Theoretical Computer Science 609 (2016) 328–335

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Algorithms solving the Matching Cut problem ☆

Dieter Kratsch [a], Van Bang Le [b,*]

[a] Laboratoire d'Informatique Théorique et Appliquée, Université de Lorraine, 57045 Metz Cedex 01, France
[b] Universität Rostock, Institut für Informatik, Rostock, Germany

ARTICLE INFO

Article history:
Received 30 June 2015
Received in revised form 4 October 2015
Accepted 10 October 2015
Available online 19 October 2015
Communicated by A. Marchetti-Spaccamela

Keywords:
Matching cut
Graph algorithm

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. This paper provides a first branching algorithm solving MATCHING CUT in time $O^*(2^{n/2}) = O^*(1.4143^n)$ for an n-vertex input graph, and shows that MATCHING CUT parameterized by the vertex cover number $\tau(G)$ can be solved by a single-exponential algorithm in time $2^{\tau(G)}O(n^2)$. Moreover, the paper also gives a polynomially solvable case for MATCHING CUT which covers previous known results on graphs of maximum degree three, line graphs, and claw-free graphs.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

In a graph $G = (V, E)$, a *cut* is a partition $V = A \cup B$ of the vertex set into disjoint, nonempty sets A and B, written (A, B). The set of all edges in G having an endvertex in A and the other endvertex in B, also written (A, B), is called the *edge cut* of the cut (A, B). A *matching cut* is an edge cut that is a matching. Note that, by our definition, the empty edge cut is a matching cut, but a matching whose removal disconnects the graph need not be a matching cut.

In [13], Farley and Proskurowski studied matching cuts in graphs in the context of network applications. Patrignani and Pizzonia [28] pointed out an application of matching cuts in graph drawing.

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?

This paper deals with algorithms solving the MATCHING CUT problem.

Previous results and related work. Graphs admitting a matching cut were first discussed by Graham in [16] under the name *decomposable graphs*. The first complexity results for MATCHING CUT have been obtained by Chvátal, who proved in [8] that MATCHING CUT is NP-complete, even when restricted to graphs of maximum degree four, and polynomially solvable for graphs of maximum degree three. In fact, Chvátal's reduction works even for $K_{1,4}$-free graphs of maximum degree four.

☆ A preliminary version of this paper appeared as an extended abstract in the proceedings of CIAC 2015 ([20]).
* Corresponding author.
E-mail addresses: kratsch@univ-metz.fr (D. Kratsch), van-bang.le@uni-rostock.de (V.B. Le).

http://dx.doi.org/10.1016/j.tcs.2015.10.016
0304-3975/© 2015 Elsevier B.V. All rights reserved.

Being not aware of Chvátal's result, Patrignani and Pizzonia [28] gave another NP-completeness proof for Matching Cut. Later, by modifying Chvátal's reduction, Le and Randerath [23] proved that Matching Cut remains NP-complete for bipartite graphs with one color class consisting only of vertices of degree three and the other color class consisting only of vertices of degree four. Using a completely different reduction, Bonsma proved the NP-hardness of Matching Cut for planar graphs of maximum degree four and for planar graphs of girth five [2].

Besides the case of maximum degree three mentioned above, it has been shown that Matching Cut can be solved in polynomial time for line graphs[1] and for graphs without induced cycles of length at least five (Moshi [26]), for claw-free graphs (Bonsma [2]), for cographs and graphs of bounded tree-width or clique-width (Bonsma [2]),[2] and for graphs of diameter two (Borowiecki and Jesse-Józefczyk [3]).

A closely related problem to Matching Cut is that of deciding if a given graph admits a stable cutset. Here, a *stable cutset* in a graph $G = (V, E)$ is a stable set $S \subseteq V$ such that $G - S$ is disconnected. It can be seen that, for graphs G with minimum degree at least two, G has a matching cut if and only if the line graph $L(G)$ admits a stable cutset. For information on applications and algorithmic results on stable cutsets we refer to [4–7,9,19,22–24,29].

Our contributions. First, we provide a new polynomially solvable case for Matching Cut, namely for graphs without induced $K_{1,4}$ and $K_{1,4}+e$ (the graph obtained from $K_{1,4}$ by adding an edge). Thus, extending and unifying Chvátal's results for graphs of maximum degree three, Moshi's results for line graphs, and Bonsma's results for claw-free graphs. Second, we provide, for the first time, an exact branching algorithm for Matching Cut that has time complexity $O^*(2^{n/2})$.[3] Third, we initiate the study of matching cuts from the viewpoint of parameterized complexity. We show that Matching Cut is fixed-parameter tractable when parameterized by the vertex cover number $\tau(G)$. Much stronger, we establish a single-exponential algorithm running in time $2^{\tau(G)}O(n^2)$.

Notation and terminology. Let $G = (V, E)$ be a graph with vertex set $V(G) = V$ and edge set $E(G) = E$. We assume that a (input) graph has n vertices and m edges. A *stable set* (a *clique*) in G is a set of pairwise non-adjacent (adjacent) vertices. The neighborhood of a vertex v in G, denoted by $N_G(v)$, is the set of all vertices in G adjacent to v; if the context is clear, we simply write $N(v)$. Set $\deg(v) = |N(v)|$, the degree of the vertex v. For a subset $W \subseteq V$, $G[W]$ is the subgraph of G induced by W, and $G - W$ stands for $G[V \setminus W]$. We write $N_W(v)$ for $N(v) \cap W$ and call the vertices in $N(v) \cap W$ the W*-neighbors* of v. A *vertex cover* of G is a subset $C \subseteq V$ such that every edge of G has at least one endvertex in C, i.e., $V \setminus C$ is a stable set in G. The vertex cover number of G, denoted by $\tau(G)$, is the smallest size of a vertex cover of G.

The complete graph and the cycle on n vertices is denoted by K_n and C_n, respectively; K_3 is also called a *triangle*. The tree on $t+1$ vertices with t leaves is denoted by $K_{1,t}$; $K_{1,3}$ is also called a *claw*. The graph obtained from $K_{1,t}$ by adding a new edge is denoted by $K_{1,t}+e$.

When an algorithm branches on the current instance of size n into 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 $x^n - x^{n-t_1} - x^{n-t_2} - \dots - x^{n-t_r} = 0$, written $\tau(t_1, t_2, \dots, t_r)$, is called its *branching number*. The running time of the branching algorithm is $O^*(\alpha^n)$, where $\alpha = \max_i \alpha_i$ and the maximum is taken over all branching rules. Furthermore for every i, α_i is the branching number of branching rule i. We refer to [15] for more details on exact branching algorithms.

Parameterized complexity deals with NP-hard problems whose instances come equipped with an additional integer parameter k. The objective is to design algorithms whose running time is $f(k) \cdot \mathrm{poly}(n)$ for some computable function f. Problems admitting such algorithms are called *fixed-parameter tractable*. See [11,14,27] for more information.

The paper is organized as follows. In Section 2 we show that Matching Cut can be solved in polynomial time for graphs without induced $K_{1,4}$ and $K_{1,4}+e$. In Section 3 we describe our branching algorithm and point out that Matching Cut does not admit a subexponential time algorithm, unless the exponential time hypothesis fails. In Section 4 we describe a single-exponential algorithm for Matching Cut when parameterized by the vertex cover number.

2. A polynomially solvable case of Matching Cut

In this section we will unify the known polynomially solvable cases for Matching Cut on graphs of maximum degree three, on line graphs, and on claw-free graphs, by proving the following theorem.

Theorem 1. *There is an algorithm solving* Matching Cut *for* $(K_{1,4}, K_{1,4}+e)$*-free graphs in time* $O(mn)$.

Proof. Let G be a $(K_{1,4}, K_{1,4}+e)$-free graph. Since forests and cycles of length at least 4 have matching cuts, we may assume that G properly contains a cycle. Let C be a shortest cycle in G.

If C is of length at least 5, then every vertex v of $G - V(C)$ has at most one neighbor in C (otherwise $vC[v_i, v_j]v$ or $vC[v_j, v_i]v$ would be a shorter cycle than C, where v_i and v_j are two neighbors of v on C and $C[v_i, v_j]$ and $C[v_j, v_i]$ are

[1] The line graph of a graph G is the graph whose vertices correspond to the edges of G, and two vertices are adjacent iff the corresponding edges have a common endvertex in G.

[2] We note that Matching Cut can be expressed in MSOL; see also [2].

[3] We use the O^* notation which suppresses polynomial factors.

Download English Version:

https://daneshyari.com/en/article/10333869

Download Persian Version:

https://daneshyari.com/article/10333869

Daneshyari.com