TheoretiCS

Isomorphism for Tournaments of Small Twin Width

Received Aug 14, 2024
Revised Nov 12, 2025
Accepted Feb 8, 2026
Published Feb 21, 2026

Key words and phrases
tournament isomorphism, twin width, fixed-parameter tractability, Weisfeiler-Leman algorithm

Martin Grohe[a]
Daniel Neuen[b]

[a] RWTH Aachen University, Aachen, Germany

[b] TU Dresden, Dresden, Germany

ABSTRACT. We prove that isomorphism of tournaments of twin width at most k can be decided in time $k^{O(\log k)}n^{O(1)}$. This implies that the isomorphism problem for classes of tournaments of bounded or moderately growing twin width is in polynomial time. By comparison, there are classes of undirected graphs of bounded twin width that are isomorphism complete, that is, the isomorphism problem for the classes is as hard as the general graph isomorphism problem. Twin width is a graph parameter that has been introduced only recently (Bonnet et al., J. ACM 2022), but has received a lot of attention in structural graph theory since then. On directed graphs, it is functionally smaller than clique width. We prove that on tournaments (but not on general directed graphs) it is also functionally smaller than directed tree width (and thus, the same also holds for cut width and directed path width). Hence, our result implies that tournament isomorphism testing is also fixed-parameter tractable when parameterized by any of these parameters.

Our isomorphism algorithm heavily employs group-theoretic techniques. This seems to be necessary: as a second main result, we show that the combinatorial Weisfeiler-Leman algorithm does not decide isomorphism of tournaments of twin width at most 35 if its dimension is $o(n)$. (Throughout this abstract, n is the order of the input graphs.)

The first author is funded by the European Union (ERC, SymSim, 101054974). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. A preliminary version of this article appeared at ICALP 2024 [29].

Cite as Martin Grohe, Daniel Neuen. Isomorphism for Tournaments of Small Twin Width. TheoretiCS, Volume 5 (2026), Article 4, 1-37.

https://theoretics.episciences.org
DOI 10.46298/theoretics.26.4

1. Introduction

The tournament isomorphism problem (TI) was recognized as a particularly interesting special case of the graph isomorphism problem (GI) early-on. Already in 1983, Babai and Luks [3] proved that TI is solvable in time $n^{O(\log n)}$; it took 33 more years for Babai [2] to prove that the general GI is in quasi-polynomial time. An important fact that makes TI more accessible than GI is that tournaments always have solvable automorphism groups. This is a consequence of the observation that the automorphism groups of tournaments have odd order and the famous Feit-Thompson Theorem [21] stating that all groups of odd order are solvable. However, even Babai's powerful new machinery did not help us to improve the upper bound for TI, as one might have hoped. But TI is not only special from a group-theoretic perspective. Another remarkable result, due to Schweitzer [50], states that TI reduces to the problem of deciding whether a tournament has a nontrivial automorphism; the so-called *rigidity problem*. It is an open question whether the same holds for general graphs.

While there is an extensive literature on GI restricted to classes of graphs (see [30, 45] for recent surveys), remarkably little is known for restrictions of TI. Ponomarenko [47] proved that TI is in polynomial time for tournaments whose automorphism group contains a regular cyclic subgroup, and recently Arvind, Ponomarenko, and Ryabov [1] proved that TI is in polynomial time for edge-colored tournaments where at least one edge color induces a (strongly) connected spanning subgraph of bounded degree (even fixed-parameter tractable when parameterized by the out-degree). While both of these results are very interesting from a technical perspective, they consider classes of tournaments that would hardly be called natural from a graph-theoretic point of view. Natural graph parameters that have played a central role in the structural theory of tournaments developed by Chudnovsky, Seymour and others [14, 15, 16, 17, 22] are *cut width* and *path width*. The more recent theory of structural sparsity [23, 24, 41, 42] highlights *clique width* and *twin width*. Here twin width is the key parameter. Not only is it functionally smaller than the other parameters, which means that if cut width, path width, or clique width is bounded, then twin width is bounded as well, it is also known [26] that a class of tournaments has bounded twin width if and only if it has a property known as *monadic dependence (NIP)*. Dependence is a key property studied in classical model theory. A class of graphs is monadically dependent if and only if all set systems definable in this class by a first-order transduction have bounded VC dimension. This property seems to characterize precisely the graph classes that are regarded as structurally sparse. Since twin width of graphs and binary relational structures has been introduced in [12], it received a lot of attention in algorithmic structural graph theory [6, 7, 8, 9, 10, 11, 24, 25, 26, 34, 53]. (We defer the somewhat unwieldy definition of twin width to Section 2.3.) Our main result states that tournament isomorphism is fixed-parameter tractable when parameterized by twin width.

THEOREM 1.1. *The isomorphism problem for tournaments of twin width at most k can be solved in time $k^{O(\log k)} \cdot n^{O(1)}$.*

Interestingly, isomorphism testing for undirected graphs of bounded twin width at most 4 is as hard as the general GI. This follows easily from the fact that a $\lceil 2 \log n \rceil$-subdivision of every graph with n vertices has twin width at most 4 [4]. Once more, this demonstrates the special role of the tournament isomorphism problem, though here the reason is not group-theoretic, but purely combinatorial.

Note that the dependence on the twin width k of the algorithm in Theorem 1.1 is subexponential, so our result implies that TI is in polynomial time even for tournaments of twin width $2^{O(\sqrt{\log n})}$. Since twin width is functionally smaller than clique width, our result implies that TI is also fixed-parameter tractable when parameterized by clique width. Additionally, we prove (Corollary 6.5) that the twin width of a tournament is functionally smaller than its *directed tree width*, a graph parameter originally introduced in [36]. Since the directed tree width of every directed graph is smaller than its cut width or directed path width, the same also holds for these two parameters. Hence, TI is fixed-parameter tractable also when parameterized by directed tree width, directed path width or cut width. To the best of our knowledge, this was not known for any of these parameters. The fact that twin width is functionally smaller than directed tree width, directed path width and cut width on tournaments is interesting in its own right, because this result does not extend to general directed graphs (for any of the three parameters).

Our proof of Theorem 1.1 heavily relies on group-theoretic techniques. In a nutshell, we show that bounded twin width allows us to cover a tournament by a sequence of directed graphs that have a property resembling bounded degree sufficiently closely to apply a group-theoretic machinery going back to Luks [38] and developed to great depth since then (see, e.g., [2, 3, 31, 40, 44]). Specifically, we generalize arguments that have been introduced by Arvind et al. [1] for TI on edge-colored tournaments where at least one edge color induces a spanning subgraph of bounded out-degree.

Yet one may wonder if this heavy machinery is even needed to prove our theorem, in particular in view of the fact that on many natural graph classes, including, for example, undirected graphs of bounded clique width [28], the purely combinatorial Weisfeiler-Leman algorithm is sufficient to decide isomorphism (see, e.g., [27, 37]). We prove that this is not the case for tournaments of bounded twin width.

THEOREM 1.2. *For every $k \geq 2$ there are non-isomorphic tournaments T_k and T'_k of order $|V(T_k)| = |V(T'_k)| = O(k)$ and twin width at most 35 that are not distinguished by the k-dimensional Weisfeiler-Leman algorithm.*

We remark that it was known before that the Weisfeiler-Leman algorithm fails to decide tournament isomorphism. Indeed, Dawar and Kopczynski (unpublished) proved that for every

$k \geq 2$ there are non-isomorphic tournaments U_k and U'_k of order $|V(U_k)| = |V(U'_k)| = O(k)$ that are not distinguished by the k-dimensional Weisfeiler-Leman algorithm. Theorem 1.2 strengthens this result by constructing tournaments where additionally the twin width is bounded by a fixed constant.

The paper is organized as follows. After introducing the necessary preliminaries in Section 2, Theorem 1.1 is proved in Sections 3 and 4. First, we give our main combinatorial arguments in Section 3. After that, the mainly group-theoretic isomorphism algorithm of Theorem 1.1 is presented in Section 4. Theorem 1.2 is proved in Section 5. Finally, in Section 6 we compare twin width to other width measures for directed graphs.

2. Preliminaries

2.1 Graphs

Graphs in this paper are usually directed. We often emphasize this by calling them "digraphs". However, when we make general statements about graphs, this refers to directed graphs and includes undirected graphs a special case (directed graphs with a symmetric edge relation). We denote the vertex set of a graph G by $V(G)$ and the edge relation by $E(G)$. The vertex set $V(G)$ is always finite and non-empty. The edge relation is always anti-reflexive, that is, graphs are loop-free, and there are no parallel edges. For a digraph G and a vertex $v \in V(G)$, we denote the set of out-neighbors and in-neighbors of v by $N_+(v)$ and $N_-(v)$, respectively. Also, the *out-degree* and *in-degree* of v are denoted by $\deg_+(v) := |N_+(v)|$ and $\deg_-(v) := |N_-(v)|$, respectively. Furthermore, $E_+(v)$ and $E_-(v)$ denote the set of outgoing and incoming edges into v, respectively. For $X \subseteq V(G)$, we write $G[X]$ to denote the subgraph of G induced on X. For two sets $X, Y \subseteq V(G)$ we write $E_G(X,Y) := \{(v,w) \in E(G) \mid v \in X, w \in Y\}$ to denote the set of directed edges from X to Y.

Let G be an undirected graph. A directed graph $\vec{G}$ is an *orientation* of G if, for every undirected edge $\{v,w\} \in E(G)$, exactly one of (v,w) and (w,v) is an edge of $\vec{G}$, and there are no other edges present in $\vec{G}$. A *tournament* is an orientation of a complete graph.

A tournament T is *regular* if $\deg_+(v) = \deg_+(w)$ for all $v, w \in V(T)$. In this case, $\deg_+(v) = \deg_-(v) = \frac{|V(G)|-1}{2}$ for all $v \in V(T)$. This implies that every regular tournament has an odd number of vertices.

Let G_1, G_2 be two graphs. An *isomorphism* from G_1 to G_2 is a bijection $\varphi\colon V(G_1) \to V(G_2)$ such that $(v,w) \in E(G_1)$ if and only if $(\varphi(v), \varphi(w)) \in E(G_2)$ for all $v, w \in V(G_1)$. We write $\varphi\colon G_1) \cong G_2$ to denote that φ is an isomorphism from G_1 to G_2. Also, $\mathrm{Iso}(G_1, G_2)$ denotes the set of all isomorphisms from G_1 to G_2. The graphs G_1 and G_2 are *isomorphic* if $\mathrm{Iso}(G_1, G_2) \neq \emptyset$. The *automorphism group* of G_1 is $\mathrm{Aut}(G_1) := \mathrm{Iso}(G_1, G_1)$.

An *arc coloring* of a digraph G is a mapping $\lambda\colon (E(G) \cup \{(v,v) \mid v \in V(G)\}) \to C$ for some set C of "colors". An *arc-colored graph* is a triple $G = (V, E, \lambda)$, where (V, E) is a graph an λ an arc coloring of (V, E). Isomorphisms between arc-colored graphs are required to preserve the coloring.

2.2 Partitions and Colorings

Let S be a finite set. A *partition* of S is a set $\mathcal{P} \subseteq 2^S$ whose elements we refer to as *parts*, such that any two parts are mutually disjoint, and the union of all parts is S. A partition $\mathcal{P}$ *refines* another partition $\mathcal{Q}$, denoted by $\mathcal{P} \preceq \mathcal{Q}$, if for all $P \in \mathcal{P}$ there is some $Q \in \mathcal{Q}$ such that $P \subseteq Q$. We say a partition $\mathcal{P}$ is *trivial* if $|\mathcal{P}| = 1$, which means that the only part is S, and it is *discrete* if $|P| = 1$ for all $P \in \mathcal{P}$.

Every mapping $\chi \colon S \to C$, for some set C, induces a partition $\mathcal{P}_\chi$ of S into the sets $\chi^{-1}(c)$ for all c in the range of χ. In this context, we think of χ as a "coloring" of S, the elements $c \in C$ as "colors", and the parts $\chi^{-1}(c)$ of the partition as "color classes". If $\chi' \colon S \to C'$ is another coloring, then we say that χ *refines* χ', denoted by $\chi \preceq \chi'$, if $\mathcal{P}_\chi \preceq \mathcal{P}_{\chi'}$. The colorings are *equivalent* (we write $\chi \equiv \chi'$) if $\chi \preceq \chi'$ and $\chi \preceq \chi'$, i.e., $\mathcal{P}_\chi = \mathcal{P}_{\chi'}$.

2.3 Twin Width

Twin width [12] is defined for binary relational structures, which in this paper are mostly directed graphs. We need one distinguished binary relation symbol E_{red} that plays a special role in the definition of twin width. Following [12], we refer to elements of E_{red} as *red edges*. For every structure A, we assume the relation $E_{\text{red}}(A)$ to be symmetric and anti-reflexive, that is, the edge relation of an undirected graph, and we refer to the maximum degree of this graph as the *red degree* of A. If $E_{\text{red}}(A)$ is not explicitly defined, we assume $E_{\text{red}}(A) = \emptyset$ (and the red degree of A is 0).

Let $A = (V(A), R_1(A), \dots, R_k(A))$ be a binary relational structure, where $V(A)$ is a non-empty finite vertex set and $R_i(A) \subseteq (V(A))^2$ are binary relations on $V(A)$ (possibly, $R_i = E_{\text{red}}$ for some $i \in [k]$). We call a pair (X, Y) of disjoint subsets of $V(A)$ *homogeneous* if for all $x, x' \in X$, and all $y, y' \in Y$ it holds that

(i) $(x, y) \in R_i(A) \Leftrightarrow (x', y') \in R_i(A)$ and $(y, x) \in R_i(A) \Leftrightarrow (y', x') \in R_i(A)$ for all $i \in [k]$, and

(ii) $(x, y) \notin E_{\text{red}}(A)$ and $(y, x) \notin E_{\text{red}}(A)$.

For a partition $\mathcal{P}$ of $V(A)$, we define $A/\mathcal{P}$ to be the structure with vertex set $V(A/\mathcal{P}) := \mathcal{P}$ and relations

$$R_i(A/\mathcal{P}) := \left\{(X, Y) \in \mathcal{P}^2 \;\middle|\; (X, Y) \text{ is homogeneous and } X \times Y \subseteq R_i(A)\right\}$$

for all $R_i \neq E_{\text{red}}$, and

$$E_{\text{red}}(A/\mathcal{P}) := \left\{(X, Y) \in \mathcal{P}^2 \;\middle|\; (X, Y) \text{ is not homogeneous and } X \neq Y\right\}.$$

A *contraction sequence for A* is a sequence of partitions $\mathcal{P}_1, \dots, \mathcal{P}_n$ of $V(A)$ such that $\mathcal{P}_1 = \{\{v\} \mid v \in V(A)\}$ is the discrete partition, $\mathcal{P}_n = \{V(A)\}$ is the trivial partition, and for every $i \in [n-1]$ the partition $\mathcal{P}_{i+1}$ is obtained from $\mathcal{P}_i$ by merging two parts, i.e., there are distinct $P, P' \in \mathcal{P}_i$ such that $\mathcal{P}_{i+1} = \{P \cup P'\} \cup (\mathcal{P}_i \setminus \{P, P'\})$. The *width* of a contraction sequence $\mathcal{P}_1, \dots, \mathcal{P}_n$ of A is the minimum k such that for every $i \in [n]$ the structure $A/\mathcal{P}_i$ has red degree at most k. The *twin width* of A, denoted by $\text{tww}(A)$, is the minimum $k \geq 0$ such that A has a contraction sequence of width k.

Note that red edges are introduced as we contract parts of the partitions. However, the structure A we start with may already have red edges, which then have direct impact on its twin width. In particular, the twin width of a graph G may be smaller than the twin width of the structure G^{red} obtained from G by coloring all edges red. This fact is used later.

We also remark that for our isomorphism algorithms, we never have to compute a contraction sequence of minimum width or the twin width.

We state two simple lemmas on basic properties of twin width.

LEMMA 2.1 ([12]). *Let A be a binary relational structure and $X \subseteq V(A)$. Then* $\text{tww}(A[X]) \leq \text{tww}(A)$.

LEMMA 2.2. *Let A be a structure over the vocabulary τ. Then there is a linear order $<$ on $V(A)$ such that* $\text{tww}(A, <) = \text{tww}(A)$.

PROOF. Let $\mathcal{P}_1, \dots, \mathcal{P}_n$ be a contraction sequence for A. Note that for every $i \in [n]$ the partition $\mathcal{P}_i$ has exactly $n + 1 - i$ parts $P_{i,1}, \dots, P_{i,n+1-i}$. We may choose the indices in such a way that if $\mathcal{P}_i$ is obtained from $\mathcal{P}_{i-1}$ by merging the parts P_{i-1,j_1} and P_{i-1,j_2} to $P_{i,j} = P_{i-1,j_1} \cup P_{i-1,j_2}$ then $j_1 = j, j_2 = j + 1, P_{i,k} = P_{i-1,k}$ for $k < j$ and $P_{i,k} = P_{i-1,k+1}$ for $k > j$. Let $<$ be the linear order induced by the partition $\mathcal{P}_1$ (whose parts have size 1). Then for all $i \in [n]$, the parts of $\mathcal{P}_i$ are consecutive intervals of $<$, which means that all pairs $(P_{i,j}, P_{i,k})$ of distinct parts of $\mathcal{P}_i$ are homogeneous for the relation $<$. This implies $\text{tww}(A, <) = \text{tww}(A)$. ■

2.4 Weisfeiler-Leman

In this section, we describe the k-dimensional Weisfeiler-Leman algorithm (k-WL). The algorithm has been originally introduced in its 2-dimensional form by Weisfeiler and Leman [55] (see also [54]). The k-dimensional version, coloring k-tuples, was introduced later by Babai and Mathon (see [13]).

Fix $k \geq 2$, and let G be a graph. For $i \geq 0$, we describe the coloring $\chi_{(i)}^{k,G}$ of $(V(G))^k$ computed in the i-th iteration of k-WL. For $i = 0$, each tuple is colored with the isomorphism type of the underlying ordered induced subgraph. So if H is another graph and $\bar{v} = (v_1, \dots, v_k) \in (V(G))^k$, $\bar{w} = (w_1, \dots, w_k) \in (V(H))^k$, then $\chi_{(0)}^{k,G}(\bar{v}) = \chi_{(0)}^{k,H}(\bar{w})$ if and only if, for all $i, j \in [k]$, it holds that

$v_i = v_j \Leftrightarrow w_i = w_j$ and $(v_i, v_j) \in E(G) \Leftrightarrow (w_i, w_j) \in E(H)$. If G and H are arc-colored, then the colors are also taken into account.

Now let $i \geq 0$. For $\bar{v} = (v_1, \dots, v_k)$ we define

$$\chi^{k,G}_{(i+1)}(\bar{v}) := \left(\chi^{k,G}_{(i)}(\bar{v}), \mathcal{M}_i(\bar{v})\right)$$

where

$$\mathcal{M}_i(\bar{v}) := \left\{\!\!\left\{(\chi^{k,G}_{(i)}(\bar{v}[w/1]), \dots, \chi^{k,G}_{(i)}(\bar{v}[w/k])) \;\middle|\; w \in V(G)\right\}\!\!\right\}$$

and $\bar{v}[w/i] := (v_1, \dots, v_{i-1}, w, v_{i+1}, \dots, v_k)$ is the tuple obtained from $\bar{v}$ b replacing the i-th entry by w (and $\{\{\dots\}\}$ denotes a multiset).

Clearly, $\chi^{k,G}_{(i+1)} \preceq \chi^{k,G}_{(i)}$ for all $i \geq 0$. So there is a unique minimal $i_\infty \geq 0$ such that $\chi^{k,G}_{(i_\infty+1)} \equiv \chi^{k,G}_{(i_\infty)}$ and we write $\chi^{k,G} := \chi^{k,G}_{(i_\infty)}$ to denote the corresponding coloring.

The k-dimensional Weisfeiler-Leman algorithm takes as input a (possibly colored) graph G and outputs (a coloring that is equivalent to) $\chi^{k,G}$. This can be done in time $O(k^2 n^{k+1} \log n)$ [35].

Let H be a second graph. The k-dimensional Weisfeiler-Leman algorithm *distinguishes* G and H if there is a color $c \in C$ such that

$$\left|\left\{\bar{v} \in (V(G))^k \;\middle|\; \chi^{k,G}(\bar{v}) = c\right\}\right| \neq \left|\left\{\bar{w} \in (V(H))^k \;\middle|\; \chi^{k,H}(\bar{w}) = c\right\}\right|.$$

We write $G \simeq_k H$ to denote that k-WL does not distinguish between G and H.

A graph G is *k-WL-homogeneous* if $\chi^{k,G}(v, \dots, v) = \chi^{k,G}(w, \dots, w)$ for all $v, w \in V(G)$.

2.5 Group Theory

For a general background on group theory we refer to [49], whereas background on permutation groups can be found in [20]. Also, basics facts on algorithms for permutation groups are given in [51].

Basics for Permutation Groups. A *permutation group* acting on a set Ω is a subgroup $\Gamma \leq \mathrm{Sym}(\Omega)$ of the symmetric group. The size of the permutation domain Ω is called the *degree* of Γ. If $\Omega = [n] := \{1, \dots, n\}$, then we also write S_n instead of $\mathrm{Sym}(\Omega)$. For $A \subseteq \Omega$ and $\gamma \in \Gamma$ let $\gamma(A) := \{\gamma(\alpha) \mid \alpha \in A\}$. The set A is Γ-*invariant* if $\gamma(A) = A$ for all $\gamma \in \Gamma$.

Let $\theta \colon \Omega \to \Omega'$ be a bijection. We write $\Gamma\theta := \{\gamma\theta \mid \gamma \in \Gamma\}$ for the set of bijections from Ω to Ω' obtained from concatenating a permutation from Γ and θ. Note that $(\gamma\theta)(\alpha) = \theta(\gamma(\alpha))$ for all $\alpha \in \Omega$.

A set $S \subseteq \Gamma$ is a *generating set* for Γ if for every $\gamma \in \Gamma$ there are $\delta_1, \dots, \delta_k \in S$ such that $\gamma = \delta_1 \dots \delta_k$. In order to perform computational tasks for permutation groups efficiently the groups are represented by generating sets of small size (i.e., polynomial in the size of the permutation domain). Indeed, most algorithms are based on so-called strong generating sets, which can be chosen of size quadratic in the size of the permutation domain of the group and can be computed in polynomial time given an arbitrary generating set (see, e.g., [51]).

Group-Theoretic Methods for Isomorphism Testing. In this work, we shall be interested in a particular subclass of permutation groups. Let Γ be a group and let $\gamma, \delta \in \Gamma$. The *commutator* of γ and δ is $[\gamma, \delta] := \gamma^{-1}\delta^{-1}\gamma\delta$. The *commutator subgroup* $[\Gamma, \Gamma]$ of Γ is the unique subgroup of Γ generated by all commutators $[\gamma, \delta]$ for $\gamma, \delta \in \Gamma$. Note that $[\Gamma, \Gamma]$ is a normal subgroup of Γ. The *derived series of* Γ is the sequence of subgroups $\Gamma^{(0)} \trianglerighteq \Gamma^{(1)} \trianglerighteq \Gamma^{(2)} \trianglerighteq \dots$ where $\Gamma^{(0)} := \Gamma$ and $\Gamma^{(i+1)} := [\Gamma^{(i)}, \Gamma^{(i)}]$ for all $i \geq 0$. A group Γ is *solvable* if there is some $i \geq 0$ such that $\Gamma^{(i)}$ is the trivial group (i.e., it only contains the identity element).

By the Feit-Thompson Theorem every group of odd order is solvable. Also, for every tournament T, the automorphism group $\text{Aut}(T)$ has odd order. Indeed, $\text{Aut}(T)$ cannot contain an involution (a permutation of order 2), since every involution swaps some pair v, w of distinct vertices and exactly one of $(v, w), (w, v)$ is an edge of T. Together, we obtain the following:

THEOREM 2.3. *Let T be a tournament. Then* $\text{Aut}(T)$ *is solvable.*

Next, we state several basic group-theoretic algorithms for isomorphism testing.

THEOREM 2.4 ([3, Theorem 4.1]). *There is an algorithm that, given two tournaments T_1 and T_2, computes* $\text{Iso}(T_1, T_2)$ *in time $n^{O(\log n)}$.*

Note that $\text{Iso}(T_1, T_2)$ may be of size exponential in the number of vertices of T_1 and T_2. However, if T_1 and T_2 are isomorphic (i.e., $\text{Iso}(T_1, T_2) \neq \emptyset$), we have $\text{Iso}(T_1, T_2) = \text{Aut}(T_1)\varphi$ where $\varphi \in \text{Iso}(T_1, T_2)$ is an arbitrary isomorphism from T_1 to T_2. Hence, the set $\text{Iso}(T_1, T_2)$ can be represented by a generating set for $\text{Aut}(T_1)$ of size polynomial in $|V(T_1)|$ and a single element $\varphi \in \text{Iso}(T_1, T_2)$. Let us stress at this point that all isomorphism sets computed by the various algorithms discussed in this work are represented in this way.

Let G_1 and G_2 be two (colored) directed graphs. Also let $\Gamma \leq \text{Sym}(V(G_1))$ be a permutation group and let $\theta\colon V(G_1) \to V(G_2)$ be a bijection. We define

$$\text{Iso}_{\Gamma\theta}(G_1, G_2) := \text{Iso}(G_1, G_2) \cap \Gamma\theta = \{\varphi \in \Gamma\theta \mid \varphi\colon G_1 \cong G_2\}$$

and $\text{Aut}_\Gamma(G_1) := \text{Iso}_\Gamma(G_1, G_1)$. Note that $\text{Aut}_\Gamma(G_1) \leq \Gamma$ and moreover, if $\text{Iso}_{\Gamma\theta}(G_1, G_2) \neq \emptyset$, then $\text{Iso}_{\Gamma\theta}(G_1, G_2) = \text{Aut}_\Gamma(G_1)\varphi$ where $\varphi \in \text{Iso}_{\Gamma\theta}(G_1, G_2)$ is an arbitrary isomorphism from G_1 to G_2.

THEOREM 2.5 ([3, Corollary 3.6]). *Let $G_1 = (V_1, E_1, \lambda_1)$ and $G_1 = (V_2, E_2, \lambda_2)$ be two arc-colored directed graphs. Also let $\Gamma \leq \text{Sym}(V_1)$ be a solvable group and $\theta\colon V_1 \to V_2$ a bijection. Then* $\text{Iso}_{\Gamma\theta}(G_1, G_2)$ *can be computed in polynomial time.*

A *hypergraph* is a pair $\mathcal{H} = (V, \mathcal{E})$ where V is a finite, non-empty set of vertices and $\mathcal{E} \subseteq 2^V$ is a subset of the powerset of V. Two hypergraphs $\mathcal{H}_1 = (V_1, \mathcal{E}_1)$ and $\mathcal{H}_2 = (V_2, \mathcal{E}_2)$ are *isomorphic* if there is a bijection $\varphi\colon V_1 \to V_2$ such that $E \in \mathcal{E}_1$ if and only if $\varphi(E) \in \mathcal{E}_2$ for all subsets $E \subseteq V$. We write $\varphi\colon \mathcal{H}_1 \cong \mathcal{H}_2$ to denote that φ is an *isomorphism* from $\mathcal{H}_1$ to $\mathcal{H}_2$. As before, we write $\mathrm{Iso}(\mathcal{H}_1, \mathcal{H}_2)$ to denote the set of all isomorphisms from $\mathcal{H}_1$ to $\mathcal{H}_2$. Also, for a permutation group $\Gamma \leq \mathrm{Sym}(V_1)$ and a bijection $\theta\colon V(G_1) \to V(G_2)$, we define

$$\mathrm{Iso}_{\Gamma\theta}(\mathcal{H}_1, \mathcal{H}_2) := \mathrm{Iso}(\mathcal{H}_1, \mathcal{H}_2) \cap \Gamma\theta = \{\varphi \in \Gamma\theta \mid \varphi\colon \mathcal{H}_1 \cong \mathcal{H}_2\}$$

and $\mathrm{Aut}_\Gamma(\mathcal{H}_1) = \mathrm{Iso}_\Gamma(\mathcal{H}_1, \mathcal{H}_1)$.

THEOREM 2.6 ([40]). *Let $\mathcal{H}_1 = (V_1, \mathcal{E}_1)$ and $\mathcal{H}_2 = (V_2, \mathcal{E}_2)$ be two hypergraphs. Also let $\Gamma \leq \mathrm{Sym}(V_1)$ be a solvable group and $\theta\colon V_1 \to V_2$ a bijection. Then $\mathrm{Iso}_{\Gamma\theta}(\mathcal{H}_1, \mathcal{H}_2)$ can be computed in polynomial time.*

Wreath Products. Finally, we describe wreath products of groups which repeatedly appear within the main algorithm of this paper. For simplicity, we focus on the application cases appearing in this work.

Let $G_1, \dots, G_\ell$ be directed graphs with pairwise disjoint vertex sets $V_1, \dots, V_\ell$. Also assume that $G_i \cong G_j$ for all $i, j \in [\ell]$. Then

$$\mathrm{Aut}(G_j) = \varphi^{-1}\,\mathrm{Aut}(G_i)\varphi := \{\varphi^{-1}\gamma\varphi \mid \gamma \in \mathrm{Aut}(G_i)\}$$

for all $\varphi \in \mathrm{Iso}(G_i, G_j)$. Now let $\Delta \leq S_\ell$ be a permutation group with domain $\{1, \dots, \ell\}$. Also let $V := V_1 \cup \dots \cup V_\ell$. We define $\Gamma \leq \mathrm{Sym}(V)$ to be the permutation group containing all elements $\gamma \in \mathrm{Sym}(V)$ such that there are $\delta \in \Delta$ and, for each $i \in [\ell]$, an isomorphism $\varphi_i \in \mathrm{Iso}(G_i, G_{\delta(i)})$ such that for all $v \in V$ it holds that

$$\gamma(v) = \varphi_i(v)$$

where $i \in [\ell]$ is the unique index such that $v \in V_i$. The group Γ is the *wreath product* of $\mathrm{Aut}(G_1)$ by Δ. Observe that Γ is solvable if and only if Δ and $\mathrm{Aut}(G_1)$ are solvable.

Now let $S_1, \dots, S_\ell$ be generating sets for $\mathrm{Aut}(G_1), \dots, \mathrm{Aut}(G_\ell)$, respectively. Also let S_Δ be a generating set for Δ. Then we obtain a generating set S_Γ as follows. For every $\gamma_i \in S_i$ we define γ_i^* to be the extension of γ_i to V which fixes all elements outside of V_i. More formally, $\gamma_i^*(v) = \gamma_i(v)$ for every $v \in V_i$, and $\gamma_i^*(v) = v$ for every $v \in V \setminus V_i$. Let $S_i^* := \{\gamma_i^* \mid \gamma_i \in S_i\}$.

Additionally, for every $\delta \in S_\Delta$, we fix an element $\varphi_i \in \mathrm{Iso}(G_i, G_{\delta(i)})$ for every $i \in [\ell]$. We set $\delta^*(v) := \varphi_i(v)$ for every $v \in V$ where $i \in [\ell]$ is the unique index such that $v \in V_i$. Let $S_\Delta^* := \{\delta^* \mid \delta \in S_\Delta\}$. Then

$$S_\Gamma := S_1^* \cup \dots \cup S_\ell^* \cup S_\Delta^*$$

is a generating of Γ. Note that S_Γ can be computed in polynomial time given the generating sets $S_1, \dots, S_\ell$ and S_Δ.

3. Small Degree Partition Sequences

In the following, we design an isomorphism test for tournaments of twin width k which runs in time $k^{O(\log k)}n^{O(1)}$. On a high level, the algorithm essentially proceeds in three phases. First, we use well-established group-theoretic methods going back to [3, 38] to reduce to the case where both input tournaments are 2-WL-homogeneous (without increasing the twin width). In the second step, we identify a substructure of an input tournament T (that is 2-WL-homogeneous) that has some kind of bounded-degree property. More concretely, we apply the 2-dimensional Weisfeiler-Leman algorithm and compute a sequence of colors $c_1, \dots, c_\ell$ in the image of the 2-WL coloring $\chi^{2,T}$ so that the subgraph induced by all edges with a color from $c_1, \dots, c_\ell$ has a certain type of bounded-degree property. After that, we rely on the computed bounded-degree structure to determine isomorphisms based on the group-theoretic graph isomorphism machinery. Similar tools have also been used in [1] to solve isomorphism of k-spanning tournaments. However, as we shall see below, the bounded-degree property guaranteed by the second step is weaker than the notion of k-spanning tournaments, which requires us to further extend the methods from [1].

In this section, we implement the second phase and prove the key combinatorial lemma (Lemma 3.6) underlying our isomorphism algorithm. Our arguments rely on the notion of *mixed neighbors* for a pair of vertices. For a pair $v, w \in V(T)$ of vertices we let

$$M(v,w) := \Big(N_-(v) \cap N_+(w)\Big) \cup \Big(N_+(v) \cap N_-(w)\Big). \tag{1}$$

We call the elements of $M(v,w)$ the *mixed neighbors* of (v,w), and we call $\mathrm{md}(v,w) := |M(v,w)|$ the *mixed degree* of (v,w). The following simple observation links the mixed degree to twin width.

OBSERVATION 3.1. *There is an edge* $(v,w) \in E(T)$ *such that* $\mathrm{md}(v,w) \leq \mathrm{tww}(T)$.

PROOF. Let $k := \mathrm{tww}(T)$ and let $\mathcal{P}_1, \dots, \mathcal{P}_n$ be a contraction sequence of T of width k. Let $\{v,w\}$ be the unique 2-element part in $\mathcal{P}_2$. Then $\mathrm{md}(v,w) = \mathrm{md}(w,v) \leq k$, and either $(v,w) \in E(T)$ or $(w,v) \in E(T)$. ■

In the following, let G_T be the directed graph with vertex set $V(G_T) := V(T)$ and edge set $E(G_T) := \{(v,w) \in E(T) \mid \mathrm{md}(v,w) \leq \mathrm{tww}(T)\}$. The next lemma implies that G_T has maximum out-degree at most $2 \cdot \mathrm{tww}(T) + 1$.

LEMMA 3.2. *Suppose* $k \geq 1$. *Let T be a tournament and let* $v \in V(T)$. *Also let*

$$W := \{w \in N_+(v) \mid \mathrm{md}(v,w) \leq k\}.$$

Then $|W| \leq 2k+1$.

PROOF. Let $\ell := |W|$. The induced subtournament $T[W]$ has a vertex w of in-degree at least $(\ell-1)/2$. Since $\mathrm{md}(v,w) \leq k$ and $(v,w') \in E(T)$ for all $w' \in W$, we have

$$|\{w' \in W \mid (w',w) \in E(T)\}| \leq k.$$

Thus $\frac{\ell-1}{2} \leq k$, which implies that $|W| = \ell \leq 2k+1$. ■

So G_T is a subgraph of T of maximum out-degree $d := 2 \cdot \mathrm{tww}(T)+1$. We remark that similar arguments also show that G_T has maximum in-degree at most d (technically, this property is not required by our algorithm, but it is helpful for the following explanations). Now, first suppose that G_T is strongly connected. Then the edges of G_T define a (strongly) connected spanning subgraph of maximum degree $2d$ (in-degree plus out-degree). In this situation, we can directly use the algorithm from [1] to test isomorphism in time $d^{O(\log d)}n^{O(1)}$.

So suppose G_T is not strongly connected. If T is 2-WL-homogeneous, then G_T is also not weakly connected (i.e., the undirected version of G_T is not connected); see Lemma 3.5. In this case, the basic idea is to identify further edges to be added to decrease the number of connected components while keeping some kind of bounded-degree property.

In the following, let $\mathcal{Q}$ be a partition of $V(T)$ that is non-trivial, that is, has at least two parts. The reader is encouraged to think of $\mathcal{Q}$ as the partition into the (weakly) connected components of G_T, but the following results hold for any non-trivial partition $\mathcal{Q}$. We call an edge $(v,v') \in E(T)$ *cross-cluster with respect to* $\mathcal{Q}$ if it connects distinct $Q,Q' \in \mathcal{Q}$. For a cross-cluster edge (v,v') with $Q \ni v, Q' \ni v'$, we let

$$\mathcal{M}_{\mathcal{Q}}(v,v') := \left\{Q'' \in \mathcal{Q} \setminus \{Q,Q'\} \;\middle|\; Q'' \cap M(v,v') \neq \emptyset\right\}$$

and $\mathrm{md}_{\mathcal{Q}}(v,v') := |\mathcal{M}_{\mathcal{Q}}(v,v')|$.

The next two lemmas generalize Observation 3.1 and Lemma 3.2.

LEMMA 3.3. *Let T be a tournament and suppose $\mathcal{Q}$ is a non-trivial partition of $V(T)$. Then there is a cross-cluster edge $(v,w) \in E(T)$ such that $\mathrm{md}_{\mathcal{Q}}(v,w) \leq \mathrm{tww}(T)$.*

PROOF. Let $k := \mathrm{tww}(T)$ and let $\mathcal{P}_1,\dots,\mathcal{P}_n$ be a contraction sequence of T of width k. Note that $\mathcal{P}_1$ refines $\mathcal{Q}$ and $\mathcal{P}_n$ does not refine $\mathcal{Q}$, because $\mathcal{Q}$ is nontrivial. Let $i \geq 1$ be minimal such that $\mathcal{P}_{i+1}$ does not refine $\mathcal{Q}$.

Let $P,P' \in \mathcal{P}_i$ denote the parts merged in the step from $\mathcal{P}_i$ to $\mathcal{P}_{i+1}$. Since $\mathcal{P}_i$ refines $\mathcal{Q}$, there are $Q,Q' \in \mathcal{Q}$ such that $P \subseteq Q$ and $P' \subseteq Q'$. Moreover, $Q \neq Q'$, because $\mathcal{P}_{i+1}$ does not refine $\mathcal{Q}$. We pick arbitrary elements $v \in P$ and $w \in P'$ such that $(v,w) \in E(T)$ (if $(w,v) \in E(T)$, we swap the roles of P and P'). Then (v,w) is a cross-cluster edge with respect to $\mathcal{Q}$.

Let $P_1,\dots,P_{k'}$ be a list of all $P'' \in \mathcal{P}_{i+1} \setminus \{P \cup P'\}$ such that the pair $(P \cup P', P'')$ is not homogeneous. Then $k' \leq k$ by the definition of twin width. Since $\mathcal{P}_i \setminus \{P,P'\} = \mathcal{P}_{i+1} \setminus \{P \cup P'\}$ and $\mathcal{P}_i$ refines $\mathcal{Q}$, there are $Q_1,\dots,Q_{k'} \in \mathcal{Q}$ such that $P_i \subseteq Q_i$ for all $i \in [k']$.

Now let $Q'' \in \mathcal{Q} \setminus \{Q, Q', Q_1, \dots, Q_{k'}\}$. Suppose for contradiction that $Q'' \cap M(v,w) \neq \emptyset$, and pick an element $w' \in Q'' \cap M(v,w)$. Then there is a $P'' \in \mathcal{P}_i \setminus \{P, P', P_1, \dots, P_{k'}\} = \mathcal{P}_{i+1} \setminus \{P \cup P', P_1, \dots, P_{k'}\}$ such that $w' \in P''$. But then the pair $(P \cup P', P'')$ is not homogeneous, which is a contradiction. So $Q'' \cap M(v,w) = \emptyset$. This implies that $\mathcal{M}_{\mathcal{Q}}(v,w) \subseteq \{Q_1, \dots, Q_{k'}\}$. In particular, $\mathrm{md}_{\mathcal{Q}}(v,w) \leq k' \leq k$. ■

LEMMA 3.4. *Suppose $k \geq 1$. Let T be a tournament and let $\mathcal{Q}$ be a non-trivial partition of $V(T)$. Also let $Q \in \mathcal{Q}$ and $v \in Q$. Let*

$$\mathcal{W} := \{Q' \in \mathcal{Q} \setminus \{Q\} \mid \exists w \in Q' \colon (v,w) \in E(T) \wedge \mathrm{md}_{\mathcal{Q}}(v,w) \leq k\}.$$

Then $|\mathcal{W}| \leq 2k+1$.

The proof is very similar to the proof of Lemma 3.2.

PROOF. Let $\ell := |\mathcal{W}|$ and suppose $\mathcal{W} = \{Q_1, \dots, Q_\ell\}$. For every $i \in [\ell]$ pick an element $w_i \in Q_i$ such that $(v, w_i) \in E(T)$ and $\mathrm{md}_{\mathcal{Q}}(v, w_i) \leq k$. We define $W := \{w_1, \dots, w_\ell\}$. Then there is some $w \in W$ such that

$$|\{w' \in W \mid (w', w) \in E(T)\}| \geq \frac{\ell - 1}{2},$$

because the induced subtournament $T[W]$ has a vertex of in-degree at least $(\ell - 1)/2$.

Since $\mathrm{md}_{\mathcal{Q}}(v,w) \leq k$ and $(v,w') \in E(T)$ for all $w' \in W$, it follows that

$$|\{w' \in W \mid (w', w) \in E(T)\}| \leq k.$$

Thus $\frac{\ell-1}{2} \leq k$, which implies that $|\mathcal{W}| = \ell \leq 2k+1$. ■

Now, suppose we color all edges (v,w) of T with $\mathrm{md}(v,w) \leq \mathrm{tww}(T)$ using the color c_1 = blue (see Figure 1). Let $\mathcal{Q}_1$ be the partition into the (weakly) connected components of the graph induced by the blue edges and suppose that $\mathcal{Q}_1$ is non-trivial. We can compute isomorphisms between the different parts of $\mathcal{Q}_1$ using the algorithm from [1]. Next, let us color all cross-cluster edges $(v,w) \in E(T)$ with $\mathrm{md}_{\mathcal{Q}}(v,w) \leq \mathrm{tww}(T)$ using the color c_2 = green. Then every vertex has outgoing green edges to at most $2\,\mathrm{tww}(T) + 1$ other parts of $\mathcal{Q}_1$ (see Lemma 3.4). However, since a vertex may have an unbounded number of green neighbors in a single part, the out-degree of the graph induced by the green edges may be unbounded. So it is not possible to use the algorithm from [1] as a black-box on the components induced by blue and green edges. Luckily, the methods used in [1] can be extended to work even in this more general setting (see Section 4). So if the graph induced by the blue and green edges is connected, then we are again done. Otherwise, we let $\mathcal{Q}_2$ denote the partition into (weakly) connected components of the graph induced by the blue and green edges. Now, we can continue in the same fashion identifying colors $c_3, c_4, \dots$ and corresponding partitions $\mathcal{Q}_3, \mathcal{Q}_4, \dots$ until the graph induced by all edges of colors $c_1, \dots, c_\ell$ is eventually connected.

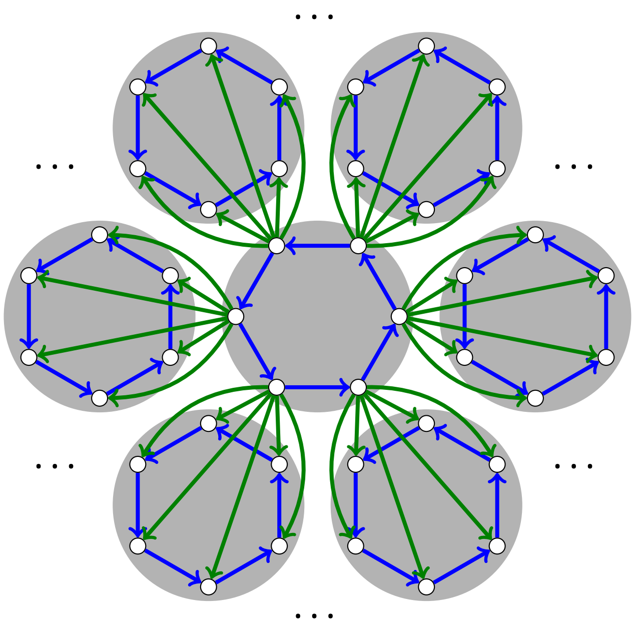

Figure 1. The figure shows part of a tournament T. The colors c_1 and c_2 are shown in blue and green, respectively. Also, the parts of the partition Q_1 are highlighted in gray. Note that only green edges, which are outgoing from the middle part, are shown.

Below, we provide a lemma that computes the corresponding sequence of partitions and edge colors using 2-WL. To state the lemma in its cleanest form, we restrict our attention to tournaments that are 2-WL-homogeneous. Recall that a tournament T is 2-WL-homogeneous if for all $v, w \in V(T)$ is holds that $\chi^{2,T}(v,v) = \chi^{2,T}(w,w)$.

We also require another piece of notation. For a directed graph G and a set of colors $C \subseteq \{\chi^{2,G}(v,w) \mid (v,w) \in E(G)\}$ we write $G[C]$ for the directed graph with vertex set $V(G[C]) := V(G)$ and edge set

$$E(G[C]) := \{(v,w) \in E(G) \mid \chi^{2,G} \in C\}.$$

To prove Lemma 3.6 we need the following lemma about the connected components of the graphs $G[C]$.

LEMMA 3.5. *Let G be a 2-WL-homogeneous graph, and let C be a set of colors in the range of $\chi^{2,G}$. Then the weakly connected components of $G[C]$ equal the strongly connected components of $G[C]$.*

PROOF. Let A be a weakly connected component of $G[C]$ and suppose towards a contradiction that A is not strongly connected in $G[C]$. Let $B_1, \dots, B_\ell$ be the strongly connected components of $G[C]$ within A. We may assume that $B_1, \dots, B_\ell$ are ordered topologically, i.e., there is an edge in $G[C]$ from a vertex in B_i to a vertex in B_j only if $i < j$. Now consider $B := B_1$, i.e., $B \subseteq A$ is a strongly connected component of $G[C]$ such that no edges of a color from C are incoming into B. Also, let $w \in A \setminus B$ denote a vertex such that there is a directed path from v to w for every $v \in B$. Also, fix a vertex $v \in B$.

Now, let $C^{\rightarrow}$ denote the set of all colors $\chi^{2,G}(x,y)$ where $x,y \in V(G)$ are distinct vertices such that there is a path from x to y in $G[C]$, but no path from y to x. By the properties of 2-WL, for every $x,y \in V(G)$ such that $\chi^{2,G}(x,y) \in C^{\rightarrow}$, there is a path from x to y in $G[C]$, but no path from y to x.

By definition, we have that $\chi^{2,G}(v,w) \in C^{\rightarrow}$, but there is no $u \in V(G)$ such that $\chi^{2,G}(u,v) \in C^{\rightarrow}$ (no edges of a color from C are incoming into B). It follows that $\chi^{2,G}(v,v) \neq \chi^{2,G}(w,w)$ which is a contradiction. ∎

LEMMA 3.6. *Let T be a 2-WL-homogeneous tournament of twin width $\operatorname{tww}(T) \leq k$. Then there is a sequence of partitions $\{\{v\} \mid v \in V(T)\} = \mathcal{Q}_0, \dots, \mathcal{Q}_\ell = \{V(T)\}$ of $V(T)$ where $\mathcal{Q}_{i-1}$ refines $\mathcal{Q}_i$ for all i, and a sequence of colors $c_1, \dots, c_\ell$ in the range of $\chi^{2,T}$ such that*

1. *$\mathcal{Q}_i$ is the partition into the strongly connected components of $T[\{c_1, \dots, c_i\}]$ for every $i \in [\ell]$, and*
2. *for every $i \in [\ell]$ and every $v \in V(T)$ it holds that*

$$\left|\left\{Q \in \mathcal{Q}_{i-1} \;\middle|\; \exists w \in Q \colon \chi^{2,T}(v,w) = c_i\right\}\right| \leq 2k+1.$$

Moreover, there is a polynomial-time algorithm that, given a tournament T and an integer $k \geq 1$, computes the desired sequences $\mathcal{Q}_0, \dots, \mathcal{Q}_\ell$ and $c_1, \dots, c_\ell$ or concludes that $\operatorname{tww}(T) > k$.

PROOF. We set $\mathcal{Q}_0 = \{\{v\} \mid v \in V(T)\}$ and inductively define a sequence of partitions and colors as follows. Let $i \geq 0$ and suppose we already defined partitions $\mathcal{Q}_0 \prec \dots \prec \mathcal{Q}_i$ and colors $c_1, \dots, c_i$. If $\mathcal{Q}_i = \{V(T)\}$, we set $\ell := i$ and complete both sequences. Otherwise, there is a cross-cluster edge (v_{i+1}, w_{i+1}) with respect to $\mathcal{Q}_i$ such that $\operatorname{md}_{\mathcal{Q}_i}(v_{i+1}, w_{i+1}) \leq k$ by Lemma 3.3. We set $c_{i+1} := \chi^{2,T}(v_{i+1}, w_{i+1})$ and define $\mathcal{Q}_{i+1}$ to be the set of weakly connected components of $T[c_1, \dots, c_{i+1}]$. By Lemma 3.5, these are also the strongly connected components.

First observe that $\mathcal{Q}_i \prec \mathcal{Q}_{i+1}$ since (v_{i+1}, w_{i+1}) is a cross-cluster edge with respect to $\mathcal{Q}_i$, and v_{i+1}, w_{i+1} are contained in the same part of $\mathcal{Q}_{i+1}$ by Lemma 3.5. Also, Property 1 is satisfied by definition. For Property 2 note that every edge $(v,w) \in E(T)$ such that $\chi^{2,G}(v,w) = c_{i+1}$ is a cross-cluster edge with respect to $\mathcal{Q}_i$. So Property 2 follows directly from Lemma 3.4.

Finally, it is clear from the description above that $\mathcal{Q}_0 \prec \dots \prec \mathcal{Q}_\ell$ and $c_1, \dots, c_\ell$ can be computed in polynomial time, or we conclude that $\operatorname{tww}(T) > k$. ∎

4. The Isomorphism Algorithm

Based on the structural insights summarized in Lemma 3.6, we now design an isomorphism test for tournaments of small twin width.

The strategy of our algorithm is the following. We are given two tournaments T_1 and T_2, and we want to compute $\mathrm{Iso}(T_1, T_2)$. First, we reduce to the case where both T_1 and T_2 are 2-WL-homogeneous.

Towards this end, we start by applying 2-WL and, for $j = 1, 2$, compute the coloring χ^{2,T_j}. If 2-WL distinguishes the two tournaments, we can immediately conclude that they are non-isomorphic and return $\mathrm{Iso}(T_1, T_2) = \emptyset$.

So suppose that 2-WL does not distinguish the tournaments. Then T_1 is 2-WL-homogeneous if and only if T_2 is 2-WL-homogeneous.

If T_1 and T_2 are not 2-WL-homogeneous, we rely on the following standard argument. Let $c_1, \dots, c_p$ be the vertex colors. For $i \in [p]$ and $j = 1, 2$, let $P_{j,i}$ be the set of all $v \in V(T_j)$ such that $\chi^{2,T_j}(v, v) = c_i$. We recursively compute the sets $\Lambda_i := \mathrm{Iso}(T_1[P_{1,i}], T_2[P_{2,i}])$ for all $i \in [p]$. If there is some $i \in [p]$ such that $\Lambda_i = \emptyset$, then T_1 and T_2 are non-isomorphic, and we return $\mathrm{Iso}(T_1, T_2) = \emptyset$. Otherwise, the set Λ_i is a coset of $\Gamma_i := \mathrm{Aut}(T_1[P_{1,i}])$ for all $i \in [p]$, i.e., $\Lambda_i = \Gamma_i\theta_i$ for some bijection $\theta_i \colon P_{1,i} \to P_{2,i}$. As the automorphism group of a tournament, Γ_i is solvable (see Theorem 2.3). Moreover, since the color classes $P_{1,i}$ are invariant under automorphisms of T_1, the automorphism group $\Gamma := \mathrm{Aut}(T_1)$ is a subgroup of the direct product $\prod_i \Gamma_i$, which is also a solvable group. Also, $\mathrm{Iso}(T_1, T_2) \subseteq \Gamma\theta$ where $\theta \colon V(T_1) \to V(T_2)$ is the unique bijection defined via $\theta(v) := \theta_i(v)$ for all $v \in V(T_1)$, where $i \in [p]$ is the unique index such that $v \in P_{1,i}$. So $\mathrm{Iso}(T_1, T_2) = \mathrm{Iso}_{\Gamma\theta}(T_1, T_2)$ can be computed in polynomial time using Theorem 2.5.

So we may assume that T_1 and T_2 are 2-WL-homogeneous. In this case, we apply Lemma 3.6 and obtain colors $c_1, \dots, c_\ell$ and, for $j = 1, 2$, a partition sequence $\{\{v\} \mid v \in V(T_j)\} = \mathcal{Q}_{j,0}, \dots, \mathcal{Q}_{j,\ell} = \{V(T_j)\}$ where $\mathcal{Q}_{j,i-1}$ refines $\mathcal{Q}_{j,i}$ for all $i \in [\ell]$.

Now, we iteratively compute for $i = 0, \dots, \ell$ the sets $\mathrm{Iso}(T_j[Q], T_{j'}[Q'])$ for all $j, j' \in \{1, 2\}$ and all $Q \in \mathcal{Q}_{j,i}$ and $Q' \in \mathcal{Q}_{j',i}$. For $i = 0$ this is trivial since all parts have size 1. The next lemma describes the key subroutine of the main algorithm which allows us to compute the isomorphism sets for level $i \in [\ell]$ given all the sets for level $i - 1$. Note that, on the last level ℓ, we compute the set $\mathrm{Iso}(T_1, T_2)$ since $\mathcal{Q}_{j,\ell} = \{V(T_j)\}$ for both $j \in \{1, 2\}$.

To state the lemma, we need additional terminology. Let $T = (V, E, \lambda)$ be an arc-colored tournament. A partition $\mathcal{Q}$ of V is *λ-definable* if there is a set of colors $C \subseteq \{\lambda(v, w) \mid v = w \vee (v, w) \in E\}$ such that

$$v \sim_{\mathcal{Q}} w \quad \iff \quad \lambda(v, w) \in C$$

for all $v, w \in V$ such that $v = w$ or $(v, w) \in E$. We also say that *$\mathcal{Q}$ is λ-defined by C*. If $\mathcal{Q}$ is λ-defined by C then we can partition the colors in the range of λ into the colors in C, which we call *intra-cluster* colors, and the remaining colors, which we call *cross-cluster* colors. Note that if a color c is intra-cluster, then for all $(v, w) \in E$ with $\lambda(v, w) = c$ it holds that $v, w \in Q$ for some $Q \in \mathcal{Q}$, and if c is cross-cluster, then for all $(v, w) \in E$ with $\lambda(v, w) = c$ it holds that (v, w) is a cross-cluster edge, that is, $v \in Q$ and $w \in Q'$ for distinct $Q, Q' \in \mathcal{Q}$.

LEMMA 4.1. *There is an algorithm that, given*

(A) an integer $d \geq 1$;

(B) two arc-colored tournaments $T_1 = (V_1, E_1, \lambda_1)$ and $T_2 = (V_2, E_2, \lambda_2)$;

(C) a set of colors C and for $j = 1, 2$ a partition $\mathcal{Q}_j$ of V_j that is λ_j-defined by C;

(D) a color c^* that is cross-cluster with respect to $\mathcal{Q}_j$ for $j = 1, 2$ and

- *for every $v \in V_j$ it holds that*

$$\left|\left\{Q \in \mathcal{Q}_j \;\middle|\; \exists w \in Q\colon (v, w) \in E_j \wedge \lambda_j(v, w) = c^*\right\}\right| \leq d;$$

- *for*

$$F_j := \left\{(Q, Q') \in \mathcal{Q}_j^2 \;\middle|\; Q \neq Q', \exists w \in Q, w' \in Q'\colon (w, w') \in E_j \wedge \lambda_j(w, w') = c^*\right\}$$

the directed graph $G_j = (\mathcal{Q}_j, F_j)$ is strongly connected;

(E) $\mathrm{Iso}\left(T_j[Q], T_{j'}[Q']\right)$ for every $j, j' \in \{1, 2\}$ and every $Q \in \mathcal{Q}_j$, $Q' \in \mathcal{Q}_{j'}$,

computes $\mathrm{Iso}(T_1, T_2)$ in time $d^{O(\log d)} \cdot n^{O(1)}$.

PROOF. The algorithm fixes an arbitrary vertex $r_1 \in V_1$. For every $r_2 \in V_2$ we compute the set $\mathrm{Iso}((T_1, r_1), (T_2, r_2))$ of all isomorphisms $\varphi \in \mathrm{Iso}(T_1, T_2)$ such that $\varphi(r_1) = r_2$. Observe that

$$\mathrm{Iso}(T_1, T_2) = \bigcup_{r_2 \in V_2} \mathrm{Iso}((T_1, r_1), (T_2, r_2)).$$

So for the remainder of the proof, let us also fix some $r_2 \in V_2$. Let $R_1 \in \mathcal{Q}_1$ be the unique set such that $r_1 \in R_1$, and similarly let $R_2 \in \mathcal{Q}_2$ be the unique set such that $r_2 \in R_2$.

We compute sets $\mathrm{Iso}((T_1[W_{1,i}], r_1), (T_2[W_{2,i}], r_2))$ for increasingly larger subsets $R_1 \subseteq W_{1,i} \subseteq V_1$ and $R_2 \subseteq W_{2,i} \subseteq V_2$. In other words, in each iteration $i \geq 0$, we build increasingly larger "windows" $W_{i,1}$ and $W_{i,2}$ and compute all isomorphisms between the sub-tournaments induced by $W_{i,1}$ and $W_{i,2}$. To obtain $W_{i+1,1}$ and $W_{i+1,2}$ in iteration $i+1$, we add a positive number of vertices to $W_{i,1}$ and $W_{i,2}$ and update the isomorphism set computed in the previous round to take the newly added vertices into account. Throughout, we maintain the property that

(I.1) there is some $\{R_j\} \subseteq \mathcal{W}_{j,i} \subseteq \mathcal{Q}_j$ such that $W_{j,i} = \bigcup \mathcal{W}_{j,i}$, and

(I.2) $\varphi(W_{1,i}) = W_{2,i}$ for every $\varphi \in \mathrm{Iso}((T_1, r_1), (T_2, r_2))$

for every $i \geq 0$.

Eventually, the algorithm either concludes that $\mathrm{Iso}((T_1, r_1), (T_2, r_2)) = \emptyset$ and terminates, or it reaches a point where $W_{i,1} = V_1$ and $W_{i,2} = V_2$ which gives the desired isomorphism set $\mathrm{Iso}((T_1, r_1), (T_2, r_2))$.

For the base case $i = 0$ we initialize $\mathcal{W}_{j,0} := \{R_j\}$ for both $j \in \{1, 2\}$. Also, we set $W_{j,0} := R_j$. This means we are given the set $\Gamma_0'\theta_0' := \mathrm{Iso}(T_1[W_{1,0}], T_2[W_{2,0}])$ as part of the input. Observe that Γ_0' is solvable by Theorem 2.3. So

$$\Gamma_0\theta_0 := \mathrm{Iso}((T_1[W_{1,0}], r_1), (T_2[W_{2,0}], r_2))$$

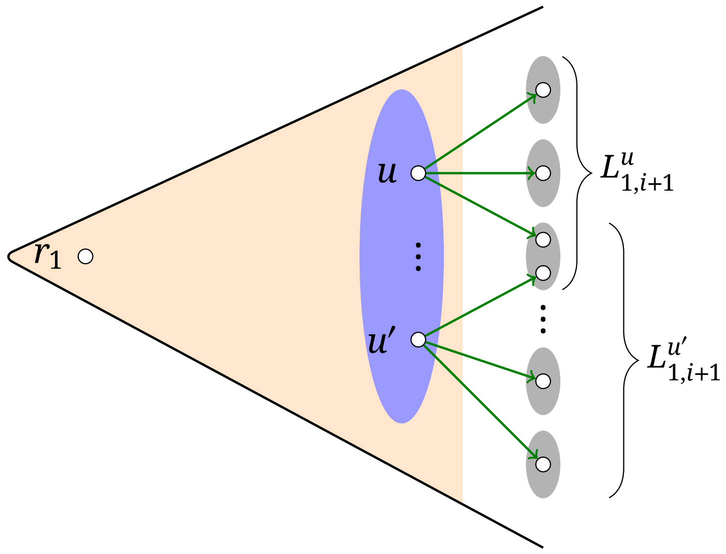

Figure 2. The figure shows the sets $W_{1,i}$ (orange), $U_{1,i}$ (blue) and $L^u_{1,i+1}$ computed in the proof of Lemma 4.1. The color c^* is shown in green and gray regions depict parts of the partition $\mathcal{Q}_1$.

can be computed in polynomial time using Theorem 2.5.

So assume $i \geq 0$ and let $W_{1,i} \subsetneq V_1$ and $W_{2,i} \subsetneq V_2$ be two sets satisfying (I.1) and (I.2). Also suppose we have already computed the set

$$\Gamma_i\theta_i := \mathrm{Iso}((T_1[W_{1,i}], r_1), (T_2[W_{2,i}], r_2)). \tag{2}$$

We may assume the isomorphism set is non-empty; otherwise $\mathrm{Iso}((T_1, r_1), (T_2, r_2)) = \emptyset$ by Invariant (I.2) and we terminate. For $j \in \{1, 2\}$ let

$$U_{j,i} := \{u \in W_{j,i} \mid \exists w \in V_j \setminus W_{j,i} \colon (u, w) \in E_j \wedge \lambda_j(u, w) = c^*\}.$$

Note that $U_{j,i} \neq \emptyset$ for both $j \in \{1, 2\}$ by Property (D). Also, for every $j \in \{1, 2\}$ and $u \in U_{j,i}$, we define

$$\mathcal{L}^u_{j,i+1} := \{Q \in \mathcal{Q}_j \setminus \mathcal{W}_{j,i} \mid \exists w \in Q \colon (u, w) \in E_j \wedge \lambda_j(u, w) = c^*\}.$$

We have

$$|\mathcal{L}^u_{j,i+1}| \leq d$$

by Property (D). We define

$$L^u_{j,i+1} := \bigcup \mathcal{L}^u_{j,i+1}$$

and the tournament

$$T^u_j := T_j[L^u_{j,i+1}]$$

for every $j \in \{1, 2\}$ and $u \in U_{j,i}$. A visualization is also given in Figure 2.

CLAIM 4.2. *For every $j, j' \in \{1, 2\}$ and every $u \in U_{j,i}$, $u' \in U_{j',i}$ the set $\mathrm{Iso}(T^u_j, T^{u'}_{j'})$ can be computed in time $d^{O(\log d)} \cdot n^{O(1)}$.*

Proof. Fix $j, j' \in \{1,2\}$ and $u \in U_{j,i}$, $u' \in U_{j',i}$. We define $\mathcal{L} := \mathcal{L}^{u}_{j,i+1} \uplus \mathcal{L}^{u'}_{j',i+1}$ (i.e., we take the disjoint union of the two sets).

For each $Q \in \mathcal{L}$ we fix an orbit $A_Q \subseteq Q$ of the group $\mathrm{Aut}(T_b[Q])$, where $b \in \{j, j'\}$ is the unique index such that $Q \in \mathcal{Q}_b$, subject to the following compatibility condition: If $b, b' \in \{j, j'\}$, $Q \in \mathcal{L} \cap \mathcal{Q}_b$, $Q' \in \mathcal{L} \cap \mathcal{Q}_{b'}$, and $\varphi \in \mathrm{Iso}(T_b[Q], T_{b'}[Q'])$ then $\varphi(A_Q) = A_{Q'}$. To see that this is possible, recall that if $\varphi \in \mathrm{Iso}(T_b[Q], T_{b'}[Q'])$ then $\varphi^{-1}\,\mathrm{Aut}(T_j[Q])\varphi = \mathrm{Aut}(T_{b'}[Q'])$. Observe that $|A_Q|$ is odd for every $Q \in \mathcal{L}$.

We define the tournament $\widetilde{T}$ with vertex set $V(\widetilde{T}) := \mathcal{L}^{u}_{j,i+1}$ and edge set

$$E(\widetilde{T}) := \left\{ (Q, Q') \;\middle|\; |E_{T_j}(A_Q, A_{Q'})| > |E_{T_j}(A_{Q'}, A_Q)| \right\}.$$

Similarly, we define the tournament $\widetilde{T}'$ with vertex set $V(\widetilde{T}') := \mathcal{L}^{u'}_{j',i+1}$ and edge set

$$E(\widetilde{T}') := \left\{ (Q, Q') \;\middle|\; |E_{T_{j'}}(A_Q, A_{Q'})| > |E_{T_{j'}}(A_{Q'}, A_Q)| \right\}.$$

Also, we color the vertices of both tournaments in such a way that vertices $Q, Q' \in \mathcal{L}$ receive the same color if and only if $\mathrm{Iso}(T_b[Q], T_{b'}[Q']) \neq \emptyset$, where $b, b' \in \{j, j'\}$ are the unique indices such that $Q \in \mathcal{Q}_b$ and $Q' \in \mathcal{Q}_{b'}$.

Observe that $|V(\widetilde{T})| \leq d$ and $|V(\widetilde{T}')| \leq d$. So we can compute the set $\mathrm{Iso}(\widetilde{T}, \widetilde{T}')$ in time $d^{O(\log d)}$ by Theorem 2.4. If $\mathrm{Iso}(\widetilde{T}, \widetilde{T}') = \emptyset$ then also $\mathrm{Iso}(T^{u}_{j}, T^{u'}_{j'}) = \emptyset$ and we are done. So suppose that $\mathrm{Iso}(\widetilde{T}, \widetilde{T}') \neq \emptyset$, and let us write

$$\widetilde{\Delta}\widetilde{\theta} := \mathrm{Iso}(\widetilde{T}, \widetilde{T}'),$$

i.e., $\widetilde{\Delta} = \mathrm{Aut}(\widetilde{T})$ and $\widetilde{\theta} \in \mathrm{Iso}(\widetilde{T}, \widetilde{T}')$.

We define $\Delta \leq \mathrm{Sym}(L^{u}_{j,i+1})$ to be the permutation group containing all elements $\delta \in \mathrm{Sym}(L^{u}_{j,i+1})$ such that there are $\widetilde{\delta} \in \widetilde{\Delta}$ and, for every $Q \in \mathcal{L}^{u}_{j,i+1}$, some $\varphi_Q \in \mathrm{Iso}(T_j[Q], T_j[\widetilde{\delta}(Q)])$ such that

$$\delta(v) = \varphi_Q(v)$$

for all $v \in L^{u}_{j,i+1}$ where $Q \in \mathcal{L}^{u}_{j,i+1}$ is the unique element such that $v \in Q$. Observe that Δ is solvable since it is a wreath product of solvable groups (see Theorem 2.3). Moreover, a generating set for Δ can be computed in polynomial time (see Section 2.5 for details).

Next, we define a bijection $\theta\colon L^{u}_{j,i+1} \to L^{u'}_{j',i+1}$ as follows. For each $Q \in \mathcal{L}^{u}_{j,i+1}$ pick an arbitrary isomorphism $\varphi_Q \in \mathrm{Iso}(T_j[Q], T_{j'}[\widetilde{\theta}(Q)])$. Then, set

$$\theta(v) := \varphi_Q(v)$$

for all $v \in L^{u}_{j,i+1}$ where $Q \in \mathcal{L}^{u}_{j,i+1}$ is the unique element such that $v \in Q$. We have that

$$\mathrm{Iso}(T^{u}_{j}, T^{u'}_{j'}) \subseteq \Delta\theta.$$

So

$$\mathrm{Iso}(T_j^u, T_{j'}^{u'}) = \mathrm{Iso}_{\Delta\theta}(T_j^u, T_{j'}^{u'})$$

can be computed in polynomial time using Theorem 2.5. ◆

Let $U_i := U_{1,i} \cup U_{2,i}$. We define an equivalence relation $\sim$ on U_i by setting $u \sim u'$ if $\mathrm{Iso}(T_j^u, T_{j'}^{u'}) \neq \emptyset$ where $j, j' \in \{1,2\}$ are the unique indices such that $u \in U_{j,i}$ and $u' \in U_{j',i}$. We pick an arbitrary equivalence class $U_i^* \subseteq U_i$ of the relation $\sim$. Let $U_{j,i}^* := U_{j,i} \cap U_i^*$ for every $j \in \{1,2\}$. Also, let

$$L_{j,i+1} := \bigcup_{u \in U_{j,i}^*} L_{j,i+1}^u$$

for every $j \in \{1,2\}$.

CLAIM 4.3. *$\varphi(U_{1,i}^*) = U_{2,i}^*$ and $\varphi(L_{1,i+1}) = L_{2,i+1}$ for every $\varphi \in \mathrm{Iso}((T_1,r_1),(T_2,r_2))$.*

Proof. Let $\varphi \in \mathrm{Iso}((T_1,r_1),(T_2,r_2))$. By (I.2) we get that $\varphi(W_{1,i}) = W_{2,i}$. This implies that $\varphi(U_{1,i}) = U_{2,i}$.

Let $u \in U_{1,i}^*$. Then φ restricts to an isomorphism from T_1^u to $T_2^{\varphi(u)}$. In particular, $u \sim \varphi(u)$ which implies that $\varphi(u) \in U_{2,i}^*$. So $\varphi(U_{1,i}^*) \subseteq U_{2,i}^*$. Also note that $\varphi(L_{1,i+1}^u) = L_{2,i+1}^{\varphi(u)}$. So $\varphi(L_{1,i+1}) \subseteq L_{2,i+1}$.

In the other direction, let $u' \in U_{2,i}^*$. Then φ^{-1} restricts to an isomorphism from $T_1^{\varphi^{-1}(u')}$ to T_2^u. In particular, $\varphi^{-1}(u') \sim u'$ which implies that $\varphi^{-1}(u') \in U_{1,i}^*$. So $U_{2,i}^* \subseteq \varphi(U_{1,i}^*)$. Also note that $\varphi^{-1}(L_{2,i+1}^{u'}) = L_{1,i+1}^{\varphi^{-1}(u')}$. So $L_{2,i+1} \subseteq \varphi(L_{1,i+1})$. ◆

If $|U_{1,i}^*| \neq |U_{2,i}^*|$ or $|L_{1,i+1}| \neq |L_{2,i+1}|$ then $\mathrm{Iso}((T_1,r_1),(T_2,r_2)) = \emptyset$ and the algorithm terminates. So suppose that $|U_{1,i}^*| = |U_{2,i}^*|$ and $|L_{1,i+1}| = |L_{2,i+1}|$. We set

$$W_{j,i+1} := W_{j,i} \cup L_{j,i+1}$$

for both $j \in \{1,2\}$. Clearly, (I.1) and (I.2) remain satisfied using Claim 4.3. It remains to compute

$$\mathrm{Iso}((T_1[W_{1,i+1}], r_1), (T_2[W_{2,i+1}], r_2)).$$

Recall that $\Gamma_i\theta_i = \mathrm{Iso}((T_1[W_{1,i}], r_1), (T_2[W_{2,i}], r_2))$ denotes the isomorphism set computed in the previous iteration; see Equation (2). We start by computing the set

$$\Gamma_i'\theta_i' := \mathrm{Iso}_{\Gamma_i\theta_i}((T[W_{1,i}], r_1, U_{1,i}^*), (T[W_{2,i}], r_2, U_{2,i}^*))$$

of all bijections from $\Gamma_i\theta_i$ mapping $U_{1,i}^*$ to $U_{2,i}^*$ in polynomial time using Theorem 2.6 (recall that $\Gamma_i = \mathrm{Aut}(T[W_{1,i}], r_1)$ is solvable by Theorem 2.3). As before, we may assume the isomorphism set is non-empty; otherwise $\mathrm{Iso}((T_1,r_1),(T_2,r_2)) = \emptyset$ using Invariant (I.2) and we terminate.

Now, we consider the sets

$$A_{j,i+1} := \{(u,w) \mid u \in U_{j,i}^*, w \in L_{j,i+1}^u\}$$

for both $j \in \{1,2\}$. Intuitively speaking, the reader is encouraged to view $A_{j,i+1}$ as a disjoint union over all sets $L^{u}_{j,i+1}$, $u \in U^{*}_{j,i}$. The basic idea is to first compute suitable "isomorphisms" between the sets $A_{1,i+1}$ and $A_{2,i+1}$. Afterwards, we "merge" elements in the sets $A_{j,i+1}$ with the same second coordinate to obtain the isomorphism set between $L_{1,i+1}$ and $L_{2,i+1}$.

CLAIM 4.4.

$$\varphi(A_{1,i+1}) := \{(\varphi(u),\varphi(w)) \mid (u,w) \in A_{1,i+1}\} = A_{2,i+1}$$

for every $\varphi \in \mathrm{Iso}((T_1,r_1),(T_2,r_2))$.

Proof. Let $\varphi \in \mathrm{Iso}((T_1,r_1),(T_2,r_2))$. Also let $(u,w) \in A_{1,i+1}$. Then $\varphi(u) \in U^{*}_{2,i}$ by Claim 4.3. Also, $\varphi(L^{u}_{1,i+1}) = L^{\varphi(u)}_{2,i+1}$. In particular, $\varphi(w) \in L^{\varphi(u)}_{2,i+1}$. So we get that $(\varphi(u),\varphi(w)) \in A_{2,i+1}$.

In the other direction, let $(u',w') \in A_{2,i+1}$. Then $\varphi^{-1}(u') \in U^{*}_{1,i}$ by Claim 4.3. Also, $\varphi^{-1}(L^{u'}_{2,i+1}) = L^{\varphi^{-1}(u')}_{1,i+1}$. In particular, $\varphi^{-1}(w') \in L^{\varphi^{-1}(u')}_{1,i+1}$. So we get that $(\varphi^{-1}(u'),\varphi^{-1}(w')) \in A_{1,i+1}$. ◆

We now define $\Delta_{i+1} \leq \mathrm{Sym}(A_{1,i+1})$ to be the permutation group containing all elements $\delta \in \mathrm{Sym}(A_{1,i+1})$ such that there is a $\gamma \in \Gamma'_i$ and, for each $u \in U^{*}_{1,i}$, a permutation $\varphi_u \in \mathrm{Iso}(T^{u}_{1}, T^{\gamma(u)}_{1})$ such that

$$\delta(u,w) = (\gamma(u),\varphi_u(w))$$

for all $(u,w) \in A_{1,i+1}$. Observe that Δ_{i+1} is solvable since it is a wreath product of solvable groups. Also, a generating set for Δ_{i+1} can be computed in polynomial time (see Section 2.5 for details).

We define a bijection $\delta_{i+1}\colon A_{1,i+1} \to A_{2,i+1}$ as follows. For each $u \in U^{*}_{1,i}$ pick an arbitrary isomorphism $\varphi_u \in \mathrm{Iso}(T^{u}_{1}, T^{\theta'_i(u)}_{2})$. Then, set

$$\delta_{i+1}(u,w) := (\theta'_i(u),\varphi_u(w))$$

for all $(u,w) \in A_{1,i+1}$.

For $j \in \{1,2\}$ we define a hypergraph $\mathcal{H}_{j,i+1} = (A_{j,i+1},\mathcal{E}_{j,i+1})$ where

$$\mathcal{E}_{j,i+1} := \{\{(u,w') \in A_{j,i+1} \mid w' = w\} \mid w \in L_{j,i+1}\}.$$

We compute

$$\Delta'_{i+1}\delta'_{i+1} := \mathrm{Iso}_{\Delta_{i+1}\delta_{i+1}}(\mathcal{H}_{1,i+1},\mathcal{H}_{2,i+1})$$

and the natural action $\Delta^{*}_{i+1}\delta^{*}_{i+1}$ of $\Delta'_{i+1}\delta'_{i+1}$ on $L_{1,i+1}$ (via the correspondence between $L_{j,i+1}$ and $\mathcal{E}_{j,i+1}$).

We get that

$$\mathrm{Iso}((T_1[W_{1,i+1}],r_1),(T_2[W_{2,i+1}],r_2)) \subseteq \Gamma'_i\theta'_i \times \Delta^{*}_{i+1}\delta^{*}_{i+1}.$$

Indeed, let $\varphi \in \mathrm{Iso}((T_1[W_{1,i+1}],r_1),(T_2[W_{2,i+1}],r_2))$. Then $\varphi(U^{*}_{1,i}) = U^{*}_{2,i}$ (see Claim 4.3) and thus $\varphi|_{W_{1,i}} \in \Gamma'_i\theta'_i$. Moreover, the action of φ on $A_{1,i+1} = U^{*}_{1,i} \times L_{1,i+1}$ is in $\Delta'_{i+1}\delta'_{i+1}$ and thus

$\varphi|_{L_{1,i+1}} \in \Delta^*_{i+1}\delta^*_{i+1}$. In combination, the mapping $\varphi \mapsto (\varphi|_{W_{1,i}}, \varphi|_{L_{1,i+1}})$ embeds the isomorphism set $\mathrm{Iso}((T_1[W_{1,i+1}], r_1), (T_2[W_{2,i+1}], r_2))$ into $\Gamma'_i\theta'_i \times \Delta^*_{i+1}\delta^*_{i+1}$.

Since $\Gamma'_i \times \Delta^*_{i+1}$ is solvable, we can compute $\mathrm{Iso}((T_1[W_{1,i+1}], r_1), (T_2[W_{2,i+2}], r_2))$ in polynomial time using Theorem 2.5.

This entire procedure is repeated until we reach the stage where $W_{1,\ell} = V_1$ and $W_{2,\ell} = V_2$ (or the algorithm concludes that $\mathrm{Iso}((T_1, r_1), (T_2, r_2)) = \emptyset$). At this point the algorithm has computed $\mathrm{Iso}((T_1, r_1), (T_2, r_2))$ as desired. Note that $\ell \leq |V_1|$ since in each iteration at least one element is added, i.e., $W_{1,i} \subsetneq W_{1,i+1}$ for all $i \in [\ell - 1]$.

Also observe that all steps can be performed in polynomial time except for the subroutine from Claim 4.2 which runs in time $d^{O(\log d)} \cdot n^{O(1)}$. Overall, this gives the desired bound on the running time of the algorithm. ■

Building on the subroutine from Lemma 4.1, we can now design an isomorphism test for 2-WL-homogeneous tournaments of bounded twin width.

LEMMA 4.5. *There is an algorithm that, given* 2*-WL-homogeneous tournaments* T_1 *and* T_2 *and an integer* $k \geq 1$*, either concludes that* $\mathrm{tww}(T_1) > k$ *or computes* $\mathrm{Iso}(T_1, T_2)$ *in time* $k^{O(\log k)} \cdot n^{O(1)}$.

PROOF. We run 2-WL on T_1 and T_2. If 2-WL distinguishes between T_1 and T_2, then we return that $\mathrm{Iso}(T_1, T_2) = \emptyset$. So suppose that $T_1 \simeq_2 T_2$.

For $j \in \{1, 2\}$ we define an arc-coloring via $\lambda_j(v, w) := \chi^{2,T_j}(v, w)$ for every $(v, w) \in E(T) \cup \{(v, v) \mid v \in V(T)\}$. We write $\widehat{T}_j = (V(T_j), E(T_j), \lambda_j)$ for the corresponding arc-colored version of T_j.

We apply Lemma 3.6 to the graph T_1 and obtain a sequence of partitions $\{\{v\} \mid v \in V(T_1)\} = \mathcal{Q}_{1,0} \prec \cdots \prec \mathcal{Q}_{1,\ell} = \{V(T_1)\}$ of $V(T_1)$ and a sequence of colors $c_1, \ldots, c_\ell \in \{\chi^{2,T_1}(v, w) \mid (v, w) \in E(T_1)\}$ satisfying Properties 1 and 2 (or $\mathrm{tww}(T_1) > k$). Also, we define $\mathcal{Q}_{2,i}$ to be the partition into the strongly connected components of $T_2[c_1, \ldots, c_i]$. By the properties of the 2-dimensional Weisfeiler-Leman algorithm we conclude that for every $i \in [\ell]$ and every $v \in V(T_2)$ it holds that

$$|\{Q \in \mathcal{Q}_{2,i-1} \mid \exists w \in Q \colon \chi^{2,T_2}(v, w) = c_i\}| \leq 2k + 1,$$

i.e., Property 2 is also satisfied for the sequence $\mathcal{Q}_{2,0} \prec \cdots \prec \mathcal{Q}_{2,\ell}$.

Now, for every $i \in \{0, \ldots, \ell\}$, every $j, j' \in \{1, 2\}$, and every $Q \in \mathcal{Q}_{j,i}$, $Q' \in \mathcal{Q}_{j',i}$ we inductively compute the set $\mathrm{Iso}(\widehat{T}_j[Q], \widehat{T}_{j'}[Q'])$. For $i = 0$ this trivial since $|Q| = |Q'| = 1$.

So suppose $i \in \{0, \ldots, \ell - 1\}$, $j, j' \in \{1, 2\}$, and $Q \in \mathcal{Q}_{j,i+1}$, $Q' \in \mathcal{Q}_{j',i+1}$. Let $d := 2k + 1$. We set $\widetilde{T}_1 := \widehat{T}_j[Q]$ and $\widetilde{T}_2 := \widehat{T}_{j'}[Q']$. Also, we set

$$\widetilde{\mathcal{Q}}_1 := \{\widetilde{Q} \in \mathcal{Q}_{j,i} \mid \widetilde{Q} \subseteq Q\}$$

and

$$\widetilde{\mathcal{Q}}_2 := \{\widetilde{Q} \in \mathcal{Q}_{j',i} \mid \widetilde{Q} \subseteq Q'\}.$$

Note that $\widetilde{Q}_1$ is a partition of $V(\widetilde{T}_1)$ and $\widetilde{Q}_2$ is a partition of $V(\widetilde{T}_2)$. Finally, we set $c^* := c_{i+1}$ and apply Lemma 4.1 to compute the set

$$\mathrm{Iso}(\widehat{T}_j[Q], \widehat{T}_{j'}[Q']) = \mathrm{Iso}(\widetilde{T}_1, \widetilde{T}_2)$$

in time $d^{O(\log d)} \cdot n^{O(1)} = k^{O(\log k)} \cdot n^{O(1)}$. Observe that all isomorphism sets required in Lemma 4.1(E) have already been computed in the previous iteration.

Finally, observe that $Q_{1,\ell} = \{V(T_1)\}$ and $Q_{2,\ell} = \{V(T_2)\}$. So in iteration $i = \ell$ the algorithm computes

$$\mathrm{Iso}(\widehat{T}_1, \widehat{T}_2) = \mathrm{Iso}(T_1, T_2).$$

To bound the running algorithm, one can observe that all steps can be performed in polynomial time except for the subroutine in Lemma 4.1. However, this subroutine is only called at most $\ell \cdot (|V(T_1)| + |V(T_2)|)^2$ many times, which is polynomial in the input size, and runs in time $k^{O(\log k)} \cdot n^{O(1)}$. Overall, this gives the desired running time. ■

Finally, relying on well-established arguments, we can remove the restriction that all vertices need to receive the same color under 2-WL to obtain Theorem 1.1. We reformulate our main algorithmic result in a slightly more precise manner.

THEOREM 4.6. *There is an algorithm that, given two tournaments T_1 and T_2 and an integer $k \geq 1$, either concludes that $\mathrm{tww}(T_1) > k$ or computes $\mathrm{Iso}(T_1, T_2)$ in time $k^{O(\log k)} \cdot n^{O(1)}$.*

PROOF. We run 2-WL on T_1 and T_2. If 2-WL distinguishes between T_1 and T_2, then we return that $\mathrm{Iso}(T_1, T_2) = \emptyset$. So suppose that $T_1 \simeq_2 T_2$.

Let $C_V := \{\chi^{2,T_1}(v, v) \mid v \in V(T_1)\}$ denote the set of *vertex colors* of the coloring χ^{2,T_1}. For $c \in C_V$ and $j \in \{1, 2\}$ we define

$$X_{j,c} := \{v \in V(T_j) \mid \chi^{2,T_j}(v, v) = c\}.$$

For every $c \in C_V$ we use the algorithm from Lemma 4.5 to compute the set

$$\Gamma_c\theta_c := \mathrm{Iso}(T_1[X_{1,c}], T_2[X_{2,c}]).$$

If there is some $c \in C_V$ for which the algorithm from Lemma 4.5 concludes that $\mathrm{tww}(T_1[X_{1,c}]) > k$, then $\mathrm{tww}(T_1) > k$ by Lemma 2.1.

Moreover, if there is a color $c \in C_V$ such that $\mathrm{Iso}(T_1[X_{1,c}], T_2[X_{2,c}]) = \emptyset$, then $\mathrm{Iso}(T_1, T_2) = \emptyset$. Otherwise, let

$$\Gamma := \bigtimes_{c \in C_V} \Gamma_c \leq \mathrm{Sym}(V(T_1))$$

and $\theta \colon V(T_1) \to V(T_2)$ be the bijection defined via $\theta(v) := \theta_c(v)$ for the unique $c \in C_V$ such that $v \in X_{1,c}$. Then

$$\mathrm{Iso}(T_1, T_2) \subseteq \Gamma\theta.$$

Moreover, the group Γ_c is solvable for every $c \in C_V$ by Theorem 2.3. This implies that Γ is solvable. So

$$\mathrm{Iso}(T_1, T_2) = \mathrm{Iso}_{\Gamma\theta}(T_1, T_2)$$

can be computed in polynomial time using Theorem 2.5. ■

5. The WL-Dimension of Tournaments of Bounded Twin Width

In this section, we prove Theorem 1.2. The proof relies on well-known game-characterizations of the Weisfeiler-Leman algorithm as well as the graph parameter tree width.

5.1 Games

First, we cover the cops-and-robber game that characterizes tree width of graphs. Fix some integer $k \geq 1$. For an undirected graph G, we define the cops-and-robber game $\mathrm{CopRob}_k(G)$ as follows:

- The game has two players called *Cops* and *Robber*.
- The game proceeds in rounds, each of which is associated with a pair of positions $(\bar{v}, u)$ with $\bar{v} \in \big(V(G)\big)^k$ and $u \in V(G)$.
- To determine the initial position, the Cops first choose a tuple $\bar{v} = (v_1, \dots, v_k) \in \big(V(G)\big)^k$ and then the Robber chooses some vertex $u \in V(G) \setminus \{v_1, \dots, v_k\}$ (if no such u exists, the Cops win the play). The initial position of the game is then set to $(\bar{v}, u)$.
- Each round consists of the following steps. Suppose the current position of the game is $(\bar{v}, u) = ((v_1, \dots, v_k), u)$.
  (C) The Cops choose some $i \in [k]$ and $v' \in V(G)$.
  (R) The Robber chooses a vertex $u' \in V(G)$ such that there exists a path from u to u' in the graph $G \setminus \{v_1, \dots, v_{i-1}, v_{i+1}, \dots, v_k\}$. After that, the game moves to the new position $\big((v_1, \dots, v_{i-1}, v', v_{i+1}, \dots, v_k), u'\big)$.

  If $u \in \{v_1, \dots, v_k\}$ the Cops win. If there is no position of the play such that the Cops win, then the Robber wins.

We say that the Cops (and the Robber, respectively) *win* $\mathrm{CopRob}_k(G)$ if the Cops (and the Robber, respectively) have a winning strategy for the game. We also say that *k cops can catch a robber on G* if the Cops have a winning strategy in this game.

THEOREM 5.1 ([52]). *An undirected graph G has tree width at most k if and only if $k + 1$ cops can catch a robber on G.*

Next, we discuss a game-theoretic characterization of the Weisfeiler-Leman algorithm.

Let $k \geq 1$. For graphs G and H on the same number of vertices, we define the *bijective k-pebble game* $\mathrm{BP}_k(G, H)$ as follows:

- The game has two players called *Spoiler* and *Duplicator*.
- The game proceeds in rounds, each of which is associated with a pair of positions $(\bar{v}, \bar{w})$ with $\bar{v} \in \big(V(G)\big)^k$ and $\bar{w} \in \big(V(H)\big)^k$.
- To determine the initial position, Duplicator plays a bijection $f\colon \big(V(G)\big)^k \to \big(V(H)\big)^k$ and Spoiler chooses some $\bar{v} \in \big(V(G)\big)^k$. The initial position of the game is then set to $(\bar{v}, f(\bar{v}))$.
- Each round consists of the following steps. Suppose the current position of the game is $(\bar{v}, \bar{w}) = ((v_1, \dots, v_k), (w_1, \dots, w_k))$.
  (S) Spoiler chooses some $i \in [k]$.
  (D) Duplicator picks a bijection $f\colon V(G) \to V(H)$.
  (S) Spoiler chooses $v \in V(G)$ and sets $w := f(v)$. Then the game moves to position $\big(\bar{v}[i/v], \bar{w}[i/w]\big)$ where $\bar{v}[i/v] := (v_1, \dots, v_{i-1}, v, v_{i+1}, \dots, v_k)$ is the tuple obtained from $\bar{v}$ by replacing the i-th entry by v.

  If mapping each v_i to w_i does not define an isomorphism of the induced subgraphs of G and H, Spoiler wins the play. More precisely, Spoiler wins if there are $i, j \in [k]$ such that $v_i = v_j \not\Leftrightarrow w_i = w_j$ or $v_iv_j \in E(G) \not\Leftrightarrow w_iw_j \in E(H)$. If there is no position of the play such that Spoiler wins, then Duplicator wins.

We say that Spoiler (and Duplicator, respectively) *wins* $\mathrm{BP}_k(G, H)$ if Spoiler (and Duplicator, respectively) has a winning strategy for the game. Also, for a position $(\bar{v}, \bar{w})$ with $\bar{v} \in \big(V(G)\big)^k$ and $\bar{w} \in \big(V(H)\big)^k$, we say that Spoiler (and Duplicator, respectively) *wins* $\mathrm{BP}_k(G, H)$ *from position* $(\bar{v}, \bar{w})$ if Spoiler (and Duplicator, respectively) has a winning strategy for the game started at position $(\bar{v}, \bar{w})$.

THEOREM 5.2 ([13, 33]). *Suppose $k \geq 2$ and let G and H be two directed graphs. Then $G \simeq_k H$ if and only if Duplicator wins the game* $\mathrm{BP}_{k+1}(G, H)$.

5.2 CFI Graphs and Tournaments

To prove that the WL algorithm on its own is unable to determine isomorphisms between tournaments of bounded twin width, we adapt a construction of Cai, Fürer and Immerman [13]. Towards this end, we first describe a construction of directed graphs with large WL dimension, and then argue how to translate those graphs into tournaments while preserving their WL dimension.

More concretely, the Cai-Fürer-Immerman (CFI) construction [13] takes an undirected base graph G, and replaces every vertex with a certain gadget, and those gadgets are connected along the edges of G. To obtain a lower bound on the WL dimension that is linear in the number of vertices, one typically uses 3-regular expander graphs as a base graph (more precisely, we require a bounded-degree graph G with tree width that is linear in the number of vertices). In order to translate this construction to tournaments of small twin-width, we need to tackle some hurdles. First of all, in the standard CFI construction, each gadget encodes a linear equation over

$\mathbb{Z}_2$ via its automorphism group. However, the automorphism group of a tournament always has odd order (see the comments before Theorem 2.3). For this reason, we rely on a simple variant of the gadgets that encode linear equations over $\mathbb{Z}_3$. At this point, we can employ a simple idea to translate a CFI graph into a tournament: We fix a linear order $<$ on the vertex set of G (to be specified later), and represent edges of the CFI graph by a "forward" edge in the tournament (i.e., an edge that points to the larger element in the linear order) whereas non-edges are represented by "backward" edges.

To bound the twin width of the resulting tournament, the basic idea to build the contraction sequence is to first contract each of the CFI gadgets to a single vertex. At this point, intuitively speaking, we obtain (the tournament representation of) the trigraph G^{red} obtained from G by replacing every edge with a red edge. So, in order to bound the twin width, we choose G to be a 3-regular expander graph such that G^{red} has small twin width; the existence of such graphs is shown in [6]. Finally, for the contraction sequence of G^{red} to be compatible with the tournament representation described above, we choose the linear order $<$ so that the twin width of $(G^{\mathsf{red}}, <)$ is small; this is possible by Lemma 2.2.

In the following, we formally describe the construction of tournaments with large WL dimension. Let G be a connected, 3-regular undirected graph. Let G^{red} denote the structure obtained from G be replacing every edge with a red edge and let $t := \operatorname{tww}(G^{\mathsf{red}})$. By Lemma 2.2 there is some linear order $<$ on $V(G)$ such that

$$\operatorname{tww}(G^{\mathsf{red}}, <) = t. \tag{3}$$

Also, let $\vec{G}$ be an arbitrary orientation of G.

We start by providing the basic CFI construction (where gadgets encode linear equations over $\mathbb{Z}_3$). We stress that the specific construction described below is heavily tailored towards enabling the later translation into tournaments.

Recall that for every $v \in V(G)$ we denote by $E_+(v)$ the set of outgoing edges and $E_-(v)$ the set of incoming edges in $\vec{G}$. Also, we write $E(v)$ to denote the set of incident (undirected) edges in G. For $a \in \mathbb{Z}_3$ we define

$$M_a(v) := \left\{ f \colon E(v) \to \mathbb{Z}_3 \;\middle|\; \sum_{(v,w)\in E_+(v)} f(\{v,w\}) - \sum_{(w,v)\in E_-(v)} f(\{v,w\}) = a \pmod 3 \right\}$$

We also define $F_a(v)$ to contain all pairs $(f,g) \in (M_a(v))^2$ such that, for the minimal element $w \in N_G(v)$ (with respect to $<$) such that $f(vw) \neq g(vw)$, it holds that

$$f(vw) + 1 = g(vw) \pmod 3.$$

Observe that, for every distinct $f, g \in M_a(v)$, either $(f,g) \in F_a(v)$ or $(g,f) \in F_a(v)$.

Let $\alpha\colon V(G) \to \mathbb{Z}_3$ be a function. We define the graph $\mathsf{CFI}_3(\vec{G}, <, \alpha)$ with vertex set

$$V(\mathsf{CFI}_3(\vec{G}, <, \alpha)) := \bigcup_{v \in V(G)} \{v\} \times M_{\alpha(v)}(v)$$

and edge set

$$\begin{aligned} E(\mathsf{CFI}_3(\vec{G}, <, \alpha)) := \; & \Big\{\{(v, f), (w, g)\} \;\Big|\; vw \in E(G) \wedge f(vw) = g(vw)\Big\} \\ & \cup \Big\{((v, f), (v, g)) \;\Big|\; (f, g) \in F_{\alpha(v)}(v)\Big\}. \end{aligned}$$

Observe that $\mathsf{CFI}_3(\vec{G}, <, \alpha)$ is a *mixed graph*, i.e., it contains both directed and undirected edges. Also, we color the vertices of $\mathsf{CFI}_3(\vec{G}, <, \alpha)$ using the coloring $\lambda\colon V(\mathsf{CFI}_3(\vec{G}, <, \alpha)) \to V(G)$ defined via $\lambda(v, f) := v$ for all $(v, f) \in V(\mathsf{CFI}_3(\vec{G}, <, \alpha))$, i.e., each set $\{v\} \times M_{\alpha(v)}(v)$ forms a color class under λ.

Let us analyze the relevant properties of the graphs $\mathsf{CFI}_3(\vec{G}, <, \alpha)$. We start by investigating isomorphisms between the graphs $\mathsf{CFI}_3(\vec{G}, <, \alpha)$ for different mappings $\alpha\colon V(G) \to \mathbb{Z}_3$.

LEMMA 5.3. *Let $\alpha\colon V(G) \to \mathbb{Z}_3$ be a function and let $v, w \in V(G)$ be distinct vertices. Also let P be a path from v to w in G. Let $\beta\colon V(G) \to \mathbb{Z}_3$ denote the function defined via*

$$\beta(u) := \begin{cases} \alpha(u) + 1 \pmod 3 & \text{if } u = v \\ \alpha(u) - 1 \pmod 3 & \text{if } u = w \\ \alpha(u) & \text{otherwise} \end{cases}.$$

Then there is an isomorphism $\varphi\colon \mathsf{CFI}_3(\vec{G}, <, \alpha) \cong \mathsf{CFI}_3(\vec{G}, <, \beta)$ such that

$$\varphi(u, f) = (u, f)$$

for all $u \in V(G) \setminus V(P)$ and $f \in M_{\alpha(u)}(u)$.

PROOF. Let $E(P)$ denote the set of directed edges on the path P (all edges are directed so that we obtain a path from v to w). For every $u \in V(G)$ we define the function $h_u\colon E(u) \to \mathbb{Z}_3$ via

$$h_u(\{u, u'\}) = \begin{cases} 1 & \text{if } (u, u') \in E(P) \wedge (u, u') \in E(\vec{G}) \\ 2 & \text{if } (u, u') \in E(P) \wedge (u', u) \in E(\vec{G}) \\ 2 & \text{if } (u', u) \in E(P) \wedge (u, u') \in E(\vec{G}) \\ 1 & \text{if } (u', u) \in E(P) \wedge (u', u) \in E(\vec{G}) \\ 0 & \text{otherwise} \end{cases}.$$

We define $\varphi(u, f) := (u, f + h_u)$ for all $(u, f) \in \mathsf{CFI}_3(\vec{G}, <, \alpha)$. It is easy to check that $\varphi\colon \mathsf{CFI}_3(\vec{G}, <, \alpha) \cong \mathsf{CFI}_3(\vec{G}, <, \beta)$ and the desired properties are satisfied. ■

COROLLARY 5.4. *Let $\alpha, \beta \colon V(G) \to \mathbb{Z}_3$ be two functions such that*

$$\sum_{v \in V(G)} \alpha(v) = \sum_{v \in V(G)} \beta(v) \pmod 3.$$

Then $\mathsf{CFI}_3(\vec{G}, <, \alpha) \cong \mathsf{CFI}_3(\vec{G}, <, \beta)$.

PROOF. This follows from repeatedly applying Lemma 5.3 to gradually align the two functions α and β. ■

Now fix an arbitrary vertex $u_0 \in V(G)$. For every $p \in \mathbb{Z}_3$ we define the mapping $\alpha_p \colon V(G) \to \mathbb{Z}_3$ via $\alpha_p(u_0) \coloneqq p$ and $\alpha_p(w) \coloneqq 0$ for all $w \in V(G) \setminus \{u_0\}$.

LEMMA 5.5. $\mathsf{CFI}_3(\vec{G}, <, \alpha_0) \not\cong \mathsf{CFI}_3(\vec{G}, <, \alpha_1)$.

PROOF. Let $X \coloneqq \{(v, \mathbf{0}_{E(v)}) \mid v \in V(G)\}$ where $\mathbf{0}_{E(v)}$ denotes the function that maps every element in $E(v)$ to 0. Note that $(\mathsf{CFI}_3(\vec{G}, <, \alpha_0))[X]$ is isomorphic to G and contains exactly one vertex from each color class.

Now suppose towards a contradiction that $\varphi \colon \mathsf{CFI}_3(\vec{G}, <, \alpha_0) \cong \mathsf{CFI}_3(\vec{G}, <, \alpha_1)$. Let $Y \coloneqq \varphi(X)$. Since φ is an isomorphism, the set Y also contains exactly one vertex from every color class, i.e., for every $v \in V(G)$ there is a unique $f_v \in M_{\alpha_1(v)}(v)$ such that $(v, f_v) \in Y$. Also, $f_v(\{v, w\}) = f_w(\{v, w\})$ for all edges $\{v, w\} \in E(G)$ because $\{(v, f_v), (w, f_w)\} \in E(\mathsf{CFI}_3(\vec{G}, <, \alpha_1))$. This means we can define combine all the mappings f_v, $v \in V(G)$, into one assignment $f \colon E(G) \to \mathbb{Z}_3$ via

$$f(\{v, w\}) \coloneqq f_v(\{v, w\}) = f_w(\{v, w\}).$$

For every $v \in V(G)$ it holds that

$$\sum_{(v,w) \in E_+(v)} f(\{v, w\}) - \sum_{(w,v) \in E_-(v)} f(\{v, w\}) = \alpha_1(v) \pmod 3.$$

So

$$\sum_{v \in V(G)} \left(\sum_{(v,w) \in E_+(v)} f(\{v, w\}) - \sum_{(w,v) \in E_-(v)} f(\{v, w\}) \right) = \sum_{v \in V(G)} \alpha_1(v) \pmod 3.$$

However, we have that

$$\sum_{v \in V(G)} \left(\sum_{(v,w) \in E_+(v)} f(\{v, w\}) - \sum_{(w,v) \in E_-(v)} f(\{v, w\}) \right) = 0 \pmod 3$$

whereas

$$\sum_{v \in V(G)} \alpha_1(v) = 1 \pmod 3.$$

This is a contradiction. ■

Now, we analyze the WL algorithm on the graphs $\mathsf{CFI}_3(\vec{G}, <, \alpha)$ for different functions $\alpha \colon V(G) \to \mathbb{Z}_3$. Similar results have been obtained in prior works; see, e.g., [19, 43].

LEMMA 5.6. *Let k be an integer such that $\operatorname{tw}(G) \geq k+1$. Also let $\alpha, \beta \colon V(G) \to \mathbb{Z}_3$ be two functions. Then $\mathsf{CFI}_3(\vec{G}, <, \alpha) \simeq_k \mathsf{CFI}_3(\vec{G}, <, \beta)$.*

PROOF. Using Corollary 5.4 it suffices to show that $\mathsf{CFI}_3(\vec{G}, <, \alpha_p) \simeq_k \mathsf{CFI}_3(\vec{G}, <, \alpha_q)$ for all $p, q \in \mathbb{Z}_3$. In fact, since $\simeq_k$ is an equivalence relation, we only need to prove that $\mathsf{CFI}_3(\vec{G}, <, \alpha_0) \simeq_k \mathsf{CFI}_3(\vec{G}, <, \alpha_q)$ for all $q \in \{1, 2\}$.

So let us fix $q \in \{1, 2\}$. Since $\operatorname{tw}(G) \geq k+1$, the Robber has a winning strategy in the cops-and-robber game $\mathrm{CopRob}_{k+1}(G)$ by Theorem 5.1. We translate the winning strategy for the Robber in $\mathrm{CopRob}_{k+1}(G)$ into a winning strategy for Duplicator in the $(k+1)$-bijective pebble game played on $\mathsf{CFI}_3(\vec{G}, <, \alpha_0)$ and $\mathsf{CFI}_3(\vec{G}, <, \alpha_q)$. Using Theorem 5.2 this implies that $\mathsf{CFI}_3(\vec{G}, <, \alpha_0) \simeq_k \mathsf{CFI}_3(\vec{G}, <, \alpha_q)$ as desired.

For a tuple $\bar{x} = (x_1, \dots, x_{k+1}) \in (V(\mathsf{CFI}_3(\vec{G}, <, \alpha_0)))^{k+1}$, we define $A(\bar{x}) := (v_1, \dots, v_{k+1})$ where $v_i \in V(G)$ is the unique vertex such that $x_i \in \{v_i\} \times M_0(v_i)$.

We first construct the bijection f for the initialization round. Let $\bar{v} = (v_1, \dots, v_{k+1}) \in (V(G))^{k+1}$, and consider a tuple $\bar{x} = (x_1, \dots, x_{k+1}) \in (V(\mathsf{CFI}_3(\vec{G}, <, \alpha_0)))^{k+1}$ with $A(\bar{x}) = \bar{v}$. Now let u be the vertex chosen by the Robber if the Cops initially place themselves on $A(\bar{x})$. We define $\alpha_{u,q} \colon V(G) \to \mathbb{Z}_3$ via $\alpha_{u,q}(u) := q$ and $\alpha_{u,q}(w) := 0$ for all $w \in V(G) \setminus \{u\}$. Let P be a path from u to u_0 (recall that G is connected), and let φ denote the isomorphism from $\mathsf{CFI}_3(\vec{G}, <, \alpha_{u,q})$ to $\mathsf{CFI}_3(\vec{G}, <, \alpha_q)$ constructed in Lemma 5.3. We set $f(\bar{x}) := (\varphi(x_1), \dots, \varphi(x_{k+1}))$. It is easy to see that this gives a bijection f (we use the same isomorphism φ for all tuples $\bar{x}$ such that $A(\bar{x}) = \bar{v}$). Observe that $x_i \in V(\mathsf{CFI}_3(\vec{G}, <, \alpha_{u,q}))$ since $\alpha_{u,q}(v_i) = 0 = \alpha_0(v_i)$ for all $i \in [k+1]$.

Now, throughout the game, Duplicator maintains the following invariant. Let $(\bar{x}, \bar{y})$ denote the current position. Then there is a vertex $u \in V(G)$ and a bijection $\varphi \colon \mathsf{CFI}_3(\vec{G}, <, \alpha_{u,q}) \to \mathsf{CFI}_3(\vec{G}, <, \alpha_q)$ such that

- $\varphi(\bar{x}) = \bar{y}$,
- u does not appear in the tuple $A(\bar{x})$, and
- the Robber wins from the position $(A(\bar{x}), u)$, i.e., if the Cops are placed on $A(\bar{x})$ and the Robber is on u.

Note that this condition is satisfied by construction after the initialization round. Also observe that Duplicator never looses the game in such a position. Indeed, since u does not appear in the tuple $A(\bar{x})$, the graphs $\mathsf{CFI}_3(\vec{G}, <, \alpha_0)$ and $\mathsf{CFI}_3(\vec{G}, <, \alpha_{u,q})$ are identical when restricted to vertices from $\bar{x}$. Since φ is an isomorphism from $\mathsf{CFI}_3(\vec{G}, <, \alpha_{u,q})$ to $\mathsf{CFI}_3(\vec{G}, <, \alpha_q)$ and $\varphi(\bar{x}) = \bar{y}$, Duplicator does not loose in the position $(\bar{x}, \bar{y})$.

So it remains to show that Duplicator can maintain the above invariant in each round of the $(k+1)$-bijective pebble game. Suppose $(\bar{x}, \bar{y})$ is the current position. Also let $(A(\bar{x}), u)$ be the associated position in the cops-and-robber game. Suppose that $A(\bar{x}) = (v_1, \dots, v_{k+1})$.

Let $i \in [k+1]$ denote the index chosen by Spoiler. We describe the bijection f chosen by Duplicator. Let $v \in V(G)$. Let u' be the vertex the Robber moves to if the Cops choose i and v (i.e., the i-th cop changes its position to v) in the position $(A(\bar{x}), u)$. Let P denote a path from u to u' that avoids $\{v_1, \dots, v_{k+1}\} \setminus \{v_i\}$. Let ψ denote the isomorphism from $\mathsf{CFI}(G, \alpha_{u',q})$ to $\mathsf{CFI}(G, \alpha_{u,q})$ constructed in Lemma 5.3. We set $f(x) := \varphi(\psi(x))$ for all $x \in \{v\} \times M_0(v)$.

It is easy to see that f is a bijection. Let x denote the vertex chosen by Spoiler and let $y := f(x)$. Let $\bar{x}' := \bar{x}[i/x]$ and $\bar{y}' := \bar{y}[i/y]$, i.e., the pair $(\bar{x}', \bar{y}')$ is the new position of the game. Also, we set $\varphi' := \psi \circ \varphi$ where ψ denotes the isomorphism from $\mathsf{CFI}(G, \alpha_{u',q})$ to $\mathsf{CFI}(G, \alpha_{u,q})$ used in the definition of $f(x)$.

We have $\varphi'(x) = y$ by definition. All the other entries of $\bar{x}'$ are fixed by the mapping ψ (see Lemma 5.3) which overall implies that $\varphi'(\bar{x}') = \bar{y}'$. Also, u' does not appear in the tuple $A(\bar{x}')$ by construction, and the Robber wins from the position $(A(\bar{x}'), u')$.

So overall, this means that Duplicator can maintain the above invariant which provides the desired winning strategy. ■

Together, Lemmas 5.5 and 5.6 provide pairs of non-isomorphic graphs that are not distinguished by k-WL assuming $\mathrm{tw}(G) > k$. Next, we argue how to turn these graphs into tournaments.

We define the tournament $T = T(\vec{G}, <, \alpha)$ with vertex set

$$V(T) := V(\mathsf{CFI}_3(\vec{G}, <, \alpha)) = \bigcup_{v \in V(G)} \{v\} \times M_{\alpha(v)}(v)$$

and edge set

$$\begin{aligned} E(T) := \; & \Big\{((v,f)(v,g)) \;\Big|\; (f,g) \in F_{\alpha(v)}(v)\Big\} \\ \cup & \Big\{((v,f)(w,g)) \;\Big|\; v < w \wedge \{(v,f)(w,g)\} \notin E(\mathsf{CFI}_3(\vec{G}, <, \alpha))\Big\} \\ \cup & \Big\{((w,g)(v,f)) \;\Big|\; v < w \wedge \{(v,f)(w,g)\} \in E(\mathsf{CFI}_3(\vec{G}, <, \alpha))\Big\}. \end{aligned}$$

We argue that the relevant properties are preserved by this translation.

LEMMA 5.7. *Let $\alpha, \beta \colon V(G) \to \mathbb{Z}_3$ be two functions. Then $T(\vec{G}, <, \alpha) \cong T(\vec{G}, <, \beta)$ if and only if $\mathsf{CFI}_3(\vec{G}, <, \alpha) \cong \mathsf{CFI}_3(\vec{G}, <, \beta)$.*

PROOF. The backward direction is immediately clear since $T(\vec{G}, <, \alpha)$ and $T(\vec{G}, <, \beta)$ are defined from $\mathsf{CFI}_3(\vec{G}, <, \alpha)$ and $\mathsf{CFI}_3(\vec{G}, <, \beta)$, respectively, in an isomorphism-invariant manner. Note that the linear order $<$ on the vertex set of G is encoded into $\mathsf{CFI}_3(\vec{G}, <, \alpha)$ and $\mathsf{CFI}_3(\vec{G}, <, \beta)$ via the vertex-coloring.

So let $\varphi \colon T(\vec{G}, <, \alpha) \cong T(\vec{G}, <, \beta)$. The following claim provides the key tool.

CLAIM 5.8. *For every $(v, f) \in V(T(\vec{G}, <, \alpha))$ there is some $g \in M_{\beta(v)}(v)$ such that $\varphi(v, f) = (v, g)$.*

Proof. Suppose $V(G) = \{v_1, \dots, v_n\}$ where $v_i < v_j$ for all $i < j \in [n]$. For $i \leq j$ we define

$$X_{i,j} \coloneqq \bigcup_{i \leq \ell \leq j} \{v_\ell\} \times M_{\alpha(v_\ell)}(v_\ell)$$

and

$$Y_{i,j} \coloneqq \bigcup_{i \leq \ell \leq j} \{v_\ell\} \times M_{\beta(v_\ell)}(v_\ell).$$

Also let $X_i \coloneqq X_{i,i}$ and $Y_i \coloneqq Y_{i,i}$. Let $T_\alpha \coloneqq T(\vec{G}, <, \alpha)$ and $T_\beta \coloneqq T(\vec{G}, <, \beta)$.

We argue by induction that $\varphi(X_i) = Y_i$ for all $i \in [n]$. So fix some $i \in [n]$ and suppose that $\varphi(X_j) = Y_j$ for all $j < i$. In particular, $\varphi(X_{i,n}) = Y_{i,n}$.

Let $w \in X_i$. We have

$$|N_-(w) \cap X_{i,n}| = 4 + 3 \cdot |N_G(v_i) \cap \{v_{i+1}, \dots, v_n\}|.$$

Let $\ell \in \{i, \dots, n\}$ such that $\varphi(w) \in Y_\ell$. Then

$$|N_-(\varphi(w)) \cap Y_{i,n}| = 4 + (\ell - i) \cdot 9 - 3 \cdot |N_G(v_\ell) \cap \{v_i, \dots, v_{\ell-1}\}| + 3 \cdot |N_G(v_\ell) \cap \{v_{\ell+1}, \dots, v_n\}|.$$

Moreover, using that $\varphi(X_j) = Y_j$ for all $j < i$, we get that

$$N_G(v_i) \cap \{v_1, \dots, v_{i-1}\} = N_G(v_\ell) \cap \{v_1, \dots, v_{i-1}\}$$

which implies that

$$|N_G(v_i) \cap \{v_i, \dots, v_n\}| = |N_G(v_\ell) \cap \{v_i, \dots, v_n\}|$$

Also

$$|N_-(w) \cap X_{i,n}| = |N_-(\varphi(w)) \cap Y_{i,n}|.$$

All this is only possible if $\ell = i$ using that G is 3-regular. So $\varphi(w) \in Y_i$ as desired. ◆

The claim immediately implies that $\varphi \colon \mathsf{CFI}_3(\vec{G}, <, \alpha) \cong \mathsf{CFI}_3(\vec{G}, <, \beta)$. ■

LEMMA 5.9. *Let $k \geq 2$ be an integer such that $\operatorname{tw}(G) \geq k + 1$. Also let $\alpha, \beta \colon V(G) \to \mathbb{Z}_3$ be two functions. Then $T(\vec{G}, <, \alpha) \simeq_k T(\vec{G}, <, \beta)$.*

PROOF. Let $(\bar{v}, \bar{w})$ be a winning position for Spoiler in the $(k+1)$-bijective pebble game played on $T(\vec{G}, <, \alpha)$ and $T(\vec{G}, <, \beta)$. Then $(\bar{v}, \bar{w})$ is also a winning position for Spoiler in the $(k+1)$-bijective pebble game played on $\mathsf{CFI}_3(\vec{G}, <, \alpha)$ and $\mathsf{CFI}_3(\vec{G}, <, \beta)$. So the lemma follows directly from Lemma 5.6. ■

To prove Theorem 1.2, we also need to bound the twin width of the resulting graph. Recall that $t \coloneqq \operatorname{tww}(G^{\mathsf{red}})$ where G^{red} denotes the version of G where every edge is colored red. Also recall that we defined the linear order $<$ on $V(G)$ in such a way that $t = \operatorname{tww}(G^{\mathsf{red}}) = \operatorname{tww}(G^{\mathsf{red}}, <)$ (see Equation (3)).

LEMMA 5.10. *For every function $\alpha\colon V(G) \to \mathbb{Z}_3$ it holds that* $\operatorname{tww}(T(\vec{G}, <, \alpha)) \leq \max(35, t)$*.*

PROOF. Throughout the proof, we define $M(v) := M_{\alpha(v)}(v)$ for every $v \in V(G)$. Since G is 3-regular, we have that $|M(v)| = 9$ for every $v \in V(G)$. So $|V(T(\vec{G}, <, \alpha))| = 9 \cdot |V(G)|$. Also note that $(M(v), M(w))$ is homogeneous for all distinct $v, w \in V(G)$ such that $\{v, w\} \notin E(G)$.

We construct a partial contraction sequence as follows. Let $n := |V(G)|$. We define $\mathcal{P}_1, \dots, \mathcal{P}_{8n+1}$ arbitrarily such that $\mathcal{P}_{8n+1} = \{M(v) \mid v \in V(G)\}$. Since G is 3-regular and $|M(v)| = 9$ for every $v \in V(G)$, we conclude that $(T(\vec{G}, <, \alpha))/\mathcal{P}_i$ has red degree at most $4 \cdot 9 - 1 = 35$ for every $i \in [8n + 1]$. Now observe that

$$\operatorname{tww}((T(\vec{G}, <, \alpha))/\mathcal{P}_{8n+1}) \leq \operatorname{tww}(G^{\text{red}}, <) = t.$$

It follows that $\operatorname{tww}(T(\vec{G}, <, \alpha)) \leq \max(35, t)$ as desired. ■

With Lemma 5.10 in hand, we apply the construction $T(\vec{G}, <, \alpha)$ to a 3-regular base graph G which has tree width linear in the number of vertices, but the twin width of G^{red} is bounded. The existence of such graphs has already been observed in [6]. More precisely, the following theorem follows from combining the arguments from [6, Lemma 5.1] and the results from [5, 39].

THEOREM 5.11. *There is a family of 3-regular graphs $(G_n)_{n\geq 1}$ such that*

1. $|V(G_n)| = O(n)$,
2. $\operatorname{tww}(G_n^{\text{red}}) \leq 6$ *(where G_n^{red} denotes the version of G_n where all edges are turned into red edges), and*
3. $\operatorname{tw}(G_n) \geq n$

for every $n \geq 1$.

Now, we are ready to give a proof for Theorem 1.2.

THEOREM 5.12. *For every $k \geq 2$ there are non-isomorphic tournaments T_k and T'_k such that*

1. $|V(T_k)| = |V(T'_k)| = O(k)$,
2. $\operatorname{tww}(T_k) \leq 35$ *and* $\operatorname{tww}(T'_k) \leq 35$*, and*
3. $T_k \simeq_k T'_k$.

PROOF. Let $k \geq 2$. Let G_{k+1} be the 3-regular graph obtained from Theorem 5.11. Note that $\operatorname{tw}(G_{k+1}) \geq k + 1$.

We also fix an arbitrary orientation $\vec{G}_{k+1}$ of G_{k+1} and let G_{k+1}^{red} denote the version of G_{k+1} where every edge is replaced by a red edge. We have $t := \operatorname{tww}(G_{k+1}^{\text{red}}) \leq 6$ by Theorem 5.11. By Lemma 2.2 there is some linear order $<$ on $V(G_{k+1})$ such that $\operatorname{tww}(G_{k+1}^{\text{red}}, <) \leq t \leq 6$.

Now fix an arbitrary $u_0 \in V(G_{k+1})$. For $p \in \mathbb{Z}_3$ we define the mapping $\alpha_p\colon V(G_{k+1}) \to \mathbb{Z}_3$ via $\alpha_p(u_0) = p$ and $\alpha_p(w) = 0$ for all $w \in V(G_{k+1}) \setminus \{u_0\}$. We define $T_k := T(\vec{G}_{k+1}, <, \alpha_0)$ and

$T'_k := T(\vec{G}_{k+1}, <, \alpha_1)$. We have $|V(T_k)| = |V(T'_k)| = 9 \cdot |V(G_{k+1})| = O(k)$. Also, $\text{tww}(T_k) \leq 35$ and $\text{tww}(T'_k) \leq 35$ by Lemma 5.10. Finally, $T_k \ncong T'_k$ by Lemmas 5.5 and 5.7, and $T_k \simeq_k T'_k$ by Lemma 5.9. ■

6. Twin Width is Smaller Than Other Widths

In this section, we compare twin width with other natural width parameters of tournaments. If f, g are mappings from (directed) graphs to the natural numbers, we say that f is *functionally smaller* than g on a class C of graphs if for every k there is a k' such that for all graphs $G \in C$, if $g(G) \leq k$ then $f(G) \leq k'$. We write $f \precsim_C g$ to denote that f is functionally smaller than g on C. We omit the subscript $_C$ if C is the class of all digraphs.

Natural width measures for directed graphs are *cut width* [14], *directed path width*, *directed tree width* [36], and *clique width* [18] (see also [48] for width measures on directed graphs). On the class of tournaments twin width turns out to be functionally smaller than all of these. For clique width, it has already been shown in [12] that twin width is functionally smaller than clique width on undirected graphs; the proof easily extends to arbitrary binary relational structures and hence to tournaments.

We start by giving definitions for the other width measures.

Let G be a digraph. For a linear order $\leq$ on $V(G)$ and a vertex $v \in V(G)$, we let $S_{\leq}(v) := \{w \in V(G) \mid w \leq v\}$ be the set of all vertices smaller than or equal to v in $\leq$. Let

$$s_{\leq}(v) := |E_G(S_{\leq}(v), V(G) \setminus S_{\leq}(v))|$$

be the number of edges from $S_{\leq}(v)$ to its complement. The *width* of $\leq$ is $\max_{v \in V(G)} s_{\leq}(v)$, and the *cut width* $\text{ctw}(G)$ is the minimum over the width of all linear orders of $V(G)$.

A *directed path decomposition* of a digraph G is a mapping $\beta\colon [p] \to 2^{V(G)}$, for some $p \in \mathbb{N}$, such that for every vertex $v \in V(G)$ there are $\ell, r \in [p]$ such that

$$v \in \beta(t) \iff \ell \leq t \leq r,$$

and for all edges $(v, w) \in E(G)$ there are $\ell, r \in [p]$ with $\ell \leq r$ such that $v \in \beta(r)$ and $w \in \beta(\ell)$. The sets $\beta(t)$, $t \in [p]$, are the *bags* of the decomposition. The *width* of the decomposition is $\max_{t \in [p]} |\beta(t)| - 1$, and the *directed path width* $\text{dpw}(G)$ is the minimum width of a directed path decomposition of G.

A digraph R is a *rooted directed tree* if there is a vertex $r_0 \in V(R)$ such that for every $t \in V(R)$ there is a unique directed walk from r_0 to t. Note that every rooted directed tree can be obtained from an undirected tree by selecting a root r_0 and directing all edges away from the root. For $t \in V(R)$ we denote by R_t the unique induced subgraph of R rooted at t.

Let G be a digraph. A *directed tree decomposition* of G is a triple (R, β, γ) where R is a rooted directed tree, $\beta\colon V(R) \to 2^{V(G)}$ and $\gamma\colon E(R) \to 2^{V(G)}$ such that

(D.1) $\{\beta(t) \mid t \in V(R)\}$ is a partition of $V(G)$, and

(D.2) for every $(s,t) \in E(R)$ the set $\gamma(s,t)$ is a hitting set for all directed walks that start and end in $\beta(R_t) := \bigcup_{t' \in V(R_t)} \beta(t')$ and contain a vertex outside of $\beta(R_t)$.

For $t \in V(R)$ we define $\Gamma(t) := \beta(t) \cup \bigcup_{(s,s') \in E(t)} \gamma(s,s')$ where $E(t)$ denotes the set of edges incident to t. The *width* of (R,β,γ) is defined as

$$\mathrm{width}(R,\beta,\gamma) := \max_{t \in V(R)} |\Gamma(t)| - 1.$$

The *directed tree width* $\mathrm{dtw}(G)$ is the minimum width of a directed tree decomposition of G.

For the reader's convenience, we include a proof of the following well-known inequalities.

PROPOSITION 6.1. *For all digraphs G, it holds that*

$$\mathrm{dtw}(G) \leq \mathrm{dpw}(G) \leq \mathrm{ctw}(G).$$

PROOF. To prove that $\mathrm{dtw}(G) \leq \mathrm{dpw}(G)$, let $\beta\colon [p] \longrightarrow 2^{V(G)}$ be a directed path decomposition of G of width k. Let R be the rooted directed tree with vertex set $V(R) := [p]$, $E(R) := \{(i,i-1) \mid 2 \leq i \leq p\}$, and root $r_0 := p$. (That is, R is the path $1,\ldots,p$ in reverse order.) We let $\beta'(1) := \beta(1)$, and for $2 \leq i \leq p$, we let $\beta'(i) := \beta(i) \setminus \beta(i-1)$ and $\gamma(i,i-1) := \beta(i) \cap \beta(i-1)$. It is easy to show that (R,β',γ) is a directed tree decomposition of G of width k.

To prove that $\mathrm{dpw}(G) \leq \mathrm{ctw}(G)$, let $\leq$ be a linear order of G of width k. Without loss of generality we may assume that $V(G) = [n]$ and $\leq$ is the natural linear order on $[n]$. For every $v \in [n]$, let $\ell_v := \max\big(\{v\} \cup N_+(v)\big)$. Note that there are at most k vertices $u \leq v$ such that $\ell_u > v$. We define $\beta\colon [n] \longrightarrow 2^{V(G)}$ by $\beta(i) := \{v \in V(G) \mid v \leq i \leq \ell_v\}$. Note that $|\beta(i)| \leq k+1$. It is easy to verify that β is a directed path decomposition of G of width at most k. ■

The proposition implies that $\mathrm{dtw} \lesssim \mathrm{dpw} \lesssim \mathrm{ctw}$. The following example shows that $\mathrm{tww} \not\lesssim \mathrm{ctw}$ and hence $\mathrm{tww} \not\lesssim \mathrm{dpw}$ and $\mathrm{tww} \not\lesssim \mathrm{dtw}$ on the class of all digraphs.

EXAMPLE 6.2. For every undirected graph G, let $\widehat{G}$ be the digraph with vertex set $V(\widehat{G}) := V(G) \cup E(G)$ and edge set

$$E(\widehat{G}) := \big\{(e,v),(e,w) \bigm| e = (v,w) \in E(G)\big\}.$$

Suppose that $V(G) = \{v_1,\ldots,v_n\}$ and $E(G) = \{e_1,\ldots,e_m\}$. Let $\leq$ be the linear order on $V(\widehat{G})$ defined by $v_i \leq v_j$ if and only if $i \leq j$, $v_i \leq e_j$ for all i,j, and $e_i \leq e_j$ if and only if $i \leq j$. Since there are only backward edges, the width of this linear order is 0. Hence, $\mathrm{ctw}(\widehat{G}) = 0$.

However, it is easy to see that the class of all digraphs $\widehat{G}$ for arbitrary G has unbounded twin width. ◆

In contrast to the last example, it turns out that on the class of tournaments, twin width is functionally smaller than directed tree width. Actually, this even holds for the larger class of semi-complete graphs. A digraph G is *semi-complete* if for all distinct $v, w \in V(G)$ at least one of the pairs $(v, w), (w, v)$ is an edge. Note that every tournament is semi-complete. We argue that

$$\operatorname{tww} \lesssim_{\mathcal{S}} \operatorname{dtw}, \tag{4}$$

where $\mathcal{S}$ denotes the class of all semi-complete digraphs.

To prove (4), we rely on the following result which shows that $\operatorname{dpw} \lesssim_{\mathcal{S}} \operatorname{dtw}$.

THEOREM 6.3 ([32, Proposition 5]). *Let G be a semi-complete graph. Then*

$$\operatorname{dpw}(G) \leq 4(\operatorname{dtw}(G) + 2)^2 + 7(\operatorname{dtw}(G) + 2) - 1.$$

Hence, it suffices to show that, on the class of semi-complete digraphs, twin width is functionally smaller than directed path width.

THEOREM 6.4. *Let G be a semi-complete graph. Then* $\operatorname{tww}(G) \leq \operatorname{dpw}(G)$.

PROOF. Let G be a semi-complete graph of directed path width k, and let $\beta\colon [p] \to 2^{V(T)}$ be a directed path decomposition of G of width k. For every $v \in V(G)$, let $r(v) \in [p]$ be maximum such that $v \in \beta(r(v))$. Let $v_1, \dots, v_n$ be an enumeration of $V(G)$ such that for all $i < j$ we have $r(v_i) \leq r(v_j)$.

We define a contraction sequence $\mathcal{P}_1, \dots, \mathcal{P}_n$ of G by contracting the vertices from left to right. Formally, we define $\mathcal{P}_i := \{\{v_1, \dots, v_i\}, \{v_{i+1}\}, \dots, \{v_n\}\}$ for all $i \in [n]$.

Consider the graph $G/\mathcal{P}_i$. Since $P_i := \{v_1, \dots, v_i\}$ is the only non-singleton part in $\mathcal{P}_i$, every red edge in $G/\mathcal{P}_i$ is incident to P_i. Suppose $\{P_i, \{v_j\}\}$ is a red edge in $G/\mathcal{P}_i$ (for some $j > i$). Then there are vertices $v, w \in P_i = \{v_1, \dots, v_i\}$ such that $(v, v_j), (v_j, w) \in E(T)$. Let $r := r(v_i)$. Since $v = v_{i'}$ for some $i' \leq i$, we have $r(v) \leq r \leq r(v_j)$. By the definition of path decompositions, since $(v, v_j) \in E(T)$ this implies $v_j \in \beta(r)$.

So if an edge $\{P_i, \{v_j\}\}$ is red in $G/\mathcal{P}_i$, then $v_j \in \beta(r)$. Since we also have $v_i \in \beta(r)$ and $|\beta(r)| \leq k + 1$, there can be at most k red edges. Hence, the width of $\mathcal{P}_1, \dots, \mathcal{P}_n$ is at most k. ■

Combining Theorems 6.3 and 6.4, we obtain the following corollary.

COROLLARY 6.5. *Let G be a semi-complete graph. Then*

$$\operatorname{tww}(G) \leq 4(\operatorname{dtw}(G) + 2)^2 + 7(\operatorname{dtw}(G) + 2) - 1.$$

Note that the corollary implies (4). We close this section with an example showing that inequality (4) is strict even on tournaments, that is,

$$\operatorname{dtw} \not\lesssim_{\mathcal{T}} \operatorname{tww}$$

where $\mathcal{T}$ denotes the class of all tournaments.

EXAMPLE 6.6. For every $n \geq 1$, let T_n be the tournament obtained by taking a cycle of length $2n+1$ and directing edges along the shorter path. Formally, we set $V(T_n) := \{0, \dots, 2n\}$ and

$$E(T_n) := \big\{(i, i+j \bmod 2n+1) \bigm| i \in \{0, \dots, 2n\}, j \in \{1, \dots, n\}\big\}.$$

Then it is easy to show that $\operatorname{tww}(T_n) \leq 1$, but the directed tree width of the family of all T_n, $n \geq 1$, is unbounded. ◆

7. Conclusion

We prove that the isomorphism problem for classes of tournaments of bounded (or slowly growing) twin width is in polynomial time. Many algorithmic problems that can be solved efficiently on (classes of) tournaments can also be solved efficiently on (corresponding classes of) semi-complete graphs, that is, directed graphs where for every pair (v, w) of vertices at least one of the pairs (v, w), (w, v) is an edge (see, e.g., [46]). Contrary to this, we remark that isomorphism of semi-complete graphs of bounded twin width is GI-complete: we can reduce isomorphism of oriented graphs to isomorphism of semi-complete graphs by replacing each non-edge by a bidirectional edge. This reduction preserves twin with.

Classes of tournaments of bounded twin width are precisely the classes that are considered to be structurally sparse. Formally, these are the classes that are monadically dependent, which means that all set systems definable over the tournaments in such a class have bounded VC dimension. The most natural set systems definable within a tournament are those consisting of the in-neighbors of the vertices and of the out-neighbors of the vertices. Bounded twin width implies that the VC dimension of these two set systems is bounded, but the converse does not hold. It is easy to see that the VC dimensions of the in-neighbors and out-neighbors systems as well as the set system consisting of the mixed neighbors of all edges are within a linear factor of one another. As a natural next step, we may ask if isomorphism of tournaments where the VC-dimension of these systems is bounded is in polynomial time.

References

[1] **Vikraman Arvind, Ilia Ponomarenko, and Grigory Ryabov.** Isomorphism testing of k-spanning tournaments is fixed parameter tractable. *Art Discret. Appl. Math.* 8(2):2, 2025. DOI (2, 3, 10–12)

[2] **László Babai.** Graph isomorphism in quasipolynomial time [extended abstract]. *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 684–697. ACM, 2016. DOI (2, 3)

[3] **László Babai and Eugene M. Luks.** Canonical labeling of graphs. *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, pages 171–183. ACM, 1983. DOI (2, 3, 8, 10)

[4] **Pierre Bergé, Édouard Bonnet, and Hugues Déprés.** Deciding twin-width at most 4 is np-complete. *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, 18:1–18:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. DOI (3)

[5] **Yonatan Bilu and Nathan Linial**. Lifts, discrepancy and nearly optimal spectral gap. *Comb.* 26(5):495–519, 2006. DOI (31)

[6] **Édouard Bonnet**, **Colin Geniet**, **Eun Jung Kim**, **Stéphan Thomassé**, **and Rémi Watrigant**. Twin-width II: small classes. *Comb. Theory*, 2(2), 2022. DOI (2, 25, 31)

[7] **Édouard Bonnet**, **Colin Geniet**, **Eun Jung Kim**, **Stéphan Thomassé**, **and Rémi Watrigant**. Twin-width III: max independent set, min dominating set, and coloring. *SIAM J. Comput.* 53(5):1602–1640, 2024. DOI (2)

[8] **Édouard Bonnet**, **Ugo Giocanti**, **Patrice Ossona de Mendez**, **Pierre Simon**, **Stéphan Thomassé**, **and Szymon Torunczyk**. Twin-width IV: ordered graphs and matrices. *J. ACM*, 71(3):21, 2024. DOI (2)

[9] **Édouard Bonnet**, **Ugo Giocanti**, **Patrice Ossona de Mendez**, **and Stéphan Thomassé**. Twin-width V: linear minors, modular counting, and matrix multiplication. *40th International Symposium on Theoretical Aspects of Computer Science, STACS 2023, March 7-9, 2023, Hamburg, Germany*, volume 254 of *LIPIcs*, 15:1–15:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. DOI (2)

[10] **Édouard Bonnet**, **Eun Jung Kim**, **Amadeus Reinald**, **and Stéphan Thomassé**. Twin-width VI: the lens of contraction sequences. *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 1036–1056. SIAM, 2022. DOI (2)

[11] **Édouard Bonnet**, **Eun Jung Kim**, **Amadeus Reinald**, **Stéphan Thomassé**, **and Rémi Watrigant**. Twin-width and polynomial kernels. *Algorithmica*, 84(11):3300–3337, 2022. DOI (2)

[12] **Édouard Bonnet**, **Eun Jung Kim**, **Stéphan Thomassé**, **and Rémi Watrigant**. Twin-width I: tractable FO model checking. *J. ACM*, 69(1):3:1–3:46, 2022. DOI (2, 5, 6, 32)

[13] **Jin-yi Cai**, **Martin Fürer**, **and Neil Immerman**. An optimal lower bound on the number of variables for graph identification. *Comb.* 12(4):389–410, 1992. DOI (6, 24)

[14] **Maria Chudnovsky**, **Alexandra Ovetsky Fradkin**, **and Paul D. Seymour**. Tournament immersion and cutwidth. *J. Comb. Theory, Ser. B*, 102(1):93–101, 2012. DOI (2, 32)

[15] **Maria Chudnovsky**, **Ringi Kim**, **Chun-Hung Liu**, **Paul D. Seymour**, **and Stéphan Thomassé**. Domination in tournaments. *J. Comb. Theory, Ser. B*, 130:98–113, 2018. DOI (2)

[16] **Maria Chudnovsky**, **Alex Scott**, **and Paul D. Seymour**. Disjoint paths in unions of tournaments. *J. Comb. Theory, Ser. B*, 135:238–255, 2019. DOI (2)

[17] **Maria Chudnovsky and Paul D. Seymour**. A well-quasi-order for tournaments. *J. Comb. Theory, Ser. B*, 101(1):47–53, 2011. DOI (2)

[18] **Bruno Courcelle and Stephan Olariu**. Upper bounds to the clique width of graphs. *Discret. Appl. Math.* 101(1-3):77–114, 2000. DOI (32)

[19] **Anuj Dawar and David Richerby**. The power of counting logics on restricted classes of finite structures. *Computer Science Logic, 21st International Workshop, CSL 2007, 16th Annual Conference of the EACSL, Lausanne, Switzerland, September 11-15, 2007, Proceedings*, volume 4646 of *Lecture Notes in Computer Science*, pages 84–98. Springer, 2007. DOI (27)

[20] **John D. Dixon and Brian Mortimer**. Permutation Groups, volume 163 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1996., pages xii+346. DOI (7)

[21] **Walter Feit and John G. Thompson**. Solvability of groups of odd order. *Pacific J. Math.* 13:775–1029, 1963. (2)

[22] **Fedor V. Fomin and Michal Pilipczuk**. On width measures and topological problems on semi-complete digraphs. *J. Comb. Theory, Ser. B*, 138:78–165, 2019. DOI (2)

[23] **Jakub Gajarský**, **Stephan Kreutzer**, **Jaroslav Nesetril**, **Patrice Ossona de Mendez**, **Michal Pilipczuk**, **Sebastian Siebertz**, **and Szymon Torunczyk**. First-order interpretations of bounded expansion classes. *ACM Trans. Comput. Log.* 21(4):29:1–29:41, 2020. DOI (2)

[24] **Jakub Gajarský**, **Michal Pilipczuk**, **and Szymon Torunczyk**. Stable graphs of bounded twin-width. *LICS '22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science, Haifa, Israel, August 2 - 5, 2022*, 39:1–39:12. ACM, 2022. DOI (2)

[25] **Robert Ganian**, **Filip Pokrývka**, **André Schidler**, **Kirill Simonov**, **and Stefan Szeider**. Weighted model counting with twin-width. *25th International Conference on Theory and Applications of Satisfiability Testing, SAT 2022, August 2-5, 2022, Haifa, Israel*, volume 236 of *LIPIcs*, 15:1–15:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. DOI (2)

[26] **Colin Geniet and Stéphan Thomassé**. First order logic and twin-width in tournaments and dense oriented graphs. *Eur. J. Comb.* 132:104247, 2026. DOI (2)

[27] **Martin Grohe**. Descriptive Complexity, Canonisation, and Definable Graph Structure Theory, volume 47 of *Lecture Notes in Logic*. Cambridge University Press, 2017. DOI (3)

[28] **Martin Grohe and Daniel Neuen**. Canonisation and definability for graphs of bounded rank width. *ACM Trans. Comput. Log.* 24(1):6:1–6:31, 2023. DOI (3)

[29] **Martin Grohe and Daniel Neuen**. Isomorphism for tournaments of small twin width. *51st International Colloquium on Automata, Languages, and Programming, ICALP 2024, Tallinn, Estonia, July 8-12, 2024*, volume 297 of *LIPIcs*, 78:1–78:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. DOI (1)

[30] **Martin Grohe and Daniel Neuen**. Recent advances on the graph isomorphism problem. *Surveys in Combinatorics, 2021: Invited lectures from the 28th British Combinatorial Conference, Durham, UK, July 5-9, 2021*, pages 187–234. Cambridge University Press, 2021. DOI (2)

[31] **Martin Grohe**, **Daniel Neuen**, and **Pascal Schweitzer**. A faster isomorphism test for graphs of small degree. *SIAM J. Comput.* 52(6):S18–1, 2023. DOI (3)

[32] **Frank Gurski**, **Dominique Komander**, **Carolin Rehs**, and **Sebastian Wiederrecht**. Directed width parameters on semicomplete digraphs. *Combinatorial Optimization and Applications - 15th International Conference, COCOA 2021, Tianjin, China, December 17-19, 2021, Proceedings*, volume 13135 of *Lecture Notes in Computer Science*, pages 615–628. Springer, 2021. DOI (34)

[33] **Lauri Hella**. Logical hierarchies in PTIME. *Inf. Comput.* 129(1):1–19, 1996. DOI (24)

[34] **Petr Hlinený and Jan Jedelský**. Twin-width of planar graphs is at most 8, and some related bounds. *SIAM J. Discret. Math.* 39(4):2003–2048, 2025. DOI (2)

[35] **Neil Immerman and Eric Lander**. Describing graphs: a first-order approach to graph canonization. **Alan L. Selman**, editor, *Complexity Theory Retrospective: In Honor of Juris Hartmanis on the Occasion of His Sixtieth Birthday, July 5, 1988*, pages 59–81. Springer New York, New York, NY, 1990. DOI (7)

[36] **Thor Johnson**, **Neil Robertson**, **Paul D. Seymour**, and **Robin Thomas**. Directed tree-width. *J. Comb. Theory, Ser. B*, 82(1):138–154, 2001. DOI (3, 32)

[37] **Sandra Kiefer**. The Weisfeiler-Leman algorithm: an exploration of its power. *ACM SIGLOG News*, 7(3):5–27, 2020. DOI (3)

[38] **Eugene M. Luks**. Isomorphism of graphs of bounded valence can be tested in polynomial time. *J. Comput. Syst. Sci.* 25(1):42–65, 1982. DOI (3, 10)

[39] **Adam W. Marcus**, **Daniel A. Spielman**, and **Nikhil Srivastava**. Interlacing families I: Bipartite Ramanujan graphs of all degrees. *Ann. of Math. (2)*, 182(1):307–325, 2015. DOI (31)

[40] **Gary L. Miller**. Isomorphism of graphs which are pairwise k-separable. *Inf. Control.* 56(1/2):21–33, 1983. DOI (3, 9)

[41] **Jaroslav Nesetril and Patrice Ossona de Mendez**. Structural sparsity. *Russian Math. Surveys*, 71(1):79–107, 2016. DOI (2)

[42] **Jaroslav Nesetril**, **Patrice Ossona de Mendez**, and **Sebastian Siebertz**. Structural properties of the first-order transduction quasiorder. *30th EACSL Annual Conference on Computer Science Logic, CSL 2022, February 14-19, 2022, Göttingen, Germany (Virtual Conference)*, volume 216 of *LIPIcs*, 31:1–31:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. DOI (2)

[43] **Daniel Neuen**. Homomorphism-distinguishing closedness for graphs of bounded tree-width. *41st International Symposium on Theoretical Aspects of Computer Science, STACS 2024, March 12-14, 2024, Clermont-Ferrand, France*, volume 289 of *LIPIcs*, 53:1–53:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. DOI (27)

[44] **Daniel Neuen**. Hypergraph isomorphism for groups with restricted composition factors. *ACM Trans. Algorithms*, 18(3):27:1–27:50, 2022. DOI (3)

[45] **Daniel Neuen**. Parameterized complexity of graph isomorphism testing. *Comput. Sci. Rev.* 60:100918, 2026. DOI (2)

[46] **Michal Pilipczuk**. Tournaments and Optimality: New Results in Parameterized Complexity. PhD thesis, University of Bergen, 2013. (35)

[47] **Ilia N. Ponomarenko**. Polynomial time algorithms for recognizing and isomorphism testing of cyclic tournaments. *Acta Appl. Math.* 29(1-2):139–160, 1992. DOI (2)

[48] **Carolin Rehs**. Comparing and Computing Parameters for Directed Graphs. PhD thesis, University of Düsseldorf, Germany, 2022. (32)

[49] **Joseph J. Rotman**. An Introduction to the Theory of Groups, volume 148 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, fourth edition, 1995., pages xvi+513. DOI (7)

[50] **Pascal Schweitzer**. A polynomial-time randomized reduction from tournament isomorphism to tournament asymmetry. *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, 66:1–66:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. DOI (2)

[51] **Ákos Seress**. Permutation Group Algorithms, volume 152 of *Cambridge Tracts in Mathematics*. Cambridge University Press, Cambridge, 2003., pages x+264. DOI (7, 8)

[52] **Paul D. Seymour and Robin Thomas**. Graph searching and a min-max theorem for tree-width. *J. Comb. Theory, Ser. B*, 58(1):22–33, 1993. DOI (23)

[53] **Stéphan Thomassé**. A brief tour in twin-width (invited talk). *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, 6:1–6:5. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. DOI (2)

[54] **Boris Weisfeiler**. On Construction and Identification of Graphs, volume 558 of *Lecture Notes in Mathematics*. Springer-Verlag, 1976. DOI (6)

[55] **Boris Weisfeiler and Andrei Leman**. The reduction of a graph to canonical form and the algebra which appears therein. *NTI, Series 2*, 1968. English translation by Grigory Ryabov available at `https://www.iti.zcu.cz/wl2018/pdf/wl_paper_translation.pdf` (6)