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 and 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 $2^{\mathcal{O}(\sqrt{k})} n^{\mathcal{O}(1)}$ time under the Exponential-Time Hypothesis.

2012 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity

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

* This paper is based on the two independent manuscripts [9] and [34]. The full version of this extended abstract containing the detailed proofs is appended for the convenience of the reader.

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:37
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

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 [22], 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: $\mathsf{FPT} \subseteq \mathsf{W}[1] \subseteq \mathsf{W}[2] \subseteq \ldots \subseteq \mathsf{XP}$. 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 can be found in [17, 20, 25, 27].

Vertex-Disjoint Cycle Packing in undirected graphs is FPT with respect to the solution size k [11, 38] but has no polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ [12]. In contrast, Edge-Disjoint Cycle Packing in undirected graphs admits a kernel with $\mathcal{O}(k \log k)$ vertices (and is therefore FPT) [12]. On directed graphs, these problems have many practical applications (for example in biology [13, 19]) and they have been extensively studied [7, 36]. It turns out that Vertex-Disjoint Cycle Packing and Arc-Disjoint Cycle Packing are equivalent and are W[1]-hard [35, 43]. 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, 40]. Tournaments have several applications in modeling round-robin tournaments and in the study of voting systems and social choice theory [30, 32].

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 [18], machine learning [16], search engine ranking [21] and have been intensively studied in various algorithmic areas. MinFAST and MinFVST are NP-hard [3, 14] while FAST and FVST are FPT when parameterized by the solution size k [4, 24, 26, 32]. Further, FAST has a kernel with $\mathcal{O}(k)$ vertices [10] and FVST has a kernel with $\mathcal{O}(k^{1.5})$ vertices

[37]. 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* [37].

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 [45], almost regular tournaments [2] and complete digraphs [29]. 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 4 and 6). 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 9). Moreover, deciding if a tournament has a cycle packing and a feedback arc set of the same size is NP-complete (Theorem 8).
- 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 10).
- ACT can be solved in $\mathcal{O}^\star(2^{\mathcal{O}(k \log k)})$ time (Theorem 16) and admits a kernel with $\mathcal{O}(k)$ vertices (Theorem 15).
- ATT can be solved in $\mathcal{O}^\star(2^{\mathcal{O}(k)})$ time and admits a kernel with $\mathcal{O}(k)$ vertices (Theorem 17).

2 Preliminaries

We denote the set $\{1, 2, \dots, n\}$ 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 $(v_1, \ldots, v_k)$ of distinct vertices such that for each $i \in [k-1]$, $v_i v_{i+1} \in A(D)$. The set $\{v_1, \ldots, v_k\}$ is denoted by $V(P)$ and the set $\{v_i v_{i+1} : i \in [k-1]\}$ is denoted by $A(P)$. A *cycle* C in D is a sequence $(v_1, \ldots, v_k)$ of distinct vertices such that $(v_1, \ldots, v_k)$ is a path and $v_k v_1 \in A(D)$. 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 $\text{minfas}(D)$ denote the size of a minimum FAS of D. Any directed acyclic graph D has an ordering $\sigma(D) = (v_1, \ldots, v_n)$ called *topological ordering* of its vertices such that for each $v_i v_j \in A(D)$, $i < j$ 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 $\sigma(T) = (v_1, \ldots, v_n)$ 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 $\overrightarrow{A}(T) = \{v_{i_1} v_{i_2} : (i_1 < i_2) \text{ and } v_{i_2} v_{i_1} \notin \overleftarrow{A}(T)\}$ 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 $T = T_1 T_2$ 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). Recall that 3-SAT(3) corresponds to 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 $x_1, \ldots, x_6$ 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 $\overleftarrow{A}_\sigma(V_i) = \{s_i r_i, t_i x_i^1\}$. 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 $\sigma_1 \sigma_2 \ldots \sigma_n$. Thus, T_v has $6n$ vertices. We also add the following backward arcs to T_v. Since $n \equiv 3 \pmod 6$, there is an

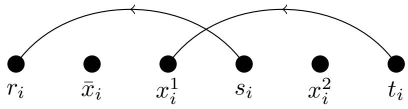

Figure 1 The variable gadget V_i. Only backward arcs are depicted, so all the remaining arcs are forward arcs.

edge-disjoint (undirected) triangle packing of K_n covering all its edges with triangles that can be computed in polynomial time [33]. 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.

Given a triangle packing Δ of T and a subset X of vertices, we define for any $x \in X$

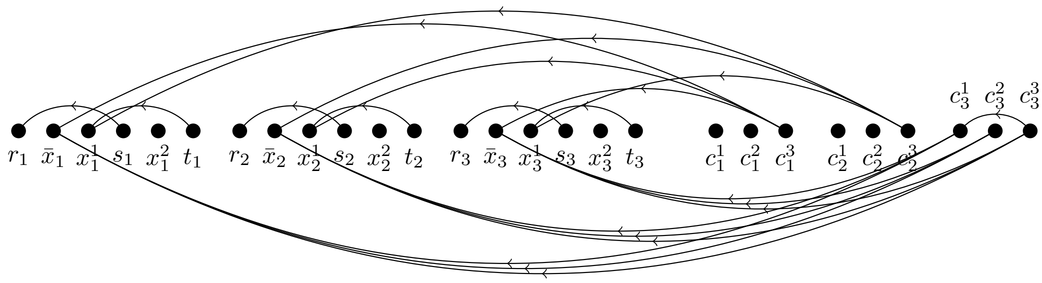

Figure 2 Example of reduction obtained when $F = \{c_1, c_2\}$ where $c_1 = \bar{v}_1 \vee v_2 \vee \bar{v}_3$ and $c_2 = v_1 \vee \bar{v}_2 \vee v_3$. Forward arcs are not depicted. In addition to the depicted backward arcs, we have the 36 backward arcs from V_3 to V_1, and the 9 backward arcs from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

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. 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 3.

▶ **Lemma 1.** *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 2.** *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 1. 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 3.** *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 [41, 44], this implies the following theorem.

▶ **Theorem 4.** 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 5.** *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 4 imply the following theorem.

▶ **Theorem 6.** 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 7.** *Let T be a tournament constructed by the reduction f, and k be the threshold value defined in Lemma 3. Then, we have $k = \text{minfas}(T)$ and we can construct (in polynomial time) an ordering of T with k backward arcs.*

▶ **Theorem 8.** TIGHT-ATT *and* TIGHT-ACT *are* NP*-hard.*

Finally, the size s of the required packing in Lemma 3 satisfies $s = \mathcal{O}((n+m)^2)$. Under the Exponential-time Hypothesis, the problem 3-SAT cannot be solved in $2^{o(n+m)}$ [17, 31]. Then, using the linear reduction from 3-SAT to 3-SAT(3) [44], we also get the following result.

▶ **Theorem 9.** *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 [39], 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 [22]. An interesting consequence of this theorem is that it leads to an FPT algorithm for VERTEX-DISJOINT CYCLE PACKING (see [38] 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 10.** *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 $v_i v_j \in A(T)$ or $v_j v_i \in A(T)$. 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 [15] in order to prove Theorem 12[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 11 was a consequence of a result of [15], as well for an improvement of the constant in Theorem 12.

▶ **Lemma 11.** [15] *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 12.** *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 11. 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 [10] for FAST and uses Theorem 12. 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 lemma known from [10]. An *interval* is a consecutive set of vertices in a linear representation $(\sigma(T), \overleftarrow{A}(T))$ of a tournament T.

▶ **Lemma 13** ([10])**.** *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 13. 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 14.** *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 13 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 15.** 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 12, 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 13, 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 [23], W[1]-hard [43] with respect to k as parameter and solvable in $n^{\mathcal{O}(k)}$ time [28]. Despite its fixed-parameter intractability, we will show that we can use the $n^{\mathcal{O}(k)}$ algorithm and Theorems 12 and 15 to describe an FPT algorithm for ACT.

▶ **Theorem 16.** 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 15, 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 12, 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 12 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 12. 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 [28]. 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 17.** 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 [**?**]. 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 18.** 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 $E(B) = \{au : a \in A_\mathcal{X}, u \in U$ 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 $S \subseteq U$, $N_F(S) = \{a \in A_{\mathcal{X}} : \exists s \in S \text{ s.t. } (t(a), h(a), s) \in F \text{ and } as \notin M\}$ and given a subset $S \subseteq A_{\mathcal{X}}$, $N_M(S) = \{u \in U : \exists a \in A_{\mathcal{X}} \text{ s.t. } au \in M\}$. 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 [42] - FAST is one of them [4, 24]. 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) Proc., 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, M. Bougeret, and J. Thiebaut. (Arc-disjoint) Cycle Packing in Tournament: Classical and Parameterized Complexity, 2018. `arXiv:1802.06669`.

10 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.

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

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

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

14 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.

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

16 W. W. Cohen, R. E. Schapire, and Y. Singer. Learning to Order Things. In *Advances in Neural Information Processing Systems Conference (NIPS)*, pages 451–457, 1997.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

31 R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci*, 63(4):512 – 530, 2001.

32 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. Springer Berlin Heidelberg, 2010.

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

34 R. Krithika, A. Sahu, S. Saurabh, and M. Zehavi. The Parameterized Complexity of Packing Arc-Disjoint Cycles in Tournaments, 2018. `arXiv:1802.07090`.

35 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.

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

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

38 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.

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

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

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

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

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

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

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

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 and 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 is fixed-parameter tractable and admits a polynomial kernel when parameterized by k. In particular, we show that ACT has a kernel with $\mathcal{O}(k)$ vertices and can be solved in $2^{\mathcal{O}(k \log k)} n^{\mathcal{O}(1)}$ time. Then, we show that ATT too has a kernel with $\mathcal{O}(k)$ vertices and can be solved in $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$ time. Afterwards, we describe polynomial-time algorithms for ACT and ATT when the input tournament has a feedback arc set that is a matching. We also prove that ACT and ATT cannot be solved in $2^{\mathcal{O}(\sqrt{k})} n^{\mathcal{O}(1)}$ time under the Exponential-Time Hypothesis.

2012 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity

* This paper is based on the two independent manuscripts [10] and [38].

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:23
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

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

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23
593

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 [26], 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 $f(k)n^{\mathcal{O}(1)}$ where f grows super-polynomially with k 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* and we say that the problem admits 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 of rules (called *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: $\mathsf{FPT} \subseteq \mathsf{W}[1] \subseteq \mathsf{W}[2] \subseteq \ldots \subseteq \mathsf{XP}$. 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. As mentioned before, the set of parameterized problems that admit a polynomial kernel is contained in the class FPT and it is believed that this subset relation is also strict. Further details on parameterized algorithms can be found in [21, 24, 29, 31].

Vertex-Disjoint Cycle Packing in undirected graphs is FPT with respect to the solution size k [12, 43] but has no polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ [13]. In contrast, Edge-Disjoint Cycle Packing in undirected graphs admits a kernel with $\mathcal{O}(k \log k)$ vertices (and is therefore FPT) [13]. On directed graphs, these problems have many practical applications (for example in biology [14, 23]) and they have been extensively studied [7, 40, 44]. It turns out that Vertex-Disjoint Cycle Packing and Arc-Disjoint Cycle Packing are equivalent and are W[1]-hard [39, 52]. 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, 46]. A *tournament* is a directed graph in which there is a single arc between every pair of distinct vertices. Tournaments have several applications in modeling round-robin tournaments and in the study of voting systems and social choice theory [34, 36, 42]. Further, the combinatorics of inclusion relations of tournaments is reasonably well-understood [16]. A seminal result in the theory of undirected graphs is the Graph Minor Theorem (also known as the Robertson and Seymour theorem) that states that undirected graphs are well-quasi-ordered under the *minor relation* [50]. Developing a similar theory of inclusion relations of directed graphs has been a long-standing research challenge. However, there is such a result known for tournaments that states that tournaments are well-quasi-ordered under the *strong immersion relation* [16].[59] This is another reason why tournaments is one of the most well-studied classes of directed graphs. In fact, this result on containment theory also holds for a superclass of tournaments, namely, semicomplete digraphs [8]. A *semicomplete digraph* is a directed graph in which there is at least one arc between every pair of distinct vertices. Many results (including some of the ones described in this work) for tournaments straightaway hold for semicomplete digraphs too.

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 [22, 42], machine learning [18], search engine ranking [25] and have been intensively studied in various algorithmic areas. MinFAST and MinFVST are NP-hard [3, 15, 19, 53] while FAST and FVST are FPT when parameterized by the solution size k [4, 28, 30, 36, 49]. Further, FAST has a kernel with $\mathcal{O}(k)$ vertices [11] and FVST has a kernel with $\mathcal{O}(k^{1.5})$ vertices [41]. 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 [9]. 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* [41].

It is easy to verify that 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. Further, it also hints that the problems of packing arc-disjoint cycles and arc-disjoint triangles in tournaments could have 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 is the task of determining if T has k arc-disjoint cycles and ATT is the task of determining if T has k arc-disjoint triangles. MaxATT is a special case of 3-Set Packing, by creating the hypergraph on the arc set of the tournament and each triangle becomes a hyperedge. The 3-Set Packing problem admits a $\frac{4}{3}+\varepsilon$ approximation [20], implying the same result for MaxATT. 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 [55], almost regular tournaments [2] and complete digraphs [33]. 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. First, we show that MaxACT and MaxATT are NP-hard. Then, we show that ACT is FPT and admits a linear vertex kernel when parameterized by k. Next, we show that ATT is FPT and admits a linear vertex kernel when parameterized by k. Finally, we show that MaxACT and MaxATT are polynomial-time solvable on *sparse tournaments* (tournaments that have a feedback arc set that is a matching). This class of tournaments is interesting for cycle packing problems and packing vertex-disjoint triangles (and hence cycles) in sparse tournaments is NP-complete [9]. In particular, we show the following results.

- MaxATT and MaxACT are NP-hard (Theorems 4 and 6). 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 (Theorem 18) and admits a kernel with $\mathcal{O}(k)$ vertices (Theorem 19).
- MaxATT and MaxACT restricted to sparse tournaments is polynomial-time solvable (Theorem 22).

Road Map. The paper is organized as follows. In Section 2, we give some definitions related to directed graphs, paths, cycles and tournaments. In Section 3, we show the result on the NP-hardness of the problems considered. In Section 4, we show the parameterized complexity results of ACT. Then, in Section 5, we show the parameterized complexity results of ATT. Then, we show the polynomial-time solvability of MaxATT and MaxACT restricted to sparse tournaments in Section 6. Finally, we conclude with some remarks in Section 7.

2 Preliminaries

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

Directed Graphs. A *directed graph* (or *digraph*) is a pair consisting of a set V of vertices and a set A of arcs. An arc is specified as an ordered pair of vertices (called its endpoints). We will consider only simple unweighted digraphs. For a digraph D, $V(D)$ and $A(D)$ denote the set of its vertices and the set of its arcs, respectively. Two vertices u, v are said to be *adjacent* in D if $uv \in A(D)$ or $vu \in A(D)$. For an arc $e = uv$, we define $\mathrm{h}(e) = v$ as the head of e and $\mathrm{t}(e) = u$ as the tail of e. 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 *in-neighbourhood*, denoted by $N^-(v)$, is the set $\{u \in V(D) \colon uv \in A(D)\}$. 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. For a set $F \subseteq A(D)$, $D - F$ denotes the digraph obtained from D by deleting F.

Paths and Cycles. A *path* P in a digraph D is a sequence $(v_1, \ldots, v_k)$ of distinct vertices such that for each $i \in [k-1]$, $v_i v_{i+1} \in A(D)$. The set $\{v_1, \ldots, v_k\}$ is denoted by $V(P)$ and the set $\{v_i v_{i+1} \colon i \in [k-1]\}$ is denoted by $A(P)$. A path $P = (v_1, \ldots, v_k)$ is called an *induced* (or *chordless*) path if $A(P)$ are the only arcs of $D[V(P)]$. A *cycle* C in D is a sequence $(v_1, \ldots, v_k)$ of distinct vertices such that $(v_1, \ldots, v_k)$ is a path and $v_k v_1 \in A(D)$. The set

$\{v_1, \dots, v_k\}$ is denoted by $V(C)$ and the set $\{v_i v_{i+1} : i \in [k-1]\} \cup \{v_k v_1\}$ is denoted by $A(C)$. A cycle $C = (v_1, \dots, v_k)$ is called an *induced* (or *chordless*) cycle if $A(C)$ are the only arcs of $D[V(C)]$. The length of a path or cycle X is the number of vertices in it and is denoted by $|X|$. For a set $\mathcal{C}$ of paths or cycles, $V(\mathcal{C})$ denotes the set $\{v \in V(D) : \exists C \in \mathcal{C}, v \in V(C)\}$ and $A(\mathcal{C})$ denotes the set $\{e \in A(D) : \exists C \in \mathcal{C}, e \in A(C)\}$. A cycle on three vertices is called a *triangle*. A digraph is said to be *triangle-free* if it has no triangles. A set of pairwise arc-disjoint cycles is called a *cycle packing* and a set of pairwise arc-disjoint triangles is called a *triangle packing*. 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 $v_i v_j \in A(D)$, $i < j$ 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 $\overrightarrow{A}(T) = \{v_{i_1} v_{i_2} : (i_1 < i_2) \text{ and } v_{i_2} v_{i_1} \notin \overleftarrow{A}(T)\}$ 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. It is clear that for any linear representation $(\sigma(T), \overleftarrow{A}(T))$ of T the set $\overleftarrow{A}(T)$ is an FAS of T. A tournament is *sparse* if it admits an FAS which is a matching. Given a linear representation $(\sigma(T), \overleftarrow{A}(T))$ of a tournament T, a triangle C in T is a triple $(v_{i_1}, v_{i_2}, v_{i_3})$ with $i_l < i_{l+1}$ such that either $v_{i_3} v_{i_1} \in \overleftarrow{A}(T)$, $v_{i_3} v_{i_2} \notin \overleftarrow{A}(T)$ and $v_{i_2} v_{i_1} \notin \overleftarrow{A}(T)$ (in this case we call C a *triangle with backward arc* $v_{i_3} v_{i_1}$), or $v_{i_3} v_{i_1} \notin \overleftarrow{A}(T)$, $v_{i_3} v_{i_2} \in \overleftarrow{A}(T)$ and $v_{i_2} v_{i_1} \in \overleftarrow{A}(T)$ (in this case we call C a *triangle with two backward arcs* $v_{i_3} v_{i_2}$ and $v_{i_2} v_{i_1}$). 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 $T = T_1 T_2$ 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). Recall that 3-SAT(3) corresponds to 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.

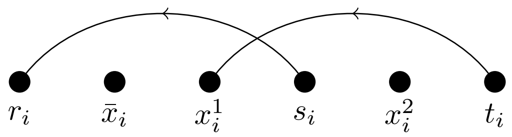

Figure 1 The variable gadget V_i. Only backward arcs are depicted, so all the remaining arcs are forward arcs.

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 $\overleftarrow{A}_\sigma(V_i) = \{s_i r_i, t_i x_i^1\}$. 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 $6n$ 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 [37]. Let $\{u_1, \dots, u_n\}$ 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 $\sigma'(T_{K_n}) = (u_1, \dots, u_n)$ 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 $\sigma(T_c) = (c_1^1, c_1^2, c_1^3, \dots, c_m^1, c_m^2, c_m^3, c_{m+1}^1, c_{m+1}^2, c_{m+1}^3)$. 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 $c_j = \ell_1 \vee \ell_2 \vee \ell_3$, 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

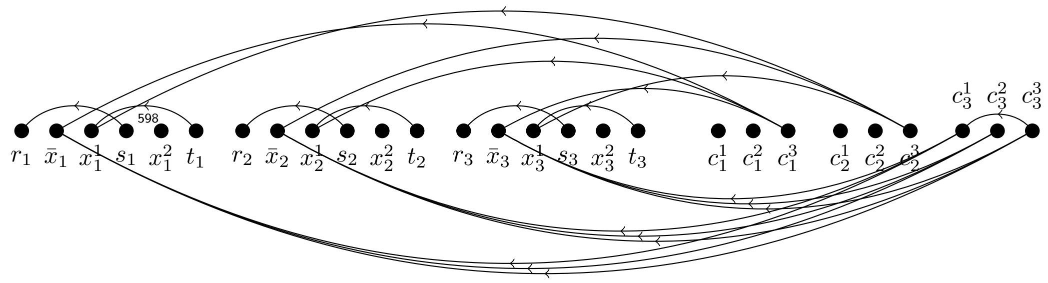

Figure 2 Example of reduction obtained when $F = \{c_1, c_2\}$ where $c_1 = \bar{v}_1 \vee v_2 \vee \bar{v}_3$ and $c_2 = v_1 \vee \bar{v}_2 \vee v_3$. Forward arcs are not depicted. In addition to the depicted backward arcs, we have the 36 backward arcs from V_3 to V_1, and the 9 backward arcs from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

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.

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

(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 $\Delta_i^{\top'}$ 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.

▶ **Lemma 1.** *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 $\{f_i \colon i \in [6]\}$, $\{g_i \colon i \in [6]\}$ and $\{h_i \colon i \in [6]\}$ and consider the tripartite tournament $K_{6,6,6}$ given by $V(K_{6,6,6}) = \{f_i, g_i, h_i \colon i \in [6]\}$ and $A(K_{6,6,6}) = \{f_ig_j, g_ih_j, h_if_j \colon i, j \in [6]\}$. Then it is easy to check that $\{(f_i, g_j, h_{i+j \pmod 6}) \colon i, j \in [6]\}$ 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 2.** *For any triangle packing Δ of the tournament T, we have the following inequalities:*

- **(i)** $|\Delta_{V,V,V}| + |\Delta_{C,C,C}| \leq 6n(n-1) + 3m(m+1)/2$,
- **(ii)** $|\Delta_{2V,C}| + |\Delta_{V,2C}| + |\Delta_{V,C,C}| + |\Delta_{V,V,C}| \leq |\overleftarrow{A}^{vc}(T)|$, *where* $|\overleftarrow{A}^{vc}(T)| = |\overleftarrow{A}^{vc}(T)|$,
- **(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 1. 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. ◀

These two lemmas allow us to prove the following.

▶ **Lemma 3.** *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.*

Proof. First, let suppose that there exists an assignment a of the variables which satisfies F, and let $a^\top$ (resp. $a^\perp$) be the set of variables set to true (resp. false).

We construct a triangle packing Δ of T with the desired number of triangles. First, we pick all the disjoint triangles of Δ^v and Δ^c. By Lemma 2, if we also add the dummy triangle $(c^1_{m+1}, c^2_{m+1}, c^3_{m+1})$ we have $6n(n-1) + 3m(m+1)/2 + 1$ triangles in Δ until now.

Then, for any variable v_i of the formula F, if $v_i \in a^\top$, then we add in Δ the triangles $\Delta_i^\top$. Otherwise, we add $\Delta_i^\perp$. One can check that in both cases, these triangles are disjoint to the triangles we just added. Thus, in each V_i, we made an locally-optimal solution, so we added $2n$ triangles in Δ.

Now we add in Δ the triangles $(\bar{x}_i, t_i, c^1_{m+1})$, $(\bar{x}_i, x^1_i, c^2_{m+1})$ and $(\bar{x}_i, x^2_i, c^3_{m+1})$ which will consume all the dummy arcs of the tournament. Recall that in Remark 3 we mentioned that the vertices x^1_i and x^2_i (resp. $\bar{x}_i$) have an $\Delta_i^\top$-local out-degree both equal to 1 (resp. $\Delta_i^\perp$-local out-degree equals to 4). Then given a clause c_j, let ℓ be one literal which satisfies c_j. Assume that the clause is of size 3, since the reasoning is the same for clauses of size 2.

If ℓ is a positive literal, say v_i, then let u be the number such that $c_j^3 x_i^u$ is a backward arc of T. By Remark 3, we know that there exists $v \in V_i$ such that the arc $x_i^u v$ is available to make the triangle (x_i^u, v, c_j^3). Otherwise, that is if ℓ is a negative literal, say $\bar{v}_i$, then we have $d^+_{V_i \setminus \Delta_i^\perp}(\bar{x}_i) = 4$. Three of these four available arcs are used in the triangles which consume the dummy arcs, then we can still make the triangle $(\bar{x}_i, s_i, c_j^3)$. Let also ℓ_1 and ℓ_2 be the two other literals of c_j (which do not necessarily satisfy c_j). Denote by a_1 and a_2 the vertices of T_v connected to c_j^3 corresponding to the literals ℓ_1 and ℓ_2, respectively. Then we add the two following triangles: (a_1, c_j^1, c_j^3) and (a_2, c_j^2, c_j^3). So we used all the backward arc from T_c to T_v, and there are no triangles which use two arcs of $\overleftarrow{A}^{vc}(T)$. Then in the packing Δ there are in total $6n(n-1) + 3m(m+1)/2 + 2n + |\overleftarrow{A}^{vc}(T)| + 1$ triangles.

Conversely let Δ be a triangle packing of T with $|\Delta| = 6n(n-1) + 3m(m+1)/2 + 2n + |\overleftarrow{A}^{vc}(T)| + 1$. In the same way as we already did before, we partition Δ into the different subsets we defined before. We have $|\Delta| = |\Delta_{V,V,V}| + |\Delta_{V,V,C}| + |\Delta_{V,C,C}| + |\Delta_{C,C,C}| + |\Delta_{2V,C}| + |\Delta_{V,2C}| + |\Delta_{3V}| + |\Delta_{3C}|$. By Lemma 2 all the upper bounds described above are tight, that is:

- $|\Delta_{V,V,V}| + |\Delta_{C,C,C}| = 6n(n-1) + 3m(m+1)/2$,
- $|\Delta_{2V,C}| + |\Delta_{V,2C}| + |\Delta_{V,C,C}| + |\Delta_{V,V,C}| = |\overleftarrow{A}^{vc}(T)|$,
- $|\Delta_{3V}| = 2n$,
- $|\Delta_{3C}| = 1$.

Let us first prove that $|\Delta_{V,V,C}| + |\Delta_{V,C,C}| = 0$. Let $x = |\Delta_{V,V,C}| + |\Delta_{V,C,C}|$. Since each triangle of the sets $\Delta_{V,V,C}, \Delta_{V,C,C}, \Delta_{2V,C}$ and $\Delta_{V,2C}$ uses exactly one backward arc of $\overleftarrow{A}^{vc}(T)$, it implies that $|\Delta_{2V,C}| + |\Delta_{V,2C}| \leq |\overleftarrow{A}^{vc}(T)| - x$. Moreover, if $x \neq 0$, then we have $|\Delta_{V,V,V}| < |\Delta^v|$ or $|\Delta_{C,C,C}| < |\Delta^c|$ because each triangle in $\Delta_{V,V,C}$ (resp. $\Delta_{V,C,C}$) will use one arc between two distinct variable gadgets (resp. clause gadgets) and according to Lemma 1, Δ^v (resp. Δ^c) uses all the arcs between distinct variable gadgets (resp. clause gadgets). Finally, we always have $|\Delta_{3V}| \leq 2n$ and $|\Delta_{3C}| \leq 1$ by construction. Therefore, if $x \neq 0$, we have $|\Delta| < |\Delta^v| + |\Delta^c| + x + (|\overleftarrow{A}^{vc}(T)| - x) + 2n + 1$ that is $|\Delta| < 6n(n-1) + 3m(m+1)/2 + 2n + |\overleftarrow{A}^{vc}(T)| + 1$, which is impossible. So we must have $x = 0$, which implies $\Delta_{V,V,C} = \Delta_{V,C,C} = \emptyset$.
Since $|\Delta_{3V}| = 2n$ and we have at most two arc-disjoint triangles in each variable gadget V_i, it implies that $\Delta[V_i] \in \{\Delta_i^\perp, \Delta_i^\top, \Delta_i^{\top'}\}$. In the following, we will simply write Δ_i instead of $\Delta[V_i]$. Let us consider the following assignment a: for any variable v_i, if $\Delta_i = \Delta_i^\perp$, then $a(v_i) = false$ and $a(v_i) = true$ otherwise. Let us see that the assignment a satisfies the formula F. We have just proved that the backward arcs from T_c to T_v are all used in $\Delta_{2V,C}$ and $\Delta_{V,2C}$. As $|\Delta_{3C}| = 1$ the dummy triangle C_{m+1} belongs to Δ. So every dummy arc $c^u_{m+1}\bar{x}_i$ is contained in a triangle of Δ which uses an arc of V_i. Therefore in each V_i we have $d^+_{V_i \setminus \Delta_i}(\bar{x}_i) \geq 3$. Moreover, for each clause of size q with $q \in \{2, 3\}$, there are q triangles which use the backward arcs coming from the clause to variable gadgets. Let C_j be a clause gadget of size 3 (we can do the same reasoning if C_j has size 2). By construction the 3 triangles cannot all lie in $\Delta_{V,2C}$. Thus, there is at least one of these triangles which is in $\Delta_{2V,C}$. Let t be one of them, V_i be the variable gadget where t has two out of its three vertices and $\tilde{x}$ be the vertex of V_i which is also the head of the backward arc from C_j to V_i. By construction, $\tilde{x}$ corresponds to a literal ℓ in the clause c_j. If ℓ is positive, then $\tilde{x} = x_i^1$ or $\tilde{x} = x_i^2$. In both cases, since t has a second vertex in V_i, we have $d^+_{V_i \setminus \Delta_i}(\tilde{x}) > 0$. Thus, using Figure 3 we cannot have $\Delta_i = \Delta_i^\perp$ so the assignment sets the positive literal ℓ to $true$, which satisfies c_j. Otherwise, ℓ is negative so $\tilde{x} = \bar{x}_i$. Since $\bar{x}_i$ has to use three out-going arcs to consume the dummy arcs and one out-going arc to consume t, we have $d^+_{V_i \setminus \Delta_i}(\bar{x}_i) \geq 4$ and so $\Delta_i = \Delta_i^\perp$ by Figure 3. Therefore, c_j is satisfied in that case too. Thus, the assignment a satisfies the whole formula F. ◀

As 3-SAT(3) is NP-hard [47, 54], this directly implies the following theorem.

▶ **Theorem 4.** 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 5.** *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.*

Proof. Given a cycle packing O of T of size $6n(n-1)+3m(m+1)/2+2n+|\overleftarrow{A}^{vc}(T)|+1$, we partition it into the following sets:

- $O_V = \{(v_1,\dots,v_p) \in O\colon \exists i \in [n], \forall k \in [p], v_k \in V_i\}$,
- $O_C = \{(v_1,\dots,v_p) \in O\colon \exists j \in [m+1], \forall k \in [p], v_k \in C_j\}$,
- $O_{V^*} = \{(v_1,\dots,v_p) \in O\colon \forall k \in [p], \exists i \in [n], v_k \in V_i \text{ and } (v_1,\dots,v_p) \notin O_V\}$,
- $O_{C^*} = \{(v_1,\dots,v_p) \in O\colon \forall k \in [p], \exists j \in [m+1], v_k \in C_j \text{ and } (v_1,\dots,v_p) \notin O_C\}$,
- $O_{V^*,C^*} = \{(v_1,\dots,v_p) \in O\colon \exists i \in [n], \exists j \in [m+1], \exists k_1,k_2 \in [p], v_{k_1} \in V_i, v_{k_2} \in C_j\}$.

As we did in the previous proof, we begin by finding upper bounds on each of these sets. First, recall that the FAS of each V_i is 2. Thus, we have $|O_V| \leq 2n$. By construction, we also have $|O_C| \leq 1$. Secondly, notice that a cycle of O_{V^*} cannot belong to exactly two distinct variable gadgets since the arcs between them are all in the same direction. Thus, the cycles of O_{V^*} have at least three vertices which implies $|O_{V^*}| \leq 6n(n-1)$. We obtain $|O_{C^*}| \leq 3m(m+1)/2$ using the same reasoning on O_{C^*}. Finally, we have $|O_{V^*,C^*}| \leq |\overleftarrow{A}^{vc}(T)|$ since each cycle must have at least one backward arc.

Putting these upper bounds together, we obtain that $|O| \leq 6n(n-1)+3m(m+1)/2+2n+|\overleftarrow{A}^{vc}(T)|+1$ which implies that the bounds are tight. In particular, cycles of O_{V^*} (resp. O_{C^*}) use exactly three arcs that are between distinct variable gadgets (resp. clause gadgets) and all these arcs are used. So we can construct a new cycle packing O' where we replace the cycles of O_{V^*} and O_{C^*} by the triangle packings Δ^v and Δ^c defined in Lemma 1. The new solution uses a subset of arcs of O and has the same size.

The cycles of O_{V^*,C^*} use exactly one backward arc of $\overleftarrow{A}^{vc}(T)$ due to the tight upper bound $|\overleftarrow{A}^{vc}(T)|$. Moreover, by the previous reasoning, two vertices of a cycle of O_{V^*,C^*} cannot belong to two different variable gadgets (resp. clause gadgets). Let C_j be a clause gadget which has three literals (if it has only two literals, the reasoning is analogous). Let $\tilde{x}_{i_k} \in V_{i_k}$ be the head of a backward arc from c_j^3 where $k \in [3]$. By the previous arguments each arc $c_j^3\tilde{x}_{i_k}$ is contained in a cycle o_k of O for $k \in [3]$. There is at least one $\tilde{x}_{i_k}$ whose next vertex in o_k, say y, belongs to V_{i_k} since C_j has only two other vertices in addition to c_j^3. Without loss of generality, we may assume that $\tilde{x}_{i_3}$ is that vertex. Then, we can replace o_1 and o_2 by the triangles $(\tilde{x}_{i_1}, c_j^1, c_j^3)$ and $(\tilde{x}_{i_2}, c_j^2, c_j^3)$. The arcs $c_j^1c_j^3$ and $c_j^2c_j^3$ cannot have already been used because C_j is acyclic and we previously consumed all the arcs between clause gadgets. In the same way, we replace the cycle o_3 by the triangle $(\tilde{x}_{i_3}, y, c_j^3)$. The arc yc_j^3 is available since it could have been used only in the cycle o_3.

We now prove that given a V_i, we can restructure every cycle of $O_V[V_i]$ into triangles. Recall that $O_V[V_i]$ have exactly 2 cycles, and notice that by construction one cannot have two cycles each having a size greater than 3. First, if the two cycles are triangles, we are done. Then $O_V[V_i]$ contains a triangle, say δ, and a cycle, say o, of size greater than 3. If o contains the backward arc s_ir_i, then by construction $o = (r_i, \bar{x}_i, x_i^1, s_i)$. In that case, we necessary have $\delta = (x_i^1, x_i^2, t_i)$ and we can restructure o in the triangle (r_i, x_i^1, s_i). The arc

$r_i x_i^1$ is not contained in O since the only arcs inside V_i we may have imposed until now are out-going arcs of x_i^1, x_i^2 and $\bar{x}_i$. If o contains the backward arc $t_i x_i^1$, then by construction $o = (x_i^1, s_i, x_i^2, t_i)$ and $t = (r_i, \bar{x}_i, s_i)$. In the same way, we can restructure o into (x_i^1, s_i, t_i) whose all the arcs are available.

As O_C is already a triangle, T finally has a triangle packing of size $6n(n-1) + 3m(m+1)/2 + 2n + |\overleftarrow{A}^{vc}(T)| + 1$. The other direction of the equivalence is straightforward. ◀

The previous lemma and Theorem 4 directly imply the following theorem.

▶ **Theorem 6.** 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 now prove that we can construct in polynomial time an ordering of T, the tournament of the reduction, with k backward arcs (where k is the threshold value defined in Lemma 3).

▶ **Lemma 7.** *Let T be a tournament constructed by the reduction f, and k be the threshold value defined in Lemma 3. Then, we can construct (in polynomial time) an ordering of T with k backward arcs implying that T has an FAS of size k.*

Proof. Let us define a linear representation $(\sigma(T), \overleftarrow{A}(T))$ such that $|\overleftarrow{A}(T)| = k$. Remember that since $n \equiv 3 \pmod 6$, the edges of the n-clique K_n can be packed into a packing O of $n(n-1)/6$ (undirected) triangles. Let us first prove that there exists an orientation T_{K_n} of K_n and a linear ordering σ of T_{K_n} with $|O|$ backward arcs. Let $\sigma = 1 \dots n$. For each undirected triangle ijk in O where $i < j < k$, we set $ki \in \overleftarrow{A}(T_{K_n})$ (implying that ij and jk are forward arcs). As all edges are used in O this defines an orientation for all edges. Thus, there is only $|O|$ backward arcs in σ. Thus, when using the previous orientations T_{K_n} to construct the variable tournament T_v of the reduction (remember that we blow up each vertex u_i into 6 vertices V_i), we get an ordering with $36n(n-1)/6 = 6n(n-1)$ backward arcs between two different V_i (more formally, $|\{a \in \overleftarrow{A}(T_v) \colon \exists i_1 \neq i_2, h(a) \in V_{i_1}, t(a) \in V_{i_2}\}| = 6n(n-1)$). Following the same construction for the clause tournament T_c we get an ordering with $3m(m+1)/2$ backward arcs between two distinct C_j. Now, as there are two backward arcs in each V_i, one backward arc in C_{m+1}, and $|\overleftarrow{A}^{vc}(T)|$ backward arcs from T_c to T_v, the total number of backward arcs is k. ◀

We also prove that $k = \text{minfas}(T)$.

▶ **Lemma 8.** *Let $T = (V, A)$ be a tournament constructed by the reduction f and k be the threshold value defined in Lemma 3. Then,* $\text{minfas}(T) \geq k$*.*

Proof. We suppose that T is equipped with the ordering defined in Lemma 7. Let F be an optimal FAS of T. Given an arc a, let $v(a) = \{t(a), h(a)\}$. Let us partition the arcs of T into the following sets. For any $i \in [n], j \in [m+1]$, let us define

- $A_{V_i} = \{a \in A \colon v(a) \subseteq V_i\}$
- $A_{C_j} = \{a \in A \colon v(a) \subseteq C_j\}$
- $A_{V_i C_j} = \{a \in A \colon |v(a) \cap V_i| = |v(a) \cap C_j| = 1\}$
- $A_{V_i V_{i'}} = \{a \in A \colon |v(a) \cap V_i| = |v(a) \cap V_{i'}| = 1\}$ where $i \neq i'$
- $A_{C_j C_{j'}} = \{a \in A \colon |v(a) \cap C_j| = |v(a) \cap C_{j'}| = 1\}$ where $j \neq j'$

For any $i, i' \in [n]$, $j, j' \in [m+1]$ and $X \in \{V_i, C_j, V_iC_j, V_iV_{i'}, C_jC_{j'}\}$, we also define the corresponding sets F_X in F, where for example $F_{V_i} = F \cap A_{V_i}$. In addition, for any $j \in [m+1]$ we define $F_{*C_j} = \bigcup_{i \in [n]} F_{V_iC_j}$. Let T'_v be the directed graph (T'_v is not a tournament) obtained by starting from T_v and only keeping arcs in $A_{V_iV_{i'}}$ for any $i, i' \in [n]$ with $i \neq i'$. As F is FAS of T, $F_{VV} = \bigcup_{i,i' \in [n], i \neq i'} F_{V_iV_{i'}}$ must be an FAS of T'_v. As according to Lemma 1 there is a cycle packing of size $6n(n-1)$ in T'_v, we get $|F_{VV}| \geq 6n(n-1)$. The same arguments hold for the clause part, and thus with $F_{CC} = \bigcup_{j,j' \in [m+1], j \neq j'} F_{C_jC_{j'}}$, we get $|F_{CC}| \geq 3m(m+1)/2$. As C_{m+1} is a triangle, we also get $|F_{C_{m+1}}| \geq 1$.

For any $j \in [m]$, let $u_j \in \{2, 3\}$ be equal to the size of the clause j (we also have $u_j = |\{a \in \overleftarrow{A}(T) : \exists i \in [n], h(a) \in V_i \text{ and } t(a) \in C_j\}|$). Let $L = \{j \in [m] : |F_{*C_j} \cup F_{C_j}| \geq u_j\}$ be informally the set of clauses where F spends a large (in fact larger than the u_j required) amount of arcs, and $S = [m] \setminus L$. Let us prove that for any $j \in S$, $|F_{C_j}| \geq u_j - 1$. Let us first consider the case where $u_j = 3$. Suppose by contradiction than $F_{C_j} = \{a\}$ (arguments will also hold for $F_{C_j} = \emptyset$). Remember that $\sigma(C_j) = (c_j^1, c_j^2, c_j^3)$ (there are only forward arcs). As $|F_{*C_j}| \leq 1$, there exists $i \in [n]$ and two arcs a_1, a_2 not in F such that $t(a_1) = c_j^3$, $h(a_1) \in V_i$, $t(a_2) = h(a_1)$, and $h(a_2) \neq t(a)$. Thus, $(t(a_1), t(a_2), h(a_2))$ is a triangle using no arc of F, a contradiction. As the same kind of arguments holds for the case where $u_j = 2$, we get that for any $j \in S$, $|F_{C_j}| \geq u_j - 1$ (implying also $|F_{*C_j}| = 0$).

Let us now prove that $|S| \leq 1$. Suppose by contradiction that $|S| \geq 2$. Let j_1 and j_2 be in S. For any $l \in [2]$, let define a_l such that there exists $i_l \in [n]$ with $t(a_l) \in C_{j_l}$ and $h(a_1) \in V_{i_l}$. Notice that we may have $i_1 = i_2$, but we always have $h(a_1) \neq h(a_2)$. Moreover, as a_i is the unique backward arc of T with $t(a) \in \bigcup_{j \in [m]} C_j$, we get that $a_3 = h(a_1)t(a_2)$ and $a_4 = h(a_2)t(a_1)$ are forward arcs of T. As $|F_{*C_{j_1}}| = |F_{*C_{j_2}}| = 0$ we know that $a_l \notin F$ for $l \in [4]$. Thus, $(t(a_1), h(a_1), t(a_2), h(a_2), t(a_1))$ is a cycle using no arc of F, a contradiction.

Let $L' = \{i \in [n] : \exists a \in T \text{ s.t. } h(a) \in V_i \text{ and } t(a) \in C_j,\ j \in S\}$. Notice that if $S = \emptyset$ then $L' = \emptyset$, and otherwise $|L'| = u_{j_0}$, where $S = \{j_0\}$. Let $S' = [n] \setminus L'$. For any $i \in [n]$, let $\overleftarrow{A}_{V_iC_{m+1}} = \overleftarrow{A}(T) \cap A_{V_iC_{m+1}}$. Recall that $\overleftarrow{A}_{V_iC_{m+1}} = c_{m+1}^u \bar{x}_i$ for $u \in [3]$ where $\bar{x}_i \in V_i$. Moreover, for any $x \in \{\bar{x}_i, x_i^1, x_i^2\}$, let $A_{xV_i} = \{a \in T : t(a) = x \text{ and } h(a) \in V_i\}$. Notice that $|A_{\bar{x}_iV_i}| = 4$, $|A_{x_i^1V_i}| = 2$ and $|A_{x_i^2V_i}| = 1$.

Let us prove that for any $i \in S'$, $|F_{V_i} \cup F_{V_iC_{m+1}}| \geq 5$. If $A_{\bar{x}_iV_i} \subseteq F$, then as F_{V_i} must be an FAS of V_i and $A_{\bar{x}_iV_i}$ is not an FAS of V_i, there exists at least another arc in F_{V_i} and we get $|F_{V_i}| \geq 5$. Otherwise, $\overleftarrow{A}_{V_iC_{m+1}} \subseteq F$ (if it is not the case, there is a cycle $c_{m+1}^u \bar{x}_i v$ where $v \in V_i$ is a out-neighbour of $\bar{x}_i$). Then, as $\text{minfas}(V_i) \geq 2$, $|F_{V_i} \cup F_{V_iC_{m+1}}| \geq 5$.

Let us finally prove that for any $i \in L'$, $|F_{V_i} \cup F_{V_iC_{m+1}}| \geq 6$. As $i \in L'$, there is an arc $a \in T$ with $h(a) \in V_i$ and $t(a) \in C_{j_0}$ where $S = \{j_0\}$. Let $x = h(a)$. Notice that $x \in \{\bar{x}_i, x_i^1, x_i^2\}$. As $|F_{*C_{j_0}}| = 0$ we get that $A_{xV_i} \subseteq F_{V_i}$ (otherwise there would be a cycle with one vertex in C_{j_0}, x, and an out-neighbour of x in V_i).
Case 1: $x = \bar{x}_i$**.**As F_{V_i} must be an FAS of V_i, F needs two other arcs in A_{V_i} and we get $|F_{V_i}| \geq 6$.
Case 2: $x = x_i^1$**.**If $A_{\bar{x}_iV_i} \subseteq F$ then $|F_{V_i} \cup F_{V_iC_{m+1}}| \geq 6$. Otherwise, as before we get $\overleftarrow{A}_{V_iC_{m+1}} \subseteq F$, and as $A_{x_i^1V_i}$ is not an FAS of V_i, F need another arc in V_i, implying $|F_{V_i} \cup F_{V_iC_{m+1}}| \geq 6$.
Case 3: $x = x_i^2$**.**If $A_{\bar{x}_iV_i} \subseteq F$ then as $A_{x_i^2V_i} \cup A_{\bar{x}_iV_i}$ is not an FAS of V_i, F need another arc in V_i, implying $|F_{V_i}| \geq 6$. Otherwise, as before we get $\overleftarrow{A}_{V_iC_{m+1}} \subseteq F$, and as $A_{x_i^1V_i}$ is not an FAS of V_i, F need two other arcs in V_i, implying $|F_{V_i} \cup F_{V_iC_{m+1}}| \geq 6$.

Putting all the pieces together, we get the following.

$$\begin{aligned} |F| = {} & |F_{VV}| + |F_{CC}| + |F_{C_{m+1}}| + \sum_{j \in L}(|F_{*C_j} \cup F_{C_j}|) + \sum_{j \in S}(|F_{*C_j} \cup F_{C_j}|) \\ & + \sum_{i \in S'}(|F_{V_i} \cup F_{V_i C_{m+1}}|) + \sum_{i \in L'}(|F_{V_i} \cup F_{V_i C_{m+1}}|) \\ & \geq 6n(n-1) + \frac{3m(m+1)}{2} + 1 + \sum_{j \in L} u_j + \sum_{j \in S}(u_j - 1) + 5|S'| + 6|L'| \\ & \geq 6n(n-1) + \frac{3m(m+1)}{2} + 1 + \sum_{j \in [m]} u_j + 5n = k \end{aligned}$$

◀

Then, using Lemma 7 and Lemma 8, we get the NP-hardness of Tight-ATT and Tight-ACT.

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

Finally, the size s of the required packing in Lemma 3 satisfies $s = \mathcal{O}((n+m)^2)$. Under the Exponential-time Hypothesis, the problem 3-SAT cannot be solved in $2^{o(n+m)}$ [21, 35]. Then, using the linear reduction from 3-SAT to 3-SAT(3) [54], 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 [45], 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 there exists a function $f(k) = \mathcal{O}(k \log k)$ such that for each non-negative integer k, every undirected graph either contains k vertex-disjoint cycles or has a feedback vertex set consisting of $f(k)$ vertices [26]. An interesting consequence of this theorem is that it leads to an FPT algorithm for Vertex-Disjoint Cycle Packing. It is well known that the treewidth (tw) of a graph is not larger than the size of its feedback vertex set, and that a naive dynamic programming scheme solves Vertex-Disjoint Cycle Packing in $\mathcal{O}^\star(2^{\mathcal{O}(tw \log tw)})$ time (see, e.g., [21]). Thus, the existence of an $\mathcal{O}^\star(2^{\mathcal{O}(k \log^2 k)})$ time algorithm can be viewed as a direct consequence of the Erdős-Pósa theorem (see [43] 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.

4.1 An Erdős-Pósa Type Theorem

In this section, we show certain interesting combinatorial results on arc-disjoint cycles in tournaments.

▶ **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 $v_i v_j \in A(T)$ or $v_j v_i \in A(T)$. 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 [17] in the process[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 $vu \in A(D)$. Recall that for a digraph D, minfas(D) denotes the size of a minimum FAS of D.

▶ **Lemma 12.** [17] *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.*

This leads to the following main result of this section.

▶ **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 $|F^*| \leq 5(k-1)$. ◀

4.2 A Linear Vertex Kernel

Next, we show that ACT has a linear vertex kernel. This kernel is inspired by the linear kernelization described in [11] 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 not in any cycle, then delete v from T.*

[1] The authors would like to thank F. Havet for pointing out that Lemma 12 was a consequence of a result of [17], as well for an improvement of the constant in Theorem 13.

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 lemma known from [11]. An *interval* is a consecutive set of vertices in a linear representation $(\sigma(T), \overleftarrow{A}(T))$ of a tournament T.

▶ **Lemma 14 ([11]).** [2] *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. ◀

[2] Lemma 14 is Lemma 3.9 of [11] that has been rephrased to avoid the use of several definitions and terminology introduced in [11].

4.3 An FPT Algorithm

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 [27], W[1]-hard [52] with respect to k as parameter and solvable in $n^{\mathcal{O}(k)}$ time [32]. 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 [32]. 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

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 [51]. 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 which belong 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 MaxACT and MaxATT in Sparse Tournaments

Recall that a tournament is *sparse* if it admits an FAS which is a matching. In this section, we show that MaxACT and MaxATT are polynomial-time solvable on sparse tournaments. Note that packing vertex-disjoint triangles (and hence cycles) in sparse tournaments is NP-complete [9].

Let T be a sparse tournament according to the ordering of its vertices $\sigma(T)$, that is the set of its backward arcs $\overleftarrow{A}(T)$ is a matching. If a backward arc xy of T lies between two consecutive vertices, then we can exchange the position of x and y in $\sigma(T)$ to obtain a sparse tournament with fewer backward arc. So we can assume that the backward arcs of T do not contain consecutive vertices. Moreover, if a vertex x of T is contained in no backward arc of T then call A (resp. B) the vertices of T which are before (resp. after) x in $\sigma(T)$. Let X_0 be the set of triangles made from a backward arc from B to A and the vertex x. As T is sparse it is clear that X_0 is a set of disjoint triangles. Moreover, it can easily be seen that there exists an optimal packing of triangles (resp. cycles) of T which is the union of an optimal packing of triangles (resp. cycles) of $T[A]$, one of $T[B]$ and X_0. Thus to solve MaxATT or MaxACT on T we can solve the problem on $T[A]$ and on $T[B]$ and build the optimal solution for T. Therefore we can focus on the case where every vertex of T is the beginning or the end of a backward arc $\overleftarrow{A}(T)$. We will call such a tournament a *fully sparse tournament*. So we focus on solving MaxATT in fully sparse tournaments. In the following, let Π be the problem of finding a collection of arc-disjoint triangles of maximum size on fully sparse tournament.

Now order the arcs $e_1, \ldots, e_b$ of $\overleftarrow{A}(T)$ such that for any $i \in [b-1]$, $h(e_i) <_\sigma h(e_{i+1})$. Moreover, let G' be the digraph with vertex set $V' = \{e_i : i \in [b]\}$ and arc set A' defined by: $(e_i e_j) \in A'$ if $(h(e_i), h(e_j), t(e_i))$ or $(h(e_i), t(e_j), t(e_i))$ is a triangle of T. Let Π' be the problem such that, given a digraph $G' = (V', A')$, the objective is to find a maximum sized subset of A' such that the digraph induced by the arcs of the subset is a functional and digon-free digraph. Remind that a functional digraph is a digraph such that any of its vertices has out-degree at most 1.

Let X be a solution (not necessary optimal) of $\Pi'(G')$, and $e_i e_j$ an arc of X. We denote by $\Pi(e_i e_j)$ the triangle $(h(e_i), h(e_j), t(e_i))$ if $i < j$ and otherwise. Given a triangle $\Pi(e_i e_j)$, let $s(e_j)$ be the second vertex of $\Pi(e_i e_j)$; in other words, if $\Pi(e_i e_j) = (h(e_i), t(e_j), t(e_i))$, then $s(e_j) = t(e_j)$ and $s(e_j) = h(e_j)$ otherwise. Informally, $\Pi(e_i e_j)$ corresponds to the triangle formed by the backward arc e_i and one vertex of e_j, that vertex being $s(e_j)$. In the same way, we define $\Pi(X) = \bigcup_{x \in X} \Pi(x)$.

▶ **Claim 19.1.** *Let X be a solution of $\Pi'(G')$. The set X is an optimal solution if and only if $\Pi(X)$ is an optimal solution of $\Pi(T)$.*

Proof. Let $e_i e_j$ and $e_k e_l$ be two distinct arcs of X. We cannot have $e_i = e_k$ as X induces a functional digraph in G'. Without loss of generality, we may assume that $i < k$, that is

$h(e_i) <_\sigma h(e_k)$. Moreover, we cannot have $t(e_i) = t(e_k)$ without contradicting that T is a sparse tournament. As $h(e_i) <_\sigma h(e_k)$ the arc $h(e_i)s(e_j)$ is not an arc of $\Pi(e_k e_l)$. Thus if $\Pi(e_i e_j)$ and $\Pi(e_k e_l)$ share a common arc, it means that $s(e_j)t(e_i) = h(e_k)s(e_l)$. But in this case $e_i = e_l$ and $e_j = e_k$, implying $\{e_i e_j, e_k e_l\}$ is a digon of G', which contradict the fact that X is a solution $\Pi'(G')$. So, if X is a solution of $\Pi'(G')$, then $\Pi(X)$ is an solution of $\Pi(T)$. Notice that the size of the solution does not change.

On the other hand, if X is a subset of the arcs of G' such that $\Pi(X)$ is a solution of $\Pi(T)$. We cannot have a vertex e_i of G' such that $d^+_X(e_i) > 1$, since it would imply that the backward arc e_i of T is covered by at least two triangles of $\Pi(X)$. So X induces a functional subdigraph of G'. As previously the digraph induced by X is also digon-free otherwise we would have two arc-disjoint triangles on only four vertices in $\Pi(X)$, which is impossible. Thus, X is a solution of $\Pi'(G')$, and the solution of the same size.

The two problems Π and Π' being both maximization problems, they have the same optimal solution. ◀

Now we show how to solve Π' in polynomial time.

▶ **Claim 19.2.** *If G' is strongly connected and has a cycle C of size at least 3 then the solution of $\Pi'(G')$ is the number of vertices of G'.*

Proof. We construct the arc set X as follows: we start by taking the arcs of C. Then, while there is a vertex x which is not covered by any arcs of X, we add to X the arcs of the shortest path from x to any vertex of X. By construction, every vertex x of every arc of X verify $d^+_X(x) = 1$, and X is digon free. Since X covers every vertex of G', $|X|$ is a maximum solution of $\Pi'(G')$, that is the number of vertices of G'. ◀

A digraph D is a *digoned tree* if D arises from a non-trivial tree whose each edge is replaced by a digon.

▶ **Claim 19.3.** *If G' is strongly connected and has only cycles of size 2 then G' is a digoned tree.*

Proof. Since G' is strongly connected, then for any arc xy of G' there exists a path from y to x. As G' only contains cycles of size 2, the only path from y to x is the directed arc yx. So every arc of G' is contained in a digon. If H is the underlying graph of G' (without multiple edges) then it is clear that H is a tree otherwise G' would contain a cycle of size more than 2. ◀

▶ **Claim 19.4.** *If G' is a digoned tree or if $|V(G')|= 1$, then the optimal solution of $\Pi'(G')$ is $|V(G')|-1$.*

Proof. The case $|V(G')|= 1$ is clear. So assume that G' is a digoned tree and let X be a set of arcs of G' corresponding to an optimal solution of $\Pi'(G')$. Then X is acyclic and then has size at most $|V(G')|-1$. Moreover, any in-branching of G' provides a solution of size $|V(G')|-1$. ◀

▶ **Lemma 20.** *Let G' be a digraph with n vertices. Denote by $S_1, \dots, S_p$ terminal strong components of G' such that for any i with $1 \leq i \leq k$, S_i is a digoned tree or an isolated vertex and for any $i > k$, S_i contains a cycle of length at least 3. Then an optimal solution of $\Pi'(G')$ has size $n - k$ and we can construct one in polynomial time.*

Proof. We can assume that G' is connected otherwise we apply the result on every connected component of G' and the disjoint union of the solutions produces an optimal solution on the whole digraph G'.

So assume that G' is connected and let S be a terminal strong component of G'. If X is an optimal solution of $\Pi'(G')$ then the restriction of X to the arcs of $G'[S]$ is an optimal solution of $\Pi'(G'[S])$. Indeed otherwise we could replace this set of arcs in X by an optimal solution of $\Pi'(G'[S])$ and obtain a better solution for $\Pi'(G')$, a contradiction.

So by Claim 19.2 and Claim 19.4 the set X contains at most $\sum_{i=1,\dots,p}|S_i|-k$ arcs lying in a terminal component of G'. Now as every vertex of $G' \setminus \bigcup_{i=1,\dots,p} S_i$ is the beginning of at most one arc of X, the set X has size at most $n-k$. Conversely by growing in-branchings in G' from the union of the optimal solutions of $\Pi'(G'[S_i])$ for $i=1,\dots,p$, by Claim 19.2 and 19.4 we obtain a solution of $\Pi'(G')$ of size $n-k$ which is then optimal. Moreover, this solution can clearly be built in polynomial time. ◀

Using Claim 19.1 and Lemma 20 we can solve MaxATT in polynomial time.

▶ **Lemma 21.** *In a fully sparse tournament T the size of a maximum cycle packing is equal to the size of a maximum triangle packing.*

Proof. First if T has an optimal triangle packing of size $|\overleftarrow{A}(T)|$ then as $\overleftarrow{A}(T)$ is an FAS of T, every optimal cycle packing of T has size $|\overleftarrow{A}(T)|$. Otherwise, we build from T the digraph G' as previously. By Lemma 20, G' has some terminal components $S_1,\dots,S_k$ which are either a single vertex or induces a digoned tree and every optimal triangle packing of T has size $|\overleftarrow{A}(T)|-k$. Let see that no S_i can be a single vertex. Indeed if $S_i=\{e\}$ where e is a backward arc of T, it means that no backward of T begins or ends between $h(e)$ and $t(e)$ in $\sigma(T)$. As T is fully sparse, it means that $h(e)$ and $t(e)$ are consecutive in $\sigma(T)$ what we forbid previously. Now consider a component S_i which induces a digoned tree in G'. Let π_i be the order $\sigma(T)$ restricted to the heads and tails of the arcs of T corresponding to the vertices of S_i. First notice that π_i is an interval of the order $\sigma(T)$. Indeed otherwise there exists two backward arcs a and b of T such that $a \in S_i$, $b \notin S_i$ and $h(a)$ is before the head or the of b which is before $t(a)$ in $\sigma(T)$. But in this case there is an arc in G' from a to b contradicting the fact that S_i is a terminal component of G'. So we denote π_i by $(x_1,x_2,\dots,x_l)$ and notice that x_1 and x_2 are then forced to be the heads of backward arcs belonging to S_i. If x_3 is also the head of backward arc of S_i, then we obtain that the three corresponding backward arcs form a 3-cycle in G' contradicting the fact that S_i induces a digoned tree in G'. Repeating the same argument we show that l is even and that the backward arcs corresponding to the elements of S_i are exactly x_3x_1, x_lx_{l-2} and x_jx_{j-3} for all odd $j \in [l] \setminus \{1,3\}$. In other words S_i induces a 'digoned path' in G'. Now consider Δ an optimal cycle packing of T. Let X_1 be the set of backward arcs of $\overleftarrow{A}(T)$ with head strictly before x_1 and tail strictly after x_l in $\sigma(T)$. And let Δ_1 be the cycles of Δ using at least one arc of X_1. It is easy to check that $\Delta' = (\Delta \setminus \Delta_1) \cup \{(h(e),x_1,t(e))\colon e \in X_1\}$ is also an optimal cycle packing of T. Now every cycle of Δ' which uses a backward arc of S_i only uses backward arcs of S_i (otherwise it must one arc of X_1, which is not possible). Let Δ_i be the set of cycles of Δ using backward arcs of S_i. It is easy to see that $\{x_ix_{i+1}\colon i \text{ even and } i \in [l-2]\}$ is an FAS of $T[\{x_1,\dots,x_l\}]$ and has size $l/2-1=|S_i|-1$. So we have $|\Delta_i| \leq |S_i|-1$.
Repeating this argument for $i=1,\dots,k$ we obtain that $|\Delta| \leq |\overleftarrow{A}(T)|-k$. Thus by Lemma 20 Δ has the same size than an optimal triangle packing of T. ◀

This leads to the following main result of this section.

▶ **Theorem 22.** MaxATT *and* MaxACT *restricted to sparse tournaments can be solved in polynomial time.*

7 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. We also showed that these problems are polynomial-time solvable in sparse tournaments. 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 [48] - FAST is one of them [4, 28]. 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) Proc., 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 F. Barbero, C. Paul, and M. Pilipczuk. Strong Immersion is a Well-quasi-ordering for Semi-complete Digraphs. *CoRR*, abs/1707.03563, 2017.

9 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.

10 S. Bessy, M. Bougeret, and J. Thiebaut. (Arc-disjoint) Cycle Packing in Tournament: Classical and Parameterized Complexity, 2018. `arXiv:1802.06669`.

11 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.

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

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

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

15 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.

16 M. Chudnovsky and P. Seymour. A Well-quasi-order for Tournaments. *J. of Combinatorial Theory, Series B*, 101(1):47–53, 2011.

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

18 W. W. Cohen, R. E. Schapire, and Y. Singer. Learning to Order Things. In *Advances in Neural Information Processing Systems Conference (NIPS)*, pages 451–457, 1997.

19 V. Conitzer. Computing Slater Rankings Using Similarities Among Candidates. In *21st National Conf. on Artificial Intelligence - Volume 1*, pages 613–619, 2006.

20 M. Cygan. Improved Approximation for 3-Dimensional Matching via Bounded Pathwidth Local Search. In *54th Annual IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 509–518, 2013.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

35 R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci*, 63(4):512 – 530, 2001.

36 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. Springer Berlin Heidelberg, 2010.

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

38 R. Krithika, A. Sahu, S. Saurabh, and M. Zehavi. The Parameterized Complexity of Packing Arc-Disjoint Cycles in Tournaments, 2018. `arXiv:1802.07090`.

39 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.

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

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

42 M. le Marquis de Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix.* Imprimerie Royale, Paris, 1785.

43 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.

44 C. L. Lucchesi and D. H. Younger. A Minimax Theorem for Directed Graphs. *J. of the London Mathematical Society*, 17(2):369–374, 1978.

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

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

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

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

49 V. Raman and S. Saurabh. Parameterized Algorithms for Feedback Set problems and their Duals in Tournaments. *Theor. Comput. Sci.*, 351(3):446–458, 2006.

50 N. Robertson and P.D. Seymour. Graph Minors. XX. Wagner's Conjecture. *J. of Combinatorial Theory, Series B*, 92(2):325–357, 2004.

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

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

53 E. Speckenmeyer. On Feedback Problems in Digraphs. In *15th International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 218–231. Springer Berlin Heidelberg, 1990.

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

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