Graph Isomorphism Parameterized by Elimination Distance to Bounded Degree*

Jannis Bulian and Anuj Dawar

University of Cambridge Computer Laboratory

June 19, 2014

A commonly studied means of parameterizing graph problems is the deletion distance from triviality [10], which measures the number of vertices that need to be deleted from a graph to place it in some class for which efficient algorithms are known. In the context of graph isomorphism, we define triviality to mean a graph with maximum degree bounded by a constant, as such graph classes admit polynomial-time isomorphism tests. We generalise deletion distance to a measure we call elimination distance to triviality, based on elimination trees or tree-depth decompositions. We establish that graph isomorphism is \( \mathsf{FPT} \) when parameterized by elimination distance to bounded degree, generalising results of Bouland et al. [2] on isomorphism parameterized by tree-depth.

1 Introduction

The graph isomorphism problem (\( \mathsf{GI} \)) is the problem of determining, given a pair of graphs \( G \) and \( H \), whether they are isomorphic. This problem has an unusual status in complexity theory as it is neither known to be in \( \mathsf{P} \) nor known to be \( \mathsf{NP} \)-complete, one of the few natural problems for which this is the case. Polynomial-time algorithms are known for a variety of special classes of graphs. Many of these lead to natural parameterizations of \( \mathsf{GI} \) by means of structural parameters of the graphs. For instance, it is known that \( \mathsf{GI} \) is in \( \mathsf{XP} \) parameterized by the genus of the graph, [15, 7], by maximum

*Research supported in part by EPSRC grant EP/H026835.
degree [14] and by the size of the smallest excluded minor [17], or more generally, the smallest excluded topological minor [9]. For each of these parameters, it remains an open question whether the problem is FPT. On the other hand, GI has been shown to be FPT when parameterized by eigenvalue multiplicity [5], tree distance width [20], the maximum size of a simplicial component [18, 19] and minimum feedback vertex set [11]. Bouland et al. [2] showed that the problem is FPT when parameterized by the tree depth of a graph and in a recent advance on this, Lokshtanov et al. [13] have announced that it is also FPT parameterized by tree width.

Our main result extends the results of Bouland et al. and is incomparable with that of Lokshtanov et al. We show that graph isomorphism is FPT parameterized by elimination distance to degree d, for any constant d. The structural graph parameter we introduce is an instance of what Guo et al. [10] call distance to triviality and is worthy of further investigation in the context of other graph problems.

To put this parameter in context, consider the simplest notion of distance to triviality for a graph G: the number k of vertices of G that must be deleted to obtain a graph with no edges. This is, of course, just the size of the minimal vertex cover in G and is a parameter that has been much studied (see for instance [6]). Indeed, it is also quite straightforward to see that GI is FPT when parameterized by vertex cover number. Consider two ways this observation might be strengthened. The first is to relax the notion of what we consider to be “trivial”. For instance, as there is for each d a polynomial time algorithm deciding graph isomorphism among graphs with maximum degree d, we may take this as our trivial base case. We then parameterize a graph G by the number k of vertices that must be deleted to obtain a subgraph of G with maximum degree d. This yields the parameter deletion distance to bounded degree, which we consider in Section 4 below. Alternatively, we relax the notion of “distance” so that rather than considering the sequential deletion of k vertices, we consider the recursive deletion of vertices in a tree-like fashion. To be precise, say that a graph G has elimination distance k + 1 from triviality if, in each connected component of G we can delete a vertex so that the resulting graph has distance k to triviality. If triviality is understood to mean the empty graph, this just yields a definition of the tree depth of G. In our main result, we combine these two approaches by parameterizing G by the elimination distance to triviality, where a graph is trivial if it has maximum degree d. We show that, for any fixed d, this gives a structural parameter on graphs for which graph isomorphism is FPT. Along the way, we establish a number of characterisations of the parameter that may be interesting in themselves. The key idea in the proof is the separation of any graph of elimination distance k to degree d into two subgraphs, one of which has degree bounded by d and the other tree-depth bounded by a function of k and d, in a canonical way.

In Section 2 we recall some definitions from graph theory and parameterized complexity theory. We discuss known algorithms for graph isomorphism and canonisation on bounded degree graphs in Section 3 and also state the generalised versions for coloured graphs. Section 4 introduces the notion of deletion distance to bounded degree and presents a kernelisation procedure that allows us to decide isomorphism. In Section 5 we introduce the main parameter of our paper, elimination distance to bounded degree, and establish its key properties. The main result on FPT graph isomorphism is
Parameterized complexity theory is a two-dimensional approach to the study of the complexity of computational problems. A language (or problem) \( L \) is a set of strings \( L \subseteq \Sigma^* \) over a finite alphabet \( \Sigma \). A parameterization is a function \( \kappa : \Sigma^* \rightarrow \mathbb{N} \). We say that \( L \) is fixed-parameter tractable with respect to \( \kappa \) if we can decide whether an input \( x \) is in \( L \) in time \( O(f(\kappa(x)) \cdot |x|^c) \), where \( c \) is a constant and \( f \) is a computable function. For a thorough discussion of the subject we refer to the books by Downey and Fellows \([4]\), Flum and Grohe \([8]\) and Niedermeier \([16]\).

A graph \( G \) is a set of vertices \( V(G) \) and a set of edges \( E(G) \subseteq V(G) \times V(G) \). We will usually assume that graphs are loop-free and undirected, i.e. that \( E \) is irreflexive and symmetric. If \( E \) is not symmetric, we call \( G \) a directed graph. We mostly follow the notation in Diestel’s book \([3]\). If \( v \in G \) and \( S \subseteq V(G) \), we write \( E_G(v, S) \) for the set of edges \( \{vw \mid w \in S\} \) between \( v \) and \( S \).

The neighbourhood of a vertex \( v \) is \( N_G(v) := \{w \in V(G) \mid vw \in E(G)\} \). The degree of a vertex \( v \) is the size of its neighbourhood \( \deg_G(v) := |N_G(v)| \). For a set of vertices \( S \subseteq V(G) \) its neighbourhood is defined to be \( N_G(S) := \bigcup_{v \in S} N_G(v) \). The degree of a graph \( G \) is the maximal degree of its vertices \( \Delta(G) := \max\{\deg_G(v) \mid v \in V(G)\} \). If it is clear from the context what the graph is, we will sometimes omit the subscript.

A subgraph \( H \) of \( G \) is a graph with vertices \( V(H) \subseteq V(G) \) and edges \( E(H) \subseteq (V(H) \times V(H)) \cap E(G) \). If \( A \subseteq V(G) \) is a set of vertices of \( G \), we write \( G[A] \) for the subgraph induced by \( A \), i.e. \( V(G[A]) = A \) and \( E(G[A]) = E(G \cap (A \times A)) \). If \( A \) is a subset of \( V(G) \), we write \( G \setminus A \) for \( G[V(G) \setminus A] \). For a vertex \( v \in V(G) \), we write \( G \setminus v \) for \( G \setminus \{v\} \).

A vertex \( v \) is said to be reachable from a vertex \( w \) in \( G \) if \( v = w \) or if there is a sequence of edges \( a_1a_2, \ldots, a_{s-1}a_s \in E(V) \) with the \( a_i \) pairwise distinct and \( w = a_1 \) and \( v = a_s \). We call the subgraph \( P \) of \( G \) with vertices \( V(P) = \{a_1, \ldots, a_s\} \) and edges \( E(P) = \{a_1a_2, \ldots, a_{s-1}a_s\} \) a path from \( w \) to \( v \). Let \( H \) be a subgraph of \( G \) and \( v, w \in V(G) \). A path through \( H \) from \( w \) to \( v \) is a path \( P \) from \( w \) to \( v \) in \( G \) with all vertices, except possibly the endpoints, in \( V(H) \), i.e. \( (V(P) \setminus \{v, w\}) \subseteq V(H) \).

It is easy to see that for undirected graphs reachability defines an equivalence relation on the vertices of \( G \). A subgraph of an undirected graph induced by a reachability class is called a component.

Two graphs \( G, G' \) are isomorphic if there is a bijection \( \varphi : V(G) \rightarrow V(G') \) such that for all \( v, w \in V(G) \) we have that \( vw \in E(G) \) if and only if \( \varphi(v)\varphi(w) \in V(G') \). We write \( G \cong G' \) if \( G \) and \( G' \) are isomorphic. We write \( GI \) to denote the problem of deciding, given \( G \) and \( G' \) whether \( G \cong G' \).

A \((k-)\)colouring of a graph \( G \) is a map \( c : V(G) \rightarrow \{1, \ldots, k\} \) for some \( k \in \mathbb{N} \). We call a graph together with a colouring a coloured graph. Two coloured graphs \( G, G' \) with respective colourings \( c, c' : V(G) \rightarrow \{1, \ldots, k\} \) are isomorphic if there is a bijection...
\( \varphi : V(G) \to V(G') \) such that:

- for all \( v, w \in V(G) \) we have that \( vw \in E(G) \) if and only if \( \varphi(v) \varphi(w) \in V(G') \);
- for all \( v \in V(G) \), we have that \( c(v) = c(\varphi(v)) \).

Note that we require the colour classes to match exactly, and do not allow a permutation of the colour classes.

Let \( C \) be a class of (coloured) graphs closed under isomorphism. A canonical form for \( C \) is a function \( F : C \to C \) such that

- for all \( G \in C \), we have that \( F(G) \cong G \);
- for all \( G, H \in C \), we have that \( G \cong H \) if, and only if, \( F(G) = F(H) \).

A partial order is a binary relation \( \leq \) on a set \( S \) which is reflexive, antisymmetric and transitive. If \( \leq \) is a partial order on \( S \), and for each element \( a \in S \), the set \( \{ b \in S \mid b \leq a \} \) is totally ordered by \( \leq \), we say \( \leq \) is a tree order.

**Definition 2.1.** An elimination order \( \leq \) is a tree order on the vertices of a graph \( G \), such that for each edge \( uv \in E(G) \) we have either \( u \leq v \) or \( v \leq u \).

We say that an order has height \( k \) if the length of the longest chain it contains is \( k \).

Note that there is an elimination order \( \leq \) of height \( k \) for a graph \( G \) if, and only if, \( G \) has tree-depth \( \text{td}(G) \leq k \), which is defined as follows

\[
\text{td}(G) := \begin{cases} 
0, & \text{if } V(G) = \emptyset; \\
1 + \min\{ \text{td}(G \setminus v) \mid v \in V(G) \}, & \text{if } G \text{ is connected}; \\
\max\{ \text{td}(H) \mid H \text{ a component of } G \}, & \text{otherwise}. 
\end{cases}
\]

3 Isomorphism on (coloured) bounded degree graphs

In this section we collect some well known results about isomorphism tests and canonicalisation of bounded degree graphs that will be used in the later sections. Luks [14] showed that isomorphism of bounded-degree graphs is decidable in polynomial time. This result extends, by an easy reduction, to coloured graphs of bounded-degree. For completeness, we present this reduction explicitly.

**Proposition 3.1.** The isomorphism problem for coloured graphs can be reduced to \( GI \) in polynomial time.

**Proof.** Let \( G, G' \) be graphs and let \( c, c' : V(G) \to \{1, \ldots, k\} \) be colourings of \( G, G' \) respectively for some \( k \in \mathbb{N} \).

We define \( H \) to be the graph whose vertices include \( V(G) \) and, additionally, for each \( v \in V(G) \), \( c(v) + 1 \) new vertices \( u_1^v, \ldots, u_{c(v)+1}^v \). The edges of \( H \) are the edges \( E(G) \) plus additional edges so that the vertices \( v \) and \( u_1^v, \ldots, u_{c(v)+1}^v \) form a simple cycle of length \( c(v) + 2 \). We obtain \( H' \) in a similar way from \( G' \).
We claim that $G \cong G'$ if, and only if, $H \cong H'$. Clearly, if $G \cong G'$ and $\varphi$ is an isomorphism witnessing this, it can be extended to an isomorphism from $H$ to $H'$ by mapping $u_i^v$ to $u_i^\varphi(v)$. For the converse, suppose $H \cong H'$ and let $\varphi : H \to H'$ be an isomorphism. We use it to define an isomorphism $\varphi'$ from $G$ to $G'$. Note that, if $v \in V(G)$ is not an isolated vertex of $G$, then it has degree at least 3 in $H$. Since $\varphi(v)$ has the same degree, it is in $V(G')$, and we let $\varphi'(v) = \varphi(v)$. If $v$ is an isolated vertex of $G$, then its component in $H$ is a simple cycle of length $c(v) + 2$. The image of this component under $\varphi$ is a simple cycle of $H'$ which must contain exactly one vertex $v'$ of $V(G')$. We let $\varphi'(v) = v'$. It is easy to see that there is an edge between $v_1, v_2$ in $G$ if, and only if, there is an edge between $\varphi'(v_1)$ and $\varphi'(v_2)$ in $G'$. To see that $\varphi'$ also preserves colours, note that $\varphi$ must map the cycle containing $u_i^v$ to the cycle containing $u_i^\varphi(v)$ and therefore $c(v) = c(\varphi'(v))$.

Remark. Note that the construction in the proof increases the degree of each vertex by 2, so if $G$ and $G'$ are graphs of degree $d$, then $H, H'$ are graphs of degree $d + 2$.

As Luks [14] proved that isomorphism of bounded degree graphs can be decided in polynomial time, we have the following:

**Theorem 3.2.** We can test in polynomial time whether two (coloured) graphs with maximal degree bounded by a constant are isomorphic.

Babai and Luks [1] give a polynomial time canonisation algorithm for bounded degree graphs. Just as above we can reduce canonisation of coloured bounded degree graphs to the bounded degree graph canonisation problem.

**Theorem 3.3.** Let $C$ be a class of (coloured) bounded degree graphs closed under isomorphism. Then there is a canonical form $F$ for $C$ that allows us to compute $F(G)$ in polynomial time.

### 4 Deletion distance to bounded degree

We first study the notion of deletion distance to bounded degree. Though the result in this section is subsumed by the more general one in the following sections, it provides a useful warm-up. The notion of deletion distance to bounded degree is a particular instance of the general notion of distance to triviality introduced by Guo et al. [10]. In the context of graph isomorphism, we have chosen triviality to mean graphs of bounded degree.

**Definition 4.1.** A graph $G$ has deletion distance $k$ to degree $d$ if there are $k$ vertices $v_1, \ldots, v_k \in V(G)$ such that $G \setminus \{v_1, \ldots, v_k\}$ has degree $d$. We call such a set $\{v_1, \ldots, v_k\}$ a $d$-deletion set.

Remark. To say that $G$ has deletion distance 0 from degree $d$ is just to say that $G$ has maximum degree $d$. Also note that if $d = 0$, then the $d$-deletion set is just a vertex cover and the deletion distance the vertex cover number of $G$. 


We show that isomorphism is fixed-parameter tractable on such graphs parameterized by \( k \) with fixed degree \( d \); in particular we give a procedure that computes a polynomial kernel in linear time.

**Theorem 4.2.** For any graph \( G \) and integers \( d, k > 0 \), we can identify in linear time a subgraph \( G' \) of \( G \), a set of vertices \( U \subseteq V(G') \) with \( |U| = O(k(k+d)^2) \) and a \( k' \leq k \) such that: \( G \) has deletion distance \( k \) to degree \( d \) if and only if \( G' \) has deletion distance \( k' \) to degree \( d \) and, moreover, if \( G' \) has deletion distance at most \( k' \), then any minimal \( d \)-deletion set for \( G' \) is contained in \( U \).

**Proof.** Let \( H := \{v \in V(G) \mid \deg(v) > k+d\} \) and let \( R \) be a minimal \( d \)-deletion set for \( G \). So if \( G \) has deletion distance at most \( k \) to degree \( d \), then \( |R| \leq k \) and the vertices in \( V(G \setminus R) \) have degree at most \( k+d \) in \( G \). So \( H \subseteq R \). This means that if \( |H| > k \), then \( G \) must have deletion distance greater than \( k \) to degree \( d \) and in that case we let \( G' := G, k' := k \) and \( U = \emptyset \).

Otherwise let \( G' := G \setminus H \) and \( k' := k - |H| \). We have shown that every \( d \)-deletion set of size at most \( k \) must contain \( H \). Thus \( G \) has deletion distance \( k \) to degree \( d \) if and only if \( G' \) has deletion distance \( k' \) to degree \( d \).

Let \( S := \{v \in V(G') \mid \deg_G(v) > d\} \) and \( U := S \cup N_G(S) \). Let \( R' \subseteq V(G') \) be a minimal \( d \)-deletion set for \( G' \). We show that \( R' \subseteq U \). Let \( v \notin U \). Then by the definition of \( U \) we know that \( \deg_G(v) \leq d \) and all of the neighbours of \( v \) have degree at most \( d \) in \( G' \). So if \( v \in R' \), then \( G \setminus (R' \setminus \{v\}) \) also has maximal degree \( d \), which violates the minimality of \( R' \). Thus \( v \notin R' \).

Note that the vertices in \( G' \setminus \left( R' \cup N(R') \right) \) have the same degree in \( G' \) as in \( G \) and thus all have degree at most \( d \). So \( S \subseteq R' \cup N(R') \) and thus \( |U| \leq k' + k'(k + d) + k'(k + d)^2 = O(k(k + d)^2) \).

Finally, the sets \( H \) and \( U \) defined as above can be found in linear time, and \( G', k' \) can be computed from \( H \) in linear time. \( \square \)

**Remark.** Note that if \( U = \emptyset \) and \( k' > 0 \), then there are no \( d \)-deletion sets of size at most \( k' \).

So if there is a \( d \)-deletion set of size at most \( k \), we can compute the kernel \( U \) and then find the deletion set by brute force by checking all the \( k'(k+d)^2 \) possibilities. If there is no such \( d \)-deletion set, either the computation of the kernel fails, or the brute force attempt will fail.

Next we see how the kernel \( U \) can be used to determine whether two graphs with deletion distance \( k \) to degree \( d \) are isomorphic by reducing the problem to isomorphism of coloured graphs of degree at most \( d \).

Suppose we are given two graphs \( G \) and \( H \) with \( d \)-deletion sets \( S = \{v_1, \ldots, v_k\} \) and \( T = \{w_1, \ldots, w_k\} \) respectively. Further suppose that the map \( v_i \mapsto w_i \) is an isomorphism on the induced subgraphs \( G[S] \) and \( H[T] \). We can then test if this map can be extended to an isomorphism from \( G \) to \( H \) using Theorem [4.2]. To be precise, we define the coloured graphs \( G' \) and \( H' \) which are obtained from \( G \setminus S \) and \( H \setminus T \) respectively, by colouring vertices. A vertex \( u \in V(G') \) gets the colour \( \{i \mid v_i \in N_G(u)\} \), i.e. the set of its neighbours in \( S \). Vertices in \( H' \) are similarly coloured by the sets of their neighbours in \( T \). It is
clear that $G'$ and $H'$ are isomorphic if, and only if, there is an isomorphism between $G$ and $H$, extending the fixed map between $S$ and $T$. The coloured graphs $G'$ and $H'$ have degree bounded by $d$, so Theorem 3.2 gives us a polynomial-time isomorphism test on these graphs.

Now, given a pair of graphs $G$ and $H$ which have deletion distance $k$ to degree $d$, let $A$ and $B$ be the sets of vertices of degree greater than $k + d$ in the two graphs respectively. Also, let $U$ and $V$ be the two kernels in the graphs obtained from Theorem 4.2. Thus, any $d$-deletion set in $G$ contains $A$ and is contained in $A \cup U$ and similarly, any $d$-deletion set for $H$ contains $B$ and is contained in $B \cup V$. Therefore to test $G$ and $H$ for isomorphism, it suffices to consider all $k$-element subsets $S$ of $A \cup U$ containing $A$ and all $k$-element subsets $T$ of $B \cup V$ containing $B$, and if they are $d$-deletion sets for $G$ and $H$, check for all $k!$ maps between them whether the map can be extended to an isomorphism from $G$ to $H$. As $d$ is constant this takes time

$$O^* \left( \left( \frac{k^3}{k} \right)^2 \cdot k! \right) = O^* \left( 2^{7k \log k} \right).$$

5 Elimination distance to bounded degree

In this section we introduce a new structural parameter for graphs. We generalise the idea of deletion distance to triviality by recursively allowing deletions from each component of the graph. This generalises the idea of elimination height or tree-depth, and is equivalent to it when the notion of triviality is the empty graph. In the context of graph isomorphism we again define triviality to mean bounded degree, so we will look at the elimination distance to bounded degree in this section.

**Definition 5.1.** The elimination distance to degree $d$ of a graph $G$ is defined as follows:

$$ed_d(G) := \begin{cases} 
0, & \text{if } \Delta(G) \leq d; \\
1 + \min\{ed_d(G \setminus v) \mid v \in V(G)\}, & \text{if } G \text{ is connected}; \\
\max\{ed_d(H) \mid H \text{ a connected component of } G\}, & \text{otherwise}.
\end{cases}$$

We first introduce other equivalent characterisations of this parameter. If $G$ is a graph that has elimination distance $k$ to degree $d$, then we can associate a certain tree order $\leq$ with it:

**Definition 5.2.** A tree order $\leq$ on $V(G)$ is an elimination order to degree $d$ for $G$ if for each $v \in V(G)$ the set

$$S_v := \{u \in V(G) \mid uv \in E(G) \text{ and } u \not< v \text{ and } v \not< u\}$$

satisfies either:

- $S_v = \emptyset$; or
- $v$ is $\leq$-maximal, $|S_v| \leq d$, and for all $u \in S_v$, we have $\{w \mid w < u\} = \{w \mid w < v\}$. 


Remark. Note that if \( S_v = \emptyset \) for all \( v \in V(G) \), then an elimination order to degree \( d \) is just an elimination order, in the sense of Definition 2.1.

**Proposition 5.3.** A graph \( G \) has \( ed_d(G) = k \) if and only if there is an elimination order \( \leq \) to degree \( d \) of height \( k \) for \( G \).

**Proof.** Let \( S_v \) be as in Definition 5.2. We prove the proposition by induction on \( k \). If \( k = 0 \), then the graph has no vertex of degree larger than \( d \) and we can take \( \leq \) to be the identity relation on \( V(G) \). Then every \( v \in V(G) \) is maximal, we have \( |S_v| \leq d \), and for all \( u \in S_v \) we have \( \{ w \mid w < u \} = \emptyset = \{ w \mid w < v \} \).

Suppose \( k > 0 \) and the statement is true for smaller values. If \( G \) is not connected, we apply the following argument to each component. So in the following we assume that \( G \) is connected.

Suppose \( ed_d(G) = k \). Then there is a vertex \( a \in V(G) \) such that the components \( C_1, \ldots, C_r \) of \( G \setminus a \) all have \( ed_d(C_i) \leq k - 1 \). So by the induction hypothesis each \( C_i \) has a tree order \( \leq_i \) to degree \( d \) of height at most \( k - 1 \) with the properties in Definition 5.2. For each \( v \in V(C_i) \) define

\[
S_v^i := \{ u \in V(C_i) \mid uv \in E(G) \text{ and } u \not\leq v \text{ and } v \not\leq u \}.
\]

Let

\[
\leq := \{(a, w) \mid w \in V(G)\} \cup \bigcup_i \leq_i.
\]

Then \( \leq \) is clearly a tree order for \( G \). Note that \( S_a = \emptyset \). Let \( v \in V(G) \setminus a \) be a vertex different from \( a \), say \( v \in V(C_i) \). Note \( S_v^i = S_v \). If \( S_v \neq \emptyset \), then \( v \) is \( \leq_i \)-maximal, and thus also \( \leq \)-maximal. Moreover, \( |S_v^i| = |S_v| \leq d \). Lastly for any \( u \in S_v \):

\[
\{ w \mid w < u \} = \{ a \} \cup \{ w \mid w <_i u \} = \{ a \} \cup \{ w \mid w <_i v \} = \{ w \mid w < v \}.
\]

Conversely assume there is an elimination order \( \leq \) to degree \( d \) of height \( k \) for \( G \). There is a single minimal element \( v \) of \( \leq \) because \( G \) is connected and \( k > 0 \). Note that \( \leq \) restricted to a component \( C \) of \( G \setminus v \) has height \( k - 1 \) and thus by the induction assumption we have that \( ed_d(C) \leq k - 1 \). 

We can split a graph with an elimination order to degree \( d \) in two parts: one of low degree, and one with an elimination order defined on it. So if \( G \) is a graph that has elimination distance \( k \) to degree \( d \), we can associate an elimination order \( \leq \) for a subgraph \( H \) of \( G \) of height \( k \) with \( G \), so that each component of \( G \setminus V(H) \) has degree at most \( d \) and is connected to \( H \) in a certain way:

**Proposition 5.4.** Let \( G \) be a graph and \( \leq \) an elimination order to degree \( d \) for \( G \) of height \( k \). If \( A \) is the set of vertices in \( V(G) \) that are not \( \leq \)-maximal, then:

1. \( \leq \) restricted to \( A \) is an elimination order of height \( k \) of \( G[A] \); and
2. \( G \setminus A \) has degree at most \( d \);
3. if $C$ is the vertex set of a component of $G \setminus A$, and $u, v \in A$ are $\leq$-incomparable, then either $E(u, C) = \emptyset$ or $E(v, C) = \emptyset$.

Proof. As any $v \in A$ is non-maximal, by Definition 5.2, $S_v = \emptyset$. Hence if there is an edge between $u, v \in A$, either $u < v$ or $v < u$, and (1) follows.

Since $G \setminus A$ contains the $\leq$-maximal elements, they are all incomparable. By definition of an elimination order to degree $d$, this means that each vertex in $G \setminus A$ has at most $d$ neighbours in $G \setminus A$, so this graph has degree at most $d$, establishing (2).

To show (3), let $C$ be the vertex set of a component of $G \setminus A$ and let $u, v \in A$ be incomparable. Suppose for a contradiction that $E(u, C) \neq \emptyset$ and $E(v, C) \neq \emptyset$. Then there are $a, b \in C$ such that $au, bv \in E(G)$. By Definition 5.2, $\{w \mid a < w\} = \{w \mid w < b\}$, so $a < v$ and $a < v$, so $u$ and $v$ must be comparable because $\leq$ is a tree-order – a contradiction.

We also have a converse to the above in the following sense.

Proposition 5.5. Suppose $G$ is a graph with $A \subseteq V(G)$ a set of vertices and $\leq_A$ an elimination order of $G[A]$ of height $k$, such that:

1. $G \setminus A$ has degree at most $d$;
2. if $C$ is the vertex set of a component of $G \setminus A$, and $u, v \in A$ are incomparable, then either $E(u, C) = \emptyset$ or $E(v, C) = \emptyset$.

Then, $\leq_A$ can be extended to an elimination order to degree $d$ for $G$ of height $k + 1$.

Proof. Let

$$\leq := \leq_A \cup \{(v, v) \mid v \in (V(G) \setminus A)\}$$
$$\cup \{(u, v) \mid u \in A, v \in C, C \text{ a component of } G \setminus A, E(u, C) \neq \emptyset\}.$$ 

Then $\leq$ is a tree order on $G$. Let $v \in V(G)$ and let $S_v$ be as in Definition 5.2. Suppose $S_v \neq \emptyset$. Then $v \in (V(G) \setminus A)$ and has degree at most $d$ in $G \setminus A$. By the construction $v$ is $\leq$-maximal. Let $u \in S_v$. Then there is a component $C$ of $G \setminus A$ that contains both $u$ and $v$ and thus $\{w \mid w < u\} = \{w \mid w < v\}$.

Remark. In the following, given a graph $G$ and an elimination order to degree $d$, $\leq$, we call the subgraph induced by the non-maximal elements of the order $\leq$ the non-maximal subgraph of $G$ under $\leq$.

These alternative characterisations turn out to be very useful. In the following series of lemmas, we use them to show that it is sufficient to consider elimination orders of a super-graph of the high degree part of a graph $G$ in order to obtain an elimination order to degree $d$, whose height is bounded by a function of the parameter.

Lemma 5.6. Let $G$ be a graph with maximal degree $\Delta(G) \leq k + d$. Let $\leq$ be an elimination order to degree $d$ of height $k$ of $G$ with non-maximal subgraph $H$.

Then $H' = G[V(H) \cup \{v \in V(G) \mid \deg_G(v) > d\}]$ has an elimination order $\subseteq$ of height at most $k(k + d + 1)$ that can be extended to an elimination order to degree $d$ of $G$.
Proof. Let $G, H, H'$ and $\leq$ be as in the statement of the lemma. We will adapt $\leq$ to an elimination order $\sqsubseteq$ of $H'$.

Let $B$ be the set of $\leq$-maximal elements in $V(H)$. Let $w \in H' \setminus V(H)$ and let $C_w$ be the component of $G \setminus V(H)$ that contains $w$. Note that $N(C_w) \neq \emptyset$, because $\deg(w) > d$, so at least one vertex in $H$ must be adjacent to $w$. By Definition 5.2, all vertices in $N(C_w)$ are $\leq$-comparable, so they are linearly ordered and there is a $b_w \in B$ such that $b_w \geq a$ for all $a \in N(C_w)$. This way we can associate a vertex $b_w \in B$ with every $w \in H' \setminus V(H)$. For each $b \in B$, let $W_b := \{w \in H' \setminus V(H) \mid b_w = b\}$, and let $\sqsubseteq_b$ be an arbitrary linear order on $W_b$.

For any $u, v \in V(G)$, define $u \sqsubseteq v$ if one of the following holds:

- $u = v$;
- $u, v \in H$ and $u \leq v$;
- $u \in H$, $v \in G \setminus V(H')$ and $u \leq v$;
- $u \in H$, $u \leq b$ for some $b \in B$ with $v \in W_b$;
- $u \in W_b$ for some $b \in B$, $w \in G \setminus V(H')$ and $b \leq w$;
- $u, v \in W_b$ for some $b \in B$ and $u \sqsubseteq_b v$.

It follows from the construction that $\sqsubseteq$ restricted to $H'$ is an elimination order of $H'$, and that $\sqsubseteq$ is an elimination order to degree $d$ of $G$.

The height of $\sqsubseteq$ is at most $k(k + d + 1)$, because $G$ has maximum degree $k + d$ and for each $b \in B$, the set $W_b \subseteq N(\{v \in H \mid v \leq b\})$, and $N(\{v \in H \mid v \leq b\})$ contains at most $k(k + d)$ vertices. So the length of a chain in $H'$ will be at most $k + k(k + d)$. \hfill \Box

Lemma 5.7. Let $G$ be a graph. Let $\leq$ be an elimination order to degree $d$ of $G$ of height $k$ with non-maximal subgraph $H$, such that $H$ contains all vertices of degree greater than $d$.

Then $H' = G[\{v \in V(H) \mid \deg_G(v) > d\}]$ has an elimination order $\sqsubseteq$ of height at most $k((k + 1)d)^{2^k}$ that can be extended to an elimination order to degree $d$ of $G$.

Proof. Let $G, H, H'$ and $\leq$ be as in the statement of the lemma. We assume that $G$ is connected – if not, we can apply the argument to each component of $G$. We construct an elimination order $\sqsubseteq$ of $H'$ from $\leq$, making sure that it has height at most $k((k + 1)d)^{2^k}$. This extends to an elimination order to degree $d$ of $G$ by making all vertices not in $H'$ maximal, as in the Proposition 5.5.

Let $J := H \setminus V(H')$. For $v \in V(J)$, let $K_v$ be the set of vertices $w \in H'$ such that:

- there is a path from $v$ to $w$ with no vertex except $w$ in $V(H')$; and
- $v \leq w$; and
- $v$ is the $\leq$-minimal element with those properties.
Note that because $\leq$ is a tree order and the third condition, the sets $K_v$ are pairwise disjoint. Let $\overline{K} := V(H') \setminus (\bigcup_{v \in V(J)} K_v)$ be the set of vertices in $H'$ that are not contained in any of the $K_v$.

For each $v \in V(J)$, let $\subseteq_v$ be an arbitrary linear order on $K_v$. For any $u, w \in V(G)$, define $u \subseteq w$ if one of the following holds:

- $u = w$;
- $u \in K_v$, $w \in G \setminus V(H')$ and $v \leq w$;
- $u \in \overline{K}$, $w \in G \setminus V(H')$ and $u \leq w$;
- $u, w \in K_v$ and $u \subseteq_v w$;
- $u \in K_v$, $w \in K_v'$ and $v < v'$;
- $u \in \overline{K}$, $w \in K_v$ and $u \leq v$;
- $u \in K_v$, $w \in \overline{K}$ and $v \leq w$;
- $u, w \in \overline{K}$ and $u \leq w$.

We first show that $\subseteq$ is an elimination order for $H'$. The construction makes sure $\subseteq$ is a tree order. Let $u, w \in V(H')$. We show that if $u \leq w$, then either $u \subseteq w$ or $w \subseteq u$. We go through all possible cases: If $u = w$, we have $u \subseteq w$. If there is some $v \in V(J)$ such that $u, w \in K_v$, then $u \subseteq w$ or $w \subseteq u$. If $u \in K_v$, $w \in K_v'$ for two different $v, v' \in V(J)$, then $v \leq u \leq w$ and $v' \leq w$, so $v' \leq v$ and thus $w \subseteq u$. If $u \in \overline{K}$ and $w \in K_v$, then both $u, v \leq w$, so either $u \leq v$ or $v \leq u$, and thus either $u \subseteq w$ or $w \subseteq u$. The case where $u \in K_v$, $w \in \overline{K}$ is symmetric. Finally, if both $u, w \in \overline{K}$, then $u \subseteq w$. Thus if $uw \in E(H')$, we have $u \leq w$ or $w \leq u$ and therefore $u \subseteq w$ or $w \subseteq u$. Hence $\subseteq$ is an elimination order for $H'$.

Let $Z$ be a component of $G \setminus V(H')$. We assumed that $H$ contains all vertices of degree greater than $d$, and by the construction $H'$ also contains all those vertices. Thus $Z$ has maximum degree $d$.

Suppose $u, v \in V(H')$ are two vertices that are connected to $Z$, i.e. $E_G(u, V(Z)) \neq \emptyset \neq E_G(v, V(Z))$. We show that either $u \subseteq v$ or $v \subseteq u$. Note that there is a path $P$ through $Z \subseteq G \setminus V(H')$ from $u$ to $v$, i.e. all vertices in $P$, except for the endpoints, lie outside of $H'$. If $P$ contains no vertices from $J$, then the connected component $Z'$ of $G \setminus V(H)$ containing $P \setminus \{u, v\}$ satisfies $E_G(u, V(Z')) \neq \emptyset \neq E_G(v, V(Z'))$ and thus $u \leq v$ or $v \leq u$, and therefore by the above $u \subseteq v$ or $v \subseteq u$. Otherwise, $P$ contains vertices from $J$. Let $w$ be a $\leq$-minimal vertex in $V(P) \cap V(J)$. Then there is a path outside of $H'$ from $w$ to $u$, and also to $v$ (both part of $P$). Moreover, if neither $u \leq v$ nor $v \leq u$, then $w \leq u$ and $w \leq v$. Thus $u$ and $v$ are in $K_w$ (or in $K_{w'}$ for some $w' < w$), and therefore $u \subseteq v$ or $v \subseteq u$.

It remains to show that the size of $K_v$ is bounded by $k((k + 1)d)^{2k}$ for all $v \in V(J)$. Let $G'$ be the graph obtained from $G$ by adding an edge between two vertices $s, t \in V(J)$ whenever there is a path through $G \setminus V(H)$ between $s$ and $t$. This increases the degree of
vertices in $V(J)$ by at most $kd$, because each of these vertices is connected to at most $d$ components of $G \setminus V(H)$ and each of these is connected to at most $k$ vertices in $H$. Now there is a path between two vertices $s, t \in J$ in $G$ outside of $H'$ if and only if there is a path between $s$ and $t$ in $G'[V(J)]$. Moreover, $\leq$ is also an elimination order for $G'[V(J)]$. So, as $G'[V(J)]$ has tree-depth at most $k$ it does not contain a path of length more than $2^k$. Since each vertex on the path has degree at most $(k + 1)d$, we can reach at most $((k + 1)d)^2^k$ vertices in $V(H')$ on paths only containing vertices outside of $V(H')$. Thus $|K_u| \leq ((k + 1)d)^2^k$ and the height of $\subseteq$ is bounded by $k|K_u| \leq k((k + 1)d)^2^k$.

Next we introduce the notion of $d$-degree torso and prove that it captures the properties that we require of an elimination tree to degree $d$.

**Definition 5.8.** Let $G$ be a graph, let $d > 0$ and let $H$ be the induced subgraph of $G$ containing the vertices of degree larger than $d$. The $d$-degree torso of $G$ is the graph $C$ obtained from $H$ by adding an edge between two vertices $u, v \in H$ if there is a path through $G \setminus V(H)$ from $u$ to $v$ in $G$.

**Lemma 5.9.** Let $G$ be a graph and let $C$ be the $d$-degree torso of $G$. Let $H = G[V(C)]$ and let $\leq$ be an elimination order for $H$. Then $\leq$ is an elimination order for $C$ if and only if $\leq$ can be extended to an elimination order to degree $d$ for $G$.

**Proof.** Let $G, C, H$ and $\leq$ be as above.

Suppose $\leq$ is an elimination order for $C$. Since $C$ is a supergraph of $H$, this means that $\leq$ is an elimination order for $H$. Let $Z$ be a component of $G \setminus V(H)$. Since $C$ contains all vertices of degree greater than $d$, $Z$ has maximal degree $d$. If $E(Z, u) \neq \emptyset$ and $E(Z, v) \neq \emptyset$ for two vertices $u, v \in H$, then there is a path through $Z \subseteq G \setminus V(H)$ connecting $u$ and $v$, so by the definition of the $d$-degree torso $uv \in E(C)$ and thus $u, v$ are comparable. We can extend $\leq$ to an tree order $\leq'$ where all the vertices from $V(G) \setminus V(H)$ are maximal.

Conversely assume that $\leq$ can be extended to an elimination order to degree $d$ for $G$. Let $uv \in E(C)$. If $uv \in E(H)$, then $u$ and $v$ must be $\leq$-comparable. Otherwise $uv \notin E(H)$, so there is a path through $G \setminus V(H)$ from $u$ to $v$ in $G$, i.e. both $u$ and $v$ are connected to a component $Z$ of $G \setminus V(H)$ and thus comparable. Therefore $\leq$ is an elimination order for $C$.

**Lemma 5.10.** Let $G$ be a graph with elimination distance $k$ to degree $d$ and maximum degree $\Delta(G) \leq k + d$. Let $C$ be the $d$-degree torso of $G$ and let $\leq$ be a minimal height elimination order for $C$. Then $\leq$ has height at most $k(k+d+1)((k(k+d+1)+1)d)^2^k$. 

**Proof.** Let $\subseteq$ be a minimal elimination order to degree $d$ of $G$. Since $G$ has elimination distance to degree $d$ at most $k$, the height of $\subseteq$ is at most $k$. Let $H$ be the non-maximal subgraph of $G$ under $\subseteq$ and define

$$H' = G[V(H) \cup \{v \in V(G) \mid \deg_G(v) > d\}].$$

By Lemma 5.6, the subgraph $H'$ has an elimination order $\leq$ of height at most $k(k+d+1)$ that can be extended to an elimination order to degree $d$ for $G$. 

12
Let \( H'' = G\{v \in V(H') \mid \deg_G(v) > d\} \). By Lemma \([5.7]\), the subgraph \( H'' \) has an elimination order \( \leq \) of height at most \( k(k + d + 1)((k(k + d + 1) + 1)d^{2k(k+d+1)} \) that can be extended to an elimination order to degree \( d \) for \( G \).

Lastly note that \( V(H'') = V(C) \), so that by Lemma \([5.9]\) \( \leq \) is an elimination order for \( C \).

We are ready to prove the main result now:

**Theorem 5.11.** Let \( G \) be a graph that has elimination distance \( k \) to degree \( d \). Let \( \leq \) be a minimal elimination order of the \( d \)-degree torso \( C \) of \( G \). Then

(i) \( \leq \) is an elimination order to degree \( d \) of \( G \);

(ii) the height of \( \leq \) is at most

\[
k(k + d)^{2k+1} + k(1 + k + d)(k(1 + k + 2d))^{2k(1+k+d)+1}.
\]

**Proof.** (i) This follows from Lemma \([5.9]\).

(ii) Let \( C \) be the \((k + d)\)-degree torso of \( G \). We first show that the tree-depth of \( C \) is bounded by \( k((k+1)d)^{2k} \). To see this, let \( \subseteq \) be an elimination order to degree \( d \) of \( G \) of minimal height with non-maximal subgraph \( H \). Note that \( H \) contains all vertices of degree greater than \( k + d \), because the vertices in \( G \setminus V(H) \) are adjacent to at most \( k \) vertices in \( H \).

Let \( H' = G\{v \in V(H) \mid \deg_G(v) > k + d\} \). By Lemma \([5.7]\), the subgraph \( H' \) has an elimination order \( \leq \) of depth at most \( k((k+1)d)^{2k} \) that can be extended to an elimination order to degree \( k + d \) for \( G \). Note that \( V(H') = V(C) \), so by Lemma \([5.9]\), the order \( \leq \) is an elimination order for \( C \).

Let \( Z \) be a component of \( G \setminus V(H') \) and let \( C_Z \) be the \( d \)-degree torso of \( Z \). By Lemma \([5.10]\), there is an elimination order \( \leq_Z \) for \( C_Z \) of height at most \( k(k + \) \( d + 1) + 1)d^{2k(k+d+1)} \). Let \( v_Z \) be the maximal element in \( C \) such that there is a \( w \in C_Z \) with \( v_Z \leq w \). Define

\[
\leq' := \leq \cup \bigcup_Z \leq_Z \cup \bigcup_Z \{(v, w) \mid v \leq v_Z, w \in C_Z\} \cup \bigcup_Z \{(v, w) \mid v \leq v_Z \text{ or } v \in C_Z, w \in C, C \text{ a component of } Z \setminus V(C_Z), E(v, C) \neq \emptyset\}.
\]

Observe that \( C \cup \bigcup_Z C_Z \) is the \( d \)-degree torso of \( G \). Thus \( \leq' \) is an elimination order for the \( d \)-degree torso of \( G \), and the depth of \( \leq \) is bounded by its height.

The height of \( \leq' \) is bounded by

\[
\text{td}(C) + \max\{\text{td}(C_Z)\}_Z \leq k(k + d)^{2k+1} + k(1 + k + d)(k(1 + k + 2d))^{2k(1+k+d)+1}.
\]

\( \square \)
6 Elimination distance to bounded degree as parameter

In this section we show that graph isomorphism is FPT parameterized by elimination distance to bounded degree. The main idea is to construct a labelled directed tree \( T_G \) from a graph \( G \) (of elimination distance \( k \) to degree \( d \)) that is an isomorphism invariant for \( G \). That is, for graphs \( G_1 \) and \( G_2 \), the resulting trees \( T_{G_1} \) and \( T_{G_2} \) are isomorphic as labelled trees if, and only if, \( G_1 \cong G_2 \). We can then test the labelled trees for isomorphism using the tree canonisation algorithm from Lindell [12].

The tree \( T_G \) is obtained from \( G \) by taking a tree-depth decomposition of the \( d \)-degree torso of \( G \) and labelling the nodes with the isomorphism types of the low-degree components that attach to them. The tree-depth decomposition of a graph is just the elimination order in tree form. We formally define it as follows

**Definition 6.1.** Given a graph \( H \) and an elimination order \( \leq \) on \( H \), the *tree-depth decomposition* associated with \( \leq \) is the directed tree with nodes \( V(H) \) and an arc \( a \to b \) if, and only if, \( a < b \) and there is no \( c \) such that \( a < c < b \).

Note that, in general, the tree-depth decomposition of a graph that is not connected may be a forest. By results of Bouland et. al [2], we can construct a canonical tree-depth decomposition of a graph of tree-depth \( k \) in time that is tractable with parameter \( k \).

Before defining the tree \( T_G \) formally, we introduce one further piece of terminology.

**Definition 6.2.** Let \( G \) be a graph and let \( \leq \) be a tree order for \( G \). The *level* of a vertex \( v \in V(G) \) is the length of the chain \( \{ w \in V(G) \mid w \leq v \} \). We denote the level of \( v \) by \( \text{level}(v) \).

Given a graph \( G \) of elimination distance \( k \) to degree \( d \), let \( C \) be the \( d \)-degree torso of \( G \). Let \( Z \) be a component of \( G \setminus C \). We let \( Z^C \) denote the coloured graph that is obtained by colouring each vertex \( v \) in \( A \) by the colour \( \{ i \mid uv \in E(G) \text{ for some } u \in C \text{ with } \text{level}(u) = i \} \). We write \( F(Z^C) \) for the canonical form of this coloured graph given by Theorem 3.3. Note that, by the definition of elimination distance, there is, for each \( Z \) and \( i \) at most one vertex \( u \in C \) with \( \text{level}(u) = i \) which can occur as a neighbour of vertices in \( Z \).

We are now ready to define the labelled tree \( T_G \). Let \( C \) be the \( d \)-degree torso of \( G \) and let \( T \) be a canonical tree-depth decomposition of \( C \) and \( \leq \) the corresponding elimination order. The nodes of \( T_G \) are the nodes of \( T \) together with a new node \( r \), and the arcs are the arcs of \( T \) along with new arcs from \( r \) to the root of each tree in \( T \). Define, for each node \( u \) of \( T_G \), \( Z_u \) to be the set \( \{ Z \mid Z \text{ is a component of } G \setminus C \text{ with } u \leq -\text{maximal in } C \cap N_G(Z) \} \) (if \( u \neq r \)) and \( \{ Z \mid Z \text{ is a component of } G \setminus C \text{ with } C \cap N_G(Z) = \emptyset \} \) (if \( u = r \)). Each node \( u \) in \( T \) carries a label consisting of two parts:

- \( L_u \) := \( \{ \text{level}(w) \mid w < u \text{ and } uw \in E(G) \} \); and
- the multiset \( \{ F(A^C) \mid A \in \mathcal{A}_u \} \).

**Proposition 6.3.** For any graphs \( G \) and \( G' \), \( T_G \) and \( T_{G'} \) are isomorphic labelled trees if, and only if, \( G \cong G' \).
Proof. If $G \cong G'$ then, by construction, their $d$-degree torsos induce isomorphic graphs. The canonical tree-depth decomposition of Bouland et al. then produces isomorphic directed trees and the isomorphism must preserve the labels that encode the rest of the graphs $G$ and $G'$ respectively.

For the converse direction, suppose we have an isomorphism $\varphi$ between the labelled trees $T_G$ and $T_{G'}$. Since the label $L_u$ of any node $u$ encodes all ancestors of $u$ which are neighbours, $\varphi$ must preserve all edges and non-edges in the $d$-degree torso $C$ of $G$. To extend $\varphi$ to all of $G$, for each node $u$ in $T_G$, let $\beta_u$ be a bijection from $Z_u$ to the corresponding set $Z_{\varphi}(u)$ of components of $G' \setminus C'$, such that $F(Z^C) = F(\beta_u(Z)^C)$ (such a bijection exists as $u$ and $\varphi(u)$ carry the same label). Thus, in particular, there is an isomorphism between $Z^C$ and $\beta_u(Z)^C$, since they have the same canonical form. We define, for each $v \in V(G) \setminus C$, $\varphi(v)$ to be the image of $v$ under the isomorphism taking the component $Z$ containing $v$ to $\beta_u(Z)$. Note that this gives a well-defined function on $V(G)$, because for each such $v$, there is exactly one node $u$ of $T_G$ such that the component containing $v$ is in $Z_u$. We claim that $\varphi$ is now an isomorphism from $G$ to $G'$. Let $vw$ be an edge of $G$. If both $v$ and $w$ are in $C$, then either $v < w$ or $w < v$. Assume, without loss of generality, that it is the former. Then, level($v$) $\in L_w$ in the label of $w$ in $T_G$ and since $\varphi$ is a label-preserving isomorphism from $T_G$ to $T_{G'}$, $\varphi(v)\varphi(w)$ is an edge in $G'$. If both $v$ and $w$ are in $G \setminus C$, then there is some component $Z$ of $G \setminus C$ that contains them both. Since $\varphi$ maps $Z$ to an isomorphic component of $G' \setminus C'$, $\varphi(v)\varphi(w) \in E(G')$. Finally, suppose $v$ is in $C$ and $w$ in $G \setminus C$ and let $Z$ be the component containing $w$. Then $i := \text{level}(v)$ is part of the colour of $w$ in $Z^C$ and hence part of the colour of $\varphi(w)$ in the corresponding component of $G' \setminus C'$. Moreover, if $u$ is the $\leq$-maximal element in $C \cap N_G(Z)$, then we must have $v \leq u$. Thus $\varphi(v)$ is the unique element of level $i$ in $C' \cap N_{G'}(\beta_u(Z))$ and we conclude that $\varphi(v)\varphi(w) \in E(G')$. By a symmetric argument, we have that for any edge $vw \in E(G')$, $\varphi^{-1}(v)\varphi^{-1}(w) \in E(G)$ and we conclude that $\varphi$ is an isomorphism. \hfill \square

With this, we are able to establish our main result.

Theorem 6.4. Graph Isomorphism is FPT parameterized by elimination distance to bounded degree.

Proof. Suppose we are given a pair of graphs $G$ and $G'$ with $|V(G)| = |V(G')| = n$. We first compute the $d$-degree torso $C$ and $C'$ of $G$ and $G'$ respectively in $O(n^d)$ time. Using the result from Bouland et. al [2, Theorem 11], we can find a tree-depth decomposition for $C$ and $C'$ in time $O(h(k)n^d\log(n))$ for some computable function $h$. To compute the labels of the nodes in the trees (and hence obtain) $T_G$ and $T_{G'}$, we determine, for each $u \in C$, the set $\{\text{level}(w) \mid w < u$ and $uw \in E(G)\}$. This can be done in time $O(n^2)$. Then, we identify the components of $G \setminus C$ and $G' \setminus C'$, and colour the vertices with the levels of their neighbours in $C$ (or $C'$ respectively). This can be done in $O(n^2)$ time. Finally, we compute for each coloured component $Z^C$ the canonical representative $F(Z^C)$ which, by Theorem 3.3, can be done in polynomial time (where the degree of the polynomial depends on $d$).
Having obtained $T_G$ and $T_G'$, we can test them for isomorphism in linear time, using Lindell’s canonisation algorithm \cite{12} to compare the labelled trees.

7 Conclusion

We introduce a new way of parameterizing graphs by their distance to triviality, namely by elimination distance. In the particular case of the graph isomorphism problem, taking triviality to mean graphs of bounded degree, we show that the problem is FPT.

A natural question that arises is what happens when we take other classes of graphs for which graph isomorphism is known to be tractable as our basic “trivial” classes. For instance, in the result of Kratsch and Schweitzer \cite{11} that graph isomorphism is FPT when parameterized by the size of the minimum feedback vertex set, the parameter could be described as *deletion distance to trees*. Does the result extend to graphs parameterized by *elimination distance to trees*? What can we say about isomorphism when parameterized by elimination distance to planar graphs? It should be noted that techniques such as those deployed in the present paper are unlikely to work in this case.

Our techniques rely on identifying a canonical subgraph which defines an elimination tree into the trivial class. In the case of planar graphs, consider graphs which are subdivisions of $K_5$, each of which is deletion distance 1 away from planarity. However the deletion of *any* vertex yields a planar graph and it is therefore not possible to identify a canonical such vertex.

More generally, the notion of elimination distance to triviality seems to offer promise for defining tractable parameterizations for many graph problems other than isomorphism. This is a direction that bears further investigation.

References

[1] L. Babai and E. M. Luks, *Canonical Labeling of Graphs*, Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing (New York, NY, USA), ACM, 1983, pp. 171–183.

[2] A. Bouland, A. Dawar, and E. Kopczyński, *On Tractable Parameterizations of Graph Isomorphism*, Parameterized and Exact Computation, Springer Berlin Heidelberg, 2012, pp. 218–230.

[3] R. Diestel, *Graph Theory*, Springer, January 2000.

[4] R. G. Downey and M. R. Fellows, *Parameterized Complexity*, Springer Verlag, October 2012.

[5] S Evdokimov and I Ponomarenko, *Isomorphism of coloured graphs with slowly increasing multiplicity of Jordan blocks*, Combinatorica 19 (1999), no. 3, 321–333.
[6] M. R. Fellows, D. Lokshtanov, N. Misra, F. A. Rosamond, and S. Saurabh, *Graph layout problems parameterized by vertex cover*, Proc. 19th Intl. Symp. Algorithms and Computation, 2008, pp. 294–305.

[7] I. S. Filotti and J.N. Mayer, *A polynomial-time algorithm for determining the isomorphism of graphs of fixed genus*, STOC '80: Proceedings of the twelfth annual ACM symposium on Theory of computing, ACM Request Permissions, April 1980.

[8] J Flum and M Grohe, *Parameterized Complexity Theory*, Springer, May 2006.

[9] M. Grohe and D. Marx, *Structure theorem and isomorphism test for graphs with excluded topological subgraphs*, Proc. 44th Symp. on Theory of Computing, 2012, pp. 173–192.

[10] J. Guo, F. Hüffner, and R. Niedermeier, *A Structural View on Parameterizing Problems: Distance from Triviality*, Parameterized and Exact Computation, Springer Berlin Heidelberg, 2004, pp. 162–173.

[11] S. Kratsch and P. Schweitzer, *Isomorphism for graphs of bounded feedback vertex set number*, SWAT’10: Proceedings of the 12th Scandinavian conference on Algorithm Theory (Berlin, Heidelberg), Springer-Verlag, June 2010, pp. 81–92.

[12] S. Lindell, *A logspace algorithm for tree canonization (extended abstract)*, STOC ’92: Proceedings of the twenty-fourth annual ACM symposium on Theory of computing, ACM Request Permissions, July 1992.

[13] D. Lokshtanov, M. Pilipczuk, M. Pilipczuk, and S. Saurabh, *Fixed-parameter tractable canonization and isomorphism test for graphs of bounded treewidth*, arxiv:1404.0818 [cs.DS], 2014.

[14] E. M. Luks, *Isomorphism of graphs of bounded valence can be tested in polynomial time*, Journal of Computer and System Sciences 25 (1982), no. 1, 42–65.

[15] G. Miller, *Isomorphism testing for graphs of bounded genus*, STOC ’80: Proceedings of the twelfth annual ACM symposium on Theory of computing, ACM Request Permissions, April 1980.

[16] R. Niedermeier, *Invitation to Fixed-Parameter Algorithms*, Oxford University Press, February 2006.

[17] I. N. Ponomarenko, *The isomorphism problem for classes of graphs closed under contraction*, Journal of Soviet Mathematics 55 (1991), no. 2, 1621–1643.

[18] S. Toda, *Computing Automorphism Groups of Chordal Graphs Whose Simplicial Components Are of Small Size*, IEICE - Transactions on Information and Systems E89-D (2006), no. 8, 2388–2401.
[19] R. Uehara, S. Toda, and T. Nagoya, *Graph isomorphism completeness for chordal bipartite graphs and strongly chordal graphs*, Discrete Applied Mathematics 145 (2005), no. 3, 479–482.

[20] K. Yamazaki, H. L. Bodlaender, B. De Fluiter, and D. M. Thilikos, *Isomorphism for graphs of bounded distance width*, CIAC ’97 (Berlin, Heidelberg), Springer Berlin Heidelberg, 1997, pp. 276–287.