Matching Cut in Graphs with Large Minimum Degree

Chi-Yeh Chen¹ · Sun-Yuan Hsieh¹ · Hoang-Oanh Le² · Van Bang Le³ · Sheng-Lung Peng⁴

Received: 14 February 2020 / Accepted: 4 November 2020 / Published online: 23 November 2020
© The Author(s) 2020

Abstract

In a graph, a matching cut is an edge cut that is a matching. **MATCHING CUT** is the problem of deciding whether or not a given graph has a matching cut, which is known to be NP-complete. While **MATCHING CUT** is trivial for graphs with minimum degree at most one, it is NP-complete on graphs with minimum degree two. In this paper, we show that, for any given constant $c > 1$, **MATCHING CUT** is NP-complete in the class of graphs with minimum degree c and this restriction of **MATCHING CUT** has no subexponential-time algorithm in the number of vertices unless the Exponential-Time Hypothesis fails. We also show that, for any given constant $\epsilon > 0$, **MATCHING CUT** remains NP-complete in the class of n -vertex (bipartite) graphs with unbounded minimum degree $\delta > n^{1-\epsilon}$. We give an exact branching algorithm to solve **MATCHING CUT** for graphs with minimum degree $\delta \geq 3$ in $O^*(\lambda^n)$ time, where λ is the positive root of the polynomial $x^{\delta+1} - x^\delta - 1$. Despite the hardness results, this is a very fast exact exponential-time algorithm for **MATCHING CUT** on graphs with large minimum degree; for instance, the running time is $O^*(1.0099^n)$ on graphs with minimum degree $\delta \geq 469$. Complementing our hardness results, we show that, for any two fixed constants $1 < c < 4$ and $c' \geq 0$, **MATCHING CUT** is solvable in polynomial time for graphs with large minimum degree $\delta \geq \frac{1}{c}n - c'$.

Keywords Matching cut · Graph algorithm · Exact branching algorithm · NP-hard problem

A preliminary version [10] of this article has appeared in Proceedings of the 25th International Computing and Combinatorics Conference (COCOON 2019), Lecture Notes in Computer Science 11653, pp. 301–312.

✉ Van Bang Le
van-bang.le@uni-rostock.de

Extended author information available on the last page of the article

1 Introduction

In a graph $G = (V, E)$, a *cut* is a partition $V = X \dot{\cup} Y$ of the vertex set into disjoint, non-empty sets X and Y , written (X, Y) . The set of all edges in G having an endvertex in X and the other endvertex in Y , also written (X, Y) , is called the *edge cut* of the cut (X, Y) . A *matching cut* is an edge cut that is a (possibly empty) matching. Another way to define matching cuts is as follows [5, 9]. A partition $V = X \dot{\cup} Y$ of the vertex set of the graph $G = (V, E)$ into disjoint, non-empty sets X and Y , is a matching cut if and only if each vertex in X has at most one neighbor in Y and each vertex in Y has at most one neighbor in X .

Graham [9] studied matching cuts in graphs in connection to a number theory problem called cube-numbering. In [6], Farley and Proskurowski studied matching cuts in the context of network applications. Patrignani and Pizzonia [18] pointed out an application of matching cuts in graph drawing. Matching cuts have been used by Araújo et al. [1] in studying good edge-labellings in the context of WDM (Wave-length Division Multiplexing) networks.

Not every graph has a matching cut; the MATCHING CUT problem is the problem of deciding whether or not a given graph has a matching cut:

MATCHING CUT

Instance: A graph $G = (V, E)$.

Question: Does G have a matching cut?

Observe that disconnected graphs and graphs having a bridge have a matching cut. In particular, MATCHING CUT is trivial for graphs with minimum degree at most one. It is known, however, that MATCHING CUT is NP-complete on (bipartite) graphs with minimum degree two and on (bipartite) graphs with minimum degree three (see [16]). This paper considers the computational complexity of, as well as algorithms for the MATCHING CUT problem in graphs of large minimum degree.

1.1 Previous Results

Graphs admitting a matching cut were first discussed by Graham in [9] under the name *decomposable graphs*. The first complexity and algorithmic results for MATCHING CUT have been obtained by Chvátal, who proved in [5] that MATCHING CUT is NP-complete, even when restricted to graphs of maximum degree four, and is solvable in polynomial time for graphs with maximum degree at most three. These results triggered a lot of research on the computational complexity of MATCHING CUT in graphs with additional structural assumptions; see [3, 4, 14, 16–18]. In particular, the NP-hardness of MATCHING CUT has been further strengthened for planar graphs of maximum degree four [3] and bipartite graphs of minimum degree three and maximum degree four [16]. Moreover, it follows from Bonsma’s result [3] and a simple reduction observed by Moshi [17] that MATCHING CUT remains NP-complete

on bipartite planar graphs of minimum degree two and maximum degree eight. Recently, Le and Le [15] proved that MATCHING CUT is NP-complete on graphs of diameter d for any fixed integer $d \geq 3$ and on bipartite graphs of diameter d for any fixed integer $d \geq 4$.

On the positive side, some polynomially solvable cases have been identified: graphs of maximum degree 3 [5], graphs without chordless cycle of length at least five (including chordal and chordal bipartite graphs) [17], claw-free graphs (including line graphs) [3], graphs of diameter 2 [4, 15], bipartite graphs of diameter at most 3 [15]. In [14], the cases of graphs of maximum degree 3 and of claw-free graphs have been extended to a larger class in which MATCHING CUT is still solvable in polynomial time.

Parameterized and exact exponential-time algorithms for MATCHING CUT on graphs without any restriction have been considered in [2, 13, 14]. Kratsch and Le [14] provided the first exact branching algorithm for MATCHING CUT running in time $O^*(1.4143^n)$ ¹ and noted that MATCHING CUT cannot be solved in $O^*(2^{o(n)})$ time for n -vertex graphs, assuming the Exponential Time Hypothesis (ETH) [11, 12] is true. Recently, the running time has been improved in [13] to $O^*(1.3803^n)$. We note that the running time analysis of these branching algorithms highly depends on the minimum degree ≥ 2 of the input graphs. This fact was the main motivation of the present work.

Very recently, Gomes and Sau [8] considered a natural generalization of MATCHING CUT, the d -CUT problem: for an integer $d \geq 1$, a d -cut is a bipartition of the vertex set of a graph into two non-empty sets A and B such that each vertex has at most d neighbors across the cut. (A d -cut is also called a *symmetric $K_{1,d+1}$ -free decomposition* in [19].) Thus, MATCHING CUT and 1-CUT coincide. Among others, Gomes and Sau proved that d -CUT is NP-complete for any $d \geq 1$, even in the class of $(2d + 2)$ -regular graphs.

1.2 Our Contributions

We first prove the following hardness results:

- For any constant $c > 1$, MATCHING CUT restricted on graphs with minimum degree c is NP-complete and, assuming the ETH, cannot be solved in $O^*(2^{o(n)})$ time for n -vertex graphs.
- For any fixed constant $\epsilon > 0$ and integer $d \geq 1$, d -CUT remains NP-complete when restricted to n -vertex graphs of unbounded minimum degree at least $n^{1-\epsilon}$. In case of MATCHING CUT, i.e., $d = 1$, the same result holds true even in the class of bipartite graphs. This hardness result is somehow surprising, as, intuitively, MATCHING CUT might be easy for very dense graphs, graphs with very large minimum degree.

¹ Throughout the paper we use the O^* notation which suppresses polynomial factors.

We then show that the exact branching algorithms in [13, 14] solving MATCHING CUT can be simplified for graphs with large minimum degree δ , which results in a running time $O^*(\lambda^n)$, where λ is the positive root of the polynomial $x^{\delta+1} - x^\delta - 1$. For $\delta \geq 4$, this running time is $O^*(1.3248^n)$, improving upon the previous best known $O^*(1.3803^n)$ -time algorithm.

It is quite interesting to note, as we will see, that $\lambda \rightarrow 1$ as $\delta \rightarrow \infty$. Thus, for any constant $\epsilon > 0$, a natural restriction of MATCHING CUT on graphs with (large) minimum degree $\delta = \delta(\epsilon)$ can be solved in $O^*((1 + \epsilon)^n)$ time, although this restriction has no subexponential-time algorithm (assuming the ETH). For instance, for $\delta \geq 469$, our exact algorithm solves MATCHING CUT on graphs with minimum degree δ in time $O^*(1.0099^n)$! Despite the hardness results, this is a very fast exponential-time algorithm for an NP-hard problem and we are not aware of another natural NP-hard problem that has no subexponential-time algorithm (unless the ETH fails) but admits an exponential-time algorithm with running time near $O^*(1.0099^n)$. We remark that the authors of [20] provided an $O^*(1.02220^n)$ -time algorithm for an NP-hard problem (the PARTITION INTO TRIANGLES problem restricted on graphs of maximum degree 4) that cannot be solved in subexponential-time (assuming ETH), and ask for such rare NP-hard problems with very fast exponential-time algorithms. Our results add an infinite family of such natural problems into this list.

Finally, complementing our hardness results, we show that, for any two fixed constants $1 < c < 4$ and $c' \geq 0$, MATCHING CUT is solvable in polynomial time for graphs with large minimum degree $\delta \geq \frac{1}{c}n - c'$.

1.3 Notation and Terminology

Let $G = (V, E)$ be a graph with vertex set $V(G) = V$ and edge set $E(G) = E$. An *independent set* (a *clique*) in G is a set of pairwise non-adjacent (adjacent) vertices. The neighborhood of a vertex v in G , denoted $N_G(v)$, is the set of all vertices in G adjacent to v ; the closed neighborhood of v is $N_G[v] = N_G(v) \cup \{v\}$. If the context is clear, we simply write $N(v)$ and $N[v]$. Set $\deg(v) = |N(v)|$, the degree of the vertex v , and $\delta(G) = \min\{\deg(v) \mid v \in V(G)\}$, the minimum degree of G . For a subset $W \subseteq V$, the W -neighbors of a vertex v are the vertices in $N(v) \cap W$, $G[W]$ is the subgraph of G induced by W , and $G - W$ stands for $G[V \setminus W]$. The complete graph and the path on n vertices are denoted by K_n and P_n , respectively; K_3 is also called a *triangle*. The complete bipartite graph with one color class of size p and the other of size q is denoted by $K_{p,q}$.

Given a graph $G = (V, E)$ and a partition $V = X \cup Y$, it can be decided in linear time if (X, Y) is a matching cut of G . This is because (X, Y) is a matching cut of G if and only if the bipartite subgraph $B_G(X, Y)$ of G with the color classes X and Y and edge set (X, Y) is P_3 -free. That is, (X, Y) is a matching cut of G if and only if the non-trivial connected components of the bipartite graph $B_G(X, Y)$ are edges. We say that $S \subseteq V$ is *monochromatic* if, for any matching cut (X, Y) of G (if any), $S \subseteq X$ or else $S \subseteq Y$. Observe that (the vertex set of) any K_n with $n \geq 3$, and any $K_{p,q}$ with $p \geq 2, q \geq 3$, in G is monochromatic. A *bridge* in a graph is an edge whose deletion increases the number of the connected components. Since disconnected graphs and

graphs having a bridge have a matching cut, we may assume that all graphs considered are connected and 2-edge connected.

When an algorithm branches on the current instance of size n into r subproblems of sizes at most $n - t_1, n - t_2, \dots, n - t_r$, then $(t_1, t_2, \dots, t_r)$ is called the *branching vector* of this branching, and the unique positive root of the characteristic polynomial $x^n - x^{n-t_1} - x^{n-t_2} - \dots - x^{n-t_r}$, denoted $\tau(t_1, t_2, \dots, t_r)$, is called its *branching number*. The running time of a branching algorithm is $O^*(\lambda^n)$, where $\lambda = \max_i \lambda_i$ and λ_i is the branching number of branching rule i , and the maximum is taken over all branching rules. We refer to [7] for more details on exact branching algorithms.

1.4 Structure of the Paper

The paper is organized as follows. In Sect. 2 we show that MATCHING CUT is NP-complete when restricted on graphs with minimum degree c , for any given constant $c > 1$, and that this restriction of MATCHING CUT has no subexponential-time algorithm unless the ETH fails. We then show that MATCHING CUT remains NP-complete when restricted to graphs with minimum degree $\delta \geq n^{1-\epsilon}$, for any given constant $\epsilon > 0$. The same result holds also for bipartite graphs. In Sect. 3 we describe the fast exact branching algorithm solving MATCHING CUT for graphs with large minimum degree. In Sect. 4 we show that MATCHING CUT can be solved in polynomial time for graphs with very large minimum degree at least $\frac{1}{c}n - c'$ for any two fixed constants $1 < c < 4, c' \geq 0$. We conclude the paper with Sect. 5.

2 Hardness Results

Observe that if G has a vertex v with $\deg(v) \leq 1$, then $(\{v\}, V(G) \setminus \{v\})$ is a matching cut of G . It is known that MATCHING CUT is NP-complete for graphs with minimum degree two, and that, assuming ETH, MATCHING CUT has no subexponential-time algorithm. We first strengthen these hardness results for graphs with given minimum degree $c \geq 2$ in Theorem 1 below. As pointed out by a reviewer, Chvátal's original hardness reduction [5] can be modified to show that MATCHING CUT is NP-complete for graphs with minimum degree c . However, to make the paper self-contained, we include our reduction here, which demands less technical details and is thus cleaner than discussing an adaptation of Chvátal's proof.

Theorem 1 *For any constant $c \geq 2$, MATCHING CUT remains NP-complete when restricted on graphs with minimum degree c and still admits no subexponential-time algorithm in the number of vertices unless the ETH fails.*

Proof Given constant $c > 1$, we reduce MATCHING CUT to MATCHING CUT restricted on graphs with minimum degree c in polynomial time as follows. Given a graph $G = (V, E)$ with n vertices and m edges, let G' be obtained from G and n cliques $Q_v, v \in V$, each of size c by joining edges between v and all vertices in Q_v . Then, G' has $n' = (c + 1)n$ vertices, $m' = m + \frac{1}{2}c^2(c - 1)n$ edges and minimum degree c .

Moreover, G has a matching cut if and only if G' has a matching cut (this is because the cliques $Q_v \cup \{v\}$, $v \in V$, are monochromatic). Thus, MATCHING CUT remains NP-complete when restricted on graphs with minimum degree c .

Now, suppose MATCHING CUT restricted on graphs with minimum degree c can be solved in subexponential time; that is, there is an algorithm with running time $O^*(2^{\epsilon' \cdot n'})$ on G' for all constants $\epsilon' > 0$. Then such an algorithm with $\epsilon' = \epsilon/(c+1)$ would imply, by the polynomial-time reduction above, an algorithm for MATCHING CUT on G with running time $O^*(2^{\epsilon \cdot n})$ for all $\epsilon > 0$. However, assuming ETH, no such algorithm for MATCHING CUT exists. $\square$

The question arises what happens when considering MATCHING CUT restricted on graphs with unbounded minimum degree. In such a very dense graph G having a matching cut (X, Y) , almost all edges are within X or else within Y . Thus, G consists of two disjoint very dense induced subgraphs, between which there are very few edges, namely a matching. Because of this highly restricted structure one might expect that it would be easy to check if a very dense graph has a matching cut. This is true for Dirac and Ore graphs, for example (cf. Corollary 1 and Theorem 6). But in general it is not the case as we will see below that MATCHING CUT remains NP-complete on graphs with unbounded minimum degree (Theorem 2 with $d = 1$), even on such bipartite graphs (Theorem 3).

Theorem 2 *Let $\epsilon > 0$. For every integer $d \geq 1$, d -CUT remains NP-complete on n -vertex graphs with minimum degree at least $n^{1-\epsilon}$.*

Proof Let $\epsilon > 0$ and integer $d \geq 1$ be two constants. Let $G = (E, V)$ be an instance for d -CUT with n vertices and $\delta = \delta(G) \geq 2$. We are going to construct, in polynomial time, a graph $G' = (V', E')$ with n' vertices and minimum degree $\delta' \geq (n')^{1-\epsilon}$ such that G has a d -cut if and only if G' has a d -cut. Since d -CUT is NP-complete [8], we then obtain Theorem 2.

Let G' be obtained from $t = \lceil n^{\frac{1-\epsilon}{\epsilon}} \rceil$ copies of G by making all t copies of any vertex $v \in V$ to a clique. More precisely, writing $V = \{v_1, \dots, v_n\}$ we have; see also Fig. 1.

- $V' = V^1 \cup \dots \cup V^t$, where
 - $V^s = \{v_1^s, \dots, v_n^s\}$ is the s th copy of the vertex set V , $1 \leq s \leq t$.
- $E' = E^1 \cup \dots \cup E^t \cup F_1 \cup \dots \cup F_n$, where
 - $E^s = \{v_i^s v_j^s \mid 1 \leq i, j \leq n, v_i v_j \in E\}$ is the s th copy of the edge set E , $1 \leq s \leq t$, and
 - $F_i = \{v_i^s v_i^{s'} \mid 1 \leq s, s' \leq t, s \neq s', 1 \leq i \leq n\}$.

Thus, $G'[V^s] = (V^s, E^s)$, $1 \leq s \leq t$, are the t copies of G and for each $1 \leq i \leq n$, the set $Q_i = \{v_i^1, \dots, v_i^t\}$ of the t copies of $v_i \in V$ induces a clique in G' (with the edge set F_i). Note that, given constants ϵ and d , we may assume that G has $n \geq (2d+1)^{\epsilon/(1-\epsilon)}$ vertices (otherwise, G has bounded size). Then the number t of G -copies is at least $n^{(1-\epsilon)/\epsilon} \geq 2d+1$, ensuring that the cliques Q_i in G' are monochromatic whenever G' has a d -cut.

Fig. 1 The graphs G and G'

Clearly, given $\epsilon > 0$ and G , G' can be constructed in polynomial time $t \cdot O(n^2) = O(n^{3\frac{1-\epsilon}{\epsilon}+1})$. Moreover, G has a d -cut if and only if G' has a d -cut: first, if (X, Y) is a d -cut of G , then (X', Y') with

$$X' = \bigcup_{1 \leq i \leq n : v_i \in X} Q_i \quad \text{and} \quad Y' = \bigcup_{1 \leq i \leq n : v_i \in Y} Q_i$$

is, by definition of G' , a d -cut of G' . Second, if (X', Y') is a d -cut of G' , then for each $1 \leq i \leq n$, the clique Q_i is contained in X' or else in Y' (as Q_i is monochromatic). Hence, (X, Y) with

$$X = \{v_i \mid 1 \leq i \leq n, Q_i \subseteq X'\} \quad \text{and} \quad Y = \{v_i \mid 1 \leq i \leq n, Q_i \subseteq Y'\}$$

is a d -cut of G .

Now, G' has $n' = tn$ vertices. Hence $t \geq n^{\frac{1-\epsilon}{\epsilon}} = (n'/t)^{\frac{1-\epsilon}{\epsilon}}$, implying $t \geq (n')^{1-\epsilon}$. Therefore, G' has minimum degree $\delta' = \delta + t - 1 > t \geq (n')^{1-\epsilon}$. $\square$

We remark that in the proof of Theorem 2 if G is δ -regular, then G' is δ' -regular. Since d -CUT remains NP-complete in the class of $(2d+2)$ -regular graphs, we conclude that d -CUT remains NP-complete when restricted on regular graphs of unbounded degree.

We now modify the proof of Theorem 2 to show that MATCHING CUT remains NP-complete on bipartite graphs with unbounded minimum degree. Recall that MATCHING CUT is NP-complete in the class of bipartite graphs.

Theorem 3 *Let $\epsilon > 0$. MATCHING CUT remains NP-complete on n -vertex bipartite graphs with minimum degree at least $n^{1-\epsilon}$.*

Proof Let $\epsilon > 0$ and let $G = (V, E)$ be a bipartite graph with bipartition $V = U \cup W$ and minimum degree $\delta = \delta(G) \geq 2$. The construction in the proof of Theorem 2 can be modified to obtain a desired bipartite graph $G' = (U' \cup W', E')$ consisting of $2t$ copies of G as follows; see also Fig. 2. Write $U = \{u_1, \dots, u_p\}$, $W = \{w_1, \dots, w_q\}$. Thus, $n = p + q$ is the vertex number of G .

Fig. 2 The bipartite graphs G and G' . The bicliques Q_i^U and Q_j^W , $1 \leq i \leq p$, $1 \leq j \leq q$, are drawn in blue and red, respectively (Color figure online)

- Let $t = \lceil (2n)^{\frac{1-\epsilon}{\epsilon}} \rceil$.
- $U' = U^1 \cup \dots \cup U^t \cup W^{t+1} \cup \dots \cup W^{2t}$, and
- $W' = W^1 \cup \dots \cup W^t \cup U^{t+1} \cup \dots \cup U^{2t}$, where
 - $U^s = \{u_1^s, \dots, u_p^s\}$ is the s th copy of the color class U , $1 \leq s \leq 2t$,
 - $W^s = \{w_1^s, \dots, w_q^s\}$ is the s th copy of the color class W , $1 \leq s \leq 2t$.
- $E' = E^1 \cup \dots \cup E^{2t} \cup F_1^U \cup \dots \cup F_p^U \cup F_1^W \cup \dots \cup F_q^W$, where
 - $E^s = \{u_i^s w_j^s \mid 1 \leq i \leq p, 1 \leq j \leq q, u_i w_j \in E\}$ is the s th copy of the edge set E , $1 \leq s \leq 2t$,
 - $F_i^U = \{u_i^s u_i^{s'} \mid 1 \leq s \leq t, t+1 \leq s' \leq 2t\}$, $1 \leq i \leq p$, and
 - $F_j^W = \{w_j^s w_j^{s'} \mid 1 \leq s \leq t, t+1 \leq s' \leq 2t\}$, $1 \leq j \leq q$.

Thus, $G'[U^s \cup W^s] = (U^s \cup W^s, E^s)$, $1 \leq s \leq 2t$, are the $2t$ copies of G and for each $1 \leq i \leq p$, the set $Q_i^U = \{u_i^1, \dots, u_i^t\} \cup \{u_i^{t+1}, \dots, u_i^{2t}\}$ of the $2t$ copies of $u_i \in U$ induces a complete bipartite subgraph, i.e., a biclique, in G' (with the edge set F_i^U) and for each $1 \leq j \leq q$, the set $Q_j^W = \{w_j^1, \dots, w_j^t\} \cup \{w_j^{t+1}, \dots, w_j^{2t}\}$ of the $2t$ copies of $w_j \in W$ induces a biclique in G' (with the edge set F_j^W).

Similarly to the proof of Theorem 2, it can be seen that G has a matching cut and only if G' has a matching cut: first, if (X, Y) is a matching cut of G , then (X', Y') with

$$X' = \bigcup_{1 \leq i \leq p : u_i \in X} Q_i^U \cup \bigcup_{1 \leq j \leq q : w_j \in X} Q_j^W,$$

$$Y' = \bigcup_{1 \leq i \leq p : u_i \in Y} Q_i^U \cup \bigcup_{1 \leq j \leq q : w_j \in Y} Q_j^W,$$

is, by definition of G' , a matching cut of G' . Second, if (X', Y') is a matching cut of G' , then for each $1 \leq i \leq p$ and $1 \leq j \leq q$, each of the bicliques Q_i^U and Q_j^W is contained in X' or else in Y' (as Q_i^U and Q_j^W are monochromatic.) Hence, (X, Y) with

$$X = \{u_i \mid 1 \leq i \leq p, Q_i^U \subseteq X'\} \cup \{w_j \mid 1 \leq j \leq q, Q_j^W \subseteq X'\},$$

$$Y = \{u_i \mid 1 \leq i \leq p, Q_i^U \subseteq Y'\} \cup \{w_j \mid 1 \leq j \leq q, Q_j^W \subseteq Y'\}$$

is a matching cut of G .

The obtained bipartite graph G' has $n' = 2tn$ vertices and minimum degree $\delta' = \delta + t > t \geq (2n)^{\frac{1-\epsilon}{\epsilon}} \geq (n')^{1-\epsilon}$. $\square$

Observe that the construction in the proof of Theorem 3 can be slightly modified for d -CUT in the class of bipartite graphs. Hence, assuming d -CUT is NP-complete on bipartite graphs, Theorem 3 also holds true for d -CUT. However, to our knowledge, for $d \geq 2$, the complexity of d -CUT restricted on bipartite graphs is unknown so far.

3 A Very Fast Exact Exponential-Time Algorithm for Graphs with Large Minimum Degree

In [14], the first exact algorithm solving MATCHING CUT for n -vertex graphs has the running time $O^*(1.4143^n)$. This branching algorithm has been improved to $O^*(1.3803^n)$ in [13]. We remark that the branching steps, hence the time complexity of the proposed algorithms, strongly depend on the minimum degree $\delta \geq 2$ of the input graphs. In this section, we observe that the algorithms proposed in [13, 14] can be simplified for input graphs with minimum degree $\delta \geq 3$ that results in an exact algorithm with running time $O^*(\lambda^n)$, where λ is the positive root of the polynomial $x^{\delta+1} - x^\delta - 1$. For graphs with minimum degree $\delta \geq 4$, this running time is $O^*(1.3248^n)$, improved upon the mentioned $O^*(1.3803)$ -time algorithm. For graphs with large minimum degree $\delta \geq 469$, the running time is $O^*(1.0099^n)$; see also the Table 1. It is perhaps important to recall that, by Theorem 1, for arbitrary large constant c , MATCHING CUT restricted on graphs with minimum degree $\delta \geq c$ has no sub-exponential-time algorithm (assuming the ETH).

Table 1 Roots λ (rounded up) of $x^{\delta+1} - x^\delta - 1 = 0$ for some concrete minimum degree δ

δ	3	4	25	199	469	6165	7747
λ	1.3803	1.3248	1.0976	1.0199	1.0099	1.0011	1.0009

Before describing the algorithm, we show in Fact 1 below that $\lambda \rightarrow 1$ as $\delta \rightarrow \infty$.

Fact 1 As δ tends to infinity, root λ of $x^{\delta+1} - x^\delta - 1$ tends to one.

Proof Let λ_δ be the positive root of $x^{\delta+1} - x^\delta - 1$. That is, λ_δ is the branching number $\tau(\delta + 1, 1)$. It is well-known (see, e.g., [7, Lemma 2.2]) that, for any integer $\delta \geq 1$,

$$1 < \lambda_{\delta+1} < \lambda_\delta. \quad (1)$$

Thus, λ_δ is (strictly) decreasing in δ and is bounded from below. Hence, the limit $\lambda^* := \lim_{\delta \rightarrow \infty} \lambda_\delta$ exists. We are going to show that $\lambda^* = 1$.

Note first that, by (1), for any integer $\delta \geq 1$,

$$1 \leq \lambda^* \leq \lambda_\delta. \quad (2)$$

Moreover, it follows from $(\lambda_\delta)^{\delta+1} - (\lambda_\delta)^\delta - 1 = 0$ that $\lambda_\delta - 1 = \left(\frac{1}{\lambda_\delta}\right)^\delta$, and with (2),

$$0 \leq \lambda^* - 1 \leq \lambda_\delta - 1 = \left(\frac{1}{\lambda_\delta}\right)^\delta \leq \left(\frac{1}{\lambda^*}\right)^\delta. \quad (3)$$

Now, if $\lambda^* > 1$, then the right most term in (3) would tend to zero as δ tends to infinity, and this would imply $\lambda^* - 1 = 0$, a contradiction. Therefore $\lambda^* = 1$, as claimed. $\square$

Thus, for arbitrary small $\epsilon > 0$, we can solve MATCHING CUT restricted on graphs with (large) minimum degree $\delta = \delta(\epsilon)$ in $O^*((1 + \epsilon)^n)$ time, although it does not admit an $O^*(2^{o(n)})$ -time algorithm.

The idea of our algorithm is as follows [14]. If the input graph $G = (V, E)$ has a matching cut (X, Y) , then some vertex a is contained in X and some vertex b is contained in Y . The algorithm is a branching algorithm and will be executed for all possible pairs $a, b \in V$, hence $O(n^2)$ times. To do this set $A := \{a\}$, $B := \{b\}$, and $F := V \setminus \{a, b\}$ and call the branching algorithm. At each stage of the algorithm, A and/or B will be extended or it will be determined that there is no matching cut separating A and B , that is a matching cut (X, Y) with $A \subseteq X$ and $B \subseteq Y$. We describe our algorithm by a list of reduction and branching rules given in preference order, i.e., in an execution of the algorithm on any instance of a subproblem one always applies the first rule applicable to the instance, which could be a reduction or a branching rule. A reduction rule produces one instance/subproblem while a branching rule results in at least two instances/subproblems, with different extensions of A and B . Note that G has a matching cut that separates A from B if and only if in at least one recursive branch, extensions A' of A and B' of B are obtained such that G has a matching cut that separates A' from B' . Typically a rule assigns one or more free vertices, vertices of F , either to A or to B and removes them from F , that is, we always have $F := V \setminus (A \cup B)$.

The algorithm first applies Reduction Rules (R1) – (R4) given in [14].

- (R1) If a vertex in A has two B -neighbors, or a vertex in B has two A -neighbors then STOP: “ G has no matching cut separating A, B ”. If $v \in F$, $|N(v) \cap A| \geq 2$ and $|N(v) \cap B| \geq 2$ then STOP: “ G has no matching cut separating A, B ”. If there is an edge xy in G such that $x \in A$ and $y \in B$ and $N(x) \cap N(y) \cap F \neq \emptyset$ then STOP: “ G has no matching cut separating A, B ”.
- (R2) If $v \in F$ and $|N(v) \cap A| \geq 2$ then $A := A \cup \{v\}$. If $v \in F$ and $|N(v) \cap B| \geq 2$ then $B := B \cup \{v\}$.
- (R3) If $v \in A$ has two adjacent F -neighbors w_1, w_2 then $A := A \cup \{w_1, w_2\}$. If $v \in B$ has two adjacent F -neighbors w_3, w_4 then $B := B \cup \{w_3, w_4\}$.
- (R4) If there is an edge xy in G such that $x \in A$ and $y \in B$ then add $N(x) \cap F$ to A (if $N(x) \cap F \neq \emptyset$), and add $N(y) \cap F$ to B (if $N(y) \cap F \neq \emptyset$).

If none of these reduction rules can be applied then

- the A, B -edges of G form a matching cut in $G[A \cup B] = G - F$ due to (R1),
- every vertex in F is adjacent to at most one vertex in A and at most one vertex in B due to (R2),
- the neighbors in F of any vertex in A and the neighbors in F of any vertex in B form an independent set due to (R3), and
- every vertex in A adjacent to a vertex in B has no neighbor in F and every vertex in B adjacent to a vertex in A has no neighbor in F .

In contrast to the previous two algorithms in [13, 14], our algorithm consists of only two branching rules, which are based on the following fact. See also Fig. 3.

Fig. 3 When the branching rules (B1) and (B2) are applicable

Lemma 1 *Let $a \in A$ be a vertex with two (non-adjacent) neighbors u and v in F . Then*

- (B1) *G has a matching cut separating A and B if and only if G has a matching cut separating $A \cup \{u\}$ and $B \cup (N[v] \cap F)$ or a matching cut separating $A \cup \{v\}$ and B .*
- (B2) *If $N(v) \cap B \neq \emptyset$, then G has a matching cut separating A and B if and only if G has a matching cut separating $A \cup \{u\}$ and $B \cup (N[v] \cap F)$ or a matching cut separating $A \cup (N[v] \cap F)$ and B .*

Proof (B1): Let (X, Y) be a matching cut of G such that $A \subseteq X$ and $B \subseteq Y$. If $v \in X$, then (X, Y) separates $A \cup \{v\}$ and B . If $v \in Y$, then, as v is adjacent to $a \in A \subseteq X$, u must belong to X and $N[v] \cap F$ must belong to Y . Thus, (X, Y) is a matching cut separating $A \cup \{u\}$ and $B \cup (N[v] \cap F)$. Conversely, it is clear that any matching cut separating $A \cup \{u\}$ and $B \cup (N[v] \cap F)$, or $A \cup \{v\}$ and B is particularly a matching cut separating A and B .

(B2): Assume $N(v) \cap B \neq \emptyset$, and let (X, Y) be a matching cut of G such that $A \subseteq X$ and $B \subseteq Y$. By (B1), it remains to consider the case where $v \in X$. In this case, as v has a neighbor in $B \subseteq Y$, $N[v] \cap F$ must belong to X . Thus, (X, Y) is a matching cut separating $A \cup \{u\}$ and $B \cup (N[v] \cap F)$ or separating $A \cup (N[v] \cap F)$ and B . $\square$

To determine the branching vectors which correspond to our branching rules, we set the size of an instance (G, A, B) as its number of free vertices, i.e., $|V(G)| - |A| - |B|$.

Let $a \in A$ with $|N(a) \cap F| \geq 2$, and choose two neighbors $u \neq v$ in F of a . Due to Reduction Rule (R3), u and v are non-adjacent. Due to Reduction Rule (R2), $N(v) \cap A = \{a\}$.

- If $N(v) \cap B = \emptyset$, we branch into two subproblems according to (B1): First, add u to A and add $N[v] \cap F = N[v] \setminus \{a\}$ to B . Second, add v to A . Hence the branching vector is $(\delta + 1, 1)$.
- If $N(v) \cap B \neq \emptyset$, we branch into two subproblems according to (B2): First, add u to A and add $N[v] \cap F$ to B . Second, add $N[v] \cap F$ to A . Due to Reduction Rule (R2), we have $|N(v) \cap B| = 1$, hence the branching vector is $(\delta, \delta - 1)$.

Note that, for $\delta \geq 3$, the branching number of the branching vector $(\delta, \delta - 1)$ is smaller than the branching number of the branching vector $(\delta + 1, 1)$; see also Table 2 and Claim 1 below.

Table 2 Branching numbers (rounded up) due to branching rule (B1) and (B2) for some concrete minimum degree δ

δ	3	4	25	199	469	6165	7747
(B1)	1.3803	1.3248	1.0976	1.0199	1.0099	1.0011	1.0009
(B2)	1.3248	1.2208	1.0287	1.0035	1.0015	1.0002	1.0001

Claim 1 For $\delta \geq 3$, the branching number of the branching vector $(\delta, \delta - 1)$ is smaller than the branching number of the branching vector $(\delta + 1, 1)$.

Proof The branching number of the branching vector $(\delta, \delta - 1)$ is the positive root of the polynomial $x^\delta - x - 1$. The branching number of the branching vector $(\delta + 1, 1)$ is the positive root of the polynomial $x^{\delta+1} - x^\delta - 1$. Let $f_\delta(x) = x^{\delta+1} - x^\delta - 1$ and $\lambda_{f,\delta}^*$ be the positive root of $f_\delta(x)$. Let $g_\delta(x) = x^\delta - x - 1$ and $\lambda_{g,\delta}^*$ be the positive root of $g_\delta(x)$. Both functions $f_\delta(x)$ and $g_\delta(x)$ are increasing in $x > 1$ for $\delta > 1$. We show that $g_\delta(\lambda_{f,\delta}^*) \geq 0$. Since $f_\delta(\lambda_{f,\delta}^*) = 0$, we have $g_\delta(\lambda_{f,\delta}^*) = (\lambda_{f,\delta}^*)^\delta - 1/(\lambda_{f,\delta}^*)^\delta - 2$. Solving $x - 1/x - 2 = 0$, the roots are -0.4142 and 2.4142 . We have $x - 1/x - 2 \geq 0$ for $x \geq 2.4142$. Let

$$\lambda_\delta = \begin{cases} \sqrt[\delta]{\lambda_1}, & \delta > 1; \\ 2.4142, & \delta = 1. \end{cases}$$

Since $f_\delta(\lambda_\delta) = \lambda_\delta^{\delta+1} - \lambda_\delta^\delta - 1$ is decreasing in $\delta > 1$ and $f_3(\lambda_3) \leq 0$, we have $\lambda_{f,\delta}^* \geq \lambda_\delta$ for $\delta \geq 3$. Since $x - 1/x - 2 \geq 0$ for $x \geq 2.4142$ and $(\lambda_{f,\delta}^*)^\delta \geq \lambda_\delta^\delta = 2.4142$ for $\delta \geq 3$, we have $g_\delta(\lambda_{f,\delta}^*) \geq 0$ for $\delta \geq 3$ which implies $\lambda_{f,\delta}^* \geq \lambda_{g,\delta}^*$ for $\delta \geq 3$. $\square$

Consequently, the running time of our algorithm for n -vertex graphs with minimum degree $\delta \geq 3$ is $O^*(\lambda^n)$, where λ is the branching number of the branching vector $(\delta + 1, 1)$, that is, λ is the positive root of $x^{\delta+1} - x^\delta - 1 = 0$.

It remains to show that if none of the reduction rules and none of the branching rules is applicable to an instance (G, A, B) then the graph G has a matching cut (X, Y) such that $A \subseteq X$ and $B \subseteq Y$.

In fact, in this case, $(A, V(G) \setminus A)$ is a matching cut of G : First, every vertex in $V(G) \setminus A$ has at most one neighbor in A ; this is clear because none of Reduction Rules (R1) – (R4) is applicable. Second, if some vertex $a \in A$ has at least two neighbors outside A , then no such neighbor is in B (as Reduction Rule (R4) is not applicable). So, a has two non-adjacent neighbors in $F = V(G) \setminus (A \cup B)$, and Branching Rule (B1) or (B2) is applicable, a contradiction. Thus, every vertex in A has at most one neighbor outside A . Therefore, $(A, V(G) \setminus A)$ is a matching cut of G as claimed.

To conclude, we obtain the following result:

Theorem 4 MATCHING CUT can be solved in time $O^*(\lambda^n)$ for n -vertex graphs with minimum degree $\delta \geq 3$, where λ is the positive root of the polynomial $x^{\delta+1} - x^\delta - 1$.

Perhaps, it is important to note that our algorithm labels the vertices of the input graph G by either A or B but never changes the graph G . Hence, the minimum degree δ of G remains unchanged in all subproblems. This is not the case by other minimum degree-based branching algorithms, like those computing a minimum independent set; cf. [7].

4 Polynomial-Time Solvable Cases: Graphs with Very Large Minimum Degree

In this section, we show that MATCHING CUT can be solved in polynomial time on graphs with large enough minimum degree. By Theorem 2, one can hope for polynomial-time results in terms of minimum degree δ only for graphs with $\delta \geq \frac{1}{c}n$ for some fixed constant $c > 1$. Examples of such dense graphs include Dirac graphs and Ore graphs. Graphs G with n vertices and $\delta(G) \geq \frac{1}{2}n$ are called *Dirac graphs*. *Ore graphs* are those graphs such that, for any two non-adjacent vertices x and y , $\deg(x) + \deg(y) \geq n$. In particular, Dirac graphs are a proper subclass of Ore graphs. Dirac and Ore graphs are well-studied classical graph classes, especially in the longest path and cycle community.

Our polynomial-time result is the following theorem.

Theorem 5 *Let $0 < c_1 < 2$ and $c_2 \geq 0$ be two arbitrary fixed constants. MATCHING CUT is solvable in polynomial time for n -vertex graphs, in which any two non-adjacent vertices x and y satisfy $\deg(x) + \deg(y) \geq \frac{1}{c_1}n - c_2$.*

Proof Our strategy is to decide in polynomial time, given two disjoint vertex subsets A and B of the input graph G , if G admits a matching cut separating A and B , i.e., A is in one part and B is in the other part of the matching cut.

First our algorithm applies Reduction Rules (R1) – (R4) mentioned in Sect. 3 to the current instance (in the order of the rules). In addition, we need one new reduction rule; recall from Sect. 3 that $F = V(G) \setminus (A \cup B)$.

(R5) If there are vertices $u, v \in F$ with a common neighbor in A and $|N(u) \cap N(v) \cap F| \geq 2$, then $A := A \cup \{u, v\} \cup (N(u) \cap N(v) \cap F)$.

Reduction Rule (R5) is safe: Assume that (X, Y) is a matching cut of G with $A \subseteq X$ and $B \subseteq Y$. Since $|N(u) \cap N(v) \cap F| \geq 2$, $u, v, N(u) \cap N(v) \cap F$ and the common neighbor $a \in A$ of u and v are contained in a monochromatic $K_{2,q}$ for some $q \geq 3$. Hence $u, v, N(u) \cap N(v) \cap F$ all must belong to X as $a \in A \subseteq X$. Thus, (X, Y) is a matching cut separating $A \cup \{u, v\} \cup (N(u) \cap N(v) \cap F)$ and B . The other direction is obvious: any matching cut of G separating $A \cup \{u, v\} \cup (N(u) \cap N(v) \cap F)$ and B separates A and B , too. We have seen that Reduction Rule (R5) is correct.

Now, let $G = (V, E)$ be an n -vertex graph satisfying the condition in the theorem, and recall that we may assume that G is 2-edge connected. Hence every matching cut (X, Y) of G , if any, must contain at least two edges. Our algorithm will check, for each choice of two edges $a_1b_1, a_2b_2 \in E$, if G has a matching cut containing a_1b_1 and a_2b_2 . To do this we start with $A := \{a_1, a_2\}$, $B := \{b_1, b_2\}$ and apply rules (R1) – (R5) exhaustively. If (R1) is applicable, then G has no matching cut containing a_1b_1 and a_2b_2 . So let us assume that (R1) was never applied and none of (R2), (R3), (R4) and (R5) is applicable.

Then, as (R1) was never applied, b_1 is the only neighbor in B of a_1 and a_2 is the only neighbor in A of b_2 . As (R4) is not applicable, $|A| \geq |N[a_1] \setminus \{b_1\}| = \deg(a_1)$

and $|B| \geq |N[b_2] \setminus \{a_2\}| = \deg(b_2)$. Since a_1 and b_2 are non-adjacent, we therefore have

$$|A| + |B| \geq \frac{1}{c_1}n - c_2. \quad (4)$$

It follows from $|F| = n - |A| - |B|$ and (4) that

$$|F| \leq \left(1 - \frac{1}{c_1}\right)n + c_2. \quad (5)$$

Now, as (R2) is not applicable, every vertex in F has at most one neighbor in A (and at most one neighbor in B). So, if every vertex in A has at most one neighbor in F , then $(A, B \cup F)$ is a matching cut of G separating A and B .

Thus, it remains the case that some vertex $v \in A$ has two neighbors x, y in F . Then x and y are non-adjacent as (R3) is not applicable. Recall that v is the only neighbor in A of x and of y , and each of x and y has at most one neighbor in B . Now, as (R5) is not applicable, $|N(x) \cap N(y) \cap F| \leq 1$, and so we have $|F| \geq |N[x]| + |N[y]| - 4 = \deg(x) + \deg(y) - 2$. Hence

$$|F| \geq \frac{1}{c_1}n + c_2 - 2. \quad (6)$$

By (5) and (6), $\frac{1}{c_1}n + c_2 - 2 \leq \left(1 - \frac{1}{c_1}\right)n + c_2$. Since $0 < c_1 < 2$, we therefore have $n \leq \frac{2c_1}{2-c_1}$. Thus, G has bounded size. Hence, we can determine if G has a matching cut or not in constant time.

Observe that rule (R1) can be applied in constant time and applies at most once. Each of the other rules applies at most n times because it removes at least one vertex from F . Thus, the running time for the application of the rules is roughly $n \cdot O(m) = O(nm)$. Since we have at most $O(m^2)$ many choices for the edges a_1b_1, a_2b_2 , we conclude that it can be decided in time $O(n \cdot m^3)$ if G has a matching cut. $\square$

Consider a graph G with minimum degree $\delta \geq \frac{1}{c}n - c'$, where $1 < c < 4$ and $c' \geq 0$ are given constants. Then, for every two vertices x and y in G , $\deg(x) + \deg(y) \geq 2\delta \geq \frac{1}{c_1}n - c_2$ with $c_1 = c/2 < 2$ and $c_2 = 2c' \geq 0$. Hence Theorem 5 leads to the following result.

Corollary 1 *Let $1 < c < 4$ and $c' \geq 0$ be two fixed constants. MATCHING CUT is solvable in polynomial time for n -vertex graphs with minimum degree at least $\frac{1}{c}n - c'$.*

4.1 Dirac and Ore Graphs

In this subsection, we show that there is a good characterization for Dirac graphs and Ore graphs having a matching cut. It turns out that Ore graphs having a matching cut

are particularly $\frac{1}{2}n$ -regular Dirac graphs. Ore graphs having a matching cut can be characterized as follows.

Theorem 6 *An Ore graph G has a matching cut if and only if G is a co-bipartite graph with a bipartition into cliques A and B such that every vertex in A is adjacent to exactly one vertex in B and vice versa.*

Proof For the non-trivial direction, let (A, B) be a matching cut of G . Observe first that, for each vertex $a \in A$,

$$|A| \geq \deg(a), \text{ with equality if and only if } |N(a) \cap B| = 1.$$

This is because $|A| \geq |N[a]| - 1$, as any vertex $a \in A$ has at most one neighbor in B . Similarly, for each vertex $b \in B$,

$$|B| \geq \deg(b), \text{ with equality if and only if } |N(b) \cap A| = 1.$$

Thus, for every vertex $a \in A$, $|N(a) \cap B| = 1$: if some $a \in A$ has no neighbor in B , then, as G is an Ore graph, we would have the contradiction $n = |A| + |B| > \deg(a) + \deg(b) \geq n$, where $b \in B$ is an arbitrary vertex. Similarly, for every vertex $b \in B$, $|N(b) \cap A| = 1$.

That is, as (A, B) is a matching cut, and every vertex in A has exactly one neighbor in B , every vertex in B has exactly one neighbor in A . Therefore $|A| = |B|$, and $\deg(a) = |A|$ for any $a \in A$ and $\deg(b) = |B|$ for any $b \in B$. In particular, A and B are cliques. $\square$

Since co-bipartite graphs can be recognized in quadratic time, Theorem 6 implies the following result.

Corollary 2 *MATCHING CUT can be solved in quadratic time on Ore graphs.*

5 Conclusion

We showed that, for any constant $c > 1$, MATCHING CUT restricted on graphs with minimum degree c is NP-complete and cannot be solved by a subexponential-time algorithm unless the Exponential-Time Hypothesis fails. We also showed that MATCHING CUT remains NP-complete for graphs with unbounded minimum degree, even for n -vertex bipartite graphs with minimum degree at least $n^{1-\epsilon}$, where $\epsilon > 0$ is any given constant. It would be interesting to clear the complexity of MATCHING CUT for other special classes of graphs with minimum degree at least $n^{1-\epsilon}$ such as graphs with bounded diameter or graphs with large girth.

For n -vertex graphs with minimum degree $\delta \geq 3$, however, we provided an exact branching algorithm to solve MATCHING CUT in $O^*(\lambda^n)$ time, where λ is the positive root of the polynomial $x^{\delta+1} - x^\delta - 1$. For large minimum degree $\delta \geq 469$, the running time is $O^*(1.0099^n)$, and for $\delta \geq 7747$, the running time is $O^*(1.0009^n)$. In fact,

as $\lambda \rightarrow 1$ when $\delta \rightarrow \infty$, we have that MATCHING CUT is solvable in $O^*((1 + \epsilon)^n)$ time for graphs with large minimum degree $\delta = \delta(\epsilon)$ for arbitrary small constant $\epsilon > 0$. This is a very fast exponential-time algorithm for a natural NP-hard problem that cannot be solved in subexponential time (assuming the ETH). So, an interesting question arises whether or not MATCHING CUT restricted on graphs with unbounded minimum degree δ , say $\delta = O(\log n)$, can be solved in subexponential time.

For graphs with very large minimum degree, MATCHING CUT becomes easier. We showed that, for any fixed constants $1 < c < 4$ and $c' \geq 0$, MATCHING CUT is solvable in polynomial time for graphs with minimum degree at least $\frac{1}{c}n - c'$. We leave the question open if MATCHING CUT can be solved in polynomial time for graphs with minimum degree at least $\frac{1}{c}n - c'$, where $c \geq 4$ and $c' \geq 0$ are given constants.

Finally, another interesting open question is whether there is any analogous result of Theorem 5 or Corollary 1 for d -CUT.

Acknowledgements The authors are grateful to a reviewer for meticulous reading and helpful comments and suggestions. The fourth author thanks Ray Li (UNSW Sydney, Australia) for stimulating discussions on branching numbers.

Funding Open Access funding enabled and organized by Projekt DEAL.

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

References

1. Araújo, J., Cohen, N., Giroire, F., Havet, F.: Good edge-labelling of graphs. *Discrete Appl. Math.* **160**(18), 2502–2513 (2012). <https://doi.org/10.1016/j.dam.2011.07.021>
2. Aravind, N.R., Kalyanasundaram, S., Kare, A.S.: On structural parameterizations of the matching cut problem. In: *Combinatorial Optimization and Applications—11th International Conference, COCOA 2017, Shanghai, China, 16–18 December 2017, Proceedings, Part II*, pp. 475–482 (2017). https://doi.org/10.1007/978-3-319-71147-8_34
3. Bonsma, P.S.: The complexity of the matching-cut problem for planar graphs and other graph classes. *J. Graph Theory* **62**(2), 109–126 (2009). <https://doi.org/10.1002/jgt.20390>
4. Borowiecki, M., Jesse-Józefczyk, K.: Matching cutsets in graphs of diameter 2. *Theor. Comput. Sci.* **407**(1–3), 574–582 (2008). <https://doi.org/10.1016/j.tcs.2008.07.002>
5. Chvátal, V.: Recognizing decomposable graphs. *J. Graph Theory* **8**(1), 51–53 (1984). <https://doi.org/10.1002/jgt.3190080106>
6. Farley, A.M., Proskurowski, A.: Networks immune to isolated line failures. *Networks* **12**(4), 393–403 (1982). <https://doi.org/10.1002/net.3230120404>
7. Fedor, V.: *Fomin and Dieter Kratsch. Exact Exponential Algorithms*. Springer, Berlin (2010)
8. Gomes, G.C.M., Sau, I.: Finding cuts of bounded degree: complexity, FPT and exact algorithms, and kernelization. In: *14th International Symposium on Parameterized and Exact Computation, IPEC 2019, 11–13 September 2019, Munich, Germany*, pp. 19:1–19:15 (2019). <https://doi.org/10.4230/LIPIcs.IPEC.2019.19>

9. Graham, R.L.: On primitive graphs and optimal vertex assignments. *Ann. N. Y. Acad. Sci.* **175**(1), 170–186 (1970)
10. Hsieh, S.-Y., Le, H.-O., Le, V.B., Peng, S.-L.: Matching cut in graphs with large minimum degree. In: *Computing and Combinatorics—25th International Conference, COCOON 2019, Xi'an, China, 29–31 July 2019, Proceedings*, pp. 301–312 (2019). https://doi.org/10.1007/978-3-030-26176-4_25
11. Impagliazzo, R., Paturi, R.: On the complexity of k -sat. *J. Comput. Syst. Sci.* **62**(2), 367–375 (2001). <https://doi.org/10.1006/jcss.2000.1727>
12. Impagliazzo, R., Paturi, R., Zane, F.: Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.* **63**(4), 512–530 (2001). <https://doi.org/10.1006/jcss.2001.1774>
13. Komusiewicz, C., Kratsch, D., Le, V.B.: Matching cut: kernelization, single-exponential time FPT, and exact exponential algorithms. *Discrete Appl. Math.* **283**, 44–58 (2020). <https://doi.org/10.1016/j.dam.2019.12.010>
14. Kratsch, D., Le, V.B.: Algorithms solving the matching cut problem. *Theor. Comput. Sci.* **609**, 328–335 (2016). <https://doi.org/10.1016/j.tcs.2015.10.016>
15. Le, H.-O., Le, V.B.: A complexity dichotomy for matching cut in (bipartite) graphs of fixed diameter. *Theor. Comput. Sci.* **770**, 69–78 (2019). <https://doi.org/10.1016/j.tcs.2018.10.029>
16. Le, V.B., Randerath, B.: On stable cutsets in line graphs. *Theor. Comput. Sci.* **1–3**(301), 463–475 (2003). [https://doi.org/10.1016/S0304-3975\(03\)00048-3](https://doi.org/10.1016/S0304-3975(03)00048-3)
17. Moshé, A.M.: Matching cutsets in graphs. *J. Graph Theory* **13**(5), 527–536 (1989). <https://doi.org/10.1002/jgt.3190130502>
18. Patrignani, M., Pizzonia, M.: The complexity of the matching-cut problem. In: *Graph-Theoretic Concepts in Computer Science, 27th International Workshop, WG 2001, Boltenhagen, Germany, 14–16 June 2001, Proceedings*, pp. 284–295 (2001). https://doi.org/10.1007/3-540-45477-2_26
19. Rusu, I., Spinrad, J.P.: Forbidden subgraph decomposition. *Discrete Math.* **247**(1–3), 159–168 (2002). [https://doi.org/10.1016/S0012-365X\(01\)00173-X](https://doi.org/10.1016/S0012-365X(01)00173-X)
20. van Rooij, J.M.M., van Kooten Niekerk, M.E., Bodlaender, H.L.: Partition into triangles on bounded degree graphs. *Theory Comput. Syst.* **52**(4), 687–718 (2013). <https://doi.org/10.1007/s00224-012-9412-5>

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

Affiliations

Chi-Yeh Chen¹ · Sun-Yuan Hsieh¹ · Hoang-Oanh Le² · Van Bang Le³ · Sheng-Lung Peng⁴

Chi-Yeh Chen
chency@csie.ncku.edu.tw

Sun-Yuan Hsieh
hsiehsy@mail.ncku.edu.tw

Hoang-Oanh Le
LeHoangOanh@web.de

Sheng-Lung Peng
slpeng@ntub.edu.tw

¹ Department of Computer Science and Information Engineering, National Cheng Kung University, No. 1, University Road, Tainan 70101, Taiwan

² Berlin, Germany

³ Institut für Informatik, Universität Rostock, Rostock, Germany

⁴ Department of Creative Technologies and Product Design, National Taipei University of Business, Taipei, Taiwan