



HAL
open science

Complexity of Finding Maximum Locally Irregular Induced Subgraphs

Foivos Fioravantes, Nikolaos Melissinos, Theofilos Triommatis

► **To cite this version:**

Foivos Fioravantes, Nikolaos Melissinos, Theofilos Triommatis. Complexity of Finding Maximum Locally Irregular Induced Subgraphs. [Research Report] Inria; I3S; Université Côte d'Azur. 2021. hal-03358273v1

HAL Id: hal-03358273

<https://hal.science/hal-03358273v1>

Submitted on 29 Sep 2021 (v1), last revised 26 Apr 2022 (v2)

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.

Complexity of Finding Maximum Locally Irregular Induced Subgraphs

Foivos Fioravantes¹, Nikolaos Melissinos², and Theofilos Triommatis³

¹Université Côte d’Azur, Inria, CNRS, I3S, Valbonne, France

²Université Paris-Dauphine, PSL University, CNRS, LAMSADE, 75016, Paris, France

³School of Electrical Engineering, Electronics and Computer Science, University of Liverpool, Liverpool, L69-3BX, UK

Abstract

If a graph G is such that no two adjacent vertices of G have the same degree, we say that G is *locally irregular*. In this work we introduce and study the problem of identifying a largest induced subgraph of a given graph G that is locally irregular. Equivalently, given a graph G , find a subset S of $V(G)$ of minimum order, such that by deleting the vertices of S from G results in a locally irregular graph; we denote with $I(G)$ the order of such a set S . We first treat some easy graph families, namely paths, cycles, trees, complete bipartite and complete graphs. However, we show that the decision version of the introduced problem is \mathcal{NP} -Complete, even for restricted families of graphs, such as subcubic bipartite, or cubic graphs.

Then, looking for more positive results, we turn towards computing the parameter $I(G)$ through the lens of parameterised complexity. In particular, we provide two algorithms that compute $I(G)$, each one considering different parameters. The first one considers the size of the solution k and the maximum degree Δ of G with running time $(2\Delta)^k n^{\mathcal{O}(1)}$, while the second one considers the treewidth tw and Δ of G , and has running time $\Delta^{2tw} n^{\mathcal{O}(1)}$. Therefore, we show that the problem is FPT by both k and tw if the graph has bounded maximum degree Δ . Since these algorithms are not FPT for graphs with unbounded maximum degree (unless we consider $\Delta + k$ or $\Delta + tw$ as the parameter), it is natural to wonder about the existence of an algorithm that does not include additional parameters (other than k or tw) in its dependency. We manage to settle negatively this question, and we show that our algorithms are essentially optimal. In particular, we prove that there is no algorithm that computes $I(G)$ with dependence $f(k)n^{\mathcal{O}(k)}$ or $f(tw)n^{\mathcal{O}(tw)}$, unless the ETH fails.

Keywords: Locally irregular, largest induced subgraph, FPT, treewidth, W-hardness

1 Introduction

A graph G is said to be *locally irregular*, if every two adjacent vertices of G have different degrees. In this paper, we introduce and study the problem of finding a largest locally irregular induced subgraph of a given graph. This problem is equivalent to identifying what is the minimum number k of vertices that must be deleted from G , so that what remains is a locally irregular graph.

The notion of locally irregular graphs was first introduced in [6]. The most interesting aspect of locally irregular graphs, comes from their connection to the so-called 1-2-3 Conjecture, proposed in [23]. Formally, the 1-2-3 Conjecture tells us that for almost every graph, we should be able to put weights from $\{1, 2, 3\}$ on the edges of that graph, so that the colouring that assigns a colour to each vertex equal to the sum of the weights on its adjacent edges, is a proper vertex-colouring of the graph.

As we said earlier, the 1-2-3 Conjecture seems to have some very interesting links to locally irregular graphs. An obvious connection is that this conjecture holds for irregular graphs. Indeed,

assigning weight 1 to all the edges of a locally irregular graph, suffices to produce a proper vertex-colouring, as each vertex receives a colour equal to its degree. Furthermore, there have been taken some steps towards proving that conjecture, which involve edge-decomposition of a graph into a constant number of locally irregular subgraphs, *i.e.*, given G , find an edge-colouring of G using a constant number of colours, such that each colour induces a locally irregular subgraph of G . This is the main motivation behind [6], and it seems to remain interesting enough to attract more attention [8, 26, 29].

Note that the class of locally irregular graphs can be seen as an antonym to that of regular graphs. It is important to state here that there exist several additional such notions. This is mainly due to the very well known fact that non-trivial *irregular* graphs, *i.e.*, graphs with no two vertices (not necessarily adjacent) having the same degree, do not exist (see [13]). Thus, the literature has plenty of slightly different definitions of irregularity (see for example [2, 13, 14, 21, 30]). One way to deal with the nonexistence of irregular graphs, is to define a notion of *local* irregularity. Intuitively, instead of demanding for all vertices of a graph to have different degrees, we are now considering each vertex v separately, and request that the vertices “around” v verify some properties of irregularity. For example, the authors of [3] study graphs G such that for every vertex v of G , no two neighbours of v have the same degree. For an overview of other interesting notions of irregularity (local or otherwise), we refer the reader to [4].

The problem we introduce belongs in a more general and well studied family of problems, which is about identifying a largest induced subgraph of a given graph that verifies a specific property Π . That is, given a graph $G = (V, E)$ and an integer k , does there exist a set $V' \subseteq V$ such that $|V'| \leq k$ and $G[V \setminus V']$ has the specified property Π ? In our case, the property Π is “the induced subgraph is locally irregular”. This generalisation is indeed a classic problem in graph theory, appearing as the INDUCED SUBGRAPH WITH PROPERTY Π (ISP Π for short) problem in [22]. Unfortunately, it was shown in [25], that ISP Π is a hard problem for any property Π that is *hereditary*, *i.e.*, all induced subgraphs of G verify Π if G itself verifies that property.

However, the ISP Π problem remains interesting (one could say that it actually becomes more interesting) even when considering properties Π that are not hereditary. Recently, the authors of [7] studied the problem for Π being “all vertices of the induced subgraph have odd degree”, which clearly is not a hereditary property. Nevertheless, they showed that this is an \mathcal{NP} -hard problem, and they gave an FPT algorithm that solves the problem when parameterised by the rank-width. Continuing with this idea of Π being a property that focuses on the degrees of the given graph, we would also like to mention [12], in which the authors study the ISP Π problem for Π being “the number of vertices of the induced subgraph with degree equal to the maximum degree of G , is at least k ”, where the graph G and the integer k are given in the input. Also, the authors of [1, 5, 28] studied the ISP Π problem, where Π is the rather natural property “the induced subgraph is d -regular”, where d is an integer given in the input (recall that a graph is said to be *d-regular* if all of its vertices have the same degree d). In particular, in [5] it is shown that finding a largest (connected) induced subgraph that is d -regular, is \mathcal{NP} -hard to approximate, even when restricted on bipartite or planar graphs. The authors of [5] also provide linear-time algorithms to solve the problems for graphs with bounded treewidth. In contrast, the authors of [1] take a more practical approach, as they focus on solving the problem for the particular values of $d = 1$ and $d = 2$, their tools including bounds from quadratic programming, Lagrangian relaxation and integer programming.

It is quite clear that, in some sense, the property that interests us lies at the opposite side of the one studied in [1, 5, 28]. However, both properties, “the induced subgraph is locally regular” and “the induced subgraph is locally irregular” are not hereditary. This means that we do not get an \mathcal{NP} -hardness result directly from [25]. Furthermore, in contrast to the hereditary properties, for which the ISP Π problem always admits FPT algorithms (proven in [11, 24]), for the non-hereditary ones this is not always true. Indeed in [28], the authors proved that when considering Π as “the induced subgraph is regular”, the ISP Π problem is W[1]-hard when parameterised by the size of the solution. That is, there should be no $\mathcal{O}^*(f(k)n^c)$ time algorithm for this problem, where c is a constant. For such problems, it is also interesting to see if there exists any algorithm with running time $\mathcal{O}^*(n^{o(k)})$ or $\mathcal{O}^*(f(k)n^{o(k)})$. The authors of [15, 16, 17] provide techniques that can be used to strongly indicate the non-existence of such algorithms, being applied on a variety of W[1]-hard and W[2]-hard problems, such as the INDEPENDENT SET and the DOMINATING SET, parameterised

by the size of their solutions. Usually these lower bounds are shown under the assumption of a weaker version of the EXPONENTIAL TIME HYPOTHESIS, which states that SAT can not be solved in $2^{o(n+m)}$.

Our contribution: We begin in Section 2 by providing the basic notations and definitions that are going to be used throughout this paper. In Section 3, we deal with the complexity of the introduced problem. In particular, in Section 3.1, we show that the problem belongs in \mathcal{P} if the input graph is a path, cycle, tree, complete bipartite or complete graph. We then move on to Section 3, where we prove that finding the maximum induced locally irregular subgraph of a given graph G is \mathcal{NP} -hard, even if G is restricted to being a subcubic bipartite, or a cubic graph.

As the problem seems to be hard even for rather restricted families of graphs, we decide to look into its parameterised complexity. In Sections 4.1 and 4.2 we present two algorithms that compute $I(G)$, each one considering different parameters. The first considers the size of the solution k and the maximum degree Δ of G , and has running time $(2\Delta)^k n^{\mathcal{O}(1)}$, while the second considers the treewidth tw and Δ of G , and has running time $\Delta^{2tw} n^{\mathcal{O}(1)}$. Unfortunately, these algorithms can be considered as being FPT only if Δ is part of the parameter. Therefore, we wondered if we can eliminate the Δ from the dependence. In Section 4.3 we answer this question negatively and we show that our algorithms are essentially optimal. In particular, we present two linear fpt-reductions which prove that the problem is W[2]-hard when parameterised only by the size of the solution and W[1]-hard when parameterised only by the treewidth. Through these reductions, we also show that we can not even have an algorithm that computes $I(G)$ in time $\mathcal{O}^*(f(k)n^{o(k)})$ or $\mathcal{O}^*(f(tw)n^{o(tw)})$, unless the ETH fails.

2 Preliminaries

For notions and definitions on graph theory not explained here, we refer the reader to [19].

Let $G = (V, E)$ be a graph and $G' = (V', E')$ be a subgraph of G (*i.e.*, created by deleting vertices and/or edges of G). Recall first that the subgraph G' is *induced* if it can be created only by deleting vertices of G . That is, for each edge $uv \in E$, if $u, v \in V'$, then $uv \in E'$. For any vertex $v \in V$, let $N_G(v) = \{u \in V : uv \in E\}$ denote the *neighbourhood* of v in G , and let $d_G(v) = |N_G(v)|$ denote the *degree* of v in G . We also define $N_G[v] = N_G(v) \cup \{v\}$. Finally, for any $X \subseteq V$, we define $N_G[X] = \bigcup_{v \in X} N_G[v]$. Note that, whenever the graph G is clear from the context, we will omit the subscript and simply write $N(v)$, $d(v)$, $N[v]$ and $N[X]$.

A *rooted tree* T is an acyclic connected graph with one special vertex r , designated to be the *root* of T . Any vertex u of T with $d(u) = 1$, is called a *leaf* of T . Any vertex of T that is not a leaf, is known as an *internal* vertex. The parent of a vertex $v \neq r$, is the unique vertex that is closest to v in the unique path from v to r .

Let $G = (V, E)$ be a graph. We say that G is *locally irregular* if for every edge $uv \in E$, we have that $d(u) \neq d(v)$. Now, let $S \subseteq V$ be such that $G[V \setminus S]$ is a locally irregular graph; any set S that has this property, is said to be an *irregulator* of G . For short, we will say that S is an *ir*(G). Moreover, let $I(G)$ be the minimum order that any *ir*(G) can have. We will say that S is a *minimum* irregulator of G , for short S is an *ir**(G), if S is an *ir*(G) and $|S| = I(G)$.

We also define the following notion, which generalises *ir*(G). Let $G = (V, E)$ be a graph, $S, X \subseteq V$ and let $G' = G[V \setminus S]$. Now, let $S \subseteq V$ be such that, for each two neighbouring vertices u, v in $X \setminus S$, we have that $d_{G'}(u) \neq d_{G'}(v)$; any set S that has this property, is said to be an *irregulator of X in G* , for short *ir*(G, X). We define the notions of *ir**(G, X) and $I(G, X)$ analogously to the previous definitions.

Finally, recall that a *fixed parameter-tractable* (FPT for short) algorithm, is an algorithm with running time $f(k)n^{\mathcal{O}(1)}$, where f is a computable function and k is the considered parameter. We also make use of what is known as a *linear fpt-reduction*, a type of polynomial reduction such that the size of the parameter of the new problem is linear in regards to the size of the parameter of the original problem. Observe that if we have a linear fpt-reduction from a problem Q with parameter k to a problem Q' with parameter k' and the assumption that Q can not be solved in time $f(k)n_1^{o(k)}$ (where n_1 is the size of the input of Q), then we can conclude that there is no $f(k')n_2^{o(k')}$ time algorithm for Q (where n_2 is the size of the input of Q).

We will now provide some lemmas that will be useful throughout this paper. In the first three lemmas below, we investigate the relationship between $I(G)$ and $I(G, X)$.

Lemma 2.1. *Let $G = (V, E)$ be a graph and let $X \subseteq V$. Then $I(G, X) \leq I(G)$.*

Proof. Let S be an $ir^*(G)$, $G' = G[V \setminus S]$ and $X' = X \setminus S$. Observe that for each pair of vertices u, v such that $u \in X'$ and $v \in N_{G'}(u) \cap X'$, we have that $d_{G'}(u) \neq d_{G'}(v)$, since S is an $ir^*(G)$. It follows that S is also an irregulator of X in G , i.e. S is an $ir(G, X)$, and thus we have that $I(G, X) \leq |S| = I(G)$. \square

Lemma 2.2. *Let $G = (V, E)$ be a graph and $S, X \subseteq V$ such that S is an $ir^*(G, X)$. Then, $S \subseteq N[X]$ and $I(G, X) = I(G[N[X]], X)$.*

Proof. Let S be an $ir^*(G, X)$, $S_1 = S \cap N[X]$ and $S_2 = S \setminus S_1$. It suffices to prove that S_1 is an $ir(G, X)$. Indeed, if $S_1 \subseteq S$ is an $ir(G, X)$, since S is an $ir^*(G, X)$, we can conclude that $S = S_1$ and that $S \subseteq N[X]$ (by definition of S_1).

Assume now that S_1 is not an $ir(G, X)$. Then there exists a pair of vertices u, v where uv is an edge in $G[X \setminus S_1]$ and $d_{G[V \setminus S_1]}(u) = d_{G[V \setminus S_1]}(v)$. Observe that $N[\{u, v\}] \subseteq N[X]$, and thus $N[\{u, v\}] \cap S_2 = \emptyset$. Therefore, $d_{G[V \setminus S]}(u) = d_{G[V \setminus S_1]}(u) = d_{G[V \setminus S_1]}(v) = d_{G[V \setminus S]}(v)$. This is a contradiction since S is an $ir^*(G, X)$.

Now, we will prove that $I(G, X) = I(G[N[X]], X)$. Let S be an $ir^*(G, X)$. Since $S \subseteq N[X]$ and any vertex $v \in X \setminus S$ has $N(v) \subseteq N[X]$, we have that $d_{G[V \setminus S]}(v) = d_{G[N[X] \setminus S]}(v)$. Thus, S is an $ir(G[N[X]], X)$ and $I(G, X) \geq I(G[N[X]], X)$. Now for the opposite direction, let S' be an $ir^*(G[N[X]], X)$. We will show that S' is also an $ir(G, X)$. Since for all $v \in X \setminus S'$, we have $d_{G[V \setminus S']}(v) = d_{G[N[X] \setminus S']}(v)$ (again because $N(v) \subseteq N[X]$) we have that S' is an $ir(G, X)$. Therefore, $I(G, X) \leq I(G[N[X]], X)$. \square

Lemma 2.3. *Let $G = (V, E)$ be a graph, and $X_1, \dots, X_n \subseteq V$ such that $N[X_i] \cap N[X_j] = \emptyset$ for every $1 \leq i < j \leq n$. Then $\sum_{i=1}^n I(G, X_i) \leq I(G)$.*

Proof. Let $X = \bigcup_{i=1}^n X_i$. For every $1 \leq i \leq n$, let S_i be an $ir^*(G, X_i)$ and $G'_i = G[V \setminus S_i]$, and let $S = \bigcup_{i=1}^n S_i$ and $G' = G[V \setminus S]$. Observe first that for every $i \neq j$, since $N[X_i] \cap N[X_j] = \emptyset$, we have that $S_i \cap S_j = \emptyset$ as well. Thus, $|S| = \sum_{i=1}^n |S_i|$.

We will now show that S is an $ir^*(G, X)$. Assume that there exists an S' such that $|S'| < |S|$ and S' is an $ir(G, X)$. Then, there exists a $k \leq n$ such that the set $S'_k = S' \cap N[X_k]$, is such that $|S'_k| < |S_k|$, as otherwise $|S'|$ can not be smaller than $|S|$. Observe that S'_k must be an $ir(G, X_k)$; this holds because for any vertex $u \in S' \setminus S'_k$, we know that $u \notin N[X_k]$. This is a contradiction since we have assumed that S_k is an $ir^*(G, X_k)$ and S'_k is an $ir(G, X_k)$ with size smaller than S_k . Therefore, S is an $ir^*(G, X)$, and the statement follows by Lemma 2.1. \square

Lemma 2.4. *Let $G = (V, E)$ be a graph such that, G is not locally irregular, and S be an $ir^*(G)$. Furthermore let $G_v = (V', E')$ be the graph $G[V \setminus \{v\}]$ for a vertex $v \in S$. Then $I(G_v) = I(G) - 1$.*

Proof. First observe that $S' = S \setminus \{v\}$ must be an $ir(G_v)$ as $G_v[V' \setminus S'] = G[V \setminus S]$. It follows that $I(G_v) \leq I(G) - 1$. Assume that $I(G_v) < I(G) - 1$. Then there exists an S'' such that $|S''| < I(G) - 1$ and S'' is an $ir(G_v)$. Since $G_v[V' \setminus S''] = G[V \setminus (S'' \cup \{v\})]$, we have that $S'' \cup \{v\}$ is an $ir(G)$ and $|S'' \cup \{v\}| = |S''| + 1 < I(G)$. This is a contradiction. \square

The following, almost trivial, observation, will be useful throughout the rest of the paper.

Observation 2.5. *Let $G = (V, E)$ be a graph and S be an $ir(G)$. Then, for each edge $uv \in E$, if $d(u) = d(v)$, then S contains at least one vertex in $N[\{u, v\}]$.*

3 Complexity

In this section, we deal with the (classical) complexity of the problem we introduced. First, we calculate $I(G)$ for some easy families of graphs. Specifically, we show that $I(G)$ can be calculated in linear time when G is a path, a cycle, a complete or a complete bipartite graph, and in polynomial time when G is a tree. Then, we show that finding a minimum irregulator of a graph is \mathcal{NP} -hard. Interestingly, this remains true even for quite restricted families of graphs, such as 3-regular graphs (also known as cubic) and bipartite graphs of maximum degree at most 3.

3.1 Polynomial Cases

Theorem 3.1. *Let G be a graph. If $G = K_n$, then $I(G) = n - 1$. Also, if $G = K_{n,m}$ with $0 \leq n \leq m$, then $I(G) \leq 1$ with the equality holding if and only if $n = m$.*

Proof. Let $G = (V, E)$. Assume that $G = K_n$, and let S be an $ir(G)$ with $|S| < n - 1$. Then $G' = G[V \setminus S]$ is a complete graph of order $n' > n - (n - 1) = 1$, and for any $n' \geq 2$, we have that $K_{n'}$ is not locally irregular, leading to a contradiction.

Observe that $K_{n,m}$, with $0 \leq n < m$, is locally irregular, and thus $I(K_{n,m}) = 0$ in this case. Assume now that $G = K_{n,n}$ with $n \geq 1$. We have that $I(G) \geq 1$ as $K_{n,n}$ is not locally irregular. Let L, R be the two bipartitions of V , with $|L| = n$ and $|R| = n$. Consider the set $S = \{v\}$, where v is any vertex of L . Clearly, after the deletion of v , the graph $G' = G[V \setminus S]$ is isomorphic to $K_{n-1,n}$ which is locally irregular. \square

Theorem 3.2. *Let P_n be the path on n vertices, then*

$$I(P_n) = \begin{cases} \lfloor \frac{n}{4} \rfloor, & \text{if } n \not\equiv 2 \pmod{4} \\ \lfloor \frac{n}{4} \rfloor + 1, & \text{if } n \equiv 2 \pmod{4} \end{cases}$$

Proof. We will begin our proof by examining the cases of P_1, P_2, P_3 , and P_4 . Observe first of all; that P_1 and P_3 are locally irregular graphs. It follows that $I(P_1) = I(P_3) = 0$.

On the other hand, it is also easy to check that P_2 is not locally irregular, but that deleting any one of its vertices suffices to turn it into P_1 (which is locally irregular). It follows that $I(P_2) = 1$. We will now show that $I(P_4) = 1$. Let $P_4 = v_1v_2v_3v_4$ and note that P_4 is not locally irregular (we have that $d(v_2) = d(v_3) = 2$). Moreover, deleting either v_1 or v_4 from P_4 , results in the graph P_3 , which is locally irregular. Thus $I(P_4) = 1$. Observe moreover that any path on more than 4 vertices is not locally irregular.

We are now ready to continue with the proof. Let $n, k, d \in \mathbb{N}$, with $n \geq 5$, $n \equiv k \pmod{4}$, $d = \lfloor \frac{n}{4} \rfloor$ and $G = P_n = v_1 \dots v_n$. We have the following two cases:

- $k \neq 2$. Consider the set $S = \{v_i : i \equiv 0 \pmod{4}\}$. We have that $|S| = d$. Also, observe that the graph $G[V(G) \setminus S]$ has d connected components, each one of which is isomorphic to P_3 , which are locally irregular, and a connected component isomorphic to P_k , where $k \in \{0, 1, 3\}$, which is also locally irregular (the graph P_0 is the null graph). It follows that S is an $ir(P_n)$ and that $I(P_n) \leq |S| = d$. All that is left to show is that $I(P_n) \geq d$. Let us assume that there exists a set S_0 that is an $ir(P_n)$ and $|S_0| < d$. Now observe that $G[V(G) \setminus S_0]$ contains at least one connected component isomorphic to P_m , with $m \geq 4$. This is a contradiction, since P_m is not locally irregular.
- $k = 2$. Consider the set $S = \{v_i : i \equiv 0 \pmod{4}\} \cup \{v_n\}$. We have that $|S| = d + 1$. Similarly to the previous case, we have that $G[V(G) \setminus S]$ contains d connected components isomorphic to P_3 and one connected component isomorphic to P_1 . Thus S is an $ir(P_n)$ and $I(P_n) \leq |S| \leq d + 1$. All that is left to show is that $I(P_n) \geq d + 1$. Observe that the arguments supporting that $I(P_n) > d$ are the same as the previous case. So, we assume that there exists a set S_0 that is an $ir(P_n)$ and $|S_0| = d$. Observe that all the connected components of $G[V(G) \setminus S_0]$ are paths. Also, if there exists a connected component isomorphic to a P_m , with $m \geq 4$, then $G[V(G) \setminus S_0]$ is not locally irregular. So we may assume that all the connected components of $G[V(G) \setminus S_0]$ are isomorphic to paths on at most 3 vertices. It follows that one of these components must be isomorphic to P_2 , and P_2 is not locally irregular. This is a contradiction. \square

Corollary 3.3. *Let C_n be the cycle on $n \geq 3$ vertices, then $I(C_n) = I(P_{n-1}) + 1$*

To explain the above statement, observe that for every vertex v belonging to the cycle $G = C_n$, we have that $d(v) = 2$. Thus, we know that $I(C_n) \geq 1$ and that any S that is an $ir(G)$ contains at least one vertex, say vertex v . The statement follows by observing that the graph $G[V(G) \setminus v]$ is isomorphic to P_{n-1} .

Another interesting case is when G is a tree. In this case, we can calculate $I(G)$ in polynomial time as a direct consequence of Theorem 4.2 (see Section 4.2). Indeed, in that theorem we provide

an algorithm that computes $I(G)$ in time $\Delta^{2tw}n^{\mathcal{O}(1)}$. The next corollary follows directly from the fact that trees have $tw = 1$ (by definition) and $\Delta = \mathcal{O}(n)$.

Corollary 3.4. *Let T be a tree. There exists a polynomial time algorithm that computes $I(T)$.*

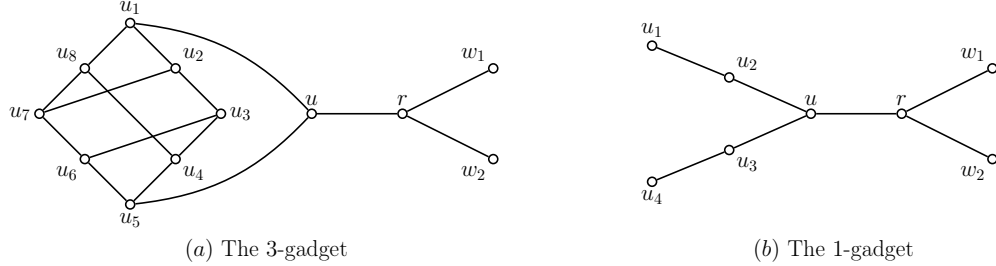


Figure 1: The two gadgets used in the proof of Theorem 3.5.

3.2 \mathcal{NP} -Hard Cases

Theorem 3.5. *Let G be graph and $k \in \mathbb{N}$. Deciding if $I(G) \leq k$ is \mathcal{NP} -complete. The same is true if G is a*

- cubic graph, or
- a bipartite graph with maximum degree $\Delta \leq 3$.

Proof. Let us focus first on proving the first item of the statement. Since the problem is clearly in \mathcal{NP} , we will focus on proving it is also \mathcal{NP} -hard. The reduction is from 2-BALANCED 3-SAT, which was proven to be \mathcal{NP} -complete in [9]. In that problem, a 3CNF formula F is given as an input, comprised by a set C of clauses over a set of Boolean variables X . In particular, we have that each clause contains exactly 3 literals, and each variable $x \in X$ appears in F exactly twice as a positive and twice as a negative literal. The question is, whether there exists a truth assignment to the variables of X satisfying F .

Let F be a 3CNF formula with m clauses C_1, \dots, C_m and n variables x_1, \dots, x_n that is given as input to the 2-BALANCED 3-SAT problem. We will construct a cubic graph G such that F is satisfiable if and only if $I(G) \leq 3n$. To construct $G = (V, E)$, we start with the following graph: for each literal x_i ($\neg x_i$ resp.) in F , add a *literal vertex* v_i (v'_i resp.) in V , and for each clause C_j of F , add a *clause vertex* c_j in V . Next, for each $1 \leq j \leq m$, add the edge $v_i c_j$ ($v'_i c_j$ resp.) if the literal x_i ($\neg x_i$ resp.) appears in C_j according to F . Observe that the resulting graph is bipartite, for each clause vertex c we have $d(c) = 3$ and for each literal vertex v we have $d(v) = 2$ (since in F , each variable appears twice as a positive and twice as a negative literal). To finish the construction of G , we will make use of the 3-gadgets, illustrated in Figure 1(a). When we say that we *attach* a copy H of the 3-gadget to the vertices v_i and v'_i (for some $1 \leq i \leq n$), we mean that we add H to G , and we identify the vertices w_1 and w_2 to the vertices v_i and v'_i respectively. Now, for each pair of literal vertices $\{v_i, v'_i\}$, attach one copy H_i of the 3-gadget to the vertices v_i and v'_i (see Figure 2). Clearly this construction is achieved in linear time in regards to $n + m$. Note also that the resulting graph G is cubic. Before we move on with the reduction, we state the following claim:

Claim 3.6. *Let H be a copy of the 3-gadget, shown in Figure 1(a), and $X = V(H) \setminus \{w_1, w_2\}$. We have the following:*

- $I(H, X) = 3$
- For any S that is an $ir^*(H, X)$, we have that $r \notin S$;
- for $w \in \{w_1, w_2\}$, if $w \in S$, with S being an $ir^*(H, X)$, then $S = \{u_4, u_8, w\}$. Furthermore, these are the only optimal irregularators of X in H that contain either w_1 or w_2 .

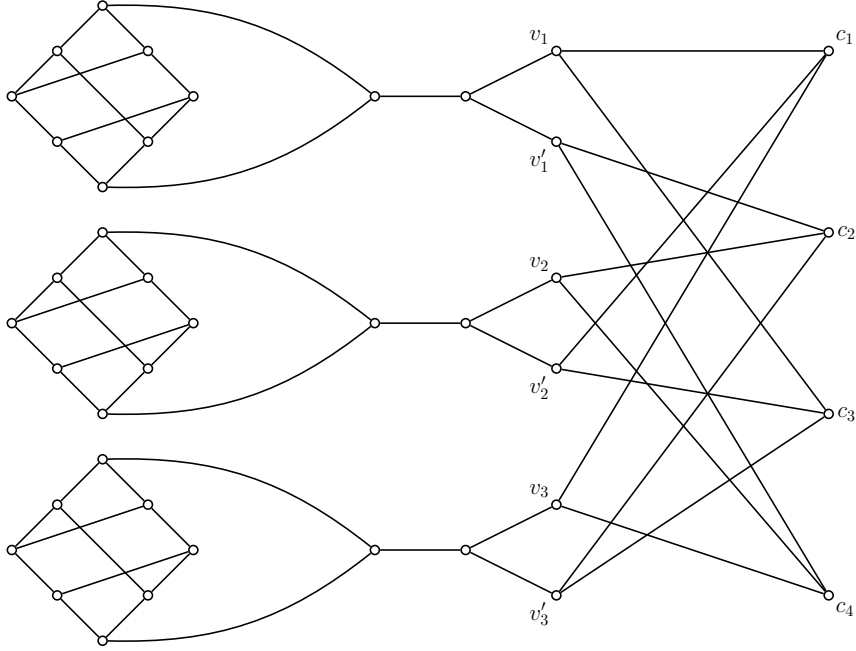


Figure 2: An example of the construction of the cubic graph G in the proof of Theorem 3.5, starting the input formula $F = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3) \wedge (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$. The formula has the $n = 3$ variables x_1, x_2, x_3 , and $m = 4$ clauses. For $1 \leq i \leq n$, the vertex v_i (v'_i resp.) corresponds to the appearances of the literal x_i ($\neg x_i$ resp.) in F .

Proof of the claim. First we will show that if S is an $ir(H, X)$, then $|S| \geq 3$. Clearly, if $|S| = 1$, then S cannot be an $ir(H, X)$. Assume now that $|S| = 2$ and consider the edges ur , u_2u_7 and u_4u_5 . These edges have in common that both of their incident vertices have the same degree (which is equal to 3). It follows from Observation 2.5 that S contains at least one vertex in each one of the sets $S_1 = N(uv) = \{u_1, u_5, u, r, w_1, w_2\}$, $S_2 = N(u_2u_7) = \{u_1, u_2, u_3, u_6, u_7, u_8\}$ and $S_3 = N(u_4u_5) = \{u_3, u_4, u_5, u_6, u_8\}$. Assume first that $u_1 \in S$. In Figure 4 we illustrate all possible subsets of vertices of H of order 2 that contain u_1 . Clearly none of them is an $ir(H, X)$. It follows that $u_1 \notin S$. Note that due to symmetry, we can also deduce that $u_5 \notin S$. It follows that $S \cap S_1 \subseteq \{u, r, w_1, w_2\}$ and, since $|S| = 2$, that the remaining vertex w of S belongs to $(S_2 \cap S_3) \setminus \{u_1, u_5\} = \{u_3, u_6, u_8\}$. Let $H' = H[V(H) \setminus S]$. It is easy to see that if $w = u_3$ then $d_{H'}(u_7) = d_{H'}(u_8)$, if $w = u_6$ then $d_{H'}(u_3) = d_{H'}(u_4)$ and that if $w = u_8$ then $d_{H'}(u_3) = d_{H'}(u_6)$ (and this holds true for any possible combination of vertices $w \in \{u_3, u_6, u_8\}$ and $w' \in \{u, r, w_1, w_2\}$). Thus S cannot be an $ir(H, X)$ and $I(H, X) \geq 3$. For the rest of the claims, it suffices to find all the irregularators of X in H of order 3. By doing an exhaustive search, we were able to identify these irregularators. They are (up to symmetry) the following: $\{u_1, u_8, u_7\}$, $\{u_1, u_5, u_7\}$, $\{u_1, u_3, u_7\}$, $\{u_4, u_8, u\}$, $\{u_4, u_8, w_1\}$ and $\{u_4, u_8, w_2\}$. \diamond

We are now ready to show the equivalence between finding a satisfying truth assignment t of F , and finding an S that is an $ir(G)$ such that $|S| = 3n$ (from which follows that $I(G) \leq 3n$).

Let t be a satisfying truth assignment of F , and let S' be the set of literal vertices v_i (v'_i resp.) such that the corresponding literals x_i ($\neg x_i$ resp.) are assigned value *true* by t . Now, for each copy H_i of the gadget which was used in the construction of G , let $S'_i = \{u_4, u_8, \alpha\}$, where $\alpha \in S'$, and consider the set $S = \bigcup_{i=1}^n S'_i$. Note that $|S| = 3n$. We will show that S is an $ir(G)$. Since t is a satisfying truth assignment of F , each clause C_j contains at least one literal that is set to *true*. In other words, the clause vertex c_j is adjacent to at least one literal vertex that belongs to S . Let $G' = G[V(G) \setminus S]$, and note that $d_{G'}(c_j) \leq 2$ (for every $1 \leq j \leq m$), while the degree of all the literal vertices of G' is equal to 3. It follows, from the previous observations and from Claim 3.6, that S is an $ir(G)$. Furthermore, by the construction of S , we know that for any literal vertex v of G' , the copy of the vertex r which incident to v , has degree 2 in G' . Finally, observe that $S \cap V(H_i)$ is an $ir(H_i, X_i)$ (by the construction of S) and that deleting any vertex of G that does

not belong to $N[X_i]$, does not change the degree of any vertex in $N[X_i]$. So, we can conclude that there are no two vertices in X_i that have the same degrees in G' . Thus S is an $ir(G)$.

For the other direction, assume that $I(G) \leq 3n$ and let S be an $ir^*(G)$. For $1 \leq i \leq n$, let $X_i = V(H_i) \setminus \{v_i, v'_i\}$ and observe that $N[X_i]$ contains exactly the vertices of the gadget H_i . Also, let $S_i = S \cap V(H_i)$, for all $i \in \{1, \dots, n\}$. First we are going to prove some properties of S :

Claim 3.7. *For the given set S , the following properties hold for all $i \in \{1, \dots, n\}$:*

1. S_i is an $ir^*(H_i, X_i)$.
2. $S = \bigcup_{i=1}^n S_i$.
3. the vertex r belonging to the H_i gadget, does not belong to S .
4. If $v_i \in S$ (for some i), then $v'_i \notin S$ and vice versa.
5. For all $j \in \{1, \dots, m\}$, we have that $c_j \notin S$.

Proof of the claim. For the first item, let us first show that S_i is an $ir(H_i, X_i)$. Assume that S_i is not an $ir(H_i, X_i)$; then there exist two vertices u, v in $X_i \setminus S_i$ that have the same degree in $G[V \setminus S_i]$. Since $S \setminus S_i$ does not include any vertices of H_i , we know that u, v belong in $G[V \setminus S]$. This is a contradiction since S is an $ir^*(G)$. In order to show that S_i is actually an $ir^*(H_i, X_i)$, we need to take in consideration the order of S . Since S_i is an $ir(H_i, X_i)$, we have that $|S_i| \geq 3$. Assume that there exist an i such that $|S_i| > 3$. Since $S_i \subseteq N_G[X_i]$ and $N_G[X_i] \cap N_G[X_j] = \emptyset$, for every $i \neq j$, we have that $|S| \geq \sum_{i=1}^n |S_i| > 3n$. This is a contradiction because $|S| \leq 3n$. Thus the first item holds.

The rest of the items follow from the first item. For the second item, we just need to observe that $\sum_{i=1}^n |S_i| = 3n$ and $|S| \leq 3n$. Therefore, S can not contain any other vertex. The third and four item follow by the facts that $S_i = S \cap V(H_i)$, the first item and Claim 3.6. Finally, the fifth item holds because $c_j \notin N_G[X_i]$, for any $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, m\}$, and $S = \bigcup_{i=1}^n S_i \subseteq \bigcup_{i=1}^n N_G[X_i]$. \diamond

Before we give the truth assignment let us note two more things for some vertices of $G' = G[V \setminus S]$. First, any literal vertex v that belongs in G' , has $d_{G'}(v) = 3$ since S does not contain any of the neighbours of v . Furthermore, for each clause vertex c_j , there must exist a literal vertex $v \in N(c_j)$ such that $v \in S$, as otherwise c_j would have the same degree as all its neighbours in G' .

Now consider the following truth assignment: we assign the value *true* to every variable x_i if the corresponding literal vertex v_i belongs in S , and value *false* to every other variable. Now, since for every $1 \leq j \leq m$ we have that $d_{G'}(c_j) < 3$, it follows that each clause C_j contains either a positive literal x_i which has been set to *true*, or a negative literal $\neg x_i$ which has been set to *false*. Thus F is satisfied.

Now, for the second item of the statement, the proof is similar to the one described above. The reduction is done once more from the 2-BALANCED 3-SAT problem. The main difference lies in the gadget used in the construction of G . For this construction, we make use of the 1-gadget H (shown in Figure 1(b)). Other than that, the construction of G is exactly the same. Observe that this time, the resulting graph G is subcubic bipartite. Furthermore, for $X = V(H) \setminus \{w_1, w_2\}$, it is fairly easy to check that $I(H, X) = 1$. Also, any S that is an $ir^*(H, X)$, is such that $S = \{w\}$, with $w \in \{w_1, w_2\}$, and these are the optimal irregularators of X in H (this plays a similar role as Claim 3.6). The rest of the proof consists in showing the equivalence between finding a satisfying truth assignment for a formula F and deciding if $I(G) \leq n$, where G is the graph that corresponds to F , and it works exactly as the proof of the first item. \square

4 Parameterised Complexity

As the problem of computing a minimal irregularator of a given graph G seems to be rather hard to solve, we focused our efforts towards finding parameterised algorithms that can solve it. In Section 4.1 we present an FPT algorithm that calculates $I(G)$ when parameterised by the size of the solution and Δ , the maximum degree of the graph. Then, we turn our attention towards graphs that are “close to being trees”, that is graphs of bounded treewidth. Indeed, in Section 4.2 we provide an FPT algorithm that finds a minimum irregularator of G , when parameterised by the

treewidth of the input graph and by Δ . Observe that both of our algorithms have to consider Δ as part of the parameter if they are to be considered as FPT. The natural question to ask at this point is whether we can have an FPT algorithm, when parameterised only by the size of the solution, or the treewidth of the input graph. In Section 4.3, we give a strong indication towards the negative answer for both cases, proving that, in some sense, the algorithms provided in Sections 4.1 and 4.2 are optimal.

4.1 FPT by the Size of the Solution and Δ

Theorem 4.1. *For a given graph $G = (V, E)$ with $|V| = n$ and maximum degree Δ , there exists an algorithm that decides if $I(G) \leq k$ in time $(2\Delta)^k n^{\mathcal{O}(1)}$.*

Proof. In order to decide if $I(G) \leq k$ we are going to use a recursive algorithm. The algorithm has input (G, k) , where $G = (V, E)$ is a graph and $k \geq 0$ is an integer. The basic idea of this algorithm, is to take advantage of Observation 2.5. We present the exact procedure in Algorithm 1.

Algorithm 1 [IsIrregular(G, k) decision function]

Input: A graph $G = (V, E)$ and an integer $k \geq 0$.

Output: Is $I(G) \leq k$ or not?

```

1: if  $G$  is irregular then
2:   return yes
3: else if  $k = 0$  then
4:   return no
5: else ▷  $k > 0$  and  $G$  is not irregular
6:    $ans \leftarrow no$ 
7:   find an edge  $vu \in E$  such that  $d_G(v) = d_G(u)$ 
8:   for all  $w \in N_G[\{u, v\}]$  do
9:     set  $G_w = G[V \setminus \{w\}]$ 
10:    if IsIrregular( $G_w, k - 1$ ) returns yes then
11:       $ans \leftarrow yes$ 
12:   return  $ans$ 

```

Now, let us argue about the correctness and the efficiency of this algorithm. We claim that for any graph $G = (V, E)$ and any integer $k \geq 0$, Algorithm 1 returns yes if $I(G) \leq k$ and no otherwise. Furthermore, the number of steps that the algorithm requires, is $f(k, n) = (2\Delta)^k n^{\mathcal{O}(1)}$, where $n = |V|$. We will prove this by induction on k .

Base of the induction ($k = 0$): Here, we only need to check if G is locally irregular. Algorithm 1 does this in line 1 and returns yes if it is (line 2) and no otherwise (line 4). Furthermore, we can check if G is locally irregular in polynomial time. So, the claim is true for the base.

Induction hypothesis ($k = k_0 \geq 0$): We assume that we have a $k_0 \geq 0$ such that Algorithm 1 can decide if any graph G with n vertices and maximum degree Δ has $I(G) \leq k_0$ in $f(k_0, n) = (k_0 + 1)(2\Delta)^{k_0} n^{\mathcal{O}(1)}$ steps.

Induction step ($k = k_0 + 1$): Let $G = (V, E)$ be a graph. If G is locally irregular then $I(G) = 0$ and Algorithm 1 answers correctly (in line 2). Assume that G is not locally irregular; then there exist an edge $vu \in E$ such that $d_G(v) = d_G(u)$. Now, let S be an $ir^*(G)$. It follows from Observation 2.5 that S must include at least one vertex $w \in N_G[\{v, u\}]$. Since Algorithm 1 considers all the vertices in $N_G[\{v, u\}]$, at some point it also considers the vertex $w \in S \cap N_G[\{v, u\}]$. Now, observe that for any $x \in S$, the set $S_x = S \setminus \{x\}$ is an $ir^*(G_x)$, where $G_x = G[V \setminus \{x\}]$. Furthermore, by Lemma 2.4, we have $I(G_x) \leq k - 1 = k_0$ iff $I(G) \leq k$. By the induction hypothesis, we know that the algorithm answers correctly for all the instances (G_x, k_0) . Thus, if $I(G) \leq k = k_0 + 1$, there must exist one instance (G_w, k_0) , where $w \in S \cap N_G[\{v, u\}]$, for which the Algorithm 1 returns yes. Therefore the algorithm answers for $(G, k_0 + 1)$ correctly. Finally, this process request $n^{\mathcal{O}(1)}$ steps in order to check if the graph is locally irregular and $2\Delta f(k - 1, n - 1)$ steps (by induction hypothesis) in order to check if for any graph G_x we have $I(G_x) \leq k - 1 = k_0$ (where $x \in N[\{u, v\}]$). So, the algorithm decides in $n^{\mathcal{O}(1)} + 2\Delta f(k - 1, n - 1) \leq n^{\mathcal{O}(1)} + 2\Delta k (2\Delta)^{k-1} (n - 1)^{\mathcal{O}(1)} \leq$

$n^{\mathcal{O}(1)} + k(2\Delta)^k n^{\mathcal{O}(1)} \leq (k+1)(2\Delta)^k n^{\mathcal{O}(1)}$ steps. Finally, note that $k \leq n-1$, and the result follows. \square

It is worth noting that Algorithm 1 can be slightly modified in order to also return a set S that is an $ir^*(G)$. To do that, it suffices to start with S being the empty set, and update this set every time Algorithm 1 “deletes” a vertex w , by adding this w to S .

4.2 FPT by Treewidth and Δ

Theorem 4.2. *For a given a graph $G = (V, E)$ with treewidth tw and maximum degree Δ , there exists an algorithm that returns $I(G)$ in time $\Delta^{2tw} n^{\mathcal{O}(1)}$.*

Proof. As the techniques we are going to use are standard, we are sketching some of the introductory details. For more details on tree decompositions (definition and terminology) see [20]. Assuming that we have a *nice tree decomposition* (defined in [10]) of the graph G rooted at a node r , we are going to perform dynamic programming on the nodes of this tree decomposition. For a node t of the given tree decomposition of G , we denote by B_t the bag of this node and by B_t^\downarrow the set of vertices of the graph that appears in the bags of the nodes of the subtree with t as a root. Observe that $B_t \subseteq B_t^\downarrow$.

The idea behind our algorithm, is that for each node t we store all the sets $S \subseteq B_t^\downarrow$ such that S is an $ir(G, B_t^\downarrow \setminus B_t)$. We will also store the necessary “conditions” (explained more in what follows) such that if there exists a set S' , where $S' \setminus S \subseteq V \setminus B_t^\downarrow$, that meets these conditions, then S' is an $ir(G, B_t^\downarrow)$. Observe that if we manage to do such a thing for every node of the tree decomposition, then we can find $I(G)$. To do so, it suffices to check the size of all the irregularators we stored for the root r of the tree decomposition, which also meet the conditions we have set. In that way, we can find a set S that is an $ir(G, B_r^\downarrow \setminus B_r)$, satisfies our conditions and is of minimum order, and since $B_r^\downarrow = V$, this set S is a minimum irregularator of G and $I(G) = |S|$.

Let us now present the actual information we are keeping for each node. Assume that t is a node of the tree decomposition and $S \subseteq B_t^\downarrow$ is an irregularator of $B_t^\downarrow \setminus B_t$ in G , *i.e.*, S is an $ir^*(G, B_t^\downarrow \setminus B_t)$. For this S we want to remember which vertices of B_t belong to S as well as the degrees of the vertices $v \in B_t \setminus S$ in $G[B_t^\downarrow \setminus B_t]$. This can be done by keeping a table D of size $tw+1$ where, if $v \in B_t \setminus S$ we set $D(v) = d_{G[B_t^\downarrow \setminus B_t]}(v)$ and if $v \in B_t \cap S$ we set $D(v) = \emptyset$ (slightly abusing the notation, by $D(v)$ we mean the position in the table D that corresponds to the vertex v). Like we have already said, we are going to keep some additional information about the conditions that could allow these sets to be extended to irregularators of B_t^\downarrow in G if we add vertices of $V \setminus B_t^\downarrow$. For that reason, we are also going to keep a table with the “*target degree*” of each vertex; in this table we assign to each vertex $v \in B_t \setminus S$ a degree d_v such that, if there exists S' where $S' \setminus S \subseteq V \setminus B_t^\downarrow$ and for all $v \in B_t \setminus S$ we have $d_{G[V \setminus S]}(v) = d_v$, then S is an $ir(G, B_t^\downarrow)$. This can be done by keeping a table T of size $tw+1$ where for each $v \in B_t \setminus S$ we set $T(v) = i$, where i is the target degree, and for each $v \in B_t \cap S$ we set $T(v) = \emptyset$. Such tables T will be called *valid* for S in B_t . Finally, we are going to keep the set $X = S \cap B_t$ and value $min = |S|$. Note that the set X does not give us any extra information, but we keep it as it will be useful to refer to it directly.

To sum up, for each node t of the tree decomposition of G , we keep a set of quadruples (X, D, T, min) , each quadruple corresponding to a valid combination of a set S that is an $ir(G, B_t^\downarrow \setminus B_t)$ and the target degrees for the vertices of $B_t \setminus S$. Here it is important to say that when treating the node B_t , for every two quadruples (X_1, D_1, T_1, min_1) and (X_2, D_2, T_2, min_2) such that for all $v \in B_t$ we have that $D_1(v) = D_2(v)$ and $T_1(v) = T_2(v)$ (this indicates that $X_1 = X_2$ as well), then we are only going to keep the quadruple with the minimum value between min_1 and min_2 as we will prove that this is enough in order to find $I(G)$.

Claim 4.3. *Assume that for a node t , we have two sets S_1 and S_2 that are both $ir(G, B_t^\downarrow \setminus B_t)$, and that T is a target table that is common to both of them. Furthermore, assume that $(X_1, D_1, T, |S_1|)$ and $(X_2, D_2, T, |S_2|)$ are the quadruples we have to store for S_1 and S_2 respectively (both respecting T), with $D_1(v) = D_2(v)$ for every $v \in B_t$. Then for any set $S \subseteq V \setminus B_t^\downarrow$ such that $d_{G[V \setminus (S_1 \cup S)]}(v) = T(v)$ for all $v \in B_t$, we also have that $d_{G[V \setminus (S_2 \cup S)]}(v) = T(v)$ for all $v \in B_t$.*

Proof of the claim. Assume that we have such an S for S_1 , let v be a vertex in B_t and $H = G[v \cup ((V \setminus B_t^\downarrow) \setminus S)]$ (observe that H does not depend on S_1 or S_2). Since $d_{G[V \setminus (S_1 \cup S_2)]}(v) = T(v)$, we know that in the graph H , v has exactly $T(v) - D_1(v)$ neighbours (as $D_1(v) = d_{G[B_t^\downarrow \setminus S_1]}(v)$). Now, since $D_1(v) = D_2(v) = d_{G[B_t^\downarrow \setminus S_2]}(v)$ we have that $d_{G[V \setminus S_2 \cup S]}(v) = T(v)$. Therefore, the claim holds. \diamond

Simply put, Claim 4.3 states that for any two quadruples $Q_1 = (X, D, T, \min_1)$ and $Q_2 = (X, D, T, \min_2)$, any extension S of S_1 is also an extension of S_2 (where S_1 and S_2 are the two sets that correspond to Q_1 and Q_2 respectively). Therefore, in order to find the minimum solution, it is sufficient to keep the quadruple that has the minimum value between \min_1 and \min_2 .

Now we are going to explain how we create all the quadruples (X, D, T, \min) for each type of node in the tree decomposition. First we have to deal with the Leaf Nodes. For a Leaf node t we know that $B_t = B_t^\downarrow = \emptyset$. Therefore, we have only one quadruple (X, D, T, \min) , where the size of both D and T are zero (so we do not need to keep any information in them), $S = \emptyset$ and $\min = |S| = 0$.

Now let t an Introduce node; assume that we have all the quadruples (X, D, T, \min) for its child c and let v be the introduced vertex. By construction, we know that v is introduced in B_t and thus it has no neighbours in $B_t^\downarrow \setminus B_t$. It follows that if $S \subseteq B_c^\downarrow$ is an irregulator for $B_c^\downarrow \setminus B_c$, then both S and $S \cup \{v\}$ are irregulators for $B_t^\downarrow \setminus B_t$ in G . Furthermore, there is no set $S \subseteq B_t^\downarrow \setminus \{v\}$ that is an irregulator of $B_t^\downarrow \setminus B_t$ and is not an irregulator of $B_c^\downarrow \setminus B_c$. So, we only need to consider two cases for the quadruples we have to store for c ; if v belongs in the under-construction irregulator of $B_t^\downarrow \setminus B_t$ in G or not.

Case 1. (v is in the irregulator): Observe that for any S that is an $ir(G, B_c^\downarrow \setminus B_c)$, which is stored in the quadruples of B_c , for every $u \in B_c \setminus S$, we have that $d_{G[B_c^\downarrow \setminus S]}(u) = d_{G[B_t^\downarrow \setminus (S \cup \{v\])]}(u)$. Moreover, for any target table T which is valid for S in c , the target table T' is valid for $S \cup \{v\}$ in t , where T' is almost the same as T , the only difference being that T' also contains the information about v , i.e., $T'(v) = \emptyset$. So, for each quadruple (X, D, T, \min) in c , we need to create one quadruple $(X \cup \{v\}, D', T', \min + 1)$ for t , where D' is the almost the same as D , except that it also contains the information about v , i.e., $D'(v) = \emptyset$.

Case 2. (v is not in the irregulator): Let $q = (X, D, T, \min)$ be a stored quadruple of c and S be the corresponding $ir(G, B_c^\downarrow \setminus B_c)$. We will first explain how to construct D' of t , based on q . Observe that the only change between $G[B_c^\downarrow \setminus S]$ and $G[B_t^\downarrow \setminus S]$, is that in the latter there exist some new edges from v to some of the vertices of B_c . Therefore, for each vertex $u \in B_c \setminus X$ we set $D'(u) = D(u) + 1$ if $u \in N[v]$ and $D'(u) = D(u)$ otherwise. Finally, for the introduced vertex v , we set $D'(v) = |N(v) \cap (B_c \setminus X)|$. We will now treat the target degrees for t . Observe that the target degrees for each vertex in $B_t \setminus \{v\}$ are the same as in T , since v only has edges incident to vertices in B_t . Now, we only need to decide which are the valid targets for v . Since $d_{G[B_t^\downarrow \setminus S]}(v) = D'(v)$, we know that for every target t' , we have that $D'(v) \leq t' \leq \Delta$. Furthermore, we can not have the target degrees of v to be the same as the targets of one of its neighbours in B_c (these values are stored in T), as, otherwise, any valid target table T' of t would lead to adjacent vertices in B_t having the same degree. Let $\{t_1, \dots, t_k\} \subset \{D(v), \dots, \Delta\}$ be an enumeration of all the valid targets for v (i.e. $t_i \neq T(u)$ for all $u \in N[v] \cap B_c \setminus X$). Therefore, for each quadruple (X, D, T, \min) in c , and for each $i = 1, \dots, k$, we need to create the quadruple (X, D', T_i, \min) , such that $T_i(u) = T(u)$ for all $u \in B_c$ and $T_i(v) = t_i$. In total, we have $k \leq \Delta$ such quadruples.

Now, let us explain how we deal with a Join node. Assume that t is a Join Node with c_1 and c_2 as its two children in the tree decomposition. Here, it is important to mention that $B_{c_1} = B_{c_2}$ and $(B_{c_1}^\downarrow \setminus B_{c_1}) \cap (B_{c_2}^\downarrow \setminus B_{c_2}) = \emptyset$. Assume that there exists an irregulator S of $B_t^\downarrow \setminus B_t$ in G , a valid target table T of S , and let (X, D, T, \min) be the quadruple we need to store in t for this pair (S, T) . Observe that this pair (S, T) is valid for both c_1 and c_2 , so we must already have stored at least one quadruple in each node. Let $X \subseteq B_t$ and a target table T such that (X, D_1, T, \min_1) and (X, D_2, T, \min_2) are stored for c_1 and c_2 respectively. We create the quadruple (X, D, T, \min) for t by setting $D(u) = D_1(u) + D_2(u) - d_{G[B_t \setminus X]}(u)$ for all $u \in B_t \setminus X$, $D(u) = \emptyset$ for all $u \in X$ and $\min = \min_1 + \min_2 - |X|$. Observe that these are the correct values for the $D(u)$ and \min , as otherwise we would count $d_{G[B_t \setminus X]}(u)$ and $|X|$ twice. Finally, we need to note that we do not store any quadruple (X, D, T, \min) we create for the Join Node such that $D(u) > T(u)$ for a vertex

$u \in B_t \setminus X$. This is because for such quadruples, the degree of vertex u will never be equal to any of the target degrees we have set, as it can only increase when we consider any of the ancestor (*i.e.* parent, grandparent etc.) nodes of B_t .

Finally, we need to treat the Forget nodes. Let t be a Forget node, c be the its child and v be the forgotten vertex. Assume that we have to store in t a quadruple (X, D, T, \min) . Then, since $X = B_t \cap S$ for an irregulator S of B_t in G , we know that in c we must have already stored a quadruple (X', D', T', \min') such that, $X' = S \cap B_c$, $D'(u) = D(u)$ for all $u \in B_c$, $T'(u) = T(u)$ for all $u \in B_c$ and $\min' = \min$. Therefore, starting from the stored quadruples in c , we can create all the quadruples of t . For each quadruple (X', D', T', \min') in c , we create at most one quadruple (X, D, T, \min) for t by considering two cases; the forgotten vertex v_f belongs to X' or not.

Case 1. (v belongs to X'): then the quadruple (X, D, T, \min) is almost the same as (X', D', T', \min') , with the following differences: $X = X' \setminus \{v\}$, $\min = \min'$, $D(u) = D'(u)$ and $T(u) = T'(u)$ for all $u \in B_t$ and the tables D and T do not include any information for v as this vertex does not belong to B_t anymore.

Case 2. (v does not belong to X'): we will first check if $D'(v_f) = T'(v_f)$ or not. This is important because the degree of the v will never again be considered by our algorithm, and thus its degree will remain unchanged. So, if $D'(v_f) = T'(v_f)$, we create the quadruple (X, D, T, \min) where $X = X'$, $\min = \min'$, $D(u) = D'(u)$ and $T(u) = T'(u)$ for all $u \in B_t$ and the tables D and T do not include any information for v .

For the running time, we need to observe that the number of nodes of a nice tree decomposition is $\mathcal{O}(tw \cdot n)$ and all the other calculations are polynomial in $n + m$. Thus we only need to count the different quadruples in each node. Now, for each vertex v , we either include it in X or we have $\Delta + 1$ options for the value $D(u)$ (the degree could have any value between 0 and Δ) and $\Delta + 1 - i$ for the value $T(u)$ if the $D(u) = i$ (we can not have $T(u) < D(u)$). Also, for sufficiently large Δ , we have that $1 + \sum_{i=0}^{\Delta} (\Delta + 1 - i) < \Delta^2$. Furthermore, the set X and the value \min do not increase the number of quadruples because $X = \{u \mid D(u) = \emptyset\}$ and from all quadruples (X, D_1, T_1, \min_1) , (X, D_2, T_2, \min_2) such that $D_1(u) = D_2(u)$ and $T_1(u) = T_2(u)$ for all $u \in B_t$ we only keep one of them (by Claim 4.3).

In total, the number of different quadruples in each node are Δ^{2tw} , and therefore the algorithm decides in $\Delta^{2tw} n^{\mathcal{O}(1)}$ time. \square

Once more, the algorithm presented in the proof of Theorem 4.2 can be adapted to also return an $ir^*(G)$. To do so, it suffices to also store a set S' for every node c , such that S' is an $ir(G, B_c^{\downarrow} \setminus B_c)$ that satisfies the properties demanded by the quadruple.

4.3 W-Hardness

In this section we present two linear-fpt reductions. The first is from the DOMINATING SET problem, when parameterised by the size of the solution, and the second is from the LIST COLOURING problem, when parameterised by the treewidth of the graph.

Theorem 4.4. *Let G be a graph and $k \in \mathbb{N}$. Deciding if $I(G) \leq k$ is W[2]-hard, when parameterised by k .*

Proof. The reduction is from the DOMINATING SET problem, which was shown to be W[2]-complete when parameterised by the size of the solution (*e.g.* in [18]). In that problem, a graph $H = (V, E)$ and an integer k are given as input. The question asked, is whether there exists a set $D \subseteq V$ of order at most k (called a *dominating set of H*), such that $V = N[D]$.

Let $H = (V, E)$ be a graph and $k \in \mathbb{N}$. We will construct a graph $G = (V', E')$ such that H has a dominating set of order at most k if and only if G has an irregulator of order at most k . We begin by defining an arbitrary enumeration of the vertices of V . That is, $V = \{v_1, \dots, v_n\}$. The graph G is built starting from a copy of the graph H . To avoid any confusion in what is to follow, we will always use H to denote the original graph, and $G|_H = G[\{v'_1, \dots, v'_n\}]$ to denote the copy of H that lies inside G (where the indices of the v'_i 's are the same as the indices of the corresponding v_i 's). Then, for each $1 \leq i \leq n$, we attach the necessary number of pending vertices (meaning vertices of degree 1) to the vertex v'_i , so that the degree of v'_i becomes equal to $i \cdot n$. Finally, for each v'_i , let u'_i be one of its newly attached pending vertices, and attach the necessary

number of new pending vertices to u'_i , so that its degree becomes equal to that of v'_i . The resulting graph is G . To be clear, for every vertex v of G , we either have that $v = v'_i$ or $v = u'_i$, or that v is a vertex pending from v'_i or u'_i (for some $1 \leq i \leq n$). Note also that for each $1 \leq i \leq n$, we have that $d_G(v'_i) = d_G(u'_i) = i \cdot n$.

Now let D be a dominating set of H , with $|D| = m \leq k$, and let D' be the subset of V' that corresponds to the vertices of D . That is, $D' = \{v'_i \in V' : v_i \in D\}$. We claim that the graph $G' = G[V' \setminus D']$ is locally irregular. Indeed, for every $1 \leq i \leq n$, let $\alpha(i)$ be the number of neighbours of v_i that belong to D . Observe that since D is a dominating set of H , we have that $1 \leq \alpha(i) \leq n-1$. Now, for every vertex v'_i in V' , we have that either $v'_i \in D'$, in which case v'_i does not belong to G' , or $d_{G'}(v'_i) = d_G(v'_i) - \alpha(i) < d_{G'}(u'_i)$. Moreover, for every $1 \leq i < j \leq n$, if $v'_i, v'_j \notin D'$, we have that $d_G(v'_j) - d_G(v'_i) \geq n$, and thus $d_{G'}(v'_j) - d_{G'}(v'_i) = d_G(v'_j) - \alpha(j) - d_G(v'_i) + \alpha(i) \geq n + \alpha(i) - \alpha(j) \geq 2$. Finally, every pending vertex l of G' is attached to either u'_i or v'_i , which have degree (in G') strictly larger than 1. It follows that D' is an irregularator of G with $|D'| = m \leq k$, and thus $I(G) \leq k$.

For the other direction, assume that $I(G) \leq k$ and let S be an $ir(G)$, with $|S| = k$, and $G' = G[V' \setminus S]$. For each $1 \leq i \leq n$, let $S_i = N[v'_i] \cup N(u'_i)$. We claim that for every i , we have $S \cap S_i \neq \emptyset$. Assume that this is not true, *i.e.*, that there exists an i_0 such that $S_{i_0} \cap S = \emptyset$. Then, by deleting the vertices of S from G , the degrees of v'_{i_0} and u'_{i_0} remain unchanged. Formally, we have that $d_{G'}(v'_{i_0}) = d_G(v'_{i_0}) = d_G(u'_{i_0}) = d_{G'}(u'_{i_0})$. This is a contradiction since S is an irregularator of G . Now, we consider the set S' , defined as follows:

- Start with $S' = S$.
- For each i , while there exists a vertex $v \in S_i \cap S'$ such that $d_G(v) = 1$ or $v = u'_i$, remove v from S' and add v'_i to S' .

Clearly, we have that S' only contains vertices from $V(G|_H)$ and that $|S'| \leq |S| = k$. Also, from the construction of S' , for every i , we have that $S_i \cap S' \neq \emptyset$. It follows that for every vertex v'_i , we either have $v'_i \in S'$ or there exists a vertex $v \in N(v'_i) \cap V(G|_H)$ such that $v \in S'$. Going back to H , let $D = \{v_i : v'_i \in S'\}$. It is clear that D is a dominating set of H of order at most k . This finishes our reduction.

Finally, note that throughout the above described reduction, the value of the parameter of the two problems is the same (in both of them, the parameter has the value k). Moreover, the construction of the graph G is achieved in polynomial time in regards to n . These observations conclude our proof. \square

Corollary 4.5. *Let G be a graph of order n and $k \in \mathbb{N}$. Unless the ETH fails, there is no algorithm that decides if $I(G) \leq k$ in time $\mathcal{O}^*(f(k)n^{o(k)})$.*

Proof. The corollary is a result of the following observations:

1. the reduction we presented from DOMINATING SET in the proof of Theorem 4.4 is a linear fpt-reduction and
2. there is no algorithm that answers if a graph G of order n has a Dominating Set of size at most k in time $\mathcal{O}^*(f(k)n^{o(k)})$ unless the ETH fails [27].

\square

Theorem 4.6. *Let G be a graph with treewidth tw , and $k \in \mathbb{N}$. Deciding if $I(G) = k$ is $W[1]$ -hard when parameterised by tw .*

Proof. We will present a reduction from the LIST COLOURING problem: the input consists in a graph $H = (V, E)$ and a list function $L : V \rightarrow \mathcal{P}(\{1, \dots, k\})$ that specifies the available colours for each vertex $u \in V$. The goal is to find a proper colouring $c : V \rightarrow \{1, \dots, k\}$ such that $c(u) \in L(u)$ for all $u \in V$. This problem is known to be $W[1]$ -hard when parameterised by the treewidth tw of H [20].

Now, starting from an instance (H, L) of LIST COLOURING, we will construct a graph $G = (V', E')$ (see Figure 3 (a)) such that:

- $|V'| = \mathcal{O}(|V|^6)$,
- $tw(G) = tw(H)$ and

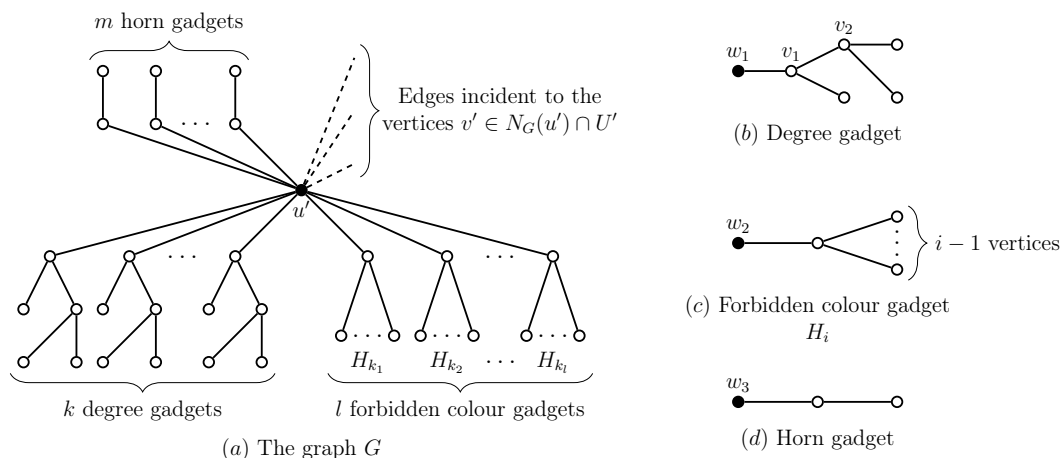


Figure 3: In (a) we illustrate the construction of G , as it is described in the proof of Theorem 4.6. The black vertex represents every vertex that belongs in U' . For the specific vertex u' shown in the figure, we have that $\bar{L}(u) = \{c_1, \dots, c_l\}$ and $k_i = n^3 - c_i$ for all $i = 1, \dots, l$. We also have that $m = 2n^3 - d_G(u) - k - l$.

- $I(G) = nk$ if and only if (H, L) is a yes-instance of LIST COLOURING.

Before we start with the construction of G , let us give the following observation.

Observation 4.7. *Let (H, L) an instance of LIST COLOURING where $H = (V, E)$ and there exists a vertex $u \in V$ such that $|L(u)| > d(u)$. Then the instance $(H[V \setminus \{u\}], L')$, where $L'(v) = L(v)$ for all $v \in V \setminus \{u\}$, is a yes-instance of LIST COLOURING if and only if (H, L) is a yes-instance of LIST COLOURING.*

Indeed, observe that for any vertex $u \in V$, by any proper colouring c of H , $c(u)$ only has to avoid $d(u)$ colours. Since $|L(u)| > d(u)$, we will always have a spare colour to use on u that belongs in $L(u)$. From the previous observation, we can assume that in our instance, for all $u \in V$, we have $|L(u)| \leq d(u)$. Furthermore, we can deduce that $k \leq n(n-1)$ as the degree of any vertex is at most $(n-1)$. Finally, let us denote by $\bar{L}(u)$ the set $\{0, 1, \dots, k\} \setminus L(u)$. It is important to note here that for every $u \in V$, the list $L(u)$ contains at least one element belonging in $\{1, \dots, k\}$. It follows that $\bar{L}(u)$ also contains at least one element, the colour 0. To sum up, we have that $1 \leq |\bar{L}(u)| \leq k$.

Now, we present the three gadgets we are going to use in the construction of G . First, we have the “forbidden colour gadget” H_i , which is a star with i leaves (see Figure 3(c)). When we say that we attach a copy of H_i on a vertex v of a graph G , we mean that we add H_i to G and we identify the vertices v and w_2 (where here and in what follows, we are using the naming illustrated in Figure 3 when talking about the vertices w_1, w_2, w_3, v_1 and v_2). The second, will be the “degree gadget”, which is presented in Figure 3(b). Finally, we have the “horn gadget”, which is a path on three vertices (see Figure 3(d)). We define the operation of attaching these two gadgets on a vertex v of a graph G similarly to how we defined this operation for the forbidden colour gadget (each time using the appropriate w_1 or w_3 , according to if it is a degree or a horn gadget respectively).

In order to construct G , we start from a copy of H . Let us use $G|_H$ to denote the copy of H that lies inside of G and, for each vertex $u \in V$, let u' be its copy in V' . Furthermore, we will call the set of these vertices U' . That is, $U' = \{v \in V(G|_H)\}$. Then, we are going to attach several copies of each gadget to u' , for each vertex $u' \in U'$. We start by attaching k copies of the degree gadget to each vertex $u' \in U'$. Then, for each $u \in V$ and each $i \in \bar{L}(u)$, we attach one copy of the forbidden colour gadget H_{2n^3-i} to the vertex u' . Finally, for each $u' \in U'$, we attach to u' as many copies of the horn gadget as are needed, in order to have $d_G(u') = 2n^3$.

Before we continue, observe that, for sufficiently large n , we have attached more than n^3 horn gadgets to each vertex of U' . Indeed, before attaching the horn gadgets, each vertex $u' \in U'$ has $d_G(u) \leq n-1$ neighbours in U' , k neighbours from the degree gadgets and at most $k < n^2$ neighbours from the forbidden colour gadgets (recall that $|\bar{L}(u)| \leq k$). We will now show that $|V'| = \mathcal{O}(n^6)$. For that purpose, let us calculate the number of vertices in all the gadgets attached

to a single vertex $u' \in U$. First, we have $5k < 5n^2$ vertices in the degree gadgets. Then, we have less than $4n^3$ vertices in the horn gadgets (as we have less than $2n^3$ such gadgets). Finally, we have at most $k < n^2$ forbidden colour gadgets, each one of which containing at most $2n^3$ vertices. So, for each vertex $u' \in U'$, we have at most $2n^5 + 4n^3 + 5n^2$ vertices in the gadgets attached to u' . Therefore, we have $|V'| = \mathcal{O}(n^6)$.

Before we prove that $I(G) \leq nk$ if and only if (H, L) is a yes-instance of LIST COLOURING, we need to argue about two things. First, about the treewidth of the graph G and second, about the minimum value of $I(G)$. Since our construction only attaches trees to each vertex of $G|_H$ (and recall that a tree has a treewidth of 1 by definition), we know that $tw(G) = tw(G|_H) = tw(H)$. As for $I(G)$, we will show that it has to be at least equal to nk . For that purpose we have the following two claims.

Claim 4.8. *Let S be an $ir(G)$ and $S \cap U' \neq \emptyset$. Then $|S| > n^3$.*

Proof of the claim. Let $u' \in S \cap U'$. By construction, G contains more than n^3 horn gadgets that are attached to u' . Therefore, by deleting u' , we create more than n^3 copies of the P_2 graph, each one of which forces us to include at least one of its vertices in S . Hence, $|S| > n^3$. \diamond

Claim 4.9. *Let S be an $ir(G)$ and $S \cap U = \emptyset$. Then $|S| \geq nk$. In particular, S includes at least one vertex from each copy of the degree gadget used in the construction of G .*

Proof of the claim. Let D be a copy of the degree gadget, attached to some vertex $u' \in U'$. Observe that we have $d_G(v_1) = d_G(v_2)$. It follows by Observation 2.5, that S contains at least one vertex v in $N(v_1, v_2)$, and since $u' \notin S$, this v is a vertex other than w_1 . The result follows from the fact that the same arguments hold for any degree gadget attached to any vertex of U' (recall that $|U'| = n$ and we have attached k copies of the degree gadget to each one of the vertices of U'). \diamond

By the previous two claims, we conclude that $I(G) \geq nk$.

We are ready to show that, if (H, L) is a yes-instance of LIST COLOURING, then there exists a set $S \subseteq V'$ such that S is an $ir(G)$ and $|S| = nk$. Let c be a proper colouring of H such that $c(u) \in L(u)$ for all $u \in V$. We will construct an irregularator of G as follows. For each $u \in V$ we partition (arbitrarily) the k degree gadgets attached to the vertex u' to $c(u)$ “good” and $(k - c(u))$ “bad” degree gadgets. For each good degree gadget, we add the copy of the vertex v_1 of that gadget to S and for each bad degree gadget we add the copy of the vertex v_2 of that gadget to S . This process creates a set S of size nk , as it includes k distinguished vertices for each vertex $u' \in U$.

Now we need to show that S is an $ir(G)$. Let $G' = G[V' \setminus S]$; observe that each vertex $u' \in U'$ has degree $d_{G'}(u') = 2n^3 - c(u)$. Therefore, u' does not have the same degree as any of its neighbours that do not belong in U' . Indeed, for every $v \in N_{G'}(u') \setminus U'$, we have that $d_{G'}(v) \in \{1, 2\}$ (if v belongs to a bad degree or a horn gadget) or $d_{G'}(v) \in \{2n^3 - i : i \in \bar{L}(u)\}$ (if v belongs to a forbidden colour gadget). Furthermore, since c is a proper colouring of H , for all $uv \in E$, we have that $c(u) \neq c(v)$. This gives us that for any edge $u'v' \in E'$ with $u', v' \in U'$, we have that $d_{G'}(u') = 2n^3 - c(u) \neq 2n^3 - c(v) = d_{G'}(v')$.

So, we know that for every vertex $u' \in U$, there is no vertex $w \in N_{G'}(u')$ such that $d_{G'}(u') = d_{G'}(w)$. It remains to show that, in G' , there exist no two vertices belonging to the same gadget, which have the same degrees. First of all, we have that S does not contain any vertex from any of the horn and forbidden colour gadgets, nor from U' . Thus any adjacent vertices belonging to these gadgets have different degrees. Last, it remains to check the vertices of the degree gadgets. Observe that for any copy of the degree gadget, S contains either v_1 or v_2 . In both cases, after the deletion of the vertices of S , any adjacent vertices belonging to any degree gadget have different degrees. Therefore, S is an $ir(G)$ of order nk and since $I(G) \geq nk$ we have that $I(G) = nk$.

Now, for the opposite direction, assume that there exists a set $S \subseteq V'$ such that S is an $ir^*(G)$ and $|S| = kn$. Let $G' = (V'', E'')$ be the graph $G[V' \setminus S]$. It follows from Claim 4.8 and Claim 4.9, that $S \cap U = \emptyset$ and that S contains exactly one vertex from each copy of the degree gadget in G and no other vertices. Consider now the colouring c of H defined as $c(u) = 2n^3 - d_{G'}(u')$. We will show that c is a proper colouring for H and that $c(u) \in L(u)$. First, we have that c is a proper colouring of H . Indeed, for any edge $uv \in E$, there exists an edge $u'v' \in E''$ (since $S \cap U' = \emptyset$). Since G' is locally irregular we have that $d_{G'}(u') \neq d_{G'}(v')$, and thus $c(u) \neq c(v)$. It remains to show that $c(u) \in L(u)$ for all $u \in V$. First observe that, during the construction of G , we attached

exactly k degree gadgets to each $u' \in U'$. It follows that $d_{G'}(u') = 2n^3 - j$ and $c(u) = j$ for a $j \in \{0, 1, \dots, k\}$. It is sufficient to show that $j \notin \bar{L}(u)$. Since S contains only vertices from the copies of the degree gadgets, we have that each $u' \in U'$ has exactly one neighbour of degree $2n^3 - i$ for each $i \in \bar{L}(u)$ (this neighbour is a vertex of the H_i forbidden colour gadget that was attached to u'). Furthermore, for all $u' \in U'$, since G' is locally irregular, we have that, $d_{G'}(u') \neq 2n^3 - i$ for all $i \in \bar{L}(u)$. Equivalently, $d_{G'}(u') = 2n^3 - j$ for any $j \in L(u)$. Thus, $c(u) \in L(u)$ for all $u \in V$. \square

Corollary 4.10. *Let G be a graph of order n and treewidth tw . Unless the ETH fails, there is no algorithm that computes $I(G)$ in time $\mathcal{O}^*(f(tw)n^{o(tw)})$.*

Proof. The corollary is a result of the following observations:

1. the reduction from LIST COLOURING that we presented in the proof of Theorem 4.6 is a linear fpt-reduction and
2. there is no algorithm that answers if an instance (G, L) of the LIST COLOURING is a yes-instance in time $\mathcal{O}^*(f(tw)n^{o(tw)})$ unless the ETH fails [20].

\square

5 Conclusion

In this work we introduce the problem of identifying the largest locally irregular induced subgraph of a given graph. There are many interesting directions that could be followed for further research. An obvious one is to investigate whether the problem of calculating $I(G)$ remains \mathcal{NP} -hard for other, more restricted families of graphs. The first candidate for such a family would be the one of cubic bipartite graphs. Such a result would also serve as a good motivation to conceive an approximation algorithm for these graphs, which we believe to be quite possible. On the other hand, there are some interesting families, for which the problem of computing an optimal irregularator could very well be decided in polynomial time, such as chordal and series-parallel graphs. The last aspect we find intriguing, is to study the parameterised complexity of calculating $I(G)$ when considering other parameters, like the size of the minimum vertex cover of G , with the goal of identifying a parameter that suffices, by itself, in order to have an FPT algorithm.

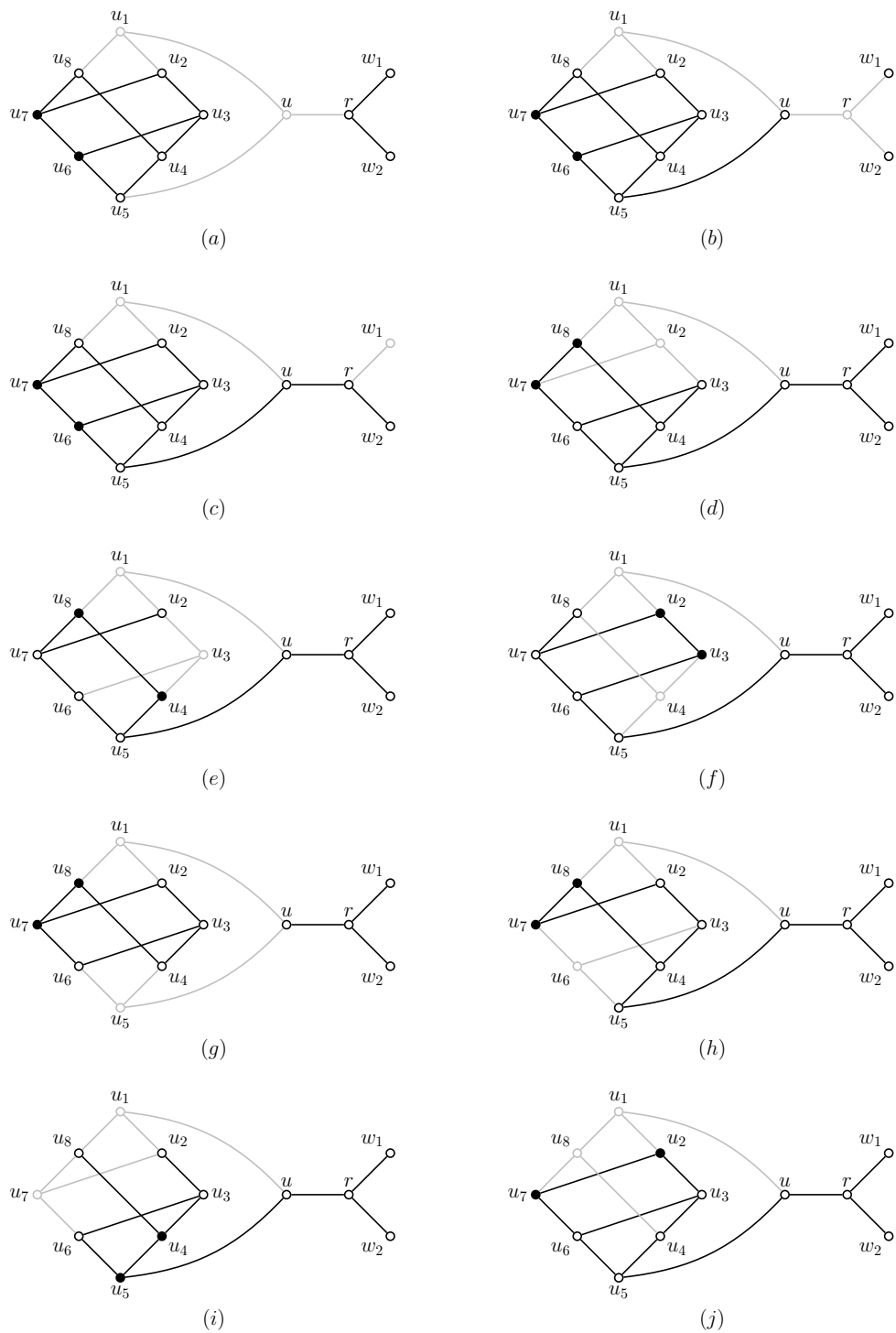


Figure 4: An illustration of all the possible $S \subseteq V(H)$ of order 2 that contain u_1 , used in the proof of Claim 3.6. The case $S = \{u_1, w_2\}$ is omitted because it is the same as the case(c). Gray vertices and edges represent the vertices and edges that do not appear in $H[V(H) \setminus S]$. By black vertices, we represent a pair of adjacent vertices with the same degree in $H[V(H) \setminus S]$.

References

- [1] A. Agra, G. Dahl, T. A. Haufmann, and S. J. Pinheiro. The k -regular induced subgraph problem. *Discrete Applied Mathematics*, 222:14–30, 2017.
- [2] Y. Alavi, A. Boals, G. Chartrand, O. Oellermann, and P. Erdős. K -path irregular graphs. *Congressus Numerantium*, 65, 01 1988.
- [3] Y. Alavi, G. Chartrand, F. R. K. Chung, P. Erdős, R. L. Graham, and O. R. Oellermann. Highly irregular graphs. *Journal of Graph Theory*, 11(2):235–249, 1987.
- [4] A. Ali, G. Chartrand, and P. Zhang. *Irregularity in Graphs*. Springer briefs in mathematics. Springer, 2021.
- [5] Y. Asahiro, H. Eto, T. Ito, and E. Miyano. Complexity of finding maximum regular induced subgraphs with prescribed degree. *Theoretical Computer Science*, 550:21–35, 2014.
- [6] O. Baudon, J. Bensmail, J. Przybyło, and M. Wozniak. On decomposing regular graphs into locally irregular subgraphs. *European Journal of Combinatorics*, 49:90–104, 2015.
- [7] R. Belmonte and I. Sau. On the complexity of finding large odd induced subgraphs and odd colorings. *Algorithmica*, 83(8):2351–2373, 2021.
- [8] J. Bensmail, M. Merker, and C. Thomassen. Decomposing graphs into a constant number of locally irregular subgraphs. *European Journal of Combinatorics*, 60:124–134, 2017.
- [9] P. Berman, M. Karpinski, and A. D. Scott. Approximation hardness of short symmetric instances of MAX-3SAT. *Electronic Colloquium on Computational Complexity*, (049), 2003.
- [10] H. L. Bodlaender. A partial k -arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1):1–45, 1998.
- [11] L. Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58(4):171–176, 1996.
- [12] Y. Caro and R. Yuster. Large induced subgraphs with equated maximum degree. *Discrete Mathematics*, 310(4):742–747, 2010.
- [13] G. Chartrand, P. Erdős, and O. Oellermann. How to define an irregular graph. *The College Mathematics Journal*, 19, 01 1988.
- [14] G. Chartrand, M. Jacobon, J. Lehel, O. Oellermann, S. Ruiz, and F. Saba. Irregular networks. *Congressus Numerantium*, 64, 01 1986.
- [15] J. Chen, B. Chor, M. Fellows, X. Huang, D. W. Juedes, I. A. Kanj, and G. Xia. Tight lower bounds for certain parameterized np-hard problems. *Information and Computation*, 201(2):216–231, 2005.
- [16] J. Chen, X. Huang, I. A. Kanj, and G. Xia. On the computational hardness based on linear fpt-reductions. *Journal of Combinatorial Optimization*, 11(2):231–247, 2006.
- [17] J. Chen, X. Huang, I. A. Kanj, and G. Xia. Strong computational lower bounds via parameterized complexity. *Journal of Computer and System Sciences*, 72(8):1346–1367, 2006.
- [18] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.
- [19] R. Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.
- [20] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.
- [21] A. M. Frieze, R. J. Gould, M. Karonski, and F. Pfender. On graph irregularity strength. *Journal of Graph Theory*, 41(2):120–137, 2002.
- [22] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- [23] M. Karoński, T. Łuczak, and A. Thomason. Edge weights and vertex colors. *Journal of Combinatorial Theory*, 91:151–157, 05 2004.

- [24] S. Khot and V. Raman. Parameterized complexity of finding subgraphs with hereditary properties. *Theoretical Computer Science*, 289(2):997–1008, 2002.
- [25] J. M. Lewis and M. Yannakakis. The node-deletion problem for hereditary properties is np-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980.
- [26] C. N. Lintzmayer, G. O. Mota, and M. Sambinelli. Decomposing split graphs into locally irregular graphs. *Discrete Applied Mathematics*, 292:33–44, 2021.
- [27] D. Lokshtanov, D. Marx, and S. Saurabh. Lower bounds based on the exponential time hypothesis. *Bulletin of the European Association for Theoretical Computer Science*, 105:41–72, 2011.
- [28] H. Moser and D. M. Thilikos. Parameterized complexity of finding regular induced subgraphs. *Journal of Discrete Algorithms*, 7(2):181–190, 2009.
- [29] J. Przybyło. On decomposing graphs of large minimum degree into locally irregular subgraphs. *Electronic Journal of Combinatorics*, 23(2):2–31, 2016.
- [30] J. Przybyło. Irregularity strength of regular graphs. *Electronic Journal of Combinatorics*, 15, 06 2008.