Discrete Applied Mathematics 226 (2017) 10–16

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

An algorithm for identifying cycle-plus-triangles graphs

Kristóf Bérczi [a,*], Yusuke Kobayashi [b]

[a] MTA-ELTE Egerváry Research Group, Eötvös University, Budapest, Hungary

[b] University of Tsukuba, Tsukuba, Japan

ARTICLE INFO

Article history:
Received 3 December 2016
Accepted 18 April 2017
Available online 17 May 2017

Keywords:
Cycle-plus-triangles graph
C_4-free 2-matching
3-colouring

ABSTRACT

The union of n node-disjoint triangles and a Hamiltonian cycle on the same node set is called a cycle-plus-triangles graph. Du, Hsu and Hwang conjectured that every such graph has independence number n. The conjecture was later strengthened by Erdős claiming that every cycle-plus-triangles graph has a 3-colouring, which was verified by Fleischner and Stiebitz using the Combinatorial Nullstellensatz. An elementary proof was later given by Sachs. However, these proofs are non-algorithmic and the complexity of finding a proper 3-colouring is left open. As a first step toward an algorithm, we show that it can be decided in polynomial time whether a graph is a cycle-plus-triangles graph. Our algorithm is based on revealing structural properties of cycle-plus-triangles graphs.

1. Introduction

An undirected, not necessarily simple graph $G = (V, E)$ on $3n$ nodes is called a *cycle-plus-triangles (CPT) graph* if it is the edge-disjoint union of a Hamiltonian cycle and n node-disjoint triangles. In [2], Du, Hsu and Hwang conjectured that such graphs have independence number n. A stronger statement was proposed by Erdős claiming that every CPT graph has a 3-colouring (see e.g. [9]). Fleischner and Stiebitz verified [5] the conjecture in an even more general form by showing that the list-chromatic number of a CPT graph is 3, that is, for every assignment of 3-length lists to the nodes, there exists a proper colouring giving each node a colour from its list. Their proof is based on the Combinatorial Nullstellensatz of Alon and Tarsi [1], hence does not yield any algorithm for finding a proper 3-colouring. An elementary, but still non-algorithmic proof for 3-colourability was later given by Sachs [8]. The complexity of finding a proper 3-colouring is left open.

Generalizations of CPT graphs are also well investigated. A subset $M \subseteq E$ of edges is called a *2-matching* if $d_M(v)$, the *degree* of v in M, is at most 2 for all $v \in V$. If equality holds for every node, then M is called a *2-factor*. We call M *spanning* if $d_M(v) \geq 1$ for all $v \in V$. Clearly, a 2-matching is the node-disjoint union of paths and cycles. A 2-matching is *C_4-free* if it does not contain a cycle of length at most four. A *2-factor-plus-triangles (FPT) graph* is the edge-disjoint union of two 2-regular graphs G_1 and G_2 on the same node set, where G_2 consists of triangles. As a natural generalization of the original conjecture, Erdős asked in [4] whether an FPT graph is 3-colourable whenever G_1 is C_4-free. The conjecture was answered negatively in [6,7]. Examples in [3] show that an FPT graph on $3n$ nodes may not have an independent set of size n either.

The motivation of our investigations was to find a proper 3-colouring of CPT graphs algorithmically. However, the problem has two natural formulations depending on whether a decomposition of G into G_1 and G_2 is given in the input or not. This distinction motivates the following subproblem, which is the focus of the present paper.

* Corresponding author.
E-mail addresses: berkri@cs.elte.hu (K. Bérczi), kobayashi@sk.tsukuba.ac.jp (Y. Kobayashi).

http://dx.doi.org/10.1016/j.dam.2017.04.021

CPT Graph Recognition Problem

Input: A 4-regular graph G on $3n$ nodes.

Output: A decomposition of G into a Hamiltonian cycle G_1 and n node-disjoint triangles G_2, if such a decomposition exists.

We are going to show that this problem can be solved efficiently.

Theorem 1. *The* CPT Graph Recognition Problem *can be solved in polynomial time.*

Theorem 1 implies that the complexity of the two variants of the colouring problem is the same. The theorem will be proved in a more general form by introducing a further generalization of FPT (and thus CPT) graphs. We call a graph a *2-matching-plus-triangles (MPT) graph* if it is the edge-disjoint union of two 2-matchings G_1 and G_2 on the same node-set, where G_1 is a spanning 2-matching and G_2 consists of node-disjoint triangles. We consider a restricted counterpart of the recognition problem for MPT graphs.

C_4-free MPT Graph Recognition Problem

Input: A graph G of maximum degree at most 4.

Output: A decomposition of G into a C_4-free spanning 2-matching G_1 and node-disjoint union of triangles G_2, if such a decomposition exists.

Note that, in both problems, G is not necessarily simple. The main contribution of the paper is the following result, which also implies Theorem 1.

Theorem 2. *The* C_4-free MPT Graph Recognition Problem *can be solved in polynomial time. Moreover, if such a decomposition exists then G_1 and G_2 are uniquely determined up to isomorphism.*

Our algorithm is based on revealing structural properties of MPT graphs. First we identify a set of obstacles whose presence in G immediately shows that no proper decomposition exists. Then we introduce a set of reduction steps for fixing some of the triangles in G_2. The so-called triangle graph of the reduced graph has surprisingly well-structured properties: it is a simple graph without isolated nodes having only three types of components. We hope that these observations may lead to a polynomial time algorithm for finding a proper 3-colouring of the graph, if such a colouring exists.

Throughout the paper we use the following notations. Given a graph G, its *node* and *edge sets* are denoted by $V(G)$ and $E(G)$, respectively. The *set of triangles* in G is denoted by $\mathcal{T}(G)$. We omit the graph from the notation if it is clear from the context. The *triangle graph of* G is another graph $R(G)$ defined as follows: each node of $R(G)$ represents a triangle in $\mathcal{T}(G)$ and there are as many edges between two nodes as the number of edges shared by the corresponding triangles.

The rest of the paper is organized as follows. In Section 2, we identify certain subgraphs whose presence in G either fixes some of the triangles in G_2 or shows that the graph is not an MPT graph with G_1 being C_4-free. The algorithm for solving the C_4-free MPT Graph Recognition Problem is presented in Section 3. Finally, Section 4 concludes the paper with further remarks on the proof.

2. Preprocessing the graph

In this section G_1 and G_2 refer to the output of the C_4-free MPT Graph Recognition Problem. Recall that G_1 has to be a spanning 2-matching, hence the triangles in G_2 have to cover exactly the degree 3 and 4 nodes of G. In what follows, first a list of obstacles that make the decomposition impossible is given in Section 2.1, then a series of reduction steps is presented in Section 2.2 that fix some of the triangles in G_2.

2.1. Obstacles

The first three obstacles are coming from the fact that G_1 has to be spanning and G_1 contains no short cycles.

Obstacle 1 (*Loops*)**.** If G has a loop then no proper decomposition exists as G_1 has to be C_4-free and G_2 may contain only triangles.

Obstacle 2 (*Short Cycle Component*)**.** Since G_1 is spanning, a connected component of G consisting of a single cycle has to be also a component of G_1. If such a cycle has length at most 4, then G_1 cannot be chosen to be C_4-free.

Obstacle 3 (*Isolated Triangle With a Low Degree Node*)**.** Let $T \in \mathcal{T}$ be a triangle with $V(T) = \{u, v, w\}$ such that $d_G(w) = 2$, and there is no other triangle containing u and v simultaneously. As G_1 has to be spanning and w has degree two, both edges of w belong to G_1, that is, T cannot be added to G_2. As no other triangle contains u and v, T has to be in G_1, hence G_1 cannot be chosen to be C_4-free.

The presence of parallel edges does not necessarily mean that no proper decomposition exists. The cases not mentioned in the following description are discussed in Reduction 1.