungsweise $(H + P_4)$-freien) Graphen möglich ist. Außerdem beweisen wir, dass Perfekter Paarungsschnitt auch für bipartite Graphen mit Durchmesser maximal 3 in polynomieller Zeit gelöst werden kann. Anschließend beweisen wir, dass Paarungsschnitt und Vervollständigbarer Paarungsschnitt für Graphen mit fixierter Taillenweite, und P_r-freie Graphen NP-vollständig sind. Dabei gilt $r = 15$ für Paarungsschnitt und $r = 19$ für Vervollständigbarer Paarungsschnitt.

Wir betrachten weiterhin die beiden Maximierungsprobleme Maximaler Paarungsschnitt und Maximaler Vervollständigbarer Paarungsschnitt, bei denen ein Paarungsschnitt beziehungsweise ein vervollständigbarer Paarungschnitt maximaler Größe, das heißt mit maximaler Anzahl Kanten, gesucht wird. Für beide Probleme können wir eine Komplexitätsdichotomie für Graphen mit beschränktem Radius oder Durchmesser, für H-freie Graphen sowie für bipartite Graphen mit beschränktem Radius oder Durchmesser angeben.

Weiterhin fokussieren wir uns auf d-Schnitt, eine Verallgemeinerung von Paarungsschnitt. Beim Problem d-Schnitt, für ein $d \geq 1$, suchen wir eine Partitionierung der Knoten in zwei nichtleere Mengen, sodass jeder Knoten zu maximal d Knoten der anderen Menge benachbart ist. Wir geben eine Komplexitätsdichotomie dieses Problems für (bipartite) Graphen mit beschränktem Radius und Durchmesser an. Desweiteren zeigen wir, dass d-Schnitt für $(P_3 + P_4)$-freie und für P_5-freie Graphen in polynomieller Zeit lösbar ist. Andererseits zeigen wir, dass das Problem für Graphen mit fixierter Taillenweite, $K_{1,4}$-freie Graphen, $3P_2$-freie Graphen (für $d \geq 3$) und $3P_3$-freie Graphen (für $d = 2$) NP-vollständig ist.

Abschließend geben wir eine Übersicht über alle Probleme, die in dieser Arbeit behandelt wurden. Dieser Überblick ermöglicht es, bekannte und neue Resultate zu vergleichen und interessante offene Fragen zu identifizieren.

Contents

Chapter 1

Introduction

A *matching* in a graph $G = (V, E)$ is a set $M \subseteq E$, such that no two edges in M share an endvertex. An *edge cut* in G is a set of edges $C \subseteq E$, such that we can partition the vertex set V into two sets R and B, with $|R| \geq 1$, $|B| \geq 1$, $R \cup B = V$ and $R \cap B = \emptyset$, and C is exactly the set of edges with one endvertex in R and one in B. Both, matchings and edge cuts exist in every graph with at least two vertices and at least one edge. Any single edge is a matching and setting $R = \{v\}$ for some vertex $v \in V$ and $B = V \setminus \{v\}$ gives an edge cut consisting of every edge incident to v. Thus, the decision problems "Given a graph G, does G have a matching?" and "Given a graph G, does G have an edge cut?" are easy to solve. Hence, there is more interest in the optimization problems Maximum Matching, asking for a matching of maximum size, as well as Minimum Cut and Max Cut, asking for an edge cut of minimum size or of size at least k, for some integer $k \geq 1$, respectively. While Maximum Matching ([18]) and Minimum Cut ([22, 23]) are solvable in polynomial time on all graphs, Max Cut is NP-hard ([13, 24]) even on restricted graph classes like subcubic graphs ([60]). Since matchings and edge cuts are very well studied, it feels natural to combine them and to ask for an edge set which is both a matching and an edge cut, and which we call a *matching cut*. See Figure 1.1 (middle and right) and Figure 1.2 (left) for an example. The corresponding decision problem "Given a graph G, does G have a matching cut?" is called Matching Cut and it is an NP-complete problem (see [12]). However, this natural combination is not how matching cuts were introduced.

In 1970, Graham [30] wanted to solve a problem on cube-numbering. He introduced *decomposable* graphs as graphs with a matching cut, which he called a *simple cutset*. The aim was to minimize a given sum and he could give a relation between this sum and matching cuts on cubes. Thus, to solve the problem he was mainly interested in what he called *primitive* graphs, which are indecomposable graphs where all proper subgraphs are decomposable. As two first examples, he showed that the graph K_3 is indecomposable, that is, it has no matching cut and further that all cubes have a matching cut, see Figure 1.1. Even though the main purpose of introducing primitive graphs was to solve a problem on cube-numbering, he ends his paper with many open questions regarding primitive graphs.

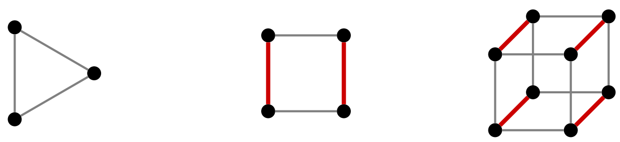


Figure 1.1: An indecomposable graph K_3 (left) and a matching cut (in bold) in the 2-dimensional cube (middle) and the 3-dimensional cube (right).

In 1982, another application of matching cuts appeared ([19]). Farley and Proskurowski were interested in communication networks which are immune to specific failures of vertices or edges. In [19] they consider *isolated line failure immune* networks (ILFI networks).

This are networks where the removal of any set of edges with pairwise different endvertices should not disconnect the graph. These networks are exactly the graphs that do not have a matching cut. In their paper, the authors construct several classes of graphs which are ILFI networks. These classes contain for example the graphs which arise from a tree after duplicating all edges as well as triangled networks, which are graphs where every edge is contained in a triangle. Further, they show how to construct an ILFI network with a minimum number of edges from a given tree.

There are several other applications than the two we already mentioned. For example, the fact that Matching Cut is an NP-complete problem was used to show the NP-completeness of Surjective Graph Homomorphism [27], where the target graphs are trees. Further, for WDM networks [2], finding graphs with *good edge-labellings* helps to solve the *Routing and Wavelength Assignment* (RWA) problem. In [2], the authors use the fact that the graphs which do not have a good edge-labelling, but where all subgraphs have a good edge-labelling, do not admit a matching cut. Thus, they can apply some structural theorems about matching cuts to better understand on which graphs the RWA problem can be solved.

The last application to mention here is graph drawing. In [14], the authors describe a method for 3D-drawings of graphs. One step is to split the drawing into two pieces. In their setting a matching cut would be a "perfect split" which does not require additional dummy vertices. Avoiding dummy vertices is good, since they make the final drawing more complicated. However, the fact that Matching Cut is NP-complete in most graph classes forces the authors to use some heuristic approach which results in a less good drawing but can be computed efficiently.

After its introduction by Graham in 1970, it took a few years until in 1984 a first paper focusing on matching cuts appeared. In this paper, Chvátal [12], who still used the name decomposable graphs, showed that recognizing a graph with a matching cut is an NP-complete problem. So it turned out that despite matchings and edge cuts exist in every graph, it is NP-complete to determine if a matching cut exists in a graph. A few years later, in 1989, Moshi [53] introduced the name "matching cut" for this problem. Over time, more results followed many of which concentrated on special graph classes, but also FPT results and exact algorithms were published. For an overview on known results we refer to Chapter 2. In this thesis, we focus on graphs of bounded diameter and radius, as well as on H-free graphs. Investigating the problem on different graph classes resulted in many open problems stated in the literature. Among them is the following question that Bonsma already asked in 2003 (see conference version of [6]).

Question 1 ([6]). Determine the complexity of Matching Cut on graphs with girth g, for $g > 5$.

This question was repeatedly given as an open problem, see [11, 39, 44, 49]. We will answer it in Theorem 5.1.4, where we show that Matching Cut is NP-complete for graphs of girth at least 3.

Figure 1.2: The graph P_6 with a matching cut (left), a disconnected perfect matching (middle) and a perfect matching cut (right).

With the increasing interest in the problem, different variants were introduced. A matching M of a graph G is a *perfect matching* if every vertex of G is incident to an edge of M. The first variant to be introduced was Perfect Matching Cut in [32], where we ask if there is a matching cut that is a perfect matching, see Figure 1.2 (right) for an example.

In [44], Le and Telle showed that Perfect Matching Cut is solvable in polynomial time for $S_{1,2,2}$-free graphs and asked the following question.

Question 2 ([44])**.** Let F be a fixed forest with maximum degree at most 3. What is the computational complexity of Perfect Matching Cut for F-free graphs?

Our results in Section 4.1.2, where we show for example that Perfect Matching Cut is solvable in polynomial time for P_6-free graphs, contribute to answering this question.

In the second variant, Disconnected Perfect Matching, we ask for the existence of a matching cut which is contained in a perfect matching, see Figure 1.2 (middle) for an example. The first result was given even before the introduction of Disconnected Perfect Matching as a variant of Matching Cut. In [16], Diwan considered disconnected 2-factors. Given a graph G, a *disconnected 2-factor* is a spanning subgraph H, consisting of at least two connected components, such that each vertex in H has degree exactly 2. Diwan showed that any planar cubic bridgeless graph G with at least 6 vertices has a disconnected 2-factor. Let H be a disconnected 2-factor in a cubic graph. Then, the set of edges not contained in H, that is $M = E(G) \setminus E(H)$, is a perfect matching in G. Since H is a disconnected 2-factor there is a minimal set $M' \subseteq M$ whose removal disconnects G. Since M' is a matching cut which can be extended to a perfect matching, we get a disconnected perfect matching. Thus, Diwan's result that any planar cubic bridgeless graph G with at least 6 vertices has a disconnected 2-factor also proved that every planar cubic bridgeless graph has a disconnected perfect matching. The first to consider this problem as a matching cut variant were Bouquet and Picouleau in [8]. They used the name perfect matching cut for it, but to avoid confusion the name disconnected perfect matching was introduced in [44] and used by other authors as well ([40, 50]). In [8], Bouquet and Picouleau showed among other results that Disconnected Perfect Matching is solvable in polynomial time for P_5-free graphs. They conclude their paper with several open problems for Disconnected Perfect Matching. Among them is the following question.

Question 3 ([8])**.** Determine the complexity of Disconnected Perfect Matching on P_k-free graphs, for $k \geq 6$.

We partially answer this question by proving that Disconnected Perfect Matching is solvable in polynomial time for P_6-free graphs in Theorem 4.2.8 and NP-complete for P_{19}-free graphs in Theorem 5.3.5. Note that the latter result got improved later in [40].

Other problems considered in this thesis are the very recent maximization variants Maximum Matching Cut and Maximum Disconnected Perfect Matching (see [51]). The problems are to find the maximum number of edges in a matching cut, respectively a disconnected perfect matching, if one exists. For both problems we give complete dichotomies for H-free graphs and (bipartite) graphs of bounded radius and diameter.

The last problem considered in this thesis is a generalization of Matching Cut. In the problem d-Cut, for some integer $d \geq 1$, we are looking for a partition of the vertex set into two sets such that every vertex is adjacent to at most d vertices in the other partition set. This problem was introduced by Gomes and Sau [29], who showed that it is NP-complete for any $d \geq 1$. Since the NP-completeness for d-Cut, for $d \geq 2$, on graphs of diameter 3 can be easily concluded from a result in [39], they ask the following question:

Question 4 ([29])**.** What is the complexity of d-Cut, for $d \geq 2$, for graphs of diameter at most 2?

In Theorem 6.2.1 we answer this question by showing that d-Cut is solvable in polynomial time for graphs of diameter at most 2.

Structure of this thesis In Chapter 2, we give the necessary preliminaries and detailed definitions of the problems dealt with in this thesis. We also give a more detailed overview about known results and the results contained in this thesis. In Chapter 3, we explain a colouring problem which is equivalent to the MATCHING CUT problem. We also consider the colouring equivalent for DISCONNECTED PERFECT MATCHING and especially for PERFECT MATCHING CUT, where we can make stronger observations. The results we obtain in Chapter 3 can be applied in Chapter 4, where we give the polynomial-time results. Our NP-hardness results follow in Chapter 5. Chapter 6 focusses solely on d-cut, for $d \geq 2$. We give again an equivalent colouring problem and prove both polynomial time and NP-completeness results. The results in Chapters 4, 5, and 6 lead to several complete and partial dichotomies. We conclude in Chapter 7 by giving these dichotomies which allow us to compare the variants and to identify interesting open problems.

Note that most of the results in this thesis have been published in [21, 48, 49, 50, 51]. The results have been reorganized in the above-mentioned chapters and for each result it is marked from which paper it was taken. The results in this thesis are joint work with Daniel Paulusma [21, 48, 49, 50, 51], Bernard Ries [21, 49, 50, 51], Carl Feghali [21], Siani Smith [48] and Ali Momeni [48].

Chapter 2

Preliminaries

2.1 General Notions

Graphs and subgraphs. A *graph* $G = (V, E)$ consists of a *vertex set* V together with an *edge set* $E \subseteq V \times V$. For an edge $e = \{u, v\}$ we say that u and v are the *endvertices* of e and that u and v are *incident* to e. Further, the vertices u and v are called *adjacent* if there exists an edge $\{u, v\}$, see also Figure 2.1. For simplicity, we normally write $e = uv$ for an edge $e = \{u, v\}$. Throughout this thesis, we only consider *finite* graphs, that is graphs with a finite number of vertices.

Let $G = (V, E)$ be a graph. A graph F is a *spanning* subgraph of G if $V(F) = V$ and $E(F) \subseteq E$. The *k-subdivision* of an edge $e = uv$ of G replaces e by k new vertices $w_1, \ldots, w_k$ and $k+1$ edges uw_1, uw_2, …, w_kv. 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 the disjoint union of s copies of a graph G by sG.

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 write $G - S = G[V \setminus S]$. Let $e \in E$ be an edge of G, then we write $G - e$ for the graph with vertex set V and edge set $E \setminus \{e\}$.

Neighbourhoods and degrees. The set $N_G(u) = \{v \in V \mid uv \in E\}$ is the *neighbourhood* of u in G, where $|N_G(u)|$ is the *degree* of u. A vertex of degree 1 is called a *leaf*. The *maximum degree* $\Delta(G)$ is the maximum of the degrees of all vertices of G. For an integer $p \geq 0$, G is *p-regular* if every $u \in V$ has degree p. A 3-regular graph is also called *cubic* and a graph of maximum degree at most 3 is called *subcubic*. Let $S \subseteq V$. The *neighbourhood* of S is the set $N(S) = \bigcup_{u \in S} N(u) \setminus S$. Further, we define the *closed neighbourhood* $N[S] = N(S) \cup S$. Let $T \subseteq V \setminus S$. The sets S and T are *complete* to each other if every vertex of S is adjacent to every vertex of T.

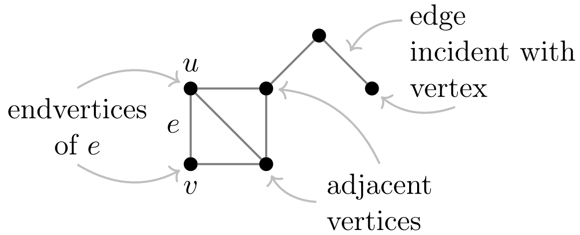


Figure 2.1: [9] A graph with definitions. The vertex v has degree 2 and u has degree 3 which is also the maximum degree of the graph.

Figure 2.2: A graph G with vertex sets S_1 (left) and S_2 (right) highlighted in blue. The sets S_1 and S_2 are both dominating and independent. While S_1 is a minimum dominating set, S_2 is a maximum independent set.

Paths and cycles. A *path* P is a graph with vertex set $V(P) = \{v_1, \ldots, v_n\}$ and edge set $E(P) = \{v_i v_{i+1} \mid i \in \{1, \ldots, n-1\}\}$. We say that the number of edges $n-1$ of P is the *length* of the path. For an integer $k \geq 1$, we denote by P_k the path on k vertices. See Figure 2.4 for an example of the graph P_4. Consider a graph $G = (V, E)$ and let $u, v \in V$. The *distance* between u and v in G, denoted by $\mathrm{dist}_G(u, v)$, is the length of a shortest path between u and v in G. If there is no path between u and v in G, then $\mathrm{dist}_G(u, v) = \infty$. If for any pair of vertices $u, v \in V$ we have that $\mathrm{dist}_G(u, v) < \infty$ we say that G is *connected*. Otherwise G is *disconnected*. A maximal connected induced subgraph of G is called a *connected component*. Further, we say that G is *k-connected*, if between any two vertices $u, v \in V$ there are at least k vertex disjoint paths, or equivalently if we have to delete at least k vertices to disconnect G. A *bridge* is an edge e of G such $G - e$ is disconnected.

A *cycle* C is a graph with vertex set $V(C) = \{v_1, \ldots, v_n\}$ and edge set $E(C) = \{v_i v_{i+1} \mid i \in \{1, \ldots, n-1\}\} \cup \{v_n v_1\}$. The *size* of a cycle is the number of its vertices. For an integer $k \geq 3$, we denote by C_k the cycle of size k. See Figure 2.4 for an example of the graph C_5. The cycle C_3 is also called a *triangle*. A graph without cycles is a *forest* and a connected forest is called a *tree*.

Graph parameters. We say that S is a *dominating* set of G, and that 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, see Figure 2.2. The set S is an *independent set* if no two vertices in S are adjacent, see again Figure 2.2, and S is a *clique* if every two vertices in S are adjacent.

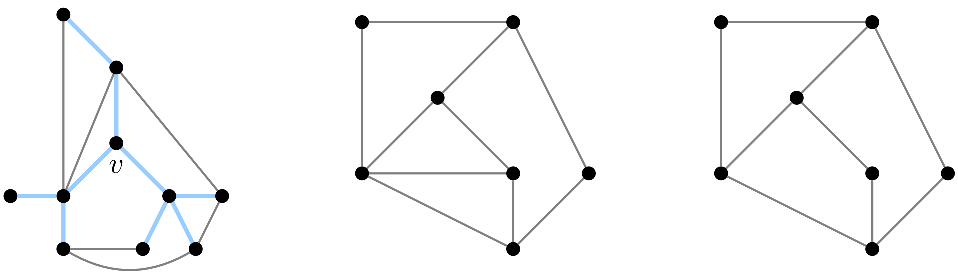


Figure 2.3: A graph with radius = 2 (left). The vertex v has distance at most 2 to all vertices of the graph. The graph in the middle has diameter = 2 and girth = 3, while the graph on the right has diameter = 3 and girth = 4.

Let $u, v \in V$. See Figure 2.3 for examples for the following definitions. 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, denoted by $\mathrm{diameter}(G)$. The *radius* of G is the minimum eccentricity over all vertices of G. We write $\mathrm{radius}(G)$ for the radius of G. Note that $\mathrm{radius}(G) \leq \mathrm{diameter}(G) \leq 2 \cdot \mathrm{radius}(G)$ and further, that a graph of radius at most 2 has a dominating star. The *girth* of G is the length of a shortest cycle in G. If G is a forest, we define $\mathrm{girth}(G) = \infty$.

The *edge expansion* $h(G)$ of a graph $G = (V, E)$ on n vertices is defined as

$$h(G) = \min_{1 \leq |S| \leq \frac{n}{2}} \frac{|\partial S|}{|S|},$$

where $\partial S := \{uv \in E \mid u \in S, v \in V \setminus S\}$ is the set of all edges of G with one endvertex in S and one endvertex outside S.

Graph classes and special graphs. Let G and H be two graphs. We say 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 $i \in \{1, \ldots, n\}$.

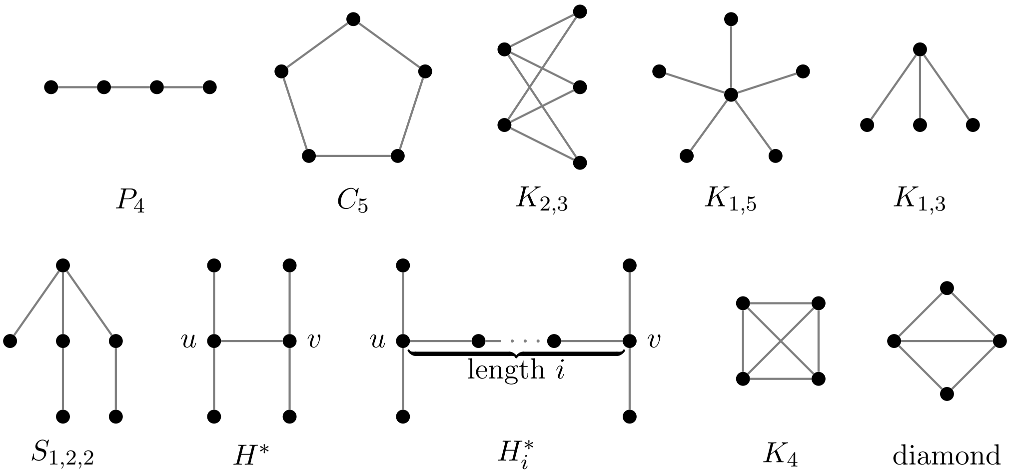


Figure 2.4: Examples for special graphs.

We refer to Figure 2.4 for examples of the following graphs. A graph $G = (V, E)$ is *bipartite* if its vertex set can be partitioned into two non-empty partition classes V_1 and V_2, such that both V_1 and V_2 are independent sets of 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 $|V_2| = \ell$, 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 observation, which was already made in [39].

Observation 2.1.1 ([39]). *Let $G = (V, E)$ be a bipartite graph of diameter at most 3 with partition classes V_1 and V_2. Let $u, v \in V_i$, for some $i \in \{1, 2\}$. Then u and v have a common neighbour.*

For examples of the following graphs, see again Figure 2.4. 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. The graph H_i^* can be obtained from H^* by subdividing the middle edge $i - 1$ times. The *complete* graph K_r is the graph on r vertices where all vertices are pairwise adjacent. We call the graph $K_4 - e$, that is the complete graph on 4 vertices with one missing edge, a *diamond*.

A graph is *chordal* if it is $\{C_4, C_5, \ldots\}$-free. Further, a graph is *quadrangulated* or *long-hole-free* if it is $\{C_5, C_6, \ldots\}$-free. The *chordality* of G is the length of a longest cycle in G. A graph of chordality k is said to be *k-chordal*. So a chordal graph is 3-chordal and a quadrangulated graph is 4-chordal. We call every bridge and every maximal 2-connected subgraph of a graph a *block* of the graph. We say that a graph is *pseudochordal* if for

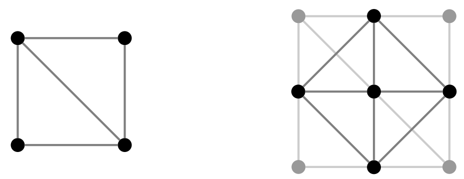


Figure 2.5: A graph G (left) and its linegraph $L(G)$ (right). The underlying graph is depicted in grey.

every block of size at least 3, every edge of the block is contained in a triangle. A graph is *planar* if it can be embedded in the plane, that is it can be drawn on the plane such that the edges do not cross. For example, the graphs in Figure 2.3 are all planar. Finally, 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 of $L(G)$, there exists an edge between e and f in $L(G)$ if and only if e and f share an endvertex in G. See Figure 2.5 for an example.

Number Theory. Let a, b be two positive integers. We call a a *divisor* of b if there exists an integer c, such that $a \cdot c = b$. A *common divisor* of a and b is an integer c which is a divisor of both a and b. An integer $p > 1$ is a *prime* number if it has exactly two divisors, namely 1 and p. Two integers a and b are *coprime*, if they do not have a common divisor greater than 1. The following result is due to Dirichlet.

Theorem 2.1.2 ([15])**.** *For two positive, coprime integers a and b, the sequence $a + bk$, for $k \in \mathbb{N}$ contains infinitely many primes.*

For an integer a and a prime p, the *Legendre symbol* is defined as $\left(\frac{a}{p}\right) \equiv a^{\frac{p-1}{2}} \bmod p$. It is well known (see, for example, [57]) that for any integer a and prime p, it holds that $\left(\frac{a}{p}\right) \equiv 0 \bmod p$, or $1 \bmod p$, or $(p-1) \bmod p$, and with slight abuse of notation one denotes these integers by 0, 1 and -1, respectively.

Computational Complexity. In the following we give the most important notions of computational complexity which are used in this thesis. For more details we refer to [25]. A *decision problem* is a problem with answer "yes" or "no". An *optimization problem* is a problem where some value has to be optimized and the answer is that optimal value. A function $f(n)$ is in $O(g(n))$ if there exists a constant $c \geq 0$ such that $f(n) \leq c \cdot g(n)$, for all n. Let n be the size of the input of an algorithm A. The *time complexity* of A is a function with n as variable and a value which gives the maximum amount of computation time over all instances of size n. An algorithm is a *polynomial-time algorithm* if there is a polynomial function $f(n)$ such that the time complexity of the algorithm is in $O(f(n))$, that is there exists $k \geq 0$ such that the algorithm is in $O(n^k)$. We say that a problem is *solvable in polynomial time* if there is a polynomial-time algorithm which solves the problem. A problem is said to be in P, if it is solvable in polynomial time. A problem is said to be in NP, if we can verify in polynomial time if a given solution is correct or not.

Consider two problems P_1 and P_2. If there exists a polynomial-time algorithm A, which transforms instances of P_1 to instances $A(P_1)$ of P_2, such that an instance I_1 of P_1 is a yes-instance of P_1 if and only if the instance $A(I_1)$ of P_2 is a yes-instance of P_2, then P_1 is *polynomial-time reducible* to P_2. A problem P is *NP-hard* if every other problem in NP is polynomial-time reducible to P. P is *NP-complete* if it is NP-hard and contained in NP.

The following problems are all NP-complete and will be used in the thesis to show other NP-hardness results.

A set $M \subseteq E$ is an *edge cut* of G if V can be partitioned into two sets B and R, such that M consists of all the edges with one endvertex in B and the other one in R, see

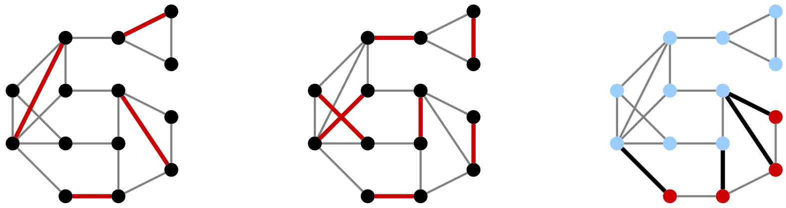


Figure 2.6: A graph G with a matching (left), a perfect matching (middle) and an edge cut (right), where the blue vertices are on one side of the cut and the red vertices on the other side.

Figure 2.6 (right). We will make use of the problem MAX CUT, which takes as input a graph G and an integer k. The question is whether G has an edge cut of size at least k.

Theorem 2.1.3 ([60]). MAX CUT *is* NP*-complete on subcubic graphs.*

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 $\mathcal{C}$. The EXACT 3-COVER problem has as input a set X with $3q$ elements, for some integer $q \geq 1$, and a collection $\mathcal{S}$ of 3-element subsets of X. The question is if $\mathcal{S}$ contains an exact 3-cover of X (which will be of size q).

Theorem 2.1.4 ([33]). EXACT 3-COVER *is* NP*-complete.*

Similarly, an *exact* 4*-cover* of a set X is a collection $\mathcal{C}$ of 4-element subsets of X, such that every $x \in X$ is in exactly one 4-element subset of $\mathcal{C}$. The EXACT 4-COVER problem has as input a set X with $4q$ elements, for some integer $q \geq 1$, and a collection $\mathcal{S}$ of 4-element subsets of X. The question is if $\mathcal{S}$ contains an exact 4-cover of X (which will be of size q). Using similar arguments than for EXACT 3-COVER, it can be easily seen that

Theorem 2.1.5. EXACT 4-COVER *is* NP*-complete.*

A *hypergraph* $H = (V, E)$ consists of a vertex set V and an edge set E, where an edge $e \in E$ is a set of vertices of H. The *size* of an edge in a hypergraph is the number of vertices it contains. The problem HYPERGRAPH 2-COLOURABILITY takes a hypergraph $H = (V, E)$ as input and asks if there exists a colouring of the vertices using 2 colours such that every edge of E contains at least one vertex of each colour.

Theorem 2.1.6 ([46]). HYPERGRAPH 2-COLOURABILITY *is* NP*-complete on hypergraphs in which all edges have size* 3.

We also need the following result from the literature, showing that there is a polynomial-time algorithm to find special dominating sets in P_6-free graphs. Note that this result has been strengthened in [10].

Theorem 2.1.7 ([59]). *A graph $G = (V, E)$ 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. Moreover, we can find such a dominating subgraph of G in $O(n^3)$ time.*

SAT. The problem k-SAT or k-SATISFIABILITY takes as input a pair (X, C), where X is a set of variables and C is a set of clauses. Every clause consists of a disjunction, that is, a boolean *and* operation, of k variables or negated variables, which are called *literals*. The conjunction, that is, a boolean *or* operation, of the clauses gives a formula ϕ. We ask if there is an assignment of the variables to the values true and false such that ϕ is satisfied. There are many polynomial-time algorithms known to solve 2-SAT. Krom (see [37]) was among the first to show the following theorem.

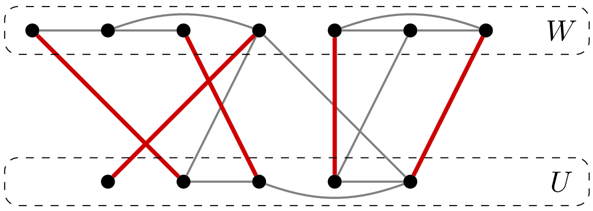


Figure 2.7: A graph with a U-saturating matching which is not W-saturating.

Theorem 2.1.8 ([37])**.** 2-SAT *is solvable in polynomial time.*

For $k \geq 3$ most variants of k-SAT are NP-hard. This is also the case for the following variant, that we will use in this thesis. The problem Exact Positive 1-in-3 SAT is a variant of 3-SAT where in addition to the restriction that every clause contains 3 literals we have that all these literals occur only positively. Moreover we may assume that each variable of X appears in exactly three clauses of C. The question is whether there exists a truth assignment, such that each clause contains exactly one true literal. This variant is NP-complete, see [56], Theorem 28.

Theorem 2.1.9 ([56])**.** Exact Positive 1-in-3 SAT *is NP-complete.*

We also need another variant of SAT. Like k-SAT, the problem Not-All-Equal SAT takes as input a pair (X, C), where X is a set of variables and C is a set of clauses. We ask for a *satisfying not-all-equal truth assignment* ϕ that is, ϕ sets, in each clause, at least one literal true and at least one literal false. This variant is well-known to be NP-complete. The theorem follows from a folklore trick; see e.g. [28].

Theorem 2.1.10 ([55])**.** Not-All-Equal SAT *is NP-complete even for instances, in which each literal occurs only positively and in two or three different clauses, and in which each clause contains either two or three literals.*

2.2 Matching Cuts

Matchings. A set $M \subseteq E$ is a *matching* if no two edges in M have a common endvertex, see Figure 2.6 (left). A matching M is *maximum* if there is no matching of G with more edges. Let $U \subseteq V$. A matching M is *perfect* if every vertex of G is incident to an edge of M, see Figure 2.6 (middle). A matching M is *U-saturating* if every vertex in U is an endvertex of an edge in M, see Figure 2.7 for an example. A U-saturating matching is *maximum* if there is no U-saturating matching of G with more edges. The following result of Plesník [54] will be used in the proofs of Section 4.2.

Theorem 2.2.1 ([54])**.** *For a graph G and a vertex set $U \subseteq V$, one can find a maximum U-saturating matching in polynomial time or conclude that G has no U-saturating matching.*

Matching Cuts. Let $M \subseteq E$. We say that M is a *matching cut* of G if M is a matching that is also an edge cut. We say that a matching cut M is a *perfect matching cut* if M is also a perfect matching. If M is a perfect matching of G which contains a matching cut M', we say that M is a *disconnected perfect matching* of G. We call a matching cut *maximum* if there is no matching cut of G containing more edges. Similarly we say that a disconnected perfect matching is *maximum* if there is no disconnected perfect matching, where the matching cut M' contained in it has more edges. See Figure 2.8 for examples. We consider the following three decision problems:

Figure 2.8: The graph P_6 with a matching cut of size 2 (left), a disconnected perfect matching of size 2 (middle) and a perfect matching cut (right). In each figure, thick edges denote matching cut edges.

Matching Cut
Input: a connected graph G
Question: does G have a matching cut?

Perfect Matching Cut
Input: a connected graph G
Question: does G have a perfect matching cut?

Disconnected Perfect Matching
Input: a connected graph G
Question: does G have a disconnected perfect matching?

Note that these decision problems are clearly contained in NP, so our NP-completeness proofs will focus on proving the NP-hardness. In the following we give an overview over results on Matching Cut, Perfect Matching Cut and Disconnected Perfect Matching.

Matching Cut. One of the first papers about Matching Cut after its introduction in 1970 is the paper of Chvátal from 1984 [12], where he proved that Matching Cut is polynomial-time solvable for subcubic graphs [12]. Later, Moshi showed that every connected subcubic graph on more than 7 vertices has a matching cut [53]. Over the years, more results appeared, giving polynomial-time algorithms for $K_{1,3}$-free graphs [6], $(K_{1,4}, K_{1,4} + e)$-free graphs [36] and quadrangulated graphs [53]. Note that the last class contains the class of chordal graphs. Further, it has been shown that Matching Cut is polynomial-time solvable on graphs of diameter at most 2 [7, 39], which we extend to graphs of radius at most 2 in Theorem 4.1.1. In Theorem 4.1.2 and Theorem 4.1.3 we extend the result of Feghali [20] that Matching Cut is solvable in polynomial time on P_5-free graphs to $(sP_3 + P_6)$-free graphs. Note that Feghali's result already generalized a result from Bonsma [6] for P_4-free graphs. In the same paper Bonsma showed that every planar graph with girth at least 6 has a matching cut. A recent result shows that Matching Cut is solvable in polynomial time for $S_{1,1,2}$-free graphs [42].

Chvátal [12] also showed an NP-completeness result, namely for graphs of maximum degree at most 4. It was pointed out in [6, 36] that Chvátal's construction is additionally $K_{1,4}$-free. Further, the NP-completeness has been shown for $K_{1,5}$-free bipartite graphs [43] and for graphs of diameter at most 3 [39]. Matching Cut is NP-complete in bipartite graphs where the vertices in one bipartition class all have degree exactly 2 [53]. From this result, the hardness follows for H_i^*-free graphs where $i \geq 1$ is odd. In Theorem 5.2.1 we extend this result to H_i^*-free graphs with even $i \geq 1$. In [6], Bonsma proved that Matching Cut is NP-complete on planar graphs of girth 5. We show in Theorem 5.1.4 that for graphs of girth at least 3, Matching Cut is NP-complete. Note that this is in contrast to the result of Bonsma for planar graphs of girth 6. Further, this answers Question 1. In [20], Feghali showed that there exists a constant integer $r > 0$ such that Matching Cut is NP-complete on P_r-free graphs. In Theorem 5.3.1 we improve this result and show the hardness for $(3P_5, P_{15})$-free graphs. After the publication of Theorem 5.3.1 in [50], Le and Le showed in [40] the hardness also for $(3P_6, 2P_7, P_{14})$-free graphs with diameter 4 and radius 3. Note that there are also results on parameterized and exact algorithms, see for instance [3, 11, 26, 29, 35, 36]. Among the first considered parameters was for example the vertex cover number (see [36]).

Perfect Matching Cut. The variant Perfect Matching Cut was introduced by Heggernes and Telle [32] who also showed that it is NP-complete. It can be verified that the graph they used to show the NP-completeness has diameter 6 and radius 3. A recent result by Le and Le [40] improves the bound on the diameter by showing the NP-completeness for $(3P_6, 2P_7, P_{14})$-free graphs with diameter 4 and radius 3. Note that they used the same construction to determine the complexity for Matching Cut, Perfect Matching Cut, and Disconnected Perfect Matching. Further, it is known that Perfect Matching Cut is NP-complete on 3-connected cubic planar bipartite graphs [5] and $K_{1,4}$-free bipartite graphs of girth g, for every $g \geq 3$ [44].

We also know that there are polynomial-time algorithms for $S_{1,2,2}$-free graphs (and thus for P_5-free and $K_{1,3}$-free graphs) [44], pseudo-chordal graphs [44] and quadrangulated graphs [41] (and thus for chordal graphs). These results will be extended to graphs of radius 2, $(sP_4 + P_6)$-free graphs and bipartite graphs of diameter 3 in Section 4.1.2.

Disconnected Perfect Matching. For the third variant of matching cuts, Diwan showed in [16] that any planar cubic bridgeless graph G with at least 6 vertices has a disconnected perfect matching. Bouquet and Picouleau showed in [8] that the problem is solvable in polynomial time for graphs of diameter 2, bipartite graphs of diameter 3, $K_{1,3}$-free graphs and P_5-free graphs. Corollary 4.2.9 and Theorem 4.2.10 generalize the last result to (sP_2+P_6)-free graphs. In [40], Le and Le show that Disconnected Perfect Matching is solvable in polynomial time for quadrangulated graphs.

On the other hand, also in [8], the authors showed that Disconnected Perfect Matching is NP-complete on graphs of diameter 3, bipartite graphs of diameter 4, $K_{1,4}$-free planar graphs, planar graphs of maximum degree 4, planar graphs of girth 5 and bipartite 5-regular graphs. Theorem 5.1.7 shows the NP-completeness of Disconnected Perfect Matching for graphs of girth at least g, with $g \geq 3$. In Theorem 5.3.5 we show NP-hardness on $(3P_7, P_{19})$-free graphs, which was published in [50]. Afterwards, Le and Le [40] published their result which shows hardness for $(3P_6, 2P_7, P_{14})$-free graphs. Recall that they use one gadget to prove the complexity of the three variants simultaneously.

Maximising Matching Cuts. In addition to the three decision problems above, we consider the following two optimization variants:

Maximum Matching Cut
Input: a connected graph G
Question: what is the maximum size of a matching cut of G?

Maximum Disconnected Perfect Matching
Input: a connected graph G
Question: what is the maximum size of a disconnected perfect matching of G?

These maximisation variants where we ask for a matching cut of maximum size or a disconnected perfect matching of maximum size were introduced recently. We note that if Matching Cut is NP-complete for some graph class, then Maximum Matching Cut is NP-hard for that graph class. Similarly, if Disconnected Perfect Matching is NP-complete on some graph class, then Maximum Disconnected Perfect Matching is NP-hard on that graph class. Furthermore, the NP-hardness of Perfect Matching Cut for some graph class implies the NP-hardness for both Maximum Matching Cut and Maximum Disconnected Perfect Matching for the same graph class. We also have some implication in the other direction. If MAXIMUM MATCHING CUT is solvable in polynomial time, then MATCHING CUT is solvable in polynomial time. Similarly if MAXIMUM DISCONNECTED PERFECT MATCHING is solvable in polynomial time then DISCONNECTED PERFECT MATCHING is solvable in polynomial time as well. Furthermore, if one of MAXIMUM MATCHING CUT and MAXIMUM DISCONNECTED PERFECT MATCHING is polynomial-time solvable for some graph class, then PERFECT MATCHING CUT is polynomial-time solvable for the same graph class.

With this knowledge, we can transfer some of the known results for MATCHING CUT, DISCONNECTED PERFECT MATCHING and PERFECT MATCHING CUT to MAXIMUM MATCHING CUT and MAXIMUM DISCONNECTED PERFECT MATCHING. For example, the result of Bonnet, Chakraborty and Duron [5] saying that PERFECT MATCHING CUT is NP-complete for 3-connected cubic planar bipartite graphs implies that the same holds for MAXIMUM MATCHING CUT. Also our result that MAXIMUM DISCONNECTED PERFECT MATCHING is solvable in polynomial time for P_6-free graphs (Theorem 4.2.8) shows that this holds as well for DISCONNECTED PERFECT MATCHING. However, apart from the result of Le and Le [40] for $(3P_6, 2P_7, P_{14})$-free graphs, which also holds for MAXIMUM MATCHING CUT, the results presented in Sections 4.2 and 5.3.2 were the first results published explicitly for these two problems. In these sections we give complete complexity dichotomies for MAXIMUM MATCHING CUT and MAXIMUM DISCONNECTED PERFECT MATCHING for graphs of bounded radius, bounded diameter and H-free graphs, as well as for bipartite graphs of bounded radius and diameter.

d-Cut. The last problem we consider is the d-CUT problem, a generalization of MATCHING CUT. For an integer $d \geq 1$ and a connected graph $G = (V, E)$, a set $M \subseteq E$ is a *d-cut* of G if V can be partitioned into two sets B and R, such that the following two conditions hold:

(i) M consists of all the edges with one endvertex in B and the other one in R; and

(ii) every vertex in B has at most d neighbours in R, and vice versa.

This leads to the following decision problem:

d-CUT
Input: a connected graph G
Question: does G have a d-cut?

Note that 1-CUT and MATCHING CUT are the same problems. Note further that d-CUT is clearly contained in NP, so our NP-completeness proofs will focus on proving the NP-hardness. The problem d-CUT was introduced only in 2021 by Gomes and Sau (see [29]), who showed that d-CUT is solvable in polynomial time for graphs of maximum degree at most $d+2$ and NP-complete when restricted to $(2d+2)$-regular graphs. As a side note they mention that the NP-completeness proof for MATCHING CUT on graphs of diameter 3 from Le and Le in [38] can be easily generalized to d-CUT, for $d \geq 2$. They also gave several exact and parametrized algorithms. We show that d-CUT is solvable in polynomial time for graphs of diameter 2 (Theorem 6.2.1), bipartite graphs of diameter 3 (Theorem 6.2.2), P_5-free graphs (Theorem 6.2.5) and $(P_3 + P_4)$-free graphs (Theorem 6.2.7). Further, if d-CUT is polynomial-time solvable for H-free graphs, then this holds as well for $(H + P_1)$-free graphs (Theorem 6.2.4). d-CUT is NP-complete for graphs of high girth (Theorem 6.3.2), $3P_3$-free graphs, for $d = 2$ (Theorem 6.3.4) and $3P_2$-free graphs, for $d \geq 3$ (Theorem 6.3.6). Moreover, it is NP-complete for $K_{1,4}$-free graphs (Theorem 6.3.7) and H_i^*-free graphs (Theorem 6.3.8). For $d \geq 3$, the stronger result that d-CUT is NP-complete for linegraphs has been shown in [52].

We need one last definition. A connected graph G is said to be *(matching-)immune* if G admits no matching cut. We also consider the following generalization for d-Cut. For an integer $d \geq 1$, we say that a connected graph G is *d-immune* if G admits no d-cut, so being 1-immune is the same as being immune. We make the following observation. Recall that $h(G)$ is the edge expansion of the graph.

Observation 2.2.2. *For every integer $d \geq 1$, every connected graph G with $h(G) > d$ is d-immune.*

Proof. Let $d \geq 1$, and let G be a connected graph with $h(G) > d$. For every set $S \subseteq V$ with $1 \leq |S| \leq \frac{n}{2}$, the average number of neighbours the vertices of S have in $V \setminus S$ is strictly larger than d. Hence, G cannot have a d-cut. In other words, G is d-immune. $\square$

Chapter 3

Colour Propagation

When working with matching cuts, we will often use an equivalent definition in terms of vertex colourings. This has been done by other authors before, see also [11, 12, 20].

3.1 Red-Blue Colourings

Let $G = (V, E)$ be a connected graph. A *red-blue colouring* of G colours every vertex of G either red or blue. If every vertex of some set $S \subseteq V$ has the same colour (red or blue), then S is said to be *monochromatic*. We also say that $G[S]$ is *monochromatic*. We say that a monochromatic set has *colour* red or blue if all its vertices are coloured red or blue, respectively. An edge $uv \in E$ is called *bichromatic* if u and v have different colours. A red-blue colouring of G is *valid* if the following holds:

1. every blue vertex has at most one red neighbour;
2. every red vertex has at most one blue neighbour; and
3. both colours red and blue are used at least once.

If a red vertex u in G has a blue neighbour v, then u and v are said to be *matched*. We refer to Figure 3.1 (left) for an example.

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, see Figure 3.1 (right).

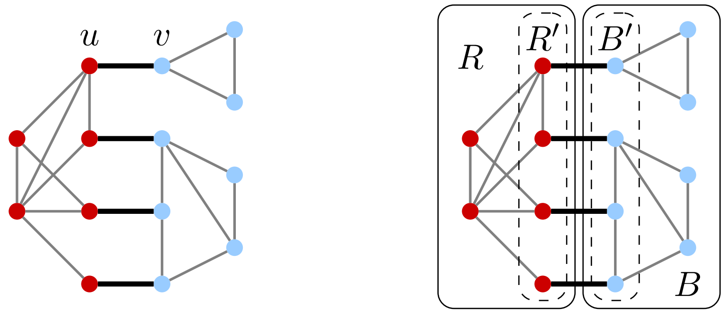


Figure 3.1: An example of a graph G with a valid red-blue colouring. The edge uv is bichromatic, u and v are matched to each other, and the thick edges form a matching cut. The sets R and B are the red and blue set, the sets R' and B' the red and blue interface.

Figure 3.2: Three different red-blue colourings of the graph P_6: a valid red-blue colouring of value 2 (left), a perfect-extendable red-blue colouring of value 2 (middle), a perfect red-blue colouring of value 3 (right).

A red-blue colouring of G is *perfect* if it is valid and moreover $R' = R$ and $B' = B$; see Figure 3.2 (right) and Figure 3.3 (right) for examples of perfect red-blue colourings. In other words, a valid red-blue colouring is perfect if every vertex is adjacent to exactly one vertex of the other colour. A valid red-blue colouring is *perfect-extendable* if there is a perfect matching in G containing all bichromatic edges, that is if $G[R \setminus R']$ and $G[B \setminus B']$ both contain a perfect matching; see Figure 3.2 (middle) and Figure 3.3 (middle) for examples of perfect-extendable red-blue colourings. In other words, the matching defined by the edges with one endvertex in R' and the other one in B' can be extended to a perfect matching in G or, equivalently, is contained in a perfect matching in G.

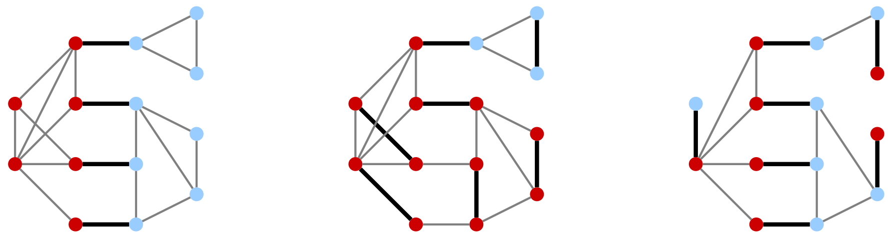


Figure 3.3: Three different red-blue colourings: a valid red-blue colouring of value 4 (left), a perfect-extendable red-blue colouring of value 1 (middle), a perfect red-blue colouring of value 7 (right).

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. Similarly, a perfect-extendable red-blue colouring is *maximum* if there is no perfect-extendable red-blue colouring of the graph with larger value.

We make two well known and very useful observations, see e.g., [20, 51]. We add a proof of the first one for completeness.

Observation 3.1.1. *For every connected graph G and integer $k \geq 1$, it holds that*

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.*

iv. *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.*

v. *G has a disconnected perfect matching with at least k edges belonging to a matching cut if and only if G has a perfect-extendable red-blue colouring of value at least k.*

Proof. i and v. Consider a matching cut M of G of size ν. We colour the vertices of one connected component of $G - M$ red and all other vertices of G blue, this clearly leads to a valid red-blue colouring of G of value ν. Consider now a valid red-blue colouring of value ν. We add all bichromatic edges to the cut. Since every vertex has at most one neighbour of the other colour this leads to a matching cut of size ν.

ii and v. Proceed as in the proof of *i.* Since the matching cut equals the set of bichromatic edges, a perfect matching contains a matching cut if and only if it contains the corresponding set of bichromatic edges. Further, this implies that the size of the matching cut equals the value of the colouring.
iii. Similar to *ii.*, a matching cut is a perfect matching if and only if the bichromatic edges of the corresponding colouring form a perfect matching. □

Observation 3.1.2. *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.*

In the following, we show that for each type of red-blue colouring, we can check whether it exists or not in polynomial time on graphs with small dominating sets. This will turn out to be very useful to solve corner cases. Note that the algorithm in Lemma 3.1.3 does not only find a maximum valid red-blue colouring, if one exists, but also answers the question if a valid red-blue colouring exists. Lemma 3.1.4 does the same for perfect-extendable red-blue colourings.

Lemma 3.1.3 ([51]). *For a connected graph G with n vertices and domination number g, it is possible to find a maximum red-blue colouring (if a valid red-blue colouring exists) in $O(2^g n^{g+2})$ time.*

Proof. Let D be a dominating set of G with $|D| \leq g$. We consider all $2^{|D|} \leq 2^g$ options of colouring the vertices of D red or blue. For every red vertex of D with no blue neighbour, we consider all $O(n)$ options of colouring at most one of its neighbours blue (and thus all of its other neighbours will be coloured red). Similarly, for every blue vertex of D with no red neighbour, we consider all $O(n)$ options of colouring at most one of its neighbours red (and thus all of its other neighbours will be coloured blue). Finally, for every red vertex in D with already one blue neighbour in D, we colour all its yet uncoloured neighbours red. Similarly, for every blue vertex in D with already one red neighbour in D, we colour all its yet uncoloured neighbours blue.

As D is a dominating set, the above means that we guessed a red-blue colouring of the whole graph G. We can check in $O(n^2)$ time if a red-blue colouring is valid and count its number of bichromatic edges. We take the valid red-blue colouring with largest value. The total number of red-blue colourings that we must consider is $O(2^g n^g)$. □

To show the corresponding result for Maximum Disconnected Perfect Matching, we simply copy the arguments of the proof of Lemma 3.1.3. Recall that we can check if a graph has a perfect matching in polynomial time by using, for instance, the Blossom algorithm [18]. Thus, for every valid red-blue colouring which we obtain in Lemma 3.1.3, we check if the graph induced by the unmatched vertices (that is, the vertices not incident to a bichromatic edge) has a perfect matching. This result is used implicitly in [51].

Lemma 3.1.4. *For a connected graph G with n vertices and domination number g, it is possible to find a maximum perfect-extendable red-blue colouring (if a red-blue colouring exists) in $O(2^g n^{g+2})$ time.*

The analogue of Lemma 3.1.3 for Perfect Matching Cut can again be obtained from the proof of Lemma 3.1.3 by colouring exactly one (instead of at most one) neighbour of each vertex with the other colour. Note that since any two perfect matching cuts in a graph have the same size, the maximization step is not necessary here.

Lemma 3.1.5 ([50]). *For a connected graph G with n vertices and domination number g, it is possible to find a perfect red-blue colouring (if a red-blue colouring exists) in $O(2^g n^{g+2})$ time.*

If we replace the condition that the dominating set D has constant size by the condition that D is monochromatic, we get a similar result, see for example [20] which uses it implicitly. We add a proof for completeness.

Lemma 3.1.6 ([49])**.** *Let D be a dominating set of a connected graph G. It is possible to check in polynomial time if G has a valid red-blue colouring in which D is monochromatic.*

Proof. Consider a valid red-blue colouring of G, in which D is monochromatic. Assume without loss of generality that every vertex of D is coloured red. Let C be a connected component of $G-D$. If C is not monochromatic, then C has an edge uv where u is red and v is blue. However, now v has at least two red neighbours, namely u and a neighbour in D (such a neighbour exists, as D is a dominating set of G). This is a contradiction. Hence, the vertex set of every connected component is monochromatic. Moreover, we may assume without loss of generality that the vertices of exactly one connected component of $G-D$ are coloured blue (else we can safely recolour the vertices of some connected component from blue to red). Hence, we can check all $O(n)$ options of choosing this unique blue connected component. For each option we check in polynomial time if the obtained red-blue colouring is valid. □

Note that with the same modification as for Lemma 3.1.4, we can show that the same holds for Disconnected Perfect Matching. However, these proofs do not work for the maximization problems Maximum Matching Cut and Maximum Disconnected Perfect Matching, since the assumption that colouring one of the connected components blue is sufficient does not hold then.

Lemma 3.1.7. *Let D be a dominating set of a connected graph G. It is possible to check in polynomial time if G has a perfect-extendable valid red-blue colouring in which D is monochromatic.*

To show the same result for Perfect Matching Cut, we adapt the proof of Lemma 3.1.6.

Lemma 3.1.8 ([50])**.** *Let D be a dominating set of a connected graph G. It is possible to check in polynomial time if G has a perfect red-blue colouring in which D is monochromatic.*

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

We claim that every F_i is monochromatic. This follows from the same argument as the one for valid red-blue colourings that are not necessarily perfect (see [49]) and we give it for completeness. For a contradiction, assume that, say, F_1 is not monochromatic. This means that F_1 contains an edge uv where u is coloured red and v is coloured blue. As D is a dominating set of G, we have that v also has a neighbour in D, which is coloured red. Hence, c is not valid and thus not perfect either, a contradiction.

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

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

3.2 Propagation of Partial Colourings

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 $X \cap Y = \emptyset$. Note that $V \setminus (X \cup Y)$ might be non-empty. A *red-blue*

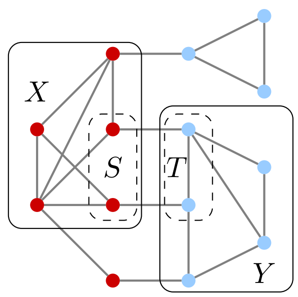


Figure 3.4: An example of a graph G with a valid red-blue (S,T,X,Y)-colouring.

(S,T,X,Y)-*colouring* of G is a red-blue colouring of all vertices in $X \cup Y$ and eventually additional vertices of G, such that the red set contains X; the blue set contains Y; the red interface contains S and the blue interface contains T. The red and blue interfaces may also contain vertices not in S and T, respectively. We refer to Figure 3.4 for an example.

To obtain a red-blue (S,T,X,Y)-colouring, we start with two non-empty, 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. For a starting pair, Le and Le [39] define five propagation rules that we give below – in our terminology – as rules R1 and R2. The goal of these rules is 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 $Z := V \setminus (X \cup Y)$.

We now state rules R1 and R2. 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 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.

Suppose 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

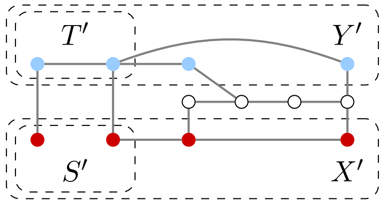


Figure 3.5: An example of a red-blue (S', T', X', Y')-colouring of a graph with an intermediate 4-tuple (S', T', X', Y').

intermediate tuple; see also Figure 3.5. 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 [39] proved the following lemma, which shows that R1 and R2 are safe and which is not difficult to verify. The fact that the value ν is preserved in Lemma 3.2.1 ii) below is implicit in their proof. Note that since ν is preserved when applying R1 and R2, they can also be applied when looking for a maximum valid red-blue colouring.

Lemma 3.2.1 ([39, 51])**.** *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'). Then:*

(i) It holds that $S^* \subseteq S'$, $S'' \subseteq X'$ 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.

(iii) It holds that

- *every vertex in S' has exactly one neighbour in Y', which belongs to T';*
- *every vertex in T' has exactly one neighbour in X', which belongs to S';*
- *every vertex in $X' \setminus S'$ has no neighbour in Y';*
- *every vertex in $Y' \setminus T'$ has no neighbour in 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 called *monochromatic* if all connected components of $G[Z]$ have to be monochromatic. Le and Le [39] proved a second lemma, using a reduction to 2-SAT. We slightly changed the formulation of their lemma so that it can be applied to the case where the graph G may also have a valid red-blue colouring which is not monochromatic.

Lemma 3.2.2 ([39, 49])**.** *Let G be a graph with a starting pair (S'', T'') with core (S^*, T^*). Assume that exhaustively applying rules R1 and R2 did not lead to a no-answer but to an intermediate tuple (S', T', X', Y'). It can be decided in $O(mn)$ time if G has a valid monochromatic red-blue (S', T', X', Y')-colouring (or equivalently, a valid monochromatic red-blue (S'', T'', S^*, T^*)-colouring).*

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.

We now present Rule R3 (which is used implicitly in [39]) 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 3.2.3 ([51])**.** *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. $\square$

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) a *final* tuple. The following lemma can be proven by a straightforward combination of the arguments of the proof of Lemma 3.2.1 with Lemma 3.2.3 and the observation that an application of R3 takes polynomial time, just as a check to see if R3 can be applied.

Lemma 3.2.4 ([51])**.** *Let G be a connected graph with a monochromatic intermediate tuple (S', T', X', Y') and a resulting final tuple (S, T, X, Y). Then:*

(i) It holds that $S' \subseteq S$, $X' \subseteq X$, $T' \subseteq T$, $Y' \subseteq Y$, 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.

(iii) It holds that

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

Note that Lemmas 3.2.1 and 3.2.4 can be adapted to perfect-extendable red-blue colourings in a straightforward way, since the propagation rules R1–R3 hold for valid red-blue colourings and every perfect-extendable red-blue colouring is valid. That is, the propagation of any partial red-blue colouring does not influence if the resulting partial red-blue colouring is perfect-extendable or not. In other words, we immediately obtain the following two lemmas. However, note that Lemma 3.2.2 can not be adapted for perfect-extendable colourings, since the condition that the graph needs to have a perfect matching cannot be modelled using 2-SAT.

Lemma 3.2.5 ([51])**.** *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 three statements hold:*

(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 perfect-extendable red-blue (S^*, T^*, S'', T'')-colouring of value ν if and only if G has a perfect-extendable red-blue (S', T', X', Y')-colouring of value ν (note that the backward implication holds by definition), and*

(iii) *every vertex in S' has exactly one neighbour in Y', which belongs to T'; every vertex in T' has exactly one neighbour in X', which belongs to S'; every vertex in $X' \setminus S'$ has no neighbour in Y'; every vertex in $Y' \setminus T'$ has no neighbour in 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.

Lemma 3.2.6 ([51]). *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 perfect-extendable (monochromatic) red-blue (S', T', X', Y')-colouring of value ν if and only if G has a perfect-extendable 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 has exactly one neighbour in Y, which belongs to T; every vertex in T has exactly one neighbour in X, which belongs to S; every vertex in $X \setminus S$ has no neighbour in Y and no two neighbours in the same connected component of $G[V \setminus (X \cup Y)]$; every vertex in $Y \setminus T$ has no neighbour in 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.

3.3 Perfect Red-Blue Colourings

3.3.1 Propagation of Partial Colourings

In Section 3.2 we presented rules R1 and R2 which hold for finding valid red-blue colourings in general. When considering perfect red-blue colourings, we can define many more rules, which will turn out to be useful. We keep the notation of starting pairs. For completeness, we state R1 and R2 again as RP1 and RP2. Rules RP3–RP7 are for finding perfect red-blue colourings; some of them are used in a slightly different form in [44].

RP1. 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 $Y \setminus T$.

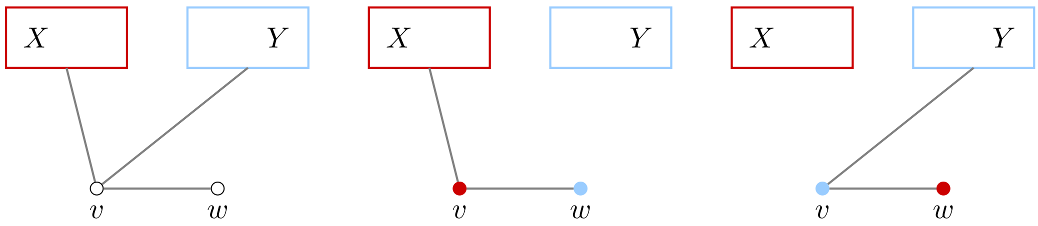
RP2. Let $v \in Z$.

(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.

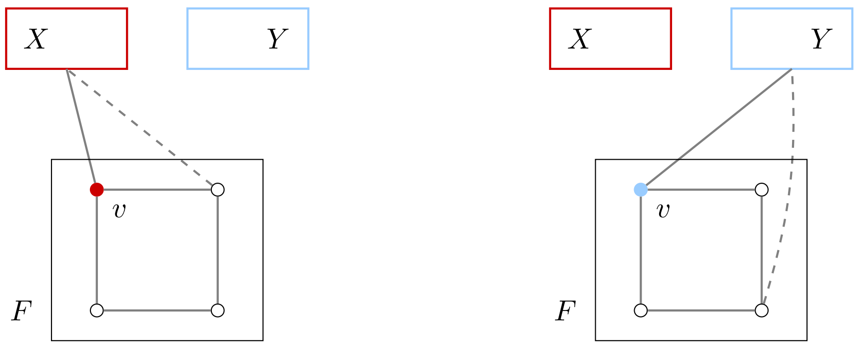
(ii) If v is adjacent to a vertex in T or to two vertices of $Y \setminus T$, then move v from Z to Y. If moreover v is also adjacent to a vertex w in X, then add v to T and w to S.

RP3. Let $v \in (X \cup Y) \setminus (S \cup T)$.

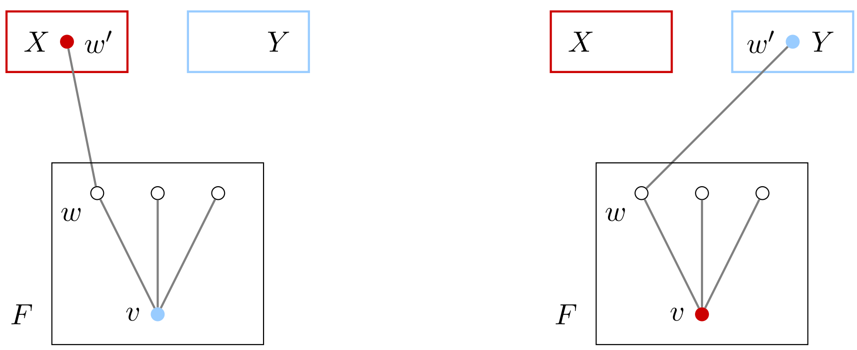
(a) RP4



(b) RP5



(c) RP6



(d) RP7

Figure 3.6: An illustration of the rules RP4–RP7.

(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.

RP4. Return `no` if

(i) a vertex $x \in X$ has no neighbours outside X or is adjacent to two vertices of Y (see Figure 3.6a), or

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

RP5. 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` (see Figure 3.6b (left)).

(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 (see Figure 3.6b (middle)).

(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 (see Figure 3.6b (right)).

RP6. 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 (see Figure 3.6c (left)).

(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 (see Figure 3.6c (right)).

RP7. 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 (see Figure 3.6d (left)).

(ii) If $w' \in Y$, then put v in X (see Figure 3.6d (right)).

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

Lemma 3.3.1 ([50])**.** *Rules RP1–RP7 are perfect safe.*

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

Rule RP1-(i) is perfect safe. A vertex adjacent to both a red vertex that already has a blue neighbour and to a blue vertex that already has a red neighbour can be coloured neither red nor blue. Rule RP1-(ii) is perfect safe, as a vertex that is adjacent to a red vertex that already has a blue neighbour must be coloured red, so it cannot also be adjacent to two blue vertices. For the same reason, RP1-(iii) is perfect safe. Finally, RP1-(iv) is perfect safe, as a vertex that is adjacent to two red vertices must be coloured red, so it cannot also be adjacent to two blue vertices.

Rule RP2-(i) is perfect safe. Any vertex adjacent to a red vertex that already has a blue neighbour or to two red vertices must be coloured red. If such a vertex is adjacent to a vertex coloured blue already, it will have its blue neighbour and thus must be added to S, whilst its blue neighbour must be added to T. For the same reason, RP2-(ii) is perfect safe as well.

Rule RP3-(i) is perfect safe. Every red vertex must have a (unique) blue neighbour, and vice versa. For the same reason, RP3-(ii) is perfect safe.

Rule RP4-(i) is perfect safe. In the first case, x will only have red neighbours in G (as x is coloured red, x needs a blue neighbour as well). In the second case, x will have two blue neighbours, while x is coloured red. This is not possible. For the same reason, RP4-(ii) is perfect safe as well.

Rule RP5-(i) is perfect safe. As v will be adjacent to a blue and red neighbour, w cannot be its matching neighbour. As w has degree 1, we find that w cannot be matched. Rule RP5-(ii) is perfect safe as well. For a contradiction, suppose that we would put v in Y. Then the matched neighbour of v is the neighbour of v that belongs to x. Hence, again we find that w does not have a matched neighbour. So we must put v in X, and then v and w will be matched to each other. For the same reason, RP5-(iii) is perfect safe.

Rule RP6-(i) is perfect safe. Suppose that we put v in Y, that is, v will be coloured blue. Consequently, v has its matching neighbour in X. This means that the neighbours of v in F will be coloured blue. As F has a cycle on four vertices as a spanning subgraph, v has at most one non-neighbour in F, and this non-neighbour will get colour blue as well. By assumption, F contains a vertex that is not adjacent to a vertex in X. This vertex is coloured blue, but will not have a red neighbour. We conclude that v must be put in X. For the same reason, RP6-(ii) is perfect safe as well.

Rule RP7-(i) is perfect safe. Suppose that we put v in X, so v will be coloured red, just like w'. Hence, w is adjacent to two red vertices, and must be coloured red as well. As w' is the only neighbour of w in $X \cup Y$, we find that every other neighbour of w is in F. As $\{v\}$ dominates F, this means that such a neighbour of w is adjacent not only to w but also to v, and hence must be coloured red. This means that w will not have any blue neighbour. We conclude that v must be put in Y. For the same reason, RP7-(ii) is perfect safe as well. □

Assume that exhaustively applying rules RP1–RP7 on a starting pair (S'', T'') did not lead to a no-answer but to a 4-tuple (S', T', X', Y'). Then we call (S', T', X', Y') an *intermediate* 4-tuple. We prove the following lemmas.

Lemma 3.3.2 ([50]). *Let G be a connected graph with a starting pair (S'', T'') with core (S^*, T^*) and a resulting intermediate 4-tuple (S', T', X', Y'). Then G has a perfect red-blue (S^*, T^*, S'', T'')-colouring if and only if G has a perfect red-blue (S', T', X', Y')-colouring. Moreover, (S', T', X', Y') can be obtained in polynomial time.*

Proof. The first part of the lemma follows from Lemma 3.3.1 and our initialisation. To prove the running time statement, we first note that each application of RP1–RP7 takes polynomial time. For each rule we can also check in polynomial time if it can be applied. Moreover, after each application of a rule we either find a no-answer or reduce the size of at least one of the sets X, Y, Z. Hence, we obtained (S', T', X', Y') in polynomial time. □

Lemma 3.3.3 ([50]). *Let G be a connected graph with an intermediate 4-tuple (S', T', X', Y'). Then:*

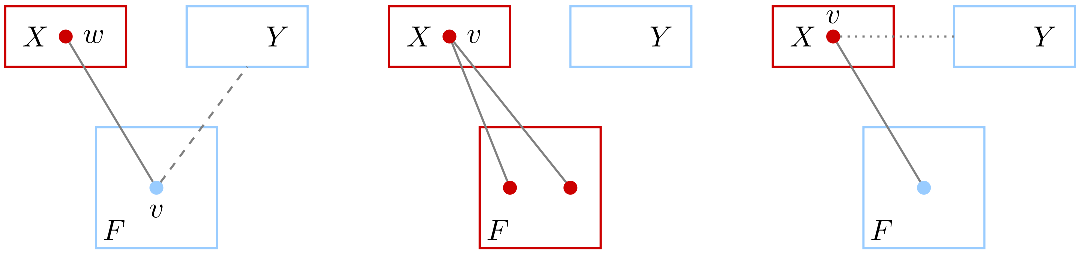
(i) every vertex in S' has exactly one neighbour in Y' and that neighbour belongs to T';

(ii) every vertex in T' has exactly one neighbour in X' and that neighbour belongs to S';

(iii) every vertex in $X' \setminus S'$ has no neighbour in Y';

(iv) every vertex in $Y' \setminus T'$ has no neighbour in X';

(v) every vertex in $V \setminus (X' \cup Y')$ has no neighbour in $S' \cup T'$, at most one neighbour in $X' \setminus S'$ and at most one neighbour in $Y' \setminus T'$.



(a) RP9 (b) RP10 (c) RP11

Figure 3.7: An illustration of the rules RP9–RP11.

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

Proof of (i). For a contradiction, first assume that some vertex u in S' has no neighbour in T'. Then u has no neighbour in $Y' \setminus T'$ either, else we would have applied RP3. However, now we would have applied RP4 (and returned a no-answer). Hence, every vertex in S' has a neighbour in $T' \subseteq Y'$. If a vertex in S' has more than one neighbour in Y', then we would have applied RP4 as well.

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

Proof of (iii). Let $u \in X' \setminus S'$. If u has a neighbour in Y', then we would have applied RP3. Hence, u has no neighbours in Y'.

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

Proof of (v). Let $u \in Z$. If u is adjacent to a vertex in $S' \cup T'$, then we would have applied RP2. If u is adjacent to two vertices in $X' \setminus S'$ or to two vertices in $Y' \setminus T'$, then we would also have applied RP2. □

3.3.2 Exploiting Monochromaticity

Let (S, T, X, Y) be an intermediate 4-tuple of a connected graph G. Let $Z = V \setminus (X \cup Y)$. Recall that a red-blue (S, T, X, Y)-colouring of G is monochromatic if all connected components of $G[Z]$ are monochromatic. Rules RP8–RP11 preserve this property; some of them were also used in [39, 44].

RP8. Let $v \in Z$. If v is not adjacent to any vertex of $X \cup Y$, then return `no`.

RP9. 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, see Figure 3.7a.

(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.

RP10. 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, see Figure 3.7b.

(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.

RP11. 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, see Figure 3.7c.

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

Lemma 3.3.4 ([50]). *Rules RP8–RP11 are perfect mono-safe.*

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

Rule RP8 is perfect mono-safe. Let F be the connected component of $G[Z]$ that contains v. Then all neighbours of v belong to F, which must be monochromatic. Thus the neighbours of v are coloured either all red or all blue. Hence, v will not have the required neighbour with a different colour than itself.

Rule RP9-(i) is perfect mono-safe. As all vertices in F will be coloured with the same colour, this means that w must receive a different colour than v. Hence, as w is coloured red, we find that v, and thus all other vertices of F, must be coloured blue. As every vertex of F only has neighbours in F and in $X \cup Y$, we find that all neighbours of every vertex in F are coloured. Hence, we can identify the unique red neighbours of the vertices of F, which in turn will be the unique blue neighbours of these vertices. For the same reason, Rule RP9-(ii) is perfect mono-safe as well.

Rule RP10-(i) is perfect mono-safe. All vertices in F will be coloured with the same colour and at least two of them are adjacent to v. Hence, the vertices in F must all get the same colour as the colour of v, which is red. Just as in the previous rule, we can now identify the unique blue neighbours of the vertices of F, which in turn will be the unique red neighbours of these vertices. For the same reason, Rule RP10-(ii) is perfect mono-safe as well.

Rule RP11-(i) is perfect mono-safe. Let w be the neighbour of v in F. Assume that $v \in X \setminus S$ and v is not adjacent to Y. As w is the only neighbour of v in Z, all other neighbours of v belong to X. As all vertices in X are coloured red, this means that w must be coloured blue. Hence, as every vertex of F will be coloured the same, every vertex of F will be coloured blue. Just as in the previous two rules, we can now identify the unique red neighbours of the vertices of F, which in turn will be the unique blue neighbours of these vertices. For the same reason, Rule RP11-(ii) is perfect mono-safe. □

Suppose that exhaustively applying rules RP1–RP11 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) a *final* 4-tuple and prove the following lemma.

Lemma 3.3.5 ([50]). *Let G be a connected graph with an intermediate 4-tuple (S', T', X', Y') and a resulting final 4-tuple (S, T, X, Y). Then G has a monochromatic perfect red-blue (S', T', X', Y')-colouring if and only if G has a monochromatic perfect red-blue (S, T, X, Y)-colouring. Moreover, (S, T, X, Y) can be obtained in polynomial time.*

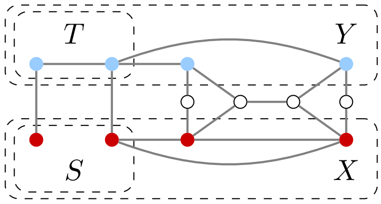


Figure 3.8: A red-blue (S, T, X, Y)-colouring of a graph with a final 4-tuple (S, T, X, Y). In this example, $G[Z]$ is isomorphic to $2P_1 + P_2$.

Proof. The first part of the lemma follows from Lemma 3.3.4 together with the fact that (S, T, X, Y) results from (S', T', X', Y'). To prove the running time statement, we first note that each application of RP1–RP11 takes polynomial time. For each rule we can also check in polynomial time if it can be applied. Moreover, after each application of a rule we either find a no-answer or reduce the size of at least one of the sets X', Y', Z'. Hence, we obtained (S', T', X', Y') in polynomial time. □

We now describe the structure of a connected graph with a final 4-tuple (S, T, X, Y); see Figure 3.8 for an example.

Lemma 3.3.6 ([50])**.** *Let G be a connected graph with a final 4-tuple (S, T, X, Y). The following holds:*

(i) every vertex in S has exactly one neighbour in Y, which belongs to T;

(ii) every vertex in T has exactly one neighbour in X, which belongs to S;

(iii) every vertex in $X \setminus S$ has no neighbour in Y, at least two neighbours in $V \setminus (X \cup Y)$ but no two neighbours in the same connected component of $G[V \setminus (X \cup Y)]$;

(iv) every vertex in $Y \setminus T$ has no neighbour in X, at least two neighbours in $V \setminus (X \cup Y)$ but no two neighbours in the same connected component of $G[V \setminus (X \cup Y)]$;

(v) every vertex of $V \setminus (X \cup Y)$ has no neighbour in $S \cup T$, exactly one neighbour in $X \setminus S$ and exactly one neighbour in $Y \setminus T$.

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

Proof of (i). For a contradiction, first assume that some vertex u in S has no neighbour in T. Then u has no neighbour in $Y \setminus T$ either, else we would have applied RP3. However, now we would have applied RP4 (and returned a no-answer). Hence, every vertex in S has a neighbour in $T \subseteq Y$. If a vertex in S has more than one neighbour in Y, then we would have applied RP4 as well.

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

Proof of (iii). Let $u \in X \setminus S$. If u has a neighbour in Y, then we would have applied RP3. Hence, u has no neighbours in Y. If u has no neighbours in $V \setminus (X \cup Y)$, then u would only have neighbours in X. In that case we would have applied RP4 (and returned a no-answer). If u only has one neighbour in Z, then we would have applied RP11. Hence, u has at least two neighbours in Z. If two neighbours of u in Z belong to the same connected component of Z, then we would have applied RP10.

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

Proof of (v). Let $u \in Z$. If u is adjacent to a vertex in $S \cup T$, then we would have applied RP2. Hence, we find that u is not adjacent to a vertex in $S \cup T$. If u is adjacent to two vertices in $X \setminus S$ or to two vertices in $Y \setminus T$, then we would have applied RP2 as well. If u has exactly one neighbour in $X \cup Y$, then we would have applied RP9. If u has no neighbour in $X \setminus S$ and no neighbour in $Y \setminus T$, then u has no neighbour in $X \cup Y$, as we already deduced that u has no neighbour in $S \cup T$. However, then we would have applied RP8 (and returned a no-answer). □

3.3.3 Reduction to 2-SAT

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

Lemma 3.3.7 ([50]). *Let G be a connected graph with a final 4-tuple (S, T, X, Y). Then it is possible to find in polynomial time a monochromatic perfect red-blue (S, T, X, Y)-colouring of G or conclude that such a colouring does not exist.*

Proof. Let $Z = V \setminus (X \cup Y)$. Let $E^* \subseteq E$ be the set of edges consisting of all edges with one endvertex in $(X \cup Y) \setminus (S \cup T)$ and the other endvertex in Z. By Lemma 3.3.6 v), we find that $|E^*| = 2|Z|$. By Lemma 3.3.6 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 3.3.6 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 [39], we now construct an instance ϕ of the 2-SAT problem. Our 2-SAT formula differs from the one in [39] 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 3.3.6 iii) and iv), C and D are different connected components of $G[Z]$. First assume that $u \in X \setminus S$. By Lemma 3.3.6 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 3.3.6 i), every vertex in S has exactly one neighbour in Y. By Lemma 3.3.6 ii), every vertex in T has exactly one neighbour in X. By Lemma 3.3.6 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 the opposite colour.

By Lemma 3.3.6 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 3.3.6 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 and D are coloured red, while the vertices of the other one are coloured blue. By Lemma 3.3.6 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. $\square$

Chapter 4

Polynomial-Time Algorithms Using Colour Propagation

In this chapter, we present algorithms to solve Matching Cut, Perfect Matching Cut, Maximum Matching Cut and Maximum Disconnected Perfect Matching on special graph classes. For Matching Cut we can extend the known results from graphs of diameter 2 ([7, 39]) to radius 2 (see Theorem 4.1.1) and from P_5-free graphs ([20]) to P_6-free graphs (see Theorem 4.1.2). Further, we show that if Matching Cut is solvable in polynomial time for H-free graphs, then it is solvable in polynomial time for $(H+P_3)$-free graphs (see Theorem 4.1.3). We give similar results for Perfect Matching Cut, see Theorem 4.1.4, 4.1.5 and 4.1.6. These results contribute to answering Question 2. We further show that Perfect Matching Cut is solvable in polynomial time for bipartite graphs of diameter 3 (see Theorem 4.1.7). For the maximization problems, we give the polynomial-time results needed to obtain the dichotomies for H-free graphs and (bipartite) graphs of bounded radius and diameter, see Section 4.2.

The algorithms in this chapter use the results from Chapter 3 on the propagation of partial colourings. That is, to obtain our results we first choose one or more partial colourings and propagate them. The resulting colourings have special properties which allow us to decide whether a valid red-blue colouring exists or not. This final step in the algorithms, where we determine the existence of a valid red-blue colouring is done using either 2-SAT, see Section 4.1, or saturating matchings, see Section 4.2.

4.1 Using 2-SAT

4.1.1 Matching Cut

We start with our results for Matching Cut. We show that Matching Cut is solvable in polynomial time for graphs of radius at most 2 (Theorem 4.1.1) and P_6-free graphs (Theorem 4.1.2), extending known results from [7, 20, 39]. The first result for graphs of radius 2 is helpful to show the result for P_6-free graphs. The proof is different from the two already known proofs for graphs of diameter 2 in [7, 39]. Further, we show that if Matching Cut is solvable in polynomial time for H-free graphs, then it is solvable in polynomial time for $(H + P_3)$-free graphs (Theorem 4.1.3). The proofs are rather straightforward. We first choose a suitable partial colouring. We show that after its propagation, the connected components of the uncoloured part of the graph have to be monochromatic. This allows us to apply Lemma 3.2.2. Note that the use of 2-SAT is hidden in Lemma 3.2.2.

Theorem 4.1.1 ([49]). Matching Cut *is polynomial-time solvable for graphs of radius at most* 2.

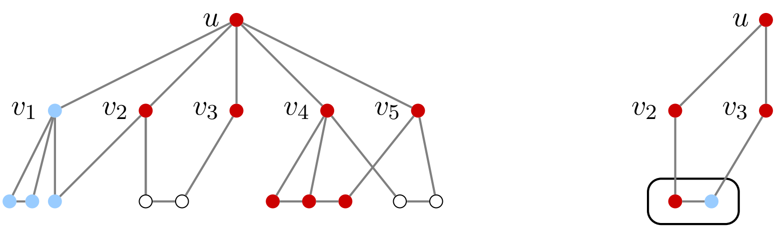


Figure 4.1: An illustration of the colouring procedure in a graph with radius 2. The neighbours of the blue vertex v_1 are all blue, except for u (left). In the figure on the right we can see that if a component is bichromatic, then there would be a blue vertex with two red neighbours.

Proof. Let G be a graph of radius at most $r \leq 2$. If $r = 1$, then G has a vertex that is adjacent to all other vertices. In this case G has a matching cut if and only if G has a vertex of degree 1; we can check the latter condition in polynomial time. 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 $s \geq 1$, see Figure 4.1. Note that H can be found in polynomial time by checking for each vertex if its neighbourhood is dominating. By Observation 3.1.1 it suffices to check if G has a valid red-blue colouring.

We first check if G has a valid red-blue colouring in which $V(H)$ is monochromatic. By Lemma 3.1.6 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 valid red-blue colouring of G (if it exists) colours u red and exactly one of $v_1, \ldots, v_s$ blue, see again Figure 4.1. That is, G has a valid red-blue colouring if and only if G has a valid 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 $S'' = S^* = \{u\}$ and $T'' = T^* = \{v_i\}$ and apply rules R1 and R2 exhaustively. The latter takes polynomial time by Lemma 3.2.1. If this exhaustive application leads to a no-answer, then by Lemma 3.2.1 we may discard the option. Suppose we obtain an intermediate tuple (S', T', X', Y'). By again applying Lemma 3.2.1, we find that G has a valid red-blue (S^*, T^*, S'', T'')-colouring if and only if G has a valid 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 valid red-blue (S', T', X', Y')-colouring c such that $G - (X' \cup Y')$ has a connected component D that is not monochromatic. Then D must contain an edge vw, where v is coloured blue and w is coloured red. Note that v cannot be adjacent to v_i, as otherwise v would have been in Y' by R2(ii) (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 w and w'), contradicting the validity of c, see Figure 4.1 (right).

From the above we conclude that for any valid (S', T', X', Y')-colouring (if it exists), every connected component $G - (X' \cup Y')$ is monochromatic. Hence, we can apply Lemma 3.2.2 to find in polynomial time whether or not G has a valid red-blue (S', T', X', Y')-colouring, or equivalently, if G has a valid red-blue (S^*, T^*, S'', T'')-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. □

Our next Theorem generalizes a result from Feghali, who showed that MATCHING CUT is

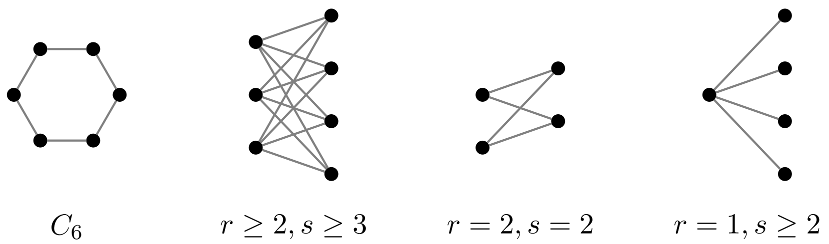


Figure 4.2: The different types of dominating sets in a P_6-free graph considered in the proof of Theorem 4.1.2.

solvable in polynomial time for P_5-free graphs. Feghali uses the fact that a P_5-free graph either has a small dominating set or a dominating clique, see for example [4]. For P_6-free graphs, using Theorem 2.1.7, we can also find dominating sets with helpful properties. We will see that the dominating set which we obtain from Theorem 2.1.7 is in most cases either small or monochromatic. This allows us to apply Lemma 3.1.3 and 3.1.6. However, we need to use Theorem 4.1.1, for the non-trivial case of a dominating star.

Theorem 4.1.2 ([49]). Matching Cut *is polynomial-time solvable for P_6-free graphs.*

Proof. Let G be a connected P_6-free graph. By Theorem 2.1.7, we find that G has a dominating induced C_6 or a dominating (not necessarily induced) complete bipartite graph $K_{r,s}$, see Figure 4.2 for all cases that may occur. By Observation 3.1.1 it suffices to check if G has a valid 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 3.1.3 to find in polynomial time if G has a valid red-blue colouring. Suppose that G has a dominating complete bipartite graph H 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 by using Observation 3.1.2 it can be seen that applying rules R1 and R2 on any starting pair $(\{u_i\}, \{v_j\})$ yields a no-answer. Hence, $V(H)$ is monochromatic for any valid red-blue colouring of G. This means that we can check in polynomial time by Lemma 3.1.6 if G has a valid 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 4.1.1. In the second case, $r \leq s \leq 2$, and thus G has domination number at most 4, and we apply Lemma 3.1.3. Hence, in both cases, we find in polynomial time whether or not G has a valid red-blue colouring. □

Our last algorithm for Matching Cut uses again the idea of colouring the graph partially to end in a situation where the uncoloured components have to be monochromatic. For this result, the arguments are slightly more involved.

Theorem 4.1.3 ([49]). *Let H be a graph. If* Matching Cut *is polynomial-time solvable for H-free graphs, then it is so for $(H + P_3)$-free graphs.*

Proof. Assume that Matching Cut is polynomial-time solvable for H-free graphs. Let G be a connected $(H + P_3)$-free graph with n vertices and m edges. We first check if G has a matching cut of size at most 2. We can do this in polynomial time by considering all $O(m^2)$ options of choosing two edges. From now on we assume that G has no matching cut of size at most 2; in particular this implies that G has no vertex of degree 1.

We may also assume that G has an induced subgraph G' that is isomorphic to H; else we are done by our assumption that Matching Cut is solvable in polynomial time for H-free graphs. Let G^* be the graph obtained from G after removing every vertex of

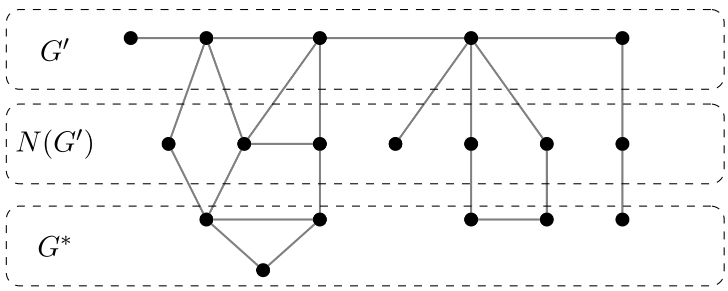


Figure 4.3: The decomposition of a graph G into the graphs G', which is isomorphic to H, the neighbourhood graph $N(G')$ of G', which is induced by all vertices not in G' but that have one or more neighbours in G', and the graph G^* induced by the remaining vertices of G. Note that since G^* is P_3-free it only consists of cliques.

$V(G') \cup N(V(G'))$. As G' is isomorphic to H and G is $(H + P_3)$-free, G^* is P_3-free, see Figure 4.3 for an example.

We continue as follows. We first consider all $O(m)$ options of choosing an edge from $E(G)$, one of whose endvertices we colour red and the other one blue. Afterwards, for each (uncoloured) vertex in G' we consider all options of colouring it either red or blue. As G' is isomorphic to H, there are $2^{|V(H)|}$ options of doing this. As H is a fixed graph, this is a constant number. Now, for every red vertex u of G' with no blue neighbour, we consider all $O(n)$ options of colouring at most one of its neighbours blue (and thus all other not yet coloured neighbours of u will be coloured red). Similarly, for every blue vertex v of G' with no red neighbour, we consider all $O(n)$ options of colouring at most one of its neighbours red (and thus all other neighbours of v will be coloured blue). Note that afterwards each vertex of $V(G') \cup N(V(G'))$ is either coloured red or blue.

There are $O(m2^{|V(H)|}n^{|V(H)|})$ options in total of colouring the endvertices of an edge in G and the vertices of G'. In each option, we have at least one red vertex and at least one blue vertex. We now consider the options one by one.

Consider an option as described above. In particular, let $e = uv$ be the chosen edge whose endvertices we coloured differently, say we coloured u red and v blue. We first check in polynomial time if every red vertex in this option has at most one blue neighbour and if every blue vertex has at most one red neighbour. If one of these two conditions does not hold, we discard the option. Now let S'' consist of u and all red vertices of $V(G') \cup N(V(G'))$, and let T'' consists of v and all blue vertices of $V(G') \cup N(V(G'))$. We let $S^* \subseteq S''$ consist of all red vertices that have (exactly) one blue neighbour, and we let $T^* \subseteq T''$ consist of all blue vertices that have (exactly) one red neighbour. By construction (recall that we started with picking an edge whose endvertices we coloured differently), $|S^*| = |T^*| \geq 1$. Hence, we can consider (S'', T'') as a starting pair with core (S^*, T^*).

Our algorithm will now check if G has a valid (S^*, T^*, S'', T'') red-blue colouring by applying rules R1 and R2 exhaustively. If we find a no-answer, then we can discard the option by Lemma 3.2.1. Otherwise, we found in polynomial time, again by Lemma 3.2.1, an intermediate tuple (S', T', X', Y'), for which the following holds:

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

(ii) G has a valid red-blue (S^*, T^*, S'', T'')-colouring if and only if G has valid red-blue (S', T', X', Y')-colouring, and

(iii) 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 $Y' \setminus T'$.

We now prove the following claim.

Claim. All connected components of $G-(X' \cup Y')$ are monochromatic in every valid red-blue (S', T', X', Y')-colouring of G.

Proof. For a contradiction, assume G has a valid red-blue (S', T', X', Y')-colouring, for which at least one connected component F of $G - (X' \cup Y')$ is not monochromatic. 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^* (recall that G^* is the P_3-free graph obtained from G after deleting the vertices of $V(G') \cup N(V(G'))$). Hence, F is P_3-free and thus as F is connected, F must be a complete graph. If F consists of one vertex or at least three vertices, then F must be monochromatic. Hence, F consists of exactly two vertices x and y.

By statement (iii), both x and y have no neighbour in $S' \cup T'$; at most one neighbour in $X' \setminus S'$ and at most one neighbour in $Y' \setminus T'$. If x and y have a common neighbour, then F must be monochromatic. If one of them, say x, has a neighbour in both $X' \setminus S'$ and a neighbour in $Y' \setminus T'$, then y must be coloured with the same colour as x and thus again F must be monochromatic. As G has minimum degree 2, this means that x and y each have exactly one neighbour in $X' \cup Y'$, and these two vertices in $X' \cup Y'$ must be different.

Let x' be the unique neighbour of x in $X' \cup Y'$, and let y' be the unique neighbour of y in $X' \cup Y'$, so $x' \neq y'$. However, now we find that G has a matching cut of size 2, namely the set $\{xx', yy'\}$, a contradiction. This completes the proof of the claim. $\lhd$

Due to the above claim, we can now check in polynomial time, by using Lemma 3.2.2, whether G has a valid red-blue (S', T', X', Y')-colouring in which every connected component in $G - (X' \cup Y')$ is monochromatic. If so, then we are done, and else we discard the option.

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

4.1.2 Perfect Matching Cut

In the following, we present our results for Perfect Matching Cut. First, we show in Theorem 4.1.4 that Perfect Matching Cut is solvable in polynomial time for graphs of radius at most 2. Then, we apply this result to show the same for P_6-free graphs (see Theorem 4.1.5). The next result, Theorem 4.1.6, shows that if Perfect Matching Cut is solvable in polynomial time for H-free graphs, then this is also the case for $(H + P_4)$-free graphs. The two latter results contribute to answering Question 2. Recall that in Question 2, Le and Le asked to determine the complexity of Perfect Matching Cut for H-free graphs, where H is a forest of maximum degree 3. Even though these results are similar to the results for Matching Cut, the proofs, especially the proof of Theorem 4.1.6, are much more involved. Afterwards, one last result follows, where we show that Perfect Matching Cut is solvable in polynomial time for bipartite graphs of diameter 3 (see Theorem 4.1.7).

Theorem 4.1.4 ([50])**.** Perfect Matching Cut *is polynomial-time solvable for graphs of radius at most* 2.

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

is adjacent to a neighbour of u. By Observation 3.1.1 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 3.1.8 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 $S'' = S^* = \{u\}$ and $T'' = T^* = \{v_i\}$. We now apply rules RP1–RP7 exhaustively. The latter takes polynomial time by Lemma 3.3.2. If this exhaustive application leads to a no-answer, then by Lemma 3.3.3 we may discard the option. Suppose we obtain an intermediate tuple (S', T', X', Y'). By again applying Lemma 3.3.3, 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 RP2-(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 vw, where v is coloured blue and w is coloured red. Note that v cannot be adjacent to v_i, as otherwise v would have been in Y' by RP3 (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 RP2-(i) and thus will be coloured red. However, now v being coloured blue is adjacent to two red vertices (namely w 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 RP1–RP11 exhaustively. The latter takes polynomial time by Lemma 3.3.5. If this exhaustive application leads to a no-answer, then by Lemma 3.3.6 we may discard the option. Suppose we obtain a final 4-tuple (S, T, X, Y). By again applying Lemma 3.3.6, 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 3.3.7 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. The proof is similar to the proof of Theorem 4.1.2. As a consequence of Theorem 2.1.7, a P_6-free graph either has a small dominating set, in which case we use Lemma 3.1.5, or a monochromatic dominating set, in which case we use Lemma 3.1.8, or it has radius at most 2, in which case we use Theorem 4.1.4.

Theorem 4.1.5 ([50]). Perfect Matching Cut *is polynomial-time solvable for P_6-free graphs.*

Proof. Let G be a connected P_6-free graph. By Theorem 2.1.7, we find that G has a dominating induced C_6 or a dominating (not necessarily induced) complete bipartite graph $K_{r,s}$. By Observation 3.1.1 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 3.1.5 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 by Observation 3.1.2 any starting pair of the form $(\{u_i\}, \{v_j\})$, $(\{u_i\}, \{u_j\})$ (if u_i and u_j are adjacent) or $(\{v_i\}, \{v_j\})$ (if v_i and v_j are adjacent) yields a no-answer. Hence, $V(D)$ is monochromatic for any perfect red-blue colouring of G. This means that we can check in polynomial time by Lemma 3.1.8 if G has a perfect red-blue colouring.

Now assume that $r = 1$ or $s \leq 2$. In the first case, G has a (not necessarily induced) dominating star and thus G has radius at most 2, and we apply Theorem 4.1.4. In the second case, $r \leq s \leq 2$, and thus G has domination number at most 4, and we apply Lemma 3.1.5. Hence, in both cases, we find in polynomial time whether or not G has a perfect red-blue colouring. □

For our next result for Perfect Matching Cut, we again apply the approach of choosing a suitable partial colouring, propagating this colouring and to reduce the problem to a 2-SAT instance. The restriction that for a perfect matching cut every vertex needs a neighbour of the other colour, allows us to prove a stronger result than for Matching Cut.

Theorem 4.1.6 ([50])**.** *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 4.4 for an example of this decomposition of G, where we have chosen $H = S_{1,2,2}$.

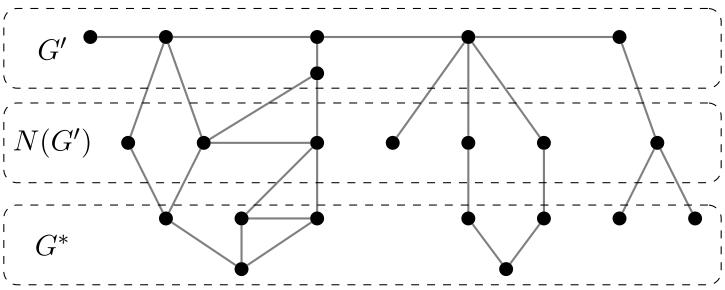


Figure 4.4: The decomposition of a graph G into the graphs G', which is isomorphic to H, the neighbourhood graph $N(G')$ of G', which is induced by all vertices not in G' but that have one or more neighbours in G', and the graph G^* induced by the remaining vertices of G.

We use Observation 3.1.1 and search for a perfect red-blue colouring. We define $n = |V(G)|$ and $m = |E(G)|$. Following our approach, we need a starting pair (S'', T'') with core (S^*, T^*). By definition, $|S^*| = |T^*| \geq 1$. Hence, we consider all $O(m)$ options of choosing an edge uv from $E(G)$, one of whose endvertices 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 RP1–RP7 exhaustively. If we find a no-answer, then we can discard the option by Lemma 3.3.3. Else

we found in polynomial time an intermediate tuple (S', T', X', Y'), such that G has a perfect red-blue (S^*, T^*, S'', T'')-colouring if and only if G has a perfect red-blue (S', T', X', Y')-colouring.
Consider a connected component F of $G - (X' \cup Y')$, for which the following holds:

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

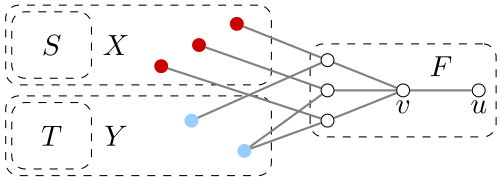


Figure 4.5: An example of a reducible connected component F of a graph G with an intermediate tuple (S', T', X', Y'). For readability only edges with at least one endvertex in F are drawn. Note that $F - \{u, v\}$ consists of three single-vertex components and that $F - \{u, v\}$ becomes a K_3 after 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 4.5 for an example.

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

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

The above operation is safe to do, as in any perfect red-blue (S', T', X', Y')-colouring of the new graph (if it exists) the vertices of $F - \{u, v\}$ will all be coloured alike. This is because the matching neighbour of every vertex of $F - \{u, v\}$ belongs to $X' \cup Y'$ and we have modified $F - \{u, v\}$ into a complete graph. We can now give v the same colour as the vertices of $F - \{u, v\}$ and u the opposite colour. In this way we obtain a perfect red-blue (S', T', X', Y')-colouring of G. Similarly, as we argued above, a perfect red-blue (S', T', X', Y')-colouring of G gives all the vertices of $F - \{u\}$ the same colour and makes u the matching neighbour of v. Hence, such a colouring (if it exists) will correspond to a perfect red-blue (S', T', X', Y')-colouring of the new graph.

We will also process any other reducible connected components of $G - (X' \cup Y')$ in the same way. Then either we found that the original graph G has no perfect red-blue (S', T', X', Y')-colouring and we discard the option, or we found a new graph that has a perfect red-blue (S', T', X', Y')-colouring if and only if G has a perfect red-blue (S', T', X', Y')-colouring. Assume that we are in the latter situation. We continue with the new graph and denote it by G again (note that the new graph is the same graph as G if G had no reducible connected components). We now prove the following claim.

Claim 4.1.6.1. *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 [34]) that every connected P_4-free graph has a spanning complete bipartite subgraph $K_{k,\ell}$ for some integers $1 \leq k \leq \ell$.

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

We conclude that every vertex of F is adjacent to at most one vertex of $X' \cup Y'$. As G is connected, F has at least one vertex v with a neighbour w in $X' \cup Y'$, say without loss of generality that $w \in X'$. Then the other three vertices of F must also have a neighbour in X' (and thus no neighbour in Y), else we would have applied RP6. The only way we can extend the red-blue (S', T', X', Y')-colouring to a perfect red-blue colouring of G is by colouring each vertex of F blue, so F is monochromatic.

It remains to consider the case where $k = 1$ and $\ell \geq 1$. In this case F contains a vertex v such that $\{v\}$ dominates F. Then every vertex in $F - v$ has either no neighbours in $X' \cup Y'$, or it has both a neighbour in X' and a neighbour in Y'; else we would have applied RP7. Let U be the set of vertices in $F - v$ with no neighbour in $X' \cup Y'$. As $\{v\}$ dominates F, every connected component of $F - v$ is monochromatic. So, v is the matching neighbour of every vertex of U. Hence, if $|U| \geq 2$, then G has no perfect red-blue (S', T', X', Y')-colouring so the claim is true. If $|U| = 0$, then the vertices in $F - v$ all have a neighbour both in X' and Y'. So, they do not have their matching neighbour in F and thus will receive the same colour as v. Hence, F is monochromatic. Assume that $|U| = 1$, say $U = \{u\}$ for some vertex u of G. As v is the matching neighbour of u, we find that v is adjacent to at most one vertex of $X' \cup Y'$.

We now have that F contains two adjacent vertices u and v, where u has no neighbours in $X' \cup Y'$ and moreover, v is dominating F, and every vertex in $F - \{u, v\}$ has both a neighbour in X' and a neighbour in Y'. Recall that F is not reducible. Hence, v is adjacent to exactly one vertex w of $X' \cup Y'$. Then u has at least one neighbour in $F - v$; else we would have applied RP5. 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. $\triangleleft$

We now apply rules RP1–RP11 exhaustively. This takes polynomial time by Lemma 3.3.5. If this leads to a no-answer, then by Lemma 3.3.6 we may discard the option. Suppose we obtain a final 4-tuple (S, T, X, Y). By Lemma 3.3.5, 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 3.3.7 to find in polynomial time if the latter holds. If so, we are done by the Claim, else we discard the option.

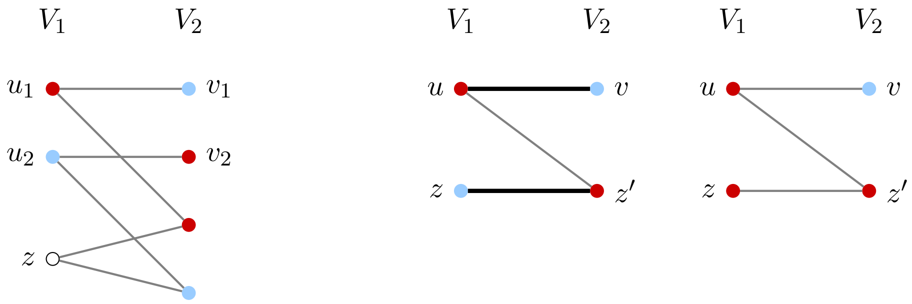


Figure 4.6: An illustration of the colouring in Phase 1 (left) and the colouring in Phase 2 (middle, right) in the proof of Theorem 4.1.7. Note that for Phase 2 the colouring in the middle leads to a contradiction to the fact that Phase 1 failed.

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

The proof of our last result in this section is based on a known result. In [39], Le and Le showed that Matching Cut is solvable in polynomial time for bipartite graphs of diameter 3. We adapt their proof to show that Perfect Matching Cut is solvable in polynomial time for bipartite graphs of diameter 3.

Theorem 4.1.7. Perfect Matching Cut *is polynomial-time solvable for bipartite graphs of diameter at most* 3.

Proof. Let $G = (V, E)$ be a connected bipartite graph of diameter at most 3 with partition classes V_1 and V_2. We may assume that $|V_1|, |V_2| \geq 2$, since otherwise it is trivial to decide if G has a perfect matching cut or not. This implies that any perfect matching cut in G has at least 2 edges, since it has size $|V|/2$. The proof consists of two phases. In Phase 1 we only search for perfect matching cuts with a special structure. In Phase 2 we consider the case where no perfect matching cut was found in Phase 1.

Phase 1: We branch over all $O(n^4)$ options of choosing two edges $u_1v_1, u_2v_2 \in E$, with $u_1, u_2 \in V_1$ and $v_1, v_2 \in V_2$, and we colour u_1 and v_2 red and v_1 and u_2 blue. We construct a starting pair (S'', T'') with core (S^*, T^*). Let $S'' = S^* = \{u_1, v_2\}$ and let $T'' = T^* = \{v_1, u_2\}$. We apply rules RP1–RP7 exhaustively. If we find a no-answer, then we can discard the option by Lemma 3.3.3. Else we found in polynomial time an intermediate 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. Let $Z = V \setminus (X' \cup Y')$.

Claim 4.1.7.1. *Every connected component of $G[Z]$ in every perfect red-blue (S', T', X', Y')-colouring of G is monochromatic.*

Proof. Let $z \in Z$. Assume first that $z \in V_1$. By Observation 2.1.1 we know that z and u_1 have a common neighbour. Further, since $u_1 \in S^*$, we get from Rule RP2 that the common neighbour of z and u_1 is red. Similarly, z and u_2 have a common neighbour which is blue. So every uncoloured vertex in V_1 has both a red and a blue neighbour, see Figure 4.6 (left). Note that by symmetry the same follows if $z \in V_2$. Thus, every vertex $z \in Z$ has at least one red and one blue neighbour. Let zz' be an edge of $G[Z]$. If we colour z blue and z' red, then z has two red neighbours but is blue, a contradiction. We conclude that the claim is true. $\lhd$

We now apply rules RP1–RP11 exhaustively. This takes polynomial time by Lemma 3.3.5. If this leads to a no-answer, then by Lemma 3.3.6 we may discard the option. Suppose we obtain a final 4-tuple (S, T, X, Y). By Lemma 3.3.5, G has a monochromatic perfect (S', T', X', Y')-colouring if and only if G has a monochromatic perfect (S, T, X, Y)-colouring. We apply Lemma 3.3.7 to find in polynomial time if the latter holds. If so, we are done by the claim, else we discard the option and proceed with the next branch.

If one of the considered branches lead to a perfect red-blue colouring, we are done. Otherwise we may assume that there does not exist a perfect red-blue colouring of G with two edges $u_1v_1, u_2v_2 \in E$, with $u_1, u_2 \in V_1$ and $v_1, v_2 \in V_2$, such that u_1 and v_2 are coloured red and v_1 and u_2 are coloured blue. That is, Phase 1 failed and we start Phase 2.

Phase 2: We branch over all options of choosing an edge $uv \in E$, with $u \in V_1$ and $v \in V_2$. We colour u red and v blue. We construct a starting pair (S'', T'') with core (S^*, T^*). Let $S'' = S^* = \{u\}$ and let $T'' = T^* = \{v\}$. We apply rules RP1–RP7 exhaustively. If we find a no-answer, then we can discard the option by Lemma 3.3.3. Else we found in polynomial time an intermediate 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 an uncoloured vertex $z \in V_1$. By Observation 2.1.1, we know that z has a common neighbour z' with u. Note that z' was coloured red by propagation. If we colour z blue, then the existence of a perfect red-blue colouring of G where we have coloured the two edges zz' and uv as described above would contradict the assumption that Phase 1 failed, see Figure 4.6 (middle). Thus, z has to be red and similarly, if we consider an uncoloured vertex $z \in V_2$ we can conclude that z has to be blue, see Figure 4.6 (right). Hence, we can colour all remaining vertices of G and check in polynomial time if this colouring is a perfect red-blue colouring. If this is the case, we are done, otherwise we discard the branch and consider the next.

If, after considering all branches in Phase 2, we did not find a perfect red-blue colouring, then G does not have a perfect red-blue colouring and thus no perfect matching cut. The correctness of our algorithm follows from its description. As the total number of branches is $O(n^4)$ and we can process each branch in polynomial time, the total running time of our algorithm is polynomial. Hence, we have proven the theorem. □

4.2 Using Saturating Matchings

In this section, we prove polynomial-time results for Maximum Matching Cut and Maximum Disconnected Perfect Matching which are based on finding special matchings. Our results imply that both problems are solvable in polynomial time for $(sP_2 + P_6)$-free graphs, which gives the polynomial part of the dichotomies for H-free graphs. See Theorem 4.2.2, Theorem 4.2.3, Theorem 4.2.8 and Theorem 4.2.10 for the corresponding results. Since Theorem 4.2.8 implies Corollary 4.2.9, we get that Disconnected Perfect Matching is solvable in polynomial time for P_6-free graphs and thus, answer partially Question 3. We further show that Maximum Matching Cut and Maximum Disconnected Perfect Matching are solvable in polynomial time for graphs of diameter at most 2, bipartite graphs of radius at most 2 and bipartite graphs of diameter at most 3. These results give the polynomial part of the dichotomies for (bipartite) graphs of bounded radius and diameter.

In the proofs, we use again the approach of choosing a partial colouring that we propagate. In all cases, we can show that the vertices which remain uncoloured after the preprocessing form an independent set. Thus, the final step will be to consider saturating matchings. The Lemmas 4.2.1 and 4.2.7 are the key lemmas of the section since they show how maximum valid red-blue colourings and saturating matchings are related.

4.2.1 Maximum Matching Cut

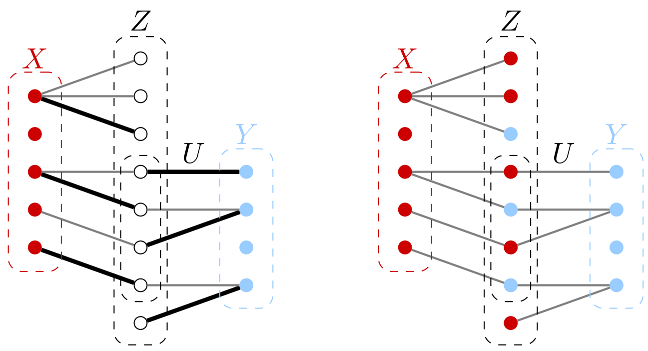


Figure 4.7: A U-saturating matching (left) and the corresponding valid red-blue colouring (right). Note that not every vertex in $X \cup Y$ belongs to $N(Z)$.

The following lemma, which is an application of Theorem 2.2.1 from Plesník, will be the final step in each of the polynomial-time results for Maximum Matching Cut.

Lemma 4.2.1 ([51])**.** *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 valid red-blue (S, T, X, Y)-colouring.*

Proof. Let $Z = V \setminus (X \cup Y)$. Let $W = N(Z)$. Recall that Z is independent. Hence, by Lemma 3.2.4 iii), every vertex of W belongs to $(X \setminus S) \cup (Y \setminus T)$. Let $U \subseteq Z$ consist of all vertices of Z that have 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 endvertex 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 4.7.

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 3.2.4 iii), every edge with an endvertex 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 in G. 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 2.2.1, this takes polynomial time. $\square$

We are now ready to give a first application of Lemma 4.2.1. Note that Theorem 4.2.2 is a stronger version of Theorem 4.1.2. Its proof uses more involved arguments.

Theorem 4.2.2 ([51]). 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 3.1.1 it suffices to find a maximum valid red-blue colouring of G. By Theorem 2.1.7, we find in polynomial time either a dominating induced C_6 or a dominating (not necessarily induced) complete bipartite graph $K_{r,s}$ in G.

If G has a dominating induced C_6, then G has domination number at most 6, and we apply Lemma 3.1.3. Suppose that G has a dominating complete bipartite graph F with partition classes $\{u_1, \dots, u_r\}$ and $\{v_1, \dots, v_s\}$. We may assume without loss of generality that $r \le s$. If $s \le 2$, then G has domination number at most 4, and we apply Lemma 3.1.3 again. So we assume that $s \ge 3$.

If $r \ge 2$, then $V(F)$ must be monochromatic in any valid red-blue colouring of G by Observation 3.1.2. In this case we colour every vertex of $V(F)$ blue. If $r = 1$, then we may assume without loss of generality that $N(u_1) = \{v_1, \dots, v_s\}$. 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 $r \ge 2$). 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 endvertex 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 3.2.1 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. We prove the following two claims for the set $Z' = V \setminus (X' \cup Y')$ of uncoloured vertices.

Claim 4.2.2.1. *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 4.2.2.2. *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, by Claim 4.2.2.1, v has two blue neighbours, u and some vertex $w \in Y' \setminus T'$, a contradiction. ◁

Since Claim 4.2.2.2 holds, we may now exhaustively apply R1–R3 to the intermediate tuple (S', T', X', Y'). By Lemma 3.2.4 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 4.2.2.1, 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

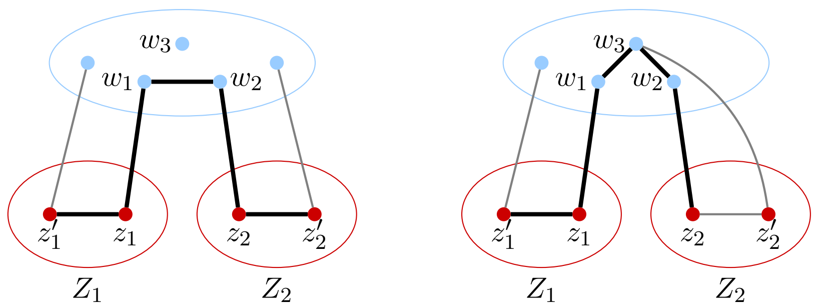


Figure 4.8: The situation in Claim 4.2.2.3 where two connected components Z_1, Z_2 of $G[Z]$, each with at least two vertices, are both coloured red. This will always yield an induced path on at least six vertices, even if w_1 and w_2 are not adjacent, as at most one of z_1', z_2' is adjacent to w_3.

- 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 4.2.2.3. *In any valid red-blue (S, T, X, Y)-colouring at most one red component of $G[Z]$ 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 4.2.2.1). 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 $z_1'z_1w_1w_2z_2z_2'$ is an induced P_6. As G is P_6-free, this is not possible. Hence, w_1 and w_2 are not adjacent, see Figure 4.8 (left).

We now use the fact that w_1 and w_2 both belong to F and that F is a complete bipartite graph. As $w_1w_2 \notin E$, the latter means that there exists a vertex $w_3 \in V(F)$ that is 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, $z_1'z_1w_1w_3w_2z_2$ is an induced P_6, a contradiction. See also Figure 4.8 (right). ◁

Due to Claim 4.2.2.3, we can now branch over all $O(n)$ options to colour at most one connected component of $G[Z]$ of size at least 2 red, and all other components of size at least 2 blue. 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 3.2.4, 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 4.2.1 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 3.2.1 ii) and 3.2.4 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. 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 branch takes polynomial time to process, and the number of branches is $O(n^3)$. This completes our proof. $\square$

The proof of our second result combines Lemma 4.2.1 with arguments used in the proof of Theorem 4.1.3. Recall that for Matching Cut, we could get the same result with P_3 instead of P_2 and for Perfect Matching Cut even P_4 was possible. However, while it is not known if the results for Matching Cut and Perfect Matching Cut can be improved, P_2 is best possible for Maximum Matching Cut, as we will see in Theorem 5.3.8.

Theorem 4.2.3 ([51]). *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.*

Proof. Assume that Maximum Matching Cut is polynomial-time solvable for H-free graphs. Let $G=(V,E)$ be a connected $(H+P_2)$-free graph on n vertices. If G is H-free, we are done by assumption. Suppose G has an induced subgraph G' isomorphic to H. Let G^* be the graph obtained from G after removing the vertices of $V(G') \cup N(V(G'))$. Since G' is isomorphic to H and G is $(H+P_2)$-free, G^* is P_2-free. Hence, $V(G^*)$ is an independent set. By Observation 3.1.1 it suffices to find a maximum valid red-blue colouring of G. Below we explain how to do this.

We first branch over all options of colouring every $u \in V(G')$ red or blue, and colouring at most one neighbour of every $u \in V(G')$ with a different colour than u. If in a branch we only used one colour, we branch over all $O(n^2)$ options of choosing a bichromatic edge. In this way we obtain, for each branch, a starting pair with a non-empty core.

Consider a branch with a starting pair (S'', T'') and core (S^*, T^*). We apply rules R1 and R2 exhaustively. If we obtain a no-answer, we may discard the branch due to Lemma 3.2.1. Else, we obtain an intermediate tuple (S,T,X,Y). Note that every vertex in $Z = V \setminus (X \cup Y)$ belongs to G^*. Hence, Z is an independent set, and thus (S,T,X,Y) is a final tuple. This means that we may apply Lemma 4.2.1. Then, in polynomial time, we either find that G has no valid red-blue (S,T,X,Y)-colouring, in which case we may discard the branch due to Lemma 3.2.1, or we find a maximum valid red-blue (S,T,X,Y)-colouring. The latter is also a maximum valid red-blue (S^*,T^*,S'',T'')-colouring, again due to Lemma 3.2.1. We remember its value. In the end, after the last branch, we output a colouring with largest value as a maximum valid red-blue colouring of G.

The correctness of our branching algorithm follows from its description. The running time is polynomial: each branch takes polynomial time to process, and the number of branches is $O(2^{|V(H)|}n^{|V(H|}) + O(n^2)$. This completes our proof. $\square$

We now show our third polynomial-time result. Again, the idea is to branch over a polynomial number of options, each of which reduces to the setting where we can apply Lemma 4.2.1. After Theorem 4.1.1, this is a second non-trivial generalization of the results in [7, 39] who both show that Matching Cut is solvable in polynomial-time for graphs of diameter at most 2.

Theorem 4.2.4 ([51]). Maximum Matching Cut *is solvable in polynomial time for graphs of diameter at most* 2.

Proof. Let $G=(V,E)$ be a graph of diameter at most 2 on n vertices. If G has diameter 1, then G is a complete graph and the problem is trivial to solve. Assume that G has diameter 2. By Observation 3.1.1 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. Now all other neighbours of u must be coloured blue. We let $D = \{u\} \cup N(u)$ and note that D dominates G, as G has diameter 2.

We 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 rules R1 and R2 on (S'', T''). By Lemma 3.2.1 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. We prove the following two claims for the set $Z' = V \setminus (X' \cup Y')$ of uncoloured vertices.

Claim 4.2.4.1. *Every vertex $z \in Z'$ has a neighbour in $Y' \setminus T'$ that belongs to D.*

Proof. As $z \in Z'$, we have that $z \notin D$. As D is dominating, z has a neighbour b in D. As every neighbour of u belongs to D and z is not in D, we find that $b \neq u$. Since D contains exactly one red vertex v, which has a blue neighbour in D (namely u), all neighbours of v in $G - D$ are coloured red, that is, belong to X. As z belongs to $G - D$ and z is not coloured red, this means that v and z are non-adjacent, and thus $b \neq v$. So, b must belong to $T'' \setminus \{u\}$, and thus to $Y' \setminus T'$, as else we could have applied R2. $\triangleleft$

Claim 4.2.4.2. *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, by Claim 4.2.4.1, v has two blue neighbours, u and some vertex $w \in Y' \setminus T'$, a contradiction. $\triangleleft$

Since Claim 4.2.4.2 holds, we may exhaustively apply R1–R3 to the intermediate tuple (S', T', X', Y'). By Lemma 3.2.4 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 4.2.4.1, the following holds for every (uncoloured) vertex $w \in Z$:

(i) w has at most one neighbour in $X \setminus S$,

(ii) w has exactly one neighbour in $Y \setminus T$, which belongs to D, and

(iii) if w' is a vertex in the same connected component of $G[Z]$ as w, then w and w' do not share a neighbour in $G - Z$.

We strengthen (i) by proving the following claim.

Claim 4.2.4.3. *Every vertex $w \in Z$ has exactly one neighbour in $X \setminus S$.*

Proof. By (i), we find that w has at most one neighbour in $X \setminus S$. For a contradiction, suppose that w has no neighbours in $X \setminus S$. We also know that w has no neighbours in S, as else we could have applied R1 or R2. Recall that v was the only red vertex of D. As v has a blue neighbour, namely u, all the other neighbours of v are coloured red due to R2. Hence, w is adjacent neither to v nor to any vertex in $N(v) \setminus \{u\}$. As all neighbours of u that are not equal to v are coloured blue and $w \in Z$ is uncoloured, we find that w is not adjacent to u either. Hence, the distance between v and w is at least 3, contradicting our assumption that G has diameter 2. $\triangleleft$

We continue by proving the following claim.

Claim 4.2.4.4. *If $G[Z]$ contains two connected components F_1 and F_2 of size at least 2, then $G[Z] = F_1 + F_2$.*

Proof. Let F_1 contain u_1 and u_2. Let F_2 contain v_1 and v_2. By combining Claim 4.2.4.3 with (ii) and (iii), we find that the vertices u_1 and u_2 have each a different red (respectively, blue) neighbour and the same holds for v_1 and v_2. However, as G has diameter 2, it holds that u_1 and u_2 each have a common neighbour with both v_1 and v_2. Thus, without loss of

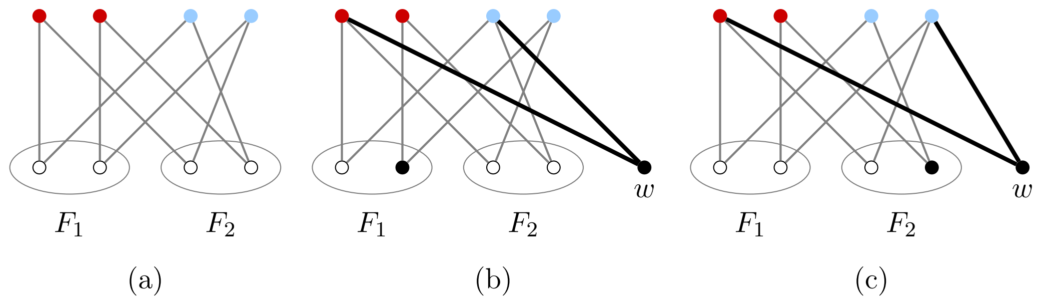


Figure 4.9: The unique way (up to symmetry) to connect two components F_1 and F_2 of $G[Z]$ of size 2 (a) and the two options to connect a vertex w in a third component F_3 to the coloured part of the graph (b) and (c). We can see that there is always an uncoloured vertex without a common neighbour with w, in the figure this vertex is highlighted black.

generality, u_1 and v_1 have a red common neighbour, u_1 and v_2 a blue one, while u_2 and v_1 have a blue common neighbour, u_2 and v_2 a red one. See also Figure 4.9 (a).

For a contradiction, assume that $G[Z]$ contains a third connected component F_3. Let w be a vertex in F_3. Then w has a common neighbour with each of u_1, u_2, v_1 and v_2. Furthermore, w has exactly one red and one blue neighbour. As can be seen in Figures 4.9 (b) and (c), there do not exist vertices $x \in X$ and $y \in Y$ such that $\{u_1, u_2, v_1, v_2\} \subseteq N_G(\{x, y\})$. Hence, w has no common neighbour with some vertex of $\{u_1, u_2, v_1, v_2\}$, contradicting our assumption that G has diameter 2. $\lhd$

From Claim 4.2.4.4, 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 4.2.4.2. 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 3.2.4, 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 4.2.1 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 3.2.1 ii) and 3.2.4 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. 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 branch takes polynomial time to process, and the number of branches is $O(n^2)$. This completes our proof. $\square$

The last theorem, where we showed that Maximum Matching Cut is solvable in polynomial time for graphs of diameter at most 2 is best possible, see Theorem 5.3.8 and [39], since Matching Cut and Maximum Matching Cut are NP-hard for graphs of diameter 3. If we add the restriction that we consider bipartite graphs, then we can give a polynomial-time algorithm for radius at most 2. This is best possible for bipartite graphs, see Theorem 5.3.9.

Theorem 4.2.5. Maximum Matching Cut *is solvable in polynomial time for bipartite graphs of radius at most* 2.

Proof. Let $G = (V, E)$ be a bipartite graph of radius at most 2 on n vertices. If G has radius 1, then G is a star and the problem is trivial to solve. Assume that G has radius 2. By Observation 3.1.1, it suffices to find a maximum valid red-blue colouring of G. Since G has radius 2 there is a dominating star D with center vertex v and leafs $\{u_1, \dots, u_k\}$, for some integer $k \geq 1$. Moreover, since G is bipartite, this star is induced. Without loss of generality, we may colour v blue. At most one neighbour of v can be blue in any maximum valid red-blue colouring of G, and thus, we branch over all $O(n)$ possible colourings of $N(v) \subseteq V(D)$.

Consider first the case where D is monochromatic, say all vertices in D are coloured blue. By definition, a maximum valid red-blue colouring of G has to contain at least one red vertex. So we branch over all $O(n)$ options of choosing a vertex w in $V - D$ to be coloured red. Recall that D is dominating so w has at least one neighbour in D. If w has more than one neighbour in D, we immediately get that the colouring is not valid, since then w, which is red, has two blue neighbours in D. So we discard the branch. Otherwise we may assume that u_1 is the unique neighbour of w in D. We set $S^* = \{w\}$, $T^* = \{u_1\}$, $S'' = \{w\}$ and $T'' = V(D)$.

In the other case, where D is not monochromatic, there is a vertex in D which is red, say this is u_1. We set $S^* = \{u_1\}$, $T^* = \{v\}$, $S'' = \{u_1\}$ and $T'' = V(D) \setminus \{u_1\}$.

In both cases, we get a starting pair (S'', T'') with core (S^*, T^*). We exhaustively apply rules R1 and R2 on (S'', T''). By Lemma 3.2.1 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. Since G is bipartite, we have that $N(V(D))$ is an independent set. So it clearly holds that every component of $V \setminus (X' \cup Y')$ is monochromatic in every maximum valid red-blue colouring. Hence, we may exhaustively apply rules R1–R3 to the intermediate tuple (S', T', X', Y'). By Lemma 3.2.4 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. Note that $V \setminus (X \cup Y) \subseteq N(V(D))$ is an independent set. This allows us to apply Lemma 4.2.1 and we either find in polynomial time a red-blue colouring of G that is a maximum red-blue (S^*, T^*, S'', T'')-colouring or that no such colouring exists. In the latter case, we discard the branch.

If somewhere in the above process we discarded a branch, 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 branch takes polynomial time to process, and the number of branches is $O(n)$. This completes our proof. □

We can also extend Theorem 4.2.4 to bipartite graphs of diameter 3, generalizing [39]. To obtain this result we use a proof which differs from the proof for Matching Cut in [39] and the proof for Perfect Matching Cut in Theorem 4.1.7. Note that Theorem 4.2.6 implies both Theorem 4.1.7 and the same result for Matching Cut. Our proof for Maximum Matching Cut is very similar to the proof of the same result for d-Cut. We refer to Theorem 6.2.2 for the details. To adjust the proof for Maximum Matching Cut we only have to make the following small change. We consider the proof for $d = 1$ and remember the value of every valid red-blue colouring that we got. In the end, we pick the valid red-blue colouring with the largest value to obtain a maximum valid red-blue colouring of G. This leads to the following Theorem.

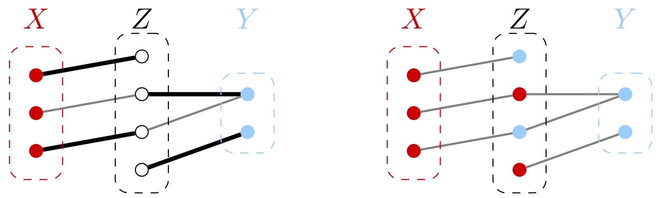


Figure 4.10: A maximum matching and the corresponding red-blue colouring. Note that this matching is not perfect and thus the colouring is not a perfect red-blue colouring.

Theorem 4.2.6. Maximum Matching Cut *is solvable in polynomial time for bipartite graphs of diameter at most* 3.

4.2.2 (Maximum) Disconnected Perfect Matching

For Maximum Disconnected Perfect Matching, we present first Lemma 4.2.7 which is proven by similar but simpler arguments as in the proof of Lemma 4.2.1. The proofs of the following Theorems follow immediately after replacing Lemmas for Matching Cut by Lemmas for Disconnected Perfect Matching. The following theorems also lead to results for Disconnected Perfect Matching which where unknown before. Especially, we get a partial answer to Question 3 from Corollary 4.2.9.

Lemma 4.2.7 ([51])**.** *Let $G = (V, E)$ be a connected graph with 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 perfect-extendable red-blue (S, T, X, Y)-colouring of G, or conclude that G has no perfect-extendable red-blue (S, T, X, Y)-colouring.*

Proof. Let $Z = V \setminus (X \cup Y)$. Let $W = N(Z)$. Recall that Z is independent. Hence, by Lemma 3.2.6 iii), every vertex of W belongs to $(X \setminus S) \cup (Y \setminus T)$. By combining this observation with Lemma 3.2.6 iii), we find that every bichromatic edge of any perfect-extendable red-blue (S, T, X, Y)-colouring (if it exists) is

(i) either an edge with one endvertex in S and the other one in T, or

(ii) an edge with one endvertex in Z and the other one in either $X \setminus S$ or $Y \setminus T$.

By Lemma 3.2.6 iii), the subgraph of G induced by $S \cup T$ has a perfect matching that consist of every edge with one endvertex in S and the other one in T. Hence, a maximum perfect-extendable red-blue (S, T, X, Y)-colouring (if it exists) is the union of

(i) the set of edges with one endvertex in S and the other one in T; and

(ii) a perfect matching in $G \setminus (S \cup T)$ that contains as many edges with one endvertex in Z (and the other one in either $X \setminus S$ or $Y \setminus T$) as possible.

We first check in polynomial time if $G \setminus (S \cup T)$ has a perfect matching (for example, by using the Blossom algorithm [18]). If not, then G has no perfect-extendable red-blue (S, T, X, Y)-colouring, and we stop. Otherwise, we found a perfect matching M of $G \setminus (S \cup T)$, and we continue as follows, see Figure 4.10 for an illustration.

We colour every vertex in X red and every vertex in Y blue. By Lemma 3.2.6 iii), every vertex in Z has at most one neighbour in X, which belongs to $X \setminus S$, and at most one neighbour in Y, which belongs to $Y \setminus T$. As Z is independent, we can do as follows for every $u \in Z$. If u has degree 1 in G and a neighbour $x \in X$, then ux must belong to M, and we colour u blue. If u has degree 1 in G and a neighbour $y \in Y$, then uy must belong to M, and we colour u red. If u has degree 2 in G, and a neighbour $x \in X$ and a neighbour $y \in Y$, then we colour u blue if $ux \in M$ and red if $uy \in M$. This takes polynomial time. As every edge of M with one endvertex in Z and the other one in either

$X \setminus S$ or $Y \setminus T$ is bichromatic, we found, in polynomial time, a maximum perfect-extendable red-blue (S, T, X, Y)-colouring of G. □

The following result is proven in exactly the same way as Theorem 4.2.2 after replacing in its proof Lemma 3.1.3 by Lemma 3.1.4; Lemma 3.2.1 by Lemma 3.2.5; Lemma 3.2.4 by Lemma 3.2.6; and Lemma 4.2.1 by Lemma 4.2.7.

Theorem 4.2.8 ([51])**.** Maximum Disconnected Perfect Matching *is solvable in polynomial time for P_6-free graphs.*

Corollary 4.2.9 follows immediately. Recall that in Question 3 Bouquet and Picouleau asked for the complexity of Disconnected Perfect Matching for P_k-free graphs, when $k \geq 6$. Here we solve one of the cases. In Theorem 5.3.5 we contribute further to answering this question.

Corollary 4.2.9 ([51])**.** Disconnected Perfect Matching *is solvable in polynomial time for P_6-free graphs.*

Our following result can be proven in the same way as Theorem 4.2.3 after replacing Lemma 3.2.4 by Lemma 3.2.6; and Lemma 4.2.1 by Lemma 4.2.7. Note that it also holds for Disconnected Perfect Matching, for which this was unknown before.

Theorem 4.2.10 ([51])**.** *Let H be a graph. If* (Maximum) Disconnected Perfect Matching *is polynomial-time solvable for H-free graphs, then it is so for $(H + P_2)$-free graphs.*

The following result is proven in exactly the same way as the proof of Theorem 4.2.4 after replacing Lemma 3.2.1 by Lemma 3.2.5; Lemma 3.2.4 by Lemma 3.2.6; and Lemma 4.2.1 by Lemma 4.2.7. It generalises a result from [8], who showed that Disconnected Perfect Matching is solvable in polynomial time for graphs of diameter at most 2.

Theorem 4.2.11 ([51])**.** Maximum Disconnected Perfect Matching *is solvable in polynomial time for graphs of diameter at most* 2.

The following result is proven in exactly the same way as the proof of Theorem 4.2.5 after replacing Lemma 3.2.1 by Lemma 3.2.5; Lemma 3.2.4 by Lemma 3.2.6; and Lemma 4.2.1 by Lemma 4.2.7.

Theorem 4.2.12. Maximum Disconnected Perfect Matching *is solvable in polynomial time for bipartite graphs of radius at most* 2.

This implies that the same result holds for Disconnected Perfect Matching.

Corollary 4.2.13. Disconnected Perfect Matching *is solvable in polynomial time for bipartite graphs of radius at most* 2.

Similar to Theorem 4.2.6, our last result in this section follows again from a result for d-Cut. We only make the following change to the proof of Theorem 6.2.2. Consider the proof for $d = 1$ and check for every valid red-blue colouring which we obtained, if it is also a perfect extendable red-blue colouring. Remember the value of every perfect extendable red-blue colouring which we get. In the end, we pick the perfect extendable red-blue colouring with the largest value to obtain a maximum perfect extendable red-blue colouring. This leads to the following Theorem.

Theorem 4.2.14. Maximum Disconnected Perfect Matching *is solvable in polynomial time for bipartite graphs of diameter at most* 3.

Chapter 5

Hardness Results

In this chapter, we prove hardness results for Matching Cut, Disconnected Perfect Matching and the corresponding maximisation problems. We reduce from different problems like Max Cut and Exact 3-Cover but also from Matching Cut, which is already known to be NP-complete. Our first result shows that Matching Cut is NP-complete for graphs of high girth. This problem was open for 20 years. We give the same result for Disconnected Perfect Matching and show further the hardness of Matching Cut and Disconnected Perfect Matching for H_i^*-free graphs, for $i \geq 1$ and for graphs with forbidden induced paths. This contributes to answering Question 3. Moreover, we give the hardness results needed to complete the complexity dichotomies for Maximum Matching Cut and Maximum Disconnected Perfect Matching. That is, we show that Maximum Matching Cut and Maximum Disconnected Perfect Matching are NP-hard for $2P_3$-free graphs and (bipartite) graphs of bounded radius and diameter (Section 5.3.2) and line graphs (Section 5.4).

5.1 Graphs of High Girth

In this section, we show that Matching Cut and Disconnected Perfect Matching remain NP-complete even for bipartite graphs of high girth and bounded maximum degree. Recall that previously both problems were known to be NP-complete for (planar) graphs of girth at least g, only for $g \leq 5$ [6, 8]. The complexity of Matching Cut for graphs of high girth was a 20-year old open problem, which was given regularly in the literature, see [11, 39, 44, 49]. We also briefly discuss how an implicit observation of Le and Telle [44] gives a simple, alternative proof for showing NP-completeness for Perfect Matching Cut (Section 5.1.3) for graphs of high girth. Further, we explain how to use this observation for Maximum Matching Cut and Maximum Disconnected Perfect Matching.

5.1.1 Matching Cut

In order to prove our hardness result for Matching Cut for bipartite graphs of high girth and bounded maximum degree, we use known results on expander graphs and number theory. We first prove a number theoretical lemma which turns out to be helpful for the construction of an immune graph with high girth.

Lemma 5.1.1 ([21]). *There are infinitely many primes q, such that*

i) $q > 13$,

ii) $q \equiv 1 \bmod 4$, *and*

iii) the Legendre symbol $\left(\frac{q}{13}\right) = -1$.

Proof. We apply Theorem 2.1.2 by Dirichlet for $a = 5$ and $b = 52 = 4 \times 13$, which are coprime. This yields that the sequence $5+52k$, for $k \in \mathbb{N}$, contains infinitely many primes, and consequently, infinitely many primes satisfying i). Since for any $k \in \mathbb{N}$, we have $5 + 52k \equiv 1 \bmod 4$, condition ii) is also satisfied.

Finally, we will show that the Legendre symbol $\left(\frac{5+52k}{13}\right) = -1$. First we note that if $a \equiv b \bmod p$, then $\left(\frac{a}{p}\right) = \left(\frac{b}{p}\right)$ (see [57]). Hence, since $5 \equiv 5 + 52k \bmod 13$, we have $\left(\frac{5}{13}\right) = \left(\frac{5+52k}{13}\right)$. We now deduce that $\left(\frac{5}{13}\right) \equiv 5^{\frac{13-1}{2}} \bmod 13 \equiv 12 \bmod 13 \equiv -1 \bmod 13$. This proves condition iii). □

Recall that $h(G)$ is the edge expansion of a graph. Lemma 5.1.2 uses known results from the theory of expander graphs. We need the fact that the graph in the statement of Lemma 5.1.2 has a perfect matching only in Section 5.1.2.

Lemma 5.1.2 ([21])**.** *For every $g \geq 3$, there is an immune 14-regular bipartite graph with girth at least g that contains a perfect matching.*

Proof. It follows from a construction of Lubotzky, Phillips and Sarnak [47] based on Caley graphs that for every two primes p and q with the following four properties

- $q > p$,
- $p \equiv 1 \bmod 4$,
- $q \equiv 1 \bmod 4$, and
- $\left(\frac{p}{q}\right) = -1$,

there exists a $(p + 1)$-regular bipartite graph G with $\lambda_2(G) \leq 2\sqrt{p}$ and girth at least $4 \log_p q - \log_p 4$; here, $\lambda_2(G)$ denotes the second largest eigenvalue of the adjacency matrix of G.

We set $p = 13$, so $p \equiv 1 \bmod 4$. We now combine the above with Lemma 5.1.1 to find that there exist infinitely many primes q, such that there exists a 14-regular bipartite graph G_q with $\lambda_2(G_q) \leq 2\sqrt{13}$ and girth at least $4 \log_{13} q - \log_{13} 4$. Moreover, Dodziuk [17] and, independently, Alon and Milman [1] showed that for every integer $\ell \geq 1$, every ℓ-regular graph G satisfies $h(G) \geq \frac{1}{2}(\ell - \lambda_2(G))$. This means that

$$h(G_q) \geq \frac{1}{2}(14 - \lambda_2(G_q)) \geq \frac{1}{2}(14 - 2\sqrt{13}) \approx 3.39 > 1.$$

Hence, by applying Observation 2.2.2, we find that G_q is immune.

By taking q sufficiently large, we conclude that for any $g \geq 3$, there exists an immune 14-regular bipartite graph G with girth at least g. Finally, as G is bipartite and regular, we find that G has a perfect matching (due to Hall's Marriage Theorem [31]). □

Remark 5.1.3. The arguments that we used to prove Lemma 5.1.2 do not allow us to set $p = 13$ to a smaller value. We need p to be prime, and moreover, it must hold that $p \equiv 1 \bmod 4$. Hence, the only alternative value for p that is smaller than 13 would be $p = 5$. However, $p = 5$ yields $h(G_q) \geq \frac{1}{2}(6 - 2\sqrt{5}) \approx 0.76$, so we cannot conclude from this that G_q is immune.

We use Lemma 5.1.2 to prove the theorem which answers Question 1.

Theorem 5.1.4 ([21])**.** *For every integer $g \geq 3$,* Matching Cut *is NP-complete for bipartite graphs of girth at least g and maximum degree at most 60.*

Proof. Let $g \geq 3$. As the class of graphs of girth at least $g+1$ is a subclass of the class of graphs of girth at least g, we may assume without loss of generality that g is divisible by 2 but not by 4, so $\frac{g}{2}$ is odd. We reduce from Matching Cut. Recall that Matching Cut is NP-complete for graphs of degree at most 4 [12]. Let G be a connected graph of maximum degree at most 4. From G we construct a graph G' with the required properties, but first we define an auxiliary graph F'.

By Lemma 5.1.2 there exists an immune 14-regular bipartite graph F with girth at least g. Note that F has constant size, so we can find F in constant time.

Figure 5.1: The graph F with designated vertices x and y at distance at least g (left) and the graph F' (right) from the proof of Theorem 5.1.4.

Let x and y be two vertices of distance $\frac{g}{2}$ in F. As $\frac{g}{2}$ is odd, x and y belong to opposite bipartition classes of F. We take two copies F_1 and F_2 of F, and we add an edge between the two copies x_1 and x_2 of x and an edge between the two copies y_1 and y_2 of y. This yields a graph F'. As F is bipartite and has girth at least g, we find that F' is bipartite and has girth at least g as well. The construction also gives us that x_1 and y_2 belong to the same bipartition class of F'. See also Figure 5.1.

Now consider an edge uv in G. The *F'-replacement* of uv is obtained from the graphs G and F' by identifying u with x_1 and v with y_2. We do an F'-replacement on every edge of G. This yields the graph G'. Since F' is bipartite such that the distance between x_1 and y_2 is even, we find that G' is bipartite as well. By construction, G' has girth at least g and maximum degree at most $4 \cdot (14+1) = 60$.

We claim that G has a matching cut if and only if G' has a matching cut. We prove this below, using Observation 3.1.1(i) implicitly.

First suppose that G has a matching cut, so G has a valid red-blue colouring c. For every edge uv in G we do as follows. Let F' with copies F_1 and F_2 of F be the corresponding F'-replacement applied on uv. If u and v have the same colour, then we colour every vertex of $V(F') \setminus \{u,v\}$ with that colour. If u and v are coloured differently, say u is red and v is blue, then we colour every vertex in $V(F_1)$ red and every vertex in $V(F_2)$ blue. This yields a valid red-blue colouring of G'. Hence, G' has a matching cut.

Now suppose that G' has a matching cut, so G' has a valid red-blue colouring c'. As F is immune, every copy of it in G' is monochromatic. Thus, for two vertices $x_1, y_2 \in V(F')$ in some graph F' in G', it holds that x_1 and y_2 each have a neighbour in F' of the opposite colour if and only if x_1 and y_2 have different colours in G'. Hence, the restriction of c' to $V(G)$ is a valid red-blue colouring of G. Hence, G has a matching cut. □

5.1.2 Disconnected Perfect Matching

We now show that Disconnected Perfect Matching is NP-complete for bipartite graphs of arbitrarily large fixed girth and bounded maximum degree. We are aware of two proofs leading to this result, with slightly different degree bounds. However for both of them we need a more involved gadget than in Section 5.1.1.

We first define some useful auxiliary graphs. Fix $g \geq 3$ such that g is divisible by 6 and $\frac{g}{6}$ is odd. Let F be a 14-regular immune bipartite graph with girth at least $2(g+1)$

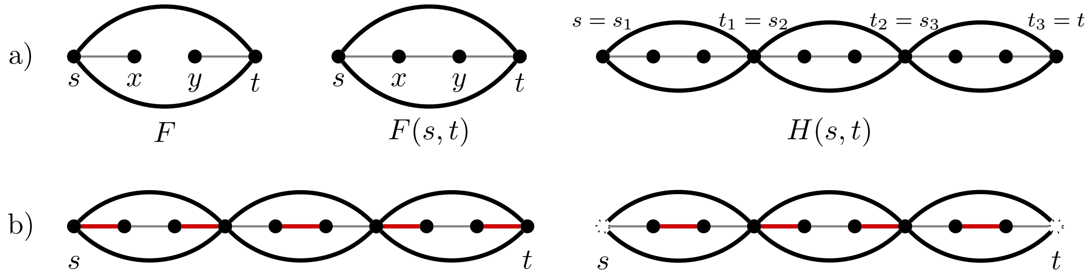


Figure 5.2: a) Illustration of the graphs F (left), $F(s,t)$ (middle) and $H(s,t)$ (right). b) Illustration of how the edges in the perfect matching of $H(s,t)$ (left) resp. $H(s,t)-\{s,t\}$ (right) are chosen (these edges are represented as red, thick edges).

containing a perfect matching. We note that F exists by Lemma 5.1.2, and as F has constant size, F can be found in constant time.

Let s and t be two designated vertices in F at distance at least $g+1$. We fix a perfect matching M of F. Let $x \in N_F(s)$ and $y \in N_F(t)$ be the (unique) neighbours of s and t in M. Then, since s and t are at distance at least $g+1$, x and y are at distance at least $g-1$. We add the edge xy and denote the resulting graph by $F(s,t)$; see also Figure 5.2 a). We make the following observation.

Lemma 5.1.5 ([21])**.** *The graph $F(s,t)$ is immune, bipartite and has girth at least g. Moreover, both $F(s,t)$ and $F(s,t)-\{s,t\}$ contain a perfect matching.*

Proof. As F is immune, $F(s,t)$ is immune. By our choice of x and y, we find that $F(s,t)$ is bipartite and has girth at least g. The fixed perfect matching M of F is of course also a perfect matching of $F(s,t)$. The new edge xy ensures that $F(s,t)-\{s,t\}$ contains a perfect matching as well; indeed, we can take $(M \setminus \{sx,ty\}) \cup \{xy\}$. □

We now take $k = \frac{g}{6}$ copies $F(s_1,t_1),\dots,F(s_k,t_k)$ of $F(s,t)$ and identify s_{i+1} with t_i for all $i \in \{1,\dots,k-1\}$. We set $s = s_1$ and $t = t_k$ and call the resulting graph $H(s,t)$; see also Figure 5.2 a).
In the following lemma, we show some useful properties of $H(s,t)$ and $H(s,t)-\{s,t\}$.

Lemma 5.1.6 ([21])**.** *The graph $H(s,t)$ is immune, bipartite, and has girth at least g. Moreover,* $\mathrm{dist}(s,t) \geq \frac{g}{2}$, *and both $H(s,t)$ and $H(s,t)-\{s,t\}$ contain a perfect matching.*

Proof. By taking two immune graphs and identifying two of their vertices (one from each graph), we obtain another immune graph. The graph $H(s,t)$ is obtained by repeatedly applying this operation on copies of $F(s,t)$, which is immune by Lemma 5.1.5. Hence, we conclude that $H(s,t)$ is immune.

By Lemma 5.1.5, every graph $F(s_i,t_i)$ for $i \in \{1,\dots,k\}$ is bipartite and has girth at least g. By constructing $H(s,t)$ from these graphs $F(s_i,t_i)$, we do not create any new cycles. Thus, $H(s,t)$ also is bipartite and has girth at least g. In every graph $F(s_i,t_i)$ we have $\mathrm{dist}(s_i,t_i)=3$, thus $\mathrm{dist}(s,t) = 3 \times \frac{g}{6} = \frac{g}{2}$.

To see that $H(s,t)$ and $H(s,t)-\{s,t\}$ contain both a perfect matching, we will use the fact that both $F(s_i,t_i)$ and $F(s_i,t_i)-\{s_i,t_i\}$ contain a perfect matching by Lemma 5.1.5. For $H(s,t)$, we take a perfect matching of $F(s_1,t_1)$, $F(s_2,t_2)-\{s_2,t_2\}$, $F(s_3,t_3)$ and continue this alternation until $F(s_k,t_k)$, see Figure 5.2 b) (left). For $H(s,t)-\{s,t\}$ we alternate as well but we start and end with $F(s_1,t_1)-\{s_1,t_1\}$ and $F(s_k,t_k)-\{s_k,t_k\}$, see Figure 5.2 b) (right). □

The first option to prove Theorem 5.1.7 is to use a similar approach as in the proof of Theorem 5.1.4. We take two copies of the graph $H(s,t)$ from Lemma 5.1.6, add two extra vertices and connect them as shown in Figure 5.3. The resulting graph F'' has girth at least g, a perfect matching containing x_1 and y_2 and a perfect matching, not containing x_1 and y_2. Thus, we can use the proof of Theorem 5.1.4 with the new gadget F'' instead of F', leading to a maximum degree of 75.

There is a second proof, using new ideas, which improves the bound of 75 to 74 and which we do give below.

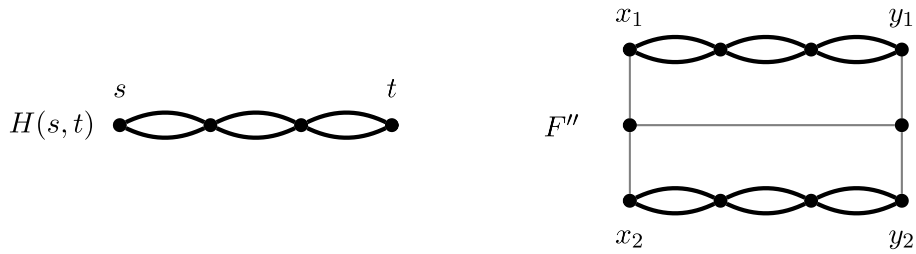


Figure 5.3: The graph $H(s,t)$ with designated vertices s and t at distance at least g (left) and the graph F'' (right).

Theorem 5.1.7 ([21]). *For every integer $g \geq 3$,* Disconnected Perfect Matching *is* NP*-complete for bipartite graphs of girth at least g and maximum degree at most* 74.

Proof. Let $g \geq 3$. We reduce from Matching Cut for bipartite graphs of girth at least g and maximum degree at most 60, which is NP-complete by Theorem 5.1.4. Similar to Theorem 5.1.4, we may assume without loss of generality that g is divisible by 12. Let G be a bipartite graph of girth at least g and maximum degree 60. We construct a graph G' by taking two copies G_1 and G_2 of G, where we connect every vertex $v \in V(G_1)$ and its copy $v' \in V(G_2)$ using the graph $H(v, v')$.

To see that G' has girth at least g, we consider first G_1 and G_2, which both have girth at least g. Any cycle containing vertices from both copies has to pass twice through a graph $H(s,t)$. Thus, it will always have length at least g, and so G' has girth g. Every vertex inside $H(s,t)$ has degree at most 28, whereas s and t only have degree 14. The degree of a vertex $v \in V(G_1) \cup V(G_2)$ is the degree of the vertex in the original graph G plus the degree in the graph $H(v, v')$. Thus, the degree of v is at most 74.

We also claim that G' is bipartite. For a contradiction, assume that G' has an odd cycle C. As G_1 and G_2 are both bipartite, C' must contain vertices from G_1 and G_2. Note that C passes through an even number of graphs $H(v, v')$, where $v \in V(G_1)$ and $v' \in V(G_2)$. Hence, the number of edges in $E(G) \setminus (E(G_1) \cup E(G_2))$ is even. Since the graph $H(v, v')$ always connects a vertex v in G_1 and its copy v' in G_2 we can find an odd cycle C' in G_1, consisting of the edges in $C \cap E(G_1)$ and the edges in $E(G_1)$ corresponding to the edges from $C \cap E(G_2)$, a contradiction.

Finally, we show that G admits a matching cut if and only if G' admits a disconnected perfect matching. Consider some vertex $v \in V(G_1)$ and its copy $v' \in V(G_2)$. Since v and v' are connected by the graph $H(v, v')$, which is immune, v and v' will always have the same colour in any valid red-blue colouring of G'. Thus, G_1 and G_2 will be coloured the same in any valid red-blue colouring. Now, if G' admits a perfect-extendable red-blue colouring, then G admits a valid red-blue colouring, as it suffices to colour G the same as G_1 (or G_2).

Conversely, if G admits a valid red-blue colouring, then we obtain a perfect-extendable colouring of G' as follows. We colour G_1 and G_2 the same as G. Notice that we colour the immune graphs connecting two copies of the same vertex such that they are monochromatic.

This gives us a valid red-blue colouring of G', i.e. a matching cut M in G'. It remains to show that the matching cut is contained in a perfect matching of G'. Since the colourings of G_1 and G_2 are the same, we have that whenever a blue vertex $v \in V(G_1)$ is matched with a red vertex $u \in V(G_1)$, i.e. $vu \in M$, then their copies $v' \in V(G_2)$ and $u' \in V(G_2)$ are matched as well, i.e. $v'u' \in M$. By Lemma 5.1.6, we know that $H(v, v') - \{v, v'\}$ and $H(u, u') - \{u, u'\}$ contain both a perfect matching which we may add to M. For every $v \in V(G_1)$ with no neighbour of the other colour, we know that its copy $v' \in V(G_2)$ has no neighbour of the other colour either. Thus, we can use that $H(v, v')$ contains a perfect matching by Lemma 5.1.6 and add it to M. Repeatedly doing this yields a perfect matching of G' containing M. □

5.1.3 Perfect and Maximum Matching Cut

It is known from [44] that PERFECT MATCHING CUT is NP-complete for graphs of high girth. The high girth in their construction is based on a simple lemma, which was implicit in their paper. The lemma is based on the observation that in any perfect red-blue colouring of a connected graph $G = (V, E)$, a vertex $v \in V$ of degree 2 has exactly one neighbour coloured the same as itself and exactly one neighbour coloured differently than itself. The proof of the lemma is short, and we include it for completeness.

Lemma 5.1.8 ([21, 44])**.** *Let $G = (V', E')$ be the graph obtained from a connected graph G by 4-subdividing an edge e of G. Now, G has a perfect matching cut if and only if G' has a perfect matching cut.*

Proof. Let $e = uv$ and denote the resulting path in G' by $uu_1u_2u_3u_4v$. First suppose G has a perfect matching cut, so G has a perfect red-blue colouring c. If u and v are coloured alike, say u and v are red, then we colour u_1 red, u_2 blue, u_3 blue and u_4 red. Else we may assume that u is red and v is blue. In that case we colour u_1 blue, u_2 blue, u_3 red and u_4 red. In both cases we obtain a perfect red-blue colouring c' of G'. So G' has a perfect matching cut. See Figure 5.4 for an illustration.

Now suppose that G' has a perfect matching cut, so G' has a perfect red-blue colouring c'. If u and v are coloured alike, say u and v are red, then u_1 must be red, u_2 blue, u_3 blue and u_4 red. Hence, the restriction of c' to V is a perfect red-blue colouring of G. Else we may assume that u is red and v is blue. In that case u_1 must be blue, u_2 blue, u_3 red and u_4 red. Again, the restriction of c' to V is a perfect red-blue colouring of G. Hence, in both cases we find that G has a perfect matching cut. □

From this lemma we can obtain a simple proof for showing that PERFECT MATCHING CUT is NP-complete for graphs of girth at least g. Apply Lemma 5.1.8, say, g times on each edge of the input graph. Combining this observation with the NP-completeness of PERFECT MATCHING CUT for 3-connected cubic planar bipartite graphs ([5]) leads to the NP-completeness of PERFECT MATCHING CUT for cubic planar bipartite graphs of girth at least g.

Theorem 5.1.9 ([5, 44])**.** PERFECT MATCHING CUT *is NP-complete for cubic planar bipartite graphs of girth at least g.*

This method can also be applied to MAXIMUM MATCHING CUT and MAXIMUM DISCONNECTED PERFECT MATCHING. For MAXIMUM MATCHING CUT we obtain the following.

Lemma 5.1.10. *Let $G = (V', E')$ be the graph obtained from a connected graph G by 4-subdividing an edge e of G. Now, G has a matching cut of size ν if and only if G' has a matching cut of size $\nu + 2$.*

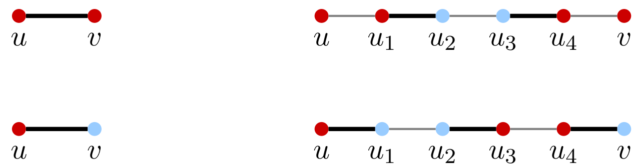


Figure 5.4: A monochromatic edge (top) and a bichromatic edge (bottom) together with the corresponding colouring of their 4-subdivision.

For Maximum Disconnected Perfect Matching we obtain the same result. We give a proof for completeness.

Lemma 5.1.11. *Let $G = (V', E')$ be the graph obtained from a connected graph G by 4-subdividing an edge e of G. Now, G has a disconnected perfect matching of size ν if and only if G' has a disconnected perfect matching of size $\nu + 2$.*

Proof. Let $e = uv$ and denote the resulting path in G' by $uu_1u_2u_3u_4v$. First suppose G has a disconnected perfect matching of size ν, so G has a perfect extendable red-blue colouring c of value ν. If u and v are coloured alike, say u and v are red, then we colour u_1 red, u_2 blue, u_3 blue and u_4 red. Else we may assume that u is red and v is blue. In that case we colour u_1 blue, u_2 blue, u_3 red and u_4 red. In both cases we obtain two additional bichromatic edges and a perfect extendable red-blue colouring c' of G' of value $\nu + 2$. So G' has a disconnected perfect matching of size $\nu + 2$. See Figure 5.4 for an illustration.

Now suppose that G' has a disconnected perfect matching of size $\nu + 2$, so G' has a perfect extendable red-blue colouring c' of value $\nu + 2$. If u and v are coloured alike, say u and v are red, then, for a maximum colouring, u_1 must be red, u_2 blue, u_3 blue and u_4 red. Hence, the restriction of c' to V is an extendable red-blue colouring of G of value ν. Else we may assume that u is red and v is blue. In that case u_1 must be blue, u_2 blue, u_3 red and u_4 red. Again, the restriction of c' to V is a perfect extendable red-blue colouring of G of value ν. Hence, in both cases we find that G has a disconnected perfect matching of size ν. □

We can apply these two lemmas to obtain NP-hardness results for Maximum Matching Cut and Maximum Disconnected Perfect Matching for graphs of high girth with much better degree bounds than the bounds for Matching Cut and Disconnected Perfect Matching. For Maximum Matching Cut, we use the hardness result for Perfect Matching Cut of [5] for cubic planar bipartite graphs.

Theorem 5.1.12. *For every $g \geq 3$,* Maximum Matching Cut *is NP-hard for cubic planar bipartite graphs of girth at least g.*

By applying Lemma 5.1.11, we get the following Theorem. Recall that the NP-completeness of Disconnected Perfect Matching on planar graphs of maximum degree 4 implies that Maximum Disconnected Perfect Matching is NP-hard for this graph class ([8]).

Theorem 5.1.13. *For every $g \geq 3$,* Maximum Disconnected Perfect Matching *is NP-hard for planar graphs of girth at least g and maximum degree at most 4.*

Proof. Let G be planar graph of maximum degree 4. By [8] we know that Maximum Disconnected Perfect Matching is NP-hard for this graph class. We apply Lemma 5.1.11 g times on each edge and obtain a bipartite graph G' with girth at least g and maximum degree 4 and the theorem follows. □

Note that to obtain these girth results, it is not necessary to use the lemmas, since the NP-completeness for Perfect Matching Cut for some graph class immediately leads to the NP-hardness of Maximum Matching Cut and Maximum Disconnected Perfect Matching for that same graph class. So for Maximum Matching Cut and Maximum Disconnected Perfect Matching on cubic planar bipartite graphs of girth at least g the NP-hardness already follows from the observation made above for Perfect Matching Cut. Still, the lemmas might also be helpful to obtain other results.

5.2 Graphs without Induced H_i^*

In the following, we show that Matching Cut and Disconnected Perfect Matching are NP-hard in H_i^*-free graphs, for an integer $i \geq 1$. We also point out that it follows from Lemma 5.1.8, which was implicit in [44], that the same holds for Perfect Matching Cut.

Let G be an H_i^*-free graph, for some integer $i \geq 1$. Let uv be an edge of G. We define an edge operation as displayed in Figure 5.5, which when applied on uv will replace uv in G by the subgraph T_{uv}^i. Note that in the new graph, the only vertices from T_{uv}^i that may have neighbours outside T_{uv}^i are u and v.

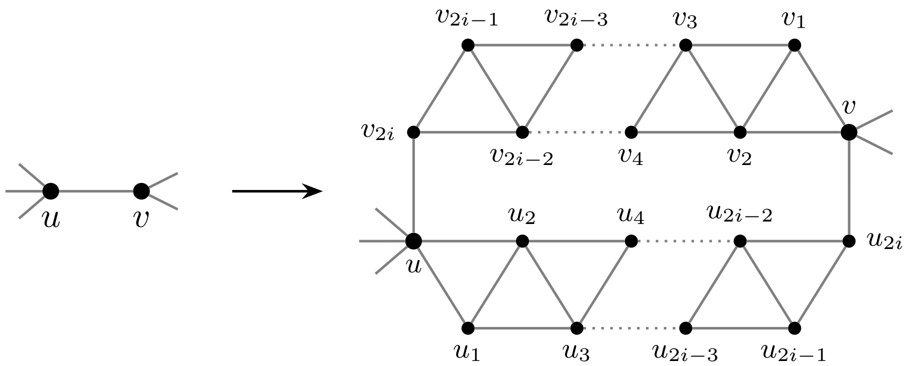


Figure 5.5: The edge uv (left) which we replace by the subgraph T_{uv}^i, for $i \geq 1$ (right).

Theorem 5.2.1 ([21]). *For all $i \geq 1$,* Matching Cut *is NP-complete for $(H_1^*, \ldots, H_i^*)$-free graphs.*

Proof. Fix $i \geq 1$. Reduce from Matching Cut. Let $G = (V, E)$ be a connected graph. Replace every $uv \in E$ by the graph T_{uv}^i (see Figure 5.5). This yields the graph $G' = (V', E')$.

We claim that G' is $(H_1^*, \ldots, H_i^*)$-free. For a contradiction, assume that G' contains an induced $H_{i'}^*$ for some $1 \leq i' \leq i$. Then G' contains two vertices x and y that are centers of an induced claw, as well as an induced path from x to y of length i'. All vertices in $V' \setminus V$ are not centers of any induced claw. Hence, x and y belong to V. By construction, any shortest path between two vertices of V has length at least $i+1$ in G', a contradiction.

We claim that G' has a matching cut if and only if G has a matching cut. First suppose G' has a matching cut M', so G' has a valid red-blue colouring c'. We prove a claim for G':

Claim 5.2.1.1. *For every edge $uv \in E(G)$ it holds that*

(a) either $c'(u) = c'(v)$, and then T_{uv}^i is monochromatic, or

(b) $c'(u) \neq c'(v)$, and then c' colours $u_1, \ldots, u_{2i}$ with the same colour as u, while c' colours all vertices of $T_{uv}^i - \{u, u_1, \ldots, u_{2i}\}$ with the same colour as v, and moreover, $uv_{2i}, vu_{2i} \in M'$.

Proof. First assume $c'(u) = c'(v)$, say c' colours u and v red. As any clique of size at least 3 is monochromatic, all vertices in T_{uv}^i are coloured red, so T_{uv}^i is monochromatic.

Now assume $c'(u) \neq c'(v)$, say u is red and v blue. As before, we find that all vertices $u_1, \ldots, u_{2i}$ have the same colour as u, so are red, while all vertices $v_1, \ldots, v_{2i}$ have the same colour as v, so are blue. By definition, every edge xy with $c'(x) \neq c'(y)$ belongs to M', so $uv_{2i}, vu_{2i} \in M'$. $\triangleleft$

We construct a subset $M \subseteq E$ in G as follows. We add an edge $uv \in E$ to M if and only if $c'(u) \neq c'(v)$ in G'. We now show that M is a matching in G. Let $u \in V$. For a contradiction, suppose that M contains edges uv and uw for $v \neq w$. Then $c'(u) \neq c'(v)$ and $c'(u) \neq c'(w)$. By Claim 5.2.1.1, we find that M' matches u in G' to vertices in T^i_{uv} and T^i_{uw}, contradicting our assumption that M' is a matching (cut). Hence, M is a matching.

Now let c be the restriction of c' to V. If c colours every vertex of G with one colour, say red, then c' would also colour every vertex of G' red by Claim 5.2.1.1, contradicting the validity of c'. Hence, c uses both colours. Moreover, for every $uv \in E$, the following holds: if $c(u) \neq c(v)$, then $c'(u) \neq c'(v)$ and thus $uv \in M$. Hence, as M is a matching, c is valid, and thus M is a matching cut of G.

Conversely, assume that G admits a matching cut, so V has a valid red-blue colouring c. We construct a red-blue colouring c' of V' as follows.

- For every edge $uv \in E$ with $c(u) = c(v)$, we let $c'(x) = c(u)$ for every $x \in V(T^i_{uv})$.
- For every edge $uv \in E$ with $c(u) \neq c(v)$, we let $c'(u) = c'(u_1) = \cdots = c'(u_{2i}) = c(u)$ and $c'(v) = c'(v_1) = \cdots = c'(v_{2i}) = c(v)$.

As c is valid, c uses both colours and thus by construction, c' uses both colours. Let $u \in V$. Again as c is valid, $c(u) \neq c(v)$ holds for at most one neighbour v of u in G. Hence, by construction, u belongs to at most one non-monochromatic gadget T^i_{uv}. Thus, c' colours in G' at most one neighbour of u with a different colour than u. Let $u \in V' \setminus V$. By construction, we find again that c' colours at most one neighbour of u with a different colour than u. Hence, c' is valid, and so G' has a matching cut. This completes the proof of Theorem 5.2.1. $\square$

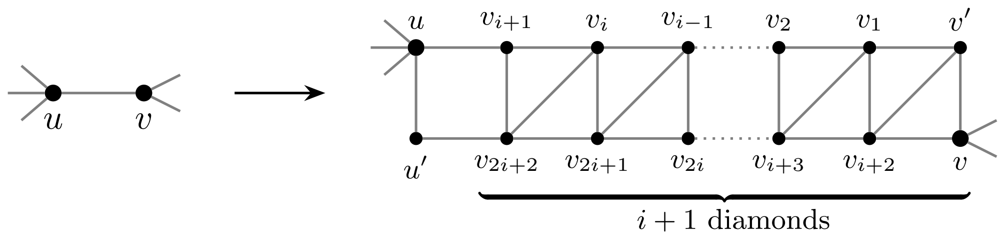


Figure 5.6: The edge uv (left) which we replace by the subgraph G^i_{uv} (right).

We show the corresponding result for Disconnected Perfect Matching in a similar way as Theorem 5.2.1. First we define the edge operation displayed in Figure 5.6. This operation replaces an edge uv in a graph G by the subgraph G^i_{uv} for some integer $i \geq 1$. Note that in the resulting graph, the only vertices from G^i_{uv} that may have neighbours outside G^i_{uv} are u and v.

Theorem 5.2.2 ([21])**.** *For every $i \geq 1$,* Disconnected Perfect Matching *is NP-complete for $(H_1^*, \ldots, H_i^*)$-free graphs.*

Proof. Fix an integer $i \geq 1$. As the class of $(H_1^*, \ldots, H_i^*)$-free graphs is contained in the class of $(H_1^*, \ldots, H_{i-1}^*)$-free graphs if $i \geq 2$, we may assume without loss of generality that i is even (we need this assumption at a later place in our proof). We reduce from Disconnected Perfect Matching itself. Let $G = (V, E)$ be a connected graph.

Let uv be an edge of G. We recall the edge operation displayed in Figure 5.6, which when applied on uv replaces uv by the subgraph G^i_{uv}. Recall that in the resulting graph, u and v are the only vertices from G^i_{uv} that may have neighbours outside G^i_{uv}.

In G we now replace every edge $uv \in E$ by the graph G^i_{uv}. Let $G' = (V', E')$ be the resulting graph.

We claim that G' is $(H_1^*, \ldots, H_i^*)$-free. For a contradiction, assume that G' contains an induced $H^*_{i'}$ for some $1 \leq i' \leq i$. Then G' contains two vertices x and y that are centers of an induced claw, as well as an induced path from x to y of length i'. The only vertices in $V' \setminus V$ that are the center of an induced claw are the vertices v_{2i+2} (see Figure 5.6). Suppose for a contradiction that $x = v_{2i+2}$. Since the graph $G[\{u, u', v_{2i+2}, v_{2i+1}, v_{i+1}\}]$ contains an induced C_4, the induced path from x to y may not contain u. Thus, either $y = v$ or the path from x to y passes through v. Since a shortest path from v_{2i+2} to v has length $i + 1$, we get a contradiction. Hence, x and y belong to V. By construction, any shortest path between two vertices of V has length at least $i + 3$ in G', a contradiction.

We claim that G' has a disconnected perfect matching if and only if G has a disconnected perfect matching. First assume that G' has a disconnected perfect matching M', so G' has a perfect-extendable red-blue colouring c'. We prove the following claim for G':

Claim 5.2.2.1. *For every edge $uv \in E(G)$, it holds that*

(a) either $c'(u) = c'(v)$, and then G^i_{uv} is monochromatic, or

(b) $c'(u) \neq c'(v)$, and then c' colours u' and all neighbours of u in $G - v$ with the same colour as u, while c' colours all vertices of $G^i_{uv} - \{u', u\}$ with the same colour as v, and moreover, $uv_{i+1} \in M'$ and either $vv' \in M'$ or $vv_{i+2} \in M'$.

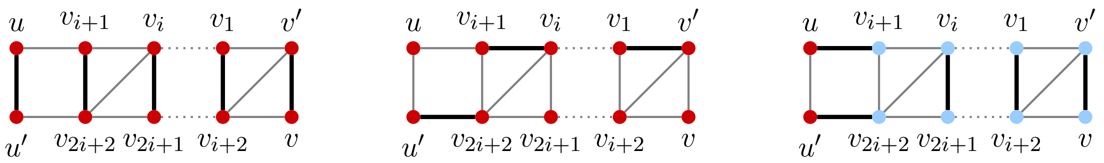
Proof. First assume $c'(u) = c'(v)$, say c' colours u and v red. As any clique of size at least 3 is monochromatic, all vertices $v', v_1, \ldots, v_{2i+2}$ are coloured the same as v, so they are red. Now u' has two red neighbours, and thus u' must be red as well. Hence, G^i_{uv} is monochromatic.

Now assume $c'(u) \neq c'(v)$, say u is red and v blue. As before, we find that $v', v_1, \ldots, v_{2i+2}$ are all coloured the same as v, so they are blue. Now u, which is red, has one blue neighbour, namely v_{i+1}, and thus, u' must be red. So u and u' have the same colour. By definition, every bichromatic edge belongs to M'. Hence, $uv_{i+1} \in M'$. Recall that M' is perfect and observe that $|V(G^i_{uv})|$ is even. As the vertices of $G^i_{uv} - \{u, v\}$ have no neighbours outside $G^i_{uv} - \{u, v\}$, these two facts imply that either $vv' \in M'$ or $vv_{i+2} \in M'$. If $c'(u) \neq c'(w)$ for some neighbour w of u in $G - v$ then, as we just argued, M' matches u with a vertex of G^i_{uw}, a contradiction, as M' is a matching in G'. This proves Claim 5.2.2.1. ◁

We now construct a subset $M \subseteq E$ in G by the following rule: add an edge $uv \in E$ to M if and only if in G', either $c'(u) = c'(v)$ and M' matches both u and v with vertices from G^i_{uv}, or $c'(u) \neq c'(v)$.

We now show that M is a perfect matching in G. Let $u \in V$. First suppose u belongs in G' to a non-monochromatic graph G^i_{uv}. By Claim 5.2.2.1, we have $c'(u) \neq c'(v)$ and $c'(u) = c'(w)$ for every neighbour w of u in $G - v$, and moreover, M' matches both u and v with vertices from G^i_{uv}. Hence, M matches u to v. As $c'(u) = c'(w)$ for every neighbour w of u in $G - v$, each G^i_{uw} is monochromatic. However, as M' already matched u to some vertex from G^i_{uv}, we find that M' (being a matching) does not match u to any vertices of any G^i_{uw} with $w \neq v$. Hence, our rule does not add any edges uw with $w \neq v$ to M. So M matches u only to v.

Now suppose that u only belongs to monochromatic graphs G^i_{uv}. By Claim 5.2.2.1, we have $c'(u) = c'(v)$ for every neighbour v of u in G. As M' is a perfect matching, u has exactly one neighbour v in G, such that M' matches u with a vertex of G^i_{uv}. As $|V(G^i_{uv})|$



(a) u and v have the same colour and $uv \in M$.

(b) u and v have the same colour and $uv \notin M$.

(c) u and v have different colours.

Figure 5.7: The different colourings and matchings in the graph G^i_{uv}.

is even and the vertices of $G^i_{uv} - \{u, v\}$ have no neighbours outside $G^i_{uv} - \{u, v\}$, this means that M' also matches v with a vertex from G^i_{uv}. Hence, M matches u to v and to no other vertex of G. We conclude that M is indeed a perfect matching in G.

Finally, let c be the restriction of c' to V. If c colours every vertex of G with one colour, say red, then c' would also colour every vertex of G' red by Claim 5.2.2.1, contradicting the perfect-extendability (and thus validity) of c'. Hence, c uses both colours. For every $uv \in E$, the following holds: if $c(u) \neq c(v)$, then $c'(u) \neq c'(v)$ and thus $uv \in M$. Hence, as M is a (perfect) matching, c is perfect-extendable. Thus M is a disconnected perfect matching of G.

Now, assume that G has a disconnected perfect matching M, so V has a perfect-extendable red-blue colouring c. We construct a red-blue colouring c' of V' and a matching M' in G' as follows:

- For every edge $uv \in E$ with $c(u) = c(v)$, we let $c'(x) = c(u)$ for every $x \in V(G^i_{uv})$.
  - If $uv \in M$, then we add $vv', v_1v_{i+2}, \dots, v_{i+1}v_{2i+2}, uu'$ to M'; see also Figure 5.7 a).
  - If $uv \notin M$, then we add $v'v_1, v_2v_3, \dots, v_{2i}v_{2i+1}, v_{2i+2}u'$ to M' (recall that i is even, so this is possible); see also Figure 5.7 b).
- For every edge $uv \in E$ with $c(u) \neq c(v)$, we let $c'(u) = c'(u') = c(u)$ and $c'(x) = c(v)$ for every $x \in V(G^i_{uv}) \setminus \{u, u'\}$. We add the edges $uv_{i+1}, u'v_{2i+2}$ to M' as well as $vv', v_1v_{i+2}, \dots, v_iv_{2i+1}$; see also Figure 5.7 c).

As c is valid, c uses both colours and thus by construction, c' uses both colours. Let $u \in V$. Again as c is valid, $c(u) \neq c(v)$ holds for at most one neighbour v of u in G. Hence, by construction, u belongs to at most one non-monochromatic gadget G^i_{uv}. Thus, c' colours in G' at most one neighbour of u with a different colour than u. Let $u \in V' \setminus V$. By construction, we find again that c' colours at most one neighbour of u with a different colour than u. Hence, c' is valid. Again by construction, M' is a perfect matching containing all edges xy of G' with $c'(x) \neq c'(y)$, and thus, M' is a disconnected perfect matching of G'. This completes the proof of Theorem 5.2.2. □

To get the same result for Perfect Matching Cut, we apply Lemma 5.1.8 (implicit in [44]) sufficiently times on every edge of a cubic planar bipartite graph of girth at least g and combine this with the NP-completeness of Perfect Matching Cut for the class of cubic planar bipartite graphs of girth at least g see Theorem 5.1.9 and [5]. Note that applying Lemma 5.1.8 does not change the degree of a graph. Hence, we immediately get the following:

Theorem 5.2.3 ([44, 5]). *For every $i \geq 1$ and $g \geq 3$,* Perfect Matching Cut *is* NP*-complete for $(H_1^*, \dots, H_i^*)$-free subcubic bipartite graphs of girth at least g.*

5.3 Graphs without Induced Paths

In this section we consider graphs without induced long paths. Our result for Matching Cut improved the current bound given by Feghali [20], while the result for Disconnected Perfect Matching was the first to show that there is a constant r, such that Disconnected Perfect Matching is NP-complete for P_r-free graphs. We also consider Maximum Matching Cut and Maximum Disconnected Perfect Matching where we can show that both are NP-hard for $2P_3$-free graphs and with a variant of that proof also that they are NP-hard for bipartite graphs of radius 3 and diameter 4.

5.3.1 Matching Cut and Disconnected Perfect Matching

For Matching Cut and Disconnected Perfect Matching, it was known before that they are solvable in polynomial time on P_5-free graphs (see [8, 20]) which we improved to P_6-free graphs in Theorem 4.1.2 and Corollary 4.2.9. We now present the corresponding hardness proofs. The first one, for Matching Cut, improves the result of Feghali [20], by showing that Matching Cut is NP-complete for P_{15}-free graphs, while the second, for Disconnected Perfect Matching, was the first result to show that there is some constant r such that Disconnected Perfect Matching is NP-complete for P_r-free graphs. We prove the two NP-completeness results in this section by reducing from Exact Positive 1-in-3 SAT.

The proof of Theorem 5.3.1 follows from Feghali's construction [20] after making some minor modifications to it. We use the modified construction as a basis for the proof of Theorem 5.3.5. Recall that Feghali [20] showed that Matching Cut is NP-complete for P_r-free graphs, for some unspecified constant r. We show that $r = 27$ in [20]; see Remark 5.3.3 below.

Theorem 5.3.1 ([50])**.** Matching Cut *is NP-complete for* $(3P_5, P_{15})$*-free graphs.*

Proof. To prove NP-hardness, we will use a reduction from Exact Positive 1-in-3 SAT, which is NP-complete by Theorem 2.1.9. Let $\mathcal{I}$ be an instance of Exact Positive 1-in-3 SAT with variable set X and clause set C. We will build a graph $G_{\mathcal{I}}$ (see also Figures 5.8 and 5.9):

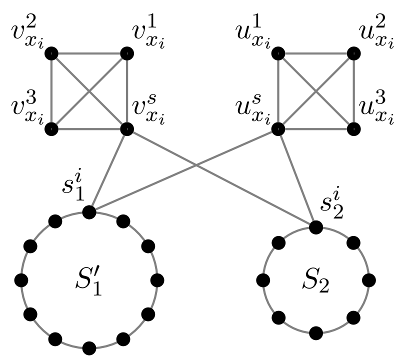


Figure 5.8: The cliques S'_1 and S_2 together with the variable gadget of x_i. For readability, the edges inside S'_1 and S_2 have been omitted. Note that S'_1 consists of all vertices of S_1 and all clause vertices.

- for every $x_i \in X$, construct a variable gadget consisting of two disjoint cliques of size 4, U_{x_i} and V_{x_i}, with vertex set $\{u^s_{x_i}, u^1_{x_i}, u^2_{x_i}, u^3_{x_i}\}$ and $\{v^s_{x_i}, v^1_{x_i}, v^2_{x_i}, v^3_{x_i}\}$, respectively;

- add two cliques S_1 and S_2 with vertex set $\{s_1^1, \dots, s_1^{|X|}\}$ and $\{s_2^1, \dots, s_2^{|X|}\}$, respectively;
- for every $i \in \{1, \dots, |X|\}$, add the edges $s_1^i u_{x_i}^s, s_1^i v_{x_i}^s, s_2^i u_{x_i}^s, s_2^i v_{x_i}^s$;
- for every $c_j \in C$, construct a clause gadget on clause vertices $v_{c_j}, u_{c_j}^1, u_{c_j}^2$, and auxiliary vertices $a_{c_j}^1, \dots, a_{c_j}^6$.
- Add edges between all clause vertices of all clause gadgets to obtain a clique and add all edges between S_1 and this clique. We call the resulting clique S_1'.
- for every $j \in \{1, \dots, |C|\}$, add the edges $u_{c_j}^1 a_{c_j}^\ell$, for $\ell = 1, 2, 3$ and the edges $u_{c_j}^2 a_{c_j}^\ell$, for $\ell = 4, 5, 6$;
- for every $x_i \in X$ occurring in clauses c_{j_1}, c_{j_2} and c_{j_3}, add the edges $v_{x_i}^k v_{c_{j_k}}$, for $k = 1, 2, 3$; and
- for every $c_j \in C$ such that $c_j = x_{i_1} \vee x_{i_2} \vee x_{i_3}$, add the edges $u_{x_{i_k}}^k a_{c_j}^k$ and $u_{x_{i_k}}^k a_{c_j}^{k+3}$, for $k = 1, 2, 3$.

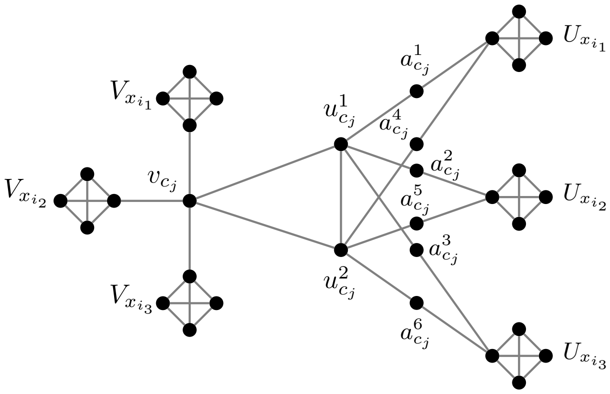


Figure 5.9: The clause gadget for clause $c_j = x_{i_1} \vee x_{i_2} \vee x_{i_3}$.

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

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

Claim 5.3.1.1. *For any variable $x_i \in X$, $i = 1, \dots, |X|$, both V_{x_i} and U_{x_i} are monochromatic. Furthermore, S_1' and S_2 are each monochromatic.*

Proof. This immediately follows from the fact these sets are cliques of size at least 3 and Observation 3.1.2. ◁

Claim 5.3.1.2. *It holds that S_1' and S_2 have different colours.*

Proof. Suppose for a contradiction that S_1' and S_2 have the same colour. We may assume without loss of generality that S_1' and S_2 are both coloured blue. Since for every variable $x_i \in X$, $i = 1, \dots, |X|$, there exist vertices $u_{x_i}^s$ and $v_{x_i}^s$ having each a neighbour in both S_1' and S_2, it follows that every variable gadget is coloured blue. Thus, both neighbours of each auxiliary vertex are blue, which forces the auxiliary vertices to be blue themselves. It follows that all vertices in $G_{\mathcal{I}}$ are coloured blue. Hence, the colouring is not valid, a contradiction. ◁

Claim 5.3.1.3. *For every variable $x_i \in X$, $i \in \{1, \dots, |X|\}$, U_{x_i} and V_{x_i} have different colours.*

Proof. Suppose for a contradiction that for some variable $x_i \in X$, $i \in \{1, \dots, |X|\}$, U_{x_i} and V_{x_i} have the same colour. We may assume without loss of generality that they are both coloured blue. Since s_1^i and s_2^i are both adjacent to $u_{x_i}^s$ and to $v_{x_i}^s$, it follows that they are both coloured blue. Now it follows from Claim 5.3.1.1, that S_1' and S_2 must both be coloured blue, a contradiction to Claim 5.3.1.2. ◁

Claim 5.3.1.4. *For every clause $c_j \in C$, exactly two neighbours of v_{c_j} outside of S_1' have the same colour as v_{c_j}.*

Proof. We may assume without loss of generality that v_{c_j} is coloured blue. Let $c_j = (x_{i_1} \vee x_{i_2} \vee x_{i_3})$. Without loss of generality, let $v_{x_{i_1}}^1, v_{x_{i_2}}^2, v_{x_{i_3}}^3$ be the three neighbours of v_{c_j} outside of S_1' and let $u_{x_{i_1}}^1, u_{x_{i_2}}^2$ and $u_{x_{i_3}}^3$ be the neighbours of the auxiliary vertices $a_{c_j}^k$, $k = 1, \dots, 6$. By definition of a valid red-blue-colouring, v_{c_j} has at least two neighbours outside of S_1' that are coloured blue. Suppose for a contradiction that the vertices $v_{x_{i_1}}^1, v_{x_{i_2}}^2, v_{x_{i_3}}^3$ are all coloured blue. Then, it follows from Claim 5.3.1.1, that $V_{x_{i_1}}, V_{x_{i_2}}, V_{x_{i_3}}$ must be coloured blue. Since v_{c_j} is coloured blue, Claim 5.3.1.1 also implies that all vertices in S_1', and in particular $u_{c_j}^1$ and $u_{c_j}^2$, are coloured blue. Let A_1 (resp. A_2) be the set of auxiliary vertices which are adjacent to $u_{c_j}^1$ (resp. $u_{c_j}^2$). Then, at least two vertices in A_1 and two vertices in A_2 are coloured blue. Since $u_{x_{i_1}}^1, u_{x_{i_2}}^2$ and $u_{x_{i_3}}^3$ have each one neighbour in A_1 and one neighbour in A_2, it follows that one of them has two blue neighbours in $A_1 \cup A_2$, and is therefore coloured blue. We may assume without loss of generality that this vertex is $u_{x_{i_1}}^1$. Using Claim 5.3.1.1 again, we get that $U_{x_{i_1}}$ is coloured blue, a contradiction to Claim 5.3.1.3. ◁

We continue as follows. By Claim 5.3.1.1, we may assume without loss of generality that S_1' is coloured blue. Then, we set every variable $x_i \in X$, $i \in \{1, \dots, |X|\}$, to true for which V_{x_i} has been coloured red. We set all other variables to false. By Claim 5.3.1.4, we know that for each clause $c_j \in C$, $j \in \{1, \dots, |C|\}$, there exists exactly one red neighbour of v_{c_j}. Hence, in every clause, exactly one literal is set to true. Since by Claim 5.3.1.1, every V_{x_i}, $i \in \{1, \dots, |X|\}$, is monochromatic, it follows that no variable gets both values true and false. Thus, $\mathcal{I}$ admits a truth assignment such that each clause contains exactly one true literal.

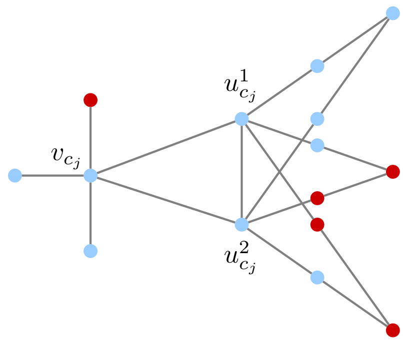


Figure 5.10: The given valid red-blue-colouring of the clause gadget for c_j.

Now suppose that $\mathcal{I}$ admits a truth assignment such that each clause contains exactly one true literal. For every variable $x_i \in X$, $i \in \{1, \dots, |X|\}$, that is set to true, we colour V_{x_i} red and U_{x_i} blue; for every other variable $x_i \in X$, we colour V_{x_i} blue and U_{x_i} red. It follows that every vertex v_{c_j}, for $c_j \in C$ and $j \in \{1, \dots, |C|\}$, has exactly one red and two blue neighbours outside of S_1', since the truth assignment is such that each clause contains exactly one true literal. Thus, we colour S_1' blue. If we consider a clause $c_j = (x_{i_1} \vee x_{i_2} \vee x_{i_3})$, we know that exactly one of $U_{x_{i_1}}, U_{x_{i_2}}, U_{x_{i_3}}$ is blue and the other ones are red. Assume without loss of generality that $U_{x_{i_1}}$ is blue and consider $a^1_{c_j}, a^4_{c_j}$, the neighbours of $u^1_{x_{i_1}} \in U_{x_{i_1}}$. We then colour $a^1_{c_j}, a^4_{c_j}$ blue, since they have two blue neighbours ($u^1_{x_{i_1}}$ and $u^1_{c_j}$ resp. $u^2_{c_j}$). To obtain a valid red-blue colouring of $G_{\mathcal{I}}$, we have to colour one of the vertices $a^2_{c_j}, a^5_{c_j}$ blue and the other one red, and similarly, one of the vertices $a^3_{c_j}, a^6_{c_j}$ blue and the other one red. Since $u^1_{c_j}, u^2_{c_j}$ are both coloured blue and each of them can have at most one red neighbour, we colour $a^2_{c_j}, a^6_{c_j}$ blue and $a^3_{c_j}, a^5_{c_j}$ red (see also Figure 5.10). Finally, the only vertices that remain uncoloured are the vertices in S_2. The cliques S'_1 and S_2 are coloured differently. Hence, as S'_1 is coloured blue, we must colour S_2 red. This gives us a valid red-blue-colouring of $G_{\mathcal{I}}$.

To complete the proof, it remains to show that $G_{\mathcal{I}}$ is $(3P_5, P_{15})$-free. We first prove that $G_{\mathcal{I}}$ is P_{15}-free. Let P be a longest induced path in $G_{\mathcal{I}}$. Then P can contain at most two vertices from a same clique, since otherwise P would not be induced. Moreover, if P contains two vertices from a clique, then these two vertices must be consecutive in P. Let W_1 and W_2 be two disjoint cliques in $\{U_{x_i}, V_{x_i} | \; x_i \in X\}$. By construction, every path from a vertex in W_1 to a vertex in W_2 contains at least one vertex from one of the cliques S'_1 or S_2. Hence, P can intersect at most three cliques belonging to some variable gadgets. Moreover, in the case where P intersects three cliques belonging to some variable gadgets, P intersects each of the cliques S'_1 and S_2 as well. Assume that P intersects W_1, S'_1 and W_2 in this order. Then P may contain at most one auxiliary vertex between intersecting W_1 and S'_1 and at most one between intersecting S'_1 and W_2; see Figure 5.11 for an example. Hence, P contains at most 14 vertices. As P was a longest path in $G_{\mathcal{I}}$, we conclude that $G_{\mathcal{I}}$ is P_{15}-free.

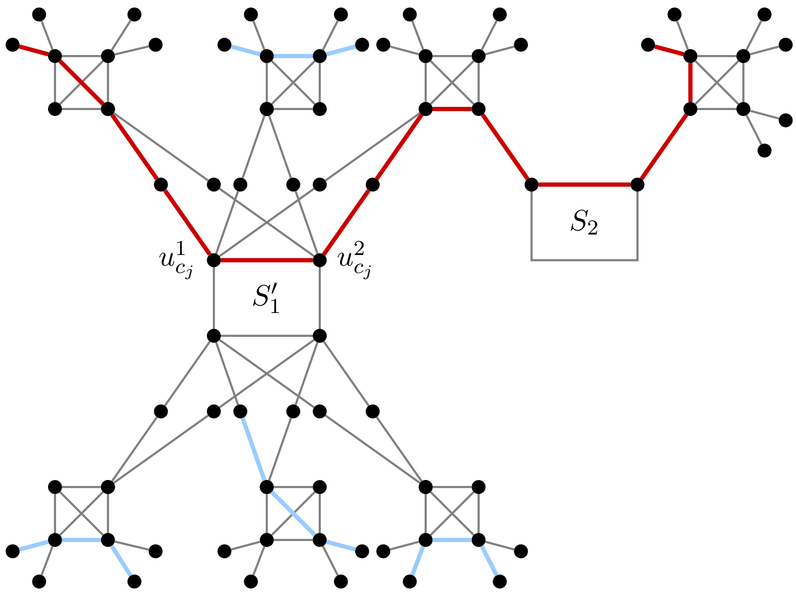


Figure 5.11: An induced $P_{14} + 4P_4$, where the induced P_{14} is displayed in red and the four induced P_4s are in blue.

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

Remark 5.3.2. It can be verified that the $(3P_5, P_{15})$-free graph $G_{\mathcal{I}}$ in the proof of Theorem 5.3.1 is not $(P_{14} + sP_4)$-free for any $s \geq 1$; see also Figure 5.11, which displays an induced $P_{14} + 4P_4$.

Remark 5.3.3. In the graph $G_{\mathcal{I}}$ from the proof of Theorem 5.3.1, no vertex of U_{x_i} is adjacent to a vertex of V_{x_i}, for any $x_i \in X$. In Feghali's construction [20], there is an edge between any two such cliques. This implies that an induced path can use four consecutive vertices inside the same variable gadget. Via similar arguments as in our proof, one can show that Feghali's construction has an induced P_{26}, but no induced P_{27} (and thus it is P_{27}-free).

Remark 5.3.4. After [50] appeared, Le and Le improved this result ([40]) and showed that Matching Cut is NP-complete even for $(3P_6, 2P_7, P_{14})$-free graphs.

We now modify the construction in the proof of Theorem 5.3.1 to obtain Theorem 5.3.5. This result addresses Question 3 by showing that Disconnected Perfect Matching is NP-complete for P_{19}-free graphs. Note that before it was not known if there is an integer r such that Disconnected Perfect Matching is NP-complete for P_r-free graphs.

To obtain this result, it is possible to use the method that we used to adapt the proof that Matching Cut is NP-hard for graphs of fixed girth to obtain the same result for Disconnected Perfect Matching. That is, we could take two copies of P_{15}-free graph and connect the two copies of a vertex using a suitable gadget. The gadget has to ensure that both copies of the vertex have the same colour in any valid red-blue colouring. But it should not increase the maximum pathlength too much. In this case, we could for example connect the vertices by cliques. However this approach increases the maximum pathlength to $2 \cdot 14 = 28$. In the following version of the proof, we modify the construction of Theorem 5.3.1 and do not duplicate everything and thus, we get a better bound on the pathlength.

Theorem 5.3.5 ([50])**.** Disconnected Perfect Matching *is NP-complete for* $(3P_7, P_{19})$-*free graphs.*

Proof. In order to prove NP-hardness, we reduce from Exact Positive 1-in-3 SAT, which is NP-complete by Theorem 2.1.9. Let $\mathcal{I}$ be an instance of Exact Positive 1-in-3 SAT with variable set X and clause set C. We build a graph $G_{\mathcal{I}}$ similarly to the graph in Theorem 5.3.1 (see also Figures 5.12 and 5.13):

- for every $x_i \in X$, construct a variable gadget consisting of two disjoint cliques of size 7, U_{x_i} and V_{x_i}, with vertex set $\{u^s_{x_i}, u^1_{x_i}, \ldots, u^6_{x_i}\}$ and $\{v^s_{x_i}, v^1_{x_i}, \ldots, v^6_{x_i}\}$, respectively;
- add two cliques S_1 and S_2 with vertex set $\{s^1_1, \ldots, s^{|X|}_1\}$ and $\{s^1_2, \ldots, s^{|X|}_2\}$, respectively;
- for every $i \in \{1, \ldots, |X|\}$, add the edges $s^i_1 u^s_{x_i}, s^i_1 v^s_{x_i}, s^i_2 u^s_{x_i}, s^i_2 v^s_{x_i}$;
- for every $c_j \in C$, construct a clause gadget on clause vertices $v^1_{c_j}, v^2_{c_j}, u^1_{c_j}, u^2_{c_j}, u^3_{c_j}, u^4_{c_j}$, and auxiliary vertices $a^1_{c_j}, \ldots, a^{12}_{c_j}$.
- Add edges between all clause vertices of all clause gadgets to obtain a clique and add all edges between this clique and S_1. We call the resulting clique S_1.
- for every $j \in \{1, \ldots, |C|\}$, add the edges $u^1_{c_j} a^\ell_{c_j}$, for $\ell = 1, 2, 3$, $u^2_{c_j} a^\ell_{c_j}$, for $\ell = 4, 5, 6$, $u^3_{c_j} a^\ell_{c_j}$, for $\ell = 7, 8, 9$ and $u^4_{c_j} a^\ell_{c_j}$, for $\ell = 10, 11, 12$. Add also the edges $a^\ell_{c_j} a^{\ell+6}_{c_j}$, for $\ell = 1, \ldots, 6$;
- for every $x_i \in X$ occurring in clauses c_{j_1}, c_{j_2} and c_{j_3}, add the edges $v^k_{x_i} v^1_{c_{j_k}}$ and $v^{k+3}_{x_i} v^2_{c_{j_k}}$, for $k = 1, 2, 3$; and

- for every $c_j \in C$ such that $c_j = x_{i_1} \vee x_{i_2} \vee x_{i_3}$, add the edges $u^k_{x_{i_k}} a^k_{c_j}$, $u^k_{x_{i_k}} a^{k+3}_{c_j}$, $u^{k+3}_{x_{i_k}} a^{k+6}_{c_j}$ and $u^{k+3}_{x_{i_k}} a^{k+9}_{c_j}$, for $k = 1, 2, 3$.

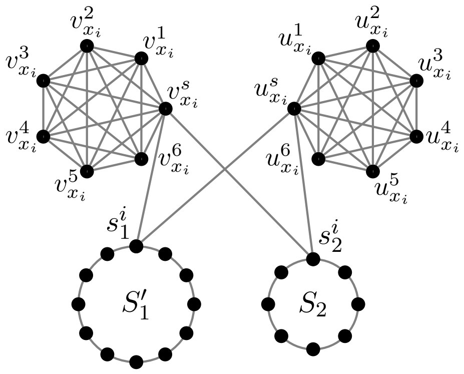


Figure 5.12: The cliques S'_1 and S_2, together with the variable gadget of x_i. For readability, the edges inside S'_1 and S_2 have been omitted. Note that S'_1 consists of all vertices of S_1 and all clause vertices.

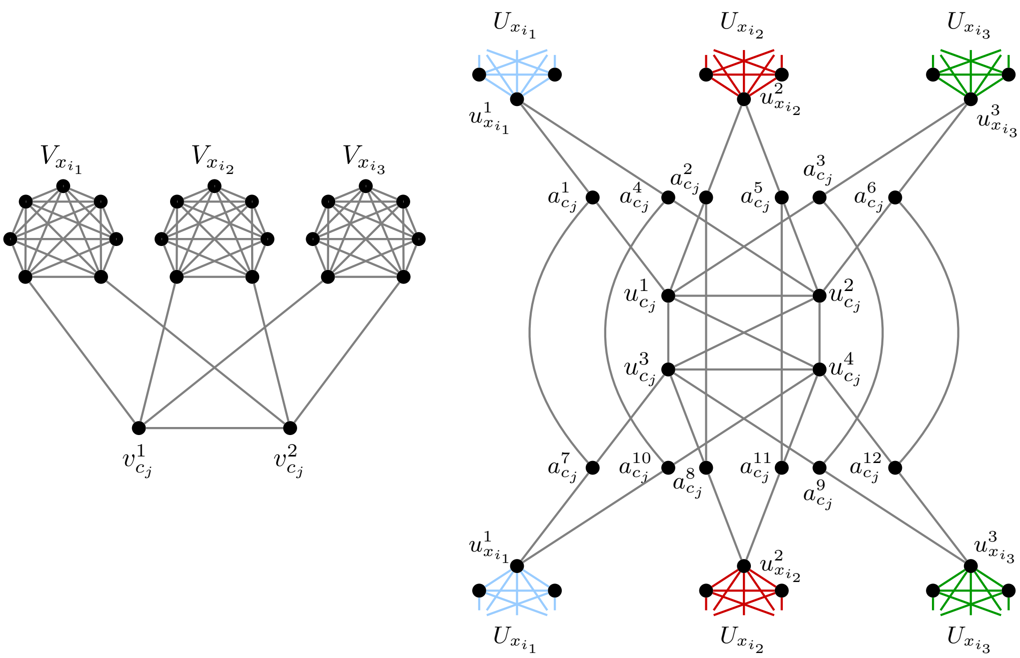


Figure 5.13: The clause gadget for the clause $c_j = x_{i_1} \vee x_{i_2} \vee x_{i_3}$. The vertices $v^1_{c_j}, v^2_{c_j}, u^1_{c_j}, \ldots, u^4_{c_j}$ belong to clique S'_1; for readability, we omitted the edges between $v^1_{c_j}, v^2_{c_j}$ and $u^1_{c_j}, \ldots, u^4_{c_j}$. Moreover, coloured edges of the same colour belong to the same clique $U_{x_{i_j}}$, for $j = 1, 2, 3$ (again, for readability).

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

First suppose that $G_{\mathcal{I}}$ admits a valid perfect-extendable red-blue-colouring. We start with some useful claims, the first three have the same proof as Claims 5.3.1.1, 5.3.1.2, 5.3.1.3 in the proof of Theorem 5.3.1.

Claim 5.3.5.1. *For any variable $x_i \in X$, $i = 1, \ldots, |X|$, both V_{x_i} and U_{x_i} are monochromatic. Furthermore, S'_1 and S_2 are also monochromatic.*

Claim 5.3.5.2. *It holds that S'_1 and S_2 have different colours.*

Claim 5.3.5.3. *For every variable $x_i \in X$, $i \in \{1, \dots, |X|\}$, the cliques U_{x_i} and V_{x_i} have different colours.*

Claim 5.3.5.4. *For every clause $c_j \in C$, $j \in \{1, \dots, |C|\}$, exactly two neighbours of $v^1_{c_j}$ (resp. $v^2_{c_j}$) outside of S'_1 have the same colour as $v^1_{c_j}$ (resp. $v^2_{c_j}$).*

Proof. We may assume without loss of generality that $v^1_{c_j}$ (resp. $v^2_{c_j}$) is coloured blue. By symmetry, it is enough to prove the claim for $v^1_{c_j}$. Let $c_j = (x_{i_1} \vee x_{i_2} \vee x_{i_3})$. Without loss of generality, let $v^1_{x_{i_1}}, v^2_{x_{i_2}}, v^3_{x_{i_3}}$ be the three neighbours of $v^1_{c_j}$ outside of S'_1 and let $u^1_{x_{i_1}}, u^2_{x_{i_2}}$ and $u^3_{x_{i_3}}$ be the neighbours of the auxiliary vertices $a^k_{c_j}$, $k = 1, \dots, 6$. By definition of a perfect-extendable red-blue-colouring, $v^1_{c_j}$ has at least two neighbours outside of S'_1 that are coloured blue. Suppose for a contradiction that the vertices $v^1_{x_{i_1}}, v^2_{x_{i_2}}, v^3_{x_{i_3}}$ are all coloured blue. Then, it follows from Claim 5.3.5.1 that $V_{x_{i_1}}, V_{x_{i_2}}, V_{x_{i_3}}$ must be coloured blue. Since $v^1_{c_j}$ is coloured blue, Claim 5.3.5.1 also implies that all vertices in S'_1, in particular $u^1_{c_j}$ and $u^2_{c_j}$ are coloured blue. Let A_1 (resp. A_2) be the set of auxiliary vertices which are a neighbour of $u^1_{c_j}$ (resp. $u^2_{c_j}$). Then, at least two vertices in A_1 and two vertices in A_2 are coloured blue. Since $u^1_{x_{i_1}}, u^2_{x_{i_2}}$ and $u^3_{x_{i_3}}$ have each one neighbour in A_1 and one neighbour in A_2, it follows that one of them has two blue neighbours in $A_1 \cup A_2$, and is thus coloured blue. We may assume without loss of generality that this vertex is $u^1_{x_{i_1}}$. Using Claim 5.3.5.1 again, we get that $U_{x_{i_1}}$ is coloured blue, a contradiction to Claim 5.3.5.3.

Since $v^1_{c_j}$ and $v^2_{c_j}$ are both in S'_1, using Claim 5.3.5.1 we get that they are both blue. For each neighbour of $v^1_{c_j}$ there is a neighbour of $v^2_{c_j}$ in the same clique and vice versa. Hence, they have the same number of blue neighbours. ◁

We continue as follows. By Claim 5.3.5.1, we may assume without loss of generality that S'_1 is coloured blue. We set every variable $x_i \in X$ for which V_{x_i} has been coloured red to true. We set all other variables to false. By Claim 5.3.5.4, we know that for each clause $c_j \in C$, $j \in \{1, \dots, |C|\}$, there is exactly one red neighbour of $v^1_{c_j}$ (resp. $v^2_{c_j}$). Hence, in every clause, exactly one literal is set to true. Since by Claim 5.3.5.1, every V_{x_i}, $i \in \{1, \dots, |X|\}$, is monochromatic, it follows that no variable is both true and false. Thus, $\mathcal{I}$ admits a truth assignment such that each clause contains exactly one true literal.

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

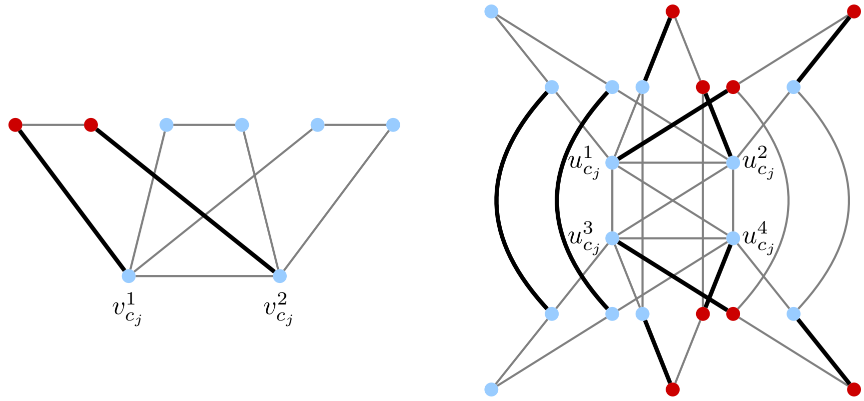


Figure 5.14: A perfect-extendable red-blue-colouring of the clause gadget. Notice that vertices $v^1_{c_j}, v^2_{c_j}, u^1_{c_j}, \ldots, u^4_{c_j}$ belong to the clique S'_1, but for the sake of readability, we omitted the edges between vertices $v^1_{c_j}, v^2_{c_j}$ and vertices $u^1_{c_j}, \ldots, u^4_{c_j}$.

We must still verify that this valid red-blue-colouring is perfect-extendable. First, consider the vertices s^i_j, for $i \in \{1, \ldots, |X|\}$ and $j \in \{1, 2\}$. We know that each vertex s^i_j has two neighbours $v^s_{x_i}$ and $u^s_{x_i}$ outside of $S'_1 \cup S_2$. Since the red-blue-colouring is valid, it follows that exactly one of s^i_1, s^i_2 and exactly one of $v^s_{x_i}$ and $u^s_{x_i}$ is blue. So every vertex s^i_j has a neighbour of the opposite colour and hence, can be matched with it. The same holds for the vertices $v^s_{x_i}, u^s_{x_i}$, for all $x_i \in X$.

Next, consider a clause $c_j \in C$, such that $c_j = x_{i_1} \vee x_{i_2} \vee x_{i_3}$. It follows from the construction of our valid red-blue-colouring that we may assume without loss of generality that $V_{x_{i_1}}, U_{x_{i_2}}, U_{x_{i_3}}$ are coloured red and $V_{x_{i_2}}, V_{x_{i_3}}, U_{x_{i_1}}$ are coloured blue. Since every clause vertex is coloured blue and has exactly one red neighbour, it follows that all clause vertices in the clause gadget of c_j can be matched and the same holds for the vertices in $V_{x_{i_1}} \setminus \left\{v^s_{x_{i_1}}\right\}$. The auxiliary vertices which are adjacent to vertices in $U_{x_{i_2}} \cup U_{x_{i_3}}$ have exactly one neighbour of the opposite colour and thus, they can be matched with either clause vertices or vertices in $U_{x_{i_2}} \cup U_{x_{i_3}}$ (see also Figure 5.14). The auxiliary vertices $a^1_{c_j}, a^4_{c_j}, a^7_{c_j}$ and $a^{10}_{c_j}$, which are neighbours of $U_{x_{i_1}}$, are all coloured blue. In this case, we consider the edges $a^1_{c_j}a^7_{c_j}$ and $a^4_{c_j}a^{10}_{c_j}$ as matching edges.

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

To complete the proof, it remains to show that $G_{\mathcal{I}}$ is indeed $(3P_7, P_{19})$-free. To show that it is P_{19}-free, we follow the same arguments as in the proof of Theorem 5.3.1, where we showed that the corresponding graph was P_{15}-free. Let P be a longest induced path in $G_{\mathcal{I}}$. It holds that P can contain at most two vertices from each clique, else P would not be induced. Also, if P contains two vertices from the same clique, then these two vertices are necessarily consecutive in P. Let W_1, W_2 be two disjoint cliques in $\{U_{x_i}, V_{x_i} | \ x_i \in X\}$. Every path from a vertex in W_1 to a vertex in W_2 contains at least one vertex from one of the cliques S'_1 and S_2. Hence, P can intersect at most three cliques belonging to some variable gadgets. Moreover, in the case where P does intersect three cliques belonging to some variable gadgets, then P intersects each of the cliques S'_1 and S_2 as well. Directly before intersecting S'_1 the path P may contain up to two auxiliary vertices (see also Figure 5.15). Similarly, directly after intersecting S'_1, the path P may contain another two auxiliary vertices. Moreover, the first two, respectively the last two, vertices in P may correspond to auxiliary vertices. Hence, P contains at most 18 vertices, and thus, $G_{\mathcal{I}}$ is indeed P_{19}-free.

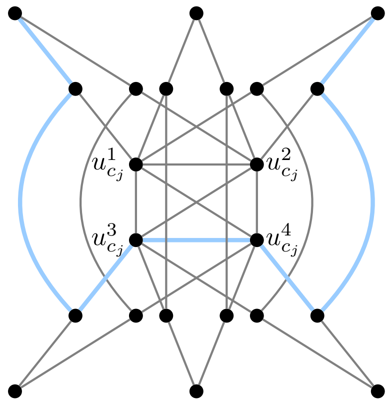


Figure 5.15: A path intersecting a clause gadget and containing four auxiliary vertices.

We now prove that $G_{\mathcal{I}}$ is $3P_7$-free. Let $G' = G[V(G_{\mathcal{I}}) \setminus (V(S_1') \cup V(S_2)]$. The graph G' consists of two types of connected components: (i) every V_{x_i}, for $x_i \in X$, corresponds to a clique of size 7; (ii) every U_{x_i}, for $x_i \in X$, together with some auxiliary vertices corresponds to a connected component as shown in Figure 5.16. Any induced path contains at most two vertices of the same clique. Also observe that any induced path contains at most six vertices from a connected component containing a clique U_{x_i} (see Figure 5.16). Hence, every induced path with length at least 7 has to contain a vertex in one of the cliques S_1' and S_2. It immediately follows that $G_{\mathcal{I}}$ is $3P_7$-free. $\square$

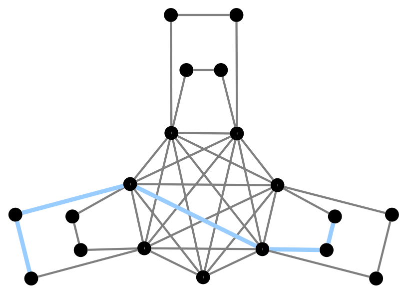


Figure 5.16: A connected component of type (ii) obtained from the removal of S_1' and S_2 and a longest path in this connected component in blue.

Remark 5.3.6. Just like we showed in Remark 5.3.2 that the gadget in the proof of Theorem 5.3.1 is not $(P_{14} + sP_4)$-free for any integer $s \geq 0$, it can be verified that the $(3P_7, P_{19})$-free graph $G_{\mathcal{I}}$ in the proof of Theorem 5.3.5 is not $(P_{18} + sP_6)$-free for any $s \geq 1$.

Remark 5.3.7. As for Theorem 5.3.1, after [50] appeared, Le and Le improved this result ([40]) and showed that Disconnected Perfect Matching is NP-complete even for $(3P_6, 2P_7, P_{14})$-free graphs.

5.3.2 Maximum Matching Cut

For Maximum Matching Cut and Maximum Disconnected Perfect Matching much stronger results can be proven. We show in the following that both are NP-hard even for $2P_3$-free graphs with radius 2 and diameter 3. We modify the construction to show that they are NP-hard also for bipartite graphs of radius 3 and diameter 4.

We start with the result 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. We reduce from Exact 3-Cover.

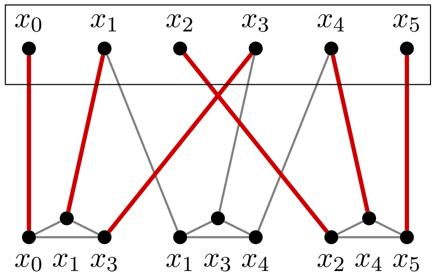


Figure 5.17: The graph G for variables $X = \{x_0, \dots, x_5\}$ and sets $\mathcal{S} = \{\{x_0, x_1, x_3\}, \{x_1, x_3, x_4\}, \{x_2, x_4, x_5\}\}$. The vertices in the rectangle form a clique. The set $\mathcal{S}' = \{\{x_0, x_1, x_3\}, \{x_2, x_4, x_5\}\}$ is an exact 3-cover of X. The thick red edges in the graph show the corresponding matching cut of size $3q = 6$.

Theorem 5.3.8 ([51])**.** Maximum Matching Cut *is NP-hard for* $2P_3$*-free quadrangulated graphs of radius at most* 2 *and diameter at most* 3.

Proof. Let $(X, \mathcal{S})$ be an instance of Exact 3-Cover where $X = \{x_1, \dots, x_{3q}\}$ and $\mathcal{S} = \{S_1, \dots, S_k\}$, such that each S_i contains exactly three elements of X. From $(X, \mathcal{S})$ we construct a graph G. We first define a clique $K_X = \{x_1, \dots, x_{3q}\}$. For each $S \in \mathcal{S}$, we do as follows. Let $S = \{x_h, x_i, x_j\}$. We add 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. See Figure 5.17 for an example.

As every induced P_3 must contain at least one vertex from the clique K_X, we find that G is $2P_3$-free. As G is not only $2P_3$-free, but also (C_5, C_6, C_7)-free, G is quadrangulated. Consider some $x_i \in K_X$. Then every other vertex is of distance at most 2 from x_i. Consider some $x_i^S \in K_S$ for some $S \in \mathcal{S}$. Then every other vertex is of distance at most 3 from x_i^S. Hence, the radius of G is at most 2 and the diameter of G is at most 3.

We claim that $\mathcal{S}$ contains an exact 3-cover of X if and only if G has a matching cut of size $3q$. First suppose that $\mathcal{S}$ contains an exact 3-cover $\mathcal{C}$ of X. We colour every vertex of K_X red. We colour a triangle K_S blue if $S \in \mathcal{C}$ and otherwise we colour it red. This yields a valid red-blue colouring of value $3q$, and thus a matching cut of size $3q$.

Now suppose that G has a matching cut M of size $3q$. As K_X is a clique of size $3q \geq 3$, the corresponding valid red-blue colouring assigns every vertex of K_X the same colour, say red. As every triangle K_S is monochromatic, this means that exactly q triangles must be coloured blue. Moreover, no two blue triangles have a common red neighbour in K_X. Hence, the blue triangles correspond to an exact 3-cover of X. See again Figure 5.17. $\square$

The construction in the proof of Theorem 5.3.8 can be adapted to be bipartite. To do so, we replace the clique K_X and the triangles K_S by complete bipartite graphs. This increases both the radius and the diameter by 1. Also longer paths and cycles become possible. We give the full proof for completeness.

Theorem 5.3.9. Maximum Matching Cut *is NP-hard for bipartite graphs of radius at most* 3 *and diameter at most* 4.

Proof. Let $(X, \mathcal{S})$ be an instance of Exact 3-Cover where $X = \{x_1, \dots, x_{3q}\}$ and $\mathcal{S} = \{S_1, \dots, S_k\}$, such that each S_i contains exactly three elements of X. From $(X, \mathcal{S})$ we construct a graph G as follows.

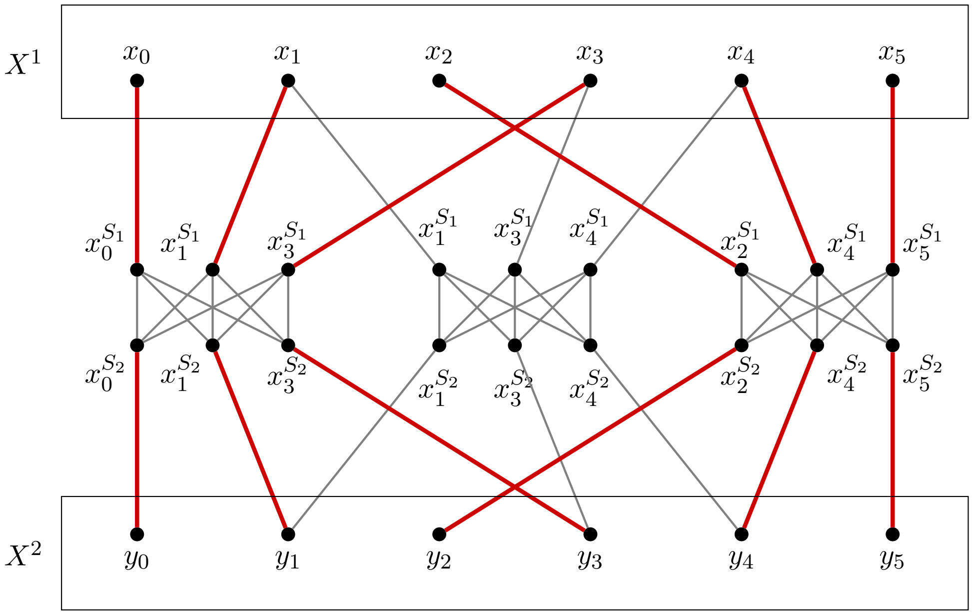


Figure 5.18: The graph G for variables $X = \{x_0, \dots, x_5\}$ and sets $\mathcal{S} = \{\{x_0, x_1, x_3\}, \{x_1, x_3, x_4\}, \{x_2, x_4, x_5\}\}$. The set $\mathcal{S}' = \{\{x_0, x_1, x_3\}, \{x_2, x_4, x_5\}\}$ is an exact 3-cover of X. The thick red edges in the graph show the corresponding matching cut of size $6q = 12$. X^1 and X^2 are the partition classes of a complete bipartite graph. For readability the edges between X^1 and X^2 are not represented.

We first define a complete bipartite graph $K_{3q,3q}$ called K_X with partition classes $X^1 = \{x_1, \dots, x_{3q}\}$ and $X^2 = \{y_1, \dots, y_{3q}\}$. For each $S \in \mathcal{S}$, we do as follows. Let $S = \{x_h, x_i, x_j\}$. We add a complete bipartite graph $K_{3,3}$ called K_S with partition classes $S^1 = \{x_h^{S^1}, x_i^{S^1}, x_j^{S^1}\}$ and $S^2 = \{x_h^{S^2}, x_i^{S^2}, x_j^{S^2}\}$. We add an edge between a vertex $x_i \in X^1$ and a vertex $u \in S^1$ if and only if $u = x_i^{S^1}$ for some $S \in \mathcal{S}$. Similarly, we add an edge between a vertex $y_i \in X^2$ and a vertex $u \in S^2$ if and only if $u = x_i^{S^2}$ for some $S \in \mathcal{S}$. This completes the construction of G. See Figure 5.18 for an example.

First note that this graph is indeed bipartite. The partition classes are $X^1 \cup \bigcup_{1 \le i \le k} S_i^2$ and $X^2 \cup \bigcup_{1 \le i \le k} S_i^1$. To see that G has radius 3 pick a vertex v in X^1. All vertices in K_X have distance at most 2 to v. Further, all vertices in some K_S are at distance at most 3 to v. To see this note that there is a path using exactly one vertex in each of the following sets. For $u \in S^2$, for some $S \in \mathcal{S}$, go from $v \in X^1$ to X^2, then to u. For $u \in S^1$, for some $S \in \mathcal{S}$, go from $v \in X^1$ to X_2, then to S^2 and then $u \in S^1$.

To see that G has also diameter 4, it remains to show that the distance between $v \in S^1$ to a vertex in $K_{S'}, S' \in \mathcal{S}$ is at most 4. Note that the case where $v \in S^2$ follows by symmetry. Let $v \in S^1$, for some $S \in \mathcal{S}$. First, note that all vertices in K_S have distance at most 2 to v. Let $S' \in \mathcal{S}$ be different from S and let $u \in S'^2$. Then there is a path of length 3 from v to u using first the neighbour of v in X^1 and then the neighbour of u in X^2. Note that these two vertices are adjacent. Let now $u \in S'^1$, then there is a path of length 4 from v to u using exactly one vertex of each of X^1, X^2, S'^2. Thus, the diameter of G is 4.

We claim that $\mathcal{S}$ contains an exact 3-cover of X if and only if G has a matching cut of size $6q$. First suppose that $\mathcal{S}$ contains an exact 3-cover $\mathcal{C}$ of X. We colour every vertex of K_X red. We colour a graph K_S blue if $S \in \mathcal{C}$ and red otherwise. This yields a valid red-blue colouring of value $6q$, and thus a matching cut of size $6q$.

Now suppose that G has a matching cut M of size $6q$. As K_X and K_S for all $S \in \mathcal{S}$ are complete bipartite graphs with at least 3 vertices on each side, they are monochromatic, see Observation 3.1.2. Say K_X is coloured red. Thus, exactly q subgraphs K_S are coloured blue. Moreover, no two blue K_S have a common red neighbour in K_X. Hence, the blue K_S correspond to an exact 3-cover of X. □

For Maximum Disconnected Perfect Matching we get the same results by modifying the proofs given above. To make the proof of Theorem 5.3.8 hold for Maximum Disconnected Perfect Matching, small modifications to the construction are needed. Thus, we give the full proof for completeness.

Theorem 5.3.10 ([51]). Maximum Disconnected Perfect Matching *is NP-hard for $2P_3$-free quadrangulated graphs of radius at most* 2 *and diameter at most* 3.

Proof. Let $(X, \mathcal{S})$ be an instance of Exact 4-Cover where $X = \{x_1, \dots, x_{4q}\}$ and $\mathcal{S} = \{S_1, \dots, S_k\}$, such that each S_i contains exactly four elements of X. From $(X, \mathcal{S})$ we construct a graph G. We first define a clique $K_X = \{x_1, \dots, x_{4q}\}$. For each $S \in \mathcal{S}$, we do as follows. Let $S = \{x_h, x_i, x_j, x_\ell\}$. We add a clique K_S on four vertices x_h^S, x_i^S, x_j^S and x_ℓ^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.

As every induced P_3 must contain at least one vertex from the clique K_X, we find that G is $2P_3$-free. As G is not only $2P_3$-free, but also (C_5, C_6, C_7)-free, G is quadrangulated. Consider some $x_i \in K_X$. Then every other vertex is of distance at most 2 from x_i. Consider some $x_i^S \in K_S$ for some $S \in \mathcal{S}$. Then every other vertex is of distance at most 3 from x_i^S. Hence, the radius of G is at most 2 and the diameter of G is at most 3.

We claim that $\mathcal{S}$ contains an exact 4-cover of X if and only if G has a matching cut of size $4q$. First suppose that $\mathcal{S}$ contains an exact 4-cover $\mathcal{C}$ of X. We colour every vertex of K_X red. We colour a clique K_S blue if $S \in \mathcal{S}$ and otherwise we colour it red. This yields a valid red-blue colouring of value $4q$, and thus a matching cut of size $4q$. Further, all vertices which are not incident to a bichromatic edge are in some clique K_S and for a clique K_S either all vertices are matched or none. Since K_S contains 4 vertices, these can be matched and thus the matching cut can be extended to a perfect disconnected matching.

Now suppose that G has a matching cut M of size $4q$. As K_X is a clique of size $4q \geq 3$, the corresponding valid red-blue colouring assigns every vertex of K_X the same colour, say red. As every clique K_S is monochromatic, this means that exactly q cliques K_S must be coloured blue. Moreover, no two blue cliques K_S have a common red neighbour in K_X. Hence, the blue cliques K_S correspond to an exact 4-cover of X. □

We can also give the analogue result for Maximum Disconnected Perfect Matching to the result presented in Theorem 5.3.9 for Maximum Matching Cut. Note that in contrast to Theorem 5.3.10 for this result no changes to the construction are needed to make the maximum matching cut extendable to a perfect matching. Since we have an exact 3-cover, if there is a matching cut in the graph, then all vertices in K_X are matched. Further, each K_S is either completely matched to K_X or no vertex of K_S has a matching partner in this matching cut. In this case, we can match every vertex in K_S with another vertex from the same K_S, since K_S is a complete bipartite graph with partition classes of equal size. Thus, after adding the above argument, the proof of Theorem 5.3.9 leads to the following theorem.

Theorem 5.3.11. Maximum Disconnected Perfect Matching *is NP-hard for graphs of radius at most* 3 *and diameter at most* 4.

5.4 Line Graphs

In this last section of the chapter we show the NP-hardness of Maximum Matching Cut and Maximum Disconnected Perfect Matching for line graphs of triangle-free graphs. Recall that the class of line graphs is contained in the class of claw-free graphs. Thus, our results for Maximum Matching Cut and Maximum Disconnected Perfect Matching differ from the known results for Matching Cut, Disconnected Perfect Matching and Perfect Matching Cut, where we know for all three variants that they are solvable in polynomial time for claw-free graphs, see [6, 8, 44].

To prove this result, we reduce from Max Cut on subcubic graphs which is NP-hard by Theorem 2.1.3.

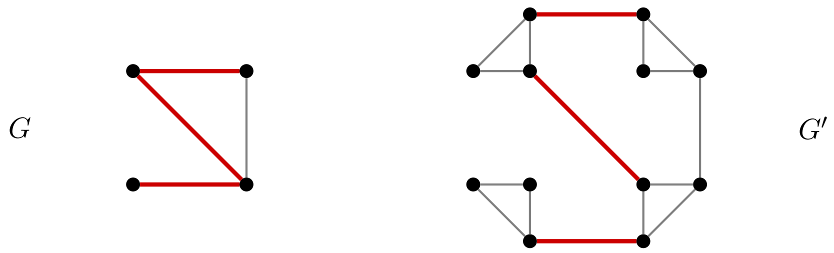


Figure 5.19: A graph G (left) where the thick red edges form a maximum edge cut, and the graph G' (right) from the proof of Theorem 5.4.1, where the thick red edges form a maximum matching cut.

Theorem 5.4.1 ([51]). Maximum Matching Cut *is NP-hard for subcubic line graphs of triangle-free graphs.*

Proof. Let (G, k) be an instance of Maximum Cut, where G is a subcubic graph. From G, we construct a graph G' as follows. First replace every vertex $v \in V(G)$ by a triangle C_v. Next, 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 of C_v. This is possible since G is subcubic. See Figure 5.19 for an example. The graph G' is 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, or equivalently, the line graph of a triangle-free graph.

We claim that G has an edge cut of size at least k if and only if G' has a matching cut of size at least k.

First suppose that G has an edge cut M of size at least k. So, $V(G)$ can be partitioned into sets R and B, such that for every $e \in E(G)$, it holds that $e \in M$ if and only if e has one endvertex in R and the other one in B. We define the edge set

$$M' = \{u'v' \in E(G') \mid u' \in C_u, v' \in C_v, uv \in M\}.$$

Note that $|M'| = |M| \geq k$. Moreover, M' contains no edge from any triangle C_u, so M' is a matching. For every $v \in V(G)$, we put all vertices of C_v in a set B' if $v \in B$, and else we put all vertices of C_v in a set R'. We now find that for every edge $e \in E(G')$, it holds that e belongs to M' if and only if e has one endvertex in R' and the other one in B'. Hence, M' is an edge cut, and thus a matching cut, of G' with $|M'| \geq k$.

Now suppose that G' has a matching cut M' of size at least k. Let R' and B' be the corresponding sets of red and blue vertices, respectively. We define the edge set

$$M = \{uv \in E(G) \mid u'v' \in M', u' \in C_u, v' \in C_v\}.$$

Note that $|M| = |M'| \geq k$. Due to Observation 3.1.2, every triangle C_u is monochromatic. For every $u \in V(G)$, we put u in a set R if C_u is coloured red, else we put u in a set B.

We now find for every edge $e \in E(G)$ that e belongs to M if and only if one endvertex of e belongs to R and the other one to B. Hence, M is an edge cut in G of size at least k. $\square$

The result for Maximum Disconnected Perfect Matching follows with a similar proof. Note that the maximum degree bound is no longer 3 but 6.

Theorem 5.4.2 ([51])**.** Maximum Disconnected Perfect Matching *is* NP*-hard for graphs of maximum degree* 6 *that are line graphs of triangle-free graphs.*

Proof. Let (G, k) be an instance of Maximum Cut, where G is a subcubic graph. From G, we construct a graph G' as follows. First replace every vertex $v \in V(G)$ by a clique C_v of size 6. Next, for every edge $uv \in E(G)$, add two edges between C_u and C_v, such that every vertex in C_v has at most one neighbour outside C_v. The resulting graph G' is still $(K_{1,3}, \text{diamond})$-free and thus the line graph of a triangle-free graph but has maximum degree 6.

We claim that G has an edge cut of size at least k if and only if G' has a disconnected perfect matching of size at least $2k$.

First suppose that G has an edge cut M of size at least k. So, $V(G)$ can be partitioned into sets R and B, such that for every $e \in E(G)$, it holds that $e \in M$ if and only if e has one endvertex in R and the other one in B. We define the edge set

$$M' = \left\{u'v' \in E(G') \mid u' \in C_u, v' \in C_v, uv \in M\right\}.$$

Note that $|M'| = 2 * |M| \geq 2k$. Moreover, M' contains no edge from any clique C_u, so M' is a matching. For every $v \in V(G)$, we put all vertices of C_v in a set B' if $v \in B$, and else we put all vertices of C_v in a set R'. We now find that for every edge $e \in E(G')$, it holds that e belongs to M' if and only if e has one endvertex in R' and the other one in B'. Hence, M' is an edge cut, and thus a matching cut, of G' with $|M'| \geq k$. Observe that for $u \in C_u$ with a neighbour $v \in C_v$ and $u' \in C_u$, adjacent to $v' \in C_v$, either both edges $uv, u'v'$ are in M' or none of the edges. Thus, the number of vertices in C_u not incident with an edge of M' is even and we can extend M' to a perfect matching.

Now suppose that G' has a disconnected perfect matching M' of size at least $2k$. Let R' and B' be the corresponding sets of red and blue vertices, respectively. We define the edge set

$$M = \left\{uv \in E(G) \mid u'v' \in M', u' \in C_u, v' \in C_v\right\}.$$

Note that $2|M| = |M'| \geq 2k$. Due to Observation 3.1.2, every clique C_u is monochromatic. For every $u \in V(G)$, we put u in a set R if C_u is coloured red, else we put u in a set B. We now find for every edge $e \in E(G)$ that e belongs to M if and only if one endvertex of e belongs to R and the other one to B. Hence, M is an edge cut in G of size at least k. $\square$

Chapter 6

d-Cut

In this chapter, we consider the d-Cut problem, a generalization of Matching Cut. We give polynomial-time algorithms and hardness results, which show that there exist graphs H, such that Matching Cut and d-Cut, for $d \geq 2$, have different time complexities for the class of H-free graphs. We show that d-Cut is solvable in polynomial time for graphs of diameter at most 2 and bipartite graphs of diameter at most 3. Further, we give algorithms for P_5-free and $(P_3 + P_4)$-free graphs. On the other hand, we show that d-Cut is NP-complete for graphs of high girth, $K_{1,4}$-free and H_i^*-free graphs. Moreover, it is NP-complete for $3P_3$-free graphs, for $d = 2$, and even for $3P_2$-free graphs, for $d \geq 3$. We start with a generalization of valid red-blue colourings.

6.1 Propagation of Partial Colourings

We generalize the colouring terminology from Chapter 3. Recall that a *red-blue colouring* of a graph G assigns every vertex of G either the colour red or blue. For $d \geq 1$, a red-blue colouring is a *red-blue d-colouring* if every blue vertex has at most d red neighbours; every red vertex has at most d blue neighbours; and both colours red and blue are used at least once. See Figure 6.1 for examples of a red-blue 1-colouring and a red-blue 3-colouring. We make the following observation, which is similar to Observation 3.1.1.

Observation 6.1.1. *For every $d \geq 1$, a connected graph G has a d-cut if and only if it has a red-blue d-colouring.*

Note that a red-blue 1-colouring is a valid red-blue colouring. Recall that a set where all vertices have the same colour is called *monochromatic* and that an edge with a blue and a red endvertex is *bichromatic*. The next observation resembles Observation 3.1.2 for valid red-blue colourings.

Observation 6.1.2. *For every $d \geq 1$, every complete graph K_r with $r \geq 2d+1$ and every complete bipartite graph $K_{r,s}$ with* $\min\{r, s\} \geq 2d$ *and* $\max\{r, s\} \geq 2d+1$ *is monochromatic in every red-blue d-colouring.*

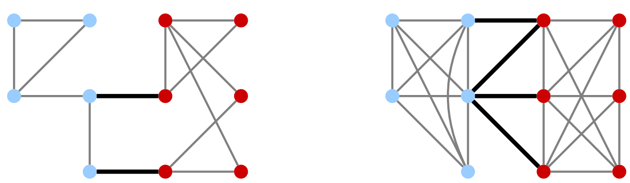


Figure 6.1: An example for a graph with a 1-cut, that is a matching cut, (left) and a 3-cut (right).

Observation 6.1.3. *For every $d \geq 1$, every connected graph with a red-blue d-colouring contains a bichromatic edge.*

Similar to our polynomial-time algorithms for Matching Cut, our polynomial-time algorithms for solving d-Cut on special graph classes are based on guessing a small set S of red vertices and a small set T of blue vertices in the input graph G. The goal is to extend S and T to sets R and B of red and blue vertices, respectively, such that $R \cup B = V(G)$. However, for d-Cut with $d \geq 2$, many arguments that were valid for $d = 1$ cannot be used, and different arguments will be needed.

Formally, let $d \geq 1$. Let $G = (V, E)$ be a connected graph and $S, T \subseteq V$ be two disjoint sets. A *red-blue (S,T)-d-colouring* of G is a red-blue d-colouring of G that colours all the vertices of S red and all the vertices of T blue. We call (S,T) a *precoloured pair* of (G,d) which is *colour-processed* if every vertex of $V \setminus (S \cup T)$ is adjacent to at most d vertices of S and to at most d vertices of T. The next lemma shows that we may assume without loss of generality that a precoloured pair (S,T) is always colour-processed. Recall that we say that a propagation rule is *safe* if a graph G has a red-blue (S,T)-d-colouring before the application of the rule if and only if it has a red-blue (S,T)-d-colouring after the application of the rule.

Lemma 6.1.4 ([48])**.** *Let G be a connected graph with a precoloured pair (S,T). It is possible, in polynomial time, to either colour-process (S,T) or to find that G has no red-blue (S,T)-d-colouring.*

Proof. We apply the following rules on G. Let $Z = V \setminus (S \cup T)$. If $v \in Z$ is adjacent to $d+1$ vertices in S, then move v from Z to S. If $v \in Z$ is adjacent to $d+1$ vertices in T, then move v from Z to T. Return `no` if a vertex $v \in Z$ at some point becomes adjacent to $d+1$ vertices in S as well as to $d+1$ vertices in T. We apply these three rules exhaustively. When at some point no rule can be applied, we stop.

It is readily seen that each of these rules is safe, and moreover, can be verified and applied in polynomial time. After each application of a rule, we either stop or have decreased the size of Z by at least one vertex. Hence, the procedure is correct and takes polynomial time. □

We now generalize Lemma 3.1.3, which shows that we can decide in polynomial time if a graph with a small dominating set has a d-cut.

Lemma 6.1.5 ([48])**.** *For $d, g \geq 1$, it is possible to find in $O(2^g n^{d+2})$-time a red-blue d-colouring (if it exists) of a graph G with n vertices and domination number g.*

Proof. Let $d, g \geq 1$ and G be a graph on n vertices with domination number g. Let D be a dominating set D of G that has size at most g.

We consider all $2^{|D|} \leq 2^g$ options of giving the uncoloured vertices of D either colour red or blue. For each red-blue colouring of D we do as follows. For every red vertex of D, we consider all $O(n^d)$ options of colouring at most d of its uncoloured neighbours blue, and we colour all of its other uncoloured neighbours red. Similarly, for every blue vertex of D, we consider all $O(n^d)$ options of colouring at most d of its uncoloured neighbours red, and we colour all of its other uncoloured neighbours blue. As D dominates G, we obtained a red-blue colouring c of the whole graph G. We discard the option if c is not a red-blue d-colouring of G.

We note that any red-blue d-colouring of G, if it exists, will be found by the above algorithm. As the total number of options is $O(2^g n^d)$ and checking if a red-blue colouring is a red-blue d-colouring takes $O(n^2)$ time, our algorithm has total running time $O(2^g n^{d+2})$. □

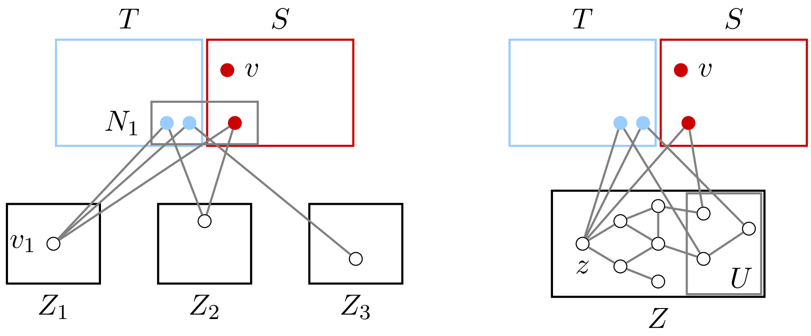


Figure 6.2: Cases 1 (left) and 2 (right) in the proof of Theorem 6.2.1.

6.2 Polynomial-Time Results

In the following, we give the polynomial-time algorithms for d-Cut. We start with graphs of bounded diameter. Recall that in Question 4, Gomes and Sau asked for the complexity of d-Cut for graphs of diameter at most 2. We answer this question in Theorem 6.2.1.

6.2.1 Graphs with Bounded Diameter

We show that d-Cut is solvable in polynomial time for graphs of diameter at most 2 (Theorem 6.2.1) and bipartite graphs of diameter at most 3 (Theorem 6.2.2). Both proofs use the same idea. We first guess a pair (S, T) where we colour S red and T blue. Then, we consider the connected components of the uncoloured part of the graph. We will see that if we have more than one of these components, we can colour the remaining part of the graph. Otherwise we use again that the graph has bounded diameter to colour a small vertex set which, after some preprocessing, dominates the remaining graph. This procedure will be repeated a constant number of times, since afterwards every uncoloured vertex has more than d coloured neighbours of at least one colour and will be coloured by propagation.

For $d = 1$ this result is already known from [7, 39] and also Theorem 4.1.1 and 4.2.4, which both show more general results. Note that these four proofs are all different. The following proof provides again a new way of how to show this result and also works for $d \geq 2$.

Theorem 6.2.1 ([48])**.** *For $d \geq 1$, d-Cut is polynomial-time solvable for graphs of diameter at most 2.*

Proof. Let $G = (V, E)$ be a graph of diameter at most 2. Note that this implies that G is connected. Let $v \in V$. Without loss of generality we may colour v red. Since v has at most d blue neighbours in any red-blue d-colouring, we can branch over all $O(n^d)$ options to colour the neighbourhood of v. We consider each option separately.

Let R and B be the red and the blue set, respectively. We define a precoloured pair (S, T), with $S' = \{v\} \cup \{u \mid u \in N(v), u \in R\}$ and $T' = \{u \mid u \in N(v), u \in B\}$. We use Lemma 6.1.4 to colour-process (S', T') in polynomial time, resulting in a pair (S, T). Let $Z = V \setminus (S \cup T)$ be the set of uncoloured vertices. As (S, T) is colour-processed, every vertex in Z has at most d red and d blue neighbours and thus in total at most $2d$ coloured neighbours. We distinguish between the following two cases:

Case 1. $G[Z]$ consists of at least two connected components.
Let $Z_1, \ldots, Z_r$ be the connected components of $G[Z]$. Let $v_1 \in V(Z_1)$ and $v_2 \in V(Z_2)$. Since G has diameter 2, we know that v_1 has a common neighbour with every vertex in $Z_2, \ldots, Z_r$. Let N_1 be the set of common neighbours of v_1 and $Z_2, \ldots, Z_r$, see Figure 6.2

(left). Note that $N_1 \subseteq S \cup T$. Thus, $|N_1| \leq 2d$, since v_1 has at most $2d$ coloured neighbours. Using the same arguments, we can find a set $N_2 \subseteq S \cup T$, with $|N_2| \leq 2d$ containing the common neighbours of v_2 and Z_1. In a red-blue d-colouring, every vertex in $N_1 \cup N_2$ has at most d neighbours of the other colour. Thus, we can try all $O(n^{4d^2})$ options to colour $N_{G[Z]}(N_1 \cup N_2) = Z$. If in the resulting colouring any vertex has more than d neighbours of the other colour we discard the branch, otherwise we found a red-blue d-colouring and thus a d-cut.

Case 2. $G[Z]$ is connected.
We first assume that $\text{radius}(G[Z]) > 2$. We let $z \in Z$, and we let $U = \{u \in Z : \text{dist}_{G[Z]}(z, u) > 2\}$, see Figure 6.2 (right). Note that $U \neq \emptyset$. As G has diameter 2, we find that z has a common neighbour with every vertex in U. These common neighbours are not contained in Z and thus they are in $S \cup T$, the set of coloured vertices. Vertex z has at most $2d$ coloured neighbours. Each of them has at most d neighbours of the other colour. So we can branch over all $O(n^{2d^2})$ options to colour U. If any colouring of U leads to a vertex having more than d neighbours of the other colour, we discard the branch. For each remaining colouring, we know that the graph $G[Z \setminus U]$ which remains uncoloured has radius at most 2. Let $Z = Z \setminus U$.

So we may assume (possibly after some branching) that $\text{radius}(G[Z]) \leq 2$. Let $z \in Z$ such that $\text{dist}_{G[Z]}(z, u) \leq 2$ for all $u \in Z$. We branch over all $O(n^d)$ colourings of $\{z\} \cup N_{G[Z]}(z)$. Since $\text{radius}(G[Z]) \leq 2$, we get that $\{z\} \cup N_{G[Z]}(z)$ is a dominating set of $G[Z]$. Hence, every uncoloured vertex got a new coloured neighbour. We colour-process the pair of red and blue sets, which takes polynomial time by Lemma 6.1.4. This results in a new uncoloured set of vertices Z'. If there is a vertex with $d+1$ neighbours of the other colour or an uncoloured vertex with more than d neighbours of each colour we discard the branch. Otherwise we can proceed by choosing either Case 1 or Case 2.

If at some point we apply Case 1, we are done. Every time we apply Case 2, all vertices that remain uncoloured get a new coloured neighbour. Hence, we can apply Case 2 at most $2d$ times, since afterwards every uncoloured vertex has more than $2d$ coloured neighbours and thus, they are coloured when we colour-process the sets of red and blue vertices.

The correctness of our algorithm follows from its description. We now discuss its running time. We always have $O(n^d)$ branches in the beginning. Case 1 give $O(n^{4d^2})$ branches, while Case 2 only gives $O(n^{2d^2+d})$ branches. Since after one application of Case 1 we are done, in the worst case we apply Case 2 $2d-1$ times and then apply Case 1. This leads to

$$O\left(n^d * \left(n^{2d^2+d}\right)^{2d-1} * n^{4d^2}\right) = O\left(n^{4d^2(d+1)}\right)$$

branches, a polynomial number, to consider in the worst case. Since by Lemma 6.1.4 the colour-processing can be done in polynomial time, the algorithm runs in polynomial time. □

For the next result we apply Observation 2.1.1 on the overlapping neighbourhoods in bipartite graphs of diameter 3. We generalize the result from Le and Le [39] that MATCHING CUT is solvable in polynomial time for bipartite graphs of diameter 3. Our proof works for all $d \geq 1$ and is different from the proof of [39]. It uses ideas from the proof of Theorem 6.2.1.

Theorem 6.2.2. *For $d \geq 1$, d-CUT is polynomial-time solvable for bipartite graphs of diameter at most 3.*

Proof. Let $G = (V, E)$ be a bipartite graph of diameter at most 3. Note that this implies that G is connected. Let $v \in V$. Without loss of generality, we may colour v red. Since v has at most d blue neighbours in any red-blue d-colouring, we can branch over all $O(n^d)$ options to colour the neighbourhood of v. We consider each option separately.

Let R and B be the red and the blue set, respectively. We define a precoloured pair (S', T'), where $S' = \{v\} \cup \{u \mid u \in N(v), u \in R\}$ and $T' = \{u \mid u \in N(v), u \in B\}$. We use Lemma 6.1.4 to colour-process (S', T') in polynomial time, resulting in a pair (S, T). Let $Z = V \setminus (S \cup T)$ be the set of uncoloured vertices. As (S, T) is colour-processed, every vertex in Z has at most d red and d blue neighbours and thus in total at most $2d$ coloured neighbours. Note further, that $G[Z]$ and every connected component of $G[Z]$ is bipartite. We distinguish between the following cases:

Case 1. $G[Z]$ consists of at least two connected components.
Let Z^1 and Z^2 be the partition classes of $G[Z]$. Let $Z_1, \ldots, Z_r$ be the connected components of $G[Z]$. We denote by Z_i^1 and Z_i^2 the partition classes of the bipartite component Z_i, for $i \in \{1, \ldots, r\}$. Assume first that each of Z^1 and Z^2 consists of a single vertex. Then, we branch over all colourings of these two vertices.

Assume now that one of Z^1 and Z^2, say Z^1, consists of a single vertex. Then, we branch over all possible colourings of Z^1. Since $G[Z]$ consists of at least two connected components we can choose $v_1 \in Z_1^2$ and $v_2 \in Z_2^2$. Since G is bipartite and has diameter 3, it follows from Observation 2.1.1 that v_1 has a common neighbour with every vertex in $Z_2^2, \ldots, Z_r^2$. Let N_1 be the set of common neighbours of v_1 and $Z_2^2, \ldots, Z_r^2$. Then, $N_1 \subseteq S \cup T$ and thus, $|N_1| \leq 2d$, since v_1 has at most $2d$ coloured neighbours. Similarly, let N_2 be the set of common neighbours of v_2 and Z_1^2. Then, $|N_2| \leq 2d$ as well. Thus, we can try all $O(n^{4d^2})$ options to colour $N_{G[Z^2]}(N_1 \cup N_2) = Z^2$.

Consider next the case where all vertices in one partition class of $G[Z]$, say Z^1 are contained in a single component, say Z_1. Hence, the components $Z_2, \ldots, Z_r$ only contain vertices in Z^2 and thus, since G is bipartite, each component except of Z_1 consists of exactly one vertex, which is contained in Z^2. We pick a vertex $v \in Z_1^2$. Let $N_1 \subseteq S \cup T$ be the set of common neighbours of v and the vertices in $Z_2, \ldots, Z_r$. Using the same arguments as before, we can colour all vertices in $Z_2, \ldots, Z_r$. Then, only the component Z_1 remains uncoloured and we can proceed as in Case 2.

If none of the aforementioned cases occurred, we know that we can find two vertices per partition class which lie in different components of $G[Z]$. Let $v_1 \in V(Z_1^1)$, $v_2 \in V(Z_1^2)$, $u_1 \in V(Z_2^1)$ and $u_2 \in V(Z_2^2)$. If Z_1 or Z_2 does not contain enough vertices to choose v_1, v_2, u_1 and u_2, for the missing one, we pick a vertex from another component which is in the same partition class as the missing one.

We apply the same method as before to colour all components. Since G is bipartite and has diameter 3, we know by Observation 2.1.1 that v_1 has a common neighbour with every vertex in $Z_2^1, \ldots, Z_r^1$. Similarly, v_2 has a common neighbour with every vertex in $Z_2^2, \ldots, Z_r^2$. Let N_1 be the set of common neighbours of v_1 and $Z_2^1, \ldots, Z_r^1$ and let N_2 be the set of common neighbours of v_2 and $Z_2^2, \ldots, Z_r^2$. Note that $N_1 \subseteq S \cup T$ and $N_2 \subseteq S \cup T$. Thus, $|N_1| \leq 2d$, since v_1 has at most $2d$ coloured neighbours. Similarly, $|N_2| \leq 2d$. Using the same arguments, we can find sets N_3 and N_4 with $|N_3| \leq 2d$ and $|N_4| \leq 2d$ containing the common neighbours of u_1 and Z_1^1 and of u_2 and Z_1^2. In a red-blue d-colouring, every vertex in $N_1 \cup N_2 \cup N_3 \cup N_4$ has at most d neighbours of the other colour. Thus, we can try all $O(n^{8d^2})$ options to colour $N_{G[Z]}(N_1 \cup N_2 \cup N_3 \cup N_4)$. Note that $N_{G[Z]}(N_1 \cup N_2 \cup N_3 \cup N_4) = Z$. If in the resulting colouring any vertex has more than d neighbours of the other colour we discard it, otherwise we found a red-blue d-colouring and thus a d-cut.

Case 2. $G[Z]$ is connected.
Let Z^1 and Z^2 be the partition classes of $G[Z]$. We first assume that there do not exist vertices $u \in Z^1$ and $v \in Z^2$ such that $\text{dist}_{G[Z]}(u, z) \leq 2$ for all $z \in Z^1$ and $\text{dist}_{G[Z]}(v, z) \leq 2$ for all $z \in Z^2$. Let $u \in Z^1$, and we let $U^1 = \{z \in Z^1 : \text{dist}_{G[Z]}(u, z) > 2\}$. Note that $U^1 \neq \emptyset$. By Observation 2.1.1 we find that u has a common neighbour with every vertex

in U^1. These common neighbours are not contained in Z and thus they are in $S \cup T$, the set of coloured vertices. Since it is uncoloured, the vertex u has at most $2d$ coloured neighbours. Each of them has at most d neighbours of the other colour. So we can branch over all $O(n^{2d^2})$ options to colour U^1. If any colouring of U^1 leads to a vertex having more than d neighbours of the other colour, we discard the branch. We repeat this procedure for a vertex $v \in Z^2$ with the set $U^2 = \{z \in Z^2 : \mathrm{dist}_{G[Z]}(u, z) > 2\}$. Thus, U_1 and U_2 are both coloured now. The graph $G[Z \setminus (U^1 \cup U^2)]$ remains uncoloured. Let $Z = Z \setminus (U^1 \cup U^2)$, that is we update Z such that it contains only uncoloured vertices.

So we may assume (possibly after some branching) that there exist vertices $u \in Z^1$ and $v \in Z^2$ such that $\mathrm{dist}_{G[Z]}(u, z) \leq 2$ for all $z \in Z^1$ and $\mathrm{dist}_{G[Z]}(v, z) \leq 2$ for all $z \in Z^2$. We branch over all $O(n^{2d})$ colourings of $\{u\} \cup N_{G[Z]}(u)$ and $\{v\} \cup N_{G[Z]}(v)$. Due to our assumption on u and v, we get that $\{u, v\} \cup N_{G[Z]}(u) \cup N_{G[Z]}(v)$ is a dominating set of $G[Z]$. Hence, every uncoloured vertex got a new coloured neighbour. We colour-process the pair of red and blue sets, which takes polynomial time by Lemma 6.1.4. This results in a new uncoloured set of vertices Z'. If there is a vertex with $d+1$ neighbours of the other colour or an uncoloured vertex with more than d neighbours of each colour we discard the branch. Otherwise we can proceed by choosing either Case 1 or Case 2.

If at some point we apply Case 1, we are either done or proceed with Case 2. Every time we apply Case 2, all vertices that remain uncoloured get a new coloured neighbour. Hence, we can apply Case 1 and Case 2 each at most $2d$ times, since afterwards every uncoloured vertex has more than $2d$ coloured neighbours and thus, they are coloured when we colour-process the sets of red and blue vertices.

The correctness of our algorithm follows from its description. We now discuss its running time. We always have $O(n^d)$ branches in the beginning. Case 1 gives at most $O(n^{8d^2})$ branches, while Case 2 gives $O(n^{4d^2+2d})$ branches. In the worst case we apply Case 1 and Case 2 $2d$ times. This leads to

$$O\left(n^d \cdot \left(n^{8d^2}\right)^{2d} \cdot \left(n^{4d^2+2d}\right)^{2d}\right) = O\left(n^{24d^3+4d^2+d}\right)$$

branches, a polynomial number, to consider in the worst case. Since by Lemma 6.1.4 the colour-processing can be done in polynomial time, the algorithm runs in polynomial time. □

Remark 6.2.3. Note that, with minor modifications, this proof also works for Maximum Matching Cut and Maximum Disconnected Perfect Matching, see Theorem 4.2.6 and 4.2.14.

6.2.2 Graphs without Induced Paths

We now show that d-Cut is solvable in polynomial time for P_5-free graphs and $(P_3 + P_4)$-free graphs. First, we show that if d-Cut is solvable in polynomial time for H-free graphs, then this also holds for $(H + P_1)$-free graphs. Note that in Theorem 6.3.6 we show that d-Cut is NP-hard for $3P_2$-free graphs, for $d \geq 3$. Thus, this result is best possible.

Theorem 6.2.4 ([48]). *For every graph H and every $d \geq 2$, if d-Cut is polynomial-time solvable for H-free graphs, then it is so for $(H + P_1)$-free graphs.*

Proof. Suppose that d-Cut is polynomial-time solvable for H-free graphs. Let G be a connected $(H + P_1)$-free graph. If G is H-free, the result follows by assumption. If G is not H-free, then $V(G)$ contains a set U such that $G[U]$ is isomorphic to H. As G is $(H + P_1)$-free, U dominates G. As H is fixed, $|U| = |V(H)|$ is a constant. Hence, we can apply Lemma 6.1.5 and the theorem follows. □

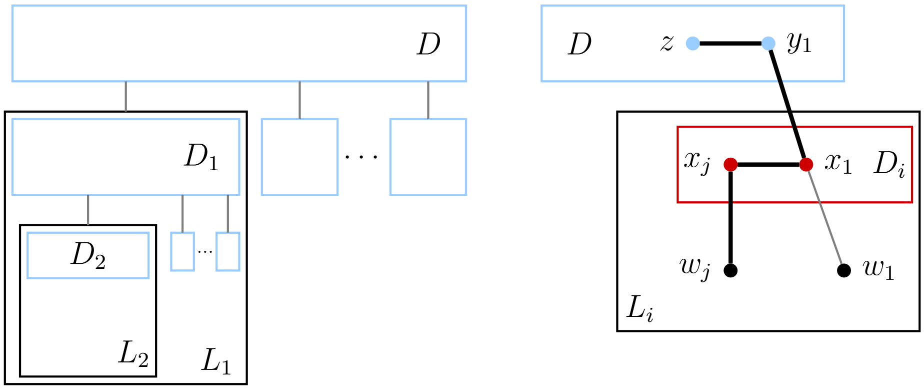


Figure 6.3: The graph G in the blue phase (left) and in the red phase: Case 2 (right).

The next result is on P_5-free graphs. For $d = 1$, the same approach is used for P_5-free graphs [20]. We find a dominating set D and branch over all colourings of D. The difference is that for $d = 1$, the algorithm and analysis is much shorter: one only has to check, if there is a component of $G - D$, whose vertices can all be safely coloured red. Then one either finds a matching cut, or G has no matching cut.

Theorem 6.2.5 ([48])**.** *For $d \geq 2$, d-*Cut *is polynomial-time solvable for P_5-free graphs.*

Proof. Let $d \geq 2$. Let $G = (V, E)$ be a connected P_5-free graph on n vertices. As G is P_5-free and connected, G has a dominating set D, such that either D induces a cycle on five vertices or D is a clique [45]. Moreover, we find such a dominating set D in $O(n^3)$ time 2.1.7. If $|D| \leq 3d$, then we apply Lemma 6.1.5. Now assume that $|D| \geq 3d + 1 \geq 7$, and thus D is a clique. If $D = V$, then G has no d-cut, as $|D| \geq 3d + 1$ and D is a clique. Assume that $D \subsetneq V$, so $G - D$ has at least one connected component. Below we explain how to find in polynomial time a d-cut of G, or to conclude that G does not have a d-cut. By Observation 6.1.1, we need to decide in polynomial time if G has a red-blue d-colouring.

We first enter the *blue phase* of our algorithm. As $|D| \geq 3d + 1$ and D is a clique, D is monochromatic in any red-blue d-colouring of G. Hence, we may colour, without loss of generality, all the vertices of D blue, and we may colour all vertices of every connected component of $G - D$ except one connected component blue as well. We branch over all $O(n)$ options of choosing the connected component L_1 of $G - D$ that will contain a red vertex. For each option, we are going to repeat the process of colouring vertices blue until we colour at least one vertex red. We first explain how this process works if we do not do this last step.

As L_1 is connected and P_5-free, we find in $O(n^3)$ time (using the algorithm of 2.1.7) a dominating set D_1 of L_1 that is either a cycle on five vertices or a clique. We colour the vertices of D_1 blue. As we have not used the colour red yet, we colour all vertices of every connected component of $L_1 - D_1$ except one connected component blue. So, we branch over all $O(n)$ options of choosing the connected component L_2 of $L_1 - D_1$ that will contain a red vertex. Note that every uncoloured vertex of G, which belongs to L_2, has both a neighbour in D (as D dominates G) and a neighbour in D_1 (as D_1 dominates L_1). We now find a dominating set D_2 of L_2 and a connected component L_3 of $G - D_2$ with a dominating set D_3, and so on. See also Figure 6.3.

If we repeat the above process more than d times, we have either coloured every vertex of G blue, or we found $d + 1$ pairwise disjoint, blue sets $D, D_1, \ldots, D_d$, such that every uncoloured vertex u has a neighbour in each of them. The latter implies that u has $d + 1$ blue neighbours, so u must be coloured blue as well. Hence, in each branch, we would eventually end up with the situation where all vertices of G will be coloured blue. To prevent this from happening, we must colour, in each branch, at least one vertex of D_i red, for some $1 \le i \le d-1$. As soon as we do this, we end the blue phase for the branch under consideration, and our algorithm enters the *red phase*.

Each time we have $O(n)$ options to select a connected component L_i and, as argued above, we do this at most d times. Hence, we enter the red phase for $O(n^d)$ branches in total. From now on, we call these branches the *main branches* of our algorithm. For a main branch, we say that we *quit the blue phase at level* i if we colour at least one vertex of D_i red. If we quit the blue phase for a main branch at level i, for some $1 \le i \le d-1$, then we have constructed, in polynomial time, pairwise disjoint sets D, $D_1, \ldots, D_i$ and graphs $L_1, \ldots, L_i$, such that:

- for every $h \in \{1, \ldots, i-1\}$, L_{h+1} is a connected component of $L_h - D_h$;
- every vertex of G that does not belong to L_i has been coloured blue;
- for every $h \in \{1, \ldots, i\}$, D_h induces a cycle on five vertices or is a clique;
- D dominates G, so, in particular, D dominates L_i; and
- for every $h \in \{1, \ldots, i\}$, D_h dominates L_h.

We now prove the following claim, which shows that we branched correctly.

Claim 6.2.5.1. *The graph G has a red-blue d-colouring that colours every vertex of D without loss of generality blue, if and only if we have quit a main branch at level i for some $1 \le i \le d-1$, such that G has a red-blue d-colouring that colours at least one vertex of D_i red and all vertices not in L_i blue.*

Proof. Suppose G has a red-blue d-colouring c that colours every vertex of D blue (the reverse implication is immediate). By definition, c has coloured at least one vertex u in $G - D$ red. As we branched in every possible way, there is a main branch that quits the blue phase at level i, such that u belongs to L_i for some $1 \le i \le d-1$. We pick a main branch with largest possible i, so u is not in L_{i+1}. Hence, $u \in D_i$. We also assume that no vertex u' that belongs to some D_h with $h < i$ is coloured red by c, as else we could take u' instead of u. Hence, c has coloured every vertex in $D_1 \cup \ldots \cup D_{i-1}$ (if $i \ge 2$) blue. Therefore, we may assume that every vertex $v \notin D \cup D_1 \cup \ldots \cup D_{i-1} \cup V(L_i)$ has been coloured blue by c. If not, may just recolour all such vertices v blue for the following reasons: u is still red and no neighbour of v is red, as after a possible recolouring all red vertices belong to L_i, while v belongs to a different component of $G-(D \cup D_1 \cup \cdots \cup D_{i-1})$ than L_i. So, we proved the claim. ◁

Claim 6.2.5.1 allows us to do some specific branching once we quit the blue phase for a certain main branch at level i. Namely, all we have to do is to consider all options to colour at least one vertex of D_i red. We call these additional branches *side branches*. We distinguish between the following two cases:

Case 1. $|D_i| \le 2d+1$.
We consider each of the at most 2^{2d+1} options to colour the vertices of D_i either red or blue, such that at least one vertex of D_i is coloured red. Next, for each vertex $u \in D_i$, we consider all $O(n^d)$ options to colour at most d of its uncoloured neighbours blue if u is red, or red if u is blue. Note that the total number of side branches is $O(2^{2d+1}n^{d(2d+1)})$. As D_i dominates L_i and the only uncoloured vertices were in L_i, we obtained a red-blue colouring c of G. We check in polynomial time if c is a red-blue d-colouring of G. If so, we stop and return c. If none of the side branches yields a red-blue d-colouring of G, then by Claim 6.2.5.1 we can safely discard the main branch under consideration.

Case 2. $|D_i| \ge 2d+2$.
Recall that D_i either induces a cycle on five vertices or is a clique. As $|D_i| \ge 2d+2 \ge 6$, we find that D_i is a clique. As $|D_i| \ge 2d+2$, this means that D_i must be monochromatic,

and thus every vertex of D_i must be coloured red. We check in polynomial time if D_i contains a vertex with more than d neighbours in D (which are all coloured blue), or if D contains a vertex with more than d neighbours in D_i (which are all coloured red). If so, we may safely discard the main branch under consideration due to Claim 6.2.5.1.

From now on, assume that every vertex in D_i has at most d neighbours in D, and vice versa. By construction, every uncoloured vertex belongs to $L_i - D_i$. Hence, if $V(L_i) = D_i$, we have obtained a red-blue colouring c of G. We check, in polynomial time, if c is also a red-blue d-colouring of G. If so, we stop and return c. Otherwise, we may safely discard the main branch due to Claim 6.2.5.1.

Now assume $V(L_i) \supsetneq D_i$. We colour all vertices in $L_i - D_i$ that are adjacent to at least $d+1$ vertices in D blue; we have no choice as the vertices in D are all coloured blue. If there are no uncoloured vertices left, we check in polynomial time if the obtained red-blue colouring is a red-blue d-colouring of G. If so, we stop and return it; else we may safely discard the main branch due to Claim 6.2.5.1.

Assume that we still have uncoloured vertices left. We recall that these vertices belong to $L_i - D_i$, and that by construction they have at most d neighbours in D. Consider an uncoloured vertex w_1. As D_i dominates L_i, we find that w_1 has a neighbour x_1 in D_i. As D dominates G, we find that x_1 is adjacent to some vertex $y_1 \in D$. We consider all $O(n^{2d})$ possible ways to colour the uncoloured neighbours of x_1 and y_1, such that x_1 (which is red) has at most d blue neighbours, and y_1 (which is blue) has at most d red neighbours. If afterwards there is still an uncoloured vertex w_2, then we repeat this process: we choose a neighbour x_2 of w_2 in D_i and a neighbour y_2 of w_2 in D, and we branch again by colouring the uncoloured neighbours of x_2 and y_2, and so on. We repeat this process until there are no more uncoloured vertices. This gives us $2p$ distinct vertices $w_1, \ldots, w_p$, $x_1, \ldots, x_p$, together with vertices $y_1, \ldots, y_p$ (with possibly $y_a = y_b$ for some $1 \leq a < b \leq p$) for some integer $p \geq 1$. The total number of side branches for the main branch is $O(n^{2pd})$.

We claim that $p \leq d$. For a contradiction, assume that $p \geq d+1 \geq 3$. Let $2 \leq j \leq p$. As D_i is a clique that contains x_1 and x_j, we find that x_1 and x_j are adjacent. Hence, G contains the 4-vertex path $w_j x_j x_1 y_1$. By construction, w_j was uncoloured after colouring the neighbours of x_1 and y_1, so w_j is neither adjacent to x_1 nor to y_1. This means that $w_j x_j x_1 y_1$ is an induced P_4 if and only if x_j is not adjacent to y_1. Recall that w_j and every vertex of D_i, so including x_1 and x_j, has at most d neighbours in D. As $|D| \geq 3d+1$, this means that D contains a vertex z that is not adjacent to any of w_j, x_j, x_1. As D is a clique, z is adjacent to y_1. Hence, x_j must be adjacent to y_1, as otherwise $w_j x_j x_1 y_1 z$ is an induced P_5; see also Figure 6.3. The latter is not possible as G is P_5-free. We now find that y_1 is adjacent to $x_1, \ldots, x_p$, which all belong to D_i. As we assumed that $p \geq d+1$ and every vertex of D, including y_1, is adjacent to at most d vertices of D_i, this yields a contradiction. We conclude that $p \leq d$.

As $p \leq d$, the total number of side branches is $O(n^{2d^2})$. Each side branch yields a red-blue colouring of G. We check in polynomial time if it is a red-blue d-colouring of G. If so, then we stop and return it; else we can safely discard the side branch, and eventually the associated main branch, due to Claim 6.2.5.1.

The correctness of our algorithm follows from its description. With regard to its running time, the total number of main branches is $O(n^d)$. For each main branch we have either $O(2^{2d+1} n^{d(2d+1)})$ side branches (Case 1) or $O(n^{2d^2})$ side branches (Case 2). Hence, the total number of branches is $O(n^d 2^{2d+1} n^{d(2d+1)})$. As d is fixed, this is a polynomial number. As processing each branch takes polynomial time (including the construction of the D_i-sets and L_i-graphs), we conclude that our algorithm runs in polynomial time. This completes the proof. $\square$

Remark 6.2.6. We expect that this result can be generalized to P_6-free graphs ([58]).

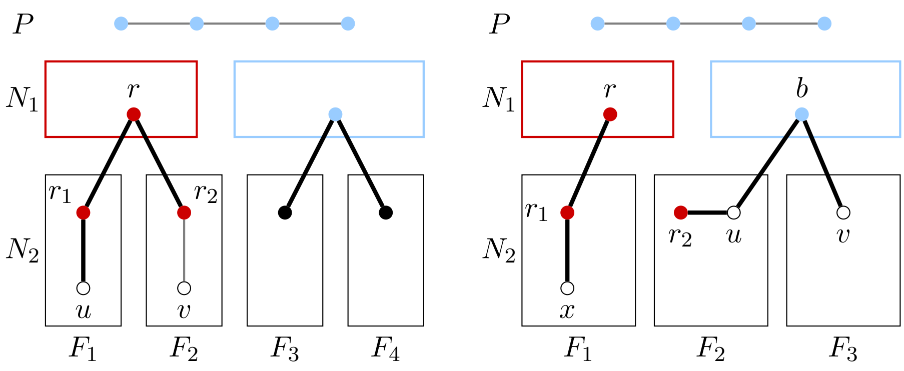


Figure 6.4: Illustration of the proof of Theorem 6.2.7. The white vertices indicate uncoloured vertices, while for the black vertices we do not know if they are uncoloured or which colour they have.

We now show that d-Cut, for $d \geq 2$, is solvable in polynomial time for $(P_3 + P_4)$-free graphs. This proof also works for $d = 1$. Recall that 1-Cut is the same problem as Matching Cut. Combining Theorem 4.1.2 and 4.1.3, we get the stronger result that Matching Cut is solvable in polynomial time for $(sP_3 + P_6)$-free graphs. Let $G = (V, E)$ be a graph and let $S \subseteq V$. Recall that we differ between the neighbourhood $N(S)$ and the closed neighbourhood $N[S] = N(S) \cup S$.

Theorem 6.2.7 ([48])**.** *For every $d \geq 2$, d-Cut is polynomial-time solvable for $(P_3 + P_4)$-free graphs.*

Proof. Let G be a connected $(P_3 + P_4)$-free graph. We may assume that G contains an induced P_4, else we apply Theorem 6.2.5.

We divide the proof into two cases. First, we decide whether there exists a red-blue d-colouring of G in which some induced P_4 is monochromatic. We then search for a red-blue d-colouring in which every induced P_4 is bichromatic.

Case 1. There exists a monochromatic P_4 in some red-blue d-colouring of G.
We branch over all induced P_4s of G. Note that there are at most $O(n^4)$ induced P_4s in G. Let P be the current induced P_4 of G which we assume is monochromatic. Without loss of generality, every vertex of P is blue. Let $N_1 = N(P)$ and $N_2 = V(G) \setminus N[P]$. Since G is $(P_3 + P_4)$-free we get that $G[N_2]$ is P_3-free and thus every connected component of $G[N_2]$ is a clique. We branch over all $O(n^{4d})$ possible colourings of N_1. If $V(P) \cup N_1$ is monochromatic, that is N_1 is blue, then it is enough to decide whether there is a connected component F of $G[N_2]$ such that $G[V(P) \cup N_1 \cup V(F)]$ has a red-blue d-colouring. All other components can safely be coloured blue. Since F is a clique, F is monochromatic if $|V(F)| \geq 2d+1$ and has constant size otherwise. In both cases only a constant number of possible colourings exist for F and we can try each of them to decide if it gives a red-blue d-colouring of $G[V(P) \cup N_1 \cup V(F)]$. If we find a red-blue d-colouring we are done, otherwise we discard the branch and continue with the case where $V(P) \cup N_1$ is not monochromatic.

Let S be the set of red vertices in N_1. Since we handled the case where all vertices in N_1 are blue, we may assume that S is not empty. We then branch over all $O(n^{4d^2})$ possible colourings of $N(S)$. Note that every vertex which remains uncoloured belongs to N_2 and that all its neighbours in N_1 are blue. Recall that each connected component of $G[N_2]$ is a clique. Therefore each connected component either has size at most $2d$ or is monochromatic. This implies that there are a constant number of possible red-blue colourings for each component. We colour-process the current red-blue colouring.

Consider a connected component of $G[N_2]$ which contains an uncoloured vertex. We call this component F_1. If F_1 contains no red vertex with a red neighbour in N_1 then all vertices in F_1 have only blue neighbours in N_1, since the neighbourhood of S is coloured. Thus, we may recolour the whole connected component blue.

We now consider the case that there exists a red vertex $r \in N_1$ which has red neighbours in two different connected components of N_2 which contain uncoloured vertices. Let F_1 and F_2 be these connected components and $r_1 \in V(F_1)$ the red neighbour of r in F_1 and $u \in V(F_1)$ an uncoloured vertex in F_1. Similarly let $r_2 \in V(F_2)$ be a red neighbour of r in F_2 and $v \in V(F_2)$ an uncoloured vertex in F_2. Note that neither u nor v is a neighbour of r since $r \in S$ and $N(S)$ is already coloured. Then $Q = u - r_1 - r - r_2$ is an induced P_4. By $(P_3 + P_4)$-freeness, any induced P_3 in G contains a vertex which is adjacent to at least one vertex of Q. In particular, this implies hat every blue vertex in N_1 with uncoloured neighbours in more than one connected component of $G[N_2]$ which is neither F_1, nor F_2 has a neighbour on Q. See Figure 6.4 (left). Let T be the set of these vertices, that is the set of blue vertices in N_1 with uncoloured neighbours in at least two connected components, neither of which is F_1 or F_2. Since the vertices of Q are either red or uncoloured, they may have at most d blue neighbours and thus $|T| \leq 4d$.

We branch over all $O(n^{4d^2})$ colourings of $N(T) \cup V(F_1) \cup V(F_2)$. Let $F_3, \dots, F_k$ be the connected components of $G[N_2]$ which contain an uncoloured vertex. Let $x \in V(F_i)$, for $i \in \{3, \dots, k\}$, be a vertex that remains uncoloured. The neighbours of x in N_1 have uncoloured neighbours only in F_i.

Therefore, it is enough to decide whether we can extend the current red-blue colouring to a red-blue d-colouring of $G[V(F_i) \cup N_1]$ for each connected component F_i of $G[N_2]$ with $i \geq 3$. To see that these colourings are independent, recall that any remaining vertex of N_1 with an uncoloured neighbour has uncoloured neighbours in only one component F_i. That is, we branch over each of the constant number of possible red-blue colourings of the uncoloured vertices of F_i and decide whether this leads to a red-blue colouring of $G[V(F_i) \cup N_1]$. If this is the case for every component F_i of $G[N_2]$ then G is a yes-instance. Otherwise we discard the branch.

Next we consider the case where every red vertex r in N_1 has red neighbours in at most one connected component of $G[N_2]$ containing uncoloured vertices. Let F_1 be a connected component of $G[N_2]$ containing an uncoloured vertex x together with a neighbour r_1 of r. Note that x and r are non-adjacent since $r \in S$ and the neighbourhood of S is coloured. Then $J = x - r_1 - r$ is an induced P_3 in G and at most $3d$ blue vertices in G have a neighbour on this path. Let Γ be the set of such vertices and consider each of $O(n^{3d^2})$ colourings of $N[\Gamma]$.

Any remaining blue vertex $b \in N_1$ with an uncoloured neighbour has no neighbours belonging to Γ. If b has uncoloured neighbours u and v in two different components F_2 and F_3 and a red non-neighbour r_2 in one of them, say F_2, we obtain an induced $P_3 + P_4$ consisting of these four vertices together with J, see Figure 6.4 (right). Therefore, b either has neighbours in at most one connected component of $G[N_2]$ different from F_1 or is adjacent to every red or uncoloured vertex in every connected component different from F_1 in which it has a neighbour.

Note that b has at most $d - 1$ red neighbours, else its remaining neighbours would be coloured blue. Since every connected component of $G[N_2]$ containing an uncoloured vertex contains a red vertex, b has uncoloured neighbours in at most $d - 1$ components.

Let T be the set of blue vertices in N_1 with uncoloured neighbours. For $t \in T$, let M_t be the set of cliques in N_2 in which t has uncoloured neighbours. For any two vertices t_i and $t_j \in T$, M_{t_i} and M_{t_j} are either disjoint or one is contained in the other. Assume otherwise. Then there exist blue vertices $t_i, t_j \in T$ and distinct integers $a, b, c \geq 2$ such that:

- $x \in F_a$ is adjacent to both t_i and t_j.
- $y \in F_b$ is adjacent to t_i but not t_j.
- $z \in F_c$ is adjacent to t_j but not t_i.

Then t_i and t_j are adjacent by $(P_3 + P_4)$-freeness. But then $y - b_i - b_j - z$ is an induced P_4 with no neighbours in J a contradiction.
Hence, we obtain a partition $D_1 \dots D_l$ of the cliques of N_2 containing uncoloured vertices such that, for each part D:

- No vertex of T with a neighbour in D_i has neighbours in any other part.
- Each part contains at most $d-1$ cliques, each of size at most $2d$.

Therefore, for each part D in turn, we test all of $2^{2d(d-1)}$ possible colourings of the uncoloured vertices in $G[V(D_i)]$ to decide whether there exists an extension of our red-blue colouring of N_1 to a red-blue d-colouring of $G[V(D) \cup N_1]$. If such a colouring exists, we have found a d-cut otherwise we discard the branch.

Case 2. In any red-blue d-colouring of G, every induced P_4 is bichromatic.
Note that we may assume that both the set B of blue vertices in a red-blue d-colouring of G and the set R of red vertices induce connected subgraphs of G. Otherwise, assume that $G[B]$ is disconnected. Then we may recolour all but one connected component of $G[B]$ red. If then $G[R]$ is disconnected, each of its connected components contains a vertex with a neighbour in B, else G is disconnected. We obtain a red-blue d-colouring where both $G[R]$ and $G[B]$ are connected by recolouring all but one connected component of $G[R]$ blue. Therefore in this case $G[B]$ and $G[R]$ have diameter at most 2, else we have a monochromatic P_4.

Let P be an induced P_4 in G. We can find P in time $O(n^4)$ and we branch over all 16 colourings of P. Further, we branch over all $O(n^{4d})$ colourings of its neighbourhood N_1. Let $N_2 = V(G) \setminus N[P]$. If $N_2 = \emptyset$ we are done. Otherwise, since G is (P_3+P_4)-free, every connected component of $G[N_2]$ is a clique. Let $F_1, \dots, F_k$, for some integer $k \geq 1$, be the connected components of $G[N_2]$. We colour-process the current colouring.

If all components $F_1, \dots, F_k$ are coloured we are done, otherwise there is a connected component, say F_1, of $G[N_2]$ with an uncoloured vertex x. Then x has at most d red neighbours and at most d blue neighbours in N_1, since otherwise it would have been coloured by colour-processing. Let $B_x \subseteq N_1$ be the set of blue neighbours of x in N_1 and $R_x \subseteq N_1$ the set of red neighbours. We branch over all $O(n^{2d^2})$ colourings of $G[N(B_x \cup R_x) \cup V(F_1)]$.

We colour N_2 as follows. If x is coloured blue then any blue vertex of N_2 not belonging to F_1 must have a neighbour in B_x since $G[B]$ has diameter 2. This implies that any uncoloured vertex must be red since it has no neighbour in B_x. Similarly, if x is red then any uncoloured vertex must be blue. We check in polynomial time if the resulting colouring is a red-blue d-colouring. If yes, we have found a solution, otherwise we discard the branch.

If in any case any branch led to a red-blue d-colouring of G we immediately returned that G has a red-blue d-colouring and thus a d-cut. Otherwise if no branch led to a red-blue d-colouring we conclude that G has no d-cut.

The correctness of the algorithm follows from its description. The maximum number of branches in Case 1 is

$$O\left(n^4 * n^{4d} * \left(n^{2d} + n^{4d^2} * \left(n^{4d^2} * n^{2d} + n^{3d^2}\right)\right)\right)$$

and in Case 2 there are $O(16 * n^4 * n^{4d} * n^{2d^2})$ branches. In total we have at most $n^{O(d^3)}$ branches and each branch can processed in polynomial time. Thus, the running time of our algorithm is polynomial. $\square$

Remark 6.2.8. We expect that this result can be generalized to $2P_4$-free graphs ([58]).

6.3 Hardness Results

We now give several hardness results for d-Cut. Among these are results for graphs with forbidden induced paths, see Section 6.3.2 and graphs with forbidden induced stars, see Section 6.3.3. The results for graphs with forbidden induced paths complement the polynomial-time results nicely. We will see that for $d \geq 3$ only three cases remain open, two of which we expect to be settled due to [58]. Further, the graphs constructed for the hardness results for forbidden induced paths have bounded radius and diameter and thus, contribute to the corresponding dichotomies. We start with our result for graphs of high girth.

6.3.1 Graphs of High Girth

Our first result is that d-Cut, for $d \geq 2$, is NP-complete for graphs of girth at least g, with $g \geq 3$. Recall that we already showed the same result for Matching Cut, see Theorem 5.1.4. The following proof is a generalization of the proof for Matching Cut. This requires us to replace Lemma 5.1.1 by the following lemma.

Lemma 6.3.1 ([21])**.** *There are infinitely many primes p, such that*

a) $p \geq 13$,

b) $p \equiv 1 \bmod 4$, *and*

c) there exists an infinite set Q_p such that the following holds for every $q \in Q_p$:

  i. $q > p$

  ii. $q \equiv 1 \bmod 4$, *and*

  iii. the Legendre symbol $\left(\frac{q}{p}\right) = -1$.

Proof. We first apply Theorem 2.1.2 to $a = 13$ and $b = 20$, which are coprime. In this way we find that the sequence $13 + 20k_1$, for $k_1 \in \mathbb{N}$, contains infinitely many primes $p \geq 13$. Hence, condition a) is satisfied. Since $p = 13 + 20k_1$ for some integer $k_1 \geq 0$, we find that $p \equiv 1 \bmod 4$. Hence, condition b) is satisfied as well.

We now show condition c). We apply Theorem 2.1.2 to $a = 5$ and $b = 4p$. In this way we find that the sequence $5 + 4pk_2$, for $k_2 \in \mathbb{N}$, contains infinitely many primes q, so in particular an infinite set Q_p with $q > p$ for every $q \in Q_p$. This shows that condition c)i is satisfied. For every $q \in Q_p$ it holds that $q = 5+4pk_2$ for some $k_2 \geq 0$ and thus $q \equiv 1 \bmod 4$. Hence, condition c)ii is satisfied as well.

Finally, we will show condition c)iii by proving that $\left(\frac{q}{p}\right) = -1$ holds for every $q \in Q_p$. First, we recall that if $a \equiv b \bmod c$, then $\left(\frac{a}{c}\right) = \left(\frac{b}{c}\right)$ (see [57]). We will also use the law of reciprocity, of which several equivalent versions exist in the literature. We use the following partial version (which can be found in [57]). For every two primes $a > 2$ and $b > 2$, it holds that

$$\left(\frac{a}{b}\right) = \left(\frac{b}{a}\right) \text{ if } a \equiv 1 \bmod 4 \text{ or } b \equiv 1 \bmod 4.$$

Combining both properties of the Legendre symbol we can now deduce that

$$\begin{aligned}\left(\frac{q}{p}\right) &= \left(\frac{5+4pk_2}{13+20k_1}\right) = \left(\frac{5+4k_2(13+20k_1)}{13+20k_1}\right) = \left(\frac{5}{13+20k_1}\right) \\ &= \left(\frac{13+20k_1}{5}\right) = \left(\frac{3}{5}\right) \equiv -1 \bmod 3.\end{aligned}$$

Thus, condition c)iii is satisfied as well, and we completed the proof of the lemma. □

We are now ready to generalize Theorem 5.1.4. Recall that $h(G)$ is the edge expansion of G.

Theorem 6.3.2 ([21]). *For every integer $d \geq 1$ and every integer $g \geq 3$, there is a function $f(d)$ that only depends on d, such that d-*Cut *is* NP*-complete for bipartite graphs of girth at least g and maximum degree at most $f(d)$.*

Proof. Let $d \geq 1$. By Observation 2.2.2, every graph G with $h(G) > d$ is d-immune. Hence, we can construct a $(p+1)$-regular bipartite gadget F using the arguments from the proof of Lemma 5.1.2, where Lemma 6.3.1 plays the role of Lemma 5.1.1. That is, we can choose p such that

(i) p is a prime congruent to 1 mod 4, and

(ii) $\frac{1}{2}(p+1-\lambda_2(G)) \geq \frac{1}{2}(p+1-2\sqrt{p}) > d$.

We now reduce from d-Cut. Gomes and Sau [29] proved that d-Cut is NP-hard for graphs of maximum degree at most $2d+2$. Hence, we may assume that the instance G from d-Cut has maximum degree at most $2d+2$. By applying the arguments of the proof of Theorem 5.1.4, we can use F and G to construct for every integer $g \geq 3$, a bipartite graph G' of girth at least g and maximum degree at most $f(d)$, such that G has a d-cut if and only if G' has a d-cut. □

Remark 6.3.3. Since it is not possible to give the smallest prime p satisfying conditions (i) and (ii) in the proof of Theorem 6.3.2, we can merely state that the maximum degree of G' is bounded by some function $f(d)$ that only depends on d.

6.3.2 Graphs without Induced Paths

In the following, we consider graphs with forbidden induced paths. Our first result, Theorem 6.3.4, for $3P_3$-free graphs of radius 2 and diameter 3 can be modified to hold for bipartite graphs of radius 3 and diameter 4 (see Theorem 6.3.5.) Further, it can be easily generalized to hold for $d \geq 2$. However, this is not needed, since the last result, Theorem 6.3.6, which only holds for $d \geq 3$, is stronger. Note that Theorem 6.3.4 and 6.3.6 show that there is a difference in complexity for Matching Cut and d-Cut, for $d \geq 2$, since for Matching Cut, Theorem 4.1.3 gives a polynomial-time algorithm for sP_3-free graphs. There also is a difference for graphs of radius 2 and diameter 3. Theorem 6.3.4 and 6.3.6 show the NP-completeness for d-Cut, for $d \geq 2$, on these graph classes, while we know that Matching Cut is solvable in polynomial time on graphs of radius 2 or diameter 3, see Theorem 4.1.1 and [39].

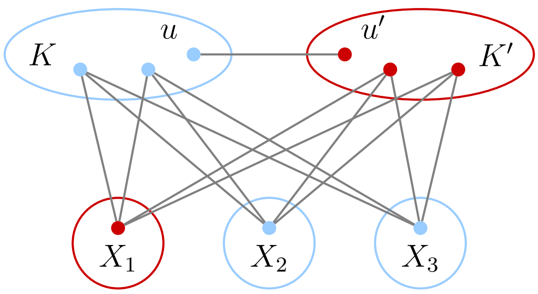


Figure 6.5: The graph G from the proof of Theorem 6.3.4 with a red-blue d-colouring.

Theorem 6.3.4 ([48]). *For $d = 2$, d-*Cut *is* NP*-complete for $3P_3$-free graphs of radius 2 and diameter 3.*

Proof. We reduce from Not-All-Equal SAT. Let $(X, \mathcal{C})$ be an instance of this problem, where $X = \{x_1, x_2, \ldots, x_n\}$ and $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$. By Theorem 2.1.10 we may assume that each literal occurs only positively and in at most three different clauses, and each C_i contains either two or three literals.

From $(X, \mathcal{C})$ we construct a graph G as follows. We introduce a clique K with vertex set $\{C_{1_1}, C_{1_2}, C_{2_1}, C_{2_2}, \ldots, C_{m_1}, C_{m_2}\}$ and a clique K' with vertex set $\{C'_{1_1}, C'_{1_2}, C'_{2_1}, C'_{2_2}, \ldots, C'_{m_1}, C'_{m_2}\}$. For every variable x_i we add a variable clique X_i of size 5. For x_i occurring in clause C_j choose a vertex v_{C_j} of X_i and add edges to C_{j_1}, C_{j_2}, C'_{j_1} and C'_{j_2}, see Figure 6.5 for an example. Ensure that every vertex in X_i is associated to at most one clause. For every clause C_i consisting of only two literals x_j and x_k we pick a yet unselected vertex u in one of the cliques X_j and X_k. If no such vertex exists, we add an additional vertex to X_j. We then connect u to $C_{i_1}, C_{i_2}, C'_{i_1}$ and C'_{i_2}. We call those vertices of X_i which are not selected to represent an occurrence of x_i auxiliary vertices. For each of these auxiliary vertices x, we add one new vertex adjacent to x to K and one to K'. To complete the construction, we add vertices u to K and u' to K'. Finally, we add the edge uu'.

We show that G is $3P_3$-free, has radius 2 and has diameter 3. The graph $G \setminus (K \cup K')$ consists of n independent cliques, each of which is P_3-free. Thus, every P_3 in G contains at least one vertex of $K \cup K'$. Since each of K and K' may only contain vertices of one P_3, G is $3P_3$-free. Consider the vertex $u \in V(K)$ connecting K and K'. Since every vertex in the variable cliques has a neighbour in K, the distance from u to any vertex of G is at most 2 and thus, G has radius 2. Let $v \in V(X_i)$ and $v' \in V(X_j)$ for $i, j \in \{1, \ldots, n\}$. Then, there is a path from v to v' of length at most 3 using their neighbours K which are either the same or adjacent. Similarly for $v \in V(K)$ and $v' \in V(K')$, the path $v - u - u' - v'$ has length 3. This shows that G has diameter 3.

We claim that $(X, \mathcal{C})$ is a yes-instance of Not-All-Equal SAT if and only if G has a 2-cut.

First suppose $(X, \mathcal{C})$ is a yes-instance of Not-All-Equal SAT. Then X has a truth assignment ϕ that sets, in each C_i, at least one literal true and at least one literal false. We colour a clique X_i red if the corresponding variable is set to true and blue otherwise. We colour the vertices in K red and those in K' blue. Consider a vertex v in X_i, say v is coloured red. Since v has the same colour as its clique X_i it has at most two neighbours of the other colour which are either in K or in K'. Now consider a vertex C_{i_j} in K, which is coloured red. As each clause contains two or three literals and ϕ sets at least one literal true, we find that C_{i_j} is adjacent to at most two blue vertices in the variable cliques. The other cases follow by symmetry. Hence, we obtain a red-blue 2-colouring of G. By Observation 6.1.1, this means G has a 2-cut.

Now suppose G has a 2-cut. By Observation 6.1.1, this means that G has a red-blue 2-colouring c. We may assume without loss of generality that $m \geq 2$, which implies that K and K' have size at least 5 and are thus, monochromatic. Say c colours every vertex of K red. For a contradiction, assume c colours every vertex of K' red as well. Then, in each variable clique there is a vertex with 4 red neighbours so this vertex has to be coloured red. Since the variable cliques have size 5, they are monochromatic and thus, the whole graph is coloured red, a contradiction. We conclude that c must colour every vertex of K' blue.

Assume for another contradiction that there exists a clause such that all its corresponding variable cliques are coloured blue. Then C_{i_j} in K, which is coloured red, has three blue neighbours, a contradiction. By symmetry, we get the same contradiction for C'_{i_j} in K'. Hence, setting x_i to true if X_i is red in G and to false otherwise gives us the desired truth assignment for X. □

We can modify this proof to hold for d-Cut, with $d \geq 2$, for bipartite graphs of radius 3 and diameter 4. The idea is to replace every clique by a complete bipartite graph. By

Observation 6.1.2, complete bipartite graphs have to be monochromatic if each partition class is large enough. The edges between variable gadgets and clause gadgets exist then twice – once for every partition class. This assures that the radius and diameter of the graph only increase by 1. Note that the fact that $d \geq 2$ is important in this proof.

Theorem 6.3.5. *For $d \geq 2$, d-CUT is NP-complete for bipartite graphs of radius 3 and diameter 4.*

Proof. We reduce from NOT-ALL-EQUAL SAT. Let $(X, \mathcal{C})$ be an instance of this problem, where $X = \{x_1, x_2, \ldots, x_n\}$ and $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$. By Theorem 2.1.10, we may assume that each literal occurs only positively and in at most three different clauses, and each C_i contains either two or three literals.

From $(X, \mathcal{C})$ we construct a graph G as follows. We introduce a complete bipartite graph K with partition classes $K_a = \{C^a_{1_1}, \ldots, C^a_{1_d}, \ldots, C^a_{m_1}, \ldots, C^a_{m_d}\}$ and $K_b = \{C^b_{1_1}, \ldots, C^b_{1_d}, \ldots, C^b_{m_1}, \ldots, C^b_{m_d}\}$ and a complete bipartite graph K' with partition classes $K'_a = \{C'^a_{1_1}, \ldots, C'^a_{1_d}, \ldots, C'^a_{m_1}, \ldots, C'^a_{m_d}\}$ and $K'_b = \{C'^b_{1_1}, \ldots, C'^b_{1_d}, \ldots, C'^b_{m_1}, \ldots, C'^b_{m_d}\}$. For every variable x_i we add a complete bipartite graph X_i with partition classes X^a_i and X^b_i of size $3d$ each. For each clause $C_j = \{x_i, x_k, x_\ell\}$ consisting of three literals we choose

- $d-1$ vertices of X^a_i and make them complete to $C^b_{j_1}, \ldots, C^b_{j_d}$ and $C'^b_{j_1}, \ldots, C'^b_{j_d}$;
- $d-1$ vertices of X^b_i and make them complete to $C^a_{j_1}, \ldots, C^a_{j_d}$ and $C'^a_{j_1}, \ldots, C'^a_{j_d}$;
- one vertex of X^a_k and add edges to $C^b_{j_1}, \ldots, C^b_{j_d}$ and $C'^b_{j_1}, \ldots, C'^b_{j_d}$;
- one vertex of X^b_k and add edges to $C^a_{j_1}, \ldots, C^a_{j_d}$ and $C'^a_{j_1}, \ldots, C'^a_{j_d}$;
- one vertex of X^a_ℓ and add edges to $C^b_{j_1}, \ldots, C^b_{j_d}$ and $C'^b_{j_1}, \ldots, C'^b_{j_d}$; and
- one vertex of X^b_ℓ and add edges to $C^a_{j_1}, \ldots, C^a_{j_d}$ and $C'^a_{j_1}, \ldots, C'^a_{j_d}$.

For each clause $C_j = \{x_i, x_k\}$ consisting of two literals we choose

- d vertices of X^a_i and make them complete to $C^b_{j_1}, \ldots, C^b_{j_d}$ and $C'^b_{j_1}, \ldots, C'^b_{j_d}$;
- d vertices of X^b_i and make them complete to $C^a_{j_1}, \ldots, C^a_{j_d}$ and $C'^a_{j_1}, \ldots, C'^a_{j_d}$;
- one vertex of X^a_k and add edges to $C^b_{j_1}, \ldots, C^b_{j_d}$ and $C'^b_{j_1}, \ldots, C'^b_{j_d}$; and
- one vertex of X^b_k and add edges to $C^a_{j_1}, \ldots, C^a_{j_d}$ and $C'^a_{j_1}, \ldots, C'^a_{j_d}$.

In all cases, ensure that every vertex in every X_i is associated to at most one clause. We call those vertices of X_i which are not selected to represent an occurrence of x_i *auxiliary vertices*. For each of these auxiliary vertices x, we add one new vertex adjacent to x to K_b and one to K'_b if $x \in X^a_i$ and to K_a and to K'_a if $x \in X^b_i$. To complete the construction, we add vertices u_a to K_a, u_b to K_b, u'_a to K'_a and u'_b to K'_b. Finally, we add the edges $u_a u'_b$ and $u_b u'_a$.

We show that G has radius 3 and diameter 4. Consider the vertex $u_a \in V(K)$ connecting K and K'. Since every vertex in the variable gadgets has a neighbour in K_a or K_b, the distance from u to any vertex of G is at most 3 and thus, G has radius 3. Let $v \in V(X_i)$ and $v' \in V(X_j)$ for $i, j \in \{1, \ldots, n\}$. Then, there is a path from v to v' of length at most 4 using their neighbours in K which are at distance at most 2. Similarly for $v \in V(K)$ and $v' \in V(K')$, v is adjacent to either u_a or u_b and the distance from v' to u'_a and u'_b is at most 2, resulting in a path of length at most 4. This shows that G has diameter 4.

We claim that $(X, \mathcal{C})$ is a yes-instance of NOT-ALL-EQUAL SAT if and only if G has a d-cut.

First suppose $(X, \mathcal{C})$ is a yes-instance of NOT-ALL-EQUAL SAT. Then X has a truth assignment ϕ that sets, in each clause C_i, at least one literal true and at least one literal false. We colour a variable gadget X_i red if the corresponding variable is set to true and blue otherwise. We colour the vertices in K red and those in K' blue. Consider a vertex

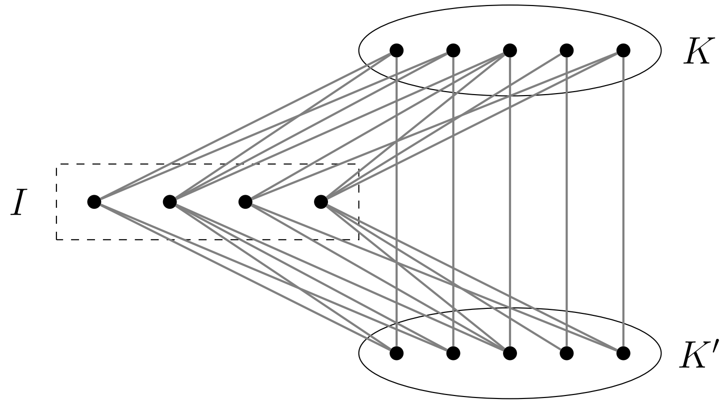


Figure 6.6: The $3P_2$-free graph G constructed in the proof of Theorem 6.3.6, where we did not draw edges inside K and K'.

v in X_i, say v is coloured red. Since v has the same colour as its variable gadget X_i it has at most d neighbours of the other colour which are either in K or in K'. Now consider a vertex C_{i_j} in K, which is coloured red. As each clause contains two or three literals and ϕ sets at least one literal true, we find that C_{i_j} is adjacent to at most d blue vertices in the variable cliques. The other cases follow by symmetry. Hence, we obtain a red-blue d-colouring of G. By Observation 6.1.1, this means G has a d-cut.

Now suppose G has a d-cut. By Observation 6.1.1, this means that G has a red-blue d-colouring c. We may assume without loss of generality that $m \geq 2$, which implies that the partition classes of K and K' have size at least $2d + 1$ and thus, K and K' are monochromatic. Say c colours every vertex of K red. For a contradiction, assume c colours every vertex of K' red as well. Then, in each variable gadget there is a vertex with $2d$ red neighbours so this vertex has to be coloured red. Since the partition classes of the variable gadgets have size $3d$, the variable gadgets are monochromatic and thus, the whole graph is coloured red, a contradiction. We conclude that c must colour every vertex of K' blue.

Assume for another contradiction that there exists a clause C_i, for $i \in \{1, \ldots, m\}$ such that all its corresponding variable gadgets are coloured blue. Then $C^a_{i_j}$ in K_a, for $j \in \{1, \ldots, d\}$, which is coloured red, has $d+1$ blue neighbours, a contradiction. By symmetry, we get the same contradiction for $C^b_{i_j}$ in K_b and $C'^a_{i_j}$ and $C'^b_{i_j}$ in K'. Hence, setting x_i to true if X_i is red in G and to false otherwise gives us the desired truth assignment for X. □

We now give a stronger result than the result in Theorem 6.3.4 for $d \geq 3$. Note that the construction is a simplified version of the construction of Theorem 6.3.4. This simplified construction enables us to achieve $3P_2$-freeness. The assumption that $d \geq 3$ is essential in the proof of Theorem 6.3.6 though. Note that using another variant of 3-SAT, we expect that we can generalize this result to $d = 2$ ([58]).

Theorem 6.3.6 ([48])**.** *For $d \geq 3$, d-Cut is NP-complete for $3P_2$-free graphs of radius 2 and diameter 3.*

Proof. First, let $d = 3$. We reduce from Not-All-Equal SAT. Let $(X, \mathcal{C})$ be an instance of this problem, where $X = \{x_1, x_2, ..., x_n\}$ and $\mathcal{C} = \{C_1, C_2, ..., C_m\}$. By Theorem 2.1.10 we may assume that each literal occurs only positively and appears in two or three different clauses, and each C_i contains either two or three literals.

From $(X, \mathcal{C})$, we construct a graph $G = (V, E)$ as follows. We introduce a clique $K = \{C_1, \ldots, C_m\}$; a clique $K' = \{C'_1, \ldots, C'_m\}$ and an independent set $I = \{x_1, \ldots, x_n\}$, such that K, K', I are pairwise disjoint and $V = K \cup K' \cup I$. We modify K and K' into larger cliques by adding a new vertex z_i to K and a new vertex z'_i to K' for each C_i that consists of two literals. We add an edge between x_h and C_i and an edge between x_h and C_i' if and only if x_h occurs as a literal in C_i. In other words, for every $i \in \{1, \ldots, m\}$, we have that C_i' is a *copy* of C_i. For every $i \in \{1, \ldots, m\}$, we also add an edge between C_i and C_i'. Finally, we add an edge between C_i and z_i' and between C_i' and z_i for each C_i that consists of two literals. See Figure 6.6 for an example.

As every edge must have at least one endvertex in K or K', and K and K' are cliques, we find that G is $3P_2$-free. Moreover, G has radius 2, as the distance from a vertex in $K \cup K'$ to any other vertex in G is at most 2. In addition, the distance from a vertex in I to any other vertex in G is at most 3. Hence, G has diameter at most 3.

We claim that $(X, \mathcal{C})$ is a yes-instance of Not-All-Equal SAT if and only if G has a 3-cut.

First suppose $(X, \mathcal{C})$ is a yes-instance of Not-All-Equal SAT. Then X has a truth assignment ϕ that sets, in each C_i, at least one literal true and at least one literal false. In I, we colour the true literals red and the false literals blue. We colour the vertices in K red and the vertices in K' blue. Consider a vertex x_h in I, say x_h is coloured red. As each literal appears in at most three clauses, we find that x_h has at most three blue neighbours (which all belong to K'). Now consider a vertex C_i in K, which is coloured red. As each clause contains two or three literals and ϕ sets at least one literal true, we find that C_i is adjacent to at most two blue vertices in I if C_i consists of three literals, whereas C_i is adjacent to at most one blue literal in I if C_i consists of two literals. By construction, C_i is adjacent to exactly one other blue vertex, which belongs to K', if C_i contains three literals, while C_i is adjacent to exactly two other blue vertices, which both belong to K', if C_i contains two literals. Hence, every C_i is adjacent to at most three blue literals. The other cases follow by symmetry. Hence, we obtained a red-blue 3-colouring of G. By Observation 6.1.1, this means that G has a 3-cut.

Now suppose G has a 3-cut. By Observation 6.1.1, this means that G has a red-blue 3-colouring c. We may assume without loss of generality that $m \geq 2d + 1$. This means that both K and K' are monochromatic. Say c colours every vertex of K red. For a contradiction, assume c colours every vertex of K' red as well. As c must colour at least one vertex of G blue, this means that I contains a blue vertex x_i. As each literal is contained in two or three clauses, we now find that a blue vertex, x_i, has at least two red neighbours in K and at least two red neighbours in K', so at least four red neighbours in total, a contradiction. We conclude that c must colour every vertex of K' blue.

For another contradiction, assume there exists a clause C_i whose three literals are all coloured blue in G. This means that a red vertex in G, namely C_i, has three blue neighbours in I and one blue neighbour in K', namely C_i', so at least four blue neighbours in total, a contradiction. Now assume there exists a clause C_i with two literals that are both coloured blue in G. This means that a red vertex in G, namely C_i, has two blue neighbours in I and two blue neighbour in K', namely C_i' and z_i', so at least four blue neighbours in total, a contradiction. As the vertices in K' are copies of the vertices in K, there is no clause either consisting of two or three literals that are all coloured red in G. Hence, setting x_i to true if x_i is red in G and to false if x_i is blue in G gives us the desired truth assignment for X.

Now, we consider the case where $d \geq 3$. We adjust G as follows. We first modify K into a larger clique by adding a set L_h of $d-3$ new vertices for each $h \in \{1, \ldots, m\}$. We also modify K' into a larger clique by adding a set L_h' of $d-3$ vertices for each $h \in \{1, \ldots, m\}$. For each $h \in \{1, \ldots, m\}$ we make x_h complete to L_h and to L_h'. Finally, we add additional edges between vertices in $K \cup L_1 \cup \ldots \cup L_m$ and vertices in $K' \cup L_1' \cup \ldots \cup L_m'$, in such a way that every vertex in K has $d-2$ neighbours in K', and vice versa. The modified graph G is still $3P_2$-free, has radius 2 and diameter 3, and also still has size polynomial in m and n. The remainder of the proof uses the same arguments as before. □

6.3.3 Graphs without Induced $K_{1,4}$ and H_i^*

In this last section, we show the hardness of d-Cut for $K_{1,4}$-free and H_i^*-free graphs. Both results work for any $d \geq 1$ and are generalizations of already known proofs for Matching Cut. For $d \geq 3$, a stronger result is known, namely that d-Cut is NP-complete for line graphs, see [52]. Recall that the class of line graphs is contained in the class of $K_{1,3}$-free graphs.

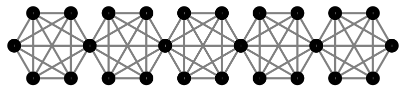


Figure 6.7: A chain of cliques of length 5 with cliques of size 6.

The first result is a generalization of a result of Chvátal, who showed in [12] that Matching Cut is NP-complete for $K_{1,4}$-free graphs. Our proof uses the following gadget. We call a *chain of cliques* of length k a set of k cliques of the same size, such that every clique, except two, has exactly two neighbouring cliques, with each of which it shares exactly one vertex, see Figure 6.7.

Theorem 6.3.7 ([48])**.** *For $d \geq 2$, d-Cut is NP-complete for $K_{1,4}$-free graphs.*

Proof. We reduce from Hypergraph 2-Colourability. Let H be a hypergraph. We construct a graph G as follows. Figure 6.8 illustrates the construction. Let $n = |V(H)|$ be the number of vertices of H and let $m = |E(H)|$ be the number of edges of H.

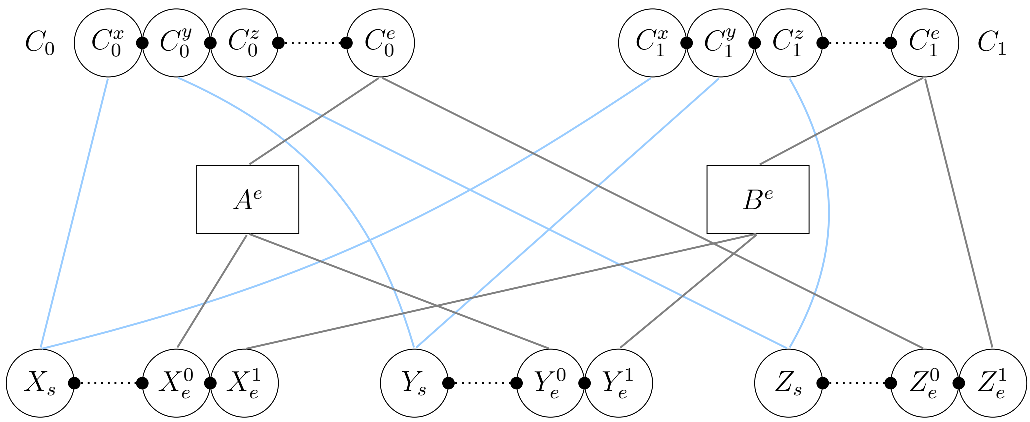


Figure 6.8: The construction of G. In this example there is only one edge $e = xyz$. Every cycle represents a clique of size $2d + 1$. The boxes represent independent sets.

Let C_0 and C_1 each be a chain of $n+m$ cliques of size $2d+1$. For every vertex $v \in V(H)$ there are cliques C_0^v in C_0 and C_1^v in C_1 and for every edge $e \in V(H)$ there are cliques C_0^e in C_0 and C_1^e in C_1. For every vertex $v \in V(H)$ we make a chain of cliques of size $2d + 1$, containing two cliques per edge in which v appears, and one extra clique. We denote the two cliques corresponding to v and the edge e as V_e^0 and V_e^1. We denote the extra clique by V_s. For an edge $e = (u, v, w)$ we add two independent sets, each of d vertices, A_e and B_e. We define $A = \bigcup_{e \in E(H)} A_e$ and $B = \bigcup_{e \in E(H)} B_e$.

We add all edges between A_e and d vertices of U_e^0, all edges between A_e and d vertices of V_e^0, all edges between A_e and d vertices $s_1, \ldots, s_d$ of C_0^e and all edges between $s_1, \ldots, s_d$ and d vertices of W_e^0. We do the same for B_e and the sets U_e^1, V_e^1, W_e^1 and C_1^e. For every vertex $v \in V(H)$ we add all edges between d vertices $s_1, \ldots, s_d$ in V_s and d vertices in C_0^v as well as all edges between $s_1, \ldots, s_d$ and d vertices in C_1^v.

We claim that H is bicolourable if and only if G has a d-cut. We assume that G has a d-cut. That is G has a red-blue d-colouring. Consider such a colouring.

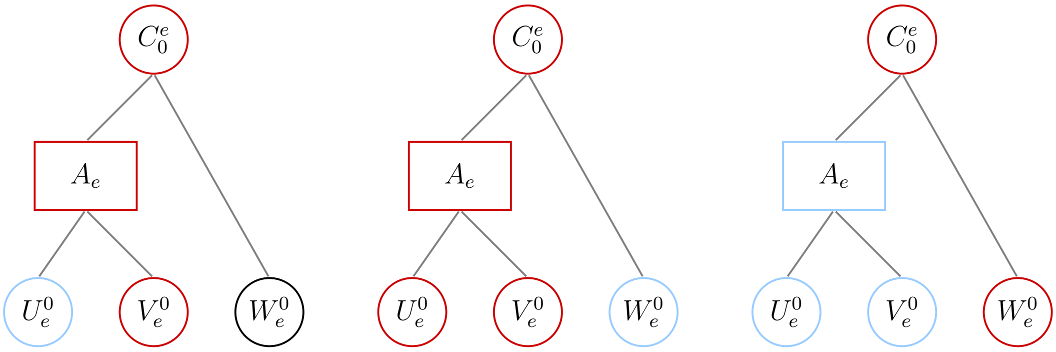


Figure 6.9: The three possible red-blue d-colourings of an edge gadget.

Claim 6.3.7.1. *Every chain of cliques is monochromatic.*

Proof. Recall that by Observation 6.1.2 every clique of size $2d+1$ is monochromatic in any red-blue d-colouring. Since the cliques of the chain share vertices, they all have the same colour. $\triangleleft$

Claim 6.3.7.2. *C_0 and C_1 have different colours.*

Proof. Suppose that C_0 and C_1 are both coloured blue. Then the cliques V_s will all be coloured blue for every $v \in V(H)$, since they have d neighbours in each of C_0 and C_1. Thus, by Claim 6.3.7.1 all chains of cliques corresponding to vertices are coloured blue. The only vertices which remain uncoloured are the vertices in $A \cup B$. Since any vertex in $A \cup B$ has $3d$ blue neighbours, A and B are both coloured blue and thus the whole graph is coloured blue and the colouring is not a red-blue d-colouring. Hence, C_0 and C_1 are coloured differently. $\triangleleft$

Claim 6.3.7.3. *Consider an edge $e = (u, v, w) \in E(H)$. The cliques U_e^0, V_e^0 and W_e^0 do not have all the same colour.*

Proof. Suppose that there is an edge e in H, such that U_e^0, V_e^0 and W_e^0 are coloured the same. Without loss of generality we may assume they are all coloured blue. By Claim 6.3.7.1 we know that the chains in which U_e^0, V_e^0 and W_e^0 are contained are coloured blue as well and especially that U_e^1, V_e^1 and W_e^1 are blue. Without loss of generality we may further assume that U_e^0 and V_e^0 are connected to A_e and similarly that U_e^1 and V_e^1 are connected to B_e. Then every vertex in A_e and B_e has $2d$ blue neighbours (the vertices U_e^0 and V_e^0 or the vertices in U_e^1 and V_e^1). Thus, A_e and B_e are both coloured blue. Hence, the connector vertices in C_0^e and C_1^e have each $2d$ blue neighbours (the vertices in W_e^0 and A_e or the vertices in W_e^1 and B_e) and are also coloured blue. Due to Claim 6.3.7.1 this leads to C_0 and C_1 being both blue, a contradiction to Claim 6.3.7.2. $\triangleleft$

Since Claim 6.3.7.3 holds for every edge in $E(H)$, we get that if G has a red-blue d-colouring, then H is bicolourable, by colouring every vertex of H according to the colour of its corresponding chain in a red-blue d-colouring.

We assume now that H is bicolourable. We consider a bicolouring of the vertices of H. We colour the vertices of G as follows. For each chain of cliques in G corresponding to a vertex $v \in V(H)$, we colour the chain of cliques according to the colour of v in the bicolouring of H. Further, we colour C_0 red and C_1 blue. Every vertex in A or B has d neighbours in each of its adjacent three cliques. We colour every vertex in A and B with the colour that at least two of its neighbouring cliques have. This clearly leads to a red-blue d-colouring of the vertices in A and B. For $v \in V(H)$, every vertex in V_e^0, V_e^1,

C_0^v and C_1^v has at most d neighbours outside of its monochromatic chain. Thus, no matter how we colour the vertices in A_e and B_e, the vertices in V_e^0, V_e^1, C_0^v and C_1^v have at most d neighbours of the other colour. The vertices in V_s which are connected to C_0 and C_1 have d neighbours of each colour, so also for them the colouring is a red-blue d-colouring.

Consider an edge $e = (u, v, w) \in E(H)$. It remains to show that the vertices in C_0^e and C_1^e have at most d neighbours of the other colour. Consider a vertex $x \in C_0^e$. Note that the analysis for vertices in C_1^e works analogously. The vertex x has d neighbours in A_e and d neighbours in W_e^0. Since we consider a bicolouring of H we know that one or two of U_e^0, V_e^0 and W_e^0 are blue and one or two are red. We distinguish several cases. First suppose that U_e^0 and V_e^0 have different colours (see Figure 6.9 (left)). Say without loss of generality that U_e^0 is blue and V_e^0 is red. Independently of the colour of W_e^0 we colour A_e red, since it has two neighbouring red cliques (C_0^e and V_e^0). Thus, x is red and has d red neighbours in A_e and either 0 or d blue neighbours in W_e^0. Now suppose that U_e^0 and V_e^0 are both red (see Figure 6.9 (middle)). Then, due to the bicolouring of H, A_e is red and W_e^0 is blue. Thus, x has d blue and d red neighbours. The last case we consider is where U_e^0 and V_e^0 are both blue (see Figure 6.9 (right)). Then A_e is blue and W_e^0 is red. Again, x has d blue and d red neighbours. We get that the colouring is a red-blue d-colouring and thus the theorem follows. □

Our last result in this section shows the NP-hardness of d-Cut for H_i^*-free graphs. This result is again mostly relevant for the case of $d = 2$ since the hardness result for line graphs in [52] includes the NP-completeness of d-Cut, for $d \geq 3$, for H_i^*-free graphs. Note that the proof is a (very) straightforward generalization of the proof of Theorem 5.2.1 for the corresponding result for $d = 1$, and below we just give a full proof for completeness.

Let uv be an edge of a graph G. We define an edge operation as displayed in Figure 5.5, which when applied on uv will replace uv in G by the subgraph T_{uv}^i. Note that in the new graph, the only vertices from T_{uv}^i that may have neighbours outside T_{uv}^i are u and v.

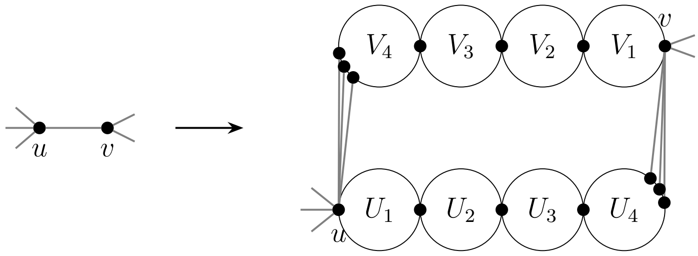


Figure 6.10: The edge uv (left) which we replace by the subgraph T_{uv}^i (right), here in the case where $d = 3$ and $i = 4$. Every cycle represents a clique of size $2d + 1$. The vertex u (v, resp.) has d neighbours in V_i (U_i, resp.).

Theorem 6.3.8 ([48])**.** *For $d \geq 2$ and $i \geq 1$, d-Cut is NP-complete for $(H_1^*, \ldots, H_i^*)$-free graphs.*

Proof. Fix $i \geq 1$. We reduce from Matching Cut. Let $G = (V, E)$ be a connected graph. Replace every $uv \in E$ by the graph T_{uv}^i (see Figure 6.10). This yields the graph $G' = (V', E')$.

We claim that G' is $(H_1^*, \ldots, H_i^*)$-free. For a contradiction, assume that G' contains an induced $H_{i'}^*$ for some $1 \leq i' \leq i$. Then G' contains two vertices x and y that are centers of an induced claw, as well as an induced path from x to y of length i'. All vertices in $V' \setminus V$ are not centers of any induced claw. Hence, x and y belong to V. By construction, any shortest path between two vertices of V has length at least $i + 1$ in G', a contradiction.

We claim that G' has a d-cut if and only if G has a matching cut. First suppose G' has a d-cut M', so G' has a red-blue d-colouring c'. We prove a claim for G':

Claim 6.3.8.1. *For every edge $uv \in E(G)$ it holds that*

(a) either $c'(u) = c'(v)$, and then T^i_{uv} is monochromatic, or

(b) $c'(u) \neq c'(v)$, and then c' colours $U_1, \dots, U_i$ with the same colour as u, while c' colours all vertices of $T^i_{uv} - \{u, U_1, \dots, U_i\}$ with the same colour as v, and moreover, the d edges from u to V_i and the d edges from v to U_i are in M'.

Proof. First assume $c'(u) = c'(v)$, say c' colours u and v red. As any clique of size at least $2d+1$ is monochromatic, all vertices in T^i_{uv} are coloured red, so T^i_{uv} is monochromatic.

Now assume $c'(u) \neq c'(v)$, say u is red and v blue. As before, we find that all cliques $U_1, \dots, U_i$ have the same colour as u, so are red, while all cliques $V_1, \dots, V_i$ have the same colour as v, so are blue. By definition, every edge xy with $c'(x) \neq c'(y)$ belongs to M'. $\lhd$

We construct a subset $M \subseteq E$ in G as follows. We add an edge $uv \in E$ to M if and only if $c'(u) \neq c'(v)$ in G'. We now show that M is a matching in G. Let $u \in V$. For a contradiction, suppose that M contains edges uv and uw for $v \neq w$. Then $c'(u) \neq c'(v)$ and $c'(u) \neq c'(w)$. By Claim 6.3.8.1, we find that M' matches u in G' to d vertices in each of T^i_{uv} and T^i_{uw}, contradicting our assumption that M' is a d-cut. Hence, M is a matching.

Now let c be the restriction of c' to V. If c colours every vertex of G with one colour, say red, then c' would also colour every vertex of G' red by Claim 6.3.8.1, contradicting that c' is a red-blue d-colouring. Hence, c uses both colours. Moreover, for every $uv \in E$, the following holds: if $c(u) \neq c(v)$, then $c'(u) \neq c'(v)$ and thus $uv \in M$. Hence, as M is a matching, c is a red-blue 1-colouring, and thus M is a matching cut of G.

Conversely, assume that G admits a matching cut, so V has a red-blue 1-colouring c. We construct a red-blue d-colouring c' of V' as follows.

- For every edge $uv \in E$ with $c(u) = c(v)$, we let $c'(x) = c(u)$ for every $x \in V(T^i_{uv})$.
- For every edge $uv \in E$ with $c(u) \neq c(v)$, we let $c'(u) = c(u)$ and $c'(u') = c(u)$, for every $u' \in \bigcup_{j=1}^{i} U_j$ and similarly, $c'(v) = c(v)$ as well as $c'(v') = c(v)$, for every $v' \in \bigcup_{j=1}^{i} V_j$.

As c is a red-blue 1-colouring, c uses both colours and thus by construction, c' uses both colours. Let $u \in V$. Again as c is a red-blue 1-colouring, $c(u) \neq c(v)$ holds for at most one neighbour v of u in G. Hence, by construction, u belongs to at most one non-monochromatic gadget T^i_{uv}. Thus, c' colours in G' at most d neighbours of u with a different colour than u. Let $u \in V' \setminus V$. By construction, we find that c' colours at most d neighbours of u with a different colour than u. Hence, c' is a red-blue d-colouring, and so G' has a d-cut. This completes the proof. $\square$

Chapter 7

Conclusion

In this thesis we considered the problem Matching Cut together with its two variants Disconnected Perfect Matching and Perfect Matching Cut. Further, we presented the first results explicitly proven for the maximisation problems Maximum Matching Cut and Maximum Disconnected Perfect Matching. Afterwards, in Chapter 6 we presented results for d-Cut, a generalization of Matching Cut. The results in this thesis focused on (bipartite) graphs of bounded radius and diameter and on H-free graphs. They contribute to different dichotomies. In the following we give an overview of the current state of the dichotomies, some of which are already complete and some are still incomplete. This overview allows us to detect interesting open problems.

7.1 Radius and Diameter

For all variants of matching cut considered in this thesis, there has been research on their complexity for graphs of bounded radius and bounded diameter. Most of the dichotomies are already complete and only two cases are missing. See also Tables 7.1 and 7.2 for an overview.

For Matching Cut, a result from this thesis completed the dichotomy for the radius, while the dichotomy for graphs of bounded diameter was already complete before.

Theorem 7.1.1 ([39, 49]). Matching Cut *is solvable in polynomial time for graphs of radius at most* 2 *and* NP*-complete on graphs of radius at least* 3.

Proof. The first part follows from Theorem 4.1.1, the second follows from [39] since the class of graphs of radius 3 is contained in the class of graphs of diameter 3. □

Theorem 7.1.2 ([7, 39]). Matching Cut *is solvable in polynomial time for graphs of diameter at most* 2 *and* NP*-complete on graphs of diameter at least* 3.

For Disconnected Perfect Matching the dichotomy for graphs of bounded diameter was proven in [8].

Theorem 7.1.3 ([8]). Disconnected Perfect Matching *is solvable in polynomial time for graphs of diameter at most* 2 *and* NP*-complete on graphs of diameter at least* 3.

However, for the radius we only know that Disconnected Perfect Matching is NP-complete for graphs of radius at least 3 by [40]. Since the case of radius 1 is trivial, we are left with one open case and thus, a first open problem.

Open Problem 1. What is the complexity of Disconnected Perfect Matching for graphs with radius 2?

	Matching Cut	Disconnected Perfect Matching	Perfect Matching Cut
P NP	diameter ≤ 2 ([7]) diameter ≥ 3 ([39])	diameter ≤ 2 ([8]) diameter ≥ 3 ([8])	diameter ≤ 2 (4.1.4) diameter ≥ 4 ([40])
P NP	radius ≤ 2 (4.1.1) radius ≥ 3 ([39])	radius ≤ 1 radius ≥ 3 ([40])	radius ≤ 2 (4.1.4) radius ≥ 3 ([40])

Table 7.1: The complexity of Matching Cut, Disconnected Perfect Matching and Perfect Matching Cut for graphs of bounded diameter and radius.

On a first glance it might be surprising that this question is still open, since for both Matching Cut and Perfect Matching Cut we know that they are solvable in polynomial time on graphs with radius 2. However, in the proofs for Matching Cut and Perfect Matching Cut, the final step, hidden in Lemma 3.2.2 (respectively Lemma 3.3.7), includes formulating the colouring problem as a 2-SAT formula, which can be efficiently solved. Since we can not encode the additional condition "the matching cut has to be contained in a perfect matching" using 2-SAT, solving the problem Disconnected Perfect Matching on graphs of radius 2 requires new ideas.

For Perfect Matching Cut the dichotomy for the radius is complete, while incomplete for the diameter where the case of diameter 3 remains open.

Theorem 7.1.4 ([40, 50])**.** Perfect Matching Cut *is solvable in polynomial time for graphs of radius at most* 2 *and* NP*-complete on graphs of radius at least* 3.

Proof. The first part follows from Theorem 4.1.4, the second is due to [40]. □

From [40] we know that Perfect Matching Cut is NP-hard for graphs of diameter 4 and from Theorem 4.1.4 it follows that it is solvable in polynomial time for graphs of diameter at most 2. For Perfect Matching Cut there are currently two ways known to obtain gadgets for NP-hardness proofs. The graphs constructed in the hardness reductions contain either some (subdivided) cubes or 4-subdivided edges. Both the cube (and thus all subdivisions) and a 4-subdivided edge have diameter 3 already, so it seems to be hard to construct a reduction graph with diameter 3. Hence, either Perfect Matching Cut is solvable in polynomial time on graphs of diameter 3 or new ideas are needed to prove the NP-hardness.

For now it is only known that the complexity of Matching Cut and Perfect Matching Cut differs for cubic graphs. Every subcubic graph with at least 7 vertices has a matching cut (see [53]) and it is NP-hard to decide if a 3-connected cubic bipartite planar graph has a perfect matching cut (see [5]). For graphs of bounded diameter and radius, the results are already rather similar. So the answer to the following question will show if Matching Cut and Perfect Matching Cut differ in complexity for graphs of bounded radius and diameter or not.

Open Problem 2. What is the complexity of Perfect Matching Cut for graphs with diameter 3?

For Maximum Matching Cut, Maximum Disconnected Perfect Matching and d-Cut, for $d \geq 2$, both dichotomies are settled. It is interesting to see that these three problems have the same dichotomies, see also Table 7.2. Note that for graphs of bounded diameter, these dichotomies are equal to the dichotomy for Matching Cut. However, in the case of graphs of bounded radius, Matching Cut is easier to solve in the sense

	Maximum Matching Cut	Maximum Disconnected Perfect Matching	d-Cut ($d \geq 2$)
P NP	diameter ≤ 2 (4.2.4) diameter ≥ 3 (5.3.8)	diameter ≤ 2 (4.2.11) diameter ≥ 3 (5.3.10)	diameter ≤ 2 (6.2.1) diameter ≥ 3 ([39], 6.3.4, 6.3.6)
P NP	radius ≤ 1 radius ≥ 2 (5.3.8)	radius ≤ 1 radius ≥ 2 (5.3.10)	radius ≤ 1 radius ≥ 2 (6.3.4, 6.3.6)

Table 7.2: The complexity of Maximum Matching Cut, Maximum Disconnected Perfect Matching and d-Cut, for $d \geq 2$, for graphs of bounded diameter and radius.

that for graphs of radius at most 2, Matching Cut is solvable in polynomial time while Maximum Matching Cut, Maximum Disconnected Perfect Matching and d-Cut, for $d \geq 2$, are NP-hard on this graph class. Before we state the dichotomies, we make the following straightforward observation that allows us to find a matching cut in a graph of radius 1.

Observation 7.1.5. *A graph with radius* 1 *has a vertex v which is dominating, that is, it is adjacent to every other vertex. This implies that G has a matching cut if and only if G has a vertex of degree* 1 *and a matching cut can be found in polynomial time.*

Theorem 7.1.6 ([51])**.** Maximum Matching Cut *is solvable in polynomial time for graphs of diameter at most* 2 *and NP-complete on graphs of diameter at least* 3.

Proof. The first part follows from Theorem 4.2.4, the second is due to Theorem 5.3.8. □

Theorem 7.1.7 ([51])**.** Maximum Matching Cut *is solvable in polynomial time for graphs of radius at most* 1 *and NP-complete on graphs of radius at least* 2.

Proof. From Observation 7.1.5 it follows that every matching cut of G has size exactly 1. So a maximum matching cut can be found in polynomial time, if it exists, and thus, the first part follows. The second is due to Theorem 5.3.8. □

Theorem 7.1.8 ([51])**.** Maximum Disconnected Perfect Matching *is solvable in polynomial time for graphs of diameter at most* 2 *and NP-complete on graphs of diameter at least* 3.

Proof. The first part follows from Theorem 4.2.11, the second is due to Theorem 5.3.10. □

Theorem 7.1.9 ([51])**.** Maximum Disconnected Perfect Matching *is solvable in polynomial time for graphs of radius at most* 1 *and NP-complete on graphs of radius at least* 2.

Proof. From Observation 7.1.5 it follows that every matching cut of G has size exactly 1. Thus, G has a disconnected perfect matching if and only if it has a vertex of degree 1 and a perfect matching. So the first part follows and the second is due to Theorem 5.3.10. □

Theorem 7.1.10 ([29, 48])**.** *d-*Cut *is solvable in polynomial time for graphs of diameter at most* 2 *and NP-complete on graphs of diameter at least* 3.

Proof. The first part follows from Theorem 6.2.1, the second is due to [29, 48]. □

Theorem 7.1.11 ([48])**.** *d-*Cut *is solvable in polynomial time for graphs of radius at most* 1 *and NP-complete on graphs of radius at least* 2.

	Matching Cut	Disconnected Perfect Matching	Perfect Matching Cut
P NP	diameter ≤ 3 ([39]) diameter ≥ 4 ([39])	diameter ≤ 3 ([8]) diameter ≥ 4 ([8])	diameter ≤ 3 (4.1.7)
P NP	radius ≤ 2 (4.1.1) radius ≥ 4 ([39])	radius ≤ 2 (4.2.13) radius ≥ 4 ([8])	radius ≤ 2 (4.1.4)

	Maximum Matching Cut	Maximum Disconnected Perfect Matching	d-Cut ($d \geq 2$)
P NP	diameter ≤ 3 (4.2.6) diameter ≥ 4 (5.3.9)	diameter ≤ 3 (4.2.14) diameter ≥ 4 (5.3.11)	diameter ≤ 3 (6.2.2) diameter ≥ 4 (6.3.5)
P NP	radius ≤ 2 (4.2.5) radius ≥ 3 (5.3.9)	radius ≤ 2 (4.2.12) radius ≥ 3 (5.3.11)	radius ≤ 1 radius ≥ 3 (6.3.5)

Table 7.3: The complexity of all six problems for bipartite graphs of bounded diameter and radius.

Proof. If the radius is 1, we can simply branch over all colourings of the neighbourhood of the dominating vertex and see if any of these colourings is a red-blue d-colouring. So the first part follows and the second is due to [48]. □

For Matching Cut and Disconnected Perfect Matching not only the complexity on graphs of bounded diameter has been considered but also on bipartite graphs of bounded diameter. For bipartite graphs the obtained results are different to the results for general graphs. In this thesis we gave many results for bipartite graphs of bounded radius and diameter. In the following we give an overview over the known results, see also Table 7.3. For bipartite graphs of diameter 3 there exist polynomial-time algorithms for Matching Cut (see [39]) and Disconnected Perfect Matching (see [8]), while for diameter 4 both problems become NP-complete (see again [8, 39]).

For Perfect Matching Cut, we saw in Theorem 4.1.7 that it is polynomial-time solvable for bipartite graphs of diameter 3, by slightly modifying the proof of Le and Le for Matching Cut in the same graph class. However, to the best of our knowledge there is no result showing that Perfect Matching Cut is NP-hard on bipartite graphs of bounded diameter. We expect that there exists a constant r such that Perfect Matching Cut is NP-hard for bipartite graphs of diameter r, but for now this remains open.

Open Problem 3. What is the complexity of Perfect Matching Cut for bipartite graphs of diameter g, with $g \geq 4$?

In Theorem 4.2.5 and 4.2.12, we have seen that Maximum Matching Cut and Maximum Disconnected Perfect Matching are solvable in polynomial time for bipartite graphs of radius at most 2 and thus, diameter 2. Further, in Theorem 4.2.6 and 4.2.14 we showed that they are as well solvable in polynomial time for bipartite graphs of diameter 3. On the other hand Theorem 5.3.9 and 5.3.11 show the NP-hardness for bipartite graphs of radius 3 and diameter 4 and thus, we obtain the following dichotomies.

Theorem 7.1.12. Maximum Matching Cut *and* Maximum Disconnected Perfect Matching *are solvable in polynomial time for bipartite graphs of diameter at most* 3 *and* NP*-complete for bipartite graphs of diameter at least* 4.

Theorem 7.1.13. Maximum Matching Cut *and* Maximum Disconnected Perfect Matching *are solvable in polynomial time for bipartite graphs of radius at most* 2 *and* NP-*complete for bipartite graphs of radius at least* 3.

These results add up nicely to the results for Matching Cut and Disconnected Perfect Matching. Recall that there when considering bipartite graphs, the problems stay solvable in polynomial time even if we increase the diameter by 1.

For d-Cut, we have seen in Theorem 6.2.2 a polynomial-time algorithm for bipartite graphs of diameter 3. In Theorem 6.3.5 we saw the counterpart that showed the NP-hardness of d-Cut for graphs of radius 3 and diameter 4. Thus, the only open case to complete both dichotomies is the case of bipartite graphs of radius 2. So we ask:

Open Problem 4. What is the complexity of d-Cut for bipartite graphs of radius 2?

Concerning the radius, we have already seen that we got the complete dichotomy for Maximum Matching Cut and Maximum Disconnected Perfect Matching. However, for Matching Cut we only know that it is solvable in polynomial time for graphs of radius 2 (see Theorem 4.1.1) and thus for bipartite graphs of radius 2 and moreover, that it is NP-complete on bipartite graphs for diameter 4 and thus of radius 4 (see [39]). So the case of radius 3 remains open.

Open Problem 5. What is the complexity of Matching Cut for bipartite graphs of radius 3?

Note that for Disconnected Perfect Matching we have the same gap. Here it follows from Corollary 4.2.13 that Disconnected Perfect Matching is solvable in polynomial time for bipartite graphs of radius 2 and we get the hardness for graphs of radius 4 from [8]. So we ask:

Open Problem 6. What is the complexity of Disconnected Perfect Matching for bipartite graphs of radius 3?

Finally, for Perfect Matching Cut even less is known. We only know that a polynomial-time algorithms for bipartite graphs of radius 2 follows from Theorem 4.1.4 and no hardness result is known for bipartite graphs of bounded radius. This leads to our last open problem in this section.

Open Problem 7. What is the complexity of Perfect Matching Cut for bipartite graphs of radius r, with $r \geq 3$?

7.2 H-free graphs

In this section we discuss the many results that exist for H-free graphs. We will see that for Maximum Matching Cut and Maximum Disconnected Perfect Matching the dichotomies are already complete. Further, for d-Cut only a small number of cases is open. For Matching Cut, Disconnected Perfect Matching and Perfect Matching Cut more cases are open. The overview which we get in the following allows us to identify cases that might be helpful to complete these partial dichotomies.

First, we consider Maximum Matching Cut and Maximum Disconnected Perfect Matching. For these problems this thesis gives all necessary results to obtain the two dichotomies, see also Table 7.4. Note that the dichotomies are the same.

Theorem 7.2.1 ([51])**.** *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*

	Maximum Matching Cut	Maximum Disconnected Perfect Matching
P	$sP_2 + P_6$ (4.2.2, 4.2.3)	$sP_2 + P_6$ (4.2.8, 4.2.10)
NP	$2P_3$ (5.3.8) P_7 (5.3.8) $K_{1,3}$ (5.4.1) $C_r, r \geq 3$ (5.1.12)	$2P_3$ (5.3.10) P_7 (5.3.10) $K_{1,3}$ (5.4.2) $C_r, r \geq 3$ (5.1.13)

Table 7.4: The complete dichotomies for Maximum Matching Cut and Maximum Disconnected Perfect Matching on H-free graphs.

- *NP-hard if $H \supseteq_i K_{1,3}$, $2P_3$ or $H \supseteq_i C_r$ for some $r \geq 3$.*

Proof. Let H be a graph. If H contains a cycle, then Maximum Matching Cut is NP-hard due to Theorem 5.1.12. Now suppose that 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 can apply Theorem 5.4.1.

Now suppose that 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 4.2.2 in combination with s applications of Theorem 4.2.3. Else H contains an induced $2P_3$ and we apply Theorem 5.3.8. This completes the proof. □

Theorem 7.2.2 ([51]). *For a graph H,* Maximum Disconnected Perfect Matching *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 $r \geq 3$.*

Proof. Let H be a graph. First suppose that H has a cycle. Then, Maximum Disconnected Perfect Matching is NP-hard for due to Theorem 5.1.13. If H is not a cycle, then 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, which contains the class of line graphs, so we apply Theorem 5.4.2. Otherwise H is a linear forest. If $H \subseteq_i sP_2 + P_6$ for some $s \geq 0$, then we apply Theorem 4.2.8 in combination with s applications of Theorem 4.2.10. Else H contains an induced $2P_3$ and we apply Theorem 5.3.10. □

For the classical Matching Cut problem and the other variants the dichotomies are still incomplete. In the following we give the current partial dichotomies.

Theorem 7.2.3 ([6, 12, 21, 40, 42, 49, 50, 53]). *For a graph H,* Matching Cut *on H-free graphs is*

- *polynomial-time solvable if $H \subseteq_i sP_3 + S_{1,1,2}$ or $sP_3 + P_6$ for some $s \geq 0$, and*
- *NP-complete if $H \supseteq_i K_{1,4}$, P_{14}, $2P_7$, $3P_5$, C_r for some $r \geq 3$, or H_j^* for some $j \geq 1$.*

Theorem 7.2.4 ([8, 21, 40, 51]). *For a graph H,* Disconnected Perfect Matching *on H-free graphs is*

- *polynomial-time solvable if $H \subseteq_i sP_2 + K_{1,3}$ or $sP_2 + P_6$ 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 $j \geq 1$.*

Theorem 7.2.5 ([21, 40, 44, 50]). *For a graph H,* Perfect Matching Cut *on H-free graphs is*

	Matching Cut	Disconnected Perfect Matching	Perfect Matching Cut
P	$sP_3 + S_{1,1,2}$ ([42], 4.1.3) $sP_3 + P_6$ (4.1.2, 4.1.3)	$sP_2 + K_{1,3}$ ([8], 4.2.10) $sP_2 + P_6$ (4.2.9, 4.2.10)	$sP_4 + S_{1,2,2}$ ([44], 4.1.6) $sP_4 + P_6$ (4.1.5, 4.1.6)
NP	$3P_5$ (5.3.1) $2P_7$ ([40]) P_{14} ([40]) $K_{1,4}$ ([12]) $C_r, r \geq 3$ (5.1.4) $H_i^*, i \geq 1$ (5.2.1)	$3P_6$ ([40]) $2P_7$ ([40]) P_{14} ([40]) $K_{1,4}$ ([8]) $C_r, r \geq 3$ (5.1.7) $H_i^*, i \geq 1$ (5.2.2)	$3P_6$ ([40]) $2P_7$ ([40]) P_{14} ([40]) $K_{1,4}$ ([44]) $C_r, r \geq 3$ ([44]) $H_i^*, i \geq 1$ ([44])

Table 7.5: Comparison of Matching Cut, Disconnected Perfect Matching and Perfect Matching Cut for different classes of H-free graphs.

- *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}*,* $2P_7$*,* $3P_6$*,* C_r *for some* $r \geq 3$ *or* H_j^* *for some* $j \geq 1$*.*

Table 7.5 illustrates the state of the art for Matching Cut, Disconnected Perfect Matching and Perfect Matching Cut for H-free graphs. For all three problems the dichotomies are still incomplete, leading to the general problem:

Open Problem 8. Complete the dichotomies of Matching Cut, Disconnected Perfect Matching and Perfect Matching Cut for H-free graphs.

The known results restrict the graphs H which we still have to consider to complete the dichotomies. In the following we concentrate on Matching Cut. Due to the similarity of the known results the analysis for Disconnected Perfect Matching and Perfect Matching Cut can be done analogously. Let H be a graph such that the complexity of Matching Cut for H-free graphs is not yet known. Since Matching Cut is NP-complete when we forbid C_r, for some $r \geq 3$, we get that H should not contain a cycle and therefore is a forest. The NP-completeness for $K_{1,4}$-free graphs implies that no vertex of H may have degree more than three. Finally, knowing that Matching Cut is NP-complete for H_i^*-free graphs, for any $i \geq 1$, tells us that H is a forest with at most one vertex of degree 3 per connected component. Thus, each component of H is either a path P_r, for some integer $r \geq 1$, or a $S_{i,j,k}$, for some integers $i, j, k \geq 1$. The results on forbidden paths reduce the options further. For example, if H consists of a single connected component, the open cases contain the cases when H is a P_r with $7 \leq r \leq 13$, but also $S_{i,j,k}$ for different combinations of integers $i, j, k \geq 1$. If H consists of several components more cases remain open. Even though there is only a finite number of open cases left, they are hard to enumerate. Still there are a few open cases which we assume to be helpful to complete the dichotomy.

Open Problem 9. What is the complexity of Matching Cut for H-free graphs, when $H \in \{sP_4, P_7, S_{1,2,2}\}$, for some $s \geq 1$?

Even though the analysis for Disconnected Perfect Matching and Perfect Matching Cut is similar, there are small differences in the knowledge we have and thus, we point out different open cases.

Open Problem 10. What is the complexity of Disconnected Perfect Matching for H-free graphs, when $H \in \{sP_3, P_7, S_{1,2,2}\}$, for some $s \geq 1$?

Open Problem 11. What is the complexity of PERFECT MATCHING CUT for H-free graphs, when $H \in \{sP_5, 2P_6, P_7, S_{2,2,2}\}$ for some $s \geq 1$?

It can be seen that MATCHING CUT, DISCONNECTED PERFECT MATCHING and PERFECT MATCHING CUT differ in complexity compared to the two maximisation variants MAXIMUM MATCHING CUT and MAXIMUM DISCONNECTED PERFECT MATCHING, see Table 7.4. There are several graphs H where MATCHING CUT and PERFECT MATCHING CUT are solvable in polynomial time on H-free graphs, but MAXIMUM MATCHING CUT is not. In contrary, for MAXIMUM DISCONNECTED PERFECT MATCHING and DISCONNECTED PERFECT MATCHING a difference in complexity is currently known only for the case of $K_{1,3}$-free graphs.

Another interesting detail is that we do not know if there is a difference in complexity between MATCHING CUT, DISCONNECTED PERFECT MATCHING and PERFECT MATCHING CUT for H-free graphs. A good candidate to show that they are different could be the case of sP_r-free graphs, for some r. The current results all exclude paths of different lengths. The $H + P_3$ result for MATCHING CUT and the $H + P_4$ result for PERFECT MATCHING CUT both use 2-SAT as a final step. Recall that the condition "the matching cut has to be contained in a perfect matching" can not be expressed using 2-SAT. So it is not clear how for example Theorem 4.2.10 for DISCONNECTED PERFECT MATCHING could be generalized to $H + P_3$. The generalization of Theorem 4.1.3 for MATCHING CUT to $H + P_4$-free graphs might be possible, but is not straightforward. So we pose the three following problems:

Open Problem 12. What is the complexity of DISCONNECTED PERFECT MATCHING for $(H + P_3)$-free graphs, when it is solvable in polynomial time for H-free graphs?

Open Problem 13. Can Theorem 4.1.3 for MATCHING CUT be generalized to hold for $(H + P_4)$-free graphs?

Open Problem 14. Do the problems MATCHING CUT, DISCONNECTED PERFECT MATCHING and PERFECT MATCHING CUT differ in complexity for H-free graphs for some graph H?

When considering d-CUT, we have to distinguish between the case $d = 1$, which is the standard MATCHING CUT problem, and the cases $d = 2$ and $d \geq 3$.

Theorem 7.2.6 ([21, 48]). *For a graph H, 2-CUT on H-free graphs is*

- *polynomial-time solvable if $H \subseteq_i sP_1 + P_3 + P_4$ or $sP_1 + P_5$ for some $s \geq 0$, and*
- *NP-complete if $H \supseteq_i K_{1,4}$, $3P_3$, C_r for some $r \geq 3$, or H_i^* for some $i \geq 1$.*

Theorem 7.2.7 ([21, 48, 52]). *Let $d \geq 3$. For a graph H, d-CUT on H-free graphs is*

- *polynomial-time solvable if $H \subseteq_i sP_1 + P_3 + P_4$ or $sP_1 + P_5$ for some $s \geq 0$, and*
- *NP-complete if $H \supseteq_i K_{1,3}$, $3P_2$ or C_r for some $r \geq 3$.*

In general d-CUT, for $d \geq 2$, seems to behave complexitywise more like the maximisation variants than like MATCHING CUT, see Table 7.6. For $d \geq 3$ the dichotomy is almost complete. Only three cases remain open, of which we expect the cases P_6 and $2P_4$ to be already settled [58]. We state them as our next open problem:

Open Problem 15. What is the complexity of d-CUT, for $d \geq 3$, on H-free graphs, where $H \in \{P_6, P_7, 2P_4\}$?

For 2-CUT more cases remain open, even after considering it to be polynomial-time solvable on $2P_4$-free and P_6-free graphs and NP-complete for $3P_2$-free graphs [58], since especially the case of claw-free graphs remains open for 2-CUT. Recall that this case is settled for d-CUT, for $d \geq 3$, and all other problems considered in this thesis.

	Matching Cut	2-Cut	d-Cut ($d \geq 3$)
P	$K_{1,3} + sP_3$ ([6], 4.1.3) $P_6 + sP_3$ (4.1.2, 4.1.3)	$P_4 + P_3 + sP_1$ (6.2.7, 6.2.4) $P_5 + sP_1$ (6.2.5, 6.2.4)	$P_4 + P_3 + sP_1$ (6.2.7, 6.2.4) $P_5 + sP_1$ (6.2.5, 6.2.4)
NP	$3P_5$ (5.3.1) $2P_7$ ([40]) P_{14} ([40]) $K_{1,4}$ ([12]) $C_r, r \geq 3$ (5.1.4) $H_i^*, i \geq 1$ (5.2.1)	$3P_3$ (6.3.4) P_{11} (6.3.4) $K_{1,4}$ (6.3.7) $C_r, r \geq 3$ (6.3.2) $H_i^*, i \geq 1$ (6.3.8)	$3P_2$ (6.3.6) P_8 (6.3.6) $K_{1,3}$ ([52]) $C_r, r \geq 3$ (6.3.2) $H_i^*, i \geq 1$ (6.3.8)

Table 7.6: Comparison of Matching Cut, 2-Cut and d-Cut, for $d \geq 3$, for different classes of H-free graphs.

Open Problem 16. What is the complexity of 2-Cut on claw-free graphs?

The case of claw-free graphs might be a candidate to answer the following question, since Matching Cut is solvable in polynomial time on claw-free graphs, while d-Cut, for $d \geq 3$, is NP-complete on claw-free graphs.

Open Problem 17. Do the complexities of d-Cut for $d = 2$ and $d \geq 3$ differ on H-free graphs?

After answering the case of claw-free graphs, it is a natural question to complete the dichotomy also for $d = 2$ and we ask:

Open Problem 18. Complete the dichotomy for 2-Cut on H-free graphs.

7.3 Further Results

Matching Cut and its variants are not only interesting for H-free graphs and graphs of bounded radius and diameter. Research has been done for other graphs classes. In the following we present a few open problems related to further results for Matching Cut. We first consider graphs of bounded degree. For Matching Cut we know that it is solvable in polynomial time for graphs of maximum degree $\Delta = 3$ [12] and NP-complete when $\Delta = 4$ [43]. For Disconnected Perfect Matching, the dichotomy is not yet complete. It is trivially solvable in polynomial time for graphs of maximum degree $\Delta = 2$ and NP-complete for planar graphs of maximum degree $\Delta = 4$ [8]. Bouquet and Picouleau further showed in [8] that Disconnected Perfect Matching is solvable in polynomial time for planar cubic graphs and give the case of cubic graphs as an open problem.

Open Problem 19. Determine the complexity of Disconnected Perfect Matching for cubic graphs.

This is an interesting open problem, since Perfect Matching Cut is known to be NP-complete for graphs of maximum degree $\Delta = 3$ [5].

Even though for Disconnected Perfect Matching the complexity for cubic graphs is open, we know that Maximum Disconnected Perfect Matching, is NP-hard on this graph class. This is due the fact that Perfect Matching Cut is NP-complete if $\Delta = 3$. Recall that NP-completeness results for Perfect Matching Cut immediately give NP-hardness results for Maximum Matching Cut and Maximum Disconnected Perfect Matching. Since Maximum Matching Cut and Maximum Disconnected

Perfect Matching are trivial if $\Delta = 2$, the dichotomy is complete for graphs of bounded degree.

For d-Cut, the problem of determining a dichotomy for graphs of bounded degree is of special interest. Gomes and Sau gave a polynomial-time algorithm for graphs of maximum degree $d + 2$ and proved the NP-hardness for $2d + 2$-regular graphs, see [29]. They posed the following question:

Open Problem 20. What is the complexity of d-Cut for graphs of maximum degree Δ with $d + 3 \leq \Delta \leq 2d + 1$?

This question is interesting, since an NP-hardness proof for graphs of smaller degree than $2d + 2$ would disprove a conjecture on internal partitions, see [29] for more explanation. Thus, Gomes and Sau suppose that their bound can not be improved. Recall that for $d = 1$, that is for Matching Cut, the problem is already solved. For $d = 2$ only the case of $\Delta = 5$ remains open, which still seems to be hard to solve, but it might be a first step to completely answer Open Problem 21. So we ask:

Open Problem 21. What is the complexity of 2-Cut for graphs of maximum degree 5?

Recall that the maximum degree in the NP-hardness construction for graphs of fixed girth was 60 for Matching Cut, 74 for Disconnected Perfect Matching and an unspecified constant, which depends on d for d-Cut. This leads to following the natural question.

Open Problem 22. Can we reduce the degree bounds in Theorems 5.1.4, 5.1.7 and 6.3.2?

To achieve this goal a new proof is needed, since for the currently used method, the degree bounds are best possible. For Perfect Matching Cut, Maximum Matching Cut and Maximum Disconnected Perfect Matching, the 4-subdivision method explained in Section 5.1.3 gave proofs for graphs with small maximum degree. Note that the obtained results are best possible with respect to the maximum degree.

When forbidding induced cycles not only the girth is of interest but also the chordality of a graph. For Matching Cut, Disconnected Perfect Matching and Perfect Matching Cut it is known that they are solvable in polynomial time for quadrangulated graphs (see [40, 41, 53]). The graph constructed by Le and Le in [40] has chordality 8. We conclude with the following problem.

Open Problem 23. Does there exist a class of graphs, which is k-chordal, for $k \leq 7$, for which one of Matching Cut, Disconnected Perfect Matching and Perfect Matching Cut is NP-complete?

Bibliography

[1] Noga Alon and Vitali D. Milman. λ_1, isoperimetric inequalities for graphs, and superconcentrators. *Journal of Combinatorial Theory, Series B*, 38:73–88, 1985.

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

[3] 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.

[4] Gábor Bacsó and Zsolt Tuza. Dominating cliques in P_5-free graphs. *Periodica Mathematica Hungarica*, 21(4):303–308, 1990.

[5] Edouard Bonnet, Dibyayan Chakraborty, and Julien Duron. Cutting Barnette graphs perfectly is hard. *Proc. WG 2023, LNCS*, 14093:116–129, 2023.

[6] 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.

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

[8] Valentin Bouquet and Christophe Picouleau. The complexity of the Perfect Matching-Cut problem. *to appear in Journal of Graph Theory*, 2024.

[9] Henning Bruhn-Fujimoto. Tikz-figures. Accessed on 20.08.2024. URL: `https://www.uni-ulm.de/fileadmin/website_uni_ulm/mawi.inst.081/Henning/figs.pdf`.

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

[11] 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.

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

[13] Clayton W. Commander. Maximum Cut problem, MAX-CUT. In *Encyclopedia of Optimization, Second Edition*, pages 1991–1999. Springer, 2009.

[14] Giuseppe Di Battista, Maurizio Patrignani, and Francesco Vargiu. A split & push approach to 3D orthogonal drawing. In *Graph Algorithms And Applications 2*, pages 139–167. World Scientific, 2004.

[15] Peter G. Lejeune Dirichlet. Beweis des Satzes, dass jede unbegrenzte arithmetische Progression, deren erstes Glied und Differenz ganze Zahlen ohne gemeinschaftlichen Faktor sind, unendlich viele Primzahlen enthält. *Abhandlungen der Königlichen Preußischen Akademie der Wissenschaften zu Berlin*, 1837.

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

[17] Jozef Dodziuk. Difference equations, isoperimetric inequality and transience of certain random walks. *Transactions of the American Mathematical Society*, 284:787–794, 1984.

[18] Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of mathematics*, 17:449–467, 1965.

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

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

[21] Carl Feghali, Felicia Lucke, Daniël Paulusma, and Bernard Ries. Matching cuts in graphs of high girth and H-free graphs. *Proc. ISAAC 2023, LIPIcs*, 283:28:1–28:16, 2023.

[22] Lester Randolph Ford and Delbert R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[23] Harold N. Gabow. A matroid approach to finding edge connectivity and packing arborescences. *Journal of Computer System Sciences*, 50:259–273, 1995.

[24] M. R. Garey, David S. Johnson, and Larry J. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1:237–267, 1976.

[25] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., USA, 1979.

[26] 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.

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

[28] Petr A. Golovach, Daniël Paulusma, and Jian Song. Closing complexity gaps for coloring problems on H-free graphs. *Information and Computation*, 237:204–214, 2014.

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

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

[31] P. Hall. On representatives of subsets. *Journal of the London Mathematical Society*, s1-10(1):26–30, 1935.

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

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

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

[35] 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.

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

[37] Melven R. Krom. The decision problem for a class of first-order formulas in which all disjunctions are binary. *Mathematical Logic Quarterly*, 13(1-2):15–20, 1967.

[38] Hoàng-Oanh Le and Van Bang Le. On the complexity of Matching Cut in graphs of fixed diameter. *Proc. ISAAC 2016, LIPIcs*, pages 50:1–50:12, 2016.

[39] 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.

[40] Hoàng-Oanh Le and Van Bang Le. Complexity results for matching cut problems in graphs without long induced paths. *Proc. WG 2023, LNCS*, 14093:417–431, 2023.

[41] Hoàng-Oanh Le and Van Bang Le. Complexity and algorithms for matching cut problems in graphs without long induced paths and cycles. *CoRR*, abs/2307.05402, 2024.

[42] Van Bang Le. Private communication, 2024.

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

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

[45] Jiping Liu and Huishan Zhou. Dominating subgraphs in graphs with some forbidden structures. *Discrete Mathematics*, 135:163–168, 1994.

[46] László Lovász. Coverings and colorings of hypergraphs. In *Proc. 4th Southeastern Conference of Combinatorics, Graph Theory, and Computing*, pages 3–12. Utilitas Mathematica Publishing, 1973.

[47] Alexander Lubotzky, Ralph Phillips, and Peter Sarnak. Ramanujan graphs. *Combinatorica*, 8:261–277, 1988.

[48] Felicia Lucke, Ali Momeni, Daniël Paulusma, and Siani Smith. Finding d-Cuts in graphs of bounded diameter, graphs of bounded radius and H-free graphs. *Proc. WG 2024, LNCS*, 2024.

[49] 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.

[50] Felicia Lucke, Daniël Paulusma, and Bernard Ries. Finding matching cuts in H-free graphs. *Algorithmica*, 85:3290–3322, 2023.

[51] Felicia Lucke, Daniël Paulusma, and Bernard Ries. Dichotomies for maximum matching cut: H-freeness, bounded diameter, bounded radius. *to appear in Theoretical Computer Science*, 2024.

[52] Ali Momeni. Finding cuts of bounded degree. Master's thesis, ENS de Lyon, 2023.

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

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

[55] Thomas J. Schaefer. The complexity of satisfiability problems. *Proc. STOC 1978*, pages 216–226, 1978.

[56] Tatjana Schmidt. *Computational complexity of SAT, XSAT and NAE-SAT for linear and mixed Horn CNF formulas.* PhD thesis, Universität zu Köln, 2010.

[57] Harold N. Shapiro. Introduction to the Theory of Numbers. *Dover Publications*, 2008.

[58] Siani Smith. Private communication, 2024.

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

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

Felicia Lucke

Curriculum Vitae

felicia.lucke@unifr.de
https://orcid.org/0000-0002-9860-2928

Education

since 2020 **Ph.D., Computer Science**, *Université de Fribourg*, Switzerland
2018 - 2020 **Master of Science, Mathematics**, *Ulm University*, Germany
2014 - 2018 **Bachelor of Science, Mathematics**, *Ulm University*, Germany
2014 **Abitur**, *Landesgymnasium für Hochbegabte*, Schwäbisch Gmünd, Germany

Teaching Experience

since 2021 **Mathématiques I et II**, *Teaching assistant (French)*, Université de Fribourg
2020 **Decision Support I**, *Teaching assistant (French)*, Université de Fribourg
2015 - 2020 **Analysis 1 and 2, Linear Algebra 1, Optimisation 1**, *Tutor*, Ulm University, Germany

Supervision of students

04.-06.2024 **Research Internship**, *Minimum Matching Cut*, Joseph Marchand from ENS Paris-Saclay, Université de Fribourg, Switzerland
2023 **Master Thesis**, *Dominating Set Transversals on Interval Graphs*, Lukas Seeholzer, Université de Fribourg, Switzerland

Awards

2023 **Best Paper Award** for "Matching Cuts in Graphs of high Girth and H-Free Graphs", 34th International Symposium on Algorithms and Computation
2019 Bronze Medal, Northwestern European Regional Contest
2018 Gold Medal, German Collegiate Programming Contest

Organization

2023 49th International Workshop on Graph-Theoretic Concepts in Computer Science, Fribourg, Switzerland
since 2022 German Collegiate Programming Contest, annual subregional of the International Collegiate Programming Contest
since 2022 Winter Contest, annual German programming competition

Short Term Visits

05.2024 Carl Feghali, ENS Lyon, Lyon, France
02.2022 Daniel Paulusma, Durham University, Durham, United Kingdom
02.2022 Christophe Picouleau, CNAM, Paris, France

Conferences, Workshops, Presentations

2024 **GROW**, *11th Workshop on Graph Classes, Optimization, and Width Parameters*, Cottbus, Germany
Presentation: Matching Cut and Variants in H-Free Graphs

2024 **WG**, *50th International Workshop on Graph-Theoretic Concepts in Computer Science*, Gozd Martuljek, Slovenia
Presentation: Finding d-Cuts in Graphs of Bounded Diameter, Graphs of Bounded Radius and H-Free Graphs

2024 **EPIT**, *Spring School: Graphs and Algorithms: Conjectures*, Aussois, France

2023 **MFCS**, *48th International Symposium on Mathematical Foundations of Computer Science*, Bordeaux, France
Presentation: Dichotomies for Maximum Matching Cut: H-Freeness, Bounded Diameter, Bounded Radius

2023 **GO XI**, *International Colloquium on Graphs and Optimization*, Spa, Belgium
Presentation: Matching Cuts in H-free graphs

2023 **WG**, *49th International Workshop on Graph-Theoretic Concepts in Computer Science*, Fribourg, Switzerland

2023 **ACiD Seminar**, Durham, United Kingdom
Presentation: Transversals of Maximum Independent Sets

2022 **ISAAC**, *33rd International Symposium on Algorithms and Computation*, Seoul, Korea
Presentation: Matching Cuts in H-free graphs

2022 **ICGT**, *11th International Colloquium on Graph Theory and Combinatorics*, Montpellier, France
Presentation: Transversals of Maximum Independent Sets

2022 **IWOCA**, *33rd International Workshop on Combinatorial Algorithms*, Trier, Germany

2021 **Summer School**, *Data Science, Optimization and Operations Research*, Zinal, Switzerland

Publications

Journal

- F. Lucke, D. Paulusma, B. Ries, **Dichotomies for Maximum Matching Cut: H-Freeness, Bounded Diameter, Bounded Radius**, *Theoretical Computer Science 1017, 2024*.
- F. Lucke, B. Ries, **On Blockers and Transversals of Maximum Independent Sets in Co-Comparability Graphs**, *Discrete Applied Mathematics, 2024*.
- V. B. Le, F. Lucke, D. Paulusma, B. Ries, **Maximising Matching Cuts**, *to appear in Encyclopedia of Optimization, 2024*
- F. Lucke, F. Mann, **Reducing Graph Parameters by Contractions and Deletions**, *Algorithmica, 2024*.
- F. Lucke, D. Paulusma, B. Ries, **Finding Matching Cuts in H-Free Graphs**, *Algorithmica, 2023*.
- F. Lucke, D. Paulusma, B. Ries, **On The Complexity of Matching Cut for Graphs of Bounded Radius and H-Free Graphs**, *Theoretical Computer Science 936, 33-42, 2022*.

Conference

- F. Lucke, A. Momeni, D. Paulusma, S. Smith, **Finding d-Cuts in Graphs of Bounded Diameter, Graphs of Bounded Radius and H-Free Graphs**, *WG 2024*
- C. Feghali, F. Lucke, D. Paulusma, B. Ries, **Matching Cuts in Graphs of high Girth and H-Free Graphs,** *Leibniz International Proceedings in Informatics 283, 31:1-31:16, ISAAC 2023*.
- F. Lucke, D. Paulusma, B. Ries, **Dichotomies for Maximum Matching Cut: H-Freeness, Bounded Diameter, Bounded Radius,** *Leibniz International Proceedings in Informatics 272, 64:1 - 63:15, MFCS 2023*.

- F. Lucke, D. Paulusma, B. Ries, **Finding Matching Cuts in H-Free Graphs**, *Leibniz International Proceedings in Informatics 248, 22:1 - 22:16, ISAAC 2022*.
- F. Lucke, F. Mann, **Using Edge Contractions and Vertex Deletions to Reduce the Independence Number and the Clique Number**, *LNCS 13270, 412-424, IWOCA 2022*.

Submitted

- C. Feghali, F. Lucke, D. Paulusma, B. Ries, **Matching Cuts in Graphs of High Girth and H-Free Graphs**, *2023 (journal version)*.