Egerváry Research Group
on Combinatorial Optimization

Technical reportS

TR-2016-12. Published by the Egerváry Research Group, Pázmány P. sétány 1/C, H–1117, Budapest, Hungary. Web site: www.cs.elte.hu/egres . ISSN 1587–4451.

An algorithm for identifying cycle-plus-triangles graphs

Kristóf Bérczi and Yusuke Kobayashi

July 2016

An algorithm for identifying cycle-plus-triangles graphs

Kristóf Bérczi* and Yusuke Kobayashi†

Abstract

The union of n node-disjoint triangles and a Hamiltonian cycle on the same node set is called a cycle-plus-triangles graph. In [2], 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 [4] using the Combinatorial Nullstellensatz. An elementary proof was later given in [7]. 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. We hope that these observations may also help to find a proper 3-colouring in polynomial time.

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 statment was proposed by Erdős claiming that every CPT graph has a 3-colouring (see eg. [8]). Fleischner and Stiebitz verified [4] 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-colorability was later given by Sachs [7]. 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

*MTA-ELTE Egerváry Research Group, Budapest, Hungary. E-mail: `berkri@cs.elte.hu`.
†University of Tsukuba, Tsukuba, Japan. E-mail: `kobayashi@sk.tsukuba.ac.jp`.

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 whether an FPT graph is 3-colourable whenever G_1 is C_4-free. The conjecture was answered negatively in [5, 6]. Examples show [3] 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.

<u>CPT Graph Recognition Problem</u>
Input: A 4-regular graph G on $3n$ nodes.
Output: A decomposition of G into a Hamiltonian cycle and n node-disjoint triangles, if such a decomposition exists.

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

Theorem 1.1. *The* CPT Graph Recognition Problem *can be solved in polynomial time.*

Theorem 1.1 implies that the complexity of the two variants of the colouring problem are 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.

<u>C_4-free MPT Graph Recognition Problem</u>
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 G is not assumed to be simple. The main contribution of the paper is the following result, which also implies Theorem 1.1.

Theorem 1.2. *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 deep structural properties of MPT graphs. We hope that these observations may lead to a polynomial time algorithm for finding a proper 3-colouring of the graph, if 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 $R(G)$ that represents the adjacencies between the triangles of G, that is, 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 fix 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

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, 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.

Obstacle 4 (Parallel edges)**.** Assume that there are multiple edges between nodes u and v in G.

Obstacle 4.1. If the number of parallel edges is at least three, then no proper decomposition exists as both G_1 and G_2 may contain at most one of these edges.

Obstacle 4.2. If there are two parallel edges between u and v, then both G_1 and G_2 have to contain one of them. If there is no triangle $T \in \mathcal{T}$ with $u, v \in V(T)$, then this immediately shows that G cannot be decomposed into G_1 and G_2.

Obstacle 4.3. Assume that there is a unique node $w \in V$ for which $\{u, v, w\}$ is the node-set of a triangle $T \in \mathcal{T}$. If there are two parallel edges between any two of these nodes, then G_1 has to contain one edge from each pair. These edges together form a triangle, showing that G_1 cannot be chosen to be C_4-free.

Obstacle 4.4. Finally assume that there are two different nodes $w_1, w_2 \in V$ for which $\{u, v, w_i\}$ form the node-set of a triangle in $\mathcal{T}$. One of these triangles has to be in G_1, showing again that G_1 cannot be chosen to be C_4-free.

The reason for adding the restriction on G_1 to be C_4-free in the C_4-free MPT Graph Recognition Problem is that the following obstacle can be excluded from the graph. As we will see, this easy observation greatly simplifies the structure of the triangle graph of G.

Obstacle 5 (Edge in three triangles)**.** Let $uv \in E$ be an edge which is contained in three triangles with distinct node sets (Figure 1a). As G_2 may contain at most one of these triangles, G_1 has to contain the remaining edges of the other two. These edges form a cycle of length 4, showing that G_1 is not C_4-free.

Finally, two further obstacles are defined that cannot be subgraphs of a properly decomposable graph.

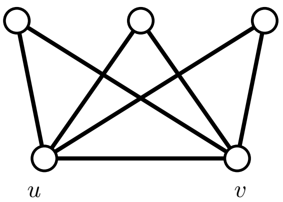
Obstacle 6 (Complete graph on 4 nodes)**.** Assume that G contains a complete graph on four nodes. G_1 is a 2-matching, hence G_2 either contains a triangle of this subgraph or uses two independent edges from it. In both cases, the remaining edges of the complete graph belong to G_1. However, in the first case the remaining edges result a node of degree at least 3 while in the second case they form a cycle of length 4, both forbidden in G_1.

Obstacle 7 (4-wheel)**.** Assume that G contains a spanned 4-wheel (Figure 1b), that is, the only triangles covering x, y, z or w are T_1, T_2, T_3 and T_4. Then these four nodes cannot be covered by node-disjoint triangles of G, hence no proper decomposition exists.

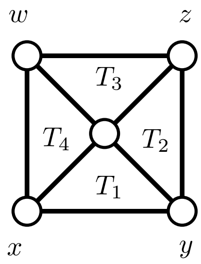
Note that it can be checked in polynomial time whether a given graph $G = (V, E)$ contains any of Obstacles 1-7.

2.2 Reductions

Now we turn our attention to reductions. In each case, a set of triangles in $\mathcal{T}$ is identified that, if a proper partition exists, has to be included in G_2. These triangles are added to G_2 and their edge sets are deleted from G. The first reduction concludes the case of parallel edges.



(a) Edge in 3 triangles



(b) 4-wheel

Figure 1: Examples for obstacles

Reduction 1 (Reduction on parallel edges)**.** The only case that is not covered by Obstacles 4.1-4.4 is when the number of parallel edges between u and v is 2, there is a unique node $w \in V$ for which $\{u, v, w\}$ is the node-set of a triangle, and there is a single edge between u and w or v and w. In this case fix an arbitrary triangle $T \in \mathcal{T}$ with $V(T) = \{u, v, w\}$ and add it to G_2.

The next two steps ensure that no isolated node will appear in the triangle graph of the reduced graph.

Reduction 2 (Reduction on uniquely covered node)**.** Let $v \in V$ be a node for which there is a unique triangle $T \in \mathcal{T}$ with $v \in V(T)$. If $d_G(v) \geq 3$, then G_2 should cover v, hence we can add T to G_2.

Reduction 3 (Reduction on H_1)**.** Assume that H_1 (Figure 2) is a subgraph of G. All three of the nodes u, v and w have to be covered by triangles in G_2. If these nodes are covered by T_2, T_3 and T_4 in G_2 then T_1 lies completely in G_1, which is not possible by C_4-freeness. Hence T_1 has to be in G_2.

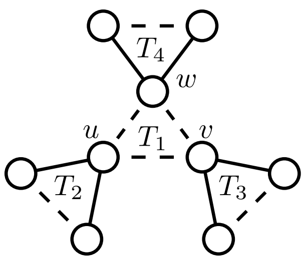
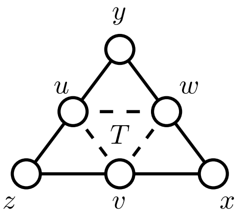


Figure 2: Reduction of an H_1 subgraph

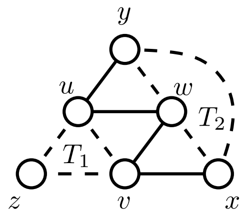
The last reduction step is crucial in simplifying the structure of the triangle graph of G.

Reduction 4 (Reduction on H_2)**.** Assume that H_2 (Figure 3a) is a subgraph of G. We distinguish several cases depending on the number of edges spanned by $\{x, y, z\}$.

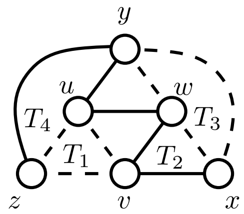
Consider first the case when $\{x, y, z\}$ spans no edges. As all three of the nodes u, v and w have to be covered by G_2, T has to be added to G_2.



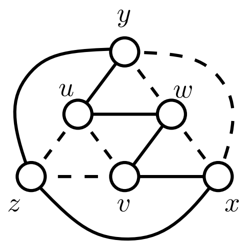
(a) $\{x, y, z\}$ spans no edge



(b) $\{x, y, z\}$ spans a single edge



(c) $\{x, y, z\}$ spans two edges



(d) $\{x, y, z\}$ spans three edges

Figure 3: Reductions of an H_2 subgraph

Assume now that $\{x, y, z\}$ spans a single edge, say $xy \in E$. Assume further that the only common neighbour of x and y is w (Figure 3b). All of the nodes u, v, w, x and y have to be covered by G_2, which is only possible by adding T_1 and T_2 to G_2. The case when x and y shares a common neighbour different from w is discussed later on.

If $\{x, y, z\}$ spans two edges, say $xy, yz \in E$ (Figure 3c), then either T_1 and T_3 or T_2 and T_4 have to be added to G_2. We can chose arbitrarily from these two choices as both pairs of triangles cover the whole node-set of the subgraph and the remaining part is a path of length 5 between x and z in both cases.

Finally, assume that $xy, yz, xz \in E$ (Figure 3d). As the maximum degree in G is at most 4, this subgraph form a connected component of G. In order to cover u, v, w, x, y and z, we need to add two node-disjoint triangles of the component to G_2. We can chose these triangles arbitrarily as the remaining part is a cycle of length 6 for all choices.

Note that it can be checked in polynomial time whether any of Reductions 1-4 can be applied for a given graph $G = (V, E)$.

3 Proof of Theorem 1.2

Let $G = (V, E)$ be an undirected graph with maximum degree at most 4. Check if the actual graph contains any of Obstacles 1-7. If the answer is yes, then the algorithm halts concluding that no proper decomposition exists. Otherwise check if any of Reductions 1-4 can be applied to the graph. If a reduction is possible, update the subgraph $G_2 = (V, E_2)$ accordingly by adding the fixed triangles to it and delete their edges from E.

Repeat these two steps as long as no more reduction is possible. Then let $G' = (V, E')$ denote the graph where $E' = E - E_2$ and let $\mathcal{T}' \subseteq \mathcal{T}$ denote the set of triangles in G'.

Claim 3.1. *G' is a simple graph.*

Proof. By Obstacles 1 and 4 and Reduction 1, G' cannot contain loops or parallel edges. □

Recall that $R(G')$ denotes the triangle graph of G'. Clearly, $R(G')$ can be constructed in polynomial time.

Claim 3.2. *$R(G')$ is a simple graph.*

Proof. By the definition of triangle graphs, $R(G')$ does not contain loops. If a pair of parallel edges appear in $R(G')$ then G' contains two triangles sharing two edges. That is, the node sets of the triangles are the same. As they have to differ in their third edges, these edges form a pair of parallel edges in G', contradicting Claim 3.1. □

Claim 3.3. *$R(G')$ has no isolated nodes.*

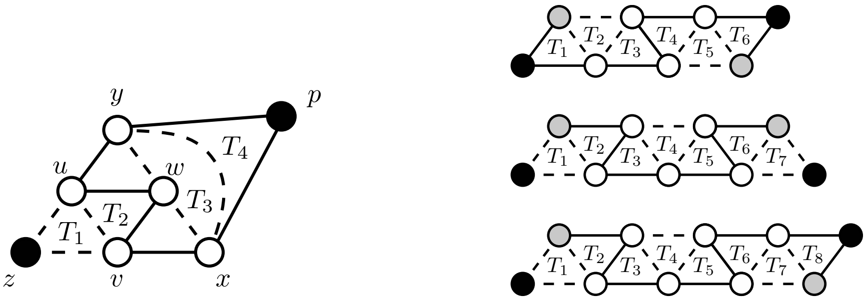
Proof. An isolated node of $R(G')$ corresponds to a triangle $T \in \mathcal{T}'$ sharing no edge with other triangles of G'. Let $V(T) = \{v_1, v_2, v_3\}$. By Obstacle 3, $d_{G'}(v_i) \geq 3$. By Reduction 2, there exist triangles T_1, T_2 and T_3 different from T such that $v_i \in V(T_i)$. As the node corresponding to T is an isolated node in $R(G')$, T_1, T_2 and T_3 are node-disjoint. But then T together with these triangles form a H_1 graph in G', contradicting Reduction 3. □

Note that not all H_2 subgraphs of G are eliminated during the reduction procedure. The single case when a H_2 subgraph of G survives is when $\{x, y, z\}$ spans a single edge, say $xy \in E$, and x and y have a common neighbour p different from w (Figure 4a). We call such a subgraph a B_0 **block**. Note that, by the degree constraints, the triangle graph of a B_0 block forms a connected component of $R(G')$.

Claim 3.4. *The maximum degree in $R(G')$ is at most 3. If a node of $R(G')$ has degree 3 then the corresponding triangle is contained in a B_0 block.*

Proof. By Obstacle 5, each edge of G is contained in at most two triangles, hence the degree of a node in $R(G')$ is at most 3.

Take a node of $R(G')$ with degree 3. Such a node corresponds to a triangle T in G' such that all of its edges are contained in other triangles. By Claim 3.2, these triangles are distinct, say T_1, T_2 and T_3. By Obstacle 6, G' does not contain a complete graph on four nodes, hence T_1, T_2 and T_3 are pairwise edge-disjoint. But then T together with T_1, T_2 and T_3 form a H_2 subgraph of G', which is only possible if they are part of a B_0 block. □



(a) B_0 block (0 exit and 2 end nodes) (b) Blocks having 2 exit and 2 end nodes

(c) Block having 2 exit and 0 end nodes (d) Block having 0 exit and 0 end nodes

Figure 4: Triangulated blocks (end nodes are black, exit nodes are grey)

Now we are ready to describe the structure of G'. By Claims 3.2-3.4, each connected component of $R(G')$ is a path (containing at least 2 nodes), a cycle (of length at least 5 by Obstacles 6 and 7), or the triangle graph of a B_0 block. A **triangulated block** is a subgraph of G' obtained by taking the union of the triangles corresponding to the node set of one of the connected components of $R(G')$. The **length** of the block is the number of its triangles, that is, the number of nodes of the corresponding component of the triangle graph. Apart from B_0 blocks, we will denote the triangles of a triangulated block of length q by $T_1, \dots, T_q$ in this order along the path or cycle of $R(G')$ the block belongs to. Note that the triangulated blocks are pairwise edge-disjoint and together contain all triangles of G'.

A triangulated block may contain only nodes of degree 2, 3 and 4. The degree 2 nodes are called the **end nodes** of the block, the degree 3 nodes are called the **exit nodes** of the block. Note that a block has either 0 or 2 end nodes, and 0 or 2 exit nodes (Figure 4). Since the maximum degree of G' is at most 4, if a connected component of $R(G')$ is a path, then the corresponding triangulated block is as in Figure 4b. When two end nodes coincide in Figure 4b, the triangulated block can be as in Figure 4c as a variant. If a connected component of $R(G')$ is a cycle, then the corresponding triangulated block is as in Figure 4d because of the degree bound.

The following observations show that a triangulated block gives a strong restriction on the set of possible triangles that can be added to G_2.

Observation 1. Assume that G' has a B_0 block (Figure 4a). Then either T_1 and T_3 or T_2 and T_4 have to be in G_2 to cover nodes u, v, w, x and y. The remaining edges of the block form a cycle of length 6.

Observation 2. Assume that G' has a triangulated block of length q with 2 exit nodes and 2 end nodes (Figure 4b). If q is congruent to 0 modulo 3, then $T_2, T_5, \dots, T_{q-1}$

have to be added to G_2 to cover the nodes of degree at least 3 in the block. If q is congruent to 1 modulo 3, then $T_1, T_4, \dots, T_q$ have to be added to G_2 for the same reason. If q is congruent to 2 modulo 3, then we have two possible choices: we can add either $T_1, T_4, \dots, T_{q-1}$ or $T_2, T_5, \dots, T_q$ to G_2. The remaining edges of the block form a path between its exit nodes in all cases.

Observation 3. Assume that G' has a triangulated block of length q with 2 exit nodes and no end nodes (Figure 4c). If q is congruent to 0 or 1 modulo 3, then there is no proper subset of triangles that cover the nodes of degree at least 3 in the block. If $q \geq 5$ and is congruent to 2 modulo 3, then we have two possible choices: we can add either $T_1, T_4, \dots, T_{q-1}$ or $T_2, T_5, \dots, T_q$ to G_2. In this case the remaining edges of the block form a path between its exit nodes.

Observation 4. Assume that G' has a triangulated block of length q with no exit and end nodes (Figure 4d). If q is congruent to 1 or 2 modulo 3, then there is no proper subset of triangles that cover the nodes of degree at least 3 in the block. If $q \geq 6$ and is congruent to 0 modulo 3, then we have three possible choices: we can add either $T_1, T_4, \dots, T_{q-2}$ or $T_2, T_5, \dots, T_{q-1}$ or $T_3, T_6, \dots, T_q$ to G_2. In this case the remaining edges of the block form a cycle.

The previous observations show that the length of a block modulo 3 plays an important role. We say that a non-B_0 triangulated block of length q is of **Type i** if q is congruent i modulo 3. Blocks isomorphic to B_0 are considered to be of Type 2.

Two blocks may meet only at their end nodes, thus the triangulated blocks of G' form 'paths' and 'cycles'. More precisely, let us call a connected component of the union of the triangulated blocks a **triangulated block component**. A component is called **linear** if it has two nodes of degree 2, otherwise it is called **circular**. A circular component is **open** if it has degree 3 nodes, otherwise it is **closed**. Note that triangulated blocks having no end nodes (Figures 4c and 4d) form circular triangulated block components.

The followings are direct corollaries of Observations 1-4.

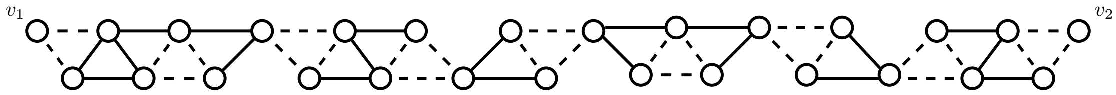
Corollary 3.5. *Take a linear triangulated block component of G' and let v_1 and v_2 denote its degree 2 nodes (Figure 5a).*

If none of v_1 and v_2 has to be covered by G_2, then a proper set of triangles can be added to G_2 if and only if the Type 0 and 1 blocks of the component are alternating along the path, possibly separated by blocks of Type 2.

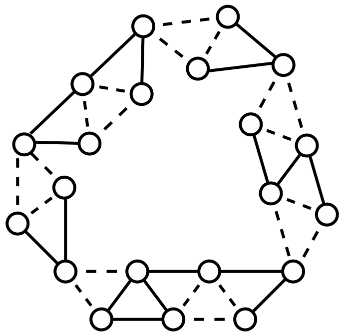
If only v_1 has to be covered by G_2, then a proper set of triangles can be added to G_2 if and only if the Type 0 and 1 blocks of the component are alternating along the path from v_1 to v_2 starting with a block of Type 1, possibly separated by blocks of Type 2.

If both v_1 and v_2 have to be covered by G_2, then a proper set of triangles can be added to G_2 if and only if the Type 0 and 1 blocks of the component are alternating along the path starting and ending with blocks of Type 1, possibly separated by blocks of Type 2.

Corollary 3.6. *Take a circular triangulated block component of G'. If the component is closed, then the component is either a single triangulated block and Observation 4*



(a) Linear triangulated block component (Types of the blocks are 2-1-2-0-2-1)



(b) Circular triangulated block component (Types of the blocks are 2-2-1-2-0)

Figure 5: Triangulated block components

applies (that is, a proper set of triangles can be added to G_2 if and only if the block is of Type 0*), or it consists of B_0 blocks in which case, according to Observation 1, there are two possible choices for the set of triangles to be added to G_2.*

If the component is open, then a proper set of triangles can be added to G_2 if and only if the Type 0 *and* 1 *blocks of the component are alternating around the cycle, possibly separated by blocks of Type* 2 *(Figure 5b).*

It can be checked in polynomial time whether the triangulated block components of G' satisfy the conditions of Corollaries 3.5 and 3.6. If any of the components fails to have a proper subset of triangles, then no proper decomposition exists and the algorithm stops.

Otherwise let $G_1 = (V, E_1)$ denote the graph obtained from $G' = (V, E')$ by deleting the edges of the triangles added to G_2 in order to cover the nodes of the blocks. We still have to check whether G_1 is a C_4-free spanning 2-matching. This can be done easily by checking the degrees of the nodes and the connected components of G_1, thus finishing the description of the algorithm.

It remains to show that the decomposition, if exists, is unique up to isomorphism. This follows from the fact that whenever we add triangles to G_2 as a reduction on some subgraph or while covering the block components, the remaining edges of the underlying structure form a cycle of fixed length or a path of fixed length with fixed end nodes, independently from the choice of the triangles where multiple choices are possible. This concludes the proof of the Theorem 1.2.

4 Conclusion

The strict structure of the triangle graph of the reduced graph revealed that the possible choices for G_2 are very limited. At each reduction step and for each triangulated block component we have at most three possible choices for the set of triangles to be added to G_2. Moreover, these choices are completely independent from each other.

These observations imply that several extensions of the C_4-free MPT Graph Recognition Problem can be solved in polynomial time. For example, we can restrict the set of triangles that can be used in G_2 to a subset of $\mathcal{T}(G)$. We can also consider an optimization version of the problem when cost functions $c_1 : E(G) \to \mathbb{R}$ and $c_2 : \mathcal{T}(G) \to \mathbb{R}$ are given, and the aim is to find a decomposition maximizing $\sum_{e \in E(G_1)} c_1(e) + \sum_{T \in \mathcal{T}(G_2)} c_2(T)$.

Our hope is that the recognition of this restricted structure may help us to find a 3-colouring of the graph, if exists, algorithmically. The difficulty lies in the reduction steps: even if a 3-colouring of the reduced graph is available, it is not clear how to transform it to a proper colouring of the original graph.

Acknowledgement

The first author were supported by the Hungarian Scientific Research Fund - OTKA, K109240. The second author was supported by JST, ERATO, Kawarabayashi Large Graph Project and by KAKENHI Grant Numbers 24106002, 24700004.

References

[1] N. Alon and M. Tarsi. Combinatorial nullstellensatz. *Combinatorics Probability and Computing*, 8(1):7–30, 1999.

[2] D. Du, D. Hsu, and F. Hwang. The hamiltonian property of consecutive-d digraphs. *Mathematical and Computer Modelling*, 17(11):61–63, 1993.

[3] D.-Z. Du and H. Q. Ngo. An extension of dhh-erd os conjecture on cycle-plus-triangle graphs. *Taiwanese Journal of Mathematics*, 6(2):pp–261, 2002.

[4] H. Fleischner and M. Stiebitz. A solution to a colouring problem of P. Erdős. *Discrete Mathematics*, 101(1):39–48, 1992.

[5] H. Fleischner and M. Stiebitz. Some remarks on the cycle plus triangles problem. In *The Mathematics of Paul Erdös II*, pages 136–142. Springer, 1997.

[6] H. Fleischner and M. Stiebitz. Some remarks on the cycle plus triangles problem. In *The Mathematics of Paul Erdős II*, pages 119–125. Springer, 2013.

[7] H. Sachs. Elementary proof of the cycle-plus-triangles theorem. *Cahiers du GERAD*, 1994.

[8] D. West. Open problems. *SIAM Activity Group Newsletter in Discrete Mathematics*, 2:3, 1991.