



HAL
open science

Packing Arc-Disjoint Cycles in Tournaments

Stéphane Bessy, Marin Bougeret, Ramaswamy Krithika, Abhishek Sahu,
Saket Saurabh, Jocelyn Thiebaut, Meirav Zehavi

► **To cite this version:**

Stéphane Bessy, Marin Bougeret, Ramaswamy Krithika, Abhishek Sahu, Saket Saurabh, et al..
Packing Arc-Disjoint Cycles in Tournaments. MFCS 2019 - 44th International Symposium on
Mathematical Foundations of Computer Science, Aug 2019, Aachen, Germany. pp.27:1-27:14,
10.4230/LIPIcs.MFCS.2019.27 . hal-02277436

HAL Id: hal-02277436

<https://hal.science/hal-02277436v1>

Submitted on 3 Sep 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

1 Packing Arc-Disjoint Cycles in Tournaments *

2 **Stéphane Bessy**

3 Université de Montpellier, LIRMM, CNRS, Montpellier, France
4 bessy@lirmm.fr

5 **Marin Bougeret**

6 Université de Montpellier, LIRMM, CNRS, Montpellier, France
7 bougeret@lirmm.fr

8 **R. Krithika**

9 Indian Institute of Technology Palakkad, India
10 krithika@iitpkd.ac.in

11 **Abhishek Sahu**

12 The Institute of Mathematical Sciences, HBNI, Chennai, India
13 asahu@imsc.res.in

14 **Saket Saurabh**

15 The Institute of Mathematical Sciences, HBNI, Chennai, India and University of Bergen,
16 Bergen, Norway
17 saket@imsc.res.in

18 **Jocelyn Thiebaut**

19 Université de Montpellier, LIRMM, CNRS, Montpellier, France
20 thiebaut@lirmm.fr

21 **Meirav Zehavi**

22 Ben-Gurion University, Beersheba, Israel
23 meiravze@bgu.ac.il

24 — Abstract —

25 A tournament is a directed graph in which there is a single arc between every pair of distinct
26 vertices. Given a tournament T on n vertices, we explore the classical and parameterized com-
27 plexity of the problems of determining if T has a cycle packing (a set of pairwise arc-disjoint
28 cycles) of size k and a triangle packing (a set of pairwise arc-disjoint triangles) of size k . We
29 refer to these problems as ARC-DISJOINT CYCLES IN TOURNAMENTS (ACT) and ARC-DISJOINT
30 TRIANGLES IN TOURNAMENTS (ATT), respectively. Although the maximization version of ACT
31 can be seen as the linear programming dual of the well-studied problem of finding a minimum
32 feedback arc set (a set of arcs whose deletion results in an acyclic graph) in tournaments, sur-
33 prisingly no algorithmic results seem to exist for ACT. We first show that ACT and ATT are
34 both NP-complete. Then, we show that the problem of determining if a tournament has a cycle
35 packing and a feedback arc set of the same size is NP-complete. Next, we prove that ACT and
36 ATT are fixed-parameter tractable, they can be solved in $2^{\mathcal{O}(k \log k)} n^{\mathcal{O}(1)}$ time and $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$
37 time respectively. Moreover, they both admit a kernel with $\mathcal{O}(k)$ vertices. We also prove that
38 ACT and ATT cannot be solved in $2^{\mathcal{O}(\sqrt{k})} n^{\mathcal{O}(1)}$ time under the Exponential-Time Hypothesis.

39 **2012 ACM Subject Classification** F.2 Analysis of Algorithms and Problem Complexity

40 **Keywords and phrases** arc-disjoint cycle packing, tournaments, parameterized algorithms, ker-
41 nelization

* 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.



© Stéphane Bessy, Marin Bougeret, R. Krithika, Abhishek Sahu, Saket Saurabh, Jocelyn Thiebaut and Meirav Zehavi;

licensed under Creative Commons License CC-BY

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



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

43 **1** Introduction

44 Given a (directed or undirected) graph G and a positive integer k , the DISJOINT CYCLE
 45 PACKING problem is to determine whether G has k (vertex or arc/edge) disjoint (directed
 46 or undirected) cycles. Packing disjoint cycles is a fundamental problem in Graph Theory
 47 and Algorithm Design with applications in several areas. Since the publication of the classic
 48 Erdős-Pósa theorem in 1965 [22], this problem has received significant scientific attention in
 49 various algorithmic realms. In particular, VERTEX-DISJOINT CYCLE PACKING in undirected
 50 graphs is one of the first problems studied in the framework of parameterized complexity.
 51 In this framework, each problem instance is associated with a non-negative integer k called
 52 *parameter*, and a problem is said to be *fixed-parameter tractable* (FPT) if it can be solved in
 53 $f(k)n^{\mathcal{O}(1)}$ time for some computable function f , where n is the input size. For convenience,
 54 the running time $f(k)n^{\mathcal{O}(1)}$ is denoted as $\mathcal{O}^*(f(k))$. A *kernelization algorithm* is a polynomial-
 55 time algorithm that transforms an arbitrary instance of the problem to an equivalent instance
 56 of the same problem whose size is bounded by some computable function g of the parameter
 57 of the original instance. The resulting instance is called a *kernel* and if g is a polynomial
 58 function, then it is called a *polynomial kernel*. A decidable parameterized problem is FPT
 59 if and only if it has a kernel (not necessarily of polynomial size). Kernelization typically
 60 involves applying a set *reduction rules* to the given instance to produce another instance.
 61 A reduction rule is said to be *safe* if it is sound and complete, i.e., applying it to the given
 62 instance produces an equivalent instance. In order to classify parameterized problems as
 63 being FPT or not, the W -hierarchy is defined: $\text{FPT} \subseteq W[1] \subseteq W[2] \subseteq \dots \subseteq \text{XP}$. It is believed
 64 that the subset relations in this sequence are all strict, and a parameterized problem that is
 65 hard for some complexity class above FPT in this hierarchy is said to be fixed-parameter
 66 intractable. Further details on parameterized algorithms can be found in [17, 20, 25, 27].

67 VERTEX-DISJOINT CYCLE PACKING in undirected graphs is FPT with respect to the
 68 solution size k [11, 38] but has no polynomial kernel unless $\text{NP} \subseteq \text{coNP}/\text{poly}$ [12]. In contrast,
 69 EDGE-DISJOINT CYCLE PACKING in undirected graphs admits a kernel with $\mathcal{O}(k \log k)$
 70 vertices (and is therefore FPT) [12]. On directed graphs, these problems have many practical
 71 applications (for example in biology [13, 19]) and they have been extensively studied [7, 36].
 72 It turns out that VERTEX-DISJOINT CYCLE PACKING and ARC-DISJOINT CYCLE PACKING
 73 are equivalent and are $W[1]$ -hard [35, 43]. Therefore, studying these problems on a subclass
 74 of directed graphs is a natural direction of research. Tournaments form a mathematically
 75 rich subclass of directed graphs with interesting structural and algorithmic properties [6, 40].
 76 Tournaments have several applications in modeling round-robin tournaments and in the
 77 study of voting systems and social choice theory [30, 32].

78 FEEDBACK VERTEX SET and FEEDBACK ARC SET are two well-explored algorithmic
 79 problems on tournaments. A *feedback vertex (arc) set* is a set of vertices (arcs) whose deletion
 80 results in an acyclic graph. Given a tournament, MINFAST and MINFVST are the problems
 81 of obtaining a feedback arc set and feedback vertex set of minimum size, respectively. We refer
 82 to the corresponding decision version of the problems as FAST and FVST. The optimization
 83 problems MINFAST and MINFVST have numerous practical applications in the areas of
 84 voting theory [18], machine learning [16], search engine ranking [21] and have been intensively
 85 studied in various algorithmic areas. MINFAST and MINFVST are NP-hard [3, 14] while
 86 FAST and FVST are FPT when parameterized by the solution size k [4, 24, 26, 32]. Further,
 87 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 $\mathcal{O}^*(2^{o(\sqrt{k})})$ 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 $\mathcal{O}^*(2^{\mathcal{O}(k \log k)})$ time (Theorem 16) and admits a kernel with $\mathcal{O}(k)$ vertices (Theorem 15).
- ATT can be solved in $\mathcal{O}^*(2^{\mathcal{O}(k)})$ 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 $\{u \in V(D) : vu \in A(D)\}$ and its *out-degree*, denoted by $d^+(v)$, 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 .

134 **Paths and Cycles.** A *path* P in a digraph D is a sequence (v_1, \dots, v_k) of distinct
 135 vertices such that for each $i \in [k-1]$, $v_i v_{i+1} \in A(D)$. The set $\{v_1, \dots, v_k\}$ is denoted by
 136 $V(P)$ 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
 137 (v_1, \dots, v_k) of distinct vertices such that (v_1, \dots, v_k) is a path and $v_k v_1 \in A(D)$. The length
 138 of a path or cycle X is the number of vertices in it. A cycle on three vertices is called a
 139 *triangle*. A digraph is called a *directed acyclic graph* if it has no cycles. A *feedback arc*
 140 *set* (FAS) is a set of arcs whose deletion results in an acyclic graph. For a digraph D , let
 141 $\text{minfas}(D)$ denote the size of a minimum FAS of D . Any directed acyclic graph D has an
 142 ordering $\sigma(D) = (v_1, \dots, v_n)$ called *topological ordering* of its vertices such that for each
 143 $v_i v_j \in A(D)$, $i < j$ holds. Given an ordering σ and two vertices u and v , we write $u <_\sigma v$ if
 144 u is before v in σ .

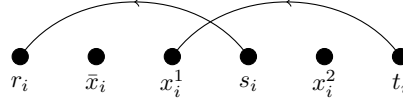
145 **Tournaments.** A *tournament* T is a digraph in which for every pair u, v of distinct
 146 vertices either $uv \in A(T)$ or $vu \in A(T)$ but not both. In other words, a tournament T on n
 147 vertices is an orientation of the complete graph K_n . A tournament T can alternatively be
 148 defined by an ordering $\sigma(T) = (v_1, \dots, v_n)$ of its vertices and a set of *backward arcs* $\overleftarrow{A}_\sigma(T)$
 149 (which will be denoted $\overleftarrow{A}(T)$ as the considered ordering is not ambiguous), where each arc
 150 $a \in \overleftarrow{A}(T)$ 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 $V(T) =$
 151 $\{v_i : i \in [n]\}$ 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
 152 the set of *forward arcs* of T in the given ordering $\sigma(T)$. The pair $(\sigma(T), \overleftarrow{A}(T))$ is called a *linear*
 153 *representation* of the tournament T . A tournament is called *transitive* if it is a directed acyclic
 154 graph and a transitive tournament has a unique topological ordering. Given two tournaments
 155 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
 156 called the *concatenation of T_1 and T_2* , where $V(T) = V(T_1) \cup V(T_2)$, $\sigma(T) = \sigma(T_1)\sigma(T_2)$ is
 157 the concatenation of the two sequences, and $\overleftarrow{A}(T) = \overleftarrow{A}(T_1) \cup \overleftarrow{A}(T_2)$.

158 **3 NP-hardness of MAXACT and MAXATT**

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

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

173 **The variable tournament T_v .** For each variable v_i of F , we define a tournament V_i
 174 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
 175 a representation of one variable gadget V_i . One can notice that the minimum FAS of V_i
 176 corresponds exactly to the set of its backward arcs. We now define $V(T_v)$ be the union
 177 of the vertex sets of the V_i s and we equip T_v with the order $\sigma_1 \sigma_2 \dots \sigma_n$. Thus, T_v has $6n$
 178 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.

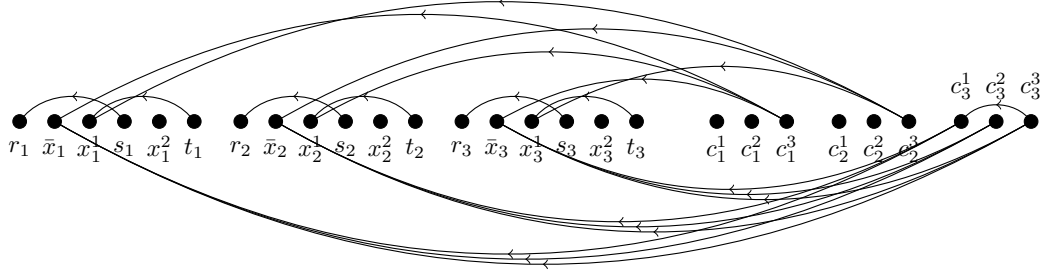
179 edge-disjoint (undirected) triangle packing of K_n covering all its edges with triangles that
 180 can be computed in polynomial time [33]. Let $\{u_1, \dots, u_n\}$ be an arbitrary enumeration of
 181 the vertices of K_n . Using a perfect triangle packing Δ_{K_n} of K_n , we create a tournament
 182 T_{K_n} such that $\sigma'(T_{K_n}) = (u_1, \dots, u_n)$ and $\overleftarrow{A}_{\sigma'}(T_{K_n}) = \{u_k u_i : (u_i, u_j, u_k) \text{ is a triangle of}$
 183 $\Delta_{K_n} \text{ with } i < j < k\}$. Now we set $\overleftarrow{A}_{\sigma}(T_v) = \{xy : x \in V(V_i), y \in V(V_j) \text{ for } i \neq j \text{ and}$
 184 $u_j u_i \in \overleftarrow{A}_{\sigma'}(T_{K_n})\} \cup \bigcup_{i=1}^n \overleftarrow{A}_{\sigma}(V_i)$. In some way, we “blew up” every vertex u_i of T_{K_n} into our
 185 variable gadget V_i .

186 **The clause tournament T_c .** For each of the m clauses c_j of F , we define a tournament
 187 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
 188 $(m+1)^{\text{th}}$ tournament denoted by C_{m+1} and defined by $\sigma(C_{m+1}) = (c_{m+1}^1, c_{m+1}^2, c_{m+1}^3)$
 189 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
 190 *dummy triangle*, and its vertices the *dummy vertices*. We now define T_c such that
 191 $\sigma(T_c)$ is the concatenation of each ordering $\sigma(C_j)$ in the natural order, that is $\sigma(T_c) =$
 192 $(c_1^1, c_1^2, c_1^3, \dots, c_m^1, c_m^2, c_m^3, c_{m+1}^1, c_{m+1}^2, c_{m+1}^3)$. So T_c has $3(m+1)$ vertices. Since $m+1 \equiv 3$
 193 (mod 6), we use the same trick as above to add arcs to $\overleftarrow{A}_{\sigma}(T_c)$ coming from a perfect packing
 194 of undirected triangles of K_{m+1} . Once again, we “blew up” every vertex u_j of $T_{K_{m+1}}$ into
 195 our clause gadget C_j .

196 **The tournament T .** To define our final tournament T let us begin with its ordering
 197 σ defined by $\sigma(T) = \sigma(T_v)\sigma(T_c)$. Then we construct $\overleftarrow{A}^{vc}(T)$ the backward arcs between T_c
 198 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
 199 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
 200 $\ell_u = \bar{v}_{i_u}$, 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
 201 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
 202 “used” by another clause, we chose $z_u = x_{i_u}^2$. Such an orientation will always be possible since
 203 each variable occurs at most two times positively and once negatively in F . If the clause c_j
 204 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
 205 $c_j^2 z_u$ 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 $\ell_u = v_{i_u}$
 206 in such a way that for any $i \in [n]$, there exists a unique arc $a \in \overleftarrow{A}^{vc}(T)$ with $h(a) = x_i^1$.
 207 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
 208 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
 209 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
 210 construction, notice also that T has $6n + 3(m+1)$ vertices and can be computed in polynomial
 211 time. Figure 2 is an example of the tournament obtained from a trivial 3-SAT(3) instance.

212 Now, we move on to proving the correctness of the reduction. First of all, observe that in
 213 each variable gadget V_i , there are only four triangles: let $\delta_i^1, \delta_i^2, \delta_i^3$ and δ_i^4 be the triangles
 214 (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
 215 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
 216 these packings Δ_i^{\top} , $\Delta_i^{\top'}$ and Δ_i^{\perp} , respectively.

217 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 .

218 the Δ -local out-degree of the vertex x , denoted $d_{X \setminus \Delta}^+(x)$, as the remaining out-degree
 219 of x in $T[X]$ when we remove the arcs of the triangles of Δ . More formally, we set:
 220 $d_{X \setminus \Delta}^+(x) = |\{xa : a \in X, xa \in A[X], xa \notin A(\Delta)\}|$.

221 **► Remark.** Given a variable gadget V_i , we have:

- 222 (i) $d_{V_i \setminus \Delta_i^\top}^+(x_i^1) = d_{V_i \setminus \Delta_i^\top}^+(x_i^2) = 1$ and $d_{V_i \setminus \Delta_i^\top}^+(\bar{x}_i) = 3$,
- 223 (ii) $d_{V_i \setminus \Delta_i^{\top'}}^+(x_i^1) = 1$, $d_{V_i \setminus \Delta_i^{\top'}}^+(x_i^2) = 0$ and $d_{V_i \setminus \Delta_i^{\top'}}^+(\bar{x}_i) = 3$,
- 224 (iii) $d_{V_i \setminus \Delta_i^\perp}^+(x_i^1) = d_{V_i \setminus \Delta_i^\perp}^+(x_i^2) = 0$ and $d_{V_i \setminus \Delta_i^\perp}^+(\bar{x}_i) = 4$,
- 225 (iv) none of $\bar{x}_i x_i^1$, $\bar{x}_i x_i^2$, $\bar{x}_i t_i$ belongs to Δ_i^\top or Δ_i^\perp .

226 Informally, we want to set the variable x_i to true (resp. false) when one of the locally-
 227 optimal $\Delta_i^{\top'}$ or Δ_i^\top (resp. Δ_i^\perp) is taken in the variable gadget V_i in the global solution. Now
 228 given a triangle packing Δ of T , we partition Δ into the following sets:

- 229 ■ $\Delta_{V,V,V} = \{(a, b, c) \in \Delta : a \in V_i, b \in V_j, c \in V_k \text{ with } i < j < k\}$,
- 230 ■ $\Delta_{V,V,C} = \{(a, b, c) \in \Delta : a \in V_i, b \in V_j, c \in C_k \text{ with } i < j\}$,
- 231 ■ $\Delta_{V,C,C} = \{(a, b, c) \in \Delta : a \in V_i, b \in C_j, c \in C_k \text{ with } j < k\}$,
- 232 ■ $\Delta_{C,C,C} = \{(a, b, c) \in \Delta : a \in C_i, b \in C_j, c \in C_k \text{ with } i < j < k\}$,
- 233 ■ $\Delta_{2V,C} = \{(a, b, c) \in \Delta : a, b \in V_i, c \in C_j\}$,
- 234 ■ $\Delta_{V,2C} = \{(a, b, c) \in \Delta : a \in V_i, b, c \in C_j\}$,
- 235 ■ $\Delta_{3V} = \{(a, b, c) \in \Delta : a, b, c \in V_i\}$,
- 236 ■ $\Delta_{3C} = \{(a, b, c) \in \Delta : a, b, c \in C_i\}$.

237 Notice that in T , there is no triangle with two vertices in a variable gadget V_i and its
 238 third vertex in a variable gadget V_j with $i \neq j$ since all the arcs between two variable gadgets
 239 are oriented in the same direction. We have the same observation for clauses.

240 In the two next lemmas, we prove some properties concerning the solution Δ , which imply
 241 the result of Lemma 3.

242 **► Lemma 1.** *There exists a triangle packing Δ^v (resp. Δ^c) which uses exactly the arcs between
 243 distinct variable gadgets (resp. clause gadgets). Therefore, we have $|\Delta_{V,V,V}| \leq 6n(n-1)$ and
 244 $|\Delta_{C,C,C}| \leq 3m(m+1)/2$ and these bounds are tight.*

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

248 and V_j in the same orientation as between u_i and u_j . Let $u_i u_j u_k$ be a triangle of the
 249 perfect packing of T_{K_n} . We temporally relabel the vertices of V_i , V_j and V_k respectively by
 250 $\{f_i, i \in [6]\}$, $\{g_i, i \in [6]\}$ and $\{h_i, i \in [6]\}$ and consider the tripartite tournament $K_{6,6,6}$ given
 251 by $V(K_{6,6,6}) = \{f_i, g_i, h_i, i \in [6]\}$ and $A(K_{6,6,6}) = \{f_i g_j, g_i h_j, h_i f_j : i, j \in [6]\}$. Then it is
 252 easy to check that $\{(f_i, g_j, h_{i+j \pmod{6}}) : i, j \in [6]\}$ is a perfect triangle packing of $K_{6,6,6}$.
 253 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
 254 use all the arcs between disjoint variable gadgets. We use the same reasoning to prove that
 255 there exists a triangle packing Δ^c which use all the arcs available in T_c between two distinct
 256 clause gadget. ◀

257 ▶ **Lemma 2.** *For any triangle packing Δ of the tournament T , we have:*

- 258 (i) $|\Delta_{V,V,V}| + |\Delta_{C,C,C}| \leq 6n(n-1) + 3m(m+1)/2$,
- 259 (ii) $|\Delta_{2V,C}| + |\Delta_{V,2C}| + |\Delta_{V,C,C}| + |\Delta_{V,V,C}| \leq |\overleftarrow{A}^{vc}(T)|$,
- 260 (iii) $|\Delta_{3V}| \leq 2n$,
- 261 (iv) $|\Delta_{3C}| \leq 1$.

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

263 **Proof.** Let Δ be a triangle packing of T . Recall that we have: $|\Delta| = |\Delta_{V,V,V}| + |\Delta_{V,V,C}| +$
 264 $|\Delta_{V,C,C}| + |\Delta_{C,C,C}| + |\Delta_{2V,C}| + |\Delta_{V,2C}| + |\Delta_{3V}| + |\Delta_{3C}|$. First, inequality (i) comes from
 265 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
 266 of these sets consumes one backward arc from T_c to T_v . We have $|\Delta_{3V}| \leq 2n$ since we have
 267 at most 2 disjoint triangles in each variable gadget. Finally we also have $|\Delta_{3C}| \leq 1$ since the
 268 dummy triangle is the only triangle lying in a clause gadget. ◀

269 ▶ **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 .*

271 As 3-SAT(3) is NP-hard [41, 44], this implies the following theorem.

272 ▶ **Theorem 4.** *MAXATT is NP-hard.*

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

276 ▶ **Lemma 5.** *Given a 3-SAT(3) instance F , and T the tournament constructed from F
 277 with the reduction f , we have a triangle packing Δ of T of size $6n(n-1) + 3m(m+1)/2 +$
 278 $2n + |\overleftarrow{A}^{vc}(T)| + 1$ if and only if there is a cycle packing O of the same size.*

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

280 ▶ **Theorem 6.** *MAXACT is NP-hard.*

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

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

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

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

295 ► **Theorem 9.** Under the Exponential-time Hypothesis, ATT and ACT cannot be solved in
 296 $\mathcal{O}^*(2^{o(\sqrt{k})})$ time.

297 In the framework of parameterizing above guaranteed values [39], the above results imply
 298 that ACT parameterized below the guaranteed value of the size of a minimal feedback arc
 299 set is fixed-parameter intractable.

300 4 Parameterized Complexity of ACT

301 The classical Erdős-Pósa theorem for cycles in undirected graphs states that for each non-
 302 negative integer k , every undirected graph either contains k vertex-disjoint cycles or has a
 303 feedback vertex set consisting of $f(k) = \mathcal{O}(k \log k)$ vertices [22]. An interesting consequence
 304 of this theorem is that it leads to an FPT algorithm for VERTEX-DISJOINT CYCLE PACKING
 305 (see [38] for more details).

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

309 ► **Theorem 10.** Let k and r be positive integers such that $r \leq k$. A tournament T contains
 310 a set of r arc-disjoint cycles if and only if T contains a set of r arc-disjoint cycles each of
 311 length at most $2k + 1$.

312 **Proof.** The reverse direction of the claim holds trivially. Let us now prove the forward
 313 direction. Let \mathcal{C} be a set of r arc-disjoint cycles in T that minimizes $\sum_{C \in \mathcal{C}} |C|$. If every
 314 cycle in \mathcal{C} is a triangle, then the claim trivially holds. Otherwise, let C be a longest cycle in
 315 \mathcal{C} and let ℓ denote its length. Let v_i, v_j be a pair of non-consecutive vertices in C . Then,
 316 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 $A(C)$
 317 forms a cycle C' of length less than ℓ with $A(C') \setminus \{e\} \subset A(C)$. By our choice of \mathcal{C} , this
 318 implies that e is an arc in some other cycle $\hat{C} \in \mathcal{C}$. This property is true for the arc between
 319 any pair of non-consecutive vertices in C . Therefore, we have $\binom{\ell}{2} - \ell \leq \ell(k - 1)$ leading to
 320 $\ell \leq 2k + 1$. ◀

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

326 We will use the following lemma known from [15] in order to prove Theorem 12¹. For a
 327 digraph D , let $\Lambda(D)$ denote the number of non-adjacent pairs of vertices in D . That is, $\Lambda(D)$
 328 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.

329 ▶ **Lemma 11.** [15] *Let D be a triangle-free digraph in which for every pair u, v of distinct*
 330 *vertices, at most one of uv or vu is in $A(D)$. Then, we can compute an FAS of size at most*
 331 *$\Lambda(D)$ in polynomial time.*

332 ▶ **Theorem 12.** *For every non-negative integer k , every tournament T either contains k*
 333 *arc-disjoint triangles or has an FAS of size at most $5(k-1)$ that can be obtained in polynomial*
 334 *time.*

335 **Proof.** Let \mathcal{C} be a maximal set of arc-disjoint triangles in T (that can be obtained greedily
 336 in polynomial time). If $|\mathcal{C}| \geq k$, then we have the required set of triangles. Otherwise, let
 337 D denote the digraph obtained from T by deleting the arcs that are in some triangle in
 338 \mathcal{C} . Clearly, D has no triangle and $\Lambda(D) \leq 3(k-1)$. Let F be an FAS of D obtained in
 339 polynomial time using Lemma 11. Then, we have $|F| \leq 3(k-1)$. Next, consider a topological
 340 ordering σ of $D - F$. Each triangle of \mathcal{C} contains at most 2 arcs which are backward in this
 341 ordering. If we denote by F' the set of all the arcs of the triangles of \mathcal{C} which are backward
 342 in σ , then we have $|F'| \leq 2(k-1)$ and $(D - F) - F'$ is acyclic. Thus $F^* = F \cup F'$ is an FAS
 343 of T satisfying $|F^*| \leq 5(k-1)$. ◀

344 Next, we show how to obtain a linear kernel for ACT. This kernel is inspired by the
 345 linear kernelization described in [10] for FAST and uses Theorem 12. Let T be a tournament
 346 on n vertices. First, we apply the following reduction rule.

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

348 This rule is clearly safe as our goal is to find k cycles and v cannot be in any of them.
 349 To describe our next rule, we need to state a lemma known from [10]. An *interval* is a
 350 consecutive set of vertices in a linear representation $(\sigma(T), \overleftarrow{A}(T))$ of a tournament T .

351 ▶ **Lemma 13** ([10]). *Let $T = (\sigma(T), \overleftarrow{A}(T))$ be a tournament on which Reduction Rule 4.1 is*
 352 *not applicable. If $|V(T)| \geq 2|\overleftarrow{A}(T)| + 1$, then there exists a partition \mathcal{J} of $V(T)$ into intervals*
 353 *(that can be computed in polynomial time) such that there are $|\overleftarrow{A}(T) \cap E| > 0$ arc-disjoint*
 354 *cycles using only arcs in E where E denotes the set of arcs in T with endpoints in different*
 355 *intervals.*

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

357 ▶ **Reduction Rule 4.2.** *Let $T = (\sigma(T), \overleftarrow{A}(T))$ be a tournament on which Reduction Rule*
 358 *4.1 is not applicable. Let \mathcal{J} be a partition of $V(T)$ into intervals satisfying the properties*
 359 *specified in Lemma 13. Reverse all arcs in $\overleftarrow{A}(T) \cap E$ and decrease k by $|\overleftarrow{A}(T) \cap E|$ where E*
 360 *denotes the set of arcs in T with endpoints in different intervals.*

361 ▶ **Lemma 14.** *Reduction Rule 4.2 is safe.*

362 **Proof.** Let T' be the tournament obtained from T by reversing all arcs in $\overleftarrow{A}(T) \cap E$. Suppose
 363 T' has $k - |\overleftarrow{A}(T) \cap E|$ arc-disjoint cycles. Then, it is guaranteed that each such cycle is
 364 completely contained in an interval. This is due to the fact that T' has no backward arc
 365 with endpoints in different intervals. Indeed, if a cycle in T' uses a forward (backward) arc
 366 with endpoints in different intervals, then it also uses a back (forward) arc with endpoints in
 367 different intervals. It follows that for each arc $uv \in E$, neither uv nor vu is used in these
 368 $k - |\overleftarrow{A}(T) \cap E|$ cycles. Hence, these $k - |\overleftarrow{A}(T) \cap E|$ cycles in T' are also cycles in T . Then,
 369 we can add a set of $|\overleftarrow{A}(T) \cap E|$ cycles obtained from the second property of Lemma 13 to
 370 these $k - |\overleftarrow{A}(T) \cap E|$ cycles to get k cycles in T . Conversely, consider a set of k cycles in

23:10 Packing Arc-Disjoint Cycles in Tournaments

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

375 Thus, we have the following result.

376 ▶ **Theorem 15.** *ACT admits a kernel with $\mathcal{O}(k)$ vertices.*

377 **Proof.** Let (T, k) denote the instance obtained from the input instance by applying Reduction
 378 Rule 4.1 exhaustively. From Lemma 12, we know that either T has k arc-disjoint triangles or
 379 has an FAS of size at most $5(k - 1)$ that can be obtained in polynomial time. In the first
 380 case, we return a trivial yes-instance of constant size as the kernel. In the second case, let F
 381 be the FAS of size at most $5(k - 1)$ of T . Let $(\sigma(T), \overleftarrow{A}(T))$ be the linear representation of T
 382 where $\sigma(T)$ is a topological ordering of the vertices of the directed acyclic graph $T - F$. As
 383 $V(T - F) = V(T)$, $|\overleftarrow{A}(T)| \leq 5(k - 1)$. If $|V(T)| \geq 10k - 9$, then from Lemma 13, there is a
 384 partition of $V(T)$ into intervals with the specified properties. Therefore, Reduction Rule 4.2
 385 is applicable (and the parameter drops by at least 1). When we obtain an instance where
 386 neither of the Reduction Rules 4.1 and 4.2 is applicable, it follows that the tournament in
 387 that instance has at most $10k$ vertices. ◀

388 Finally, we show that ACT can be solved in $\mathcal{O}^*(2^{\mathcal{O}(k \log k)})$ time. The idea is to reduce
 389 the problem to the following ARC-DISJOINT PATHS problem in directed acyclic graphs:
 390 given a digraph D on n vertices and k ordered pairs $(s_1, t_1), \dots, (s_k, t_k)$ of vertices of D , do
 391 there exist arc-disjoint paths P_1, \dots, P_k in D such that P_i is a path from s_i to t_i for each
 392 $i \in [k]$? On directed acyclic graphs, ARC-DISJOINT PATHS is known to be NP-complete
 393 [23], W[1]-hard [43] with respect to k as parameter and solvable in $n^{\mathcal{O}(k)}$ time [28]. Despite
 394 its fixed-parameter intractability, we will show that we can use the $n^{\mathcal{O}(k)}$ algorithm and
 395 Theorems 12 and 15 to describe an FPT algorithm for ACT.

396 ▶ **Theorem 16.** *ACT can be solved in $\mathcal{O}^*(2^{\mathcal{O}(k \log k)})$ time.*

397 **Proof.** Consider an instance (T, k) of ACT. Using Theorem 15, we obtain a kernel $\mathcal{I} = (\widehat{T}, \widehat{k})$
 398 such that \widehat{T} has $\mathcal{O}(k)$ vertices. Further, $\widehat{k} \leq k$. By definition, (T, k) is an yes-instance if
 399 and only if $(\widehat{T}, \widehat{k})$ is an yes-instance. Using Theorem 12, we know that \widehat{T} either contains
 400 \widehat{k} arc-disjoint triangles or has an FAS of size at most $5(\widehat{k} - 1)$ that can be obtained in
 401 polynomial time. If Theorem 12 returns a set of \widehat{k} arc-disjoint triangles in \widehat{T} , then we declare
 402 that (T, k) is an yes-instance.

403 Otherwise, let \widehat{F} be the FAS of size at most $5(\widehat{k} - 1)$ returned by Theorem 12. Let
 404 D denote the (acyclic) digraph obtained from \widehat{T} by deleting \widehat{F} . Observe that D has $\mathcal{O}(k)$
 405 vertices. Suppose \widehat{T} has a set $\mathcal{C} = \{C_1, \dots, C_{\widehat{k}}\}$ of \widehat{k} arc-disjoint cycles. For each $C \in \mathcal{C}$, we
 406 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
 407 $F = \widehat{F} \cap A(\mathcal{C})$. Then, for each cycle $C_i \in \mathcal{C}$, we can guess the arcs F_i from F that it contains
 408 and also the order π_i in which they appear. This information is captured as a partition \mathcal{F} of
 409 F into \widehat{k} sets, F_1 to $F_{\widehat{k}}$ and the set $\{\pi_1, \dots, \pi_{\widehat{k}}\}$ of permutations where π_i is a permutation
 410 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
 411 pair (u, v) , (x, y) of consecutive arcs of F_i with arcs from $A(D)$. That is, there is a path
 412 from $h(\pi_i^{-1}(j))$ and $t(\pi_i^{-1}((j + 1) \bmod |F_i|))$ with arcs from D for each $j \in [|F_i|]$. The total
 413 number of such paths in these \widehat{k} cycles is $\mathcal{O}(|F|)$ and the arcs of these paths are contained in
 414 D which is a (simple) directed acyclic graph.

415 The number of choices for F is $2^{|\widehat{F}|}$ and the number of choices for a partition $\mathcal{F} =$
 416 $\{F_1, \dots, F_{\widehat{k}}\}$ of F and a set $X = \{\pi_1, \dots, \pi_{\widehat{k}}\}$ of permutations is $2^{\mathcal{O}(|\widehat{F}| \log |\widehat{F}|)}$. Once such a
 417 choice is made, the problem of finding \widehat{k} arc-disjoint cycles in \widehat{T} reduces to the problem of
 418 finding \widehat{k} arc-disjoint cycles $\mathcal{C} = \{C_1, \dots, C_{\widehat{k}}\}$ in \widehat{T} such that for each $1 \leq i \leq \widehat{k}$ and for each
 419 $1 \leq j \leq |F_i|$, C_i has a path P_{ij} between $h(\pi_i^{-1}(j))$ and $t(\pi_i^{-1}((j+1) \bmod |F_i|))$ with arcs
 420 from $D = \widehat{T} - \widehat{F}$. This problem is essentially finding $r = \mathcal{O}(|\widehat{F}|)$ arc-disjoint paths in D and
 421 can be solved in $|V(D)|^{\mathcal{O}(r)}$ time using the algorithm in [28]. Therefore, the overall running
 422 time of the algorithm is $\mathcal{O}^*(2^{\mathcal{O}(k \log k)})$ as $|V(D)| = \mathcal{O}(k)$ and $r = \mathcal{O}(k)$. ◀

423 5 Parameterized Complexity of ATT

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

427 In this section, we provide an FPT algorithm and a linear vertex kernel for ATT. First,
 428 it is easy to obtain an $\mathcal{O}^*(2^{\mathcal{O}(k)})$ time algorithm using the classical colour coding technique
 429 [5] for packing subgraphs of bounded size.

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

431 **Proof.** Consider an instance $\mathcal{I} = (T, k)$ of ATT. Let n denote $|V(T)|$ and m denote $|A(T)|$.
 432 Let \mathcal{F} denote the family of colouring functions $c : A(T) \rightarrow [3k]$ of size $2^{\mathcal{O}(k)} \log^2 m$ that
 433 can be computed in $\mathcal{O}^*(2^{\mathcal{O}(k)})$ time using $3k$ -perfect family of hash functions [?]. For each
 434 colouring function c in \mathcal{F} , we colour $A(T)$ according to c and find a triangle packing of size
 435 k whose arcs use different colours. We use a standard dynamic programming routine to
 436 finding such a triangle packing. Clearly, if \mathcal{I} is an yes-instance and \mathcal{C} is a set of k arc-disjoint
 437 triangles in T , there is a colouring function in \mathcal{F} that colours the $3k$ arcs in these triangles
 438 with distinct colours and our algorithm will find the required triangle packing. Given a
 439 colouring $c \in \mathcal{F}$, we first compute for every set of 3 colours $\{a, b, c\}$ whether the arcs coloured
 440 with a, b or c induce a triangle using 3 different colours or not. Then, for every set S of
 441 $3(p+1)$ colours with $p \in [k-1]$, we recursively test if the arcs coloured with the colours in
 442 S induce $p+1$ arc-disjoint triangles whose arcs use all the colours of S . This is achieved by
 443 iterating over every subset $\{a, b, c\}$ of S and checking if there is a triangle using colours a, b
 444 and c and a collection of p arc-disjoint triangles whose arcs use all the colours of $S \setminus \{a, b, c\}$.
 445 For a given S , we can find this collection of triangles in $\mathcal{O}(p^3) = \mathcal{O}(k^3)$ time. Therefore, the
 446 overall running time of the algorithm is $\mathcal{O}^*(2^{\mathcal{O}(k)})$. ◀

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

448 ▶ **Theorem 18.** *ATT admits a kernel with $\mathcal{O}(k)$ vertices.*

449 **Proof.** Let \mathcal{X} be a maximal collection of arc-disjoint triangles of a tournament T obtained
 450 greedily. Let $V_{\mathcal{X}}$ denote the vertices of the triangles in \mathcal{X} and $A_{\mathcal{X}}$ denote the arcs of $V_{\mathcal{X}}$.
 451 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
 452 yes-instance of ATT. Otherwise, $|\mathcal{X}| < k$ and $|V_{\mathcal{X}}| < 3k$. Moreover, notice that $T[U]$ is acyclic
 453 and T does not contain a triangle with one vertex in $V_{\mathcal{X}}$ and two in vertices in U (otherwise
 454 \mathcal{X} would not be maximal).

455 Let B be the (undirected) bipartite graph defined by $V(B) = A_{\mathcal{X}} \cup U$ and $E(B) =$
 456 $\{au : a \in A_{\mathcal{X}}, u \in U \text{ such that } (t(a), h(a), u) \text{ forms a triangle in } T\}$. Let M be a maximum
 457 matching of B and A' (resp. U') denote the vertices of $A_{\mathcal{X}}$ (resp. U) covered by M . Define
 458 $\overline{A'} = A_{\mathcal{X}} \setminus A'$ and $\overline{U'} = U \setminus U'$.

459 We now prove that $(V_{\mathcal{X}} \cup U', k)$ is a linear kernel of (T, k) . Let \mathcal{C} be a maximum sized
 460 triangle packing that minimizes the number of vertices of $\overline{U'}$ belonging to a triangle of \mathcal{C} . By
 461 previous remarks, we can partition \mathcal{C} into $C_{\mathcal{X}} \cup F$ where $C_{\mathcal{X}}$ are the triangles of \mathcal{C} included
 462 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
 463 is clear that F corresponds to a union of vertex-disjoint stars of B with centres in U . Denote
 464 by $U[F]$ the vertices of U clause gadget g to a triangle of F . If $U[F] \subseteq U'$ then $(V_{\mathcal{X}} \cup U', k)$
 465 is immediately a kernel. Suppose there exists a vertex x_0 such that $x_0 \in U[F] \cap \overline{U'}$.

466 We will build a tree rooted in x_0 with edges alternating between F and M . For this let
 467 $H_0 = \{x_0\}$ and construct recursively the sets H_{i+1} such that

$$468 \quad H_{i+1} = \begin{cases} N_F(H_i) & \text{if } i \text{ is even,} \\ N_M(H_i) & \text{if } i \text{ is odd,} \end{cases}$$

469 where, given a subset $S \subseteq U$, $N_F(S) = \{a \in A_{\mathcal{X}} : \exists s \in S \text{ s.t. } (t(a), h(a), s) \in F \text{ and } as \notin M\}$
 470 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 $H_i \subseteq U$
 471 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
 472 union of disjoint stars and M a matching in B . Moreover, for $i \geq 1$ we call T_i the set of edges
 473 between H_i and H_{i-1} . Now we define the tree T such that $V(T) = \bigcup_i H_i$ and $E(T) = \bigcup_i T_i$.
 474 As T_i is a matching (if i is even) or a union of vertex-disjoint stars with centres in H_{i-1} (if
 475 i is odd), it is clear that T is a tree.

476 For i being odd, every vertex of H_i is incident to an edge of M otherwise B would contain
 477 an augmenting path for M , a contradiction. So every leaf of T is in U and incident to an
 478 edge of M in T and T contains as many edges of M than edges of F . Now for every arc
 479 $a \in A_{\mathcal{X}} \cap V(T)$ we replace the triangle of \mathcal{C} containing a and corresponding to an edge of F
 480 by the triangle $(t(a), h(a), u)$ where $au \in M$ (and au is an edge of T). This operation leads
 481 to another collection of arc-disjoint triangles with the same size as \mathcal{C} but containing a strictly
 482 smaller number of vertices in $\overline{U'}$, yielding a contradiction.

483 Finally $V_{\mathcal{X}} \cup U'$ can be computed in polynomial time and we have $|V_{\mathcal{X}} \cup U'| \leq |V_{\mathcal{X}}| + |M| \leq$
 484 $2|V_{\mathcal{X}}| \leq 6k$, which proves that the kernel has $\mathcal{O}(k)$ vertices. \blacktriangleleft

485 6 Concluding Remarks

486 In this work, we studied the classical and parameterized complexity of packing arc-disjoint
 487 cycles and triangles in tournaments. We showed NP-hardness, fixed-parameter tractability
 488 and linear kernelization results. An interesting problem could be to find subclasses of
 489 tournaments where these problems are polynomial-time solvable. For instance, we show
 490 in the full version of the paper that it is the case for sparse tournaments, that is for
 491 tournaments which admit an FAS that is a matching. This class of tournaments is worthy of
 492 attention for these packing problems as packing vertex-disjoint triangles (and hence cycles)
 493 in sparse tournaments is NP-complete [8]. To conclude, observe that very few problems on
 494 tournaments are known to admit an $\mathcal{O}^*(2^{\sqrt{k}})$ -time algorithm when parameterized by the
 495 standard parameter k [42] - FAST is one of them [4, 24]. To the best of our knowledge,
 496 outside bidimensionality theory, there are no packing problems that are known to admit such
 497 subexponential algorithms. In light of the $2^{o(\sqrt{k})}$ lower bound shown for ACT and ATT, it
 498 would be interesting to explore if these problems admit $\mathcal{O}^*(2^{\mathcal{O}(\sqrt{k})})$ algorithms.

499 — References —

- 500 1 F. N. Abu-Khzam. An Improved Kernelization Algorithm for r-Set Packing. *Inf. Process.*
 501 *Lett.*, 110(16):621–624, 2010.

- 502 2 I. Akaria and R. Yuster. Packing Edge-Disjoint Triangles in Regular and Almost Regular
503 Tournaments. *Discrete Math.*, 338(2):217–228, 2015.
- 504 3 N. Alon. Ranking Tournaments. *SIAM J. Discrete Math.*, 20(1):137–142, 2006.
- 505 4 N. Alon, D. Lokshtanov, and S. Saurabh. Fast FAST. In *36th International Colloquium on*
506 *Automata, Languages, and Programming (ICALP) Proc., Part I*, pages 49–58, 2009.
- 507 5 N. Alon, R. Yuster, and U. Zwick. Color-Coding. *J. ACM*, 42(4):844–856, 1995.
- 508 6 J. Bang-Jensen and G. Gutin. Paths, Trees and Cycles in Tournaments. *Congressus*
509 *Numerantium*, 115:131–170, 1996.
- 510 7 J. Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms and Applications*. Springer-
511 Verlag London, 2009.
- 512 8 S. Bessy, M. Bougeret, and J. Thiebaut. Triangle Packing in (Sparse) Tournaments: Ap-
513 proximation and Kernelization. In *25th Annual European Symp. on Algorithms (ESA 2017)*,
514 volume 87, pages 14:1–14:13, 2017.
- 515 9 S. Bessy, M. Bougeret, and J. Thiebaut. (Arc-disjoint) Cycle Packing in Tournament:
516 Classical and Parameterized Complexity, 2018. [arXiv:1802.06669](https://arxiv.org/abs/1802.06669).
- 517 10 S. Bessy, F. V. Fomin, S. Gaspers, C. Paul, A. Perez, S. Saurabh, and S. Thomassé. Kernels
518 for Feedback Arc Set in Tournaments. *J. Comput. Syst. Sci.*, 77(6):1071–1078, 2011.
- 519 11 H. L. Bodlaender. On Disjoint Cycles. *Int. J. Found. Comput. S.*, 5(1):59–68, 1994.
- 520 12 H. L. Bodlaender, S. Thomassé, and A. Yeo. Kernel Bounds for Disjoint Cycles and Disjoint
521 Paths. *Theor. Comput. Sci.*, 412(35):4570–4578, 2011.
- 522 13 A. Caprara, A. Panconesi, and R. Rizzi. Packing Cycles in Undirected Graphs. *J. Al-*
523 *gorithms*, 48(1):239 – 256, 2003.
- 524 14 P. Charbit, S. Thomassé, and A. Yeo. The Minimum Feedback Arc Set Problem is NP-Hard
525 for Tournaments. *Comb Probab Comput.*, 16(1):1–4, 2007.
- 526 15 M. Chudnovsky, P. Seymour, and B. Sullivan. Cycles in Dense Digraphs. *Combinatorica*,
527 28(1):1–18, 2008.
- 528 16 W. W. Cohen, R. E. Schapire, and Y. Singer. Learning to Order Things. In *Advances in*
529 *Neural Information Processing Systems Conference (NIPS)*, pages 451–457, 1997.
- 530 17 M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk,
531 and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.
- 532 18 Jean-Charles de Borda. Mémoire sur les élections au scrutin. *Histoire de l'Académie Royale*
533 *des Sciences*, 1781.
- 534 19 D. Dorninger. Hamiltonian Circuits Determining the Order of Chromosomes. *Discrete*
535 *Appl. Math.*, 50(2):159–168, 1994.
- 536 20 R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Springer-
537 Verlag London, 2013.
- 538 21 C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank Aggregation Methods for the
539 Web. In *Proc. of the 10th International World Wide Web Conference*, pages 613–622, 2001.
- 540 22 P. Erdős and L. Pósa. On Independent Circuits Contained in a Graph. *Canadian J. Math.*,
541 17:347–352, 1965.
- 542 23 S. Even, A. Itai, and A. Shamir. On the Complexity of Timetable and Multicommodity
543 Flow Problems. *SIAM J. Comput.*, 5(4):691–703, 1976.
- 544 24 U. Feige. Faster FAST(Feedback Arc Set in Tournaments), 2009. [arXiv:0911.5094](https://arxiv.org/abs/0911.5094).
- 545 25 J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006.
- 546 26 F. V. Fomin, D. Lokshtanov, V. Raman, and S. Saurabh. Fast Local Search Algorithm for
547 Weighted Feedback Arc Set in Tournaments. In *Proc. of the 24th AAAI Conf. on Artificial*
548 *Intelligence*, pages 65–70, 2010.
- 549 27 F. V. Fomin, D. Lokshtanov, S. Saurabh, and M. Zehavi. *Kernelization: Theory of Para-*
550 *meterized Preprocessing*. Cambridge University Press, 2019.

- 551 **28** S. Fortune, J. Hopcroft, and J. Wyllie. The Directed Subgraph Homeomorphism Problem.
552 *Theor. Comput. Sci.*, 10(2):111–121, 1980.
- 553 **29** R. B. Gardner. Optimal Packings and Coverings of the Complete Directed Graph with
554 3-Circuits and with Transitive Triples. In *Proc. of the 28th Southeastern International*
555 *Conference on Combinatorics, Graph Theory and Computing*, volume 127, pages 161–170,
556 1997.
- 557 **30** E. Hemaspaandra, H. Spakowski, and J. Vogel. The Complexity of Kemeny Elections.
558 *Theor. Comput. Sci.*, 349(3):382–391, 2005.
- 559 **31** R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential com-
560 plexity? *J. Comput. Syst. Sci.*, 63(4):512 – 530, 2001.
- 561 **32** M. Karpinski and W. Schudy. Faster Algorithms for Feedback Arc Set Tournament, Ke-
562 meny Rank Aggregation and Betweenness Tournament. In *21st International Symp. on*
563 *Algorithms and Computation (ISAAC)*, pages 3–14. Springer Berlin Heidelberg, 2010.
- 564 **33** T. P. Kirkman. On a Problem in Combinations. *Cambridge and Dublin Mathematical*
565 *Journal*, 2:191–204, 1847.
- 566 **34** R. Krithika, A. Sahu, S. Saurabh, and M. Zehavi. The Parameterized Complexity of
567 Packing Arc-Disjoint Cycles in Tournaments, 2018. [arXiv:1802.07090](https://arxiv.org/abs/1802.07090).
- 568 **35** M. Krivelevich, Z. Nutov, M. R. Salavatipour, J. V. Yuster, and R. Yuster. Approximation
569 Algorithms and Hardness Results for Cycle Packing Problems. *ACM Transactions on*
570 *Algorithms*, 3(4), 2007.
- 571 **36** M. Krivelevich, Z. Nutov, and R. Yuster. Approximation Algorithms for Cycle Packing
572 Problems. In *Proc. of the 16th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA)*,
573 pages 556–561, 2005.
- 574 **37** T. Le, D. Lokshtanov, S. Saurabh, S. Thomassé, and M. Zehavi. Subquadratic Kernels for
575 Implicit 3-Hitting Set and 3-Set Packing Problems. In *Proc. of the 29th Annual ACM-SIAM*
576 *Symp. on Discrete Algorithms (SODA)*, pages 331–342, 2018.
- 577 **38** D. Lokshtanov, A. Mouawad, S. Saurabh, and M. Zehavi. Packing Cycles Faster Than
578 Erdős-Pósa. In *44th International Colloquium on Automata, Languages, and Programming*
579 *(ICALP)*, pages 71:1–71:15, 2017.
- 580 **39** M. Mahajan and V. Raman. Parameterizing Above Guaranteed Values: MaxSat and
581 MaxCut. *J. Algorithms*, 31(2):335–354, 1999.
- 582 **40** J.W. Moon. *Topics on Tournaments*. Holt, Rinehart and Winston, New York, 1968.
- 583 **41** C. H. Papadimitriou. *Computational Complexity*. John Wiley and Sons Ltd., 2003.
- 584 **42** M. Pilipczuk. *Tournaments and Optimality: New Results in Parameterized Complexity*.
585 PhD thesis, The University of Bergen, 2013.
- 586 **43** A. Slivkins. Parameterized Tractability of Edge-Disjoint Paths on Directed Acyclic Graphs.
587 *SIAM J. Discrete Math.*, 24(1):146–157, 2010.
- 588 **44** C. A. Tovey. A simplified NP-complete satisfiability problem. *Discrete Appl. Math.*, 8(1):85
589 – 89, 1984.
- 590 **45** R. Yuster. Packing Triangles in Regular Tournaments. *J. of Graph Theory*, 74(1):58–66,
591 2013.

1 Packing Arc-Disjoint Cycles in Tournaments *

2 **Stéphane Bessy**

3 Université de Montpellier, LIRMM, CNRS, Montpellier, France
4 bessy@lirmm.fr

5 **Marin Bougeret**⁵⁹²

6 Université de Montpellier, LIRMM, CNRS, Montpellier, France
7 bougeret@lirmm.fr

8 **R. Krithika**

9 Indian Institute of Technology Palakkad, India
10 krithika@iitpkd.ac.in

11 **Abhishek Sahu**

12 The Institute of Mathematical Sciences, HBNI, Chennai, India
13 asahu@imsc.res.in

14 **Saket Saurabh**

15 The Institute of Mathematical Sciences, HBNI, Chennai, India and University of Bergen,
16 Bergen, Norway
17 saket@imsc.res.in

18 **Jocelyn Thiebaut**

19 Université de Montpellier, LIRMM, CNRS, Montpellier, France
20 thiebaut@lirmm.fr

21 **Meirav Zehavi**

22 Ben-Gurion University, Beersheba, Israel
23 meiravze@bgu.ac.il

24 — Abstract —

25 A tournament is a directed graph in which there is a single arc between every pair of distinct
26 vertices. Given a tournament T on n vertices, we explore the classical and parameterized com-
27 plexity of the problems of determining if T has a cycle packing (a set of pairwise arc-disjoint
28 cycles) of size k and a triangle packing (a set of pairwise arc-disjoint triangles) of size k . We
29 refer to these problems as ARC-DISJOINT CYCLES IN TOURNAMENTS (ACT) and ARC-DISJOINT
30 TRIANGLES IN TOURNAMENTS (ATT), respectively. Although the maximization version of ACT
31 can be seen as the linear programming dual of the well-studied problem of finding a minimum
32 feedback arc set (a set of arcs whose deletion results in an acyclic graph) in tournaments, sur-
33 prisingly no algorithmic results seem to exist for ACT. We first show that ACT and ATT are
34 both NP-complete. Then, we show that the problem of determining if a tournament has a cycle
35 packing and a feedback arc set of the same size is NP-complete. Next, we prove that ACT is
36 fixed-parameter tractable and admits a polynomial kernel when parameterized by k . In particu-
37 lar, we show that ACT has a kernel with $\mathcal{O}(k)$ vertices and can be solved in $2^{\mathcal{O}(k \log k)} n^{\mathcal{O}(1)}$
38 time. Then, we show that ATT too has a kernel with $\mathcal{O}(k)$ vertices and can be solved in $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$
39 time. Afterwards, we describe polynomial-time algorithms for ACT and ATT when the input
40 tournament has a feedback arc set that is a matching. We also prove that ACT and ATT cannot
41 be solved in $2^{\mathcal{O}(\sqrt{k})} n^{\mathcal{O}(1)}$ time under the Exponential-Time Hypothesis.

42 **2012 ACM Subject Classification** F.2 Analysis of Algorithms and Problem Complexity

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



© Stéphane Bessy, Marin Bougeret, R. Krithika, Abhishek Sahu, Saket Saurabh, Jocelyn Thiebaut and Meirav Zehavi;

licensed under Creative Commons License CC-BY

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



LIPIC Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

43 **Keywords and phrases** arc-disjoint cycle packing, tournaments, parameterized algorithms, ker-
 44 nelization

45 **Digital Object Identifier** 10.4230/LIPIcs.CVIT.2016.23

593

46 **1** Introduction

47 Given a (directed or undirected) graph G and a positive integer k , the DISJOINT CYCLE
 48 PACKING problem is to determine whether G has k (vertex or arc/edge) disjoint (directed
 49 or undirected) cycles. Packing disjoint cycles is a fundamental problem in Graph Theory
 50 and Algorithm Design with applications in several areas. Since the publication of the classic
 51 Erdős-Pósa theorem in 1965 [26], this problem has received significant scientific attention in
 52 various algorithmic realms. In particular, VERTEX-DISJOINT CYCLE PACKING in undirected
 53 graphs is one of the first problems studied in the framework of parameterized complexity.
 54 In this framework, each problem instance is associated with a non-negative integer k called
 55 *parameter*, and a problem is said to be *fixed-parameter tractable* (FPT) if it can be solved in
 56 $f(k)n^{\mathcal{O}(1)}$ time for some computable function f , where n is the input size. For convenience,
 57 the running time $f(k)n^{\mathcal{O}(1)}$ where f grows super-polynomially with k is denoted as $\mathcal{O}^*(f(k))$.
 58 A *kernelization algorithm* is a polynomial-time algorithm that transforms an arbitrary instance
 59 of the problem to an equivalent instance of the same problem whose size is bounded by some
 60 computable function g of the parameter of the original instance. The resulting instance is
 61 called a *kernel* and if g is a polynomial function, then it is called a *polynomial kernel* and
 62 we say that the problem admits a polynomial kernel. A decidable parameterized problem
 63 is FPT if and only if it has a kernel (not necessarily of polynomial size). Kernelization
 64 typically involves applying a set of rules (called *reduction rules*) to the given instance to
 65 produce another instance. A reduction rule is said to be *safe* if it is sound and complete,
 66 i.e., applying it to the given instance produces an equivalent instance. In order to classify
 67 parameterized problems as being FPT or not, the W -hierarchy is defined: $\text{FPT} \subseteq W[1] \subseteq$
 68 $W[2] \subseteq \dots \subseteq \text{XP}$. It is believed that the subset relations in this sequence are all strict, and a
 69 parameterized problem that is hard for some complexity class above FPT in this hierarchy
 70 is said to be fixed-parameter intractable. As mentioned before, the set of parameterized
 71 problems that admit a polynomial kernel is contained in the class FPT and it is believed
 72 that this subset relation is also strict. Further details on parameterized algorithms can be
 73 found in [21, 24, 29, 31].

74 VERTEX-DISJOINT CYCLE PACKING in undirected graphs is FPT with respect to the
 75 solution size k [12, 43] but has no polynomial kernel unless $\text{NP} \subseteq \text{coNP}/\text{poly}$ [13]. In contrast,
 76 EDGE-DISJOINT CYCLE PACKING in undirected graphs admits a kernel with $\mathcal{O}(k \log k)$
 77 vertices (and is therefore FPT) [13]. On directed graphs, these problems have many practical
 78 applications (for example in biology [14, 23]) and they have been extensively studied [7, 40, 44].
 79 It turns out that VERTEX-DISJOINT CYCLE PACKING and ARC-DISJOINT CYCLE PACKING
 80 are equivalent and are $W[1]$ -hard [39, 52]. Therefore, studying these problems on a subclass
 81 of directed graphs is a natural direction of research. Tournaments form a mathematically
 82 rich subclass of directed graphs with interesting structural and algorithmic properties [6, 46].
 83 A *tournament* is a directed graph in which there is a single arc between every pair of distinct
 84 vertices. Tournaments have several applications in modeling round-robin tournaments and in
 85 the study of voting systems and social choice theory [34, 36, 42]. Further, the combinatorics
 86 of inclusion relations of tournaments is reasonably well-understood [16]. A seminal result in
 87 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 $\mathcal{O}^*(2^{\mathcal{O}(\sqrt{k})})$ 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 $\mathcal{O}^*(2^{\mathcal{O}(k \log k)})$ time (Theorem 17) and admits a kernel with $\mathcal{O}(k)$ vertices (Theorem 16).
- ATT can be solved in $\mathcal{O}^*(2^{\mathcal{O}(k)})$ 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 $\{1, 2, \dots, n\}$ 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 $h(e) = v$ as the head of e and $t(e) = u$ as the tail of e . For a vertex $v \in V(D)$, its *out-neighbourhood*, denoted by $N^+(v)$, is the set $\{u \in V(D) : vu \in A(D)\}$ and its *in-neighbourhood*, denoted by $N^-(v)$, is the set $\{u \in V(D) : uv \in A(D)\}$. 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 .

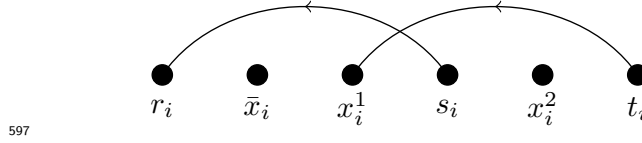
Paths and Cycles. A *path* P in a digraph D is a sequence (v_1, \dots, v_k) of distinct vertices such that for each $i \in [k - 1]$, $v_i v_{i+1} \in A(D)$. The set $\{v_1, \dots, v_k\}$ is denoted by $V(P)$ and the set $\{v_i v_{i+1} : i \in [k - 1]\}$ is denoted by $A(P)$. A path $P = (v_1, \dots, v_k)$ 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, \dots, v_k) of distinct vertices such that (v_1, \dots, v_k) is a path and $v_k v_1 \in A(D)$. The set

181 $\{v_1, \dots, v_k\}$ 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 $A(C)$.
 182 A cycle $C = (v_1, \dots, v_k)$ is called an *induced* (or *chordless*) cycle if $A(C)$ are the only arcs
 183 of $D[V(C)]$. The length of a path or cycle X is the number of vertices in it and is denoted
 184 by $|X|$. For a set \mathcal{C} of paths or cycles, $V(\mathcal{C})$ denotes the set $\{v \in V(D) : \exists C \in \mathcal{C}, v \in V(C)\}$
 185 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
 186 a *triangle*. A digraph is said to be *triangle-free* if it has no triangles. A set of pairwise
 187 arc-disjoint cycles is called a *cycle packing* and a set of pairwise arc-disjoint triangles is called
 188 a *triangle packing*. A digraph is called a *directed acyclic graph* if it has no cycles. A *feedback*
 189 *arc set* (FAS) is a set of arcs whose deletion results in an acyclic graph. For a digraph D ,
 190 let $\text{minfas}(D)$ denote the size of a minimum FAS of D . Any directed acyclic graph D has
 191 an ordering $\sigma(D) = (v_1, \dots, v_n)$ called *topological ordering* of its vertices such that for each
 192 $v_i v_j \in A(D)$, $i < j$ holds. Given an ordering σ and two vertices u and v , we write $u <_\sigma v$ if
 193 u is before v in σ .

194 **Tournaments.** A *tournament* T is a digraph in which for every pair u, v of distinct vertices
 195 either $uv \in A(T)$ or $vu \in A(T)$ but not both. In other words, a tournament T on n vertices
 196 is an orientation of the complete graph K_n . A tournament T can alternatively be defined by
 197 an ordering $\sigma(T) = (v_1, \dots, v_n)$ of its vertices and a set of *backward arcs* $\overleftarrow{A}_\sigma(T)$ (which will
 198 be denoted $\overleftarrow{A}(T)$ as the considered ordering is not ambiguous), where each arc $a \in \overleftarrow{A}(T)$ is of
 199 the form $v_{i_1} v_{i_2}$ with $i_2 < i_1$. Indeed, given $\sigma(T)$ and $\overleftarrow{A}(T)$, we define $V(T) = \{v_i : i \in [n]\}$
 200 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
 201 of *forward arcs* of T in the given ordering $\sigma(T)$. The pair $(\sigma(T), \overleftarrow{A}(T))$ is called a *linear*
 202 *representation* of the tournament T . A tournament is called *transitive* if it is a directed
 203 acyclic graph and a transitive tournament has a unique topological ordering. It is clear that
 204 for any linear representation $(\sigma(T), \overleftarrow{A}(T))$ of T the set $\overleftarrow{A}(T)$ is an FAS of T . A tournament
 205 is *sparse* if it admits an FAS which is a matching. Given a linear representation $(\sigma(T), \overleftarrow{A}(T))$
 206 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
 207 $v_{i_3} v_{i_1} \in \overleftarrow{A}(T)$, $v_{i_3} v_{i_2} \notin \overleftarrow{A}(T)$ and $v_{i_2} v_{i_1} \notin \overleftarrow{A}(T)$ (in this case we call C a *triangle with*
 208 *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
 209 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
 210 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
 211 the *concatenation of T_1 and T_2* , where $V(T) = V(T_1) \cup V(T_2)$, $\sigma(T) = \sigma(T_1)\sigma(T_2)$ is the
 212 concatenation of the two sequences, and $\overleftarrow{A}(T) = \overleftarrow{A}(T_1) \cup \overleftarrow{A}(T_2)$.

213 **3 NP-hardness of MAXACT and MAXATT**

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



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

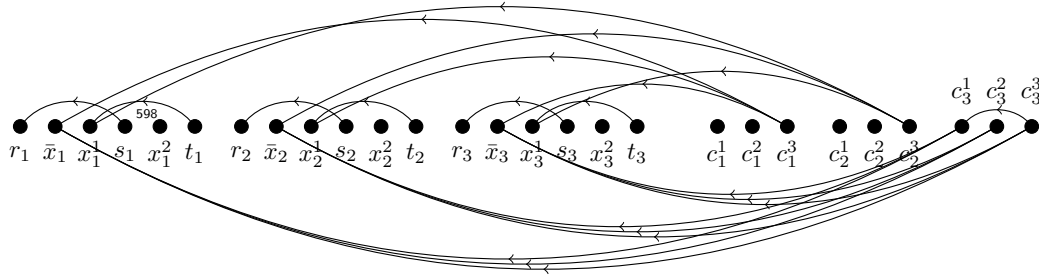
226 In the following, let f be the reduction that maps an instance F of 3-SAT(3) to a
 227 tournament T we describe now.

228 **The variable tournament T_v .** For each variable v_i of F , we define a tournament V_i of
 229 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
 230 representation of one variable gadget V_i . One can notice that the minimum FAS of V_i
 231 corresponds exactly to the set of its backward arcs. We now define $V(T_v)$ be the union
 232 of the vertex sets of the V_i s and we equip T_v with the order $\sigma_1 \sigma_2 \dots \sigma_n$. Thus, T_v has $6n$
 233 vertices. We also add the following backward arcs to T_v . Since $n \equiv 3 \pmod{6}$, there is an
 234 edge-disjoint (undirected) triangle packing of K_n covering all its edges with triangles that
 235 can be computed in polynomial time [37]. Let $\{u_1, \dots, u_n\}$ be an arbitrary enumeration of
 236 the vertices of K_n . Using a perfect triangle packing Δ_{K_n} of K_n , we create a tournament
 237 T_{K_n} such that $\sigma'(T_{K_n}) = (u_1, \dots, u_n)$ and $\overleftarrow{A}_{\sigma'}(T_{K_n}) = \{u_k u_i : (u_i, u_j, u_k) \text{ is a triangle of}$
 238 $\Delta_{K_n} \text{ with } i < j < k\}$. Now we set $\overleftarrow{A}_\sigma(T_v) = \{xy : x \in V(V_i), y \in V(V_j) \text{ for } i \neq j \text{ and}$
 239 $u_j u_i \in \overleftarrow{A}_{\sigma'}(T_{K_n})\} \cup \bigcup_{i=1}^n \overleftarrow{A}_\sigma(V_i)$. In some way, we “blew up” every vertex u_i of T_{K_n} into our
 240 variable gadget V_i .

241 **The clause tournament T_c .** For each of the m clauses c_j of F , we define a tournament C_j of
 242 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 $(m+1)^{th}$ tour-
 243 nament 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 $\overleftarrow{A}_\sigma(C_{m+1}) =$
 244 $\{c_{m+1}^3 c_{m+1}^1\}$, that is C_{m+1} is a triangle. We call this triangle the *dummy triangle*, and its ver-
 245 tices the *dummy vertices*. We now define T_c such that $\sigma(T_c)$ is the concatenation of each order-
 246 ing $\sigma(C_j)$ in the natural order, that is $\sigma(T_c) = (c_1^1, c_1^2, c_1^3, \dots, c_m^1, c_m^2, c_m^3, c_{m+1}^1, c_{m+1}^2, c_{m+1}^3)$.
 247 So T_c has $3(m+1)$ vertices. Since $m+1 \equiv 3 \pmod{6}$, we use the same trick as above to
 248 add arcs to $\overleftarrow{A}_\sigma(T_c)$ coming from a perfect packing of undirected triangles of K_{m+1} . Once
 249 again, we “blew up” every vertex u_j of $T_{K_{m+1}}$ into our clause gadget C_j .

250 **The tournament T .** To define our final tournament T let us begin with its ordering σ
 251 defined by $\sigma(T) = \sigma(T_v)\sigma(T_c)$. Then we construct $\overleftarrow{A}^{vc}(T)$ the backward arcs between T_c
 252 and T_v . For any $j \in [m]$, if the clause c_j in F has three literals, that is $c_j = \ell_1 \vee \ell_2 \vee \ell_3$,
 253 then we add to $\overleftarrow{A}^{vc}(T)$ the three backward arcs $c_j^3 z_u$ where $u \in [3]$ and such that $z_u = \bar{x}_{i_u}$
 254 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
 255 exists a unique arc $a \in \overleftarrow{A}^{vc}(T)$ with $h(a) = x_{i_u}^1$. Informally, in the previous definition, if $x_{i_u}^1$
 256 is already “used” by another clause, we chose $z_u = x_{i_u}^2$. Such an orientation will always be
 257 possible since each variable occurs at most two times positively and once negatively in F . If
 258 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
 259 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 $z_u \in \{x_{i_u}^1, x_{i_u}^2\}$
 260 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
 261 $h(a) = x_{i_u}^1$.

262 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
 263 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 .

264 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
 265 construction, notice also that T has $6n + 3(m + 1)$ vertices and can be computed in polynomial
 266 time. Figure 2 is an example of the tournament obtained from a trivial 3-SAT(3) instance.

267 Now, we move on to proving the correctness of the reduction. First of all, observe that in
 268 each variable gadget V_i , there are only four triangles: let $\delta_i^1, \delta_i^2, \delta_i^3$ and δ_i^4 be the triangles
 269 (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
 270 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
 271 these packings Δ_i^\top , $\Delta_i^{\top'}$ and Δ_i^\perp , respectively.

272 Given a triangle packing Δ of T and a subset X of vertices, we define for any $x \in X$
 273 the Δ -local out-degree of the vertex x , denoted $d_{X \setminus \Delta}^+(x)$, as the remaining out-degree
 274 of x in $T[X]$ when we remove the arcs of the triangles of Δ . More formally, we set:
 275 $d_{X \setminus \Delta}^+(x) = |\{xa : a \in X, xa \in A[X], xa \notin A(\Delta)\}|$.

276 ► **Remark.** Given a variable gadget V_i , we have:

- 277 (i) $d_{V_i \setminus \Delta_i^\top}^+(x_i^1) = d_{V_i \setminus \Delta_i^\top}^+(x_i^2) = 1$ and $d_{V_i \setminus \Delta_i^\top}^+(\bar{x}_i) = 3$,
 278 (ii) $d_{V_i \setminus \Delta_i^{\top'}}^+(x_i^1) = 1$, $d_{V_i \setminus \Delta_i^{\top'}}^+(x_i^2) = 0$ and $d_{V_i \setminus \Delta_i^{\top'}}^+(\bar{x}_i) = 3$,
 279 (iii) $d_{V_i \setminus \Delta_i^\perp}^+(x_i^1) = d_{V_i \setminus \Delta_i^\perp}^+(x_i^2) = 0$ and $d_{V_i \setminus \Delta_i^\perp}^+(\bar{x}_i) = 4$,
 280 (iv) none of $\bar{x}_i x_i^1, \bar{x}_i x_i^2, \bar{x}_i t_i$ belongs to Δ_i^\top or Δ_i^\perp .

281 Informally, we want to set the variable x_i to true (resp. false) when one of the locally-
 282 optimal $\Delta_i^{\top'}$ or Δ_i^\top (resp. Δ_i^\perp) is taken in the variable gadget V_i in the global solution. Now
 283 given a triangle packing Δ of T , we partition Δ into the following sets:

- 284 ■ $\Delta_{V,V,V} = \{(a, b, c) \in \Delta : a \in V_i, b \in V_j, c \in V_k \text{ with } i < j < k\}$,
 285 ■ $\Delta_{V,V,C} = \{(a, b, c) \in \Delta : a \in V_i, b \in V_j, c \in C_k \text{ with } i < j\}$,
 286 ■ $\Delta_{V,C,C} = \{(a, b, c) \in \Delta : a \in V_i, b \in C_j, c \in C_k \text{ with } j < k\}$,
 287 ■ $\Delta_{C,C,C} = \{(a, b, c) \in \Delta : a \in C_i, b \in C_j, c \in C_k \text{ with } i < j < k\}$,
 288 ■ $\Delta_{2V,C} = \{(a, b, c) \in \Delta : a, b \in V_i, c \in C_j\}$,
 289 ■ $\Delta_{V,2C} = \{(a, b, c) \in \Delta : a \in V_i, b, c \in C_j\}$,
 290 ■ $\Delta_{3V} = \{(a, b, c) \in \Delta : a, b, c \in V_i\}$,
 291 ■ $\Delta_{3C} = \{(a, b, c) \in \Delta : a, b, c \in C_i\}$.

292 Notice that in T , there is no triangle with two vertices in a variable gadget V_i and its
 293 third vertex in a variable gadget V_j with $i \neq j$ since all the arcs between two variable gadgets
 294 are oriented in the same direction. We have the same observation for clauses.

295 In the two next lemmas, we prove some properties concerning the solution Δ .

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

299 **Proof.** First recall that the tournament T_v is constructed from a tournament T_{K_n} which
 300 admits a perfect packing of $n(n-1)/6$ triangles. Then we replaced each vertex u_i in T_{K_n}
 301 by the variable gadget V_i and kept all the arcs between two variable gadgets V_i and V_j
 302 in the same orientation as between u_i and u_j . Let $u_i u_j u_k$ be a triangle of the perfect packing
 303 of T_{K_n} . We temporarily relabel the vertices of V_i , V_j and V_k respectively by $\{f_i: i \in [6]\}$,
 304 $\{g_i: i \in [6]\}$ and $\{h_i: i \in [6]\}$ and consider the tripartite tournament $K_{6,6,6}$ given by
 305 $V(K_{6,6,6}) = \{f_i, g_i, h_i: i \in [6]\}$ and $A(K_{6,6,6}) = \{f_i g_j, g_i h_j, h_i f_j: i, j \in [6]\}$. Then it is easy
 306 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
 307 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
 308 all the arcs between disjoint variable gadgets. We use the same reasoning to prove that there
 309 exists a triangle packing Δ^c which use all the arcs available in T_c between two distinct clause
 310 gadget. ◀

311 ► **Lemma 2.** *For any triangle packing Δ of the tournament T , we have the following*
 312 *inequalities:*

- 313 (i) $|\Delta_{V,V,V}| + |\Delta_{C,C,C}| \leq 6n(n-1) + 3m(m+1)/2$,
- 314 (ii) $|\Delta_{2V,C}| + |\Delta_{V,2C}| + |\Delta_{V,C,C}| + |\Delta_{V,V,C}| \leq |\overleftarrow{A}^{vc}(T)|$, where $|\overleftarrow{A}^{vc}(T)| = |\overleftarrow{A}^{vc}(T)|$,
- 315 (iii) $|\Delta_{3V}| \leq 2n$,
- 316 (iv) $|\Delta_{3C}| \leq 1$.

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

318 **Proof.** Let Δ be a triangle packing of T . Recall that we have: $|\Delta| = |\Delta_{V,V,V}| + |\Delta_{V,V,C}| +$
 319 $|\Delta_{V,C,C}| + |\Delta_{C,C,C}| + |\Delta_{2V,C}| + |\Delta_{V,2C}| + |\Delta_{3V}| + |\Delta_{3C}|$. First, inequality (i) comes from
 320 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
 321 of these sets consumes one backward arc from T_c to T_v . We have $|\Delta_{3V}| \leq 2n$ since we have
 322 at most 2 disjoint triangles in each variable gadget. Finally we also have $|\Delta_{3C}| \leq 1$ since the
 323 dummy triangle is the only triangle lying in a clause gadget. ◀

324 These two lemmas allow us to prove the following.

325 ► **Lemma 3.** *F is satisfiable if and only if there exists a triangle packing Δ of size $6n(n-$*
 326 *$1) + 3m(m+1)/2 + 2n + |\overleftarrow{A}^{vc}(T)| + 1$ in the tournament T .*

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

329 We construct a triangle packing Δ of T with the desired number of triangles. First, we
 330 pick all the disjoint triangles of Δ^v and Δ^c . By Lemma 2, if we also add the dummy triangle
 331 $(c_{m+1}^1, c_{m+1}^2, c_{m+1}^3)$ we have $6n(n-1) + 3m(m+1)/2 + 1$ triangles in Δ until now.

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

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

341 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
 342 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
 343 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
 344 $d_{V_i \setminus \Delta_i^+}^+(\bar{x}_i) = 4$. Three of these four available arcs are used in the triangles which consume
 345 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
 346 other literals of c_j (which do not necessarily satisfy c_j). Denote by a_1 and a_2 the vertices of
 347 T_v connected to c_j^3 corresponding to the literals ℓ_1 and ℓ_2 , respectively. Then we add the
 348 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
 349 to T_v , and there are no triangles which use two arcs of $\overleftarrow{A}^{vc}(T)$. Then in the packing Δ there
 350 are in total $6n(n-1) + 3m(m+1)/2 + 2n + |\overleftarrow{A}^{vc}(T)| + 1$ triangles.

351 Conversely let Δ be a triangle packing of T with $|\Delta| = 6n(n-1) + 3m(m+1)/2 + 2n +$
 352 $|\overleftarrow{A}^{vc}(T)| + 1$. In the same way as we already did before, we partition Δ into the different subsets
 353 we defined before. We have $|\Delta| = |\Delta_{V,V,V}| + |\Delta_{V,V,C}| + |\Delta_{V,C,C}| + |\Delta_{C,C,C}| + |\Delta_{2V,C}| + |\Delta_{V,2C}|$
 354 $+ |\Delta_{3V}| + |\Delta_{3C}|$. By Lemma 2 all the upper bounds described above are tight, that is:

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

359 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
 360 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
 361 $\overleftarrow{A}^{vc}(T)$, it implies that $|\Delta_{2V,C}| + |\Delta_{V,2C}| \leq |\overleftarrow{A}^{vc}(T)| - x$. Moreover, if $x \neq 0$, then we have
 362 $|\Delta_{V,V,V}| < |\Delta^v|$ or $|\Delta_{C,C,C}| < |\Delta^c|$ because each triangle in $\Delta_{V,V,C}$ (resp. $\Delta_{V,C,C}$) will use one
 363 arc between two distinct variable gadgets (resp. clause gadgets) and according to Lemma 1, Δ^v
 364 (resp. Δ^c) uses all the arcs between distinct variable gadgets (resp. clause gadgets). Finally,
 365 we always have $|\Delta_{3V}| \leq 2n$ and $|\Delta_{3C}| \leq 1$ by construction. Therefore, if $x \neq 0$, we have $|\Delta| <$
 366 $|\Delta^v| + |\Delta^c| + x + (|\overleftarrow{A}^{vc}(T)| - x) + 2n + 1$ that is $|\Delta| < 6n(n-1) + 3m(m+1)/2 + 2n + |\overleftarrow{A}^{vc}(T)| + 1,$
 367 which is impossible. So we must have $x = 0$, which implies $\Delta_{V,V,C} = \Delta_{V,C,C} = \emptyset$.

368 Since $|\Delta_{3V}| = 2n$ and we have at most two arc-disjoint triangles in each variable gadget V_i ,
 369 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
 370 of $\Delta[V_i]$. Let us consider the following assignment a : for any variable v_i , if $\Delta_i = \Delta_i^\perp$, then
 371 $a(v_i) = false$ and $a(v_i) = true$ otherwise. Let us see that the assignment a satisfies the
 372 formula F . We have just proved that the backward arcs from T_c to T_v are all used in $\Delta_{2V,C}$
 373 and $\Delta_{V,2C}$. As $|\Delta_{3C}| = 1$ the dummy triangle C_{m+1} belongs to Δ . So every dummy arc
 374 $c_{m+1}^u \bar{x}_i$ is contained in a triangle of Δ which uses an arc of V_i . Therefore in each V_i we have
 375 $d_{V_i \setminus \Delta_i}^+(\bar{x}_i) \geq 3$. Moreover, for each clause of size q with $q \in \{2, 3\}$, there are q triangles which
 376 use the backward arcs coming from the clause to variable gadgets. Let C_j be a clause gadget
 377 of size 3 (we can do the same reasoning if C_j has size 2). By construction the 3 triangles
 378 cannot all lie in $\Delta_{V,2C}$. Thus, there is at least one of these triangles which is in $\Delta_{2V,C}$. Let t
 379 be one of them, V_i be the variable gadget where t has two out of its three vertices and \tilde{x} be
 380 the vertex of V_i which is also the head of the backward arc from C_j to V_i . By construction,
 381 \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
 382 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
 383 cannot have $\Delta_i = \Delta_i^\perp$ so the assignment sets the positive literal ℓ to *true*, which satisfies c_j .
 384 Otherwise, ℓ is negative so $\tilde{x} = \bar{x}_i$. Since \bar{x}_i has to use three out-going arcs to consume the
 385 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 $\Delta_i = \Delta_i^\perp$
 386 by Figure 3. Therefore, c_j is satisfied in that case too. Thus, the assignment a satisfies the
 387 whole formula F . ◀

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

389 ► **Theorem 4.** MAXATT is NP-hard.

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

393 ► **Lemma 5.** Given a 3-SAT(3) instance F , and T the tournament constructed from F
 394 with the reduction f , we have a triangle packing Δ of T of size $6n(n-1) + 3m(m+1)/2 +$
 395 $2n + |\overleftarrow{A}^{vc}(T)| + 1$ if and only if there is a cycle packing O of the same size.

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

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

403 As we did in the previous proof, we begin by finding upper bounds on each of these sets. First,
 404 recall that the FAS of each V_i is 2. Thus, we have $|O_V| \leq 2n$. By construction, we also have
 405 $|O_C| \leq 1$. Secondly, notice that a cycle of O_{V^*} cannot belong to exactly two distinct variable
 406 gadgets since the arcs between them are all in the same direction. Thus, the cycles of O_{V^*}
 407 have at least three vertices which implies $|O_{V^*}| \leq 6n(n-1)$. We obtain $|O_{C^*}| \leq 3m(m+1)/2$
 408 using the same reasoning on O_{C^*} . Finally, we have $|O_{V^*, C^*}| \leq |\overleftarrow{A}^{vc}(T)|$ since each cycle must
 409 have at least one backward arc.

410 Putting these upper bounds together, we obtain that $|O| \leq 6n(n-1) + 3m(m+1)/2 +$
 411 $2n + |\overleftarrow{A}^{vc}(T)| + 1$ which implies that the bounds are tight. In particular, cycles of O_{V^*} (resp.
 412 O_{C^*}) use exactly three arcs that are between distinct variable gadgets (resp. clause gadgets)
 413 and all these arcs are used. So we can construct a new cycle packing O' where we replace
 414 the cycles of O_{V^*} and O_{C^*} by the triangle packings Δ^v and Δ^c defined in Lemma 1. The
 415 new solution uses a subset of arcs of O and has the same size.

416 The cycles of O_{V^*, C^*} use exactly one backward arc of $\overleftarrow{A}^{vc}(T)$ due to the tight upper
 417 bound $|\overleftarrow{A}^{vc}(T)|$. Moreover, by the previous reasoning, two vertices of a cycle of O_{V^*, C^*}
 418 cannot belong to two different variable gadgets (resp. clause gadgets). Let C_j be a clause
 419 gadget which has three literals (if it has only two literals, the reasoning is analogous). Let
 420 $\tilde{x}_{i_k} \in V_{i_k}$ be the head of a backward arc from c_j^3 where $k \in [3]$. By the previous arguments
 421 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
 422 next vertex in o_k , say y , belongs to V_{i_k} since C_j has only two other vertices in addition to
 423 c_j^3 . Without loss of generality, we may assume that \tilde{x}_{i_3} is that vertex. Then, we can replace
 424 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 $c_j^1 c_j^3$ and $c_j^2 c_j^3$ cannot have
 425 already been used because C_j is acyclic and we previously consumed all the arcs between
 426 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
 427 yc_j^3 is available since it could have been used only in the cycle o_3 .

428 We now prove that given a V_i , we can restructure every cycle of $O_V[V_i]$ into triangles.
 429 Recall that $O_V[V_i]$ have exactly 2 cycles, and notice that by construction one cannot have
 430 two cycles each having a size greater than 3. First, if the two cycles are triangles, we are
 431 done. Then $O_V[V_i]$ contains a triangle, say δ , and a cycle, say o , of size greater than 3. If
 432 o contains the backward arc $s_i r_i$, then by construction $o = (r_i, \bar{x}_i, x_i^1, s_i)$. In that case, we
 433 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

434 $r_i x_i^1$ is not contained in O since the only arcs inside V_i we may have imposed until now are
 435 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
 436 $o = (x_i^1, s_i, x_i^2, t_i)$ 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)
 437 whose all the arcs are available.

438 As O_C is ⁶⁰²already 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. ◀

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

441 ▶ **Theorem 6.** MAXACT is NP-hard.

442 Let us now define two special cases TIGHT-ATT (resp. TIGHT-ACT) where, given a
 443 tournament T and a linear ordering σ with k backward arcs (where $k = \text{minfas}(T)$), the
 444 goal is to decide if there is a triangle (resp. cycle) packing of size k . We call these special
 445 cases the “tight” versions of the classical packing problems because as the input admits an
 446 FAS of size k , any triangle (or cycle) packing has size at most k . We now prove that we
 447 can construct in polynomial time an ordering of T , the tournament of the reduction, with k
 448 backward arcs (where k is the threshold value defined in Lemma 3).

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

452 **Proof.** Let us define a linear representation $(\sigma(T), \overleftarrow{A}(T))$ such that $|\overleftarrow{A}(T)| = k$. Remember
 453 that since $n \equiv 3 \pmod{6}$, the edges of the n -clique K_n can be packed into a packing O of
 454 $n(n-1)/6$ (undirected) triangles. Let us first prove that there exists an orientation T_{K_n} of K_n
 455 and a linear ordering σ of T_{K_n} with $|O|$ backward arcs. Let $\sigma = 1 \dots n$. For each undirected
 456 triangle ijk in O where $i < j < k$, we set $ki \in \overleftarrow{A}(T_{K_n})$ (implying that ij and jk are forward
 457 arcs). As all edges are used in O this defines an orientation for all edges. Thus, there is
 458 only $|O|$ backward arcs in σ . Thus, when using the previous orientations T_{K_n} to construct
 459 the variable tournament T_v of the reduction (remember that we blow up each vertex u_i into
 460 6 vertices V_i), we get an ordering with $36n(n-1)/6 = 6n(n-1)$ backward arcs between
 461 two different V_i (more formally, $|\{a \in \overleftarrow{A}(T_v) : \exists i_1 \neq i_2, h(a) \in V_{i_1}, t(a) \in V_{i_2}\}| = 6n(n-1)$).
 462 Following the same construction for the clause tournament T_c we get an ordering with
 463 $3m(m+1)/2$ backward arcs between two distinct C_j . Now, as there are two backward arcs
 464 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
 465 number of backward arcs is k . ◀

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

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

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

- 472 ■ $A_{V_i} = \{a \in A : v(a) \subseteq V_i\}$
- 473 ■ $A_{C_j} = \{a \in A : v(a) \subseteq C_j\}$
- 474 ■ $A_{V_i C_j} = \{a \in A : |v(a) \cap V_i| = |v(a) \cap C_j| = 1\}$
- 475 ■ $A_{V_i V_{i'}} = \{a \in A : |v(a) \cap V_i| = |v(a) \cap V_{i'}| = 1\}$ where $i \neq i'$
- 476 ■ $A_{C_j C_{j'}} = \{a \in A : |v(a) \cap C_j| = |v(a) \cap C_{j'}| = 1\}$ where $j \neq j'$

477 For any $i, i' \in [n]$, $j, j' \in [m+1]$ and $X \in \{V_i, C_j, V_i C_j, V_i V_{i'}, C_j C_{j'}\}$, we also define the
 478 corresponding sets F_X in F , where for example $F_{V_i} = F \cap A_{V_i}$. In addition, for any $j \in [m+1]$
 479 we define $F_{*C_j} = \bigcup_{i \in [n]} F_{V_i C_j}$. Let T'_v be the directed graph (T'_v is not a tournament) obtained
 480 by starting from T_v and only keeping arcs in $A_{V_i V_{i'}}$ for any $i, i' \in [n]$ with $i \neq i'$. As F is FAS
 481 of T , $F_{VV} = \bigcup_{i, i' \in [n], i \neq i'}^{693} F_{V_i V_{i'}}$ must be an FAS of T'_v . As according to Lemma 1 there is a
 482 cycle packing of size $6n(n-1)$ in T'_v , we get $|F_{VV}| \geq 6n(n-1)$. The same arguments hold for
 483 the clause part, and thus with $F_{CC} = \bigcup_{j, j' \in [m+1], j \neq j'} F_{C_j C_{j'}}$, we get $|F_{CC}| \geq 3m(m+1)/2$.
 484 As C_{m+1} is a triangle, we also get $|F_{C_{m+1}}| \geq 1$.

485 For any $j \in [m]$, let $u_j \in \{2, 3\}$ be equal to the size of the clause j (we also have
 486 $u_j = |\{a \in \overleftarrow{A}(T): \exists i \in [n], h(a) \in V_i \text{ and } t(a) \in C_j\}|$). Let $L = \{j \in [m]: |F_{*C_j} \cup F_{C_j}| \geq u_j\}$
 487 be informally the set of clauses where F spends a large (in fact larger than the u_j required)
 488 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
 489 consider the case where $u_j = 3$. Suppose by contradiction that $F_{C_j} = \{a\}$ (arguments will
 490 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
 491 $|F_{*C_j}| \leq 1$, there exists $i \in [n]$ and two arcs a_1, a_2 not in F such that $t(a_1) = c_j^3$, $h(a_1) \in V_i$,
 492 $t(a_2) = h(a_1)$, 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
 493 contradiction. As the same kind of arguments holds for the case where $u_j = 2$, we get that
 494 for any $j \in S$, $|F_{C_j}| \geq u_j - 1$ (implying also $|F_{*C_j}| = 0$).

495 Let us now prove that $|S| \leq 1$. Suppose by contradiction that $|S| \geq 2$. Let j_1 and j_2
 496 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
 497 $h(a_l) \in V_{i_l}$. Notice that we may have $i_1 = i_2$, but we always have $h(a_1) \neq h(a_2)$. Moreover,
 498 as a_i is the unique backward arc of T with $t(a) \in \bigcup_{j \in [m]} C_j$, we get that $a_3 = h(a_1)t(a_2)$
 499 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
 500 $l \in [4]$. 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.

501 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 $S = \emptyset$
 502 then $L' = \emptyset$, and otherwise $|L'| = u_{j_0}$, where $S = \{j_0\}$. Let $S' = [n] \setminus L'$. For any $i \in [n]$,
 503 let $\overleftarrow{A}_{V_i C_{m+1}} = \overleftarrow{A}(T) \cap A_{V_i C_{m+1}}$. Recall that $\overleftarrow{A}_{V_i C_{m+1}} = c_{m+1}^u \bar{x}_i$ for $u \in [3]$ where $\bar{x}_i \in V_i$.
 504 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
 505 $|A_{\bar{x}_i V_i}| = 4$, $|A_{x_i^1 V_i}| = 2$ and $|A_{x_i^2 V_i}| = 1$.

506 Let us prove that for any $i \in S'$, $|F_{V_i} \cup F_{V_i C_{m+1}}| \geq 5$. If $A_{\bar{x}_i V_i} \subseteq F$, then as F_{V_i} must be
 507 an FAS of V_i and $A_{\bar{x}_i V_i}$ is not an FAS of V_i , there exists at least another arc in F_{V_i} and we
 508 get $|F_{V_i}| \geq 5$. Otherwise, $\overleftarrow{A}_{V_i C_{m+1}} \subseteq F$ (if it is not the case, there is a cycle $c_{m+1}^u \bar{x}_i v$ where
 509 $v \in V_i$ is a out-neighbour of \bar{x}_i). Then, as $\text{minfas}(V_i) \geq 2$, $|F_{V_i} \cup F_{V_i C_{m+1}}| \geq 5$.

510 Let us finally prove that for any $i \in L'$, $|F_{V_i} \cup F_{V_i C_{m+1}}| \geq 6$. As $i \in L'$, there is an
 511 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
 512 $x \in \{\bar{x}_i, x_i^1, x_i^2\}$. As $|F_{*C_{j_0}}| = 0$ we get that $A_{xV_i} \subseteq F_{V_i}$ (otherwise there would be a cycle
 513 with one vertex in C_{j_0} , x , and an out-neighbour of x in V_i).

514 **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
 515 $|F_{V_i}| \geq 6$.

516 **Case 2:** $x = x_i^1$. If $A_{\bar{x}_i V_i} \subseteq F$ then $|F_{V_i} \cup F_{V_i C_{m+1}}| \geq 6$. Otherwise, as before we get
 517 $\overleftarrow{A}_{V_i C_{m+1}} \subseteq F$, and as $A_{x_i^1 V_i}$ is not an FAS of V_i , F need another arc in V_i , implying
 518 $|F_{V_i} \cup F_{V_i C_{m+1}}| \geq 6$.

519 **Case 3:** $x = x_i^2$. If $A_{\bar{x}_i V_i} \subseteq F$ then as $A_{x_i^2 V_i} \cup A_{\bar{x}_i V_i}$ is not an FAS of V_i , F need another arc
 520 in V_i , implying $|F_{V_i}| \geq 6$. Otherwise, as before we get $\overleftarrow{A}_{V_i C_{m+1}} \subseteq F$, and as $A_{x_i^2 V_i}$ is not an
 521 FAS of V_i , F need two other arcs in V_i , implying $|F_{V_i} \cup F_{V_i C_{m+1}}| \geq 6$.

522 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 $\mathcal{O}^*(2^{o(\sqrt{k})})$ 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 $\mathcal{O}^*(2^{\mathcal{O}(tw \log tw)})$ time (see, e.g., [21]). Thus, the existence of an $\mathcal{O}^*(2^{\mathcal{O}(k \log^2 k)})$ 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 $\mathcal{O}^*(2^{\mathcal{O}(k \log k)})$ 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 $2k + 1$.

559 **Proof.** The reverse direction of the claim holds trivially. Let us now prove the forward
 560 direction. Let \mathcal{C} be a set of r arc-disjoint cycles in T that minimizes $\sum_{C \in \mathcal{C}} |C|$. If every
 561 cycle in \mathcal{C} is a triangle, then the claim trivially holds. Otherwise, let C be a longest cycle in
 562 \mathcal{C} and let ℓ denote its length. Let v_i, v_j be a pair of non-consecutive vertices in C . Then,
 563 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 $A(C)$
 564 forms a cycle C' of length less than ℓ with $A(C') \setminus \{e\} \subset A(C)$. By our choice of \mathcal{C} , this
 565 implies that e is an arc in some other cycle $\hat{C} \in \mathcal{C}$. This property is true for the arc between
 566 any pair of non-consecutive vertices in C . Therefore, we have $\binom{\ell}{2} - \ell \leq \ell(k-1)$ leading to
 567 $\ell \leq 2k+1$. ◀

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

573 We will use the following lemma known from [17] in the process¹. For a digraph D , let
 574 $\Lambda(D)$ denote the number of non-adjacent pairs of vertices in D . That is, $\Lambda(D)$ is the number
 575 of pairs u, v of vertices of D such that neither $uv \in A(D)$ nor $vu \in A(D)$. Recall that for a
 576 digraph D , $\text{minfas}(D)$ denotes the size of a minimum FAS of D .

577 ▶ **Lemma 12.** [17] *Let D be a triangle-free digraph in which for every pair u, v of distinct*
 578 *vertices, at most one of uv or vu is in $A(D)$. Then, we can compute an FAS of size at most*
 579 *$\Lambda(D)$ in polynomial time.*

580 This leads to the following main result of this section.

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

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

593 4.2 A Linear Vertex Kernel

594 Next, we show that ACT has a linear vertex kernel. This kernel is inspired by the linear
 595 kernelization described in [11] for FAST and uses Theorem 13. Let T be a tournament on n
 596 vertices. First, we apply the following reduction rule.

597 ▶ **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.

598 This rule is clearly safe as our goal is to find k cycles and v cannot be in any of them.
 599 To describe our next rule, we need to state a lemma known from [11]. An *interval* is a
 600 consecutive set of vertices in a linear representation $(\sigma(T), \overleftarrow{A}(T))$ of a tournament T .

601 ► **Lemma 14**² ([11]). *Let $T = (\sigma(T), \overleftarrow{A}(T))$ be a tournament on which Reduction Rule 4.1*
 602 *is not applicable. If $|V(T)| \geq 2|\overleftarrow{A}(T)| + 1$, then there exists a partition \mathcal{J} of $V(T)$ into intervals*
 603 *(that can be computed in polynomial time) such that there are $|\overleftarrow{A}(T) \cap E| > 0$ arc-disjoint*
 604 *cycles using only arcs in E where E denotes the set of arcs in T with endpoints in different*
 605 *intervals.*

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

607 ► **Reduction Rule 4.2.** *Let $T = (\sigma(T), \overleftarrow{A}(T))$ be a tournament on which Reduction Rule*
 608 *4.1 is not applicable. Let \mathcal{J} be a partition of $V(T)$ into intervals satisfying the properties*
 609 *specified in Lemma 14. Reverse all arcs in $\overleftarrow{A}(T) \cap E$ and decrease k by $|\overleftarrow{A}(T) \cap E|$ where E*
 610 *denotes the set of arcs in T with endpoints in different intervals.*

611 ► **Lemma 15.** *Reduction Rule 4.2 is safe.*

612 **Proof.** Let T' be the tournament obtained from T by reversing all arcs in $\overleftarrow{A}(T) \cap E$. Suppose
 613 T' has $k - |\overleftarrow{A}(T) \cap E|$ arc-disjoint cycles. Then, it is guaranteed that each such cycle is
 614 completely contained in an interval. This is due to the fact that T' has no backward arc
 615 with endpoints in different intervals. Indeed, if a cycle in T' uses a forward (backward) arc
 616 with endpoints in different intervals, then it also uses a back (forward) arc with endpoints in
 617 different intervals. It follows that for each arc $uv \in E$, neither uv nor vu is used in these
 618 $k - |\overleftarrow{A}(T) \cap E|$ cycles. Hence, these $k - |\overleftarrow{A}(T) \cap E|$ cycles in T' are also cycles in T . Then,
 619 we can add a set of $|\overleftarrow{A}(T) \cap E|$ cycles obtained from the second property of Lemma 14 to
 620 these $k - |\overleftarrow{A}(T) \cap E|$ cycles to get k cycles in T . Conversely, consider a set of k cycles in
 621 T . As argued earlier, we know that the number of cycles that have an arc that is in E is at
 622 most $|\overleftarrow{A}(T) \cap E|$. The remaining cycles (at least $k - |\overleftarrow{A}(T) \cap E|$ of them) do not contain any
 623 arc that is in E , in particular, they do not contain any arc from $\overleftarrow{A}(T) \cap E$. Therefore, these
 624 cycles are also cycles in T' . ◀

625 Thus, we have the following result.

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

627 **Proof.** Let (T, k) denote the instance obtained from the input instance by applying Reduction
 628 Rule 4.1 exhaustively. From Lemma 13, we know that either T has k arc-disjoint triangles or
 629 has an FAS of size at most $5(k - 1)$ that can be obtained in polynomial time. In the first
 630 case, we return a trivial yes-instance of constant size as the kernel. In the second case, let F
 631 be the FAS of size at most $5(k - 1)$ of T . Let $(\sigma(T), \overleftarrow{A}(T))$ be the linear representation of T
 632 where $\sigma(T)$ is a topological ordering of the vertices of the directed acyclic graph $T - F$. As
 633 $V(T - F) = V(T)$, $|\overleftarrow{A}(T)| \leq 5(k - 1)$. If $|V(T)| \geq 10k - 9$, then from Lemma 14, there is a
 634 partition of $V(T)$ into intervals with the specified properties. Therefore, Reduction Rule 4.2
 635 is applicable (and the parameter drops by at least 1). When we obtain an instance where
 636 neither of the Reduction Rules 4.1 and 4.2 is applicable, it follows that the tournament in
 637 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].

638 **4.3 An FPT Algorithm**

639 Finally, we show that ACT can be solved in $\mathcal{O}^*(2^{\mathcal{O}(k \log k)})$ time. The idea is to reduce
 640 the problem to the following ARC-DISJOINT PATHS problem in directed acyclic graphs:
 641 given a digraph D on n vertices and k ordered pairs $(s_1, t_1), \dots, (s_k, t_k)$ of vertices of D , do
 642 there exist arc-disjoint paths P_1, \dots, P_k in D such that P_i is a path from s_i to t_i for each
 643 $i \in [k]$? On directed acyclic graphs, ARC-DISJOINT PATHS is known to be NP-complete
 644 [27], W[1]-hard [52] with respect to k as parameter and solvable in $n^{\mathcal{O}(k)}$ time [32]. Despite
 645 its fixed-parameter intractability, we will show that we can use the $n^{\mathcal{O}(k)}$ algorithm and
 646 Theorems 13 and 16 to describe an FPT algorithm for ACT.

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

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

654 Otherwise, let \widehat{F} be the FAS of size at most $5(\widehat{k} - 1)$ returned by Theorem 13. Let
 655 D denote the (acyclic) digraph obtained from \widehat{T} by deleting \widehat{F} . Observe that D has $\mathcal{O}(k)$
 656 vertices. Suppose \widehat{T} has a set $\mathcal{C} = \{C_1, \dots, C_{\widehat{k}}\}$ of \widehat{k} arc-disjoint cycles. For each $C \in \mathcal{C}$, we
 657 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
 658 $F = \widehat{F} \cap A(\mathcal{C})$. Then, for each cycle $C_i \in \mathcal{C}$, we can guess the arcs F_i from F that it contains
 659 and also the order π_i in which they appear. This information is captured as a partition \mathcal{F} of
 660 F into \widehat{k} sets, F_1 to $F_{\widehat{k}}$ and the set $\{\pi_1, \dots, \pi_{\widehat{k}}\}$ of permutations where π_i is a permutation
 661 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
 662 pair $(u, v), (x, y)$ of consecutive arcs of F_i with arcs from $A(D)$. That is, there is a path
 663 from $h(\pi_i^{-1}(j))$ and $t(\pi_i^{-1}((j+1) \bmod |F_i|))$ with arcs from D for each $j \in [|F_i|]$. The total
 664 number of such paths in these \widehat{k} cycles is $\mathcal{O}(|F|)$ and the arcs of these paths are contained in
 665 D which is a (simple) directed acyclic graph.

666 The number of choices for F is $2^{|\widehat{F}|}$ and the number of choices for a partition $\mathcal{F} =$
 667 $\{F_1, \dots, F_{\widehat{k}}\}$ of F and a set $X = \{\pi_1, \dots, \pi_{\widehat{k}}\}$ of permutations is $2^{\mathcal{O}(|\widehat{F}| \log |\widehat{F}|)}$. Once such a
 668 choice is made, the problem of finding \widehat{k} arc-disjoint cycles in \widehat{T} reduces to the problem of
 669 finding \widehat{k} arc-disjoint cycles $\mathcal{C} = \{C_1, \dots, C_{\widehat{k}}\}$ in \widehat{T} such that for each $1 \leq i \leq \widehat{k}$ and for each
 670 $1 \leq j \leq |F_i|$, C_i has a path P_{ij} between $h(\pi_i^{-1}(j))$ and $t(\pi_i^{-1}((j+1) \bmod |F_i|))$ with arcs
 671 from $D = \widehat{T} - \widehat{F}$. This problem is essentially finding $r = \mathcal{O}(|\widehat{F}|)$ arc-disjoint paths in D and
 672 can be solved in $|V(D)|^{\mathcal{O}(r)}$ time using the algorithm in [32]. Therefore, the overall running
 673 time of the algorithm is $\mathcal{O}^*(2^{\mathcal{O}(k \log k)})$ as $|V(D)| = \mathcal{O}(k)$ and $r = \mathcal{O}(k)$. ◀

674 **5 Parameterized Complexity of ATT**

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

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

679 **Proof.** Consider an instance $\mathcal{I} = (T, k)$ of ATT. Let n denote $|V(T)|$ and m denote $|A(T)|$.
 680 Let \mathcal{F} denote the family of colouring functions $c : A(T) \rightarrow [3k]$ of size $2^{\mathcal{O}(k)} \log^2 m$ that can

681 be computed in $\mathcal{O}^*(2^{\mathcal{O}(k)})$ time using $3k$ -perfect family of hash functions [51]. For each
 682 colouring function c in \mathcal{F} , we colour $A(T)$ according to c and find a triangle packing of size
 683 k whose arcs use different colours. We use a standard dynamic programming routine to
 684 finding such a triangle packing. Clearly, if \mathcal{I} is an yes-instance and \mathcal{C} is a set of k arc-disjoint
 685 triangles in T , there is a colouring function in \mathcal{F} that colours the $3k$ arcs in these triangles
 686 with distinct colours and our algorithm will find the required triangle packing. Given a
 687 colouring $c \in \mathcal{F}$, we first compute for every set of 3 colours $\{a, b, c\}$ whether the arcs coloured
 688 with a, b or c induce a triangle using 3 different colours or not. Then, for every set S of
 689 $3(p+1)$ colours with $p \in [k-1]$, we recursively test if the arcs coloured with the colours in
 690 S induce $p+1$ arc-disjoint triangles whose arcs use all the colours of S . This is achieved by
 691 iterating over every subset $\{a, b, c\}$ of S and checking if there is a triangle using colours a, b
 692 and c and a collection of p arc-disjoint triangles whose arcs use all the colours of $S \setminus \{a, b, c\}$.
 693 For a given S , we can find this collection of triangles in $\mathcal{O}(p^3) = \mathcal{O}(k^3)$ time. Therefore, the
 694 overall running time of the algorithm is $\mathcal{O}^*(2^{\mathcal{O}(k)})$. ◀

695 Next, we show that ATTT has a linear vertex kernel.

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

697 **Proof.** Let \mathcal{X} be a maximal collection of arc-disjoint triangles of a tournament T obtained
 698 greedily. Let $V_{\mathcal{X}}$ denote the vertices of the triangles in \mathcal{X} and $A_{\mathcal{X}}$ denote the arcs of $V_{\mathcal{X}}$.
 699 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
 700 yes-instance of ATTT. Otherwise, $|\mathcal{X}| < k$ and $|V_{\mathcal{X}}| < 3k$. Moreover, notice that $T[U]$ is acyclic
 701 and T does not contain a triangle with one vertex in $V_{\mathcal{X}}$ and two in vertices in U (otherwise
 702 \mathcal{X} would not be maximal).

703 Let B be the (undirected) bipartite graph defined by $V(B) = A_{\mathcal{X}} \cup U$ and $E(B) =$
 704 $\{au : a \in A_{\mathcal{X}}, u \in U \text{ such that } (t(a), h(a), u) \text{ forms a triangle in } T\}$. Let M be a maximum
 705 matching of B and A' (resp. U') denote the vertices of $A_{\mathcal{X}}$ (resp. U) covered by M . Define
 706 $\bar{A}' = A_{\mathcal{X}} \setminus A'$ and $\bar{U}' = U \setminus U'$.

707 We now prove that $(V_{\mathcal{X}} \cup U', k)$ is a linear kernel of (T, k) . Let \mathcal{C} be a maximum sized
 708 triangle packing that minimizes the number of vertices of \bar{U}' belonging to a triangle of \mathcal{C} . By
 709 previous remarks, we can partition \mathcal{C} into $C_{\mathcal{X}} \cup F$ where $C_{\mathcal{X}}$ are the triangles of \mathcal{C} included
 710 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
 711 is clear that F corresponds to a union of vertex-disjoint stars of B with centres in U . Denote
 712 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
 713 immediately a kernel. Suppose there exists a vertex x_0 such that $x_0 \in U[F] \cap \bar{U}'$.

714 We will build a tree rooted in x_0 with edges alternating between F and M . For this let
 715 $H_0 = \{x_0\}$ and construct recursively the sets H_{i+1} such that

$$716 \quad H_{i+1} = \begin{cases} N_F(H_i) & \text{if } i \text{ is even,} \\ N_M(H_i) & \text{if } i \text{ is odd,} \end{cases}$$

717 where, given a subset $S \subseteq U$, $N_F(S) = \{a \in A_{\mathcal{X}} : \exists s \in S \text{ s.t. } (t(a), h(a), s) \in F \text{ and } as \notin M\}$
 718 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 $H_i \subseteq U$
 719 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
 720 union of disjoint stars and M a matching in B . Moreover, for $i \geq 1$ we call T_i the set of edges
 721 between H_i and H_{i-1} . Now we define the tree T such that $V(T) = \bigcup_i H_i$ and $E(T) = \bigcup_i T_i$.
 722 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
 723 is odd), it is clear that T is a tree.

724 For i being odd, every vertex of H_i is incident to an edge of M otherwise B would contain
 725 an augmenting path for M , a contradiction. So every leaf of T is in U and incident to an

726 edge of M in T and T contains as many edges of M than edges of F . Now for every arc
 727 $a \in A_{\mathcal{X}} \cap V(T)$ we replace the triangle of \mathcal{C} containing a and corresponding to an edge of F
 728 by the triangle $(t(a), h(a), u)$ where $au \in M$ (and au is an edge of T). This operation leads
 729 to another collection of arc-disjoint triangles with the same size as \mathcal{C} but containing a strictly
 730 smaller number of vertices in $\overline{U'}$, yielding a contradiction.

731 Finally $V_{\mathcal{X}} \cup U'$ can be computed in polynomial time and we have $|V_{\mathcal{X}} \cup U'| \leq |V_{\mathcal{X}}| + |M| \leq$
 732 $2|V_{\mathcal{X}}| \leq 6k$, which proves that the kernel has $\mathcal{O}(k)$ vertices. \blacktriangleleft

733 **6 MAXACT and MAXATT in Sparse Tournaments**

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

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

754 Now order the arcs e_1, \dots, e_b of $\overleftarrow{A}(T)$ such that for any $i \in [b-1]$, $h(e_i) <_{\sigma} h(e_{i+1})$.
 755 Moreover, let G' be the digraph with vertex set $V' = \{e_i : i \in [b]\}$ and arc set A' defined
 756 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
 757 problem such that, given a digraph $G' = (V', A')$, the objective is to find a maximum sized
 758 subset of A' such that the digraph induced by the arcs of the subset is a functional and
 759 digon-free digraph. Remind that a functional digraph is a digraph such that any of its
 760 vertices has out-degree at most 1.

761 Let X be a solution (not necessary optimal) of $\Pi'(G')$, and $e_i e_j$ an arc of X . We denote
 762 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 $\Pi(e_i e_j)$,
 763 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
 764 $s(e_j) = t(e_j)$ and $s(e_j) = h(e_j)$ otherwise. Informally, $\Pi(e_i e_j)$ corresponds to the triangle
 765 formed by the backward arc e_i and one vertex of e_j , that vertex being $s(e_j)$. In the same
 766 way, we define $\Pi(X) = \bigcup_{x \in X} \Pi(x)$.

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

769 **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
 770 a functional digraph in G' . Without loss of generality, we may assume that $i < k$, that is

771 $h(e_i) <_{\sigma} h(e_k)$. Moreover, we cannot have $t(e_i) = t(e_k)$ without contradicting that T is a
 772 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
 773 $\Pi(e_i e_j)$ 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
 774 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
 775 that X is a solution $\Pi'(G')$. So, if X is a solution of $\Pi'(G')$, then $\Pi(X)$ is an solution of
 776 $\Pi(T)$. Notice that the size of the solution does not change.

777 On the other hand, if X is a subset of the arcs of G' such that $\Pi(X)$ is a solution of
 778 $\Pi(T)$. We cannot have a vertex e_i of G' such that $d_X^+(e_i) > 1$, since it would imply that the
 779 backward arc e_i of T is covered by at least two triangles of $\Pi(X)$. So X induces a functional
 780 subdigraph of G' . As previously the digraph induced by X is also digon-free otherwise we
 781 would have two arc-disjoint triangles on only four vertices in $\Pi(X)$, which is impossible.
 782 Thus, X is a solution of $\Pi'(G')$, and the solution of the same size.

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

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

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

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

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

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

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

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

804 **Proof.** The case $|V(G')|=1$ is clear. So assume that G' is a digoned tree and let X be a set
 805 of arcs of G' corresponding to an optimal solution of $\Pi'(G')$. Then X is acyclic and then
 806 has size at most $|V(G')|-1$. Moreover, any in-branching of G' provides a solution of size
 807 $|V(G')|-1$. ◀

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

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

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

819 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
 820 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
 821 most one arc of X , the set X has size at most $n - k$. Conversely by growing in-branchings
 822 in G' from the union of the optimal solutions of $\Pi'(G'[S_i])$ for $i = 1, \dots, p$, by Claim 19.2
 823 and 19.4 we obtain a solution of $\Pi'(G')$ of size $n - k$ which is then optimal. Moreover, this
 824 solution can clearly be built in polynomial time. ◀

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

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

828 **Proof.** First if T has an optimal triangle packing of size $|\overleftarrow{A}(T)|$ then as $\overleftarrow{A}(T)$ is an FAS of T ,
 829 every optimal cycle packing of T has size $|\overleftarrow{A}(T)|$. Otherwise, we build from T the digraph G'
 830 as previously. By Lemma 20, G' has some terminal components S_1, \dots, S_k which are either
 831 a single vertex or induces a digoned tree and every optimal triangle packing of T has size
 832 $|\overleftarrow{A}(T)| - k$. Let see that no S_i can be a single vertex. Indeed if $S_i = \{e\}$ where e is a backward
 833 arc of T , it means that no backward of T begins or ends between $h(e)$ and $t(e)$ in $\sigma(T)$. As T
 834 is fully sparse, it means that $h(e)$ and $t(e)$ are consecutive in $\sigma(T)$ what we forbid previously.
 835 Now consider a component S_i which induces a digoned tree in G' . Let π_i be the order $\sigma(T)$
 836 restricted to the heads and tails of the arcs of T corresponding to the vertices of S_i . First
 837 notice that π_i is an interval of the order $\sigma(T)$. Indeed otherwise there exists two backward
 838 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
 839 before $t(a)$ in $\sigma(T)$. But in this case there is an arc in G' from a to b contradicting the fact
 840 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
 841 x_1 and x_2 are then forced to be the heads of backward arcs belonging to S_i . If x_3 is also
 842 the head of backward arc of S_i , then we obtain that the three corresponding backward arcs
 843 form a 3-cycle in G' contradicting the fact that S_i induces a digoned tree in G' . Repeating
 844 the same argument we show that l is even and that the backward arcs corresponding to the
 845 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
 846 S_i induces a 'digoned path' in G' . Now consider Δ an optimal cycle packing of T . Let X_1
 847 be the set of backward arcs of $\overleftarrow{A}(T)$ with head strictly before x_1 and tail strictly after x_l in
 848 $\sigma(T)$. And let Δ_1 be the cycles of Δ using at least one arc of X_1 . It is easy to check that
 849 $\Delta' = (\Delta \setminus \Delta_1) \cup \{(h(e), x_1, t(e)) : e \in X_1\}$ is also an optimal cycle packing of T . Now every
 850 cycle of Δ' which uses a backward arc of S_i only uses backward arcs of S_i (otherwise it must
 851 one arc of X_1 , which is not possible). Let Δ_i be the set of cycles of Δ using backward arcs
 852 of S_i . It is easy to see that $\{x_i x_{i+1} : i \text{ even and } i \in [l-2]\}$ is an FAS of $T[\{x_1, \dots, x_l\}]$ and
 853 has size $l/2 - 1 = |S_i| - 1$. So we have $|\Delta_i| \leq |S_i| - 1$.

854 Repeating this argument for $i = 1, \dots, k$ we obtain that $|\Delta| \leq |\overleftarrow{A}(T)| - k$. Thus by Lemma 20
 855 Δ has the same size than an optimal triangle packing of T . ◀

856 This leads to the following main result of this section.

857 ► **Theorem 22.** MAXATT and MAXACT restricted to sparse tournaments can be solved in
858 polynomial time.

859 **7** Concluding Remarks

860 In this work, we studied the classical and parameterized complexity of packing arc-disjoint
861 cycles and triangles in tournaments. We showed NP-hardness, fixed-parameter tractability and
862 linear kernelization results. We also showed that these problems are polynomial-time solvable
863 in sparse tournaments. To conclude, observe that very few problems on tournaments are
864 known to admit an $\mathcal{O}^*(2^{\sqrt{k}})$ -time algorithm when parameterized by the standard parameter
865 k [48] - FAST is one of them [4, 28]. To the best of our knowledge, outside bidimensionality
866 theory, there are no packing problems that are known to admit such subexponential algorithms.
867 In light of the $2^{o(\sqrt{k})}$ lower bound shown for ACT and ATT, it would be interesting to
868 explore if these problems admit $\mathcal{O}^*(2^{\mathcal{O}(\sqrt{k})})$ algorithms.

869 ——— References ———

- 870 **1** F. N. Abu-Khzam. An Improved Kernelization Algorithm for r-Set Packing. *Inf. Process.*
871 *Lett.*, 110(16):621–624, 2010.
- 872 **2** I. Akaria and R. Yuster. Packing Edge-Disjoint Triangles in Regular and Almost Regular
873 Tournaments. *Discrete Math.*, 338(2):217–228, 2015.
- 874 **3** N. Alon. Ranking Tournaments. *SIAM J. Discrete Math.*, 20(1):137–142, 2006.
- 875 **4** N. Alon, D. Lokshtanov, and S. Saurabh. Fast FAST. In *36th International Colloquium on*
876 *Automata, Languages, and Programming (ICALP) Proc., Part I*, pages 49–58, 2009.
- 877 **5** N. Alon, R. Yuster, and U. Zwick. Color-Coding. *J. ACM*, 42(4):844–856, 1995.
- 878 **6** J. Bang-Jensen and G. Gutin. Paths, Trees and Cycles in Tournaments. *Congressus*
879 *Numerantium*, 115:131–170, 1996.
- 880 **7** J. Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms and Applications*. Springer-
881 Verlag London, 2009.
- 882 **8** F. Barbero, C. Paul, and M. Pilipczuk. Strong Immersion is a Well-quasi-ordering for
883 Semi-complete Digraphs. *CoRR*, abs/1707.03563, 2017.
- 884 **9** S. Bessy, M. Bougeret, and J. Thiebaut. Triangle Packing in (Sparse) Tournaments: Ap-
885 proximation and Kernelization. In *25th Annual European Symp. on Algorithms (ESA 2017)*,
886 volume 87, pages 14:1–14:13, 2017.
- 887 **10** S. Bessy, M. Bougeret, and J. Thiebaut. (Arc-disjoint) Cycle Packing in Tournament:
888 Classical and Parameterized Complexity, 2018. [arXiv:1802.06669](https://arxiv.org/abs/1802.06669).
- 889 **11** S. Bessy, F. V. Fomin, S. Gaspers, C. Paul, A. Perez, S. Saurabh, and S. Thomassé. Kernels
890 for Feedback Arc Set in Tournaments. *J. Comput. Syst. Sci.*, 77(6):1071–1078, 2011.
- 891 **12** H. L. Bodlaender. On Disjoint Cycles. *Int. J. Found. Comput. S.*, 5(1):59–68, 1994.
- 892 **13** H. L. Bodlaender, S. Thomassé, and A. Yeo. Kernel Bounds for Disjoint Cycles and Disjoint
893 Paths. *Theor. Comput. Sci.*, 412(35):4570–4578, 2011.
- 894 **14** A. Caprara, A. Panconesi, and R. Rizzi. Packing Cycles in Undirected Graphs. *J. Al-*
895 *gorithms*, 48(1):239 – 256, 2003.
- 896 **15** P. Charbit, S. Thomassé, and A. Yeo. The Minimum Feedback Arc Set Problem is NP-Hard
897 for Tournaments. *Comb Probab Comput.*, 16(1):1–4, 2007.
- 898 **16** M. Chudnovsky and P. Seymour. A Well-quasi-order for Tournaments. *J. of Combinatorial*
899 *Theory, Series B*, 101(1):47–53, 2011.
- 900 **17** M. Chudnovsky, P. Seymour, and B. Sullivan. Cycles in Dense Digraphs. *Combinatorica*,
901 28(1):1–18, 2008.

- 902 **18** W. W. Cohen, R. E. Schapire, and Y. Singer. Learning to Order Things. In *Advances in*
903 *Neural Information Processing Systems Conference (NIPS)*, pages 451–457, 1997.
- 904 **19** V. Conitzer. Computing Slater Rankings Using Similarities Among Candidates. In *21st*
905 *National Conf. on Artificial Intelligence - Volume 1*, pages 613–619, 2006.
- 906 **20** M. Cygan⁶¹³. Improved Approximation for 3-Dimensional Matching via Bounded Pathwidth
907 Local Search. In *54th Annual IEEE Symp. on Foundations of Computer Science (FOCS)*,
908 pages 509–518, 2013.
- 909 **21** M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshantov, D. Marx, M. Pilipczuk, M. Pilipczuk,
910 and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.
- 911 **22** Jean-Charles de Borda. Mémoire sur les élections au scrutin. *Histoire de l'Académie Royale*
912 *des Sciences*, 1781.
- 913 **23** D. Dorninger. Hamiltonian Circuits Determining the Order of Chromosomes. *Discrete*
914 *Appl. Math.*, 50(2):159–168, 1994.
- 915 **24** R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Springer-
916 Verlag London, 2013.
- 917 **25** C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank Aggregation Methods for the
918 Web. In *Proc. of the 10th International World Wide Web Conference*, pages 613–622, 2001.
- 919 **26** P. Erdős and L. Pósa. On Independent Circuits Contained in a Graph. *Canadian J. Math.*,
920 17:347–352, 1965.
- 921 **27** S. Even, A. Itai, and A. Shamir. On the Complexity of Timetable and Multicommodity
922 Flow Problems. *SIAM J. Comput.*, 5(4):691–703, 1976.
- 923 **28** U. Feige. Faster FAST(Feedback Arc Set in Tournaments), 2009. [arXiv:0911.5094](https://arxiv.org/abs/0911.5094).
- 924 **29** J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006.
- 925 **30** F. V. Fomin, D. Lokshantov, V. Raman, and S. Saurabh. Fast Local Search Algorithm for
926 Weighted Feedback Arc Set in Tournaments. In *Proc. of the 24th AAAI Conf. on Artificial*
927 *Intelligence*, pages 65–70, 2010.
- 928 **31** F. V. Fomin, D. Lokshantov, S. Saurabh, and M. Zehavi. *Kernelization: Theory of Para-*
929 *meterized Preprocessing*. Cambridge University Press, 2019.
- 930 **32** S. Fortune, J. Hopcroft, and J. Wyllie. The Directed Subgraph Homeomorphism Problem.
931 *Theor. Comput. Sci.*, 10(2):111–121, 1980.
- 932 **33** R. B. Gardner. Optimal Packings and Coverings of the Complete Directed Graph with
933 3-Circuits and with Transitive Triples. In *Proc. of the 28th Southeastern International*
934 *Conference on Combinatorics, Graph Theory and Computing*, volume 127, pages 161–170,
935 1997.
- 936 **34** E. Hemaspaandra, H. Spakowski, and J. Vogel. The Complexity of Kemeny Elections.
937 *Theor. Comput. Sci.*, 349(3):382–391, 2005.
- 938 **35** R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential com-
939 plexity? *J. Comput. Syst. Sci.*, 63(4):512 – 530, 2001.
- 940 **36** M. Karpinski and W. Schudy. Faster Algorithms for Feedback Arc Set Tournament, Ke-
941 meny Rank Aggregation and Betweenness Tournament. In *21st International Symp. on*
942 *Algorithms and Computation (ISAAC)*, pages 3–14. Springer Berlin Heidelberg, 2010.
- 943 **37** T. P. Kirkman. On a Problem in Combinations. *Cambridge and Dublin Mathematical*
944 *Journal*, 2:191–204, 1847.
- 945 **38** R. Krithika, A. Sahu, S. Saurabh, and M. Zehavi. The Parameterized Complexity of
946 Packing Arc-Disjoint Cycles in Tournaments, 2018. [arXiv:1802.07090](https://arxiv.org/abs/1802.07090).
- 947 **39** M. Krivelevich, Z. Nutov, M. R. Salavatipour, J. V. Yuster, and R. Yuster. Approximation
948 Algorithms and Hardness Results for Cycle Packing Problems. *ACM Transactions on*
949 *Algorithms*, 3(4), 2007.

- 950 40 M. Krivelevich, Z. Nutov, and R. Yuster. Approximation Algorithms for Cycle Packing
951 Problems. In *Proc. of the 16th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA)*,
952 pages 556–561, 2005.
- 953 41 T. Le, D. Lokshtanov, S. Saurabh, S. Thomassé, and M. Zehavi. Subquadratic Kernels for
954 Implicit 3-Hitting Set and 3-Set Packing Problems. In *Proc. of the 29th Annual ACM-SIAM*
955 *Symp. on Discrete Algorithms (SODA)*, pages 331–342, 2018.
- 956 42 M. le Marquis de Condorcet. *Essai sur l'application de l'analyse à la probabilité des dé-*
957 *cisions rendues à la pluralité des voix*. Imprimerie Royale, Paris, 1785.
- 958 43 D. Lokshtanov, A. Mouawad, S. Saurabh, and M. Zehavi. Packing Cycles Faster Than
959 Erdős-Pósa. In *44th International Colloquium on Automata, Languages, and Programming*
960 *(ICALP)*, pages 71:1–71:15, 2017.
- 961 44 C. L. Lucchesi and D. H. Younger. A Minimax Theorem for Directed Graphs. *J. of the*
962 *London Mathematical Society*, 17(2):369–374, 1978.
- 963 45 M. Mahajan and V. Raman. Parameterizing Above Guaranteed Values: MaxSat and
964 MaxCut. *J. Algorithms*, 31(2):335–354, 1999.
- 965 46 J.W. Moon. *Topics on Tournaments*. Holt, Rinehart and Winston, New York, 1968.
- 966 47 C. H. Papadimitriou. *Computational Complexity*. John Wiley and Sons Ltd., 2003.
- 967 48 M. Pilipczuk. *Tournaments and Optimality: New Results in Parameterized Complexity*.
968 PhD thesis, The University of Bergen, 2013.
- 969 49 V. Raman and S. Saurabh. Parameterized Algorithms for Feedback Set problems and their
970 Duals in Tournaments. *Theor. Comput. Sci.*, 351(3):446–458, 2006.
- 971 50 N. Robertson and P.D. Seymour. Graph Minors. XX. Wagner's Conjecture. *J. of Combin-*
972 *atorial Theory, Series B*, 92(2):325–357, 2004.
- 973 51 J. P. Schmidt and A. Siegel. The Spatial Complexity of Oblivious k-Probe Hash Functions.
974 *SIAM J. Comput.*, 19(5):775–786, 1990.
- 975 52 A. Slivkins. Parameterized Tractability of Edge-Disjoint Paths on Directed Acyclic Graphs.
976 *SIAM J. Discrete Math.*, 24(1):146–157, 2010.
- 977 53 E. Speckenmeyer. On Feedback Problems in Digraphs. In *15th International Workshop on*
978 *Graph-Theoretic Concepts in Computer Science*, pages 218–231. Springer Berlin Heidelberg,
979 1990.
- 980 54 C. A. Tovey. A simplified NP-complete satisfiability problem. *Discrete Appl. Math.*, 8(1):85
981 – 89, 1984.
- 982 55 R. Yuster. Packing Triangles in Regular Tournaments. *J. of Graph Theory*, 74(1):58–66,
983 2013.