Packing Arc-Disjoint Cycles in Tournaments

Stéphane Bessy
Université de Montpellier, LIRMM, CNRS, Montpellier, France
bessy@lirmm.fr

Marin Bougeret
Université de Montpellier, LIRMM, CNRS, Montpellier, France
bougeret@lirmm.fr

R. Krithika
Indian Institute of Technology Palakkad, India
krithika@iitpkd.ac.in

Abhishek Sahu
The Institute of Mathematical Sciences, HBNI, Chennai, India
asahu@imsc.res.in

Saket Saurabh
The Institute of Mathematical Sciences, HBNI, Chennai, India
University of Bergen, Bergen, Norway
saket@imsc.res.in

Jocelyn Thiebaut
Université de Montpellier, LIRMM, CNRS, Montpellier, France
thiebaut@lirmm.fr

Meirav Zehavi
Ben-Gurion University, Beersheba, Israel
meiravze@bgu.ac.il

Abstract

A tournament is a directed graph in which there is a single arc between every pair of distinct vertices. Given a tournament T on n vertices, we explore the classical and parameterized complexity of the problems of determining if T has a cycle packing (a set of pairwise arc-disjoint cycles) of size k and a triangle packing (a set of pairwise arc-disjoint triangles) of size k. We refer to these problems as Arc-disjoint Cycles in Tournaments (ACT) and Arc-disjoint Triangles in Tournaments (ATT), respectively. Although the maximization version of ACT can be seen as the linear programming dual of the well-studied problem of finding a minimum feedback arc set (a set of arcs whose deletion results in an acyclic graph) in tournaments, surprisingly no algorithmic results seem to exist for ACT. We first show that ACT and ATT are both NP-complete. Then, we show that the problem of determining if a tournament has a cycle packing and a feedback arc set of the same size is NP-complete. Next, we prove that ACT and ATT are fixed-parameter tractable, they can be solved in $2^{\mathcal{O}(k \log k)} n^{\mathcal{O}(1)}$ time and $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$ time respectively. Moreover, they both admit a kernel with $\mathcal{O}(k)$ vertices. We also prove that ACT and ATT cannot be solved in $2^{o(\sqrt{k})} n^{\mathcal{O}(1)}$ time under the Exponential-Time Hypothesis.

2012 ACM Subject Classification Mathematics of computing → Graph theory; Theory of computation → Complexity classes; Theory of computation → Parameterized complexity and exact algorithms; Theory of computation → Design and analysis of algorithms; Mathematics of computing → Graph algorithms

Keywords and phrases arc-disjoint cycle packing, tournaments, parameterized algorithms, kernelization

Digital Object Identifier 10.4230/LIPIcs.MFCS.2019.27

Related Version This paper is based on the two independent manuscripts `arXiv:abs/1802.06669` and `arXiv:abs/1802.07090`.

44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019).
Editors: Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen; Article No. 27; pp. 27:1–27:14

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Given a (directed or undirected) graph G and a positive integer k, the Disjoint Cycle Packing problem is to determine whether G has k (vertex or arc/edge) disjoint (directed or undirected) cycles. Packing disjoint cycles is a fundamental problem in Graph Theory and Algorithm Design with applications in several areas. Since the publication of the classic Erdős-Pósa theorem in 1965 [21], this problem has received significant scientific attention in various algorithmic realms. In particular, Vertex-Disjoint Cycle Packing in undirected graphs is one of the first problems studied in the framework of parameterized complexity. In this framework, each problem instance is associated with a non-negative integer k called *parameter*, and a problem is said to be *fixed-parameter tractable* (FPT) if it can be solved in $f(k)n^{\mathcal{O}(1)}$ time for some computable function f, where n is the input size. For convenience, the running time $f(k)n^{\mathcal{O}(1)}$ is denoted as $\mathcal{O}^\star(f(k))$. A *kernelization algorithm* is a polynomial-time algorithm that transforms an arbitrary instance of the problem to an equivalent instance of the same problem whose size is bounded by some computable function g of the parameter of the original instance. The resulting instance is called a *kernel* and if g is a polynomial function, then it is called a *polynomial kernel*. A decidable parameterized problem is FPT if and only if it has a kernel (not necessarily of polynomial size). Kernelization typically involves applying a set *reduction rules* to the given instance to produce another instance. A reduction rule is said to be *safe* if it is sound and complete, i.e., applying it to the given instance produces an equivalent instance. In order to classify parameterized problems as being FPT or not, the W-hierarchy is defined: $\mathsf{FPT} \subseteq \mathsf{W}[1] \subseteq \mathsf{W}[2] \subseteq \cdots \subseteq \mathsf{XP}$. It is believed that the subset relations in this sequence are all strict, and a parameterized problem that is hard for some complexity class above FPT in this hierarchy is said to be fixed-parameter intractable. Further details on parameterized algorithms we refer to recent books [16, 19, 24, 26].

Vertex-Disjoint Cycle Packing in undirected graphs is FPT with respect to the solution size k [10, 36] but has no polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ [11]. In contrast, Edge-Disjoint Cycle Packing in undirected graphs admits a kernel with $\mathcal{O}(k \log k)$ vertices (and is therefore FPT) [11]. On directed graphs, these problems have many practical applications (for example in biology [12, 18]) and they have been extensively studied [7, 34]. It turns out that Vertex-Disjoint Cycle Packing and Arc-Disjoint Cycle Packing are equivalent and are W[1]-hard [33, 42]. Therefore, studying these problems on a subclass of directed graphs is a natural direction of research. Tournaments form a mathematically rich subclass of directed graphs with interesting structural and algorithmic properties [6, 38]. Tournaments have several applications in modeling round-robin tournaments and in the study of voting systems and social choice theory [29, 31].

Feedback Vertex Set and Feedback Arc Set are two well-explored algorithmic problems on tournaments. A *feedback vertex (arc) set* is a set of vertices (arcs) whose deletion results in an acyclic graph. Given a tournament, MinFAST and MinFVST are the problems of obtaining a feedback arc set and feedback vertex set of minimum size, respectively. We refer to the corresponding decision version of the problems as FAST and FVST. The optimization problems MinFAST and MinFVST have numerous practical applications in the areas of voting theory [17], machine learning [15], search engine ranking [20] and have been intensively studied in various algorithmic areas. MinFAST and MinFVST are NP-hard [3, 13] while FAST and FVST are FPT when parameterized by the solution size k [4, 23, 25, 31]. Further, FAST has a kernel with $\mathcal{O}(k)$ vertices and FVST has a kernel with $\mathcal{O}(k^{1.5})$ vertices [9, 35]. Surprisingly, the duals (in the linear programming sense) of MinFAST and MinFVST have not been considered in the literature until recently. Any tournament that has a cycle

also has a triangle [7]. Therefore, if a tournament has k vertex-disjoint cycles, then it also has k vertex-disjoint triangles. Thus, Vertex-Disjoint Cycle Packing in tournaments is just packing vertex-disjoint triangles. This problem is NP-hard [8]. A straightforward application of the *colour coding* technique [5] shows that this problem is FPT and a kernel with $\mathcal{O}(k^2)$ vertices is an immediate consequence of the quadratic element kernel known for 3-Set Packing [1]. Recently, a kernel with $\mathcal{O}(k^{1.5})$ vertices was shown for this problem using interesting variants and generalizations of the popular *expansion lemma* [35].

A tournament that has k arc-disjoint cycles need not necessarily have k arc-disjoint triangles. This observation hints that packing arc-disjoint cycles could be significantly harder than packing vertex-disjoint cycles. It also hints that packing arc-disjoint cycles and arc-disjoint triangles in tournaments could be problems of different complexities. This is the starting point of our study. Subsequently, we refer to a set of pairwise arc-disjoint cycles as a *cycle packing* and a set of pairwise arc-disjoint triangles as a *triangle packing*. Given a tournament, MaxACT and MaxATT are the problems of obtaining a maximum set of arc-disjoint cycles and triangles, respectively. We refer to the corresponding decision version of the problems as ACT and ATT. Formally, given a tournament T and a positive integer k, ACT (resp. ATT) is the task of determining if T has k arc-disjoint cycles (resp. triangles). From a structural point of view, the problem of partitioning the arc set of a directed graph into a collection of triangles has been studied for regular tournaments [44], almost regular tournaments [2] and complete digraphs [28]. In this work, we study the classical complexity of MaxACT and MaxATT and the parameterized complexity of ACT and ATT with respect to the solution size (i.e. the number k of cycles/triangles) as parameter.

Our main contributions.

- We prove that MaxATT and MaxACT are NP-hard (Theorems 5 and 7). As a consequence, we also show that ACT and ATT do not admit algorithms with $\mathcal{O}^\star(2^{o(\sqrt{k})})$ running time under the Exponential-Time Hypothesis (Theorem 10). Moreover, deciding if a tournament has a cycle packing and a feedback arc set of the same size is NP-complete (Theorem 9).
- A tournament T has k arc-disjoint cycles if and only if T has k arc-disjoint cycles each of length at most $2k+1$ (Theorem 11).
- ACT can be solved in $\mathcal{O}^\star(2^{\mathcal{O}(k \log k)})$ time (Theorem 17) and admits a kernel with $\mathcal{O}(k)$ vertices (Theorem 16).
- ATT can be solved in $\mathcal{O}^\star(2^{\mathcal{O}(k)})$ time and admits a kernel with $\mathcal{O}(k)$ vertices (Theorem 18).

2 Preliminaries

We denote the set $\{1, 2, \ldots, n\}$ of consecutive integers from 1 to n by $[n]$.

Directed Graphs. A *directed graph* D (or *digraph*) is a pair consisting of a finite set $V(D)$ of *vertices* of D and a set $A(D)$ of *arcs* of D, which are ordered pairs of elements of $V(D)$. For a vertex $v \in V(D)$, its *out-neighbourhood*, denoted by $N^+(v)$, is the set $\{u \in V(D) \colon vu \in A(D)\}$ and its *out-degree*, denoted by $d^+(x)$, is $|N^+(v)|$. For a set F of arcs, $V(F)$ denotes the union of the sets of endpoints of arcs in F. Given a digraph D and a subset X of vertices, we denote by $D[X]$ the digraph induced by the vertices in X. Moreover, we denote by $D \setminus X$ the digraph $D[V(D) \setminus X]$ and say that this digraph is obtained by *deleting* X *from* D.

Paths and Cycles. A *path* P in a digraph D is a sequence $(v_1, \dots, v_k)$ of distinct vertices such that for each $i \in [k-1]$, $v_iv_{i+1} \in A(D)$. The set $\{v_1, \dots, v_k\}$ is denoted by $V(P)$ and the set $\{v_iv_{i+1} : i \in [k-1]\}$ is denoted by $A(P)$. A *cycle* C in D is a sequence $(v_1, \dots, v_k)$ of distinct vertices such that $(v_1, \dots, v_k)$ is a path and $v_kv_1 \in A(D)$. The length of a path or cycle X is the number of vertices in it. A cycle on three vertices is called a *triangle*. A digraph is called a *directed acyclic graph* if it has no cycles. A *feedback arc set* (FAS) is a set of arcs whose deletion results in an acyclic graph. For a digraph D, let minfas(D) denote the size of a minimum FAS of D. Any directed acyclic graph D has an ordering $\sigma(D) = (v_1, \dots, v_n)$ called *topological ordering* of its vertices such that for each $v_iv_j \in A(D)$, $i < j$ holds. Given an ordering σ and two vertices u and v, we write $u <_\sigma v$ if u is before v in σ.

Tournaments. A *tournament* T is a digraph in which for every pair u, v of distinct vertices either $uv \in A(T)$ or $vu \in A(T)$ but not both. In other words, a tournament T on n vertices is an orientation of the complete graph K_n. A tournament T can alternatively be defined by an ordering $\sigma(T) = (v_1, \dots, v_n)$ of its vertices and a set of *backward arcs* $\overleftarrow{A}_\sigma(T)$ (which will be denoted $\overleftarrow{A}(T)$ as the considered ordering is not ambiguous), where each arc $a \in \overleftarrow{A}(T)$ is of the form $v_{i_1}v_{i_2}$ with $i_2 < i_1$. Indeed, given $\sigma(T)$ and $\overleftarrow{A}(T)$, we define $V(T) = \{v_i : i \in [n]\}$ and $A(T) = \overleftarrow{A}(T) \cup \overrightarrow{A}(T)$ where $\overrightarrow{A}(T) = \{v_{i_1}v_{i_2} : (i_1 < i_2)$ and $v_{i_2}v_{i_1} \notin \overleftarrow{A}(T)\}$ is the set of *forward arcs* of T in the given ordering $\sigma(T)$. The pair $(\sigma(T), \overleftarrow{A}(T))$ is called a *linear representation* of the tournament T. A tournament is called *transitive* if it is a directed acyclic graph and a transitive tournament has a unique topological ordering. Given two tournaments T_1, T_2 defined by $\sigma(T_l)$ and $\overleftarrow{A}(T_l)$ with $l \in \{1, 2\}$, we denote by $T = T_1T_2$ the tournament called the *concatenation of* T_1 *and* T_2, where $V(T) = V(T_2) \cup V(T_2)$, $\sigma(T) = \sigma(T_1)\sigma(T_2)$ is the concatenation of the two sequences, and $\overleftarrow{A}(T) = \overleftarrow{A}(T_1) \cup \overleftarrow{A}(T_2)$.

3 NP-hardness of MaxACT and MaxATT

This section contains our main results. We prove the NP-hardness of MaxATT using a reduction from 3-SAT(3). Here, 3-SAT(3) denotes the specific case of 3-SAT where each clause has at most three literals, and each literal appears at most two times positively and exactly one time negatively. In the following, denote by F the input formula of an instance of 3-SAT(3). Let n be the number of its variables and m be the number of its clauses. We may suppose that $n \equiv 3 \pmod 6$. If it is not the case, we can add up to 5 unused variables x with the trivial clause $x \vee \overline{x}$. This operation guarantees us we keep the hypotheses of 3-SAT(3). We can also assume that $m + 1 \equiv 3 \pmod 6$. Indeed, if it not the case, we add 6 new unused variables $x_1, \dots, x_6$ with the 6 trivial clauses $x_i \vee \overline{x_i}$, and the clause $x_1 \vee x_2$. This padding process keep both the 3-SAT(3) structure and $n \equiv 3 \pmod 6$. From F we construct a tournament T which is the concatenation of two tournaments T_v and T_c defined below.

In the following, let f be the reduction that maps an instance F of 3-SAT(3) to a tournament T we describe now.

The variable tournament T_v. For each variable v_i of F, we define a tournament V_i of order 6 as follows: $\sigma_i(V_i) = (r_i, \bar{x}_i, x_i^1, s_i, x_i^2, t_i)$ and $\overleftarrow{A}_\sigma(V_i) = \{s_ir_i, t_ix_i^1\}$. Figure 1 is a representation of one variable gadget V_i. One can notice that the minimum FAS of V_i corresponds exactly to the set of its backward arcs. We now define $V(T_v)$ be the union of the vertex sets of the V_is and we equip T_v with the order $\sigma_1\sigma_2 \dots \sigma_n$. Thus, T_v has $6n$

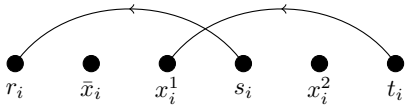

Figure 1 The variable gadget V_i. Only backward arcs are depicted, so all the remaining arcs are forward arcs.

vertices. We also add the following backward arcs to T_v. Since $n \equiv 3 \pmod 6$, there is an edge-disjoint (undirected) triangle packing of K_n covering all its edges with triangles that can be computed in polynomial time [32]. Let $\{u_1, \ldots, u_n\}$ be an arbitrary enumeration of the vertices of K_n. Using a perfect triangle packing Δ_{K_n} of K_n, we create a tournament T_{K_n} such that $\sigma'(T_{K_n}) = (u_1, \ldots, u_n)$ and $\overleftarrow{A}_{\sigma'}(T_{K_n}) = \{u_k u_i : (u_i, u_j, u_k)$ is a triangle of Δ_{K_n} with $i < j < k\}$. Now we set $\overleftarrow{A}_\sigma(T_v) = \{xy : x \in V(V_i), y \in V(V_j)$ for $i \neq j$ and $u_j u_i \in \overleftarrow{A}_{\sigma'}(T_{K_n})\} \cup \bigcup_{i=1}^{n} \overleftarrow{A}_\sigma(V_i)$. In some way, we "blew up" every vertex u_i of T_{K_n} into our variable gadget V_i.

The clause tournament T_c. For each of the m clauses c_j of F, we define a tournament C_j of order 3 as follows: $\sigma(C_j) = (c_j^1, c_j^2, c_j^3)$ and $\overleftarrow{A}_\sigma(C_j) = \emptyset$. In addition, we have a $(m+1)^{th}$ tournament denoted by C_{m+1} and defined by $\sigma(C_{m+1}) = (c_{m+1}^1, c_{m+1}^2, c_{m+1}^3)$ and $\overleftarrow{A}_\sigma(C_{m+1}) = \{c_{m+1}^3 c_{m+1}^1\}$, that is C_{m+1} is a triangle. We call this triangle the *dummy triangle* , and its vertices the *dummy vertices*. We now define T_c such that $\sigma(T_c)$ is the concatenation of each ordering $\sigma(C_j)$ in the natural order, that is $\sigma(T_c) = (c_1^1, c_1^2, c_1^3, \ldots, c_m^1, c_m^2, c_m^3, c_{m+1}^1, c_{m+1}^2, c_{m+1}^3)$. So T_c has $3(m+1)$ vertices. Since $m+1 \equiv 3 \pmod 6$, we use the same trick as above to add arcs to $\overleftarrow{A}_\sigma(T_c)$ coming from a perfect packing of undirected triangles of K_{m+1}. Once again, we "blew up" every vertex u_j of $T_{K_{m+1}}$ into our clause gadget C_j.

The tournament T. To define our final tournament T let us begin with its ordering σ defined by $\sigma(T) = \sigma(T_v)\sigma(T_c)$. Then we construct $\overleftarrow{A}^{vc}(T)$ the backward arcs between T_c and T_v. For any $j \in [m]$, if the clause c_j in F has three literals, that is $c_j = \ell_1 \vee \ell_2 \vee l_3$, then we add to $\overleftarrow{A}^{vc}(T)$ the three backward arcs $c_j^3 z_u$ where $u \in [3]$ and such that $z_u = \bar{x}_{i_u}$ when $\ell_u = \bar{v}_{i_u}$, and $z_u \in \{x_{i_u}^1, x_{i_u}^2\}$ when $\ell_u = v_{i_u}$ in such a way that for any $i \in [n]$, there exists a unique arc $a \in \overleftarrow{A}^{vc}(T)$ with $h(a) = x_i^1$. Informally, in the previous definition, if $x_{i_u}^1$ is already "used" by another clause, we chose $z_u = x_{i_u}^2$. Such an orientation will always be possible since each variable occurs at most two times positively and once negatively in F. If the clause c_j in F has only two literals, that is $c_j = \ell_1 \vee \ell_2$, then we add in $\overleftarrow{A}^{vc}(T)$ the two backward arcs $c_j^2 z_u$ where $u \in [2]$ and such that $z_u = \bar{x}_{i_u}$ when $\ell_u = \bar{v}_{i_u}$ and $z_u \in \{x_{i_u}^1, x_{i_u}^2\}$ when $\ell_u = v_{i_u}$ in such a way that for any $i \in [n]$, there exists a unique arc $a \in \overleftarrow{A}^{vc}(T)$ with $h(a) = x_i^1$. Finally, we add in $\overleftarrow{A}^{vc}(T)$ the backward arcs $c_{m+1}^u \bar{x}_i$ for any $u \in [3]$ and $i \in [n]$. These arcs are called *dummy arcs*. We set $\overleftarrow{A}_\sigma(T) = \overleftarrow{A}_\sigma(T_v) \cup \overleftarrow{A}_\sigma(T_c) \cup \overleftarrow{A}^{vc}(T)$. Notice that each $\bar{x}_i$ has exactly four arcs $a \in \overleftarrow{A}_\sigma(T)$ such that $h(a) = \bar{x}_i$ and $t(a)$ is a vertex of T_c. To finish the construction, notice also that T has $6n+3(m+1)$ vertices and can be computed in polynomial time. Figure 2 is an example of the tournament obtained from a trivial 3-SAT(3) instance.

Now, we move on to proving the correctness of the reduction. First of all, observe that in each variable gadget V_i, there are only four triangles: let δ_i^1, δ_i^2, δ_i^3 and δ_i^4 be the triangles $(r_i, \bar{x}_i, s_i)$, (r_i, x_i^1, s_i), (x_i^1, s_i, t_i) and (x_i^1, x_i^2, t_i), respectively. Moreover, notice that there are only three maximal triangle packings of V_i which are $\{\delta_i^1, \delta_i^3\}$, $\{\delta_i^1, \delta_i^4\}$ and $\{\delta_i^2, \delta_i^4\}$. We call these packings $\Delta_i^\top$, $\Delta_i^{\top'}$ and $\Delta_i^\perp$, respectively.

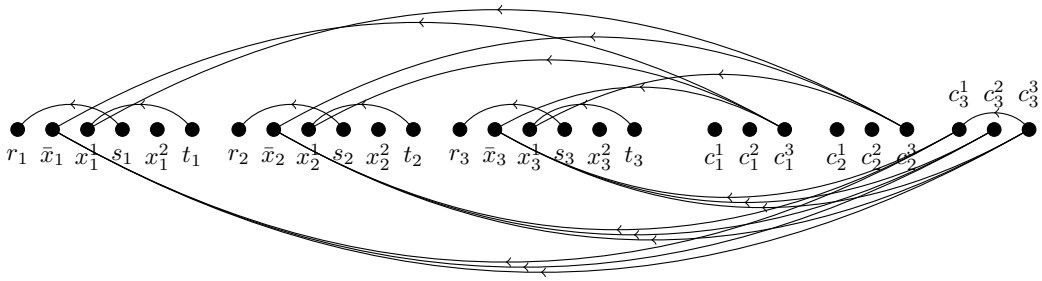

Figure 2 Example of reduction obtained when $F = \{c_1, c_2\}$ where $c_1 = \bar{v}_1 \vee v_2 \vee \bar{v}_3$ and $c_2 = v_1 \vee \bar{v}_2 \vee v_3$. Forward arcs are not depicted. In addition to the depicted backward arcs, we have the 36 backward arcs from V_3 to V_1, and the 9 backward arcs from C_3 to C_1.

Given a triangle packing Δ of T and a subset X of vertices, we define for any $x \in X$ the Δ-local out-degree of the vertex x, denoted $d^+_{X\setminus\Delta}(x)$, as the remaining out-degree of x in $T[X]$ when we remove the arcs of the triangles of Δ. More formally, we set: $d^+_{X\setminus\Delta}(x) = |\{xa\colon a \in X, xa \in A[X], xa \notin A(\Delta)\}|$.

▶ Remark 1. Given a variable gadget V_i, we have:

(i) $d^+_{V_i\setminus\Delta_i^\top}(x_i^1) = d^+_{V_i\setminus\Delta_i^\top}(x_i^2) = 1$ and $d^+_{V_i\setminus\Delta_i^\top}(\bar{x}_i) = 3$,

(ii) $d^+_{V_i\setminus\Delta_i^{\top\prime}}(x_i^1) = 1$, $d^+_{V_i\setminus\Delta_i^{\top\prime}}(x_i^2) = 0$ and $d^+_{V_i\setminus\Delta_i^{\top\prime}}(\bar{x}_i) = 3$,

(iii) $d^+_{V_i\setminus\Delta_i^\perp}(x_i^1) = d^+_{V_i\setminus\Delta_i^\perp}(x_i^2) = 0$ and $d^+_{V_i\setminus\Delta_i^\perp}(\bar{x}_i) = 4$,

(iv) none of $\bar{x}_i x_i^1$, $\bar{x}_i x_i^2$, $\bar{x}_i t_i$ belongs to $\Delta_i^\top$ or $\Delta_i^\perp$.

Informally, we want to set the variable x_i to true (resp. false) when one of the locally-optimal $\Delta_i^{\top\prime}$ or $\Delta_i^\top$ (resp. $\Delta_i^\perp$) is taken in the variable gadget V_i in the global solution. Now given a triangle packing Δ of T, we partition Δ into the following sets:

- $\Delta_{V,V,V} = \{(a,b,c) \in \Delta : a \in V_i,\ b \in V_j,\ c \in V_k \text{ with } i < j < k\}$,
- $\Delta_{V,V,C} = \{(a,b,c) \in \Delta : a \in V_i,\ b \in V_j,\ c \in C_k \text{ with } i < j\}$,
- $\Delta_{V,C,C} = \{(a,b,c) \in \Delta : a \in V_i,\ b \in C_j,\ c \in C_k \text{ with } j < k\}$,
- $\Delta_{C,C,C} = \{(a,b,c) \in \Delta : a \in C_i,\ b \in C_j,\ c \in C_k \text{ with } i < j < k\}$,
- $\Delta_{2V,C} = \{(a,b,c) \in \Delta : a,\ b \in V_i,\ c \in C_j\}$,
- $\Delta_{V,2C} = \{(a,b,c) \in \Delta : a \in V_i, b,\ c \in C_j\}$,
- $\Delta_{3V} = \{(a,b,c) \in \Delta : a,\ b,\ c \in V_i\}$,
- $\Delta_{3C} = \{(a,b,c) \in \Delta : a,\ b,\ c \in C_i\}$.

Notice that in T, there is no triangle with two vertices in a variable gadget V_i and its third vertex in a variable gadget V_j with $i \neq j$ since all the arcs between two variable gadgets are oriented in the same direction. We have the same observation for clauses.
In the two next lemmas, we prove some properties concerning the solution Δ, which imply the result of Lemma 4.

▶ **Lemma 2.** *There exists a triangle packing Δ^v (resp. Δ^c) which uses exactly the arcs between distinct variable gadgets (resp. clause gadgets). Therefore, we have $|\Delta_{V,V,V}| \leq 6n(n-1)$ and $|\Delta_{C,C,C}| \leq 3m(m+1)/2$ and these bounds are tight.*

Proof. First recall that the tournament T_v is constructed from a tournament T_{K_n} which admits a perfect packing of $n(n-1)/6$ triangles. Then we replaced each vertex u_i in T_{K_n} by the variable gadget V_i and kept all the arcs between two variable gadgets V_i

and V_j in the same orientation as between u_i and u_j. Let $u_iu_ju_k$ be a triangle of the perfect packing of T_{K_n}. We temporally relabel the vertices of V_i, V_j and V_k respectively by $\{f_i, i \in [6]\}$, $\{g_i, i \in [6]\}$ and $\{h_i, i \in [6]\}$ and consider the tripartite tournament $K_{6,6,6}$ given by $V(K_{6,6,6}) = \{f_i, g_i, h_i, i \in [6]\}$ and $A(K_{6,6,6}) = \{f_ig_j, g_ih_j, h_if_j : i, j \in [6]\}$. Then it is easy to check that $\{(f_i, g_j, h_{i+j \pmod 6}) : i, j \in [6]\}$ is a perfect triangle packing of $K_{6,6,6}$. Since every triangle of T_{K_n} becomes a $K_{6,6,6}$ in T_v, we can find a triangle packing Δ^v which use all the arcs between disjoint variable gadgets. We use the same reasoning to prove that there exists a triangle packing Δ^c which use all the arcs available in T_c between two distinct clause gadget. ◀

▶ **Lemma 3.** *For any triangle packing Δ of the tournament T, we have:*

(i) $|\Delta_{V,V,V}| + |\Delta_{C,C,C}| \leq 6n(n-1) + 3m(m+1)/2$,

(ii) $|\Delta_{2V,C}| + |\Delta_{V,2C}| + |\Delta_{V,C,C}| + |\Delta_{V,V,C}| \leq |\overleftarrow{A}^{vc}(T)|$,

(iii) $|\Delta_{3V}| \leq 2n$,

(iv) $|\Delta_{3C}| \leq 1$.

Therefore in total we have $|\Delta| \leq 6n(n-1) + 3m(m+1)/2 + 2n + |\overleftarrow{A}^{vc}(T)| + 1$.

Proof. Let Δ be a triangle packing of T. Recall that we have: $|\Delta| = |\Delta_{V,V,V}| + |\Delta_{V,V,C}| + |\Delta_{V,C,C}| + |\Delta_{C,C,C}| + |\Delta_{2V,C}| + |\Delta_{V,2C}| + |\Delta_{3V}| + |\Delta_{3C}|$. First, inequality (i) comes from Lemma 2. Then, we have $|\Delta_{2V,C}| + |\Delta_{V,2C}| + |\Delta_{V,C,C}| + |\Delta_{V,V,C}| \leq |\overleftarrow{A}^{vc}(T)|$ since every triangle of these sets consumes one backward arc from T_c to T_v. We have $|\Delta_{3V}| \leq 2n$ since we have at most 2 disjoint triangles in each variable gadget. Finally we also have $|\Delta_{3C}| \leq 1$ since the dummy triangle is the only triangle lying in a clause gadget. ◀

▶ **Lemma 4.** *F is satisfiable if and only if there exists a triangle packing Δ of size $6n(n-1) + 3m(m+1)/2 + 2n + |\overleftarrow{A}^{vc}(T)| + 1$ in the tournament T.*

As 3-SAT(3) is NP-hard [39, 43], this implies the following theorem.

▶ **Theorem 5.** MaxATT *is NP-hard.*

As mentioned in the introduction, packing arc-disjoint cycles is not necessarily equivalent to packing arc-disjoint triangles. Thus, we need to establish the following lemma to transfer the previous NP-hardness result to MaxACT.

▶ **Lemma 6.** *Given a 3-SAT(3) instance F, and T the tournament constructed from F with the reduction f, we have a triangle packing Δ of T of size $6n(n-1) + 3m(m+1)/2 + 2n + |\overleftarrow{A}^{vc}(T)| + 1$ if and only if there is a cycle packing O of the same size.*

The previous lemma and Theorem 5 imply the following theorem.

▶ **Theorem 7.** MaxACT *is NP-hard.*

Let us now define two special cases Tight-ATT (resp. Tight-ACT) where, given a tournament T and a linear ordering σ with k backward arcs, where $k = \text{minfas}(T)$, the goal is to decide if there is a triangle (resp. cycle) packing of size k. We call these special cases the "tight" versions of the classical packing problems because as the input admits an FAS of size k, any triangle (or cycle) packing has size at most k. We have the following result, directly implying the NP-hardness of Tight-ATT and Tight-ACT.

▶ **Lemma 8.** *Let T be a tournament constructed by the reduction f, and k be the threshold value defined in Lemma 4. Then, we have $k = \text{minfas}(T)$ and we can construct (in polynomial time) an ordering of T with k backward arcs.*

▶ **Theorem 9.** Tight-ATT *and* Tight-ACT *are NP-hard.*

Finally, the size s of the required packing in Lemma 4 satisfies $s = \mathcal{O}((n+m)^2)$. Under the Exponential-time Hypothesis, the problem 3-SAT cannot be solved in $2^{o(n+m)}$ [16, 30]. Then, using the linear reduction from 3-SAT to 3-SAT(3) [43], we also get the following result.

▶ **Theorem 10.** *Under the Exponential-time Hypothesis,* ATT *and* ACT *cannot be solved in* $\mathcal{O}^\star(2^{o(\sqrt{k})})$ *time.*

In the framework of parameterizing above guaranteed values [37], the above results imply that ACT parameterized below the guaranteed value of the size of a minimal feedback arc set is fixed-parameter intractable.

4 Parameterized Complexity of ACT

The classical Erdős-Pósa theorem for cycles in undirected graphs states that for each non-negative integer k, every undirected graph either contains k vertex-disjoint cycles or has a feedback vertex set consisting of $f(k) = \mathcal{O}(k \log k)$ vertices [21]. An interesting consequence of this theorem is that it leads to an FPT algorithm for Vertex-Disjoint Cycle Packing (see [36] for more details).

Analogous to these results, we prove an Erdős-Pósa type theorem for tournaments and show that it leads to an $\mathcal{O}^\star(2^{\mathcal{O}(k \log k)})$ time algorithm and a linear vertex kernel for ACT. First we obtain the following result.

▶ **Theorem 11.** *Let k and r be positive integers such that $r \leq k$. A tournament T contains a set of r arc-disjoint cycles if and only if T contains a set of r arc-disjoint cycles each of length at most $2k+1$.*

Proof. The reverse direction of the claim holds trivially. Let us now prove the forward direction. Let $\mathcal{C}$ be a set of r arc-disjoint cycles in T that minimizes $\sum_{C \in \mathcal{C}} |C|$. If every cycle in $\mathcal{C}$ is a triangle, then the claim trivially holds. Otherwise, let C be a longest cycle in $\mathcal{C}$ and let ℓ denote its length. Let v_i, v_j be a pair of non-consecutive vertices in C. Then, either $v_iv_j \in A(T)$ or $v_jv_i \in A(T)$. In any case, the arc e between v_i and v_j along with $A(C)$ forms a cycle C' of length less than ℓ with $A(C') \setminus \{e\} \subset A(C)$. By our choice of $\mathcal{C}$, this implies that e is an arc in some other cycle $\widehat{C} \in \mathcal{C}$. This property is true for the arc between any pair of non-consecutive vertices in C. Therefore, we have $\binom{\ell}{2} - \ell \leq \ell(k-1)$ leading to $\ell \leq 2k+1$. ◀

This result essentially shows that it suffices to determine the existence of k arc-disjoint cycles in T each of length at most $2k+1$ in order to determine if (T, k) is an yes-instance of ACT. This immediately leads to a quadratic Erdős-Pósa bound. That is, for every non-negative integer k, every tournament T either contains k arc-disjoint cycles or has an FAS of size $\mathcal{O}(k^2)$. Next, we strengthen this result to arrive at a linear bound.

We will use the following lemma known from [14] in order to prove Theorem 13[1]. For a digraph D, let $\Lambda(D)$ denote the number of non-adjacent pairs of vertices in D. That is, $\Lambda(D)$ is the number of pairs u, v of vertices of D such that neither $uv \in A(D)$ nor $vu \in A(D)$.

[1] The authors would like to thank F. Havet for pointing out that Lemma 12 was a consequence of a result by Chudnovsky et al. [14], as well for an improvement of the constant in Theorem 13.

▶ **Lemma 12.** [14] *Let D be a triangle-free digraph in which for every pair u, v of distinct vertices, at most one of uv or vu is in $A(D)$. Then, we can compute an FAS of size at most $\Lambda(D)$ in polynomial time.*

▶ **Theorem 13.** *For every non-negative integer k, every tournament T either contains k arc-disjoint triangles or has an FAS of size at most $5(k-1)$ that can be obtained in polynomial time.*

Proof. Let $\mathcal{C}$ be a maximal set of arc-disjoint triangles in T (that can be obtained greedily in polynomial time). If $|\mathcal{C}| \geq k$, then we have the required set of triangles. Otherwise, let D denote the digraph obtained from T by deleting the arcs that are in some triangle in $\mathcal{C}$. Clearly, D has no triangle and $\Lambda(D) \leq 3(k-1)$. Let F be an FAS of D obtained in polynomial time using Lemma 12. Then, we have $|F| \leq 3(k-1)$. Next, consider a topological ordering σ of $D - F$. Each triangle of $\mathcal{C}$ contains at most 2 arcs which are backward in this ordering. If we denote by F' the set of all the arcs of the triangles of $\mathcal{C}$ which are backward in σ, then we have $|F'| \leq 2(k-1)$ and $(D-F)-F'$ is acyclic. Thus $F^* = F \cup F'$ is an FAS of T satisfying $|F^*| \leq 5(k-1)$. ◀

Next, we show how to obtain a linear kernel for ACT. This kernel is inspired by the linear kernelization described in [9] for FAST and uses Theorem 13. Let T be a tournament on n vertices. First, we apply the following reduction rule.

▶ **Reduction Rule 4.1.** *If a vertex v is in no cycle, then delete v from T.*

This rule is clearly safe as our goal is to find k cycles and v cannot be in any of them. To describe our next rule, we need to state a result by Bessy et al. [9]. An *interval* is a consecutive set of vertices in a linear representation $(\sigma(T), \overleftarrow{A}(T))$ of a tournament T.

▶ **Lemma 14 ([9]).** *Let $T = (\sigma(T), \overleftarrow{A}(T))$ be a tournament on which Reduction Rule 4.1 is not applicable. If $|V(T)| \geq 2|\overleftarrow{A}(T)| + 1$, then there exists a partition $\mathcal{J}$ of $V(T)$ into intervals (that can be computed in polynomial time) such that there are $|\overleftarrow{A}(T) \cap E| > 0$ arc-disjoint cycles using only arcs in E where E denotes the set of arcs in T with endpoints in different intervals.*

Our reduction rule that is based on this lemma is as follows.

▶ **Reduction Rule 4.2.** *Let $T = (\sigma(T), \overleftarrow{A}(T))$ be a tournament on which Reduction Rule 4.1 is not applicable. Let $\mathcal{J}$ be a partition of $V(T)$ into intervals satisfying the properties specified in Lemma 14. Reverse all arcs in $\overleftarrow{A}(T) \cap E$ and decrease k by $|\overleftarrow{A}(T) \cap E|$ where E denotes the set of arcs in T with endpoints in different intervals.*

▶ **Lemma 15.** *Reduction Rule 4.2 is safe.*

Proof. Let T' be the tournament obtained from T by reversing all arcs in $\overleftarrow{A}(T) \cap E$. Suppose T' has $k - |\overleftarrow{A}(T) \cap E|$ arc-disjoint cycles. Then, it is guaranteed that each such cycle is completely contained in an interval. This is due to the fact that T' has no backward arc with endpoints in different intervals. Indeed, if a cycle in T' uses a forward (backward) arc with endpoints in different intervals, then it also uses a back (forward) arc with endpoints in different intervals. It follows that for each arc $uv \in E$, neither uv nor vu is used in these $k - |\overleftarrow{A}(T) \cap E|$ cycles. Hence, these $k - |\overleftarrow{A}(T) \cap E|$ cycles in T' are also cycles in T. Then, we can add a set of $|\overleftarrow{A}(T) \cap E|$ cycles obtained from the second property of Lemma 14 to these $k - |\overleftarrow{A}(T) \cap E|$ cycles to get k cycles in T. Conversely, consider a set of k cycles in

T. As argued earlier, we know that the number of cycles that have an arc that is in E is at most $|\overleftarrow{A}(T) \cap E|$. The remaining cycles (at least $k - |\overleftarrow{A}(T) \cap E|$ of them) do not contain any arc that is in E, in particular, they do not contain any arc from $\overleftarrow{A}(T) \cap E$. Therefore, these cycles are also cycles in T'. ◀

Thus, we have the following result.

▶ **Theorem 16.** ACT *admits a kernel with $\mathcal{O}(k)$ vertices.*

Proof. Let (T, k) denote the instance obtained from the input instance by applying Reduction Rule 4.1 exhaustively. From Lemma 13, we know that either T has k arc-disjoint triangles or has an FAS of size at most $5(k-1)$ that can be obtained in polynomial time. In the first case, we return a trivial yes-instance of constant size as the kernel. In the second case, let F be the FAS of size at most $5(k-1)$ of T. Let $(\sigma(T), \overleftarrow{A}(T))$ be the linear representation of T where $\sigma(T)$ is a topological ordering of the vertices of the directed acyclic graph $T - F$. As $V(T-F) = V(T)$, $|\overleftarrow{A}(T)| \leq 5(k-1)$. If $|V(T)| \geq 10k - 9$, then from Lemma 14, there is a partition of $V(T)$ into intervals with the specified properties. Therefore, Reduction Rule 4.2 is applicable (and the parameter drops by at least 1). When we obtain an instance where neither of the Reduction Rules 4.1 and 4.2 is applicable, it follows that the tournament in that instance has at most $10k$ vertices. ◀

Finally, we show that ACT can be solved in $\mathcal{O}^\star(2^{\mathcal{O}(k \log k)})$ time. The idea is to reduce the problem to the following Arc-Disjoint Paths problem in directed acyclic graphs: given a digraph D on n vertices and k ordered pairs $(s_1, t_1), \ldots, (s_k, t_k)$ of vertices of D, do there exist arc-disjoint paths $P_1, \ldots, P_k$ in D such that P_i is a path from s_i to t_i for each $i \in [k]$? On directed acyclic graphs, Arc-Disjoint Paths is known to be NP-complete [22], W[1]-hard [42] with respect to k as parameter and solvable in $n^{\mathcal{O}(k)}$ time [27]. Despite its fixed-parameter intractability, we will show that we can use the $n^{\mathcal{O}(k)}$ algorithm and Theorems 13 and 16 to describe an FPT algorithm for ACT.

▶ **Theorem 17.** ACT *can be solved in $\mathcal{O}^\star(2^{\mathcal{O}(k \log k)})$ time.*

Proof. Consider an instance (T, k) of ACT. Using Theorem 16, we obtain a kernel $\mathcal{I} = (\widehat{T}, \widehat{k})$ such that $\widehat{T}$ has $\mathcal{O}(k)$ vertices. Further, $\widehat{k} \leq k$. By definition, (T, k) is an yes-instance if and only if $(\widehat{T}, \widehat{k})$ is an yes-instance. Using Theorem 13, we know that $\widehat{T}$ either contains $\widehat{k}$ arc-disjoint triangles or has an FAS of size at most $5(\widehat{k}-1)$ that can be obtained in polynomial time. If Theorem 13 returns a set of $\widehat{k}$ arc-disjoint triangles in $\widehat{T}$, then we declare that (T, k) is an yes-instance.

Otherwise, let $\widehat{F}$ be the FAS of size at most $5(\widehat{k}-1)$ returned by Theorem 13. Let D denote the (acyclic) digraph obtained from $\widehat{T}$ by deleting $\widehat{F}$. Observe that D has $\mathcal{O}(k)$ vertices. Suppose $\widehat{T}$ has a set $\mathcal{C} = \{C_1, \ldots, C_{\widehat{k}}\}$ of $\widehat{k}$ arc-disjoint cycles. For each $C \in \mathcal{C}$, we know that $A(C) \cap \widehat{F} \neq \emptyset$ as $\widehat{F}$ is an FAS of $\widehat{T}$. We can guess that subset F of $\widehat{F}$ such that $F = \widehat{F} \cap A(\mathcal{C})$. Then, for each cycle $C_i \in \mathcal{C}$, we can guess the arcs F_i from F that it contains and also the order π_i in which they appear. This information is captured as a partition $\mathcal{F}$ of F into $\widehat{k}$ sets, F_1 to $F_{\widehat{k}}$ and the set $\{\pi_1, \ldots, \pi_{\widehat{k}}\}$ of permutations where π_i is a permutation of F_i for each $i \in [\widehat{k}]$. Any cycle C_i that has $F_i \subseteq F$ contains a (v, x)-path between every pair (u, v), (x, y) of consecutive arcs of F_i with arcs from $A(D)$. That is, there is a path from $\mathrm{h}(\pi_i^{-1}(j))$ and $\mathrm{t}(\pi_i^{-1}((j+1) \mod |F_i|))$ with arcs from D for each $j \in [|F_i|]$. The total number of such paths in these $\widehat{k}$ cycles is $\mathcal{O}(|F|)$ and the arcs of these paths are contained in D which is a (simple) directed acyclic graph.

The number of choices for F is $2^{|\widehat{F}|}$ and the number of choices for a partition $\mathcal{F} = \{F_1, \ldots, F_{\widehat{k}}\}$ of F and a set $X = \{\pi_1, \ldots, \pi_{\widehat{k}}\}$ of permutations is $2^{\mathcal{O}(|\widehat{F}| \log |\widehat{F}|)}$. Once such a choice is made, the problem of finding $\widehat{k}$ arc-disjoint cycles in $\widehat{T}$ reduces to the problem of finding $\widehat{k}$ arc-disjoint cycles $\mathcal{C} = \{C_1, \ldots, C_{\widehat{k}}\}$ in $\widehat{T}$ such that for each $1 \leq i \leq \widehat{k}$ and for each $1 \leq j \leq |F_i|$, C_i has a path P_{ij} between $\mathrm{h}(\pi_i^{-1}(j))$ and $\mathrm{t}(\pi_i^{-1}((j+1) \mod |F_i|))$ with arcs from $D = \widehat{T} - \widehat{F}$. This problem is essentially finding $r = \mathcal{O}(|\widehat{F}|)$ arc-disjoint paths in D and can be solved in $|V(D)|^{\mathcal{O}(r)}$ time using the algorithm in [27]. Therefore, the overall running time of the algorithm is $\mathcal{O}^\star(2^{\mathcal{O}(k \log k)})$ as $|V(D)| = \mathcal{O}(k)$ and $r = \mathcal{O}(k)$. ◀

5 Parameterized Complexity of ATT

It is easy to obtain an $\mathcal{O}^\star(2^{\mathcal{O}(k)})$ time algorithm using the classical colour coding technique [5] for packing subgraphs of bounded size, and in particular for ATT. Moreover, using matching techniques, we also provide a kernel with a linear number of vertices.

In this section, we provide an FPT algorithm and a linear vertex kernel for ATT. First, it is easy to obtain an $\mathcal{O}^\star(2^{\mathcal{O}(k)})$ time algorithm using the classical colour coding technique [5] for packing subgraphs of bounded size.

▶ **Theorem 18.** ATT *can be solved in* $\mathcal{O}^\star(2^{\mathcal{O}(k)})$ *time.*

Proof. Consider an instance $\mathcal{I} = (T, k)$ of ATT. Let n denote $|V(T)|$ and m denote $|A(T)|$. Let $\mathcal{F}$ denote the family of colouring functions $c : A(T) \to [3k]$ of size $2^{\mathcal{O}(k)} \log^2 m$ that can be computed in $\mathcal{O}^\star(2^{\mathcal{O}(k)})$ time using $3k$-perfect family of hash functions [41]. For each colouring function c in $\mathcal{F}$, we colour $A(T)$ according to c and find a triangle packing of size k whose arcs use different colours. We use a standard dynamic programming routine to finding such a triangle packing. Clearly, if $\mathcal{I}$ is an yes-instance and $\mathcal{C}$ is a set of k arc-disjoint triangles in T, there is a colouring function in $\mathcal{F}$ that colours the $3k$ arcs in these triangles with distinct colours and our algorithm will find the required triangle packing. Given a colouring $c \in \mathcal{F}$, we first compute for every set of 3 colours $\{a, b, c\}$ whether the arcs coloured with a, b or c induce a triangle using 3 different colours or not. Then, for every set S of $3(p+1)$ colours with $p \in [k-1]$, we recursively test if the arcs coloured with the colours in S induce $p+1$ arc-disjoint triangles whose arcs use all the colours of S. This is achieved by iterating over every subset $\{a, b, c\}$ of S and checking if there is a triangle using colours a, b and c and a collection of p arc-disjoint triangles whose arcs use all the colours of $S \setminus \{a, b, c\}$. For a given S, we can find this collection of triangles in $\mathcal{O}(p^3) = \mathcal{O}(k^3)$ time. Therefore, the overall running time of the algorithm is $\mathcal{O}^\star(2^{\mathcal{O}(k)})$. ◀

Next, we show that ATT has a linear vertex kernel.

▶ **Theorem 19.** ATT *admits a kernel with* $\mathcal{O}(k)$ *vertices.*

Proof. Let $\mathcal{X}$ be a maximal collection of arc-disjoint triangles of a tournament T obtained greedily. Let $V_{\mathcal{X}}$ denote the vertices of the triangles in $\mathcal{X}$ and $A_{\mathcal{X}}$ denote the arcs of $V_{\mathcal{X}}$. Let U be the remaining vertices of $V(T)$, i.e., $U = V(T) \setminus V_{\mathcal{X}}$. If $|\mathcal{X}| \geq k$, then (T, k) is an yes-instance of ATT. Otherwise, $|\mathcal{X}| < k$ and $|V_{\mathcal{X}}| < 3k$. Moreover, notice that $T[U]$ is acyclic and T does not contain a triangle with one vertex in $V_{\mathcal{X}}$ and two in vertices in U (otherwise $\mathcal{X}$ would not be maximal).

Let B be the (undirected) bipartite graph defined by $V(B) = A_{\mathcal{X}} \cup U$ and $E(B) = \{au \colon a \in A_{\mathcal{X}}, u \in U$ such that $(t(a), h(a), u)$ forms a triangle in $T\}$. Let M be a maximum matching of B and A' (resp. U') denote the vertices of $A_{\mathcal{X}}$ (resp. U) covered by M. Define $\overline{A'} = A_{\mathcal{X}} \setminus A'$ and $\overline{U'} = U \setminus U'$.

We now prove that $(V_{\mathcal{X}} \cup U', k)$ is a linear kernel of (T, k). Let $\mathcal{C}$ be a maximum sized triangle packing that minimizes the number of vertices of $\overline{U'}$ belonging to a triangle of $\mathcal{C}$. By previous remarks, we can partition $\mathcal{C}$ into $C_{\mathcal{X}} \cup F$ where $C_{\mathcal{X}}$ are the triangles of $\mathcal{C}$ included in $T[V_{\mathcal{X}}]$ and F are the triangles of $\mathcal{C}$ containing one vertex of U and two vertices of $V_{\mathcal{X}}$. It is clear that F corresponds to a union of vertex-disjoint stars of B with centres in U. Denote by $U[F]$ the vertices of U clause gadget g to a triangle of F. If $U[F] \subseteq U'$ then $(V_{\mathcal{X}} \cup U', k)$ is immediately a kernel. Suppose there exists a vertex x_0 such that $x_0 \in U[F] \cap \overline{U'}$.

We will build a tree rooted in x_0 with edges alternating between F and M. For this let $H_0 = \{x_0\}$ and construct recursively the sets H_{i+1} such that

$$H_{i+1} = \begin{cases} N_F(H_i) \text{ if } i \text{ is even,} \\ N_M(H_i) \text{ if } i \text{ is odd,} \end{cases}$$

where, given a subset $S \subseteq U$, $N_F(S) = \{a \in A_{\mathcal{X}} \colon \exists s \in S \text{ s.t. } (t(a), h(a), s) \in F \text{ and } as \notin M\}$ and given a subset $S \subseteq A_{\mathcal{X}}$, $N_M(S) = \{u \in U \colon \exists a \in A_{\mathcal{X}} \text{ s.t. } au \in M\}$. Notice that $H_i \subseteq U$ when i is even and that $H_i \subseteq A_{\mathcal{X}}$ when i is odd, and that all the H_i are distinct as F is a union of disjoint stars and M a matching in B. Moreover, for $i \geq 1$ we call T_i the set of edges between H_i and H_{i-1}. Now we define the tree T such that $V(T) = \bigcup_i H_i$ and $E(T) = \bigcup_i T_i$. As T_i is a matching (if i is even) or a union of vertex-disjoint stars with centres in H_{i-1} (if i is odd), it is clear that T is a tree.

For i being odd, every vertex of H_i is incident to an edge of M otherwise B would contain an augmenting path for M, a contradiction. So every leaf of T is in U and incident to an edge of M in T and T contains as many edges of M than edges of F. Now for every arc $a \in A_{\mathcal{X}} \cap V(T)$ we replace the triangle of $\mathcal{C}$ containing a and corresponding to an edge of F by the triangle $(t(a), h(a), u)$ where $au \in M$ (and au is an edge of T). This operation leads to another collection of arc-disjoint triangles with the same size as $\mathcal{C}$ but containing a strictly smaller number of vertices in $\overline{U'}$, yielding a contradiction.

Finally $V_{\mathcal{X}} \cup U'$ can be computed in polynomial time and we have $|V_{\mathcal{X}} \cup U'| \leq |V_{\mathcal{X}}| + |M| \leq 2|V_{\mathcal{X}}| \leq 6k$, which proves that the kernel has $\mathcal{O}(k)$ vertices. ◀

6 Concluding Remarks

In this work, we studied the classical and parameterized complexity of packing arc-disjoint cycles and triangles in tournaments. We showed NP-hardness, fixed-parameter tractability and linear kernelization results. An interesting problem could be to find subclasses of tournaments where these problems are polynomial-time solvable. For instance, we show in the full version of the paper that it is the case for sparse tournaments, that is for tournaments which admit an FAS that is a matching. This class of tournaments is worthy of attention for these packing problems as packing vertex-disjoint triangles (and hence cycles) in sparse tournaments is NP-complete [8]. To conclude, observe that very few problems on tournaments are known to admit an $\mathcal{O}^\star(2^{\sqrt{k}})$-time algorithm when parameterized by the standard parameter k [40] - FAST is one of them [4, 23]. To the best of our knowledge, outside bidimensionality theory, there are no packing problems that are known to admit such subexponential algorithms. In light of the $2^{o(\sqrt{k})}$ lower bound shown for ACT and ATT, it would be interesting to explore if these problems admit $\mathcal{O}^\star(2^{\mathcal{O}(\sqrt{k})})$ algorithms.

References

1 F. N. Abu-Khzam. An Improved Kernelization Algorithm for r-Set Packing. *Inf. Process. Lett.*, 110(16):621–624, 2010.

2 I. Akaria and R. Yuster. Packing Edge-Disjoint Triangles in Regular and Almost Regular Tournaments. *Discrete Math.*, 338(2):217–228, 2015.

3 N. Alon. Ranking Tournaments. *SIAM J. Discrete Math.*, 20(1):137–142, 2006.

4 N. Alon, D. Lokshtanov, and S. Saurabh. Fast FAST. In *36th International Colloquium on Automata, Languages, and Programming (ICALP) Part I*, pages 49–58, 2009.

5 N. Alon, R. Yuster, and U. Zwick. Color-Coding. *J. ACM*, 42(4):844–856, 1995.

6 J. Bang-Jensen and G. Gutin. Paths, Trees and Cycles in Tournaments. *Congressus Numerantium*, 115:131–170, 1996.

7 J. Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms and Applications*. Springer-Verlag London, 2009.

8 S. Bessy, M. Bougeret, and J. Thiebaut. Triangle Packing in (Sparse) Tournaments: Approximation and Kernelization. In *25th Annual European Symp. on Algorithms (ESA 2017)*, volume 87, pages 14:1–14:13, 2017.

9 S. Bessy, F. V. Fomin, S. Gaspers, C. Paul, A. Perez, S. Saurabh, and S. Thomassé. Kernels for Feedback Arc Set in Tournaments. *J. Comput. Syst. Sci*, 77(6):1071–1078, 2011.

10 H. L. Bodlaender. On Disjoint Cycles. *Int. J. Found. Comput. S.*, 5(1):59–68, 1994.

11 H. L. Bodlaender, S. Thomassé, and A. Yeo. Kernel Bounds for Disjoint Cycles and Disjoint Paths. *Theor. Comput. Sci.*, 412(35):4570–4578, 2011.

12 A. Caprara, A. Panconesi, and R. Rizzi. Packing Cycles in Undirected Graphs. *J. Algorithms*, 48(1):239–256, 2003.

13 P. Charbit, S. Thomassé, and A. Yeo. The Minimum Feedback Arc Set Problem is NP-hard for Tournaments. *Comb Probab Comput.*, 16(1):1–4, 2007.

14 M. Chudnovsky, P. Seymour, and B. Sullivan. Cycles in Dense Digraphs. *Combinatorica*, 28(1):1–18, 2008.

15 W. W. Cohen, R. E. Schapire, and Y. Singer. Learning to Order Things. *Journal of Artificial Intelligence Research*, 10:243–270, 1999.

16 M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.

17 Jean-Charles de Borda. Mémoire sur les élections au scrutin. *Histoire de l'Académie Royale des Sciences*, 1781.

18 D. Dorninger. Hamiltonian Circuits Determining the Order of Chromosomes. *Discrete Appl. Math.*, 50(2):159–168, 1994.

19 R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Springer-Verlag London, 2013.

20 C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank Aggregation Methods for the Web. In *10th International World Wide Web Conference*, pages 613–622, 2001.

21 P. Erdős and L. Pósa. On Independent Circuits Contained in a Graph. *Canadian J. Math.*, 17:347–352, 1965.

22 S. Even, A. Itai, and A. Shamir. On the Complexity of Timetable and Multicommodity Flow Problems. *SIAM J. Comput.*, 5(4):691–703, 1976.

23 U. Feige. Faster FAST(Feedback Arc Set in Tournaments), 2009. `arXiv:0911.5094`.

24 J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006.

25 F. V. Fomin, D. Lokshtanov, V. Raman, and S. Saurabh. Fast Local Search Algorithm for Weighted Feedback Arc Set in Tournaments. In *24th AAAI Conf. on Artificial Intelligence*, pages 65–70, 2010.

26 F. V. Fomin, D. Lokshtanov, S. Saurabh, and M. Zehavi. *Kernelization: Theory of Parameterized Preprocessing*. Cambridge University Press, 2019.

27 S. Fortune, J. Hopcroft, and J. Wyllie. The Directed Subgraph Homeomorphism Problem. *Theor. Comput. Sci.*, 10(2):111–121, 1980.

28 R. B. Gardner. Optimal Packings and Coverings of the Complete Directed Graph with 3-Circuits and with Transitive Triples. In *28th Southeastern International Conference on Combinatorics, Graph Theory and Computing*, volume 127, pages 161–170, 1997.

29 E. Hemaspaandra, H. Spakowski, and J. Vogel. The Complexity of Kemeny Elections. *Theor. Comput. Sci.*, 349(3):382–391, 2005.

30 R. Impagliazzo, R. Paturi, and F. Zane. Which Problems Have Strongly Exponential Complexity? *J. Comput. Syst. Sci*, 63(4):512–530, 2001.

31 M. Karpinski and W. Schudy. Faster Algorithms for Feedback Arc Set Tournament, Kemeny Rank Aggregation and Betweenness Tournament. In *21st International Symp. on Algorithms and Computation (ISAAC)*, pages 3–14, 2010.

32 T. P. Kirkman. On a Problem in Combinations. *Cambridge and Dublin Mathematical Journal*, 2:191–204, 1847.

33 M. Krivelevich, Z. Nutov, M. R. Salavatipour, J. V. Yuster, and R. Yuster. Approximation Algorithms and Hardness Results for Cycle Packing Problems. *ACM Transactions on Algorithms*, 3(4), 2007.

34 M. Krivelevich, Z. Nutov, and R. Yuster. Approximation Algorithms for Cycle Packing Problems. In *16th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 556–561, 2005.

35 T. Le, D. Lokshtanov, S. Saurabh, S. Thomassé, and M. Zehavi. Subquadratic Kernels for Implicit 3-Hitting Set and 3-Set Packing Problems. In *29th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 331–342, 2018.

36 D. Lokshtanov, A. Mouawad, S. Saurabh, and M. Zehavi. Packing Cycles Faster Than Erdős-Pósa. In *44th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 71:1–71:15, 2017.

37 M. Mahajan and V. Raman. Parameterizing Above Guaranteed Values: MaxSat and MaxCut. *J. Algorithms*, 31(2):335–354, 1999.

38 J.W. Moon. *Topics on Tournaments*. Holt, Rinehart and Winston, New York, 1968.

39 C. H. Papadimitriou. *Computational Complexity*. John Wiley and Sons Ltd., 2003.

40 M. Pilipczuk. *Tournaments and Optimality: New Results in Parameterized Complexity*. PhD thesis, The University of Bergen, 2013.

41 J. P. Schmidt and A. Siegel. The Spatial Complexity of Oblivious k-Probe Hash Functions. *SIAM J. Comput.*, 19(5):775–786, 1990.

42 A. Slivkins. Parameterized Tractability of Edge-Disjoint Paths on Directed Acyclic Graphs. *SIAM J. Discrete Math.*, 24(1):146–157, 2010.

43 C. A. Tovey. A simplified NP-complete satisfiability problem. *Discrete Appl. Math.*, 8(1):85–89, 1984.

44 R. Yuster. Packing Triangles in Regular Tournaments. *J. of Graph Theory*, 74(1):58–66, 2013.