Finding d-Cuts in Claw-Free Graphs

Jungho Ahn
Department of Computer Engineering, Inha University, Incheon, South Korea

Tala Eagling-Vose
Department of Computer Science, Durham University, Durham, UK

Felicia Lucke
LIP, ENS Lyon, Lyon, France

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

Siani Smith
Department of Computer Science, Loughborough University, Loughborough, UK

Abstract

The Matching Cut problem is to decide if the vertex set of a connected graph can be partitioned into two non-empty sets B and R such that the edges between B and R form a matching, that is, every vertex in B has at most one neighbour in R, and vice versa. If for some integer $d \geq 1$, we allow every vertex in B to have at most d neighbours in R, and vice versa, we obtain the more general problem d-Cut. It is known that d-Cut is NP-complete for every $d \geq 1$. However, for claw-free graphs, it is only known that d-Cut is polynomial-time solvable for $d = 1$ and NP-complete for $d \geq 3$. We resolve the missing case $d = 2$ by proving NP-completeness. This follows from our more general study, in which we also bound the maximum degree. That is, we prove that for every $d \geq 2$, d-Cut, restricted to claw-free graphs of maximum degree p, is constant-time solvable if $p \leq 2d + 1$ and NP-complete if $p \geq 2d + 3$. Moreover, in the former case, we can find a d-cut in linear time. We also show how our positive results for claw-free graphs can be generalized to $S_{1^t,\ell}$-free graphs where $S_{1^t,\ell}$ is the graph obtained from a star on $t + 2$ vertices by subdividing one of its edges exactly ℓ times.

2012 ACM Subject Classification Mathematics of computing → Graph theory; Theory of computation → Graph algorithms analysis; Theory of computation → Problems, reductions and completeness

Keywords and phrases matching cut, d-cut, claw-free, maximum degree

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2025.4

Related Version *Full Version*: https://arxiv.org/abs/2505.17993

Funding *Jungho Ahn*: supported by Leverhulme Trust Research Project Grant RPG-2024-182 and INHA UNIVERSITY Research Grant.
Felicia Lucke: supported by EPSRC Grant EP/X01357X/1 and SNSF Postdoc Mobility Grant 230578.
Daniël Paulusma: supported by Leverhulme Trust Research Project Grant RPG-2024-182 and EPSRC Grant EP/X01357X/1.

Acknowledgements The authors thank Ali Momeni and Hyunwoo Lee for providing some helpful insights about the proofs of Lemma 9 and Theorem 10, respectively.

1 Introduction

In this paper, we determine new complexity results for finding d-cuts, which form a natural generalization of matching cuts. The latter form a well-studied notion that combines two classic concepts in graph theory: matchings and edge cuts. Namely, a *matching cut* in a

© Jungho Ahn, Tala Eagling-Vose, Felicia Lucke, Daniël Paulusma, and Siani Smith;
licensed under Creative Commons License CC-BY 4.0

Figure 1 Left: a graph with a matching cut (1-cut). Right: a graph with a 3-cut but no d-cut for $d \leq 2$. Figure taken from [21].

connected graph $G = (V, E)$ is a set of edges $M \subseteq E$ that is both a *matching* (i.e., no two edges in M share an end-vertex) and an *edge cut* (i.e., V can be partitioned into two non-empty sets B and R such that M consists of all the edges between B and R). The Matching Cut problem is to decide if a connected graph has a matching cut.

Already in 1984, Chvátal [9] proved that Matching Cut is NP-complete even for $K_{1,4}$-free graphs of maximum degree 4. Here, the graph $K_{1,\ell}$ denotes the *star* on $\ell + 1$ vertices, that is, the graph with vertices $u, v_1, \ldots, v_\ell$ and edges uv_i, for $i = 1, \ldots, \ell$, and a graph is *H-free* if it does not contain H as an *induced* subgraph. In contrast, Chvátal [9] also showed that Matching Cut is polynomial-time solvable for graphs of maximum degree at most 3. In 2009, Bonsma [6] proved the same for $K_{1,3}$-free graphs (of arbitrary degree), which are also known as *claw-free* graphs. Since then a growing sequence of papers appeared in which the computational complexity of Matching Cut for various special graph classes was determined, often in a systematic way (e.g., [8, 11, 18, 22, 23]) and also from a parameterized complexity perspective (e.g. [1, 12, 13, 16, 17]) and for many closely related problem variants, such as the problems of finding perfect matching cuts [5, 15, 19], maximum matching cuts [24], minimum matching cuts [20], matching multicuts [14] and disconnected perfect matchings [7]. We refer to Section 5 for a summary of the complexity results for Matching Cut restricted to H-free graphs as part of a summary for a more general notion, namely d-cuts, the topic of our paper. 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 non-empty sets B and R such that:

- the set M is the set of all edges between B and R; and
- every vertex in B has at most d neighbours in R, and vice versa.

See Figure 1 for some examples. We note that a 1-cut is a matching cut and vice versa. For a fixed integer $d \geq 1$, the d-Cut problem is to decide if a connected graph G has a d-cut. Hence, Matching Cut and 1-Cut are the same problems.

The d-Cut problem was introduced by Gomes and Sau [13]. Apart from several parameterized complexity results (see also [2]), they showed the following two results. The first result is a structural result (shown for $d = 1$ in [9, 25]), which immediately implies the first statement of the second result (shown for $d = 1$ in [9]).

▶ **Theorem 1** (Gomes and Sau [13]). *For $d \geq 1$, every graph $G = (V, E)$ with maximum degree $\Delta(G) \leq d + 2$ and $|V| > 7$ has a d-cut, which can be found in polynomial time.*

▶ **Theorem 2** (Gomes and Sau [13]). *For $d \geq 1$, d-Cut is constant-time solvable for graphs with maximum degree $\Delta \leq d + 2$ but NP-complete for $(2d + 2)$-regular graphs (that is, graphs in which every vertex has degree $2d + 2$).*

For $d \geq 1$, the results in Theorem 2 are currently the best results for d-Cut of bounded maximum degree, so for $d \geq 2$ we have a complexity gap. Gomes and Sau [13] argued that it is unlikely that the maximum degree bound of $2d + 2$ in Theorem 2 can be lowered, as that would disprove a conjecture about the existence of so-called internal partitions of r-regular graphs for odd r; see [3].

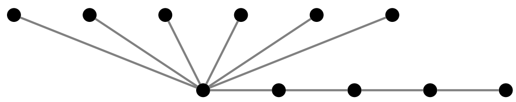

Figure 2 The graph $S_{1^6,4}$.

In particular, Theorem 2 shows that 2-Cut is constant-time solvable for graphs of maximum degree $\Delta \leq 4$ and NP-complete for graphs of maximum degree $\Delta = 6$. Gomes and Sau [13] showed that all maximum degree-5 graphs on 18 vertices without a 2-cut have some specific structure: they are 5-regular or have exactly two vertices of degree 4, which are adjacent. Interestingly, they mentioned in [13] that they were unable to find a maximum degree-5 graph on more than 18 vertices without a 2-cut.

The d-Cut problem has also been studied for H-free graphs. As mentioned, we summarize these results in Section 5. For now, we focus on the case $H = K_{1,3}$ for the following reason. It is known that for all $d \geq 3$, d-Cut is NP-complete for claw-free graphs. This was shown in [21] (the gadget constructed in the proof of Theorem 2 is not claw-free). The result for $d = 3$ contrasts the case $d = 1$ due to the aforementioned result of Bonsma [6] that 1-Cut (Matching Cut) is polynomial-time solvable on claw-free graphs and leaves open exactly the case $d = 2$. The authors of [21] could prove that 2-Cut is NP-complete for $K_{1,4}$-free graphs and asked about the case $d = 2$ in the following open problem:

▶ **Open Problem 1** (Lucke et al. [21]). *What is the complexity of* 2-Cut *for claw-free graphs?*

Our Results

Inspired by Theorems 1 and 2 of Gomes and Sau [13] and motivated by Open Problem 1, we consider *claw-free* graphs of bounded maximum degree. In Sections 3 and 4, respectively, we show the following two results for $d \geq 2$, the second of which solves Open Problem 1.

▶ **Theorem 3.** *Let $d \geq 2$. Every* claw-free *graph $G = (V, E)$ with maximum degree $\Delta(G) \leq 2d + 1$ and $|V| > 4d^2(2d + 1)$ has a d-cut, which can be found in linear time. Moreover, there exist arbitrarily large claw-free $(2d + 2)$-regular graphs with no d-cut.*

▶ **Theorem 4.** *For $d \geq 2$, d-*Cut *is constant-time solvable for claw-free graphs with maximum degree $\Delta \leq 2d + 1$ but* NP*-complete for claw-free graphs with maximum degree $\Delta = 2d + 3$.*

The first statement of Theorem 3 is the analogue of Theorem 1 for claw-free graphs (for which we can find the d-cut even in linear time). We show how this statement follows from a corresponding stronger statement on $S_{1^t,\ell}$-free graphs, where $S_{1^t,\ell}$ is the graph obtained from a star on $t + 2$ vertices by subdividing one of its edges exactly ℓ times, see Figure 2 for the case where $t = 6$ and $\ell = 4$. The second statement of Theorem 3 not only shows that the bound in the first statement is best possible, but also relates to Theorem 2: it shows that for every $d \geq 2$, there exists an infinite number of $(2d + 2)$-regular no-instances of d-Cut that are also claw-free.

The first statement of Theorem 3 immediately implies the constant-time part of Theorem 4, as d is a constant. We use the graph in the second statement of Theorem 3 as part of our hardness gadget in the NP-completeness part of Theorem 4.

Theorem 4 shows in particular that 2-Cut is constant-time solvable for claw-free graphs of maximum degree $\Delta \leq 5$ but NP-complete for claw-free graphs of maximum degree $\Delta = 7$, which solves Open Problem 1. Moreover, the gadget used in the reduction for d-Cut $(d \geq 3)$

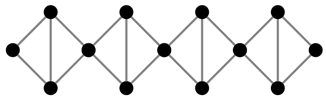

Figure 3 A chain of diamonds has no 1-cut (the diamond is the graph $K_4 - e$).

for claw-free graphs in [21] contains vertices of arbitrarily large degree. Theorem 4 strengthens this result by adding a maximum degree bound that is only one higher than the maximum degree bound in Theorem 2.

Note that d is at least 2 in both Theorems 3 and 4, which can be explained as follows. We first recall that for $d = 1$, Bonsma [6] proved that d-Cut is polynomial-time solvable even for claw-free graphs with arbitrarily large maximum degree Δ. Second, for $d = 1$, Theorem 3 would not hold: take, for example, a chain of graphs isomorphic to $K_p - e$ (complete graph minus an edge) which we obtain from k copies of $K_p - e$, for $k \geq 1$, by identifying a pair of degree $(p-2)$ vertices between consecutive copies. That is, where x_i^1, x_i^2 is the pair of degree $(p-2)$ vertices of the ith copy, we identify vertices x_i^2 and x_{i+1}^1, for all $1 \leq i \leq k-1$. The resulting chain is an arbitrarily large claw-free graph of arbitrarily large maximum degree $2p-4$ but with no 1-cut, see Figure 3 for the case where $p = 4$.

In Section 5, we give some directions for future work and show the consequence of Theorem 4 for the state-of-art summary of the complexity of d-Cut for H-free graphs.

2 Preliminaries

In this paper, we only consider finite, undirected graphs without multiple edges and self-loops. For every integer $n \geq 1$, let $[n] := \{1, \ldots, n\}$.

Let $G = (V, E)$ be a graph and let v be a vertex of G. We denote by $N_G(v) = \{u \in V \mid uv \in E\}$ the *(open) neighbourhood* of v and by $N_G[v] = N_G(v) \cup \{v\}$ the *closed neighbourhood* of v. The *degree* of v is the size of $N_G(v)$, and we denote by $\Delta(G)$ the *maximum degree* of G. For a set $S \subseteq V$, we denote by $\delta(S)$ the set of edges of G with exactly one end-vertex in S.

A *complete* graph is a graph whose vertex set is a *clique*, which is a set of pairwise adjacent vertices. For $r \geq 1$, we let K_r denote the complete graph on r vertices. A vertex set $I \subseteq V$ is an *independent set* of G if no two vertices of I are adjacent in G. In addition, G is *bipartite* if we can partition the vertex set into (possibly empty) sets A and B such that $A \cup B = V$ and A and B are each an independent set.

The *chromatic number* of a graph $G = (V, E)$ is the smallest integer k such that G has a k*-colouring*, which is a mapping $c : V \to \{1, \ldots, k\}$ with $c(u) \neq c(v)$ for every $uv \in E$. The set of all vertices that are mapped to the same colour $i \in \{1, \ldots, k\}$ is called a *colour class* of c. For $k \geq 0$, a graph G is k*-degenerate* if every subgraph of G contains a vertex of degree at most k. It is well known and readily seen that every k-degenerate graph is $(k+1)$-colourable. The *degeneracy* of G is the smallest integer k such that G is k-degenerate. The following observation holds (we give a proof for completeness).

▶ **Observation 5.** *Let $t \geq 2$ and let $G = (V, E)$ be a graph. If G has no independent set of size t, then G has a subgraph of minimum degree at least $\frac{|V|}{t-1} - 1$.*

Proof. Assume G has no independent set of size t. Let k be the degeneracy of G, so G has a subgraph of minimum degree at least k. Consider a $(k+1)$-colouring c of G. As every colour class of c is an independent set, we have $k + 1 \geq \frac{|V|}{t-1}$ and thus $k \geq \frac{|V|}{t-1} - 1$. ◀

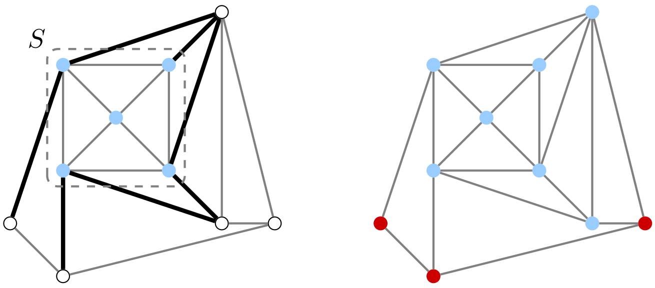

Figure 4 An example of the procedure in the proof of Lemma 9 if $d = 2$. Left: every vertex in S has at most 2 neighbours outside of S. The edges in $\delta(S)$ are highlighted in black. Right: The red-blue d-colouring we obtain from the procedure.

Let $d \geq 1$ be an integer. A *red-blue colouring* of G gives each vertex of G either colour red or blue. A *red-blue d-colouring* of G is a red-blue colouring where every vertex has at most d neighbours of the other colour and both colours, red and blue, are used at least once. See Figure 1 for an example of a red-blue 1-colouring (left figure) and a red-blue 3-colouring (right figure). In our proofs, we use the following well-known observation; see e.g. [21].

▶ **Observation 6.** *For every integer $d \geq 1$, G has a red-blue d-colouring with sets B and R if and only if the set of edges between B and R form a d-cut of G.*

Let G be a graph with a red-blue d-colouring for some $d \geq 1$. We say that a set $S \subseteq V$ is *monochromatic* if all vertices in S have the same colour.

We will also use the following observations, which can be readily seen.

▶ **Observation 7.** *Let $d \geq 1$. Let $G = (V, E)$ be a graph and let $S \subseteq V$ with $|S| \geq d+1$. For every red-blue colouring of G, in which S is monochromatic, each vertex in $V \setminus S$ with at least $d+1$ neighbours in S has the same colour as the vertices of S.*

▶ **Observation 8.** *Let $d \geq 1$. Let G be a graph with a clique K on at least $2d+1$ vertices. Then K is monochromatic in every red-blue d-colouring of G.*

3 The Proof of Theorem 3

In this section, we prove Theorem 3, which states that for all $d \geq 2$, every claw-free graph $G = (V, E)$ with maximum degree $\Delta(G) \leq 2d + 1$ and $|V| > 4d^2(2d + 1)$ has a d-cut, which we can find in linear time, and moreover, that there exist arbitrarily large claw-free $(2d + 2)$-regular graphs with no d-cut.

We start with a lemma. In its proof, we extend the argument that Chvátal [9] used to observe that 1-Cut is constant-time solvable for graphs of maximum degree at most 3.

▶ **Lemma 9.** *Let $d \geq 1$. Let $G = (V, E)$ be a graph with $\Delta(G) \leq 2d + 1$, and let S be a non-empty set of vertices with $|S| + |\delta(S)| < |V|$. If each vertex in S is incident to at most d edges in $\delta(S)$, then G has a d-cut, which can be found in linear time.*

Proof. Assume that each vertex in S is incident to at most d edges in $\delta(S)$. We will now construct a red-blue d-colouring of G, which, by Observation 6, gives that G has a d-cut. We first colour every vertex in S blue and recursively colour an uncoloured vertex blue if it has at least $d + 1$ blue neighbours. Note that this takes $\mathcal{O}(|V|)$ time, as $|E| \leq (2d + 1)|V|/2$. We colour all remaining vertices of G red. See also Figure 4.

Let R and B be the sets of red and blue vertices, respectively. Note that $S \subseteq B$. By construction, every vertex in R has at most d neighbours in B. As we assume that each vertex in S is incident to at most d edges in $\delta(S)$, each vertex in S has at most d neighbours in R. Since $\Delta(G) \leq 2d+1$ and each vertex in $B \setminus S$ has at least $d+1$ neighbours in B, each vertex in $B \setminus S$ has at most d neighbours in R. Hence, every vertex in B has at most d neighbours in R. It remains to show that $R \neq \emptyset$, or equivalently, that $B \neq V$, which we will do below.

Let $v_1, \ldots, v_b$ denote the vertices in $B \setminus S$ in the order they are coloured blue. For an arbitrary $i \in [b]$, let $B' := S \cup \{v_1, \ldots, v_i\}$. Recall that by construction, each vertex $v \in B' \setminus S$ has at least $d+1$ neighbours in B' and at most d neighbours not in B'. So when we colour v blue, that is, add v to B, we lose at least $d+1$ edges with exactly one blue end-vertex and gain at most d new edges with exactly one blue end-vertex. In other words, $|\delta(B' \setminus \{v\})| \geq |\delta(B')| + 1$. This implies $|\delta(S)| \geq |\delta(B)| + (|B| - |S|)$, and thus $|\delta(S)| + |S| \geq |\delta(B)| + |B|$. We use this and our assumption that $|V| > |\delta(S)| + |S|$ to deduce that

$$|V| > |\delta(S)| + |S| \geq |\delta(B)| + |B| \geq |B|,$$

and thus $B \neq V$. Hence, we obtained a red-blue d-colouring of G. ◀

To prove the first part of Theorem 3, we prove a stronger statement in Theorem 10 and afterwards we show how Theorem 10 implies the first part of Theorem 3. For every positive integer t and ℓ, we recall that $S_{1^t,\ell}$ is the graph obtained from $K_{1,t+1}$ by replacing one edge with a path of length ℓ, see Figure 2. Note that $S_{1^t,\ell}$ contains exactly $t+\ell$ edges. In addition, $S_{1^t,1}$ is the graph $K_{1,t+1}$.

▶ **Theorem 10.** *For $d \geq 2$, $t \geq 2$ and $\ell \geq 1$, every $S_{1^t,\ell}$-free graph $G = (V,E)$ with either maximum degree $\Delta = 2$, or $3 \leq \Delta \leq \frac{t}{t-1}d + \frac{1}{t-1}$ and $|V| > (d+1)\left(\frac{\Delta(\Delta-1)^{\ell+1}-2}{\Delta-2}\right)$ has a d-cut, which can be found in linear time.*

Proof. Let $d \geq 2$, $t \geq 2$ and $\ell \geq 1$. Let G be an $S_{1^t,\ell}$-free graph. If $\Delta = 2$, then G has a d-cut: take all edges between any vertex of G and its neighbours. From now on, assume $3 \leq \Delta \leq \frac{t}{t-1}d + \frac{1}{t-1}$ and and $|V| > (d+1)\left(\frac{\Delta(\Delta-1)^{\ell+1}-2}{\Delta-2}\right)$.

We choose an arbitrary vertex v of G and let $S_0 := \{v\}$. For each $i \in [\ell+1]$, we denote by S_i the set of vertices which are at distance exactly i from v in G. Note that $|S_1| \leq \Delta$ and for each $i \in [\ell]$, $|S_{i+1}| \leq (\Delta-1)|S_i|$. Thus,

$$\sum_{i=0}^{\ell+1} |S_i| \leq \frac{\Delta((\Delta-1)^{\ell+1}-1)}{\Delta-2} + 1 = \frac{\Delta(\Delta-1)^{\ell+1}-2}{\Delta-2}.$$

Let U be the set of vertices in S_ℓ which are incident to at least $d+1$ edges in $\delta(S_{\ell-1} \cup S_\ell)$. For each $u \in U$, we denote by N_u the set of neighbours of u not in $S_{\ell-1} \cup S_\ell$ and by G_u the subgraph of G induced on N_u.

We first show that N_u, for $u \in U$, has no independent set of size t. Suppose not. By the definition of S_ℓ, there is an induced path of length ℓ between v and u. Together with this path and the independent set in N_u, we can find $S_{1^t,\ell}$ as an induced subgraph, a contradiction. Thus, N_u has no independent set of size t.

For each $u \in U$, since G_u has no independent set of size t, by Observation 5, G_u has a subgraph H_u whose minimum degree is at least

$$\frac{|V(G_u)|}{t-1} - 1 \geq \frac{d+1}{t-1} - 1.$$

We remark that H_u can be found in constant time as $|N_u| \leq \Delta - 1$, which is a constant. Now, we let

$$S := \left(\bigcup_{i=0}^{\ell} S_i \right) \cup \left(\bigcup_{u \in U} V(H_u) \right).$$

Note that S can be found in constant time as $|U| \leq |S_\ell| \leq \Delta(\Delta-1)^{\ell-1}$, which is a constant.

We are going to apply Lemma 9 to S. Since $t \geq 2$, we have $t \leq 2(t-1)$, and therefore

$$\Delta \leq \frac{t}{t-1}d + \frac{1}{t-1} \leq 2d + 1.$$

To apply Lemma 9 to S, we first show that every vertex in S is incident to at most d edges in $\delta(S)$. By construction, this holds for every vertex in $S \setminus \bigcup_{u \in U}(N_u \cup \{u\})$. Let u be a vertex in U and let w be a vertex in $N_u \cup \{u\}$. Note that u and w might be the same. Since the minimum degree of H_u is at least

$$\frac{d+1}{t-1} - 1$$

and u is adjacent to every vertex in N_u, there are at least

$$\frac{d+1}{t-1}$$

neighbours of w in S. Then w is incident to at most

$$\Delta - \frac{d+1}{t-1} \leq \frac{t}{t-1}d + \frac{1}{t-1} - \frac{d+1}{t-1} = d$$

edges in $\delta(S)$. That is, w is incident to at most d edges in $\delta(S)$. Hence, every vertex in S is incident to at most d edges in $\delta(S)$.

We now show that $|V| > |S| + |\delta(S)|$. Since $S \subseteq \bigcup_{i=0}^{\ell+1} S_i$, we have

$$|S| \leq \sum_{i=0}^{\ell+1} |S_i| \leq \frac{\Delta(\Delta-1)^{\ell+1} - 2}{\Delta - 2}.$$

Since every vertex in S is incident to at most d edges in $\delta(S)$, we have $|\delta(S)| \leq d|S|$. Thus,

$$|V| > (d+1)\left(\frac{\Delta(\Delta-1)^{\ell+1} - 2}{\Delta - 2}\right) \geq (d+1)|S| = |S| + d|S| \geq |S| + |\delta(S)|.$$

Hence, by Lemma 9, we can find a d-cut of G in linear time. ◀

We now show that Theorem 10 implies Theorem 3, which we recall below.

▶ **Theorem 3** (first part, restated). *For $d \geq 2$, every claw-free graph $G = (V, E)$ with maximum degree $\Delta(G) \leq 2d+1$ and $|V| > 4d^2(2d+1)$ has a d-cut, which can be found in linear time.*

Proof. Let G be a claw-free graph. If $\Delta(G) = 2$, then we can apply Theorem 10 directly. Assume $\Delta(G) \geq 3$. Theorem 10 with $t = 2$ and $\ell = 1$ implies the first part of Theorem 3. In order to see this, let G be a claw-free graph with maximum degree $\Delta(G) \leq 2d+1$ and $|V| > 4d^2(2d+1)$. As $2d+1 = \frac{t}{t-1}d + \frac{1}{t-1}$ if $t = 2$, we obtain $\Delta(G) \leq \frac{t}{t-1}d + \frac{1}{t-1}$. We also observe:

$$|V| > 4d^2(2d+1) = (2d-1)\left(\frac{(2d+1)(2d)^2}{2d-1}\right) > (d+1)\left(\frac{\Delta(\Delta-1)^2 - 2}{\Delta - 2}\right)$$

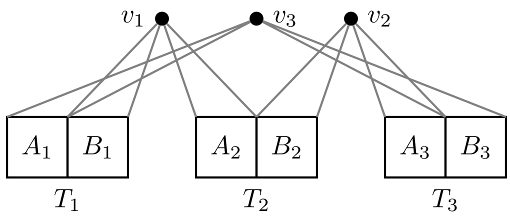

Figure 5 The graph constructed in Theorem 11 for $k = 3$. Note that for each $i \in [3]$, $A_i \cup B_i$ is a clique and v_i is complete to $B_i \cup A_{i+1}$, where $A_4 := A_1$.

where the last inequality holds from the fact that $2d - 1 \geq d + 1$ and the function $x(x-1)^2/(x-2)$ is increasing for $x \geq 3$. Hence, the conditions in Theorem 10 are satisfied, and we may conclude that G has a d-cut, which can be found in linear time. ◀

We now prove the second part of Theorem 3, which shows that the bound given in the first part of Theorem 3 is best possible. We slightly reformulate the statement as follows.

▶ **Theorem 11.** *For every integer $d \geq 2$, $k \geq 2$, and $r \geq 2d + 2$, there exists a claw-free r-regular graph on $(r+1)k$ vertices that has no d-cut.*

Proof. Let $T_1, \ldots, T_k$ be pairwise disjoint cliques of size r. For each $i \in [k]$, let $\{A_i, B_i\}$ be a partition of T_i such that $|A_i| = d + 1$. Note that

$$|B_i| = r - (d+1) \geq d + 1.$$

Let H be the graph obtained from $T_1 \cup \cdots \cup T_k$ by adding k new vertices $v_1, \ldots, v_k$ such that for each $i \in [k]$, $N_H(v_i)$ is equal to $B_i \cup A_{i+1}$ where $A_{k+1} := A_1$, see Figure 5. Note that H has $rk + k = (r+1)k$ vertices.

We first show that H is claw-free and r-regular. Let i be an arbitrary integer in $[k]$. Since the neighbourhood of v_i is the disjoint union of two cliques B_i and A_{i+1}, H has no claw centred at v_i. Note further that v_i has degree r. Consider now a vertex v in T_i. The neighbourhood of v consists of all other vertices in T_i and some vertex v_j, where $j \in \{i-1, i, k\}$. Hence, v has degree $(r-1) + 1 = r$ and since T_i is a clique, H has no claw centred at v. We conclude that H is claw-free and r-regular.

It remains to show that H has no d-cut. Note that in any red-blue d-colouring of H, T_i, for $i \in [k]$, is monochromatic by Observation 8, since it is a clique of size at least $2d + 2$. We claim that

$$T := \bigcup_{i=1}^{k} T_i$$

is monochromatic in every red-blue d-colouring of H. Suppose for a contradiction that H has a red-blue d-colouring where T is not monochromatic. Then there exists $j \in [k]$ such that T_j and T_{j+1} have different colours, where $T_{k+1} := T_1$. Without loss of generality, assume that T_j is coloured red. Consequently, T_{j+1} is coloured blue. Since v_j has at least $d + 1$ red neighbours in B_j, it is coloured red. However, v_j has $d + 1$ blue neighbours in A_{j+1}, contradicting the fact that we considered a red-blue d-colouring. Hence, if H has a red-blue d-colouring, then T is monochromatic.

Thus, if H has a red-blue d-colouring, then T is monochromatic and hence, v_i, for $i \in [k]$, is coloured the same as T since v_i has at least $2d + 2$ neighbours in T. Therefore, there is no red-blue d-colouring of H. By Observation 6, H has no d-cut. ◀

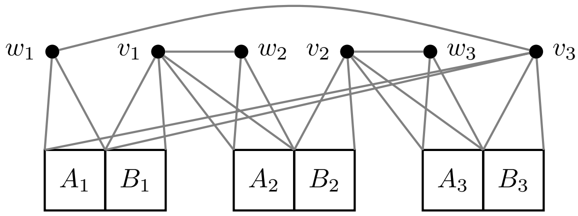

Figure 6 The graph illustrating $H_{d,k,r}$ for $k = 3$. Note that for each $i \in [3]$, $A_i \cup B_i$ is a clique, v_i is complete to $B_i \cup A_{i+1}$, where $A_4 := A_1$, and w_i is complete to A_i. If we remove w_1, w_2, and w_3, we obtain the graph in Figure 5.

4 The Proof of Theorem 4

In this section we prove Theorem 4. We first recall that the first statement of Theorem 3 immediately implies the constant-time part of Theorem 4, as d is a constant. Hence, it remains to show the NP-completeness part of Theorem 4, that is, that for every integer $d \geq 2$, d-Cut is NP-complete for claw-free graphs of maximum degree $2d + 3$. To do this, we will reduce from NAE 3-Sat 0-1, which is a variant of Not-All-Equal 3-Satisfiability (NAE 3-Sat). An instance of NAE 3-Sat consists of a set of clauses $C_1, \ldots, C_m$, each containing three variables. The problem is to decide whether there is an assignment of the variables such that every clause contains both a true and a false literal. Such an assignment is called an *NAE-satisfying* assignment.

An instance of the problem we will reduce from NAE 3-Sat 0-1 is an instance of NAE 3-Sat that is satisfied both by the assignment where every variable is true and the assignment where every variable is false. In other words, given an instance of NAE 3-Sat 0-1, every clause contains a positive literal and a negative literal. Additionally, since the clause $(\bar{x} \vee \bar{y} \vee z)$ has the same set of NAE-satisfying assignments as the clause $(x \vee y \vee \bar{z})$, we may assume that each clause contains exactly one negative literal.

The problem NAE 3-Sat 0-1 is to decide whether an instance has a third NAE-satisfying assignment. That is, we must decide whether there exists an assignment of variables such that at least one variable is true, at least one variable is false, and each clause contains both a true and a false literal. The next result was shown in [10].

▶ **Proposition 12** (Eagling-Vose et al. [10]). NAE 3-Sat 0-1 *is* NP*-complete.*

We remark that it is possible to combine the reduction from Matching Cut to NAE 3-Sat 0-1 given in [10] with the reduction given here from the latter problem to d-Cut to obtain a reduction directly from Matching Cut to d-Cut. However, we separate the two reductions for ease of explanation.

For the reduction in the proof of our NP-hardness result, we use a lemma that is based on the class of graphs from Theorem 11. For all integers $d \geq 2$, $k \geq 2$, and $r \geq 2d + 2$, let $H_{d,k,r}$ be the graph obtained from the graph in Theorem 11, with respect to the same integers, by adding k new vertices $w_1, \ldots, w_k$ such that for each $i \in [k]$, the neighbourhood of w_i in $H_{d,k,r}$ is equal to $A_i \cup \{v_{i-1}\}$ where $v_0 := v_k$. See also Figure 6.

▶ **Lemma 13.** *For all integers $d \geq 2$, $k \geq 2$, and $r \geq 2d + 2$, the graph $H_{d,k,r}$ is a claw-free graph with maximum degree $r + 1$ on $(r + 2)k$ vertices which has no d-cut. In particular, each of $w_1, \ldots, w_k$ has degree $d + 2$, and the other vertices have degree at least r.*

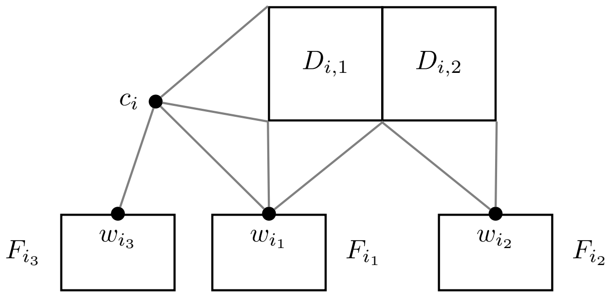

Figure 7 The clause gadget consisting of the vertex c_i and the clique $D_{i,1} \cup D_{i,2}$ representing the clause $C_i = (\overline{x}_{i_1} \vee x_{i_2} \vee x_{i_3})$, together with the corresponding variable vertices.

Proof. Let H be the graph in Theorem 11, with respect to the same integers, and let $H' := H_{d,k,r}$. Note that

$$|V(H')| = |V(H)| + k = (r+1)k + k = (r+2)k.$$

Since H is r-regular and $w_1, \dots, w_k$ have pairwise disjoint neighbourhoods, each of $w_1, \dots, w_k$ has degree $d+2$, and the other vertices have degree either r or $r+1$.

We show that H' is claw-free. Suppose that H' has a claw K centred at a vertex c. Since H is claw-free, K contains w_i for some $i \in [k]$. As $N_{H'}(w_i)$ is a clique, $c \neq w_i$. So $N_{H'}(c)$ is the union of two cliques, contradicting that K is a claw. Hence, H' is claw-free.

It remains to show that H' has no d-cut. By Observation 6, H' has a d-cut if and only if H' has a red-blue d-colouring. By Theorem 11, we know that $V(H)$ is monochromatic in every red-blue d-colouring. Further, w_i, for $i \in [k]$, has $d+2$ neighbours in H. Thus, by Observation 7, w_i is coloured the same as $V(H)$. This implies that $V(H')$ is monochromatic in every red-blue d-colouring and hence H' has no red-blue d-colouring. It follows that H' has no d-cut. ◀

We are now ready to prove the remaining part of Theorem 4, which we restate below.

▶ **Theorem 4** (NP-completeness part, restated). *For $d \geq 2$, d-Cut is NP-complete for claw-free graphs with maximum degree $\Delta = 2d+3$.*

Proof. We reduce from NAE 3-Sat 0-1. Let ϕ be an instance of NAE 3-Sat 0-1 with variables $x_1, \dots, x_n$ and clauses $C_1, \dots, C_m$. In the following, we construct an instance G of d-Cut which is claw-free and has maximum degree $2d+3$. For each $\ell \in [n]$, let k_ℓ be the number of occurrences of x_ℓ in ϕ and let F_ℓ be a copy of $H_{d,k_\ell,2d+2}$. We say that a vertex of F_ℓ is *free* if it has degree $d+2$ in F_ℓ. Note that F_ℓ has exactly k_ℓ free vertices. We remark that these free vertices will play the role of variable vertices for x_ℓ.

For each clause $C_i = (\overline{x}_{i_1} \vee x_{i_2} \vee x_{i_3})$, for $i \in [m]$, we add two disjoint cliques $D_{i,1}$ of size d and $D_{i,2}$ of size $d+1$. In addition, we add a vertex c_i. We let $D_i := D_{i,1} \cup D_{i,2}$ and call $D_i \cup \{c_i\}$ a *clause gadget*. We add all edges between $D_{i,1}$ and $D_{i,2}$ and between c_i and $D_{i,1}$. For each $j \in [3]$, we choose some free vertex of F_{i_j} and call this vertex w_{i_j}. We add edges between w_{i_1} and each vertex in $D_{i,1} \cup \{c_i\}$, edges between w_{i_2} and each vertex in $D_{i,2}$, and we also add an edge between c_i and w_{i_3}, see Figure 7. We choose these free vertices in such a way that every free vertex has neighbours in exactly one clause gadget. This is possible as, for every $\ell \in [n]$, F_ℓ has k_ℓ free vertices and x_ℓ appears k_ℓ times in ϕ. Let G be the resulting graph.

We first show that G is a claw-free graph of maximum degree $2d+3$. Let ℓ be an arbitrary integer in $[n]$. Let $v \in V(F_\ell)$ be a vertex that is not free. Then, by Lemma 13, v has degree at most $2d+3$ and further, since v has neighbours only in F_ℓ, G has no claw centred at v. Let w be a free vertex of F_ℓ. Recall that w has a neighbour in a clause gadget $D_i \cup \{c_i\}$ for some $i \in [m]$. Note that w has degree at most $(d+2)+(d+1) = 2d+3$ since it has $d+2$ neighbours within F_ℓ and at most $d+1$ neighbours in $D_i \cup \{c_i\}$. In addition, the neighbours of w in $D_i \cup \{c_i\}$ form a clique of size 1 or $d+1$. Thus, the neighbourhood of w in G is the union of this clique in $D_i \cup \{c_i\}$ and some clique in F_ℓ, so G has no claw centred at w.

Finally, we consider the clause gadget $D_i \cup \{c_i\}$. Recall that c_i is a fixed vertex with neighbourhood $D_{i,1} \cup \{w_{i_1}, w_{i_3}\}$. Thus, it has degree $d+2$. Since w_{i_1} is complete to $D_{i,1}$, the neighbourhood of c_i is the union of two cliques. Hence, the vertex c_i is not the centre of a claw of G. Let u be a vertex in $D_{i,1}$. The closed neighbourhood of u is $D_i \cup \{c_i, w_{i_1}\}$, which is the union of the two cliques $D_{i,1} \cup \{c_i, w_{i_1}\}$ and D_{i_2}. Hence, u is not the centre of a claw in G. Since $|D_i| = 2d+1$, u has degree $2d+2$. Now, let u' be a vertex in $D_{i,2}$. The closed neighbourhood of u' is $D_i \cup \{w_{i_2}\}$, which is the union of the two cliques D_i and $\{w_{i_2}\}$. So u' is not the centre of a claw in G. Again, as $|D_i| = 2d+1$, the vertex u' has degree $2d+1$. It follows that G is a claw-free graph of maximum degree at most $2d+3$.

We show that G has a d-cut if and only if ϕ has an NAE-satisfying assignment with both a true and a false variable. By Observation 6, this is equivalent to showing that G has a red-blue d-colouring if and only if ϕ has an NAE-satisfying assignment with both a true and a false variable.

We assume first that G has a red-blue d-colouring and show that ϕ has an NAE-satisfying assignment with both a true and a false variable. Recall that by Lemma 13, F_ℓ is monochromatic for each $\ell \in [n]$. We set a variable x_ℓ to true if F_ℓ is coloured blue and to false otherwise.

We first show that the resulting assignment contains both a true and a false variable. Suppose for a contradiction that this is not the case. We may assume without loss of generality that all variables are true. This implies that for every $\ell \in [n]$, F_ℓ is coloured blue. Note that D_i, for $i \in [m]$, is a clique of size $2d+1$ and thus monochromatic by Observation 8. As $w_{i_2} \in F_{i_2}$, it is coloured blue. Further, w_{i_2} has $d+1$ neighbours in $D_{i,2} \subseteq D_i$. Hence, since $D_{i,2}$ is monochromatic, $D_{i,2}$ is coloured blue. It follows that $D_{i,1}$ is blue as well and, since c_i has $d+1$ blue neighbours in $D_{i,1} \cup \{w_{i_1}\}$, it is coloured blue. Thus, the whole graph is coloured blue, contradicting the fact that we considered a red-blue d-colouring. Therefore, the assignment contains both a true and a false variable.

We now show that the assignment is NAE-satisfying for ϕ. Suppose for a contradiction that the assignment is not NAE-satisfying. Then there exists a clause $C_i = (\overline{x}_{i_1} \vee x_{i_2} \vee x_{i_3})$, for some $i \in [m]$, such that all literals have the same value. Note that this implies that both x_{i_2} and x_{i_3} take the opposite value of x_{i_1}. Without loss of generality, assume that x_{i_1} is true and x_{i_2} and x_{i_3} are both false. That is, w_{i_1} is coloured blue while w_{i_2} and w_{i_3} are coloured red. Recall that D_i is monochromatic. Since w_{i_2} has $d+1$ neighbours in $D_{i,2}$, D_i is coloured the same as w_{i_2} and is thus coloured red. This implies that the blue vertex w_{i_1} has d red neighbours in $D_{i,1}$, so its neighbour c_i is coloured blue. However, as w_{i_3} is coloured red, c_i has $d+1$ red neighbours in $D_{i,1} \cup \{w_{i_3}\}$, contradicting the fact that we considered a red-blue d-colouring. Hence, the assignment is NAE-satisfying for ϕ.

For the other direction, we now assume that ϕ has an NAE-satisfying assignment with both a true and a false variable. We construct a red-blue d-colouring of G as follows. For each $\ell \in [n]$, we colour F_ℓ blue if it corresponds to a true variable and red otherwise. For each $i \in [m]$, we colour D_i the same as w_{i_2}. In addition, we colour c_i the same as w_{i_1}.

We show that this colouring is indeed a red-blue d-colouring of G. Note first that since the assignment contains both a true and a false variable, we have both red and blue vertices. It remains to show that every vertex has at most d neighbours of the other colour. Let ℓ be an arbitrary integer in $[n]$. Note that every vertex of F_ℓ which is not a free vertex has no neighbour of the other colour. Let w be a free vertex of F_ℓ. Recall that there is exactly one clause C_i, for $i \in [m]$, such that w is adjacent to the corresponding clause gadget D_i. That is, $w \in \{w_{i_1}, w_{i_2}, w_{i_3}\}$. We distinguish between these cases:

- If $w = w_{i_3}$, then it has one neighbour outside of F_ℓ and thus at most one neighbour of the other colour.
- If $w = w_{i_2}$, then, by construction, it has no neighbour of the other colour.
- If $w = w_{i_1}$, then it has $d + 1$ neighbours outside of F_ℓ and, by construction, at least one of them, namely c_i, has the same colour as w.

Thus, w has at most d neighbours of the other colour.

Now, let u be a vertex in D_i. If $u \in D_{i,2}$, then its closed neighbourhood is $D_i \cup \{w_{i_2}\}$, which is monochromatic by construction. Thus, u has no neighbour of the other colour. If $u \in D_{i,1}$, then, since D_i is monochromatic, the only neighbours of u which may be coloured with the other colour are w_{i_1} and c_i. Since $d \geq 2$, it follows that u has at most d neighbours of the other colour. Thus, we may assume that $u = c_i$. Recall that the neighbourhood of c_i is $D_{i,1} \cup \{w_{i_1}, w_{i_3}\}$ and that c_i is coloured the same as w_{i_1}. Assume without loss of generality that c_i is coloured blue. Suppose for a contradiction that c_i has at least $d + 1$ red neighbours. That is, $D_{i,1}$ and $w_{i,3}$ are coloured red. Since D_i is monochromatic by construction, this implies that $D_{i,2}$ and thus $w_{i,2}$ are coloured red as well. Hence, $w_{i,2}$ and $w_{i,3}$ are both coloured red, while $w_{i,1}$ is coloured blue, a contradiction to the assumption that the assignment is NAE-satisfying. Thus, c_i has at most d neighbours of the other colour. Hence, the colouring is indeed a red-blue d-colouring of G. This completes the proof. ◀

We remark that for every $\Delta > 2d + 3$, d-Cut is still NP-complete on claw-free graphs with maximum degree Δ: after replacing each F_ℓ as a copy of $H_{d,k_\ell,\Delta-1}$, instead of $H_{d,k_\ell,2d+2}$, the same hardness proof works.

5 Conclusions

We proved both constant-time and NP-completeness results for d-Cut restricted to claw-free graphs of bounded maximum degree, leaving open only one case, which replaces Open Problem 1:

▶ **Open Problem 2.** *For every integer $d \geq 2$, what is the computational complexity of d-Cut on claw-free graphs of maximum degree $\Delta = 2d + 2$?*

Recall that we showed in Theorem 3 that there are many claw-free $(2d + 2)$-regular graphs with no d-cut and also that every claw-free graph of maximum degree $\Delta \leq 2d+1$ has a d-cut. This shows that our bound is best possible. On the other hand, our construction used in the proof of Theorem 4 for graphs of maximum degree $\Delta = 2d + 3$ cannot easily be modified to work for graphs of maximum degree $\Delta \leq 2d + 2$. The variable gadget is based on our construction for $(2d + 2)$-regular graphs without a d-cut. We modify the construction such that it contains vertices of degree $d + 2$ that can be used to connect the gadget to the rest of the construction. However, to make sure that we do not get a d-cut by partitioning the graph into a variable gadget and the rest, there has to be a vertex in the variable gadget that has at least $d + 1$ neighbours outside the gadget. This leads to vertices of degree $2d + 3$ in the construction. Hence, reducing the degree in this construction requires us to construct a

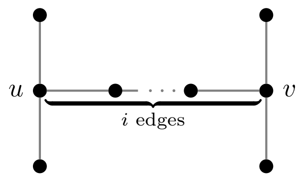

Figure 8 The graph H_i^*. Figure taken from [21].

variable gadget containing some vertices of degree smaller than $d+2$. If we want the gadget to be monochromatic, this implies we need a gadget with vertices of degree $d+1$. However, finding such a gadget seems to be a challenging task.

In our paper we also showed that our positive results for d-Cut on claw-free graphs can be generalized to even $S_{1^t,\ell}$-free graphs. A natural question, which we leave for future work, is whether our results can also be generalized to $S_{1,2,2}$-free graphs. Here, $S_{1,2,2}$ is the graph obtained from the claw by subdividing two of its edges exactly once. In order to answer this question, we need some new arguments as our current proof techniques cannot be applied immediately. For instance, in the proof of Theorem 10, a key observation is that for each $u \in U$, $G[N_u]$ has no independent set of size t, as otherwise we can find an induced $S_{1^t,\ell}$ by combining such an independent set with the path of length ℓ between u and v. This observation no longer holds for the $S_{1,2,2}$ case.

We recall that in particular, our results imply that 2-Cut, restricted to claw-free graphs of maximum degree p, is constant-time solvable if $p \leq 5$ and NP-complete if $p \geq 7$. The latter result resolved an open case in the complexity classification of d-Cut on H-free graphs, as the complexity of 2-Cut for claw-free graphs was not previously known. This brings us to the state-of-the-art summary for d-Cut on H-free graphs, which we update below.

We first give some more terminology. For $r \geq 1$, we denote by P_r and C_r, respectively, a path and a cycle on r vertices. For $s \geq 1$, we denote by sP_r the disjoint union of s copies of P_r. We denote by $S_{1,1,2}$ the graph obtained from a claw by subdividing one edge once, so that it has four edges. Let H_1^* be a graph that looks like the letter "H". It is obtained by taking the graph $2P_3$ and making the middle vertices, say u and v, of each P_3 adjacent to each other. We obtain the graph H_i^*, for $i \geq 2$, by subdividing uv exactly $i-1$ times, see Figure 8. The *girth* of a graph that is not a forest is the length of a shortest cycle in it.

We now present the updated state-of-the-art summary. For every $d \geq 1$, d-Cut is NP-complete for graphs of girth at least g for every fixed $g \geq 3$ [11], and thus for C_r-free graphs for every $r \geq 3$. It is also known that 1-Cut is NP-complete for $(H_1^*, \ldots, H_i^*)$-free graphs for every $i \geq 1$ [11], $K_{1,4}$-free graphs [9], $(3P_5, P_{15})$-free graphs [23] and $(3P_6, 2P_7, P_{14})$-free graphs [18]. On the positive side, 1-Cut is polynomial-time solvable for $(sP_3 + S_{1,1,2})$-free graphs [20], $(sP_3 + P_4 + P_6)$-free graphs [20] and $(sP_3 + P_7)$-free graphs [20] for every $s \geq 1$.

We summarize the above results and our new result for 2-Cut on claw-free graphs in the following theorem, in which all unreferenced results for d-Cut ($d \geq 2$) are shown in [21]. For two graphs H and H', we write $H \subseteq_i H'$ if H is an induced subgraph of H', and denote by $H + H'$ the disjoint union of a copy of H and a copy of H'.

▶ **Theorem 14.** *Let $d \geq 1$ and let H be a graph.*

- *If $d = 1$, then d-Cut (Matching Cut) on H-free graphs is*
  - *polynomial-time solvable if $H \subseteq_i sP_3 + S_{1,1,2}$, $sP_3 + P_4 + P_6$, or $sP_3 + P_7$ for some $s \geq 1$;*
  - NP-*complete if $H \supseteq_i K_{1,4}$, P_{14}, $2P_7$, $3P_5$, C_r for some $r \geq 3$, or H_i^* for some $i \geq 1$.*

- *If $d \geq 2$, then 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 $s \geq 1$;*
  - NP*-complete if $H \supseteq_i K_{1,3}$, $3P_2$, C_r for some $r \geq 3$.*

It is known that for $d \geq 2$, d-Cut is polynomial-time solvable on $(H + P_1)$-free graphs whenever d-Cut is polynomial-time solvable for H-free graphs. This means that the cases $\{H + sP_1 \mid s \geq 0\}$ are all polynomially equivalent. Hence, for $d \geq 2$, there are, due to our new result, now only three non-equivalent open cases left, namely when $H \in \{2P_4, P_6, P_7\}$.

Finally, we are currently exploring the consequences of our results and proofs for finding internal partitions of graphs. To explain, a partition (V_1, V_2) of the vertex set V of a graph G is an *internal partition* of G if for every $i \in \{1, 2\}$, every vertex in V_i has at least half of its neighbours in V_i. Internal partitions are well studied; see, for example, the survey of Bazgan, Tuza and Vanderpooten [4]. The corresponding decision problem that asks whether a graph has an internal partition is known as Satisfactory Partition. As we briefly mentioned in Section 1, there are connections between d-cuts and internal partitions [13].

References

1. 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. doi:10.1016/J.DAM.2021.05.016.
2. N. R. Aravind and Roopam Saxena. An FPT algorithm for Matching Cut and d-Cut. *Proc. IWOCA 2021, LNCS*, 12757:531–543, 2021. doi:10.1007/978-3-030-79987-8_37.
3. Amir Ban and Nati Linial. Internal partitions of regular graphs. *Journal of Graph Theory*, 83:5–18, 2016. doi:10.1002/JGT.21909.
4. Cristina Bazgan, Zsolt Tuza, and Daniel Vanderpooten. Satisfactory graph partition, variants, and generalizations. *European Journal of Operations Research*, 206:271–280, 2010. doi:10.1016/J.EJOR.2009.10.019.
5. Edouard Bonnet, Dibyayan Chakraborty, and Julien Duron. Cutting Barnette graphs perfectly is hard. *Theoretical Computer Science*, 1010:114701, 2024. doi:10.1016/J.TCS.2024.114701.
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. doi:10.1002/JGT.20390.
7. Valentin Bouquet and Christophe Picouleau. The complexity of the Perfect Matching-Cut problem. *Journal of Graph Theory*, 108, 2025. doi:10.1002/JGT.23167.
8. Chi-Yeh Chen, Sun-Yuan Hsieh, Hoàng-Oanh Le, Van Bang Le, and Sheng-Lung Peng. Matching Cut in graphs with large minimum degree. *Algorithmica*, 83:1238–1255, 2021. doi:10.1007/S00453-020-00782-8.
9. Vasek Chvátal. Recognizing decomposable graphs. *Journal of Graph Theory*, 8:51–53, 1984. doi:10.1002/JGT.3190080106.
10. Tala Eagling-Vose, Barnaby Martin, Daniel Paulusma, and Siani Smith. A forbidden subgraph study for cut problems on graphs permitting loops and multiedges. *CoRR*, abs/2502.07769, 2025. doi:10.48550/arXiv.2502.07769.
11. Carl Feghali, Felicia Lucke, Daniël Paulusma, and Bernard Ries. Matching cuts in graphs of high girth and H-free graphs. *Algorithmica*, 87:1199–1221, 2025. doi:10.1007/S00453-025-01318-8.
12. 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. doi:10.1016/J.JCSS.2021.07.005.
13. Guilherme Gomes and Ignasi Sau. Finding cuts of bounded degree: complexity, FPT and exact algorithms, and kernelization. *Algorithmica*, 83:1677–1706, 2021. doi:10.1007/S00453-021-00798-8.

14 Guilherme C. M. Gomes, Emanuel Juliano, Gabriel Martins, and Vinícius Fernandes dos Santos. Matching (multi)cut: Algorithms, complexity, and enumeration. *Proc. IPEC 2024, LIPIcs*, 321:25:1–25:15, 2024. doi:10.4230/LIPICS.IPEC.2024.25.

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

16 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. doi:10.1016/J.DAM.2019.12.010.

17 Dieter Kratsch and Van Bang Le. Algorithms solving the Matching Cut problem. *Theoretical Computer Science*, 609:328–335, 2016. doi:10.1016/J.TCS.2015.10.016.

18 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. doi:10.1007/978-3-031-43380-1_30.

19 Van Bang Le and Jan Arne Telle. The Perfect Matching Cut problem revisited. *Theoretical Computer Science*, 931:117–130, 2022. doi:10.1016/J.TCS.2022.07.035.

20 Felicia Lucke, Joseph Marchand, and Jannik Olbrich. Finding minimum matching cuts in H-free graphs and graphs of bounded radius and diameter. *CoRR*, abs/2502.18942, 2025. doi:10.48550/arXiv.2502.18942.

21 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*, 14760:415–429, 2024. full version: arXiv:2404.11389. doi:10.48550/arXiv.2404.11389.

22 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. doi:10.1016/J.TCS.2022.09.014.

23 Felicia Lucke, Daniël Paulusma, and Bernard Ries. Finding matching cuts in H-free graphs. *Algorithmica*, 85:3290–3322, 2023. doi:10.1007/S00453-023-01137-9.

24 Felicia Lucke, Daniël Paulusma, and Bernard Ries. Dichotomies for maximum matching cut: H-freeness, bounded diameter, bounded radius. *Theoretical Computer Science*, 1017:114795, 2024. doi:10.1016/J.TCS.2024.114795.

25 Augustine M. Moshi. Matching cutsets in graphs. *Journal of Graph Theory*, 13:527–536, 1989. doi:10.1002/JGT.3190130502.