Algorithmica
DOI 10.1007/s00453-015-0045-3

Graph Isomorphism Parameterized by Elimination Distance to Bounded Degree

Jannis Bulian[1] · **Anuj Dawar**[1]

Received: 11 October 2014 / Accepted: 27 July 2015

Abstract A commonly studied means of parameterizing graph problems is the deletion distance from triviality (Guo et al., Parameterized and exact computation, Springer, Berlin, pp. 162–173, 2004), which counts 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 canonisation, and thus graph isomorphism, is FPT when parameterized by elimination distance to bounded degree, extending results of Bouland et al. (Parameterized and exact computation, Springer, Berlin, pp. 218–230, 2012).

Keywords Graph · Isomorphism · Canonization · Parameterized complexity · Fixed-parameter tractable · Bounded degree · Graph theory · Complexity theory · Tree-depth

Research supported in part by EPSRC Grant EP/H026835, DAAD Grant A/13/05456, and DFG project *Logik, Struktur und das Graphenisomorphieproblem*.

✉ Jannis Bulian
jannis.bulian@gmail.com; jannis.bulian@cl.cam.ac.uk

Anuj Dawar
anuj.dawar@cl.cam.ac.uk

[1] Computer Laboratory, University of Cambridge, Cambridge, UK

Published online: 11 August 2015

1 Introduction

The *graph isomorphism problem* (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 P nor known to be 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 GI by means of structural parameters of the graphs which can be used to study the problem from the point of view of parameterized complexity. For instance, it is known that GI is in XP parameterized by the genus of the graph [8,16], by maximum degree [1,15] and by the size of the smallest excluded minor [18], or more generally, the smallest excluded topological minor [10]. 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 [21], the maximum size of a simplical component [19,20] and minimum feedback vertex set [12]. Bouland et al. [2] showed that the problem is FPT when parameterized by the tree depth of a graph and extended this result to a parameter they termed *generalised tree depth*. In a recent advance on this, Lokshtanov et al. [14] have shown that graph isomorphism 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 canonisation 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. [11] call *distance to triviality* and it may be of interest in the context of other graph problems. It should be noted that graphs with degree bounded by d (for $d \geq 3$) have unbounded tree-width and the same is *a fortiori* true of graphs with elimination distance k to degree d.

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 a minimal vertex cover in G and is a parameter that has been much studied (see for instance [7]). 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 GI among graphs with maximum degree d, we may take this as our trivial base case. We then parameterize 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 Sect. 3 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 canonisation

is FPT. Along the way, in Sect. 4, we establish a number of characterisations of the parameter that may be interesting in themselves. In particular, we define the notion of an *elimination order* which is a partial order on the vertices indicating the order in which they need to be eliminated to reduce the degree to d.

It is easy to see that in a graph G that has elimination distance k to degree d, all vertices of sufficiently high degree have to be eliminated, but it may be the case that in an optimal elimination order some low-degree vertices may also need to be eliminated. A key idea in our proof is to show that G contains a canonically defined set of vertices C (which we call its torso), including the high-degree vertices so that if G has elimination distance k to degree d, then we can reduce the degree to d by eliminating just vertices in C, and moreover this yields an elimination order whose height is bounded by a function of k and d. This is established in Sect. 5. It should be noted that the parameter termed *generalised tree depth* in [2] can be seen as a special case of elimination distance to degree 2.

With this characterisation established, we are able to present the canonisation algorithm in Sect. 6. The central technique used here is inspired by Lindell's tree canonisation algorithm [13]. We define an order $\sqsubseteq$ on isomorphism types of coloured graphs equipped with an elimination order recursively on the height of this order, with the base case being a canonical order on coloured graphs of bounded degree. We then show that given a graph G, separated into its torso C and a low-degree part, we can construct an elimination order for G (i.e. a tree-depth decomposition of C) which is $\sqsubseteq$-minimal. We use the result of Bouland et al. [2] on canonisation of bounded tree-depth graphs to bound the branching in the search and establish that canonisation is FPT.

2 Preliminaries

2.1 Parameterized Complexity

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 Σ. A *parameterization* is a computable function $\kappa : \Sigma^* \to \mathbb{N}$. We say that L is *fixed-parameter tractable* with respect to κ if we can decide whether an input $x \in \Sigma^*$ is in L in time $O(f(\kappa(x)) \cdot |x|^c)$, where c is a constant and f is some computable function. For a thorough discussion of the subject we refer to the books by Downey and Fellows [4], Flum and Grohe [9] and Niedermeier [17]. We follow notation and terminology from [9].

2.2 Graphs

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 [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 $N_G(S) := \bigcup_{v \in S} N_G(v) \backslash S$. The *degree* of a graph G is the maximum 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 may 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 $G \backslash A$ for $G[V(G) \backslash A]$. For a vertex $v \in V(G)$, we write $G \backslash v$ for $G \backslash \{v\}$.

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. $(V(P) \backslash \{v, w\}) \subseteq V(H)$.

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 $\varphi : V(G) \to V(G')$ such that for all $v, w \in V(G)$ we have that $vw \in E(G)$ if, and only if, $\varphi(v)\varphi(w) \in E(G')$. 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 $c : V(G) \to \{1, \ldots, k\}$ 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 $c : V(G) \to \{1, \ldots, k\}, c' : V(G') \to \{1, \ldots, k\}$ 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 $\varphi(v)\varphi(w) \in E(G')$;
- 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 $F : \mathsf{C} \to \mathsf{C}$ such that

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

A canonical form F for a class C gives rise to a natural pre-order on C, which we denote $\sqsubseteq_F$ whereby $G \sqsubseteq_F H$ just in case $F(G)$ is lexicographically smaller than $F(H)$. Then for any pair of graphs G and H in C, at least one of $G \sqsubseteq_F H$ or $H \sqsubseteq_F G$ must hold, and both hold if, and only if, $G \cong H$.

2.3 Orders

Recall that a *partial order* is a binary relation $\leq$ on a set S which is reflexive, anti-symmetric and transitive. We say a set $T \subseteq S$ is a *chain* if it is totally ordered by $\leq$. 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 a chain, we say $\leq$ is a *tree order*. (Note that the covering relation of a tree order is not necessarily a tree, but may be a forest.)

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 in it is k.

We write $td(G)$ for the *tree-depth* of G, which is defined as follows

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

Note that there is an elimination order $\leq$ of height k for a graph G if, and only if, $td(G) \leq k$.

2.4 Isomorphism on Bounded-Degree Graphs

Luks [15] shows that isomorphism of bounded-degree graphs is decidable in polynomial time. It is well-known that this result extends, by an easy reduction, to *coloured* graphs of bounded-degree. Though this reduction is folklore, for the sake of completeness, we present it here explicitly.

Proposition 2.2 *The isomorphism problem for coloured graphs of degree d can be reduced to the isomorphism problem of graphs of degree $d + 2$ 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 φ 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 φ' 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 φ 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 φ' also preserves colours, note that φ must map the cycle containing $u^v_{c(v)}$ to the cycle containing $u^{\varphi'(v)}_{c(\varphi'(v))}$ and therefore $c(v) = c(\varphi'(v))$.

Note that if G and G' are graphs of degree d, then H, H' are graphs of degree $d+2$.

As Luks [15] proves that isomorphism of bounded degree graphs can be decided in polynomial time, we have the following:

Theorem 2.3 *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 2.4 *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.*

3 Deletion Distance to Bounded Degree

We first study the notion of deletion distance to bounded degree and establish in this section that graph isomorphism is FPT with this parameter. Though the result in this section is subsumed by the more general one in Sect. 6 and also by a result of Kratsch and Schweitzer [12], it provides a useful warm-up and a tighter bound, exponential in the parameter. In the present warm-up we only give an algorithm for the graph isomorphism problem, though the result easily holds for canonisation as well (and this follows from the more general result in Sect. 6). 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. [11]. In the context of graph isomorphism, we have chosen triviality to mean graphs of degree at most d.

Definition 3.1 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 $G \backslash \{v_1, \ldots, v_k\}$ has degree d. We call the 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 a d-deletion set is just a vertex cover and the minimum 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 in linear time a set U of vertices of size polynomial in k so that any deletion set must be found in U if one exists. This then allows for exhaustive search over all potential isomorphisms. We call U a *bounding set* for the deletion set. The construction is reminiscent of a kernelization result in [6] for the problem of determining whether a graph has deletion distance k to degree d, though that does not yield a bounding set containing all deletion sets.

Theorem 3.2 *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 d and, moreover, if G' has deletion distance at most k', then any minimum size d-deletion set for G' is contained in U.*

Proof Let $A := \{v \in V(G) \mid \deg(v) > k + d\}$. Now, if R is a minimum size d-deletion set for G and G has deletion distance at most k to degree d, then $|R| \leq k$ and the vertices in $V(G \backslash R)$ have degree at most $k + d$ in G. So $A \subseteq R$. This means that if $|A| > k$, 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 $G' := G \backslash A$ and $k' := k - |A|$. We have shown that every d-deletion set of size at most k must contain A. 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 minimum size 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 $G \backslash (R' \setminus \{v\})$ also has maximal degree d, which contradicts the assumption that R' is of minimum size. Thus $v \notin R'$.

Note that the vertices in $G' \backslash (R' \cup N(R'))$ 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 A and U defined as above can be found in linear time, and G', k' can be computed from A in linear time.

Remark It may be noted that the set U is canonical in the sense that if there is an isomorphism from G to a graph H, this must take U to the corresponding set in H. But, this is not essential to our argument below. What is important is that U contains *all* possible deletion sets for G. In particular, if $U = \emptyset$ and $k' > 0$, then there are no d-deletion sets of size at most k'.

Next we see how the bounding set 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 2.4. To be precise, we define the coloured graphs G' and H' which are obtained from $G \backslash S$ and $H \backslash T$ 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 indices of its neighbours in S. Vertices in H' are similarly coloured by the sets of indices 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 2.4 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 bounding sets in the graphs obtained from Theorem 3.2. 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 $O^*\left(\binom{k^3}{k}^2 \cdot k!\right)$, which is $O^*\left(2^{7k \log k}\right)$.

4 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 and canonisation we again define triviality to mean bounded degree, so we look at the elimination distance to bounded degree.

Definition 4.1 The *elimination distance to degree d* of a graph G is defined as follows:

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

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 as defined below. The idea is that if G is a connected graph we can choose a vertex v so that deleting it strictly reduces the elimination distance to degree d. We make $v \leq u$ for all vertices u and vertices in distinct components of $G \setminus v$ are made incomparable with respect to $\leq$. We proceed in this fashion until we have components with degree at most d and all vertices in these are incomparable (and indeed $\leq$-maximal). If C is such a degree d component, then the vertices of $G \setminus C$ that are neighbours of vertices in C must be linearly ordered by $\leq$ and they all precede the vertices of C in the order $\leq$. Thus, if we associate with every vertex v, the set S_v of neighbours of v that are $\leq$-incomparable with v we can note the following: S_v has at most d elements for any v; if S_v is not empty, then v must be $\leq$-maximal; and if $u \in S_v$ then u and v must have the same $\leq$-predecessors. This can be taken as the defining property of an elimination order to degree d, as below.

Definition 4.2 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

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

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

Proof Suppose first that $ed_d(G) \leq k$. We proceed by induction on k. If $k = 0$, then G has no vertex of degree larger than d and we define the elimination order $\leq$ 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. Thus, there is a vertex $a \in V(G)$ such that the components $C_1, \ldots, C_r$ of $G \backslash a$ 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 4.2. For each $v \in V(C_i)$ define

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

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 $v \in V(G) \backslash a$ 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. We again proceed by induction on k. If $k = 0$, then $\leq$ is empty and all vertices v in G are $\leq$-maximal. Thus, S_v contains all neighbours of v and therefore v has at most d neighbours. In other words, $\Delta(G) \leq d$ as required.

Now, suppose $k > 0$. If v is an $\leq$-minimal element then for any u for which there is a path from v to u, we must have $v \leq u$. Thus, if G is connected, there is a unique $\leq$-minimal element v. Note that $\leq$ restricted to a component C of $G \backslash v$ has height $k - 1$ and thus by the induction hypothesis we have that $ed_d(C) \leq k - 1$. If G is not connected, this argument can be applied to each component of G.

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 $G \backslash V(H)$ has degree at most d and is connected to H along just one branch (this is defined more formally below).

Proposition 4.4 *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* $k-1$ *of* $G[A]$*; and*
2. $G\backslash A$ *has degree at most* d*;*
3. *if* C *is the vertex set of a component of* $G\backslash A$*, 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 4.2, $S_v = \emptyset$. Hence if there is an edge between $u, v \in A$, either $u < v$ or $v < u$, and (1) follows.

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

To show (3), let C be the vertex set of a component of $G\backslash A$ and let $u, v \in A$ be such 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 4.2, $u < a$ and $v < b$. Moreover, there is a path from a to b through C and as all vertices along this path are $\leq$-maximal, if (a', b') is an edge in the path, it must be that $\{w \mid w < a'\} = \{w \mid w < b'\}$. By transitivity, $\{w \mid w < a\} = \{w \mid w < b\}$, and so $u < b$ and $v < a$. Since $\leq$ is a tree-order, the set $\{w \mid w < a\}$ is linearly ordered and we conclude that u and v are comparable.

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

Proposition 4.5 *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. $G\backslash A$ *has degree at most* d*;*
2. *if* C *is the vertex set of a component of* $G\backslash A$*, 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

$$\begin{aligned}\leq := &\leq_A \cup \{(v, v) \mid v \in (V(G)\backslash A)\} \\ &\cup \{(u, v) \mid u \in A, v \in C, C \quad \text{a component of} G\backslash A, E(w, C) \neq \emptyset \quad \text{for some } u \leq w\}.\end{aligned}$$

Then it is easily seen that $\leq$ is a tree order on G. Indeed, $\leq_A$ is, by assumption, a tree order on A and for any $v \in V(G)\backslash A$, assumption 2 guarantees that $\{w \mid w \leq v\}$ is linearly ordered.

Let $v \in V(G)$ and let S_v be as in Definition 4.2. Suppose $S_v \neq \emptyset$. Then $v \in (V(G)\backslash A)$ and has degree at most d in $G\backslash A$. By the construction v is $\leq$-maximal. Let $u \in S_v$. Then there is a component C of $G\backslash A$ 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 $\leq$ to degree d we call the subgraph of $V(G)$ induced by the non-maximal elements of the order $\leq$ the *non-maximal subgraph of* G *under* $\leq$.

In the proof of Proposition 4.5 above, a suitable tree order on a subset A of $V(G)$ is extended to an elimination order to degree d of G by making all vertices not in A maximal in the order. This is a form of construction we use repeatedly below.

The alternative characterisations of elimination order to degree d established above are very useful. In the next sections, we use them to construct a *canonical* elimination order to degree d of G, based on an elimination order of a graph we call the *torso* of G, which consists of the high-degree vertices of G, along with some additional edges.

5 Separation into High and Low Degree Parts

The aim of this section is to show that if a graph G has elimination distance k to degree d, then there is an elimination order to degree d whose height is still bounded by a function of d and k and in which the set of non-maximal elements is *canonical*. To be precise, we identify a graph which we call the d-degree torso of G, which contains all the vertices of G of degree more than d and has additional edges to represent paths between these vertices that go through the rest of G. We show that this torso necessarily has tree-depth bounded by a function of k and d and an elimination order witnessing this can be extended to an elimination order to degree d of G. The result is established through a series of lemmas. A pattern of construction that is repeatedly used here is that we define a certain set A of vertices of G and construct an elimination order of $G[A]$. It is then shown that extending the order by making all vertices in $V(G)\setminus A$ maximal yields an elimination order to degree d of G. Necessarily, in this extended order, all the non-maximal elements are in A.

We first do this for graphs of degree bounded by $k+d$. To be precise, the following lemma establishes that if G has elimination distance k to degree d and moreover the degree of G is at most $k+d$, then we can construct an alternative elimination order to degree d on G in which all the vertices of degree greater than d are included in the non-maximal subgraph and the height of the new elimination order to degree d is still bounded by a function of k and d.

Lemma 5.1 *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, and let $A = V(H) \cup \{v \in V(G) \mid \deg_G(v) > d\}$. Then G has an elimination order $\sqsubseteq$ to degree d of height at most $k(k+d+1)$ for which A is the set of non-maximal elements.*

Proof Let G, H, A and $\leq$ be as in the statement of the lemma. We will adapt $\leq$ to an elimination order $\sqsubseteq$ of $G[A]$.

For each $w \in A\setminus V(H)$ 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 4.2, the vertices in $N(C_w)$ are linearly ordered by $\leq$. We write $b(w)$ to denote the unique $\leq$-maximal element of $N(C_w)$ for each $w \in A\setminus V(H)$. For each $b \in V(H)$, let $W_b := \{w \in A\setminus V(H) \mid b(w) = b\}$, 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 $u \leq v$;
- $u \in H$, $v \in G\setminus A$ and $u \leq v$;
- $u \in H$, $v \in A\setminus V(H)$ and $u \leq b(v)$;

- $u \in A \backslash V(H)$, $v \in G \backslash A$ and $b(u) \leq v$;
- $u, v \in A \backslash V(H)$, $b(u) = b(v)$ and $u \sqsubseteq_b v$.

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

For each $b \in V(H)$, the set $\{v \in H \mid v \leq b\}$ has at most k elements, by the assumption on the height of the order $\leq$. Since G has maximum degree $k + d$ and $W_b \subseteq N(\{v \in H \mid v \leq b\})$, we have that W_b has at most $k(k + d)$ vertices. Since the height of any $\sqsubseteq$ chain is at most the height of a $\leq$-chain plus $|W_b|$, we conclude that the height of $\sqsubseteq$ is at most $k(k + d + 1)$.

The lemma above allows us to re-arrange the elimination order so that it includes all vertices of large degree. In contrast, the next lemma gives us a means to re-arrange the elimination order so that all vertices of small degree are made maximal in the order. This is again achieved while keeping the height of the elimination order bounded by a function of k and d.

Lemma 5.2 *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, and let $A = \{v \in V(H) \mid \deg_G(v) > d\}$. Then, there is an elimination order to degree d of G of height at most $k((k + 1)d)^{2^k} + 1$ for which A is the set of non-maximal elements.*

Proof Let G, H, A 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 $G[A]$ from $\leq$, making sure that it has height at most $k((k + 1)d)^{2^k}$, and satisfying the conditions of Proposition 4.5. This then extends to an elimination order to degree d of G by making all vertices not in A maximal, as in that Proposition.

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

1. $v \leq w$;
2. there is a path from v to w through $G \backslash A$; and
3. for any u with $u < v$, there is no path from u to w through $G \backslash A$.

Note that because $\leq$ is a tree order and due to the third condition, the sets K_v are pairwise disjoint. Let $\overline{K} := A \backslash (\bigcup_{v \in V(J)} K_v)$ be the set of vertices in A that are not contained in K_v for any v.

For each $v \in V(J)$, let $\sqsubseteq_v$ be an arbitrary linear order on K_v. The idea behind the construction below is that we replace v in the elimination order by K_v, ordered by $\sqsubseteq_v$. Formally, for any $u, w \in V(G)$, define $u \sqsubseteq w$ if one of the following holds:

- $u = w$;
- $u \in K_v$, $w \in G \backslash A$ and $v \leq w$;
- $u \in \overline{K}$, $w \in G \backslash A$ and $u \leq w$;
- $u, w \in K_v$ and $u \sqsubseteq_v w$;
- $u \in K_v$, $w \in K_{v'}$ 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 $u \preceq w$.

We first show that $\sqsubseteq$ is an elimination order for $G[A]$. The construction ensures $\sqsubseteq$ is a tree order. Let $u, w \in A$. We show that if $u \preceq w$, then either $u \sqsubseteq w$ or $w \sqsubseteq u$. We go through all possible cases: If $u = w$, we have $u \sqsubseteq w$. If there is some $v \in V(J)$ such that $u, w \in K_v$, then $u \sqsubseteq w$ or $w \sqsubseteq u$. Suppose $u \in K_v$, $w \in K_{v'}$ for $v, v' \in V(J)$. By definition of K_v, $v \preceq u$ and similarly $v' \preceq w$. Since we also have $u \preceq w$, by the tree order property either $v \preceq v'$ or $v' \preceq v$ and thus $u \sqsubseteq w$ or $w \sqsubseteq u$. If $u \in \overline{K}$ and $w \in K_v$, then both $u, v \preceq w$, so either $u \preceq v$ or $v \preceq u$, and thus either $u \sqsubseteq w$ or $w \sqsubseteq u$. The case where $u \in K_v$, $w \in \overline{K}$ is symmetric. Finally, if both $u, w \in \overline{K}$, then $u \sqsubseteq w$. Thus if $uw \in E(G)$, we have $u \preceq w$ or $w \preceq u$ and therefore $u \sqsubseteq w$ or $w \sqsubseteq u$. Hence $\sqsubseteq$ is an elimination order for $G[A]$.

Let Z be a component of $G \setminus A$. By assumption H contains all vertices of degree greater than d, and hence A also contains all those vertices. Thus Z has maximum degree d.

Suppose $u, v \in A$ 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 $u \sqsubseteq v$ or $v \sqsubseteq u$. Note that there is a path P through $Z \subseteq G \setminus A$ from u to v, i.e. all vertices in P, except for the endpoints, lie outside of A. 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 $u \preceq v$ or $v \preceq u$, and therefore by the above $u \sqsubseteq v$ or $v \sqsubseteq u$.

Otherwise, P contains vertices from J. Let w be a $\preceq$-minimal vertex in $V(P) \cap V(J)$. Then there is a path outside of A from w to u, and also to v (both part of P). Moreover, if neither $u \preceq v$ nor $v \preceq u$, then $w \preceq u$ and $w \preceq v$. Thus u and v are in K_w (or in $K_{w'}$ for some $w' \prec w$), and therefore $u \sqsubseteq v$ or $v \sqsubseteq u$.

It remains to show that the size of K_v is bounded by $k((k+1)d)^{2^k}$ 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 A if and only if there is a path between s and t in $G'[V(J)]$. Moreover, $\preceq$ 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 A on paths only containing vertices outside of A. Thus $|K_v| \leq ((k+1)d)^{2^k}$ and the height of $\sqsubseteq$ is bounded by $k|K_v| \leq k((k+1)d)^{2^k}$.

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 5.3 Let G be a graph, let $d > 0$ and let H be the subgraph of G induced by the vertices of degree larger than d. The *d-degree torso of G* is the graph 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 5.4 *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 of height h if, and only if, $\leq$ can be extended to an elimination order to degree d for G of height $h + 1$.*

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 $G \backslash V(H)$. 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 $Z \subseteq G \backslash V(H)$ connecting u and v, so by the definition of the d-degree torso $uv \in E(C)$ and thus u, v are $\leq$-comparable. We can extend $\leq$ to a tree order $\leq'$ on $V(G)$ where all the vertices from $V(G) \backslash V(H)$ 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 $G \backslash V(H)$ from u to v in G, i.e. both u and v are connected to a component Z of $G \backslash V(H)$ and thus comparable. Therefore $\leq$ is an elimination order for C.

The next lemma establishes an upper bound on the tree-depth of the torso of a graph when the maximum degree is bounded.

Lemma 5.5 *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 minimum height elimination order for C. Then $\leq$ has height at most $k(k + d + 1)((k(k + d + 1) + 1)d)^{2^{k(k+d+1)}}$.*

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

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

By Lemma 5.1, the graph $G[A]$ has an elimination order $\preceq$ of height at most $k(k+d+1)$ that can be extended to an elimination order to degree d for G.

Let $A' = \{v \in A \mid \deg_G(v) > d\}$. By Lemma 5.2, the graph A' has an elimination order $\leq$ of height at most $k(k + d + 1)((k(k + d + 1) + 1)d)^{2^{k(k+d+1)}}$ that can be extended to an elimination order to degree d for G.

Lastly note that $A' = V(C)$, so that by Lemma 5.4, $\leq$ is an elimination order for C.

We are now ready to prove the main result:

Theorem 5.6 *Let G be a graph that has elimination distance k to degree d. Let $\leq$ be a minimum height elimination order of the d-degree torso G. Then $\leq$ can be extended to an elimination order to degree d of G of height at most*

$$k((k + 1)(k + d))^{2^k} + k(1 + k + d)(k(1 + k + 2d))^{2^{k(1+k+d)}} + 1.$$

Proof We show that the d-degree torso of G has an elimination order of height at most $k((k+1)(k+d))^{2^k} + k(1+k+d)(k(1+k+2d))^{2^{k(1+k+d)}}$. The theorem then follows by Lemma 5.4.

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

Let $A = \{v \in V(H) \mid \deg_G(v) > k+d\}$. By Lemma 5.2, the graph $G[A]$ has an elimination order $\preceq$ of depth at most $h := k((k+1)(k+d))^{2^k}$ that can be extended to an elimination order to degree $k+d$ of G of height $h+1$. Note that $A = V(C)$, so by Lemma 5.4, the order $\preceq$ is an elimination order for C. Let $\preceq'$ denote its extension to G.

Let Z be a component of $G \setminus A$ and let C_Z be the d-degree torso of Z. By Lemma 5.5, there is an elimination order $\preceq_Z$ for C_Z of height at most $k(k+d+1)((k(k+d+1)+1)d)^{2^{k(k+d+1)}}$. Let v_Z be the $\preceq$-maximal element in C such that there is a $w \in C_Z$ with $v_Z \preceq' w$. Define

$$\preceq' := \preceq \cup \bigcup_Z \preceq_Z \cup \bigcup_Z \{(v, w) \mid v \preceq' v_Z, w \in C_Z\}.$$

Observe that $C \cup \bigcup_Z C_Z$ has as a subgraph the d-degree torso of G. Thus $\preceq'$ is an elimination order for the d-degree torso of G. The height of $\preceq'$ is bounded by

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

□

6 Canonisation Parameterized by Elimination Distance to Bounded Degree

In this section we show that graph canonisation, and thus graph isomorphism, is FPT parameterized by elimination distance to bounded degree. The algorithm is based on Lindell's algorithm [13] that decides isomorphism of trees by establishing an ordering on them. We extend a version of this algorithm used by Bouland et al. [2] to decide isomorphism on graphs of bounded tree-depth.

The central idea is to define a pre-order $\sqsubseteq$ on graphs $(G, \preceq)$ equipped with an elimination order to degree d, with the property that $(G, \preceq) \sqsubseteq (G', \preceq')$ and $(G', \preceq) \sqsubseteq (G, \preceq')$ if, and only if, $(G, \preceq)$ is isomorphic to $(G', \preceq')$. We then exhibit an algorithm that constructs, given a graph G, the elimination order to degree d on G that makes $(G, \preceq)$ $\sqsubseteq$-minimal. This serves as a canonical representative of G. Before defining $\sqsubseteq$ formally, we need some additional terminology.

Definition 6.1 Let G be a graph and let $\preceq$ be a tree order for G. The *level* of a vertex $v \in V(G)$ is the length of the chain $\{w \in V(G) \mid w \preceq v\}$. We denote the level of v by $level_{\preceq}(v)$.

Definition 6.2 Let G be a graph. We denote the number of connected components of G by $\#G$.

The definition of $\sqsubseteq$ is by induction on the height of the order $\leq$. Inductively, we need to consider coloured graphs and so formally, $\sqsubseteq_k$ is a pre-order on triples $(G, \chi, \leq)$ where $\chi : V(G) \to X$ is a colouring of G and $\leq$ is an elimination order to degree d of G. We always assume that the set of colours X is an initial segment of the natural numbers and so, in particular, linearly ordered.

Definition 6.3 For a graph G, with a colouring $\chi : V(G) \to X$ and a vertex $v \in V(G)$, we write χ_v for the colouring of $G\backslash v$ given by $\chi_v : V(G\backslash v) \to X \times \{0, 1\}$ where $\chi_v(u) = (\chi(u), 1)$ if $(u, v) \in E(G)$ and $\chi_v(u) = (\chi(u), 0)$ if $(u, v) \notin E(G)$.

By Theorem 2.4 we can find a canonical form of each coloured graph Z of degree at most d in polynomial time, and thus a pre-order $\sqsubseteq_0$ such that for every pair of Z, Z' of such graphs, either $Z \sqsubseteq_0 Z'$ or $Z' \sqsubseteq_0 Z$ and $Z \cong Z'$ if, and only if, both $Z \sqsubseteq_0 Z'$ and $Z' \sqsubseteq_0 Z$.

In the following definition we extend this to a pre-order $\sqsubseteq$ on coloured graphs with an elimination order to degree d by induction on the height of the order.

Definition 6.4 Let G, G' be graphs with colourings χ and χ' and respective elimination orders $\leq, \leq'$ to degree d. We say that $(G, \chi, \leq) \sqsubseteq (G', \chi', \leq')$ if one of the following holds:

1. $\text{height}(\leq) < \text{height}(\leq')$;
2. $\text{height}(\leq) = \text{height}(\leq') = 0$ and $(G, \chi) \sqsubseteq_0 (G', \chi')$;
3. $\text{height}(\leq) = \text{height}(\leq') > 0$ and $\#G < \#G'$;
4. $\text{height}(\leq) = \text{height}(\leq') > 0, \#G = \#G' = 1$ and $\chi(v) < \chi'(v')$ where v, v' are minimal elements of G, G' with respect to $\leq, \leq'$ respectively;
5. $\text{height}(\leq) = \text{height}(\leq') > 0, \#G = \#G' = 1$ and

$$(G\backslash v, \chi_v, \leq) \sqsubseteq (G'\backslash v', \chi'_{v'}, \leq'),$$

   where v, v' are minimal elements of G, G' with respect to $\leq, \leq'$ respectively;
6. $\text{height}(\leq) = \text{height}(\leq') > 0, s = \#G = \#G' > 1$, and there is an enumeration $G_1, \ldots, G_s$ of the components of G and $G'_1, \ldots, G'_s$ of the components of G' such that
   (a) whenever $i \leq j$, $(G_i, \chi, \leq) \sqsubseteq (G_j, \chi, \leq)$ and $(G'_i, \chi', \leq') \sqsubseteq (G'_j, \chi', \leq')$; and
   (b) if for any i, $(G_i, \chi, \leq) \not\sqsubseteq (G'_i, \chi', \leq')$, then there is a $j < i$ such that $(G'_j, \chi', \leq') \not\sqsubseteq (G_j, \chi, \leq)$.

To see that this inductive definition is well-founded, note that the recursive use of $\sqsubseteq$ is on graphs either equipped with an elimination order to degree d of strictly smaller height [clause (5)] or with strictly fewer components [clause (6)]. The next two propositions are aimed at showing that this definition establishes a linear pre-order on coloured graphs with an elimination order that classifies them up to isomorphism

Proposition 6.5 *For any pair* $(G, \chi, \leq)$, $(G', \chi', \leq')$ *of coloured graphs with an elimination order to degree* d, *at least one of* $(G, \chi, \leq) \sqsubseteq (G', \chi', \leq')$ *or* $(G', \chi', \leq') \sqsubseteq (G, \chi, \leq)$ *holds.*

Proof The proof is by induction on the height k of $\leq$ and for a fixed k, by induction on the number of components. If $k = 0$, then either height$(\leq') > 0$ and $(G, \chi, \leq) \sqsubseteq (G', \chi', \leq')$ by clause (1) of the definition, or height$(\leq') = 0$ and at least one of $(G, \chi, \leq) \sqsubseteq (G', \chi', \leq')$ or $(G', \chi', \leq') \sqsubseteq (G, \chi, \leq)$ must hold by clause (2) and the definition of $\sqsubseteq_0$.

Now suppose $k > 0$. If one of the first four conditions of Definition 6.4 is satisfied, we are done. So suppose they fail and suppose $\#G = \#G' = 1$. Then, if v and v' are the minimal elements of G, G' with respect to $\leq, \leq'$, the height of these orders on $G \backslash v$ and $G' \backslash v'$ is strictly less than k. By induction hypothesis, either $(G \backslash v, \chi_v, \leq) \sqsubseteq (G' \backslash v', \chi'_{v'}, \leq')$ or $(G' \backslash v', \chi_{v'}, \leq') \sqsubseteq (G \backslash v, \chi_v, \leq)$, so the claim follows.

Finally, suppose height$(\leq)$ = height$(\leq') > 0$ and $s = \#G = \#G' > 1$ and we argue that in all such cases, clause (6) ensures the result. By the induction hypothesis and the previous case, for any pair of components G_i and G_j of G, one of $(G_i, \chi, \leq) \sqsubseteq (G_j, \chi, \leq)$ or $(G_j, \chi, \leq) \sqsubseteq (G_i, \chi, \leq)$ must hold and similarly for the components of G'. Thus, the components of each can be ordered so that whenever $i \leq j$, $(G_i, \chi, \leq) \sqsubseteq (G_j, \chi, \leq)$ and $(G'_i, \chi', \leq') \sqsubseteq (G'_j, \chi', \leq')$. Also, by the induction hypothesis, we have for each i, either $(G_i, \chi, \leq) \sqsubseteq (G'_i, \chi', \leq')$ or $(G'_i, \chi', \leq') \sqsubseteq (G_i, \chi, \leq)$. If the first case holds for all i, then $(G, \chi, \leq) \sqsubseteq (G', \chi', \leq')$. If the second case holds for all i, then $(G', \chi', \leq') \sqsubseteq (G, \chi, \leq)$. If neither holds for all i, consider the least i such that either $(G_i, \chi, \leq) \not\sqsubseteq (G'_i, \chi', \leq')$ or $(G'_i, \chi', \leq') \not\sqsubseteq (G_i, \chi, \leq)$. In the former case, we have $(G', \chi', \leq') \sqsubseteq (G, \chi, \leq)$ and in the latter $(G_i, \chi, \leq) \sqsubseteq (G'_i, \chi', \leq')$. In all cases, the proposition is established.

This proposition implies that it is always possible to enumerate the components $G_1, \ldots, G_s$ of G in $\sqsubseteq$-order. The next proposition shows that this order is, essentially, unique.

Proposition 6.6 *Let* G, G' *be graphs with colourings* χ *and* χ' *and let* $\leq, \leq'$ *be elimination orders to degree* d *on* G, G' *respectively.*

Then both $(G, \chi, \leq) \sqsubseteq (G', \chi', \leq')$ *and* $(G', \chi', \leq') \sqsubseteq (G, \chi, \leq)$ *if, and only if,* $(G, \chi, \leq) \cong (G', \chi, \leq')$.

Proof Suppose $(G, \chi, \leq) \cong (G', \chi, \leq')$ and suppose, without loss of generality that $(G, \chi, \leq) \sqsubseteq (G', \chi', \leq')$. We aim to show that also $(G', \chi', \leq') \sqsubseteq (G, \chi, \leq)$, and we do this by induction on the height k of $\leq$ and for a fixed k, by cases on the number of components. If $k = 0$ then since $(G, \chi) \cong (G', \chi')$, the conclusion follows from the definition of $\sqsubseteq_0$ and clause (2) of Definition 6.4. If $k > 0$ and $\#G = 1$ then the induction hypothesis and the fact that $(G \backslash v, \chi_v, \leq) \cong (G' \backslash v', \chi'_{v'}, \leq')$ together imply that $(G \backslash v, \chi_v, \leq) \sqsubseteq (G' \backslash v', \chi'_{v'}, \leq')$ and $(G' \backslash v', \chi'_{v'}, \leq') \sqsubseteq (G \backslash v, \chi_v, \leq)$, so the proposition holds by clause (4) of Definition 6.4. Finally, suppose that $k > 0$ and $\#G > 1$ and let $G_1, \ldots, G_s$ enumerate the components of G in $\sqsubseteq$ order. By the isomorphism between $(G, \chi, \leq)$ and $(G', \chi', \leq')$ and the induction hypothesis, it follows that for the corresponding enumeration $G'_1, \ldots, G'_s$ of components of G'

in $\sqsubseteq$ order, we have that $(G_i, \chi, \leq) \cong (G'_i, \chi, \leq')$ for each i. Then by clause (6) of Definition 6.4, both $(G, \chi, \leq) \sqsubseteq (G', \chi', \leq')$ and $(G', \chi', \leq') \sqsubseteq (G, \chi, \leq)$ hold.

For the other direction, suppose $(G, \chi, \leq) \sqsubseteq (G', \chi', \leq')$ and $(G', \chi', \leq') \sqsubseteq (G, \chi, \leq)$ If $\text{height}(\leq) < \text{height}(\leq')$, then $(G', \chi', \leq') \not\sqsubseteq (G, \chi, \leq)$, so assume that $\text{height}(\leq) = \text{height}(\leq') = k$ and we proceed by induction on k. If $k = 0$, we have by clause (2) that $(G, \chi) \sqsubseteq_0 (G', \chi')$ and $(G', \chi') \sqsubseteq_0 (G, \chi)$ which, by definition of $\sqsubseteq_0$ means that $(G, \chi) \cong (G', \chi)$, and the two orders are both trivial.

Suppose then that $k > 0$ and assume first that G and G' are both connected. Then both G, G' have minimal elements v, v' with respect to $\leq, \leq'$. By the induction hypothesis we have $(G \backslash v, \chi_v \leq) \cong (G' \backslash v', \chi_{v'}, \leq')$, so let $\sigma : V(G \backslash v) \to V(G' \backslash v')$ be an isomorphism witnessing this. Extend this to $\hat{\sigma} : V(G) \to V(G')$ by setting $\hat{\sigma}(v) = v'$ and we claim that this is an isomorphism between $(G, \chi, \leq)$ and $(G', \chi', \leq')$. It is clear that $\hat{\sigma}$ takes the order $\leq$ to $\leq'$ since v and v' are minimal for these respective orders. Also, $\hat{\sigma}$ preserves colours since the colourings χ_v and $\chi_{v'}$ refine the colourings χ and χ' respectively and by clause (4) we have $\chi(v) = \chi'(v')$. Finally, we need to verify that $\hat{\sigma}$ preserves the edge relation. Since σ preserves all edges, we only need to verify that $uv \in E(G)$ if, and only if, $\hat{\sigma}(u)v' \in E(G')$. But, this follows from the fact that $\chi_v(u) = \chi'_{v'}(\sigma(u))$.

Now, if G and G' have different number of connected components, then we cannot have both $(G, \chi, \leq) \sqsubseteq (G', \chi', \leq')$ and $(G', \chi', \leq') \sqsubseteq (G, \chi, \leq)$. So, we are left with the case where $\#G = \#G' = s > 1$. Let $G_1 \sqsubseteq \cdots \sqsubseteq G_s$ be the connected components of G and let $G'_1 \sqsubseteq \cdots \sqsubseteq G'_s$ be the connected components of G'. Then by definition of $\sqsubseteq$ we have that both $(G_i, \chi, \leq) \sqsubseteq (G'_i, \chi', \leq')$ and $(G'_i, \chi', \leq') \sqsubseteq (G_i, \chi, \leq)$ for each $i \leq s$. By the argument for connected graphs above, we have $(G_i, \chi, \leq) \cong (G'_i, \chi', \leq')$ for each i and we are done.

Algorithm 1 Recursive construction of minimal elimination order to degree d.

Input: A graph G, a subgraph $C \subseteq G$, and $\ell, d \in \mathbb{N}$.
Output: A $\sqsubseteq$-minimal elimination order to degree d of G, where all the sub-maximal vertices are from C.

```
1: procedure FindMinimalOrder(G, C, ℓ, d)
2:     if #G > 1 then
3:         return ⋃_i FindMinimalOrder(G_i, C[G_i ∩ C], ℓ, d), where the G_i are the connected components
               of G.
4:     else if Δ(G) ≤ d then
5:         return ∅.
6:     else
7:         find the set of potential roots R of a minimal elimination order on C.
8:         for w ∈ R do
9:             colour(u) := colour(u) ∪ {ℓ} for all neighbours u of w
10:            <_w := FindMinimalOrder(G\w, C \ w, ℓ + 1, d)
11:            <_w := <_w ∪{(w, u) | u ∈ V(G)\{w}}.
12:        return <_w that makes (G, <_w) ⊑-minimal.
```

In order to define an algorithm for constructing a minimal elimination order to degree d, we need one further ingredient. For a graph G of tree-depth k, we say that a vertex v is a *potential root* of a minimal elimination order on G if $td(G \backslash v) < k$. We know from Bouland et al. [2] that there is a computable function f such that the

number of potential roots in any graph of tree-depth k is at most $f(k)$. Moreover, the set of all such vertices can be found by an algorithm running in time $f(k)n^3$ where n is the number of vertices in G. This now allows us to define Algorithm 1 which computes a minimal elimination order to degree d. The algorithm is started with the graph G, its d-degree torso C, the level ℓ initially set to 0 and the degree d. It then recursively chooses a potential root and returns the order that is minimal with respect to $\sqsubseteq$.

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

Theorem 6.7 *Graph Canonisation is* FPT *parameterized by elimination distance to bounded degree.*

Proof Suppose we are given a graph G with $|V(G)| = n$. We first compute the d-degree torso C of G in $O(n^4)$ time. Using Algorithm 1 we compute a $\sqsubseteq$-minimal elimination order $<$ to degree d on G. By the analysis from Bouland et al. [2, Theorem 11], this takes time $O(h(k)n^3 log(n))$ for some computable function h. Note for the base case that we can compute the canonical representation of a component Z with $\Delta(Z) \leq d$ by Theorem 2.4 in polynomial time (where the degree of the polynomial depends on d).

Note that the order $\sqsubseteq$ also induces a natural order on the vertices of G and we can thus enumerate the vertices of G in a canonical way in linear time.

Corollary 6.8 *Graph Isomorphism is* FPT *parameterized by elimination distance to bounded degree.*

7 Conclusion

We introduce a new way of parameterizing graphs by their distance to triviality, i.e. by elimination distance. In the particular case of graph canonisation, and thus also graph isomorphism, 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 "trivial" classes. For instance, what can we say about GI when parameterized by elimination distance to planar graphs? Unfortunately 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.

It is easy to see that if a class of graphs C is characterised by a finite set of excluded minors, that the class $\hat{\mathsf{C}}$ of graphs with bounded elimination distance to C is characterised by a finite set of excluded minors as well. An interesting question is whether

we can, given the set of excluded minors for C, compute the excluded minors for $\hat{\mathsf{C}}$ as well?

References

1. Babai, L., Luks, E.M.: *Canonical labeling of graphs*. In: Proceedings of the 15th annual ACM symposium on theory of computing, ACM, pp. 171–183 (1983)
2. Bouland, A., Dawar, A., Kopczyński, E.: On tractable parameterizations of graph isomorphism. In: Parameterized and Exact Computation, pp. 218–230. Springer, Berlin (2012)
3. Diestel, R.: Graph Theory. Springer, Berlin (2000)
4. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer, Berlin (2012)
5. Evdokimov, S., Ponomarenko, I.: Isomorphism of coloured graphs with slowly increasing multiplicity of Jordan blocks. Combinatorica **19**(3), 321–333 (1999)
6. Fellows, M.R., Guo, J., Moser, H., Niedermeier, R.: A generalization of Nemhauser and Trotter's local optimization theorem. J. Comput. Syst. Sci. **77**(6), 1141–1158 (2011)
7. Fellows, M.R., Lokshtanov, D., Misra, N., Rosamond, F.A., Saurabh, S.: Graph layout problems parameterized by vertex cover. In: Proceedings of the 19th international symposium on algorithms and computation, pp. 294–305 (2008)
8. Filotti, I.S., Mayer, J.N.: A polynomial-time algorithm for determining the isomorphism of graphs of fixed genus. In: Proceedings of the 12th annual ACM symposium on theory of computing, ACM, pp. 236–243 (1980)
9. Flum, J., Grohe, M.: Parameterized Complexity Theory. Springer, Berlin (2006)
10. Grohe, M., Marx, D.: Structure theorem and isomorphism test for graphs with excluded topological subgraphs. In: Proceedings of the 44th annual ACM symposium on theory of computing, pp. 173–192 (2012)
11. Guo, J., Hüffner, F., Niedermeier, R.: A structural view on parameterizing problems: distance from triviality. In: Parameterized and Exact Computation, pp. 162–173. Springer, Berlin (2004)
12. Kratsch, S., Schweitzer, P.: Isomorphism for graphs of bounded feedback vertex set number. In: Proceedings of the 12th Scandinavian conference on algorithm theory, pp. 81–92. Springer (2010)
13. Lindell, S.: A logspace algorithm for tree canonization (extended abstract). In: Proceedings of the 24th annual ACM symposium on theory of computing, ACM (1992)
14. Lokshtanov, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Fixed-parameter tractable canonization and isomorphism test for graphs of bounded treewidth. In: 55th IEEE annual symposium on foundations of computer science, pp. 186–195 (2014)
15. Luks, E.M.: Isomorphism of graphs of bounded valence can be tested in polynomial time. J. Comput. Syst. Sci. **25**(1), 42–65 (1982)
16. Miller, G.: Isomorphism testing for graphs of bounded genus. In: Proceedings of the 12th annual ACM symposium on theory of computing, ACM, pp. 218–224 (1980)
17. Niedermeier, R.: Invitation to Fixed-Parameter Algorithms. Oxford University Press, Oxford (2006)
18. Ponomarenko, I.N.: The isomorphism problem for classes of graphs closed under contraction. J. Sov. Math. **55**(2), 1621–1643 (1991)
19. Toda, S.: Computing automorphism groups of chordal graphs whose simplicial components are of small size. IEICE Trans. Inf. Syst. **E89–D**(8), 2388–2401 (2006)
20. Uehara, R., Toda, S., Nagoya, T.: Graph isomorphism completeness for chordal bipartite graphs and strongly chordal graphs. Discrete Appl. Math. **145**(3), 479–482 (2005)
21. Yamazaki, K., Bodlaender, H.L., De Fluiter, B., Thilikos, D.M.: Isomorphism for graphs of bounded distance width. In: Italian/International conference on algorithms and complexity, pp. 276–287. Springer (1997)