



HAL
open science

Decomposing Subcubic Graphs into Claws, Paths or Triangles

Laurent Bulteau, Guillaume Fertin, Anthony Labarre, Romeo Rizzi, Irena Rusu

► **To cite this version:**

Laurent Bulteau, Guillaume Fertin, Anthony Labarre, Romeo Rizzi, Irena Rusu. Decomposing Subcubic Graphs into Claws, Paths or Triangles. *Journal of Graph Theory*, 2021, 98 (4), pp.557-588. 10.1002/jgt.22713 . hal-03388424

HAL Id: hal-03388424

<https://hal.science/hal-03388424v1>

Submitted on 20 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Decomposing Subcubic Graphs into Claws, Paths or Triangles*

Laurent Bulteau Guillaume Fertin Anthony Labarre Romeo Rizzi
Irena Rusu

October 20, 2021

Abstract

Let $S = \{K_{1,3}, K_3, P_4\}$ be the set of connected graphs of size 3. We study the problem of partitioning the edge set of a graph G into graphs taken from any non-empty $S' \subseteq S$. The problem is known to be NP-complete for any possible choice of S' in general graphs. In this paper, we assume that the input graph is subcubic (i.e. all its vertices have degree at most 3), and study the computational complexity of the problem of partitioning its edge set for any choice of S' . We identify all polynomial and NP-complete problems in that setting.

Keywords— decomposition, edge partition, NP-completeness, subcubic graph

1 Introduction

Given a graph G and a set S of graphs, the S -DECOMPOSITION problem asks whether G can be represented as an edge-disjoint union of subgraphs, each of which is isomorphic to a graph in S . The problem has a long history that can be traced back to Kirkman [13] and has been intensively studied ever since, both from pure mathematical and algorithmic points of view. One of the most notable results in the area is the answer of Dor and Tarsi [7] to the long-standing “Holyer conjecture”, which stated that “the problem of edge-partitioning a graph into subgraphs isomorphic to a fixed graph H is NP-complete for all graphs H with at least 3 edges” [12]. Dor and Tarsi note that this conjecture as stated is false, and prove that a slightly modified version holds:

Theorem 1.1. [7] *For every graph H , the $\{H\}$ -DECOMPOSITION problem is NP-complete whenever H contains a connected component with three edges or more.*

Theorem 1.1 leaves open the cases where all connected components of H contain at most two edges, which were eventually all shown to be tractable

Theorem 1.2. *Let $s, t \in \mathbb{N}$. There is a polynomial time algorithm to decide if an instance graph has an $\{sP_2 \cup tP_3\}$ -decomposition (see Lonc [15] for the case $s = 0$, Priesler and Tarsi [20] for the case $s = 1$ and $t \geq 1$, and Bryś and Lonc [4] for all other cases).*

Note that single-vertex components can be safely ignored, provided the input graph is large enough, since we only consider edge-disjoint (not vertex-disjoint) decompositions. Many variants of the S -DECOMPOSITION problem have been studied while attempting to prove Holyer’s conjecture or to obtain polynomial-time algorithms in restricted cases [21], and applications arise in such diverse fields as traffic grooming [19] and graph drawing [11]. In particular, Dyer and Frieze [8] studied a variant where S is the set of all connected graphs with k edges for some natural k , and proved the NP-completeness of the S -DECOMPOSITION problem for any $k \geq 3$, even under the assumption that the input graph is planar and bipartite (see Theorem 3.1 in [8]). They further claimed that the problem remains NP-complete under the additional constraint that

*This is an extended version of Bulteau et al. [5].

all vertices of the input graph have degree 2 or 3. Interestingly, if $k = 3$ and G is a bipartite *cubic* graph (i.e., each vertex has degree 3), then G can clearly be decomposed in polynomial time, using $K_{1,3}$'s only, by selecting either part of the bipartition and making each vertex in that set the center of a $K_{1,3}$. This shows that focusing on the case $k = 3$ and on cubic graphs can lead to tractable results — as opposed to general graphs, for which when $k = 3$, and for any non-empty $S' \subseteq S$, the S' -DECOMPOSITION problems all turn out to be NP-complete [8, 12].

In this paper, we study the S -DECOMPOSITION problem on subcubic graphs in the case where S is the set of all connected graphs on three edges, i.e., $S = \{K_{1,3}, K_3, P_4\}$ (respectively known as claws, triangles and paths). For any non-empty $S' \subseteq S$, we settle the computational complexity of the corresponding S' -DECOMPOSITION problem. More specifically, we give polynomial-time algorithms for:

- decomposing strictly subcubic graphs into claws, triangles, or both claws and triangles (Propositions 3.2, 3.3 and 3.6);
- decomposing cubic graphs using only claws (Proposition 4.3; triangles are trivially unusable on their own, and a polynomial-time approach to path decompositions was previously known) or both paths and triangles (Proposition 4.2);

and we show that all other decomposition problems on strictly subcubic graphs are NP-complete (Theorems 3.1 and 3.5 and Propositions 3.4 and 3.7), even if the input graph is planar and bipartite. Furthermore, we prove that deciding whether a cubic graph admits a decomposition into paths and claws (and possibly triangles) is an NP-complete problem (Theorem 5.6), even if the input graph is K_3 -free and planar.

Table 1 summarises the state of knowledge regarding the complexity of decomposing subcubic and arbitrary graphs using connected subgraphs of size 3, and puts our results into perspective. Without loss of generality, the input graph is assumed to be connected; otherwise, separate results can be used on different components depending on whether they are strictly subcubic or cubic.

| Allowed subgraphs | | | Complexity according to graph class | | |
|-------------------|-------|-------|-------------------------------------|---------------------------|------------------------------|
| $K_{1,3}$ | K_3 | P_4 | strictly subcubic | cubic | arbitrary |
| ✓ | | | in P (Proposition 3.2) | in P (Proposition 4.3) | NP-complete [8, Theorem 3.5] |
| | ✓ | | in P (Proposition 3.3) | $O(1)$ (impossible) | NP-complete [12] |
| | | ✓ | NP-complete (Theorem 3.1) | in P [14] | NP-complete [8, Theorem 3.4] |
| ✓ | | | in P (Proposition 3.6) | in P (Proposition 4.6) | NP-complete [8, Theorem 3.5] |
| ✓ | ✓ | | NP-complete (Theorem 3.5) | NP-complete (Theorem 5.6) | NP-complete [8, Theorem 3.1] |
| | ✓ | ✓ | NP-complete (Proposition 3.4) | in P (Proposition 4.2) | NP-complete [8, Theorem 3.4] |
| ✓ | ✓ | ✓ | NP-complete (Proposition 3.7) | NP-complete (Theorem 5.6) | NP-complete [8, Theorem 3.1] |

Table 1: Known complexity results on decomposing graphs using subsets of $\{K_{1,3}, K_3, P_4\}$.

A large portion of the results in this paper first appeared in a conference version [5]. The main additions and differences are:

- all results on strictly subcubic graphs (Section 3);
- explicit algorithms (Algorithms 1 to 3) and complexity analyses for the polynomial time solvable cases (see Table 1);
- a sharper characterisation of hard instances for the problems of decomposing cubic graphs in the case where $\{K_{1,3}, P_4\} \subseteq S'$: we now have a proof that those problems are hard on *planar* graphs, whereas the previous versions of those results only assumed K_3 -freeness (see Theorem 5.6);
- and some simplifications and generalisations of previous versions of our results.

2 Notation and definitions

We follow Brandstädt et al. [3] for notation and terminology. All graphs we consider are simple and nontrivial (i.e. $|V(G)| \geq 2$ and $|E(G)| \geq 1$). A *cubic* graph contains only vertices of degree 3, while a *strictly subcubic* graph contains at least one vertex of degree strictly less than 3; a *subcubic* graph may be either cubic or strictly subcubic.

Definition 2.1. Given a set S of graphs, a graph G *admits an S -decomposition*, or is *S -decomposable*, if $E(G)$ can be partitioned into subgraphs, each of which is isomorphic to a graph in S .

Throughout the paper, S denotes the set of connected graphs of size 3, i.e. $S = \{K_{1,3}, K_3, P_4\}$. We study the following problem:

S' -DECOMPOSITION

Input: a subcubic graph $G = (V, E)$, a non-empty set $S' \subseteq S$.

Question: does G admit an S' -decomposition?

During the process of building a decomposition D for a graph G , it will sometimes be convenient to distinguish between *covered* edges, i.e. edges that belong to some element of D , and *uncovered* edges, which do not belong to any element of D .

Definition 2.2. Given a graph $G = (V, E)$, *removing* a subgraph $H = (W \subseteq V, F \subseteq E)$ of G consists in removing edges in F from G as well as the possibly resulting isolated vertices. We use the notation $G \setminus H$ to denote the graph obtained from G by removing H . For a subset $U \subseteq V$, we use the lighter notation $G \setminus U$ instead of $G \setminus (U, \emptyset)$.

We let $G[U]$ denote the subgraph of G induced by $U \subseteq V(G)$. Recall that a graph G is *H -free* if there is no subset U of $V(G)$ such that $G[U]$ is isomorphic to a given graph H . If G is not H -free, we say that G *contains* (a copy of) H .

Definition 2.3. Let G and G' be two graphs:

- *subdividing* an edge $\{u, v\} \in E(G)$ consists in inserting a new vertex w into that edge, so that $V(G)$ becomes $V(G) \cup \{w\}$ and $E(G)$ is replaced with $(E(G) \setminus \{u, v\}) \cup \{u, w\} \cup \{w, v\}$;
- *attaching G' to a vertex $u \in V(G)$* means building a new graph H by identifying u and some vertex $v \in V(G')$;
- *attaching G' to an edge $e \in E(G)$* consists in subdividing e using a new vertex w , then attaching G' to w .

Figure 1 illustrates the process of attaching an edge to an edge of the cube graph, which yields what we refer to as a *buoy* graph.

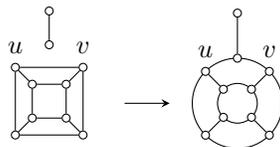


Figure 1: Attaching a new edge to $\{u, v\}$.

We shall use $N(v)$ to denote the *open neighbourhood* of a vertex v in a graph, i.e. the set that consists of all vertices adjacent to v , as opposed to its *closed neighbourhood* which we denote $N[v] = N(v) \cup \{v\}$. We use $E(v)$ to denote the set of all edges that are incident with vertex v .

3 Decomposing strictly subcubic graphs

In this section, we assume that the graph G we seek to decompose is strictly subcubic. Our hardness results will follow the strategies presented by Dyer and Frieze [8], who use the following well-known NP-complete problem [9]:

PLANAR EXACT COVER BY 3-SETS (PLANAR X3C)
 Input: a set W and a set of triplets $T \subseteq W \times W \times W$ such that the bipartite graph $G = (W \cup T, E)$ is planar, where $E = \cup_{t=(r,b,y) \in T} \{\{r, t\}, \{b, t\}, \{y, t\}\}$.
 Question: is there a subset $T' \subseteq T$ which contains all elements of W exactly once?

PLANAR X3C remains NP-complete under the following restriction [9], which we will assume holds for our reductions: each element of W appears in two or three elements of T . We will find it convenient in our reductions to view instances of PLANAR X3C immediately as planar embeddings of the bipartite graph G as described in the problem definition.

3.1 Decompositions with only one type of graph

We start our study of S' -DECOMPOSITION on strictly subcubic graphs with only P_4 's at our disposal. We show that this first problem is NP-complete.

Theorem 3.1. *The $\{P_4\}$ -DECOMPOSITION problem is NP-complete for strictly subcubic planar bipartite graphs.*

Proof. We transform a planar embedding of an instance G of PLANAR X3C into a new graph G' by replacing each vertex w in W with one of the two *vertex gadgets* described below, depending on its degree. Since we assume that w has degree either 2 or 3, we replace w with H_2 in the former case or with H_3 in the latter. The subgraph of G induced by $N[t]$, for each triplet t in T , is mapped onto a *triplet gadget* H^t attached to three gadgets denoted by $H(z)$, where $H(z)$ is either gadget H_2 or H_3 depending again on the degree of z in G . Vertices v_1 and v_2 in H_2 (and v_3 in the case of H_3) will be used for attaching each gadget to H^t , so the notation v_r (resp. v_b and v_y) corresponds to one of the v_i 's in $H(r)$ (resp. $H(b)$ and $H(y)$). Figure 2 illustrates the transformation

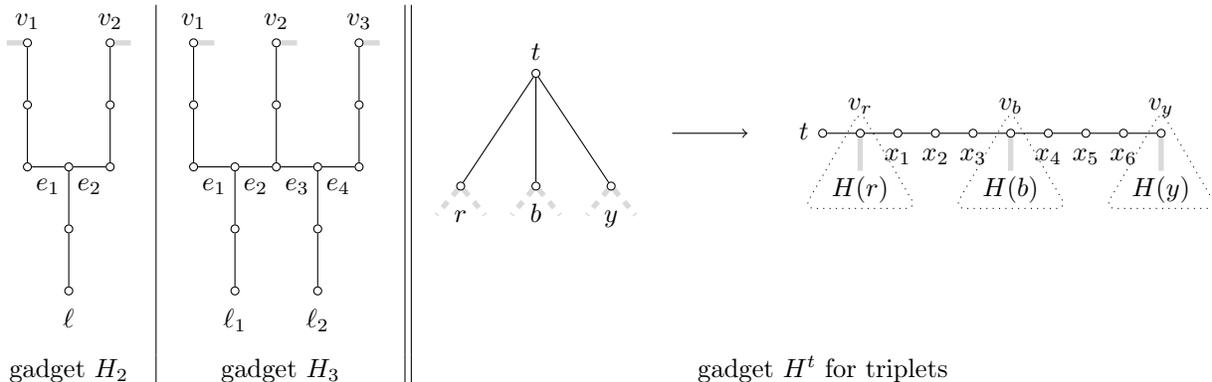


Figure 2: The gadgets used in the proof of Theorem 3.1.

The resulting graph G' is clearly strictly subcubic. We now show that G admits an exact cover by 3-sets if and only if G' admits a $\{P_4\}$ -decomposition.

\Rightarrow : we map every triplet t in an instance G of PLANAR X3C onto three P_4 's in the decomposition D that we build for G' , in a way that depends solely on whether or not t has been selected as part of a solution T' to G . More specifically, as Figure 3 illustrates:

1. if t belongs to the solution T' , then the P_4 's $G'[\{v_r, x_1, x_2, x_3\}]$ and $G'[\{v_b, x_4, x_5, x_6\}]$ are added to D , as well as the P_4 's that have two edges in their attached gadget and exactly one edge in $G'[\{t, v_r, x_1, x_2, x_3, v_b, x_4, x_5, x_6, v_y\}]$;
2. if t does not belong to the solution T' , then $G'[\{t, v_r, x_1, x_2, x_3, v_b, x_4, x_5, x_6, v_y\}]$ decomposes into three P_4 's $G'[\{t, v_r, x_1, x_2\}]$, $G'[\{x_2, x_3, v_b, x_4\}]$ and $G'[\{x_4, x_5, x_6, v_y\}]$, which are added as such to D .

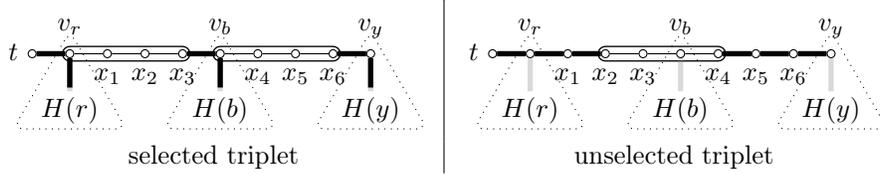


Figure 3: Expressing triplet selection using P_4 's in gadget H^t in the proof of the forward direction of Theorem 3.1.

To lighten the discussion, we will refer to a P_4 in D with only two edges in a vertex gadget as a *partial* P_4 . Note that since T' is an exact cover by 3-sets, every vertex gadget contains exactly one partial P_4 , which must involve one of its vertices with label v_1 or v_2 (or v_3 for H_3).

We now show that the choices we make for each gadget H^t completely determine the rest of D , by examining their impact on the decomposition of the attached gadgets. We distinguish between the following cases, illustrated in Figure 4:

- (a) (H_2): if v_1 is the vertex that belongs to the partial P_4 in H_2 (case v_2 is symmetric), then v_2 must be the starting point of a P_4 which contains edge e_2 , which in turn forces us to select the P_4 that starts with ℓ and contains e_1 .
- (b) (H_3): if v_1 is the vertex that belongs to the partial P_4 in H_3 (case v_3 is symmetric), then the P_4 that starts with ℓ_1 must end with e_1 . This in turn forces the P_4 that starts with v_2 to end with e_2 , which then forces the P_4 that starts with ℓ_2 to end with e_3 ; and finally, this last choice forces the selection of the P_4 that starts with v_3 and ends with e_4 .
- (c) (H_3): finally, if v_2 is the vertex that belongs to the partial P_4 in H_3 , then this forces the selection of both the P_4 from v_1 that ends with e_1 and the P_4 from v_3 that ends with e_4 . These two choices then force us to select e_2 (resp. e_4) as the final edge of the P_4 that starts with ℓ_1 (resp. ℓ_2).

Figure 4 illustrates the above three cases. Arrows indicate implications, i.e. the fact that selecting a certain P_4 forces us to select another P_4 . As a result, a $\{P_4\}$ -decomposition for G' can straightforwardly be derived from any exact cover by 3-sets for G .

\Leftarrow : to prove that a $\{P_4\}$ -decomposition for G' yields an exact cover by 3-sets for G , we first observe that each gadget's structure limits our choices for a valid $\{P_4\}$ -decomposition. The following claims characterise how our gadgets might be decomposed.

Claim 1. *If G' admits a $\{P_4\}$ -decomposition D , then D contains exactly one partial P_4 for each gadget H_2 and H_3 which must cover either v_1 or v_2 (or v_3 for H_3).*

Proof. Every $\{P_4\}$ -decomposition of H_2 must contain a P_4 that starts with its only leaf ℓ and contains either e_1 or e_2 . Choosing either edge for that P_4 forces us to build another P_4 that will contain the other edge and end with either v_2 or v_1 ; the uncovered vertex can then be covered by a partial P_4 using an additional edge from the triplet gadget to which H_2 is attached.

The case analysis is similar in the case of H_3 : every $\{P_4\}$ -decomposition must contain two P_4 's that start with ℓ_1 and ℓ_2 , respectively, and two edges out of $\{e_1, e_2, e_3, e_4\}$ will then be selected to end

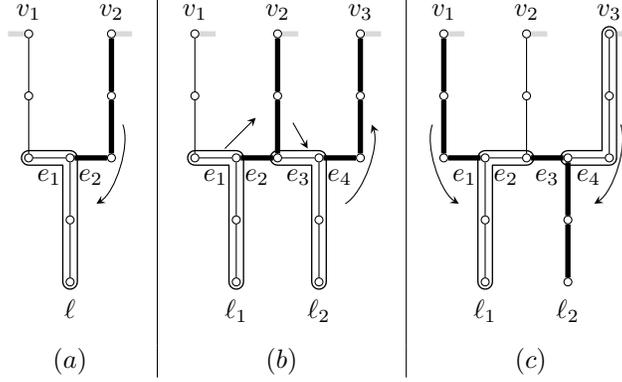


Figure 4: Expressing triplet selection using P_4 's in gadgets H_2 and H_3 in the proof of the forward direction of Theorem 3.1.

these two P_4 's. Figure 5 illustrates the following four cases, with arrows indicating implications, i.e. the fact that selecting a certain P_4 forces us to select another P_4 .

- (a) Selecting e_1 and e_3 forces us to build a P_4 that contains e_2 and ends with v_2 and another P_4 that contains e_4 and ends with v_3 . In that case, v_1 is left uncovered, and covering it with a P_4 will require the use of its other incident edge which does not belong to H_3 .
- (b) Selecting e_1 and e_4 leads to a partial decomposition which cannot be completed: any P_4 that ends with v_2 will leave out either e_2 or e_3 , which therefore cannot be part of a P_4 .
- (c) Selecting e_2 and e_3 forces us to build a P_4 that contains e_1 and ends with v_1 and another P_4 that contains e_4 and ends with v_3 . In that case, v_2 is left uncovered, and covering it with a P_4 will require the use of its other incident edge in the triplet gadget to which H_3 is attached.
- (d) Selecting e_2 and e_4 forces us to build a P_4 that contains e_1 and ends with v_1 and another P_4 that contains e_3 and ends with v_2 . In that case, v_3 is left uncovered, and covering it with a P_4 will require the use of its other incident edge in the triplet gadget to which H_3 is attached.

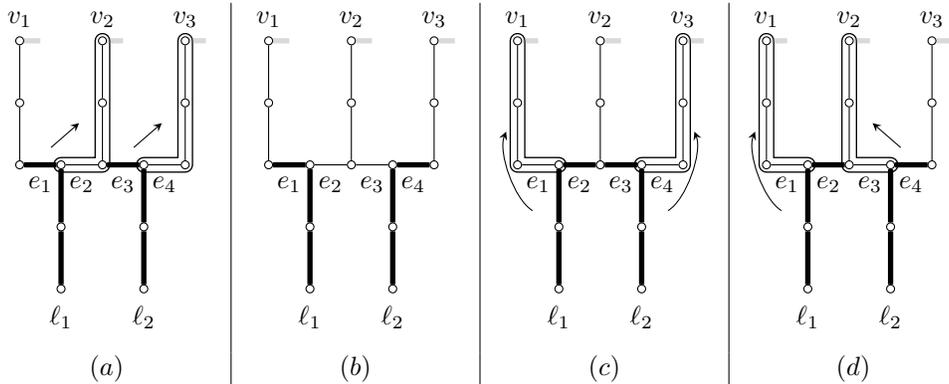


Figure 5: The four cases in the proof of Claim 1 (reverse direction of Theorem 3.1).

(end of Claim 1)

□

Claim 2. If G' admits a $\{P_4\}$ -decomposition D , then for every gadget H^t in G' :

1. either $G'[\{t, v_r, x_1, x_2, x_3, v_b, x_4, x_5, x_6, v_y\}]$ decomposes into three P_4 's that belong to D , or

2. edges $\{t, v_r\}$, $\{x_3, v_b\}$ and $\{x_6, v_y\}$ are the terminal edges of partial P_4 's in D that respectively start in $H(r)$, $H(b)$, and $H(y)$.

Proof. Claim 1 implies that $H(r)$, $H(b)$ and $H(y)$ each contain exactly one partial P_4 ; if a partial P_4 is incident with H^t , then it must be completed using an edge of G' $[\{t, v_r, x_1, x_2, x_3, v_b, x_4, x_5, x_6, v_y\}]$. We show that all three vertex gadgets attached to H^t must be in the same state, i.e. either v_r , v_b and v_y all belong to partial P_4 's in $H(r)$, $H(b)$ and $H(y)$, or none of them do:

- * if v_r belongs to a partial P_4 in D , then its last edge must be $\{t, v_r\}$, which forces the selection in D of the P_4 induced by $\{v_r, x_1, x_2, x_3\}$. The P_4 in $H(b)$ that contains v_b must be partial, otherwise the number of remaining edges to cover in H^t is not a multiple of 3 and they can therefore not be covered regardless of the state of $H(y)$. That partial P_4 must contain $\{x_3, v_b\}$, which in turn forces the selection in D of the P_4 induced by $\{v_b, x_4, x_5, x_6\}$ and thereby also the selection of a partial P_4 in $H(y)$ containing $\{x_6, v_y\}$.
- * if v_r does not belong to a partial P_4 in D , then D must contain the P_4 induced by $\{t, v_r, x_1, x_2\}$. The P_4 in D that contains x_2 , x_3 and v_b cannot end with an edge in $H(b)$, so its last edge is $\{v_b, x_4\}$ and $H(b)$ contains no partial P_4 involving v_b . This in turn forces the P_4 induced by $\{x_4, x_5, x_6, v_y\}$ to belong to D , and therefore $H(y)$ has no partial P_4 involving v_y .

(end of Claim 2) □

We can now show how to transform any $\{P_4\}$ -decomposition D for G' into an exact cover by 3-sets for G : every vertex gadget contains exactly one partial P_4 (Claim 1) which is completed using an edge of the triplet gadget to which it is attached in order to obtain a P_4 , say P' , of D . In other words, P' “points” to the triplet that must be selected to solve the original instance G of PLANAR X3C. Claim 2 guarantees that either all elements of a triplet or none of them will be selected, and Claim 1 ensures that no element can be covered by two different triplets. Therefore, the solution we deduce from D is indeed an exact cover by 3-sets.

We now have two properties left to prove:

1. G' is **bipartite**: since the gadgets we introduced are trees, they cannot create any new cycle in G' , but they will lengthen cycles that were already present in G in two ways.
 - (a) Since vertices in W are replaced by a copy of H_2 or H_3 , a P_3 with edges $\{t_a, w\}$ and $\{w, t_b\}$ in a cycle of G will now enter gadget $H(w)$ through some v_i and exit it through some $v_j \neq v_i$. Depending on the degree of w and the specific entry and exit points in $H(w)$, this adds either 6 or 8 edges to the cycle.
 - (b) Since vertices in T are replaced by a copy of H^t , a P_3 with edges $\{w_c, t\}$ and $\{t, w_d\}$ in a cycle of G will now enter gadget H^t through some v_i and exit it through some $v_j \neq v_i$. Depending on the specific entry and exit points in H^t , this adds either 4 or 8 edges to the cycle.

In all cases, all cycles in G are even to start with, and become longer but remain even in G' . Therefore, G' is also bipartite.

2. G' is **planar**: let us identify G with any of its planar embeddings. Again, since each gadget is a tree, each gadget itself admits a planar embedding; and since no two edges connecting two different elements of W to either the same triplet or different triplets in G cross, the replacement of these elements with our gadgets will not create any crossing in G' either. □

We now move on to the case of $\{K_{1,3}\}$ -decompositions, which Algorithm 1 solves in polynomial time. We prove its correctness below.

Proposition 3.2. *The $\{K_{1,3}\}$ -DECOMPOSITION problem for strictly subcubic connected graphs is in P.*

Proof. Since the input graph G is strictly subcubic, it contains a vertex v of degree 1 or 2 whose neighbour(s) must be the center of a $K_{1,3}$ if a $\{K_{1,3}\}$ -decomposition exists for G . Since G is connected, the graph G' obtained from G by removing that $K_{1,3}$ is also strictly subcubic, and therefore G admits a $\{K_{1,3}\}$ -decomposition if and only if G' does. The decomposition, if it exists, can be found by repeating this claw-removal strategy until either all edges have been removed, in which case we have found a $\{K_{1,3}\}$ -decomposition, or no new claw center can be found, in which case no $\{K_{1,3}\}$ -decomposition exists. \square

Algorithm 1 implements the strategy described in the proof of Proposition 3.2 in an iterative way. It starts by pushing all vertices of degree strictly less than 3 onto a stack \mathcal{S} , then keeps popping vertices and pushing neighbours of the newly created claws in the decomposition, which takes $O(1)$ time. Its time complexity is $O(|V| + |E|) = O(|V|)$, since G is strictly subcubic.

Algorithm 1: STRICTLYSUBCUBICCLAWDECOMPOSITION(G)

Input: A graph G whose connected components are all strictly subcubic.

Output: A $K_{1,3}$ -decomposition of G if one exists, \emptyset otherwise.

```

1  $D \leftarrow \emptyset$ ;
2  $\mathcal{S} \leftarrow$  stack with all vertices of degree  $< 3$  in  $G$ ;
3 while  $\mathcal{S} \neq \emptyset$  do
4    $x \leftarrow \mathcal{S}.\text{pop}()$ ;
5   if  $\text{deg}(x) \neq 0$  then
6      $v \leftarrow$  arbitrary neighbour of  $x$  with degree 3;
7     if no such  $v$  exists then
8       return  $\emptyset$ ;
9      $D \leftarrow D \cup E(v)$ ;           //  $v$  must be the center of a  $K_{1,3}$ , which we add to  $D$ 
10     $\mathcal{S}.\text{push}(\text{vertices in } N(v))$ ; // push vertices whose degree will decrease
11     $G \leftarrow G \setminus \{v\}$ ;       // remove recorded  $K_{1,3}$ 
12 return  $D$ ;
```

Connectedness plays a minor role in our algorithms, since the decomposition problems we study can be solved independently on each connected component. However, we will need to use Algorithm 1 when we decompose cubic graphs, and we will not be able to guarantee that the input to Algorithm 1 will be connected. This can only be an issue for cubic components, since we have not yet explained how to decompose them, which is why we require that all connected components in the input to Algorithm 1 be strictly subcubic. We conclude this section with the easiest case: decomposing strictly subcubic graphs using only K_3 's.

Proposition 3.3. *The $\{K_3\}$ -DECOMPOSITION problem for strictly subcubic graphs is in P.*

Proof. Straightforward: a strictly subcubic graph admits a $\{K_3\}$ -decomposition if and only if all its connected components are K_3 's. \square

3.2 Decompositions with two types of graphs

We now examine decomposition problems on strictly subcubic graphs using two types of graphs, starting with the following hardness result.

Proposition 3.4. *The $\{K_3, P_4\}$ -decomposition problem is NP-complete for strictly subcubic planar bipartite graphs.*

Proof. Follows directly from the proof of Theorem 3.1, since our reduction produces a bipartite graph which is therefore K_3 -free so only P_4 's can be used in a decomposition. \square

The second hardness result of this subsection requires a more involved proof but which is similar in spirit to that of Theorem 3.1.

Theorem 3.5. *The $\{K_{1,3}, P_4\}$ -decomposition problem is NP-complete for strictly subcubic planar bipartite graphs.*

Proof. We transform a planar embedding of an instance G of PLANAR X3C by replacing $G[N[w]]$ for each w in W with either of the gadgets shown in Figure 6, depending on its degree. Since we assume that the degree of w is either 2 or 3, we replace it with H_2 in the former case or with H_3 in the latter.

The resulting graph G' is clearly strictly subcubic. We now show that G admits an exact cover by 3-sets if and only if G' admits a $\{K_{1,3}, P_4\}$ -decomposition.

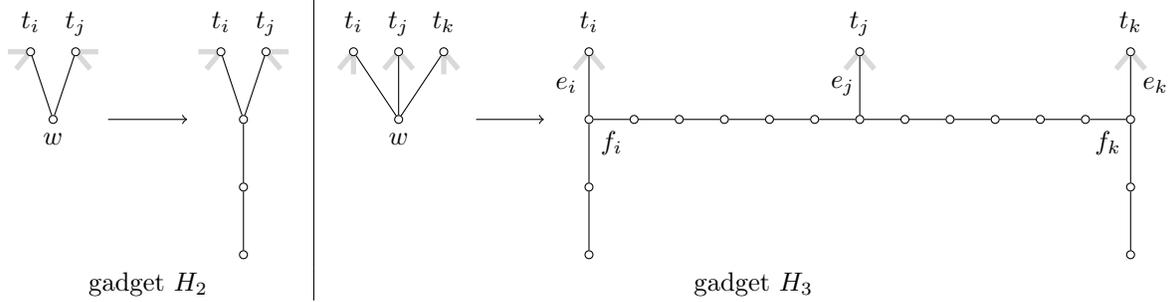


Figure 6: The gadgets used in the proof of Theorem 3.5.

\Rightarrow : every triplet in a solution T' to the instance G of PLANAR X3C naturally induces a $K_{1,3}$ which will be part of the $\{K_{1,3}, P_4\}$ -decomposition D of G' . Since every element w of W is covered exactly once by T' , exactly one edge incident to w in G is part of a $K_{1,3}$ and therefore exactly one edge incident to either t_i or t_j (or t_k if w has degree 3) in the corresponding gadget in G' is part of a $K_{1,3}$ in D .

Therefore, we only need to show that the remaining edges of each gadget admit a $\{K_{1,3}, P_4\}$ -decomposition. If w has degree 2, then that decomposition consists of a single P_4 . For the case where w has degree 3, we have two cases to distinguish: either e_i is unavailable (the case e_k is symmetric), or e_j is; in both cases, the rest of the gadget admits a $\{P_4\}$ -decomposition, as shown below. Figure 7 illustrates both cases, with arrows indicating implications, i.e. the fact that selecting a certain P_4 forces us to select another P_4 .

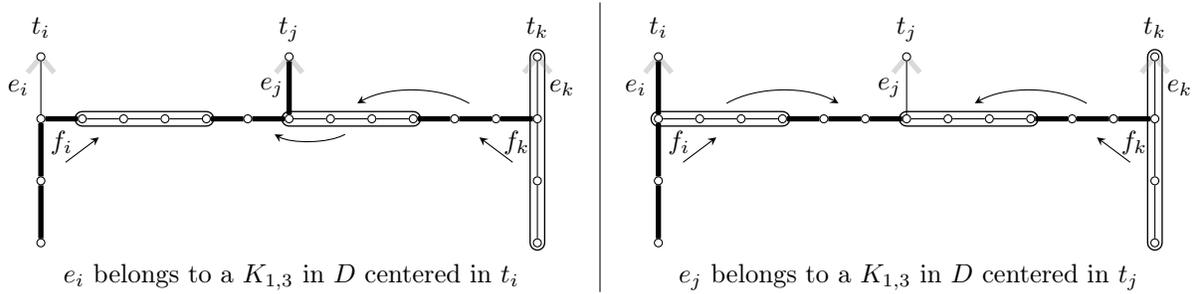


Figure 7: Propagating triplet selection using P_4 's in gadgets H_2 and H_3 in the proof of the forward direction of Theorem 3.5.

\Leftarrow : to prove that a $\{K_{1,3}, P_4\}$ -decomposition for G' yields an exact cover by 3-sets for G , we first observe that each gadget's structure limits our choices for a valid $\{K_{1,3}, P_4\}$ -decomposition.

Claim 3. *If G' admits a $\{K_{1,3}, P_4\}$ -decomposition D , then no $K_{1,3}$ in D is a subgraph of a gadget.*

Proof. A $K_{1,3}$ appearing as a subgraph of either gadget must use a vertex of degree 3 of that gadget as its center. This clearly cannot happen for gadget H_2 , or for the vertices adjacent to t_i and t_k in gadget H_3 , since in all these cases allowing those vertices to be the center of a $K_{1,3}$ in D would leave a pendant edge uncovered. The remaining case to handle is that of the vertex incident to t_j in gadget H_3 , where such a choice would force the partial decomposition of the gadget shown in Figure 8: whichever choice we make for building the remaining P_4 's in the decomposition necessarily leaves four edges of the gadget uncovered.

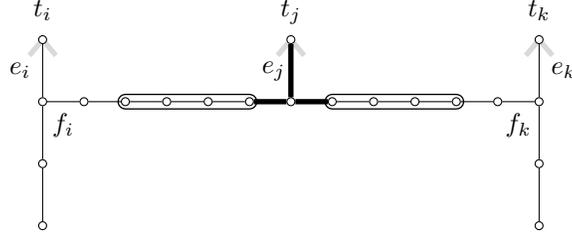


Figure 8: The impossibility of finding a $\{K_{1,3}, P_4\}$ -decomposition for a gadget when using $K_{1,3}$'s (Claim 3, reverse direction of Theorem 3.5).

(end of Claim 3) □

Claim 4. *If G' admits a $\{K_{1,3}, P_4\}$ -decomposition D , then for every gadget in G' , either e_i or e_j (or e_k in the case of H_3) belongs to an element of D which is not a subgraph of the gadget.*

Proof. The claim trivially holds for gadget H_2 , so we now turn to gadget H_3 . For this case analysis, let f_i (resp. f_k) denote the edge incident to e_i (resp. e_k) on the unique path in the gadget between t_i and t_k . The P_4 's in D that are incident to the leaves of H_3 will have to use either e_i or f_i , on one hand, and either e_k or f_k , on the other hand. This leads us to the following four cases, illustrated in Figure 9, with arrows indicating implications, i.e. the fact that selecting a certain P_4 forces us to select another P_4 .

- (a) Selecting e_i and e_k leads to building a P_4 -decomposition which is the only possible choice and leaves only e_j uncovered.
- (b) Selecting f_i and f_k leads to building a partial P_4 -decomposition which cannot be completed: the only way to cover the uncovered edges of the gadget is to use a P_4 that starts at the endpoint of either black P_4 , but that P_4 will leave at least one edge of the gadget uncovered, regardless of whether it ends with e_j .
- (c) Selecting e_i and f_k leads to building a P_4 -decomposition which is the only possible choice and leaves only e_k uncovered.
- (d) Selecting f_i and e_k leads to building a P_4 -decomposition which is the only possible choice and leaves only e_i uncovered.

(end of Claim 4) □

We can now show how to transform any $\{K_{1,3}, P_4\}$ -decomposition D for G' into an exact cover by 3-sets for G : every gadget has exactly one uncovered edge (Claim 4), and that edge is incident to the triplet t that will be part of the exact cover by 3-sets. Since D is a $\{K_{1,3}, P_4\}$ -decomposition, the other two edges incident to t are also uncovered in their respective gadgets; therefore, the triplets that belong to the exact cover by 3-sets are exactly those triplets whose closed neighbourhood induces a $K_{1,3}$ that belongs to D , and Claim 3 guarantees that only they can correspond to a $K_{1,3}$ in D . By Claim 4, the original vertex w in W is covered by exactly one triplet, so the resulting solution is indeed an exact cover by 3-sets.

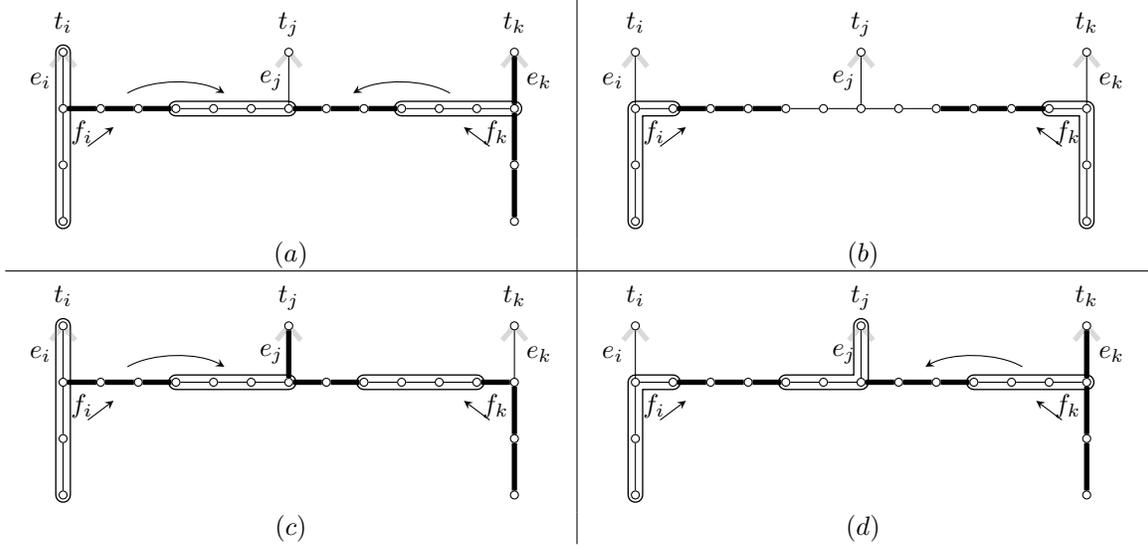


Figure 9: The four cases in the proof of Claim 4 (reverse direction of Theorem 3.5).

We only have two properties left to prove:

1. **G' is bipartite:** since the gadgets we introduced are trees, they cannot create any new cycle in G' , but they may lengthen cycles that were already present in G . The parts of cycles in G that go through a vertex of degree 2 in W are not affected by the replacement with gadget H_2 ; those that go through a vertex of degree 3 will necessarily contain two edges from $\{\{t_i, w\}, \{t_j, w\}, \{t_k, w\}\}$, and depending on the case, gadget H_3 will add either 6 or 12 edges to that cycle. Therefore, since all cycles in G are even, their (possibly lengthened) analogues in G' are also even, so G' is indeed bipartite.
2. **G' is planar:** let us identify G with any of its planar embeddings. Again, since each gadget is a tree, each gadget itself admits a planar embedding; and since no two edges connecting two different elements of W to either the same triplet or different triplets in G cross, the replacement of these elements with our gadgets will not create any crossing in G' either.

□

The only remaining case when two subgraphs are allowed is the case of $\{K_{1,3}, K_3\}$ -decompositions. Algorithm 2 finds a $\{K_{1,3}, K_3\}$ -decomposition for a strictly subcubic graph G in time $O(|V|)$ if one exists. We prove its correctness below.

Proposition 3.6. *The $\{K_{1,3}, K_3\}$ -DECOMPOSITION problem for strictly subcubic graphs is in P.*

Proof. Since G is strictly subcubic, it contains a vertex v of degree 1 or 2. We have three cases to consider for G to admit a $\{K_{1,3}, K_3\}$ -decomposition D :

1. if v is a leaf, then it must be adjacent to a vertex of degree 3 which will be the center of a $K_{1,3}$ in D ;
2. if v has degree 2, let $e_1 = \{u_1, v\}$ and $e_2 = \{u_2, v\}$:
 - (a) if $\{u_1, u_2\} \in E(G)$, the K_3 induced by $\{u_1, u_2, v\}$ must belong to D , since any other choice will leave either e_1 or e_2 uncovered and uncoverable;
 - (b) if $\{u_1, u_2\} \notin E(G)$, then v must be the intersection of two $K_{1,3}$'s in D , otherwise no $\{K_{1,3}, K_3\}$ -decomposition exists.

Algorithm 2: STRICTLYSUBCUBICCLAWTRIANGLEDECOMPOSITION(G)

Input: A graph G whose connected components are all strictly subcubic.

Output: A $\{K_{1,3}, K_3\}$ -decomposition of G if one exists, \emptyset otherwise.

```
1  $D \leftarrow \emptyset$ ;  
2  $\mathcal{S} \leftarrow$  stack with all vertices of degree  $< 3$  in  $G$ ;  
3 while  $\mathcal{S} \neq \emptyset$  do  
4    $x \leftarrow \mathcal{S}.\text{pop}()$ ;  
5   if  $\text{deg}(x) = 1$  then // case 1 in Proposition 3.6  
6      $v \leftarrow$  unique neighbour of  $x$ ;  
7     if  $\text{deg}(v) \neq 3$  then  
8       return  $\emptyset$ ;  
9      $D \leftarrow D \cup E(v)$ ;  
10     $\mathcal{S}.\text{push}(\text{vertices in } N(v) \setminus \{x\})$ ;  
11     $G \leftarrow G \setminus \{v\}$ ;  
12  else if  $\text{deg}(x) = 2$  then // case 2 in Proposition 3.6  
13     $u_1, u_2 \leftarrow N(x)$ ;  
14    if  $\{u_1, u_2\} \in E(G)$  then // case 2.(a) in Proposition 3.6  
15       $D \leftarrow D \cup G[N[x]]$ ;  
16       $\mathcal{S}.\text{push}(\text{vertices in } N(x))$ ;  
17       $G \leftarrow G \setminus G[N[x]]$ ;  
18    else // case 2.(b) in Proposition 3.6  
19       $u \leftarrow$  arbitrary neighbour of  $x$  with  $\text{deg}(u) = 3$ ;  
20      if no such  $u$  exists then  
21        return  $\emptyset$ ;  
22       $D \leftarrow D \cup E(u)$ ;  
23       $\mathcal{S}.\text{push}(\text{vertices in } N(u) \setminus \{x\})$ ;  
24       $G \leftarrow G \setminus \{u\}$ ;  
25 return  $D$ ;
```

Either one of the above three cases occur and we can produce a graph G' on which to recurse by removing a $K_{1,3}$ or a K_3 from G according to the above rules, or no decomposition for G exists. The graph G' , if non-empty, resulting from this removal is necessarily strictly subcubic: at least one of the vertices concerned by the removal ensures the connectedness of G , thus it has at least one and at most two incident edges in G' . We conclude that G admits a $\{K_{1,3}, K_3\}$ -decomposition if and only if G' does. \square

The approach implemented in Algorithm 2 uses a stack in the same way as Algorithm 1, and runs in $O(|V|)$ time. The caveats regarding disconnected graphs that we expressed in the case of Algorithm 1 apply to Algorithm 2 as well, but can be resolved in exactly the same way.

3.3 Decompositions with three types of graphs

We conclude our analysis of S' -DECOMPOSITION on strictly subcubic graphs with the case where all connected subgraphs on three edges are allowed. The hardness of this variant is a direct consequence of our previous results.

Proposition 3.7. *The $\{K_{1,3}, K_3, P_4\}$ -decomposition problem is NP-complete for strictly subcubic planar bipartite graphs.*

Proof. Follows directly from the proof of Theorem 3.5, since our reduction produces a K_3 -free graph and therefore only P_4 's and $K_{1,3}$'s can be used to decompose G . \square

4 Decomposing cubic graphs

From this section on, we study our decomposition problems on *cubic* graphs. We give polynomial-time algorithms for all subproblems tackled in this section; the next section will be devoted to proving the hardness of decomposing planar K_3 -free cubic graphs using only $K_{1,3}$'s and P_4 's.

4.1 Decompositions without a $K_{1,3}$

In this section, we study decompositions of cubic graphs that use only P_4 's or K_3 's. Note that no cubic graph is $\{K_3\}$ -decomposable, since all its vertices have odd degree. According to Bouchet and Fouquet [2], Kotzig [14] proved that a cubic graph admits a $\{P_4\}$ -decomposition iff it has a perfect matching. However, the proof of the forward direction as presented in [2] is incomplete, as it requires the use of Proposition 4.1(b) below, which is missing from their paper. Therefore, we provide the following proposition for completeness, together with another result which will also be useful for the case where $S' = \{K_3, P_4\}$.

Proposition 4.1. *Let G be a cubic graph that admits a $\{K_3, P_4\}$ -decomposition D . Then, in D , (a) no K_3 is used, and (b) no three P_4 's are incident to the same vertex.*

Proof. Partition $V(G)$ into three sets V_1, V_2 and V_3 , where V_1 (resp. V_2, V_3) is the set of vertices that are incident to exactly one P_4 (resp. two, three P_4 's) in D . Note that V_1 is exactly the set of vertices involved in K_3 's in D . Let $n_i = |V_i|$, $1 \leq i \leq 3$. Our goal is to show that $n_1 = n_3 = 0$, i.e. $V_1 = V_3 = \emptyset$. For this, note that (1) each vertex in V_3 is the extremity of three different P_4 's, (2) each vertex in V_2 is simultaneously the extremity of one P_4 and an inner vertex of another P_4 , while (3) each vertex in V_1 is the extremity of one P_4 . Since each P_4 has two extremities and two inner vertices, if p is the number of P_4 's in D , we have:

- $p = \frac{3n_3 + n_2 + n_1}{2}$ (by (1), (2) and (3) above, counting extremities);
- $p = \frac{n_2}{2}$ (by (2) above, counting inner vertices).

Putting together the above two equalities yields $n_1 = n_3 = 0$, which completes the proof. \square

The following result immediately follows from the work of Kotzig [14] and Proposition 4.1 above, since K_3 's are not allowed in $\{K_3, P_4\}$ -decompositions. As a result, $\{K_3, P_4\}$ -DECOMPOSITION for cubic graphs is in P.

Proposition 4.2. *A cubic graph admits a $\{K_3, P_4\}$ -decomposition iff it has a perfect matching.*

Micali and Vazirani [17] gave a $O(\sqrt{|V||E|})$ algorithm for finding a maximum cardinality matching in a graph; therefore, we can find a $\{K_3, P_4\}$ -decomposition for a cubic graph in time $O(|V|^{3/2})$ by computing a perfect matching and then — if one exists — converting it into a $\{P_4\}$ -decomposition as described by Kotzig [14].

4.2 Decompositions without a P_4

In this section, we study decompositions of cubic graphs that use only $K_{1,3}$'s or K_3 's.

Proposition 4.3. *A cubic graph G admits a $\{K_{1,3}\}$ -decomposition iff it is bipartite.*

Proof. For the reverse direction, select either set of the bipartition, and make each vertex in that set the center of a $K_{1,3}$. For the forward direction, let D be a $\{K_{1,3}\}$ -decomposition of G , and let C and L be the sets of vertices containing, respectively, all the centers and all the leaves of $K_{1,3}$'s in D . We show that this is a bipartition of $V(G)$. First, $C \cup L = V$ since D covers all edges and therefore all vertices. Second, $C \cap L = \emptyset$ since a vertex in $C \cap L$ would have degree at least 4. Finally, each edge in D connects the center of a $K_{1,3}$ and a leaf of another $K_{1,3}$ in D , which belong respectively to C and L . Therefore, G is bipartite. \square

As a result, $\{K_{1,3}\}$ -decompositions for cubic graphs are computable in $O(|V|)$ time. We now prove that $\{K_{1,3}, K_3\}$ -decompositions can be computed in polynomial time. Since bipartite graphs admit a $\{K_{1,3}\}$ -decomposition (by Proposition 4.3), we can restrict our attention to non-bipartite graphs that contain K_3 's (indeed, if they were K_3 -free, then only $K_{1,3}$'s would be allowed and Proposition 4.3 would imply that they admit no decomposition). Our strategy consists in iteratively removing subgraphs from G and adding them to an initially empty $\{K_{1,3}, K_3\}$ -decomposition until G is empty, in which case we have an actual decomposition, or no further removal operations are possible, in which case no decomposition exists. Our analysis relies on the following notion: a K_3 induced by vertices $\{u, v, w\}$ in a graph G is *isolated* if $V(G)$ contains no vertex x such that $\{u, v, x\}$, $\{u, x, w\}$ or $\{x, v, w\}$ induces a K_3 .

Lemma 4.1. *If a cubic graph G admits a $\{K_{1,3}, K_3\}$ -decomposition D , then every isolated K_3 in G belongs to D .*

Proof (contradiction). If an isolated K_3 were not part of the decomposition, then exactly one vertex of that K_3 would be the center of a $K_{1,3}$, leaving the remaining edge uncovered and uncoverable. \square

$\overline{C_6}$ is a minimal example of a cubic non-bipartite graph with K_3 's that admits no $\{K_{1,3}, K_3\}$ -decomposition: both K_3 's in that graph must belong to the decomposition (by Lemma 4.1), but their removal yields a perfect matching. The following simple observation will allow us to rely on the algorithms we developed in Section 3.

Observation 4.1. *Let G be a connected cubic graph. Then no sequence of at least one edge or vertex removal from G yields a cubic graph.*

Proof (contradiction). If after applying at least one removal operation on G we obtain a cubic graph G' , then the graph that precedes G' in this removal sequence must have had a vertex of degree at least 4, since G is connected. \square

Proposition 4.4. *For any connected non-bipartite cubic graph G whose K_3 's are all isolated, one can decide in polynomial time whether G is $\{K_{1,3}, K_3\}$ -decomposable.*

Proof. As seen in Lemma 4.1, all isolated K_3 's must belong to a $\{K_{1,3}, K_3\}$ -decomposition if one exists, so we can remove them all from G and add them to an initially empty set D ; therefore, G admits a $\{K_{1,3}, K_3\}$ -decomposition iff the resulting graph G' admits a $\{K_{1,3}\}$ -decomposition. Observation 4.1 implies that all connected components of G' are strictly subcubic, and Proposition 3.2 then allows us to compute a decomposition for G' if one exists, using Algorithm 2. \square

We conclude with the case where the graph may contain non-isolated K_3 's. They will necessarily belong to a *diamond*, i.e. a graph isomorphic to a K_4 from which an edge has been removed.

Proposition 4.5. *If a cubic graph G contains a diamond, then one can decide in polynomial time whether G is $\{K_{1,3}, K_3\}$ -decomposable.*

Proof. The only cubic graph on 4 vertices is K_4 , which is diamond-free and $\{K_{1,3}, K_3\}$ -decomposable, so we assume $|V(G)| \geq 6$ below. Let $H = G[\{u, v, w, x\}]$ be a diamond, with $\{u, x\} \notin E(G)$, as Figure 10(a) shows. H is connected to two other vertices u' and x' of G , which are respectively adjacent to u and x , and there are only two ways to use the edges of H in a $\{K_{1,3}, K_3\}$ -decomposition D , as shown in Figure 10(b) and (c). If $u' = x'$ (as shown in Figure 10(d)), let y be the neighbour of x' which is neither u nor x . If the decomposition we choose for H is the one depicted in Figure 10(b), then edges (x', u) and (x', y) are uncovered, while (x', x) is covered. In that case, no $\{K_{1,3}, K_3\}$ -decomposition can exist. If the decomposition we choose for H is the one depicted in Figure 10(c), a similar argument applies and leads us to conclude that no $\{K_{1,3}, K_3\}$ -decomposition can exist either. Altogether, this shows that no $\{K_{1,3}, K_3\}$ -decomposition exists for G in case (d), and therefore, we assume from now on that $u' \neq x'$.

As Figure 10(b) and (c) show, either $\{u, v, w\}$ or $\{v, w, x\}$ must form a K_3 in D , thereby forcing either $\{v, w, x, x'\}$ or $\{u', u, v, w\}$ to form a $K_{1,3}$ in D . In both cases, removing the K_3 and the $K_{1,3}$ yields a strictly subcubic graph, so we can apply the results obtained in Proposition 3.6 as follows: we apply the decomposition in Figure 10(b) then run Algorithm 2 on G' , and return the union of both decompositions. If this fails, we try again with Figure 10(c) as a starting point, and return the union of both decompositions, or \emptyset if both tries failed. \square

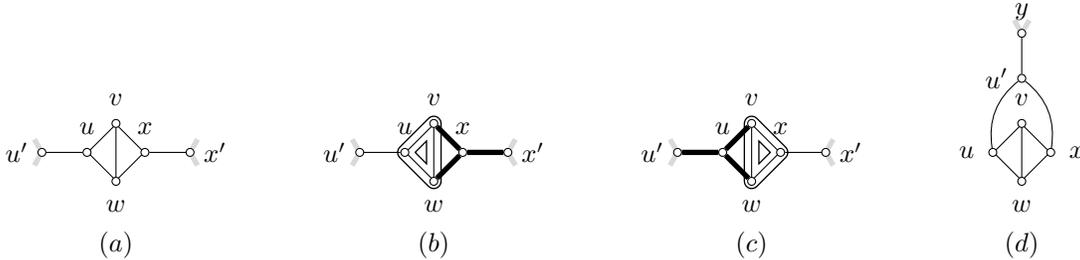


Figure 10: (a) A diamond in a cubic graph, and (b), (c) the only two ways to decompose it in a $\{K_{1,3}, K_3\}$ -decomposition. (d) The case where $u' = x'$.

All the arguments developed in this section lead to the following result.

Proposition 4.6. *The $\{K_{1,3}, K_3\}$ -DECOMPOSITION problem for a cubic graph G can be solved in time $O(|V(G)|)$.*

Proof. Algorithm 3 implements the approach proposed in this section, and its correctness follows from proofs of Propositions 4.3 to 4.5. We now examine its time complexity: lines 1-4 run in $O(|V(G)|)$ time. A diamond in a cubic graph can be found by naively searching for a claw $\{u, v, w, x\}$ with center u and then checking whether $G[\{v, w, x\}] \simeq P_3$, which takes $O(|V(G)|)$ time as well. Lines 8–17 either involve operations running in $O(1)$ or $O(|V(G)|)$ time, or calls to Algorithm 2 which, as seen in Section 3, runs in $O(|V(G)|)$ time. Finally, line 18 involves listing all isolated triangles in a cubic graph, which can be done naively by checking the intersection of $N(u)$ and $N(v)$ for each edge $\{u, v\}$. This approach takes $O(|E(G)|) = O(|V(G)|)$ time since G is cubic. \square

Without loss of generality, we require that the input to Algorithm 3 be connected. This is because Algorithm 3 uses Algorithms 1 and 2: they both require that all connected components of their inputs be strictly subcubic, and we can only guarantee this if the input to Algorithm 3 is connected (see Observation 4.1). Of course, if the input to Algorithm 3 is not connected, we can simply run it on each connected component of the input.

Algorithm 3: CUBICCLAWTRIANGLEDECOMPOSITION(G)

Input: A connected cubic graph G .
Output: A $\{K_{1,3}, K_3\}$ -decomposition of G if one exists, \emptyset otherwise.

```

1 if  $V(G)$  admits a bipartition  $V_1 \cup V_2$  then // see Proposition 4.3
2 |   return  $\{G[N[v]] \mid v \in V_1\}$ ;
3 if  $G$  is  $K_3$ -free then // see Proposition 4.3
4 |   return  $\emptyset$ ;
5 if  $G \simeq K_4$  then
6 |   return  $\{G[\{u, v, w\}], G \setminus G[\{u, v, w\}]\}$ ;
7 if  $G$  contains a diamond then // see Proposition 4.5
8 |    $u, v, w, x \leftarrow$  vertices as in Figure 10;
9 |   /* try the decomposition in Figure 10(b) as starting point */
10 |    $A \leftarrow \{G[\{u, v, w\}]\}$ ;
11 |    $D \leftarrow$  STRICTLYSUBCUBICCLAWTRIANGLEDECOMPOSITION( $G \setminus A$ );
12 |   if  $D \neq \emptyset$  then
13 |     |   return  $D \cup A$ ;
14 |     /* try the decomposition in Figure 10(c) as starting point */
15 |      $B \leftarrow \{G[\{v, x, w\}]\}$ ;
16 |      $D \leftarrow$  STRICTLYSUBCUBICCLAWTRIANGLEDECOMPOSITION( $G \setminus B$ );
17 |     if  $D \neq \emptyset$  then
18 |       |   return  $D \cup B$ ;
19 |     return  $\emptyset$ ;
20 /* last case:  $G$  is diamond-free (see Proposition 4.4) */
21  $T \leftarrow$  all (necessarily isolated) triangles in  $G$ ;
22  $D \leftarrow$  STRICTLYSUBCUBICCLAWDECOMPOSITION( $G \setminus T$ );
23 if  $D \neq \emptyset$  then
24 |   return  $D \cup T$ ;
25 return  $\emptyset$ ;
```

5 Decomposing cubic graphs using both $K_{1,3}$'s and P_4 's

In this section, we show that problems $\{K_{1,3}, P_4\}$ -DECOMPOSITION and $\{K_{1,3}, K_3, P_4\}$ -DECOMPOSITION are NP-complete, even on planar graphs. Our first hardness proof relies on intermediate problems that we define below and is structured as follows:

| | |
|---|-------------|
| CUBIC PLANAR MONOTONE 1-IN-3 SATISFIABILITY | |
| \leq_P EVEN-FACED EDGE SELECTION | (Lemma 5.4) |
| \equiv $\{K_{1,3}, P_4\}$ -DECOMPOSITION OF MARKED GRAPHS | (Lemma 5.3) |
| \leq_P $\{K_{1,3}, P_4\}$ -DECOMPOSITION in K_3 -free planar graphs | (Lemma 5.2) |

5.1 Marking edges

We first define a *marked graph* as a pair (G, M) where $G = (V, E)$ is a planar, embedded, triangle-free graph and $M \subseteq E$ is a subset of *marked* edges. The drawings that illustrate our proofs in this section show marked edges as thick and solid (i.e. $\circ\text{---}\circ$), and unmarked edges as thin and dashed (i.e. $\circ\text{---}\text{---}\circ$). We define the following intermediate decomposition problem on marked graphs:

$\{K_{1,3}, P_4\}$ -DECOMPOSITION OF MARKED GRAPHS

Input: a cubic marked graph (G, M) .

Question: does G admit a $\{K_{1,3}, P_4\}$ -decomposition D in which no edge in M is the middle edge of a P_4 ?

The proof of Lemma 5.2, which shows how the above problem relates to $\{K_{1,3}, P_4\}$ -DECOMPOSITION, uses the following result.

Lemma 5.1. *Let e be a bridge in a cubic graph G which admits a $\{K_{1,3}, K_3, P_4\}$ -decomposition. Then e must be the middle edge of a P_4 in D .*

Proof (contradiction). First note that $e = \{u, v\}$ cannot belong to a K_3 in D . Now suppose e is part of a $K_{1,3}$ in D . Without loss of generality, the situation is as shown in Figure 11 (G_u (resp. G_v) denotes the connected component that contains u (resp. v) in $G \setminus e$).

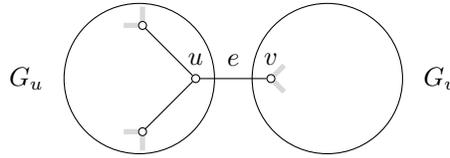


Figure 11: A bridge $e = \{u, v\}$ in a cubic graph G , and the connected component G_u (resp. G_v) of $G \setminus e$ that contains u (resp. v).

If we remove from G the $K_{1,3}$ in D that contains e , then summing the terms of the degree sequence of G_v yields $2 + 3(|V(G_v)| - 1) = 2|E(G_v)|$, which means that $|E(G_v)| \not\equiv 0 \pmod{3}$ and therefore G_v admits no decomposition into components of size 3. The very same argument shows that if e belongs to a P_4 in D , then it must be its middle edge, which completes the proof. \square

Lemma 5.2. *Let (G, M) be an instance of $\{K_{1,3}, P_4\}$ -DECOMPOSITION OF MARKED GRAPHS, where G is K_3 -free and planar, and G' be the graph obtained by attaching a buoy to every edge in M . Then (G, M) is a yes-instance iff G' admits a $\{K_{1,3}, P_4\}$ -decomposition.*

Proof. First, note that the transformation from G to G' , described in the statement of the lemma, preserves cubicity, K_3 -freeness, and planarity. Now, we prove each direction separately.

\Rightarrow : we show how to transform a decomposition D of (G, M) into a decomposition D' of G' . The subgraphs in D that have no edge in M are not modified. For the other subgraphs, a subgraph containing an edge of M is either (a) a $K_{1,3}$ or (b) a P_4 whose middle edge is not in M . Figure 12 shows how to adapt the decompositions in both cases.

\Leftarrow : we now show how to transform any $\{K_{1,3}, P_4\}$ -decomposition D' of G' into a decomposition of (G, M) . Again, the only parts of D' that will need adapting are those connected to the buoys that we attached to each marked edge when transforming G into G' . Since the leaf u of the buoy we attached has a neighbour x such that $\{u, x\}$ is a bridge in G' , $\{u, x\}$ is the middle edge of a P_4 in D' (Lemma 5.1) and we may therefore assume without loss of generality that our starting point in G' is as shown in Figure 13, where $\{v, w\} \notin E(G')$ since G is simple; we have two cases to consider:

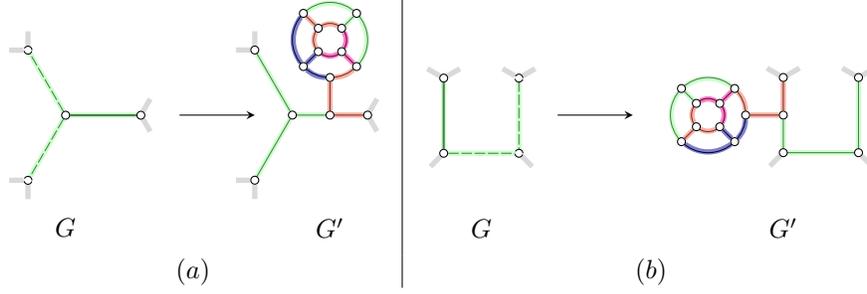


Figure 12: Adapting a $\{K_{1,3}, P_4\}$ -decomposition when attaching a buoy to a marked edge of a graph G .

- (a) if $\{u, w\}$ belongs to a $K_{1,3}$ in D' , that $K_{1,3}$ can be mapped onto a $K_{1,3}$ in D by replacing $\{u, w\}$ with $\{v, w\}$;
- (b) otherwise, $\{u, w\}$ is an extremal edge of a P_4 in D' , and that P_4 can be mapped onto a P_4 in D by replacing $\{u, w\}$ with $\{v, w\}$. Note that the resulting subgraph may only be a P_4 , and not a K_3 , since G is K_3 -free; moreover, only the extremal edge of a P_4 may be marked in (G, M) by definition.

□

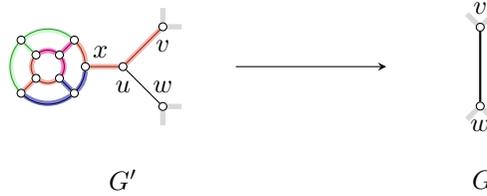


Figure 13: Obtaining a $\{K_{1,3}, P_4\}$ -decomposition for a marked graph G from a $\{K_{1,3}, P_4\}$ -decomposition for its transformed counterpart G' .

5.2 Equivalence with EVEN-FACED EDGE SELECTION

Recall that a planar embedding of a planar graph partitions \mathbb{R}^2 into connected components which are referred to as the *faces* of the embedding. *Inner faces* are bounded by edges and vertices of the graph, while the unbounded face is called the *outer face*. We represent a face using the set of vertices on its border. A face is said to be *incident* with an edge e if it contains at least one of its endpoints.

Definition 5.1. A *selection* in a marked graph (G, M) with $G = (V, E)$ is a subset $E^* \subseteq E \setminus M$ such that no two edges are adjacent. The *weight* of a face is the number of vertices of degree 3 it contains. A selection E^* is *even-faced* if every inner face of $(V, E \setminus E^*)$ has an even weight.

Note that equivalently, E^* is even-faced iff all cycles of $(V, E \setminus E^*)$ have an even number of vertices of degree 3. We define the following problem, which will prove helpful in the sequel.

EVEN-FACED EDGE SELECTION
 Input: a cubic marked graph (G, M) .
 Question: does G admit an even-faced selection?

Figure 14 illustrates this problem and its connection to $\{K_{1,3}, P_4\}$ -DECOMPOSITION.

Lemma 5.3. *The problems EVEN-FACED EDGE SELECTION and $\{K_{1,3}, P_4\}$ -DECOMPOSITION OF MARKED GRAPHS are equivalent.*

Proof. It suffices to show that a cubic marked graph (G, M) has an even-faced selection if, and only if, G admits a $\{K_{1,3}, P_4\}$ -decomposition such that no edge in M is the middle edge of a P_4 . We prove each direction separately.

\Leftarrow : assume first that G admits a $\{K_{1,3}, P_4\}$ -decomposition D . Let E^* be the set of middle edges of all P_4 's in D . E^* is clearly a subset of $E \setminus M$, and no two edges may be incident (such a configuration would require a vertex of degree 4), therefore it is a selection of (G, M) . To show that this selection is also even-faced, we orient every edge e of $E \setminus E^*$ according to D :

- if e belongs to a $K_{1,3}$, orient it from the center to the leaf of the $K_{1,3}$;
- if e belongs to a P_4 , then it must be one of the extreme edges of this path since it cannot be its middle edge: orient it towards the extremity of the P_4 .

Note that with this construction, for any node v and any incident edge e , where e belongs to some subgraph X in D (where X may be a P_4 or a $K_{1,3}$), the degree of v is 1 in X if and only if e is going towards v .

We now show that there are three types of vertices in $G[E \setminus E^*]$:

- (a) vertices with 3 outgoing edges,
- (b) vertices with 3 incoming edges, and
- (c) vertices with 1 incoming and 1 outgoing edges.

If a vertex is shared between 3 distinct parts of D , it may only have degree 1 in each, therefore it belongs to class (b). If it is shared by 2 distinct parts of D , then it has degree 1 in either one and degree 2 in the other, therefore it belongs to class (c). Finally, if a vertex is used by a single part of D , it has degree 3 in that part, and therefore belongs to class (a).

Consider now any cycle C of $G[E \setminus E^*]$. Summing over all vertices of C , the total number of outgoing edges in C must equal the total number of incoming edges. Vertices of type (a) contribute 2 to the first sum, vertices of type (b) contribute 2 to the second sum, and vertices of type (c) contribute one to each sum. So every cycle contains as many vertices of type (a) as vertices of type (b). Since types (a) and (b) together represent all vertices of degree 3, every cycle, and in particular every face, contains an even number of vertices of degree 3.

\Rightarrow : Let E^* be an even-faced selection. Build the (multi-)graph G' using $G[E \setminus E^*]$ by contracting vertices of degree 2 (so that induced paths are replaced with single edges). Then all remaining vertices have degree 3, and by the even faced property, all faces have an even degree. Therefore, G' is bipartite: partition its vertices into two independent sets A and B , and orient its edges from A to B . Now, orient all edges of every induced path of $G[E \setminus E^*]$ according to the orientation of the corresponding edge of G' . As in the first part of the proof, there are 3 types of vertices in $G[E \setminus E^*]$:

- (a) vertices with 3 outgoing edges (which belong to A),
- (b) vertices with 3 incoming edges (which belong to B), and
- (c) vertices with 1 incoming and 1 outgoing edges (which do not appear in G' since they are contracted).

Build a $\{K_{1,3}, P_4\}$ -decomposition as follows: for every vertex v of type (a), add a $K_{1,3}$ centered in v . For every edge $e = (u, v) \in E^*$, u and v have type (c), so they both are the starting point of outgoing arcs e_u, e_v respectively: $\{e_u, e, e_v\}$ therefore forms a P_4 , and we add it to the decomposition. For every vertex of type (a), (b) or (c), every outgoing edge in $E \setminus E^*$ belongs to a single part added to the decomposition, as well as every edge in E^* . Therefore, this decomposition partitions E .

□

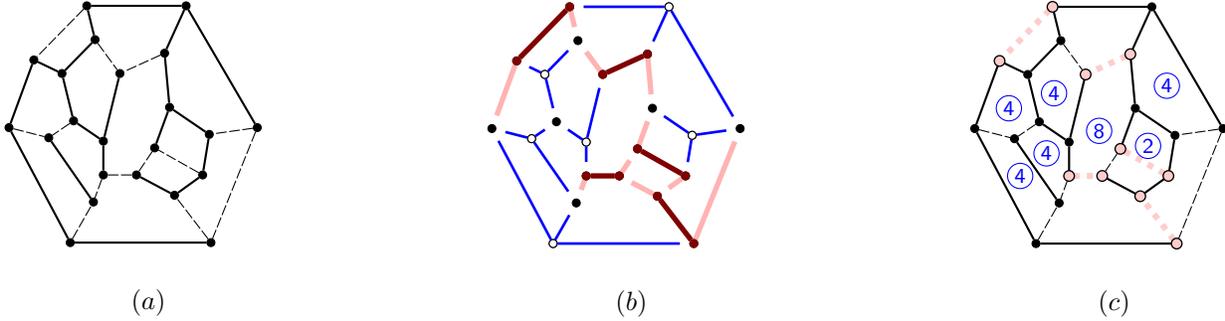


Figure 14: Illustration of $\{K_{1,3}, P_4\}$ -DECOMPOSITION and EVEN-FACED EDGE SELECTION. (a) A marked cubic graph. Recall that marked edges are indicated as solid thick lines, and unmarked ones as dashed lines. (b) A $\{K_{1,3}, P_4\}$ -decomposition of this graph, where $K_{1,3}$'s are represented by thin blue edges with a white center, and P_4 's by thick red edges. In particular, the middle edge of a P_4 (dark bold) is never marked. (c) An even-faced selection for this graph, where selected edges of E^* are indicated as dashed bold lines, and vertices of degree 3 in $(V, E \setminus E^*)$ are black. Circled numbers correspond to the weights of inner faces of $(V, E \setminus E^*)$. As described in Lemma 5.3, this selection corresponds to the $\{K_{1,3}, P_4\}$ -decomposition: selected edges are middle edges of P_4 's, and, among the remaining vertices, vertices that are at the center of a $K_{1,3}$ alternate with other vertices, which corresponds to the parity condition.

5.3 The hardness of EVEN-FACED EDGE SELECTION

We prove that EVEN-FACED EDGE SELECTION is NP-complete through an assembly of gadgets. We give a few formal definitions below to help understand how gadgets interact, after a short outline of the hardness proof to motivate their introduction. The reduction relies on the CUBIC PLANAR MONOTONE 1-IN-3 SATISFIABILITY problem, which is known to be NP-complete [18]:

| CUBIC PLANAR MONOTONE 1-IN-3 SATISFIABILITY | |
|---|---|
| Input: | a Boolean formula $\phi = C_1 \wedge C_2 \wedge \dots \wedge C_n$ without negations over a set $\Sigma = \{x_1, x_2, \dots, x_m\}$, with exactly three distinct variables per clause and where each literal appears in exactly three clauses; moreover, the graph with clauses and variables as vertices, and edges joining clauses and the variables they contain, is planar. |
| Question: | does there exist an assignment of truth values $f : \Sigma \rightarrow \{\text{TRUE}, \text{FALSE}\}$ such that exactly one literal is TRUE in every clause of ϕ ? |

Since we intend to prove the hardness of EVEN-FACED EDGE SELECTION using a reduction from a variant of the SATISFIABILITY problem, we will need a number of gadgets to encode boolean formulas in a graph-theoretic language. Those gadgets come in the form of marked graphs whose type will depend on their function in the reduction: three of them (OR, NAND and ANY) simulate boolean operations, and the remaining two (VAR and CLAUSE) allow us to encode variables and clauses, while preserving the particular additional constraints of CUBIC PLANAR MONOTONE 1-IN-3 SATISFIABILITY. All these gadgets will need to be connected in a specific way, which will be achieved using each gadget's *interface edges*.

Definition 5.2. A *gadget* g is a marked graph (G, M) with the following properties:

- all vertices have degree 1 or 3; edges incident with vertices of degree 1 are called *interface edges*;
- all interface edges are unmarked (i.e. in $E(G) \setminus M$);
- all interface edges are on the outer face;
- no two interface edges are incident.

The *interface* of a gadget g , denoted by $I(g)$, is the set of its interface edges.

Definition 5.3. A subset $B \subseteq I(g)$ is a *border* for g if there exists an even-faced selection E^* for g such that $B = E^* \cap I(g)$.

Definition 5.4. An *outer path* of g for E^* is a path in the outer face of gadget g , after removing edges from E^* , starting and ending with interface edges. It is the whole outer face if there is no remaining interface edge in g after removing E^* . Its *weight* is the number of vertices of degree 3 it contains.

Definition 5.5. A border is *even* if $B = E^* \cap I(g)$ for some E^* such that all outer paths for E^* have an even weight.

Intuitively, we build an instance as an assembly of gadgets, where adjacent gadgets share interface edges: vertices of degree 1 are removed in the process, the graph remains planar (since interface edges are on the outer faces of the gadgets), and it is K_3 -free. An even-faced selection for the whole graph can be seen as a collection of even-faced selections, one for each gadget. Conversely, a collection of even-faced selections for all gadgets yields an even-faced selection for the whole graph, when the borders of adjacent gadgets are consistent on shared edges and each border is even (this is a sufficient condition ensuring that all faces shared between several gadgets have an even weight).

Sections 5.3.1 to 5.3.5 are devoted to the introduction of our gadgets and the identification of their borders, which are all even (as will be proved in Propositions 5.1 to 5.5). We then combine them in Section 5.3.6 to obtain the hardness proof outlined above.

5.3.1 The OR gadget

The top of Figure 15 shows the OR gadget, which is our basic building block and has interface $\{\alpha, \beta, \gamma, \delta\}$. It combines two purposes: it serves as a crossing gadget (opposite interface edges are either both selected or both unselected), and as a logical OR (at least one pair among $\{\alpha, \gamma\}$ and $\{\beta, \delta\}$ is selected).

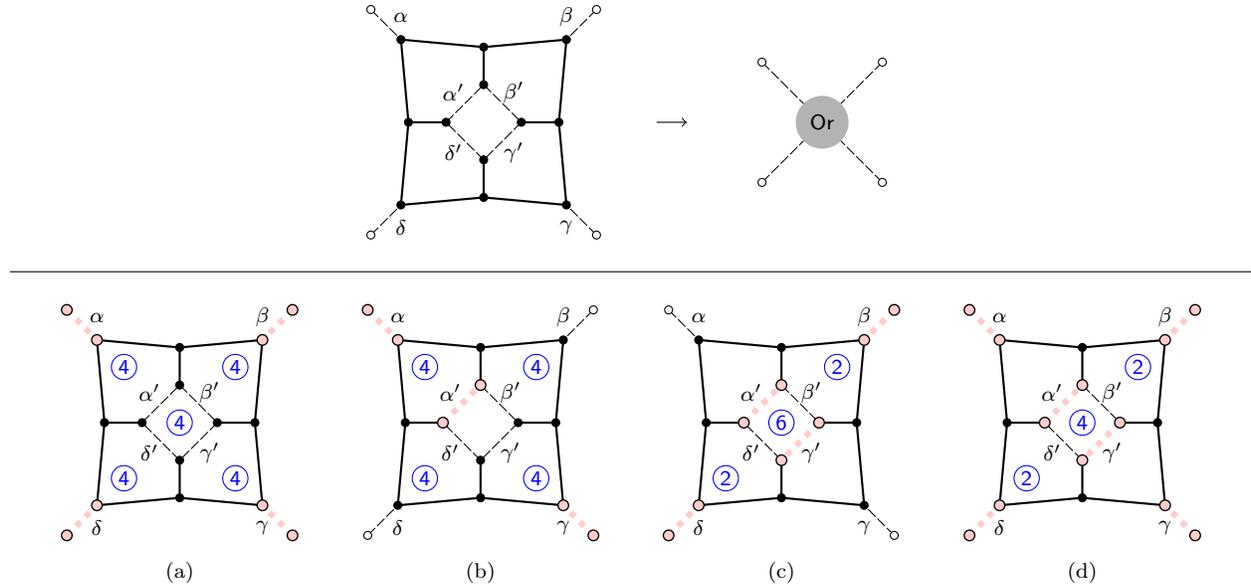


Figure 15: Top: Definition of the OR gadget. Bottom: All possible selections (up to symmetry) for this gadget. Circled numbers indicate the weights of the inner faces.

Proposition 5.1. The possible borders for an OR gadget are $\{\alpha, \gamma\}$, $\{\beta, \delta\}$, and $\{\alpha, \beta, \gamma, \delta\}$. They are all even.

Proof. The illustrations in Figure 15(a)–(d) each show an even-faced selection for the OR gadget that corresponds to each of the given sets, so they are indeed borders. Moreover, all outer paths for these selections have even weight, so these three borders are even.

We now prove that any even-faced selection E^* has one of the above three borders. Let C be the cycle of length 4 induced by $\{\alpha', \beta', \gamma', \delta'\}$ in the OR gadget. Since no two edges of E^* may be incident, we have $|E^* \cap C| \leq 2$, so we have three cases to check:

1. if $|E^* \cap C| = 0$, then consider the four weight-5 inner faces of the gadget. Each of them has a single incident unmarked edge excluding C , hence this edge must be selected (otherwise the faces would still have an odd weight in $(V, E \setminus E^*)$). This leads to $E^* = \{\alpha, \beta, \gamma, \delta\}$ (as in Figure 15(a)).
2. if $|E^* \cap C| = 1$, assume first that $E^* \cap C = \{\alpha'\}$. The graph $(V, E \setminus \{\alpha'\})$ has four inner faces: two with weight 4, incident to β and δ respectively (so $\beta, \delta \notin E^*$), and two with weight 5, incident to α and γ . Therefore, we must have $\alpha, \gamma \in E^*$ to reach weight 4 as in Figure 15(b). The same argument shows that the border must also be $\{\alpha, \gamma\}$ if $E^* \cap C = \{\gamma'\}$, or $\{\beta, \delta\}$ if $E^* \cap C = \{\beta'\}$ or $E^* \cap C = \{\delta'\}$.
3. if $|E^* \cap C| = 2$, then $E^* \cap C$ may be either $\{\alpha', \gamma'\}$ or $\{\beta', \delta'\}$. Assume first that $E^* \cap C = \{\alpha', \gamma'\}$. The graph $(V, E \setminus \{\alpha', \gamma'\})$ has three inner faces: two with weight 3, incident to β and δ respectively (so $\beta, \delta \in E^*$ in order to reach weight 2), and one with weight 6 incident to both α and γ . So either $\alpha, \gamma \in E^*$, and this face gets weight 4 (see Figure 15(c)), or $\alpha, \gamma \notin E^*$ (see Figure 15(d)). With the same argument, the border is either $\{\alpha, \beta, \gamma, \delta\}$ or $\{\alpha, \gamma\}$ if $E^* \cap C = \{\beta', \delta'\}$.

□

5.3.2 The NAND gadget

The top of Figure 16 shows the NAND gadget, which has interface $\{\alpha, \beta, \gamma, \delta\}$. Similarly to the OR gadget, it combines two purposes: it serves as a crossing gadget (opposite interface edges are either both selected or both unselected), and as a logical NAND (at most one pair among $\{\alpha, \gamma\}$ and $\{\beta, \delta\}$ is selected).

Proposition 5.2. *The possible borders for a NAND gadget are $\{\alpha, \gamma\}$, $\{\beta, \delta\}$, and \emptyset . They are all even.*

Proof. The illustrations in Figure 16(a)–(c) each show an even-faced selection for the NAND gadget that corresponds to each of the given sets, so they are indeed borders. Any of these selections can be verified to be even-faced using the fact that all outer paths of the OR gadget have an even weight.

We now prove that any even-faced selection E^* for the NAND gadget has one of the above three borders. Consider first the weight-5 face incident to α and α' . To reach an even weight, we have $|E^* \cap \{\alpha, \alpha'\}| = 1$, i.e. $\alpha \in E^* \Leftrightarrow \alpha' \notin E^*$. The same property applies to β, γ, δ .

Consider now the OR gadget included as a subgraph, with interface $\{\alpha', \beta', \gamma', \delta'\}$. Selection E^* for the whole NAND gadget yields an even-faced selection for the OR gadget, so its intersection with the interface is a border of the OR gadget. Proposition 5.1 then implies that $E^* \cap \{\alpha', \beta', \gamma', \delta'\}$ is one of $\{\alpha', \gamma'\}$, $\{\beta', \delta'\}$ and $\{\alpha', \beta', \gamma', \delta'\}$, and therefore the border is one of $\{\alpha, \gamma\}$, $\{\beta, \delta\}$, and \emptyset . □

5.3.3 The ANY gadget

The top of Figure 17 shows the ANY gadget, which has interface $\{\alpha_1, \alpha_2, \alpha_3\}$. Its purpose is to force the selection of a single interface edge α_i among all three, which corresponds to the “1-in-3” constraint of the SATISFIABILITY variant we will reduce from.

Proposition 5.3. *The possible borders for an ANY gadget are $\{\alpha_1\}$, $\{\alpha_2\}$, $\{\alpha_3\}$. They are all even.*

Proof. Each $\{\alpha_i\}$ is indeed an even border, as can be seen with the even-faced selections depicted in Figure 17(a)–(c).

We now prove that the only choices for an even-faced selection E^* for the ANY gadget must be exactly one of the $\{\alpha_i\}$. Consider first the weight-4 face incident to edges α_i and β_i for any $i \in \{1, 2, 3\}$. For this face

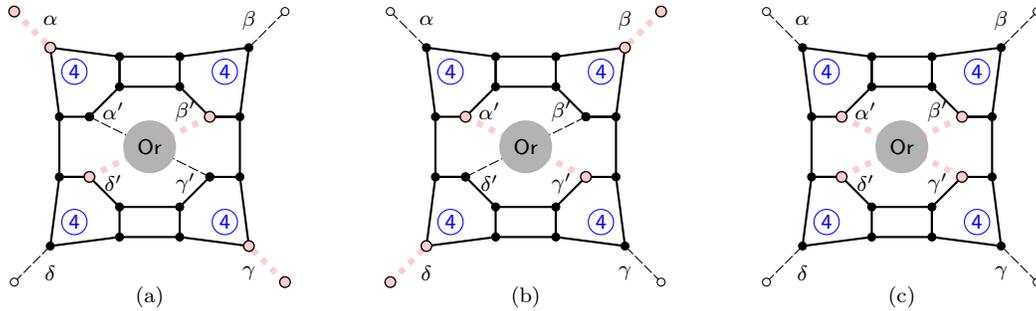
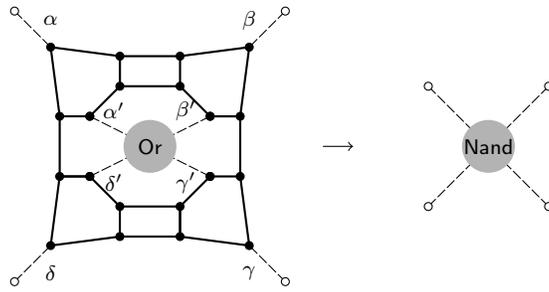


Figure 16: Top: Definition of the NAND gadget. Bottom: All possible selections for this gadget. Circled numbers indicate the weights of some inner faces.

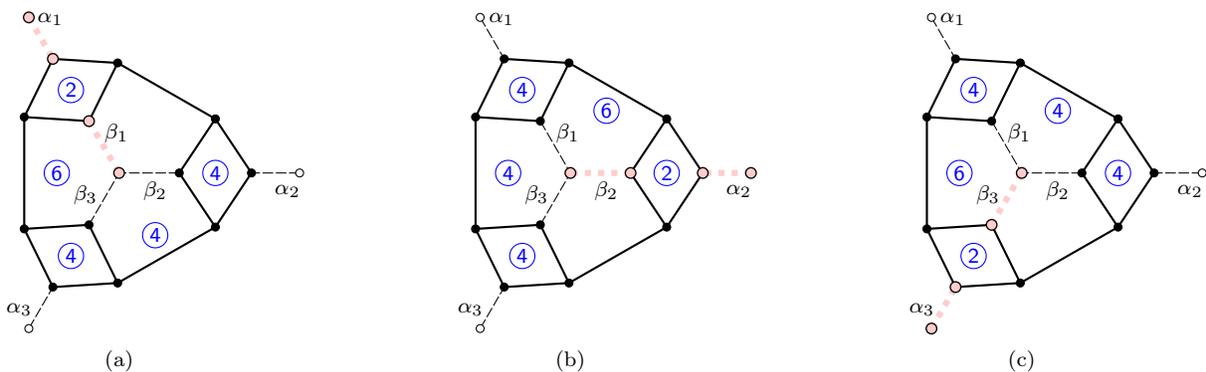
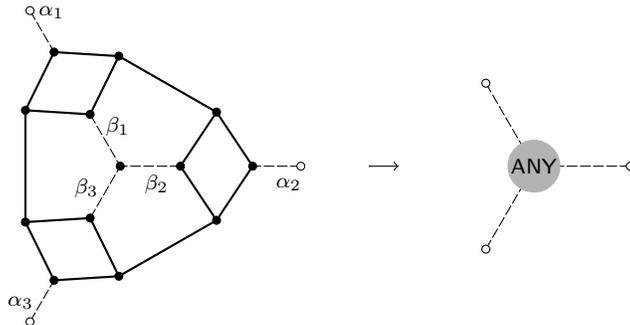


Figure 17: Top: Definition of the ANY gadget. Bottom: All possible selections for this gadget. Circled numbers indicate the weights of the inner faces.

to maintain an even weight, we have $\alpha_i \in E^* \Leftrightarrow \beta_i \in E^*$. Consider now the weight-5 faces incident to all three edges β_i . Since they have an odd weight, $E^* \cap \{\beta_1, \beta_2, \beta_3\} \neq \emptyset$. Furthermore, since those three edges are pairwise incident, $|E^* \cap \{\beta_1, \beta_2, \beta_3\}| \leq 1$. Therefore, E^* contains exactly one edge among $\{\beta_1, \beta_2, \beta_3\}$, and thus exactly one edge among $\{\alpha_1, \alpha_2, \alpha_3\}$. \square

5.3.4 The VAR gadget

The top of Figure 18 shows the VAR gadget, which has interface $\{\alpha_i \mid 1 \leq i \leq 6\}$. Its purpose is to simulate a boolean variable where either all interface edges are selected, or none at all (as will be seen later on, a literal in a SATISFIABILITY instance will be represented as a pair of edges shared between gadgets, hence a variable with three occurrences needs an interface of size 6).

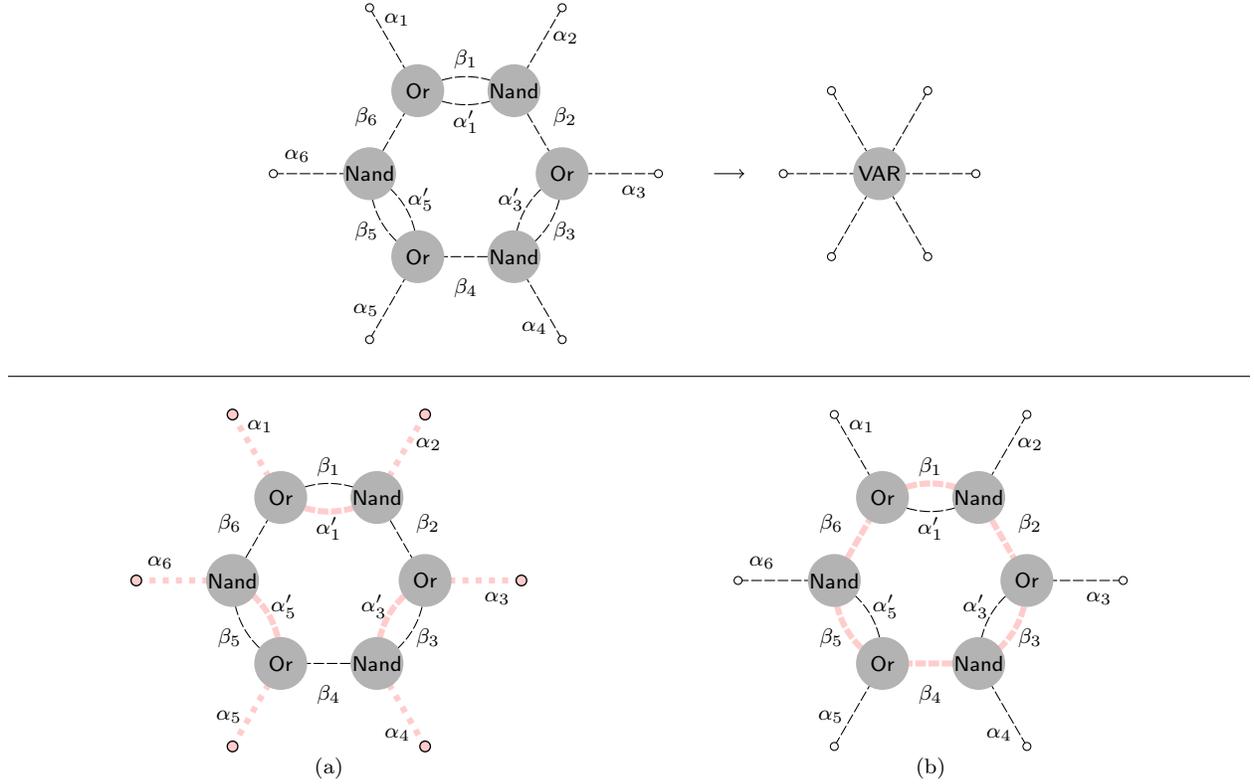


Figure 18: Top: Definition of the VAR gadget. Bottom: All possible selections for this gadget.

Proposition 5.4. *The possible borders for a VAR gadget are $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6\}$, and \emptyset . They are all even.*

Proof. Figure 18(a) and (b) show two selections creating the mentioned borders. They are indeed even-faced since for each OR and NAND gadget, they form an even border of these faces (therefore they can be extended as an even-faced selection of these gadgets, and the weight of any face not entirely included in a smaller gadget is the sum of the weights of outer paths for these gadgets, so it is even as well). Similarly, the weight of any outer path is the sum of the weights of outer paths for several gadgets, so it is also even.

We now show that any even-faced selection E^* for the VAR gadget must be either of the mentioned borders. Propositions 5.1 and 5.2 imply that $\beta_i \in E^* \Leftrightarrow \beta_j \in E^*$ for every $1 \leq i, j \leq 6$. With the same properties, $\alpha_1 \in E^* \Leftrightarrow \alpha'_1 \in E^* \Leftrightarrow \alpha_2 \in E^*$ (and similarly for $\{\alpha_3, \alpha'_3, \alpha_4\}$ and $\{\alpha_4, \alpha'_4, \alpha_5\}$). Furthermore, Proposition 5.1 implies that at least one of α'_1 and β_1 must be in E^* , and Proposition 5.2 implies that at

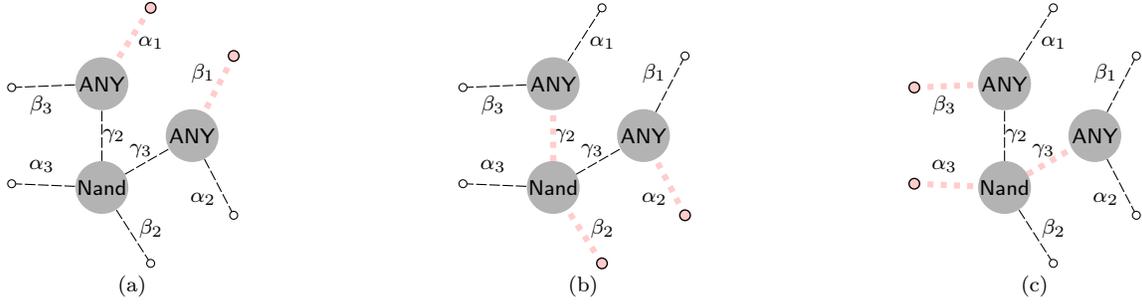
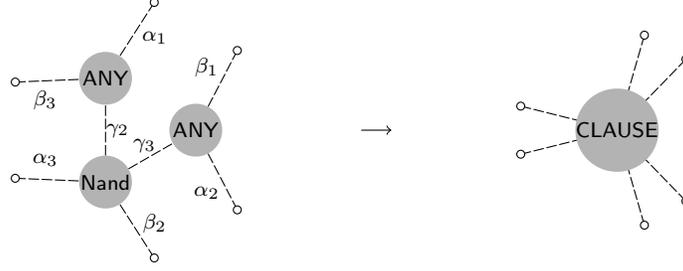


Figure 19: Top: Definition of the CLAUSE gadget. Bottom: All selections yielding a paired border (α_1 with β_1 , α_2 with β_2 , α_3 with β_3).

most one of them is in E^* . Overall, $\alpha_i \in E^* \Leftrightarrow \beta_i \notin E^*$. The same applies for (α_3, β_3) and (α_5, β_5) , and since all β_i 's are equivalent, either $\alpha_i \in E^*$ for all $1 \leq i \leq 6$, or $\alpha_i \notin E^*$ for all $1 \leq i \leq 6$. \square

5.3.5 The CLAUSE gadget

The top of Figure 19 shows the CLAUSE gadget, which has interface $\{\alpha_i, \beta_i \mid 1 \leq i \leq 3\}$. Its purpose is to simulate the behaviour of a “1-in-3” clause, where exactly one pair (α_i, β_i) of its interface may be selected (it can be seen as an extension of the ANY interval, where two edges are used for each literal). However, it does not ensure that the pairs (α_i, β_i) are selected together (for instance, $\{\alpha_1, \beta_3\}$ is an even-faced border of this gadget): this constraint will be enforced from outside, by VAR gadgets. This motivates the following definition: a border B for the CLAUSE gadget is *paired* if, for each $1 \leq i \leq 3$, $\alpha_i \in B \Leftrightarrow \beta_i \in B$.

Proposition 5.5. *The paired borders for a CLAUSE gadget are $\{\alpha_1, \beta_1\}$, $\{\alpha_2, \beta_2\}$ and $\{\alpha_3, \beta_3\}$. They are all even.*

Proof. These sets are indeed even and constitute paired borders for the CLAUSE gadget, since they can be obtained from the even-faced selections depicted in Figure 19(a)–(c) (just like for the VAR gadget, these selections are even-faced because they yield even borders of the ANY and NAND subgraphs).

Let E^* be any even-faced selection of the CLAUSE gadget with a paired border. Proposition 5.3 implies that E^* contains exactly one edge among $\alpha_1, \gamma_2, \beta_3$. Proposition 5.2 implies that $\gamma_2 \in E^* \Leftrightarrow \beta_2 \in E^*$, so E^* contains exactly one edge among $\alpha_1, \beta_2, \beta_3$. Since the border of E^* is paired, it contains exactly one pair among $\{\alpha_1, \beta_1\}$, $\{\alpha_2, \beta_2\}$, $\{\alpha_3, \beta_3\}$. \square

5.3.6 Assembling gadgets

We now combine our gadgets to show that EVEN-FACED EDGE SELECTION is NP-complete.

Lemma 5.4. *EVEN-FACED EDGE SELECTION is NP-complete.*

Proof (reduction from CUBIC PLANAR MONOTONE 1-IN-3 SATISFIABILITY). We first show how to transform an instance $\phi = C_1 \wedge C_2 \wedge \dots \wedge C_n$ of CUBIC PLANAR MONOTONE 1-IN-3 SATISFIABILITY into an instance (G, M) of EVEN-FACED EDGE SELECTION. The transformation proceeds by:

1. mapping each clause $C_j = \{x_{j,1}, x_{j,2}, x_{j,3}\}$ onto a CLAUSE gadget denoted by $\text{CLAUSE}(C_j)$, with interface $\{\alpha_{j,1}, \beta_{j,1}, \alpha_{j,2}, \beta_{j,2}, \alpha_{j,3}, \beta_{j,3}\}$;
2. mapping each variable x_i onto a VAR gadget denoted by $\text{VAR}(x_i)$, whose interface edges coincide with the three pairs of edges $\{\alpha_{j,h}, \beta_{j,h}\}$ such that x_i is the h -th literal of clause j .

Note that $G = (V, E)$ is a cubic planar graph. It is also K_3 -free; indeed, the gadgets are K_3 -free by definition because they are marked graphs. So if a triangle exists it must contain at least two interface edges from some gadget: this is not possible either as those edges may not be incident according to Definition 5.2.

We now show that ϕ is satisfiable iff (G, M) admits an even-faced selection.

\Rightarrow : We apply the following rules for transforming a satisfying assignment for ϕ into a selection E^* for (G, M) :

- if variable x_i is set to TRUE, then select the edges of $\text{VAR}(x_i)$ such that all interface edges are selected in the border, as in Figure 18(a).
- otherwise, select the edges of $\text{VAR}(x_i)$ creating an empty border, as in Figure 18(b).

By this construction, for each clause gadget $\text{CLAUSE}(C_j)$, exactly two edges $\{\alpha_{j,h}, \beta_{j,h}\}$ are selected in the interface. This pair corresponds to a border for this gadget, and we select the corresponding edges within the gadget.

It can be seen that this selection is even-faced. In $(V, E \setminus E^*)$, every face within a VAR or CLAUSE gadget has an even weight since the selection corresponds to an even-faced selection for each one, and every face containing vertices from different gadgets can be seen as a concatenation of outer paths of even weights (since all borders are even), so all faces have an even weight.

\Leftarrow : Consider now a selection E^* for (G, M) . For each variable x_i , we set x_i to TRUE if the border of $\text{VAR}(x_i)$ is the whole interface, and x_i to FALSE otherwise (i.e. if the border is empty, by Proposition 5.4). For each clause C_j , $1 \leq j \leq n$, if a literal $x_{j,h}$ is set to TRUE, then $\alpha_{j,h}, \beta_{j,h} \in E^*$, and $\alpha_{j,h}, \beta_{j,h} \notin E^*$ otherwise. Thus the border of every clause gadget is paired, and by Proposition 5.5, exactly one literal among $x_{j,1}, x_{j,2}, x_{j,3}$ is set to TRUE. Consequently, clause C_j is satisfied and, overall, the formula is satisfied by this assignment.

□

Theorem 5.6. *The $\{K_{1,3}, P_4\}$ -DECOMPOSITION and $\{K_{1,3}, K_3, P_4\}$ -DECOMPOSITION problems are NP-complete for planar cubic graphs.*

Proof. Lemma 5.4 implies that EVEN-FACED EDGE SELECTION is NP-complete. By Lemma 5.3, this problem is equivalent to $\{K_{1,3}, P_4\}$ -DECOMPOSITION OF MARKED GRAPHS. Since this problem reduces to $\{K_{1,3}, P_4\}$ -DECOMPOSITION restricted to K_3 -free planar cubic graphs (Lemma 5.2), both $\{K_{1,3}, P_4\}$ -DECOMPOSITION and $\{K_{1,3}, K_3, P_4\}$ -DECOMPOSITION are NP-complete, even when restricted to planar cubic graphs. □

6 Conclusions and future work

In this paper, we provided a complete computational complexity landscape of the S -DECOMPOSITION problem, where the input graph G is subcubic and S is the set of connected graphs of size 3. We identified all NP-complete subproblems in that setting and provided linear time algorithms for the problems that are solvable in polynomial time. All hard subproblems are NP-complete even if the input graph is both bipartite

and planar, with the exception of decompositions of cubic graphs with a set $S' \supseteq \{K_{1,3}, P_4\}$ where only planarity is guaranteed (Theorem 5.6).

The only remaining nontrivial connected graph of size *at most* 3 that could have been used is P_3 . Lonc and Pszczoła [16] proved that $\{P_3 \cup H\}$ -DECOMPOSITION is in P, where H is a connected graph of odd size. A closer examination of their results (namely, the comments made in [16, page 6373] right after their Theorem 3.6) allows us to conclude that $\{P_3 \cup S'\}$ -DECOMPOSITION for subcubic graphs is in P for any $S' \subseteq \{K_{1,3}, K_3, P_4\}$, because the problem then reduces to finding a subgraph H in G isomorphic to an element of S' and such that $G \setminus H$ contains only components of even size. Therefore, all S -DECOMPOSITION problems on subcubic graphs involving connected graphs of size at most 3 are solved, complementing the cases involving disconnected graphs of size at most 3 covered by Theorem 1.2.

A natural generalisation would be to study decompositions of k -regular graphs into connected components of size k for $k > 3$. Some partial results exist, which do not use all connected graphs of size k ; for instance, $\{P_4\}$ -DECOMPOSITION is known to be in P for 4-regular graphs, but is NP-complete for regular graphs of odd degree ≥ 5 [6]. On the other hand, $\{P_5\}$ -DECOMPOSITION is known to be in P for K_3 -free 5-regular graphs [1], which is obtained by proving a special case of a conjecture stating that if a $(2k+1)$ -regular graph admits a perfect matching, then it admits a $\{P_k\}$ -decomposition [10]. We would like to determine whether our positive results generalise in any way in that setting. It would also be interesting to identify tractable classes of graphs in the cases where those decomposition problems are hard, and to refine our characterisation of hard instances, as well as to obtain elegant graph-theoretical characterisations of easy instances.

References

- [1] F. BOTLER, G. MOTA, AND Y. WAKABAYASHI, *Decompositions of triangle-free 5-regular graphs into paths of length five*, Discrete Mathematics, 338 (2015), pp. 1845–1855.
- [2] A. BOUCHET AND J.-L. FOUQUET, *Trois types de décompositions d'un graphe en chaînes*, in Combinatorial Mathematics: Proceedings of the International Colloquium on Graph Theory and Combinatorics, C. Berge, D. Bresson, P. Camion, J. F. Maurras, and F. Sterboul, eds., vol. 75 of North-Holland Mathematics Studies, North-Holland, 1983, pp. 131–141.
- [3] A. BRANDSTÄDT, V. B. LE, AND J. P. SPINRAD, *Graph classes: a survey*, SIAM Monographs on Discrete Mathematics and Applications, Society for Industrial Mathematics, 1987.
- [4] K. BRYŚ AND Z. LONC, *Polynomial cases of graph decomposition: A complete solution of Holyer's problem*, Discrete Mathematics, 309 (2009), pp. 1294–1326.
- [5] L. BULTEAU, G. FERTIN, A. LABARRE, R. RIZZI, AND I. RUSU, *Decomposing cubic graphs into connected subgraphs of size three*, in Proceedings of the 22nd International Conference on Computing and Combinatorics (COCOON), T. N. Dinh and M. T. Thai, eds., vol. 9797 of Lecture Notes in Computer Science, Ho Chi Minh City, Vietnam, Aug. 2016, Springer, pp. 393–404.
- [6] A. A. DIWAN, J. E. DION, D. J. MENDELL, M. PLANTHOLT, AND S. K. TIPNIS, *The complexity of P_4 -decomposition of regular graphs and multigraphs*, Discrete Mathematics and Theoretical Computer Science, 17 (2015), pp. 63–76.
- [7] D. DOR AND M. TARSI, *Graph decomposition is NP-complete: A complete proof of Holyer's conjecture*, SIAM Journal on Computing, 26 (1997), pp. 1166–1187.
- [8] M. E. DYER AND A. M. FRIEZE, *On the complexity of partitioning graphs into connected subgraphs*, Discrete Applied Mathematics, 10 (1985), pp. 139–153.
- [9] M. E. DYER AND A. M. FRIEZE, *Planar 3DM is NP-complete*, Journal of Algorithms, 184 (1986), pp. 174–184.

- [10] O. FAVARON, F. GENEST, AND M. KOUIDER, *Regular path decompositions of odd regular graphs*, Journal of Graph Theory, 63 (2010), pp. 114–128.
- [11] É. FUSY, *Transversal structures on triangulations: A combinatorial study and straight-line drawings*, Discrete Mathematics, 309 (2009), pp. 1870–1894.
- [12] I. HOLYER, *The NP-completeness of some edge-partition problems*, SIAM Journal on Computing, 10 (1981), pp. 713–717.
- [13] T. P. KIRKMAN, *On a problem in combinatorics*, Cambridge Dublin Mathematical Journal, 2 (1847), pp. 191–204.
- [14] A. KOTZIG, *Z teorie konečných pravidelných grafů třetího a čtvrtého stupně*, Časopis pro pěstování matematiky, (1957), pp. 76–92.
- [15] Z. LONC, *Edge decomposition into isomorphic copies of $sK_{1,2}$ is polynomial*, Journal of Combinatorial Theory, Series B, 69 (1997), pp. 164–182.
- [16] Z. LONC AND M. PSZCZOŁA, *Edge decompositions into two kinds of graphs*, Discrete Mathematics, 309 (2009), pp. 6368–6374.
- [17] S. MICALI AND V. V. VAZIRANI, *An $O(\sqrt{|V||E|})$ algorithm for finding maximum matching in general graphs*, in Proceedings of the 21st Annual Symposium on Foundations of Computer Science (FOCS), Syracuse, New York, Oct. 1980, IEEE Computer Society, pp. 17–27.
- [18] C. MOORE AND J. M. ROBSON, *Hard tiling problems with simple tiles*, Discrete and Computational Geometry, 26 (2001), pp. 573–590.
- [19] X. MUÑOZ, Z. LI, AND I. SAU, *Edge-partitioning regular graphs for ring traffic grooming with a priori placement of the ADMs*, SIAM Journal on Discrete Mathematics, 25 (2011), pp. 1490–1505.
- [20] M. PRIESLER AND M. TARSI, *On the decomposition of graphs into copies of $P_3 \cup tK_2$* , Ars Combinatoria, 35 (1993), pp. 325–333.
- [21] R. YUSTER, *Combinatorial and computational aspects of graph packing and graph decomposition*, Computer Science Review, 1 (2007), pp. 12–26.