SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

May 2023, Vol. 66 152104:1–152104:14
https://doi.org/10.1007/s11432-021-3551-2

Improved kernels for triangle packing in tournaments

Hanchun YUAN[1], Qilong FENG[1*] & Jianxin WANG[2]

[1]School of Computer Science and Engineering, Central South University, Changsha 410083, China;
[2]Hunan Provincial Key Lab on Bioinformatics, Central South University, Changsha 410083, China

Received 30 November 2021/Revised 18 April 2022/Accepted 15 July 2022/Published online 17 April 2023

Abstract Triangle packing problem has been paid lots of attention to in the literature. In this paper, we study the kernelization of the triangle packing problem in tournaments. For the parameterized arc-disjoint triangle packing problem in tournaments, we find a maximal arc-disjoint triangle packing with the number of vertices bounded by $2.5k$. Based on the relation between the maximal arc-disjoint triangle packing and the vertices outside of the packing, a kernel of size $3.5k$ for the problem is obtained, improving the previous best one $6k$. For the parameterized vertex-disjoint triangle packing problem in sparse tournaments, several new properties between the triangles in maximal vertex-disjoint triangle packing and the arcs in the feedback arc set are presented, which result in a kernel of size $7k$ for the problem, improving the previous best one of $15k$. We also give a $7k$ vertex kernel for the parameterized feedback vertex set problem in sparse tournaments. The kernelization process presented in this paper for the parameterized arc-disjoint triangle packing can be applied to solve the parameterized arc-disjoint triangle packing problem on other restricted directed graph classes.

Keywords tournaments, triangle packing, kernelization, fixed-parameter tractable, graph algorithms

Citation Yuan H C, Feng Q L, Wang J X. Improved kernels for triangle packing in tournaments. Sci China Inf Sci, 2023, 66(5): 152104, https://doi.org/10.1007/s11432-021-3551-2

1 Introduction

Cycle packing problem forms an important class of problems in graph theory and computer science, which has applications in many fields, such as computational biology, kidney exchange program, etc. [1–3]. The problem is, given a graph G, to find the maximum number of (vertex or arc/edge) disjoint cycles of G. The cycle packing problem has been studied extensively from approximation and parameterized algorithms points of view [4–8]. As a special case of the cycle packing problem, the triangle packing problem has also been paid lots of attention. For undirected graphs, the edge-disjoint triangle packing problem is known to be NP-hard [9], even restricted to planar graphs with a maximum degree of five [10]. Kann [11] proved that in undirected graphs the edge-disjoint triangle packing problem is APX-complete even for graphs with a maximum degree of four. Chlebík and Chlebíková [12] proved that it is NP-hard to obtain an approximation factor better than 0.9929 for the edge-disjoint triangle packing problem.

Given a graph G and a nonnegative integer k, the parameterized triangle packing problem is to find at least k (vertex or arc/edge) disjoint triangles, or report that no such packing exists. Given an instance (G, k) of a problem Q, if it can be solved in $f(k) \cdot n^c$ time, then we call such problem fixed-parameter tractable (FPT). Besides the cycle/triangle packing problems, there are other important problems that have been well-studied, such as vertex cover and so on [13–16]. A problem admits a kernel if there is an algorithm that transforms (G, k) into a new instance (G', k') such that (G, k) is a yes-instance if and only if (G', k') is a yes-instance, where $k' \leqslant k$, and $|G'| \leqslant g(k)$ for some computable function g. The size $|G'|$ is also called the size of the kernel. A problem is FPT if and only if it admits a kernel [17]. Since FPT implies a kernel, one important direction is whether a problem admits a polynomial-size kernel, and many techniques are proposed for this purpose, see [18]. The existence of a polynomial kernel is also a crucial step for designing other algorithms. For example, the first step of many subexponential

* Corresponding author (email: fengcsu@csu.edu.cn)

parameterized algorithms—the running time is $2^{o(k)} \cdot n^{O(1)}$, needs a polynomial kernel for bounding the size of the input by $k^{O(1)}$ [19–25]. Although there may be other methods to compute large-scale graphs, e.g., parallel algorithms [26], the existence of a polynomial kernel can reduce the large graph to a small graph and the kernelization algorithm can be easy to implement in practice and can be easy to analyze in theory.

For the parameterized edge-disjoint triangle packing in undirected graphs, Shaw et al. [27] gave a kernel of size $4k$. Yang [28] presented an improved kernel of size $3.5k$. Lin and Xiao [29] presented a kernel of size $(3+\epsilon)k$ for the parameterized edge-disjoint triangle packing.

A tournament is a directed graph in which there is a single arc between every pair of distinct vertices, which has been studied extensively in graph theory [30]. The tournaments are also widely used to make selection models and search models [31–34] in artificial intelligence and machine learning. A feedback arc set of a given graph is a set of arcs whose deletion results in an acyclic graph. A tournament is sparse if it admits a feedback arc set that is matching in the tournament. For a set P of triangles, if no two triangles in P have common arcs, then P is called an arc-disjoint triangle packing; if no two triangles in P have common vertices, then P is called a vertex-disjoint triangle packing. In this paper, we study the following problems.

- Parameterized Arc-Disjoint Triangle Packing in Tournaments (k-ATP-T): Given a tournament T and a parameter k, the problem is to decide whether there is an arc-disjoint triangle packing of size at least k.
- Parameterized Vertex-Disjoint Triangle Packing in Sparse Tournaments (k-TP-ST): Given a sparse tournament T and a parameter k, the problem is to decide whether there is a vertex-disjoint triangle packing of size at least k.
- Parameterized Feedback Vertex Set in Sparse Tournaments (k-FVS-ST): Given a sparse tournament T and a parameter k, the problem is to decide whether there is a vertex set S of size at most k such that 'the graph' $T-S$ contains no directed cycle.

Bessy et al. [35] gave a kernel of size $6k$ for the k-ATP-T problem. Bessy et al. [36] presented a kernel of size $15k$ for the k-TP-ST problem. Le et al. [37] proved that the parameterized vertex-disjoint triangle packing in tournaments has a kernel with $O(k^{1.5})$ vertices. Whereas, the arc-disjoint triangle packing problem, the arc-disjoint cycle packing problem, and the feedback arc set problem can be solved in polynomial time in sparse tournaments [35]. The parameterized feedback vertex set in tournaments is also an interesting problem. Fomin et al. [37] gave an $O(k^{1.5})$ vertex kernel for the parameterized feedback vertex set problem in tournaments.

We point out that the kernel of size $3.5k$ for the parameterized edge-disjoint triangle packing problem on undirected graphs in [28] and the kernel of size $(3+\epsilon)k$ in [29] cannot be applied to solve the parameterized arc-disjoint triangle packing problem in tournaments, because some structures cannot be broken by the reduction rules in [28, 29], e.g., (1) a K_4 broken by Rule 3 in [28] cannot be extended to the tournaments and (2) the definition of 'span' in [29] cannot be extended to the tournaments.

In this paper, for the k-ATP-T problem, we give a process to find a maximal arc-disjoint triangle packing, which guarantees that the number of vertices in this maximal arc-disjoint triangle packing is bounded by $2.5k$. Based on the arcs in maximal arc-disjoint triangle packing and the vertices outside the packing, a bipartite graph can be constructed. By studying the relation between maximum matching in the bipartite graph and arc-disjoint triangle packing, we get that the number of vertices outside the maximal arc-disjoint triangle packing is bounded by k. Thus, a $3.5k$ vertex kernel is obtained for the k-ATP-T problem, improving the current best result of $6k$. For the k-TP-ST problem, we study the relation between maximal vertex-disjoint triangle packing and the feedback arc set. By bounding the number of arcs in the feedback arc set through the triangles in maximal vertex-disjoint triangles, a kernel of size $7k$ is given for the k-TP-ST problem, which improves the current best result of $15k$. We also give a $7k$ vertex kernel for k-FVS-ST. We point out that the kernelization process presented in this paper can be applied to solve the parameterized arc-disjoint triangle packing problem on other restricted directed graph classes.

2 Preliminaries

Given a digraph $D=(V,A)$, for two vertices $u,v \in V$, let (u,v) denote an arc oriented from u to v. For a subset $V' \subseteq V$, let $D[V']$ be the subgraph induced by V'. For a subset $A' \subseteq A$ of arcs, let $V(A')$ be the set of vertices contained in A', and let $D[A']$ denote the subgraph constructed by the vertices in $V(A')$ and all the arcs in A'. For three vertices $u, v, w \in V$, a triplet (u, v, w) is a triangle if the vertices u, v, and w construct a directed cycle. For a subset $V' \subseteq V$, let $D \backslash V'$ denote the graph obtained by removing the vertices in V' from D. Similarly, for a subset $A' \subseteq A$, let $D \backslash A'$ denote the graph obtained by removing arcs in A' from D. Given an undirected graph G, for two vertices $x, y \in G$, an edge between x and y is denoted by xy. Let $N(x)$ be the set of neighbors of x in G. For an arc-disjoint or vertex-disjoint triangle packing P, let $A(P)$ be the set of arcs contained in the triangles of P, and let $V(P)$ be the set of vertices in P.

For a given instance (T, k) of the k-ATP-T problem, an arc-disjoint triangle packing of size at least k in T is called a proper arc-disjoint triangle packing if the instance is yes. Hence, when we consider a proper arc-disjoint triangle packing, we always assume that the input instance is yes. For a proper arc-disjoint triangle packing P, if no triangle in $T \backslash A(P)$ can be added to P to get a larger packing, then P is called a maximal proper arc-disjoint triangle packing. For an arc (u, v) in a triangle (u, v, w), if (u, v) is shared with other triangles in T, then (u, v) is called a shared arc. A triangle is called a d-shared triangle if it has d shared arcs. For a 0-shared or 1-shared triangle (u, v, w), if (u, v, w) shares no arc with 2-shared or 3-shared triangles, then (u, v, w) is called a type-I triangle. A triangle that is not a type-I triangle is called a type-II triangle. For a triangle t, let $V(t)$ be the set of vertices in t, and let $S(t)$ denote the set of type-II triangles such that each triangle in $S(t)$ has one arc shared with t. Let Q be arc-disjoint triangle packing. For a triangle (u, v, w) in Q, if (u, v, w) shares no vertex with other triangles in Q, then (u, v, w) is called an independent triangle in Q. Let $I(Q)$ denote the set of independent triangles in Q.

For a given instance (T, k) of the k-TP-ST problem, a vertex-disjoint triangle packing of size at least k in T is called a proper vertex-disjoint triangle packing. A feedback arc set F that is matching in T is called a matched feedback arc set.

In this paper, we present several operations, called reduction rules, for designing the kernelization algorithm. We use reduction rules on an instance (G, k) to bound the size of the new instance (G', k'), i.e., the size of G'. When we apply a rule, we may delete some vertices, some edges, or even some components. After applying a rule, we need to prove that the rule is safe, that is, after applying a reduction rule, the new instance (G', k') is yes if and only if the original instance (G, k) is yes.

3 A 3.5k vertex kernel for k-ATP-T

In this section, we present a $3.5k$ vertex kernel for k-ATP-T. We first find two arc-disjoint triangle packings M_1 and M_2, and then bound the size of graphs containing M_1 and M_2, respectively. Note that the two graphs maybe intersect. From the size of the two graphs, we bound the size of the reduced graph by $3.5k$ vertices. Given an instance (T, k) of the k-ATP-T problem, we give the following rule, which can help us bound the size of the graph containing M_1.

Rule 1. For any vertex u in T, if u is not contained in any triangle, then delete u from T.

We can apply Rule 1 by a simple method. For each vertex v, we check whether v is contained in some triangle. If v is not contained in any triangle, then we remove it from T. If each vertex in T is contained in some triangles, then we call (T, k) a reduced instance by applying Rule 1.

Lemma 1. Given a yes-instance (T, k), for any triangle (u, v, w), if (u, v, w) has at most one shared arc in T, then there exists a proper arc-disjoint triangle packing that contains (u, v, w).

Proof. Assume that P is a maximal proper arc-disjoint triangle packing in T. We prove the lemma by contradiction; i.e., suppose that there is a 0-shared or 1-shared triangle (u, v, w) not in P. We consider the following two cases.

(1) Triangle (u, v, w) is 0-shared. In this case, the triangle (u, v, w) must be in P. Otherwise, a larger arc-disjoint packing P' can be obtained by adding (u, v, w) to P, contradicting that P is a maximal proper arc-disjoint triangle packing.

(2) Triangle (u, v, w) is 1-shared. Without loss of generality, assume that the arc (u, v) in (u, v, w) is shared with the triangle (u, v, z). Assume that (u, v) is not in $A(P)$. Then, (u, v, z) must be in P. Otherwise, a larger arc-disjoint packing P' can be obtained by adding (u, v, z) to P, contradicting that P is a maximal proper arc-disjoint triangle packing. We construct a proper arc-disjoint triangle packing P'' in the following way: add all triangles in P to P'', delete (u, v, z) from P, add (u, v, w) to P'', and add all other possible arc-disjoint triangles in P''. It is easy to see that P'' is also a maximal proper arc-disjoint triangle packing.

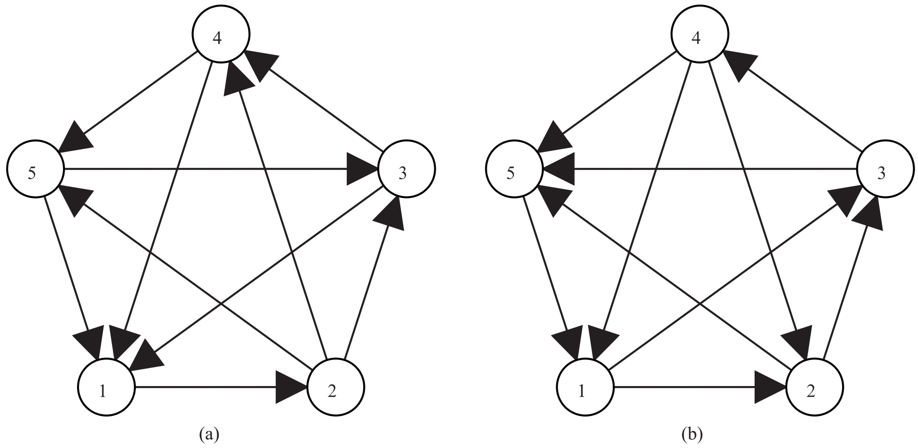

Figure 1 The illustration of $\mathrm{Tr}(u,v)$. In (a), the set of triangles containing vertex 1 is $X = \{(1,2,3),(1,2,4),(1,2,5)\}$. The set of triangles containing vertex 2 is also X. Hence, $\mathrm{Tr}(1,2) = \{(1,2,3),(1,2,4),(1,2,5)\}$. However, in (b), the set of triangles containing vertex 1 is $X_1 = \{(1,2,4),(1,2,5),(1,3,4)\}$, and the set of triangles containing vertex 2 is $X_2 = \{(1,2,4),(1,2,5),(2,3,4)\}$. Since $X_1 \neq X_2$, $\mathrm{Tr}(1,2) = \emptyset$. One can also see that in (a), $\mathrm{Tr}(2,3) = \emptyset$. In (a), since $\mathrm{Tr}(1,2) \neq \emptyset$, by applying Rule 2, the vertices 1 and 2 can be removed from (a) and the parameter k is decreased by one.

Recall that when we consider a proper arc-disjoint triangle packing, we assume that the input instance is yes. In the following, we omit this assumption.

Corollary 1. For any set Q of arc-disjoint triangles in T, if each triangle in Q has at most one shared arc in T, then there exists a proper arc-disjoint triangle packing that contains all the triangles in Q.

Proof. We prove this corollary by contradiction. Suppose that P is a proper arc-disjoint triangle packing such that $t \in Q$ is not in P. If t is 0-shared, we can add t to P since no triangle in P shares arc with t. If t is 1-shared and cannot be added to P, then t shares an arc with some triangle $t' \in P$. Since Q is arc-disjoint and $t \in Q$, we have that $t' \notin Q$. Let $P = P \backslash t' \cup \{t\}$. Since we can always add one triangle in Q to P while removing one triangle from P not in Q one by one, we have a proper arc-disjoint triangle packing containing all the triangles in Q.

To design further reduction rules, we describe a structure. For an arc (u,v), if the set X of triangles containing u is exactly the set of triangles containing v, then each triangle in X is 1-shared, and let $\mathrm{Tr}(u,v) = X$. If the set of triangles containing u is not the set of triangles containing v, then $\mathrm{Tr}(u,v) = \emptyset$. See Figure 1 for an illustration.

Lemma 2. For any two different vertices u and v in T, if $\mathrm{Tr}(u,v)$ is not empty, then for any triangle t in $\mathrm{Tr}(u,v)$, there exists a proper arc-disjoint triangle packing that contains t.

Proof. Since each triangle t in $\mathrm{Tr}(u,v)$ has one shared arc, by Lemma 1, there exists a proper arc-disjoint triangle packing that contains t.

By Lemma 2, we get the following reduction rule. The reader can take a look at Figure 1 as an example.

Rule 2. For any two different vertices u and v in T, if $\mathrm{Tr}(u,v)$ is not empty, then delete u and v from T, and decrease k by one.

Let (T,k) be the reduced instance of the k-ATP-T problem by applying Rules 1 and 2. We now give an algorithm to find two arc-disjoint triangle packings M_1 and M_2, as given in Algorithm 1. If the output of the algorithm is "yes", then in polynomial time, we have a kernel with empty size: the input instance is yes if and only if the instance $(\emptyset, 0)$ is yes. If the output is not "yes", the algorithm finds an arc-disjoint triangle packing M_1, in which each triangle is a type-I triangle. Based on M_1, a subgraph T_2 of T can be constructed such that each triangle in T_2 is a type-II triangle. Then, an arc-disjoint triangle packing M_2 from T_2 can be found such that each triangle in M_2 shares at least one vertex with other triangles in M_2.

Theorem 1. Given an input instance (T,k) of the k-ATP-T problem, by calling algorithm $\mathrm{MADP}(T,k)$, if the output is not "yes", two sets M_1, M_2 can be returned in $O(k^2 n^5)$ time (n is the number of vertices in T) such that the following:

(1) Triangle packing M_1 is arc-disjoint;

(2) Triangle packing M_2 is arc-disjoint, in which each triangle shares at least one vertex with other triangles in M_2.

Proof. Since each type-I triangle added to M_1 in step 5 shares no arc with other triangles in M_1, after

Algorithm 1 Maximal arc-disjoint packings (MADP)

Input: A tournament T, and a nonnegative parameter k.
Output: Return two arc-disjoint triangle packings M_1, M_2, and a subgraph T_2, or return "yes".
1: Apply Rules 1 and 2 exhaustively on T, and let T' denote the reduced graph;
2: $M_1 \leftarrow \emptyset$, $M_2 \leftarrow \emptyset$, $k_1 \leftarrow 0$;
3: **while** $k_1 < k$
4:   **if** exist a type-I triangle $t \in T' \backslash A(M_1)$ **then**
5:     $M_1 \leftarrow M_1 \cup \{t\}$, $k_1 \leftarrow k_1 + 1$;
6:   **else break**;
7: **if** $k_1 = k$ **then return** "yes";//Implying that the instance is yes.
8: $V_2 \leftarrow \emptyset$, $A_2 \leftarrow \emptyset$, $k_2 \leftarrow 0$;
9: $V_2 \leftarrow T' \backslash V(M_1)$;
10: Let T_2 be the subgraph induced by the vertices in V_2;
11: **for** each vertex u in $V(M_1)$ **do**
12:   **if** exist a type-II triangle C containing u in T' **then**
13:     $V_2 \leftarrow V_2 \cup \{u\}$;
14:     **for** each arc e with u as an endpoint **do**
15:       **if** e is not contained in M_1 **then** add e to T_2;
16: **while** ($k_2 < k - k_1$) and (exist a triangle t in $T_2 \backslash A(M_2)$ or $I(M_2) \neq \emptyset$)
//$I(M_2)$ denotes the set of independent triangels in M_2.
17:   **if** $M_2 = \emptyset$ **then** $M_2 \leftarrow M_2 \cup \{t\}$, $k_2 \leftarrow k_2 + 1$;
18:   **else**
19:     **if** exist a triangle $t \in T_2 \backslash A(M_2)$ such that $V(t) \cap V(M_2) \neq \emptyset$ **then**
20:       $M_2 \leftarrow M_2 \cup \{t\}$, $k_2 \leftarrow k_2 + 1$;
21:     **else** find a triangle in $T_2 \backslash A(M_2)$ arbitrarily, add it to M_2, and $k_2 \leftarrow k_2 + 1$;
22:   **if** exist no triangle in $T_2 \backslash A(M_2)$ **then**
23:     **if** exist a triangle $t \in I(M_2)$ **then**
24:       Find a triangle $t' \in S(t)$ with maximum shared arcs; //$S(t)$ denotes the set of type-II triangles sharing one arc with t.
25:       $M_2 \leftarrow (M_2 \backslash \{t\}) \cup \{t'\}$;
26: **if** $k_1 + k_2 \geqslant k$ **then return** "yes";//Implying that the instance is yes.
27: **return** M_1, M_2, and T_2.

the executions of steps 3–6, if M_1 is not empty, then no triangle in M_1 shares an arc with other triangles in M_1. Thus, M_1 is an arc-disjoint triangle packing. In steps 8–15, a subgraph T_2 can be constructed. Since no arc in $A(M_1)$ is added to T_2 in steps 8–15, no type-I triangle is contained in T_2. Thus, if T_2 has any triangle, it contains only type-II triangles. In steps 16–21, if M_2 is empty, then any triangle in T_2 can be added to M_2 if it exists. Moreover, if there exists a triangle t in $T_2 \backslash A(M_2)$, then t is added to M_2. Thus, all the triangles added to M_2 in steps 16–21 share no arc with other triangles in M_2.

Assume that the algorithm reaches step 22, i.e., no triangle is contained in $T_2 \backslash A(M_2)$. Then, M_2 obtained so far is a maximal arc-disjoint triangle packing. However, there may exist some independent triangles in M_2. In order to guarantee that each triangle in M_2 shares at least one vertex with other triangles in M_2, we need to replace the independent triangles in M_2 in steps 23–25. For the independent triangle t in M_2 found in step 23, we prove the following claims.

Claim 1. Before executing steps 22–25, if t is a 1-shared triangle contained in M_2, then for any triangle t_1 in $S(t)$, the vertex in $V(t_1) \backslash V(t)$ is contained in $V(M_2)$.

Proof. Let $t = (u, v, w)$. Without loss of generality, assume that (u, v) is the shared arc. For any triangle t_1 in $S(t)$, assume that $t_1 = (u, v, z)$. Suppose that z is not in $V(M_2)$. Thus, all the arcs with z as an endpoint are not in M_2. Because (u, v, w) is an independent triangle, all the arcs with u, v, or w as an endpoint are also not in M_2. Since (u, v, z) is a 2-shared or 3-shared triangle, there exists a triangle t'' that shares one arc (not (u, v)) with (u, v, z). Assume that t'' and (u, v, z) share the arc (v, z). Obviously, t'' shares vertex v with the triangle (u, v, w). Since z is not in M_2, all the arcs in t'' are not contained in M_2, which implies that t'' can be added to M_2 to get a larger packing, contradicting the condition in step 22. Therefore, z is contained in M_2.

Claim 2. The execution of step 25 does not change any nonindependent triangle in M_2 into an independent triangle.

Proof. Let $t = (u, v, w)$. Since (u, v, w) shares no vertex with other triangles in M_2, all the arcs in T_2 with u or v as an endpoint are not in M_2. It is not hard to observe that each arc in t' has u or v as an endpoint, implying that no arc of t' is in $M_2 \backslash \{t\}$. Thus, after the execution of step 25, nonindependent triangles are not changed.

Claim 3. Let t be the independent triangle found in step 23, and let t' be the triangle obtained in step 24. After the executions of steps 22–25, if t is a type-II triangle, then step 20 will be executed again, and after that t' is not an independent triangle in M_2.

Proof. Let $t = (u, v, w)$. Without loss of generality, assume that (u, v) and (v, w) are two shared arcs in t. Let $t' = (u, v, z)$. Since (u, v, w) shares no vertex with other triangles in M_2, all the arcs with u, v, or w as an endpoint are not contained in M_2. Thus, no arc of (u, v, z) is in $M_2 \backslash \{t\}$. In step 25, triangle (u, v, w) is deleted from M_2, and t' is added to M_2. It is easy to see that the new M_2 is still an arc-disjoint triangle packing. After the executions of steps 22–25, since t is a type-II triangle, there exists another triangle t'' such that t'' and t share one arc (not arc (u, v)), and no arc of t'' is contained in M_2, i.e., t'' is contained in $T_2 \backslash A(M_2)$. Thus, for the 'while' loop in step 16, the condition is satisfied, and a triangle that shares one vertex with t' is added to M_2 in step 20. Therefore, t' is not an independent triangle in M_2.

By Claims 1–3, we get that M_2 is an arc-disjoint triangle packing, in which each triangle shares at least one vertex with other triangles in M_2.

The triangles in T can be found in $O(n^3)$. All the triangles with bounded shared arc can be found in $O(n^4)$ time. Thus, steps 3–7 can be executed in $O(kn^4)$ time. Since the number of vertices in $V(M_1)$ is bounded by $3k$, the number of arcs with u as an endpoint is bounded by n, and the number of edges in M_1 is bounded by $3k$, steps 10–15 take $O(k^2n^5)$ time. The conditions in step 16 can be checked in $O(n^3)$ time. Steps 19–21 take time $O(n^3)$. The condition in step 22 can be checked in $O(n^3)$ time, and the condition in step 23 can be checked in $O(k^2)$ time. For step 24, $S(t)$ can be constructed in $O(n)$ time. In step 25, the triangle t' can be found in $O(n^4)$ time. Thus, steps 16–25 can be executed in $O(kn^4)$ time. Then, the running time of algorithm MADP is bounded by $O(k^2n^5)$.

For two triangles t and t' in T, if t shares at least one vertex with t', then we call that t is adjacent to t'. If t and t' have no common vertex and there exists a sequence of triangles $(t, t_1, \ldots, t_i, t')$ such that t_j is adjacent to t_{j+1} $(1 \leqslant j < i)$, t is adjacent to t_1, and t_i is adjacent to t', then we call that t' can be reached by t. Given a set Q of arc-disjoint triangles, for each pair of triangles t and t' in Q, if t is adjacent to t' or t can be reached by t' using only the triangles in Q, then we call that Q is a connected packing.

Lemma 3. Given a reduced instance (T, k) of the k-ATP-T problem, for the arc-disjoint packing M_2 returned by algorithm MADP, the number of vertices in M_2 is bounded by $2.5|M_2|$.

Proof. For the triangles in M_2, let $\{Q_1, \ldots, Q_i\}$ be the collection of maximal connected packings such that $Q_j \cap Q_{j'} = \emptyset$ $(1 \leqslant j, j' \leqslant i, j \neq j')$ and $\bigcup_{j=1}^{i} Q_j = M_2$. It is easy to get that $|Q_1| + \cdots + |Q_i| = |M_2|$. For any Q_j $(1 \leqslant j \leqslant i)$ in $\{Q_1, \ldots, Q_i\}$, since each triangle in M_2 shares at least one vertex with other triangles in M_2, the size of Q_j is at least two. We now prove the following claim which is used to prove this lemma.

Claim 4. The number of vertices contained in Q_j is bounded by $2|Q_j| + 1$.

Proof. Let h be the number of triangles contained in Q_j. We prove the claim by induction on h. For the case that $h = 2$, i.e., Q_j contains two triangles, since Q_j is a connected packing, the two triangles in Q_j share one vertex. Thus, the number of vertices in $V(Q_j)$ is five, which is $2h + 1$. Now we assume that any connected packing Q_j of size $h = d$ has at most $2d + 1$ vertices. We consider the case that $h = d + 1$. For any Q_j with $d + 1$ arc-disjoint triangles, since Q_j has $d + 1$ triangles, a subset Q' of size d of Q_j can be found such that Q' is also a connected packing. Otherwise, Q_j is not a connected packing. Let t be the triangle in $Q_j \backslash Q'$. By induction assumption, Q' has at most $2|Q'| + 1$ vertices. Since Q_j is a connected packing, t must have at least one vertex shared with the triangles in Q'. Thus, we get $V(Q_j) \leqslant 2d + 1 + 2 = 2(d + 1) + 1$.

Therefore, by Claim 4, the total number of vertices contained in $\{Q_1, \ldots, Q_i\}$ is

$$\sum_{j=1}^{i} (2|Q_j| + 1) = 2(|Q_1| + \cdots + |Q_i|) + i = 2|M_2| + i.$$

Since Q_j $(1 \leqslant j \leqslant i)$ contains at least two triangles, i is bounded by $\frac{1}{2}|M_2|$. Thus, the total number of vertices contained in $\{Q_1, \ldots, Q_i\}$ is bounded by $2|M_2| + \frac{1}{2}|M_2| = 2.5|M_2|$.

Let (T, k) be the reduced instance obtained by applying Rules 1 and 2 exhaustively. We now analyze the number of vertices in M_1. By Corollary 1, we can get the following reduction rule.

Rule 3. Given a reduced instance (T, k) of the k-ATP-T problem, for the arc-disjoint packing M_1 returned by algorithm MADP, let Q be a maximal connected packing in M_1. If $V(Q) \cap V(T_2) = \emptyset$, then delete the vertices in Q from T and M_1, and $k = k - |Q|$.

Let (T, k) be the reduced instance obtained by applying Rules 1–3 exhaustively.

Lemma 4. Given a reduced instance (T, k) of the k-ATP-T problem, for the arc-disjoint packing M_1 applying Rule 3, the number of vertices in $V(M_1)\backslash V(T_2)$ is bounded by $2|M_1|$.

Proof. We first prove that for any maximal connected packing Q of M_1, the number of vertices in $V(Q)\backslash V(T_2)$ is at most $2|Q|$. For the case that Q has only one triangle, because Rule 3 cannot be applied anymore on T, the triangle in Q must have at least one vertex shared with the triangles in T_2. Thus, the number of vertices in $V(Q)\backslash V(T_2)$ is at most two. Now we assume that the size of Q is at least two. Since Q is a connected packing, similar to the proof of Claim 4, we can get that the number of vertices in Q is bounded by $2|Q|+1$. Because there must exist at least one vertex in $V(Q)$ that is contained in $V(T_2)$, the number of vertices in $V(Q)\backslash V(T_2)$ is bounded by $2|Q|$.

Let $\{P_1, \ldots, P_i\}$ be the collection of maximal connected packings in M_1 such that $P_j \cap P_{j'} = \emptyset$ $(1 \leqslant j, j' \leqslant i,\ j \neq j')$, and $\bigcup_{j=1}^{i} P_j = M_1$. We get that

$$\begin{aligned}|V(M_1)\backslash V(T_2)| &= |V(P_1)\backslash V(T_2)| + \cdots + |V(P_i)\backslash V(T_2)| \\ &= 2|P_1| + \cdots + 2|P_i| = 2|M_1|.\end{aligned}$$

We now analyze the number of vertices in T_2. Let $V_{12} = V(M_1) \cap V(T_2)$; i.e., V_{12} is the set of vertices both in M_1 and T_2. To bound the number of vertices in T_2, we construct a bipartite graph. For each arc e in M_2, add a vertex v_e to Y. Add all the vertices in $V(T_2)\backslash V(M_2)$ to B. We construct a bipartite graph $G = (Y \uplus B, E)$ as follows. For two vertices $u \in Y$ and $v \in B$, if there is a triangle constructed by the corresponding arc of u and vertex v in T_2, then add an edge uv to E. Let H be a maximum matching in the bipartite graph G. Based on H, each edge in H corresponds to a triangle. Since H is a matching, any two edges h_1, h_2 in H share no vertex. The two triangles t_1 and t_2 corresponding to h_1 and h_2 respectively share at most one vertex, hence arc-disjoint. Thus, the matching H corresponds to an arc-disjoint triangle packing in T_2 of size $|H|$. By Corollary 1, if $|H| \geqslant k - |M_1|$, then we get that (T, k) is a yes-instance. In the following, we assume that the size of H is less than $k - |M_1|$.

Lemma 5. Let (T, k) be an input instance of k-ATP-T and let T_2 be the subgraph returned by the algorithm MADP. For any subset X of $B\backslash(V(H) \cup V_{12})$, if there exists a proper arc-disjoint triangle packing S of T containing the vertices in X, then there exists a subset X' of $V(H) \cap B$ such that a proper arc-disjoint triangle packing S' can be constructed satisfying that (1) S' contains the vertices in X' and no vertex in X; (2) $|S'| = |S|$.

Proof. Assume that S is a proper arc-disjoint triangle packing in T containing the vertices in X. Let $S_1 = S \cap M_1$. If S_1 is not empty, then delete the triangles in S_1 from S. Based on the bipartite graph $G = (Y \uplus B, E)$, we construct a new bipartite graph $G' = (Y \uplus B, E')$ as follows. Let $E' = E$. For each edge uv in $E\backslash H$, assume that u is contained in Y, and v is contained in B. If the triangle constructed by the corresponding arc of u and vertex v is not in S, then delete edge uv from E'. Let $K = E'\backslash H$. It is easy to see that for each edge uv in K, where u is contained in Y and v is contained in B, the triangle constructed by the corresponding arc of u and vertex v is contained in S.

For any vertex x in X, we now build a subgraph F_x using the edges in K and H. We will build a collection of sets $L_1, \ldots, L_h$, where L_i $(1 \leqslant i \leqslant h)$ is the set of vertices of G' contained in F_x and $L_i \cap L_j = \emptyset$ $(1 \leqslant i, j \leqslant h, i \neq j)$. First, add x to L_1. For each edge e in K with x as an endpoint, add the other endpoint of e to L_2, and add edge e to F_x. Assume that L_i $(1 \leqslant i < h)$ has been constructed. We now give how to construct L_{i+1}. For each vertex v in L_i, if i is even, then a matched edge e' in H containing v as an endpoint can be found, the other endpoint of e' is added to L_{i+1}, and e' is added to F_x. If i is odd and there exists at least one edge in K containing v as an endpoint, then add all the vertices in $N(v)\backslash L_i$ to L_{i+1}, and add all the edges $\{uv | u \in N(v)\backslash L_i\}$ to F_x. Repeat the above process until L_h is constructed. We prove the following claims.

Claim 5. The subgraph F_x is a tree, and each path in F_x from the root to a leaf has an even number of edges.

Proof. We construct a directed graph F'_x based on F_x. Let x be the root in F'_x. For each edge uv in F_x, where u is contained in L_i and v is contained in L_{i+1} $(1 \leqslant i < h)$, we add an arc from u to v to F'_x. Obviously, x has in-degree zero. For each vertex w in L_i where i is an even number, the in-degree of w is at most one. Otherwise, there exist two edges e_1 and e_2 in K such that e_1 and e_2 share one vertex contained in Y, implying that the corresponding two triangles in S share one arc, a contradiction. For each vertex w in L_i where i is an odd number, the in-degree of z is at most one. Otherwise, there exist two edges e_1 and e_2 in H such that e_1 and e_2 share one vertex in B, implying that two matched edges have a common vertex, a contradiction. Thus, each vertex except x in F'_x has in-degree one. Since x has in-degree zero and other vertices in $F'_x \backslash \{x\}$ has in-degree one, F'_x is a directed acyclic graph. We now prove that the underlying graph F_x of F'_x is a tree. Since F'_x is a directed acyclic graph, if there exists a cycle in F_x, then a vertex with in-degree two in F'_x can be found, contradicting that each vertex in F'_x has in-degree one or zero.

We now prove that each path in F_x from root to leaf has an even number of edges. Since x is contained in X, then all the edges with x as an endpoint are not in H, implying that the edges between L_1 and L_2 are unmatched edges. By the construction of F_x, the edges between L_i and L_{i+1} are unmatched edges when i is an odd number, and the edges between L_i and L_{i+1} are matched edges when i is an even number. Thus, if there exists a path in F_x from root to a leaf with an odd number of edges, then there exists an augmenting path, contradicting that H is a maximum matching.

We now prove a claim that characterizes the relationship between two trees; say F_x and F_y.

Claim 6. For any two different vertices x, y in X, the two trees F_x, F_y constructed by the above process have no common vertex.

Proof. Assume that F_x and F_y have common vertices, F_x has height h_1, and F_y has height h_2. Assume that a vertex z is both contained in F_x and F_y, where z is contained in L_i of F_x such that i is the smallest index of all the sets having a common vertex with F_y. Assume that uz is contained in F_x, where u is in L_{i-1}. Assume that vz is contained in F_y, where z is contained in L_j of F_y, and v is contained in L_{j-1} of F_y. Obviously, v is not in F_x, and u is not in F_y. We consider the following cases.

(1) Number i is odd (even) and number j is even (odd). It is easy to see that in F_x, z is contained in B, and in F_y, z is contained in Y, a contradiction. Similarly, when i is an even number and j is an odd number, a contradiction can be obtained.

(2) Number i and number j are both odd. It is easy to see that in F_x and F_y, z is contained in B. Thus, all the vertices contained in L_{i-1} and L_{j-1} are contained in Y. Since uz and vz are contained in H, v is not in F_x, and u is not in F_y, there exist two matched edges that have a common vertex, which is a contradiction.

(3) Number i and number j are both even. It is easy to see that in F_x and F_y, z is contained in Y. Thus, all the vertices in L_{i-1} and L_{j-1} are contained in B. Since uz and vz are in K, v is not in F_x, and u is not in F_y, two triangles in S share an arc, which is a contradiction.

The next claim guarantees that we can find some triangles by the constructed trees to replace some triangles in S.

Claim 7. By considering all the vertices in X, a collection F_X of trees can be obtained, and $|F_X| = |X|$. For any triangle (u, v, w) in S, if no vertex from $\{u, v, w\}$ is contained in B and there exists an edge e in H such that e is constructed by the corresponding vertex of an arc e' from (u, v, w) and a vertex b in B, then e cannot be in F_X.

Proof. Since arc e' is contained in (u, v, w) and (u, v, w) is contained in S, any other triangle except (u, v, w) in S cannot contain e'. Otherwise, S is not an arc-disjoint triangle packing. Thus, for the corresponding vertex v' of arc e' in Y, there is no edge with v' as an endpoint in K. Suppose that e is contained in F_X, then there must exist another edge e'' with v' as an endpoint in K, which is a contradiction.

For each vertex x in X, each edge in F_x corresponds to a triangle in T, and we add the triangles constructed by the edges between L_i $(1 \leqslant i < h,\ i$ is an even number$)$ and L_{i+1} in F_x to S'. By Claim 5, the number of the edges between L_i $(1 \leqslant i < h,\ i$ is an even number$)$ and L_{i+1} in F_x is equal to the number of the edges between L_i $(1 \leqslant i < h-1,\ i$ is an odd number$)$ and L_{i+1} in F_x. Note that the edges between L_i $(1 \leqslant i < h,\ i$ is an even number$)$ and L_{i+1} in F_x are all contained in H, and the edges between L_i $(1 \leqslant i < h-1,\ i$ is an odd number$)$ and L_{i+1} in F_x are all contained in K, implying that $S' \cap S = \emptyset$ and S' contains no vertex in X. By Claim 6, for any two different vertices x and y, F_x and F_y have no common edge. Thus, S' is an arc-disjoint triangle packing. By Claim 7, no triangle in $\{(u, v, w) | \{u, v, w\} \subseteq V(M_2), (u, v, w) \in S\}$ shares arcs with any triangle in S'. Thus, all the triangles in $\{(u, v, w) | \{u, v, w\} \subseteq V(M_2), (u, v, w) \in S\}$ can be added to S'. By the construction process of the trees in F_X, for a vertex w in $(V(B) \backslash V(H)) \cap V_{12}$, if w is contained in a triangle of S, then no edge with endpoint w is contained in F_X. Thus, we can add all the triangles in S with a vertex in $(V(B) \backslash V(H)) \cap V_{12}$ to S'. Since all the triangles in S_1 are type-I triangles, we have that $S_1 \cap S' = \emptyset$. Add all the triangles in S_1 to S', and we get that $|S'| = |S|$.

By Lemma 5, we get the following reduction rule.

Rule 4. For the reduced instance (T, k) of the k-ATP-T problem and based on the bipartite graph $G = (Y \uplus B, E)$, delete the vertices in $B \backslash (V(H) \cup V_{12})$ from T and B.

We then prove the final conclusion of this section.

Theorem 2. The k-ATP-T problem admits a kernel of size $3.5k$.

Proof. Assume that (T, k) is the reduced instance obtained by applying Rules 1–4 exhaustively. By calling algorithm MADP, if the output is not "yes", two maximal arc-disjoint packings M_1, M_2, and a subgraph T_2 of T can be obtained. By Lemma 3, $|V(M_2)|$ is bounded by $2.5|M_2|$. Since Rule 4 is not applicable on T, $T_2 \backslash V(M_2)$ is bounded by $k - |M_1| + |V_{12}|$. Thus, the number of vertices in T_2 is bounded by

$$2.5|M_2| + k - |M_1| + |V_{12}|,$$

and T has at most

$$\begin{aligned}|V(M_1)| - |V_{12}| + |T_2| &\leqslant 3|M_1| - |V_{12}| + 2.5|M_2| + k - |M_1| + |V_{12}| \\ &\leqslant k + 2.5(|M_1| + |M_2|)\end{aligned}$$

vertices. Since $|M_1| + |M_2| \leqslant k$, the number of vertices in T is bounded by $3.5k$.

4 A $7k$ vertex kernel for k-TP-ST

In this section, we give a $7k$ vertex kernel for k-TP-ST. The main idea contains two steps. First, we prove that the kernel size is bounded by $2|F| + k$, where F is a matched feedback arc set. Second, we prove that $|F| \leqslant 3k$. Thus, the kernel size is bounded by $7k$ vertices. We first give the following lemma which implies that F can be computed in polynomial time.

Lemma 6 ([36])**.** For a sparse tournament T, a matched feedback arc set of T can be found in polynomial time.

If a vertex v is not contained in any triangle, then we can delete it since the input instance (T, k) is yes if and only if the instance $(T \backslash \{v\}, k)$ is yes. Hence, we have the following reduction rule.

Rule 5. For any vertex u in T, if u is not contained in any triangle, then delete u from T.

Let (T, k) be the reduced instance obtained by applying Rule 5, and let F be a matched feedback arc set in T obtained by the polynomial algorithm in [36]. For each arc $e \in F$, add a vertex v_e to Y. Add the vertices in $V(T) \backslash V(F)$ to B. We construct a bipartite graph $G = (Y \uplus B, E)$ as follows. For two vertices $u \in Y$ and $v \in B$, if there is a triangle constructed by the corresponding arc of u and vertex v, then add uv to E. Let H be a maximum matching in G. For an edge $yb \in H$ where $y \in Y$ and $b \in B$, there is a triangle with vertex b and with the arc corresponding to y. Since H is a matchcing, any pair of edges in H is vertex-disjoint. Note that F is a matching, implying that H is a set of vertex-disjoint triangles. Thus, if $|H| \geqslant k$, then (T, k) is a yes-instance. In the following, we assume that the size of H is less than k.

Lemma 7. For the reduced instance (T, k) of the k-TP-ST problem, based on the bipartite graph $G = (Y \uplus B, E)$ and matched feedback arc set F, for any triangle $t = (u, v, w)$ in T, t does not satisfy any condition in the following:

(1) Triangle t contains at least two vertices in B;

(2) Triangle t contains one vertex in B and the other two vertices in $\{u, v, w\} \backslash B$ are contained in different arcs of F.

It is easy to see that if t satisfies any condition in Lemma 7, then all the arcs of t are not contained in F, contradicting that F is a matched feedback arc set in T. Hence we omit the proof of Lemma 7. For each edge e in G, a triangle t of T can be constructed based on e. For simplicity, we say that t is the corresponding triangle of edge e.

Lemma 8. For the reduced instance (T, k) of the k-TP-ST problem, for any subset X of $B \backslash V(H)$, if there exists a proper vertex-disjoint triangle packing S of T containing all the vertices in X, then there exists a subset X' of $V(H) \cap B$ such that a proper vertex-disjoint triangle packing S' can be constructed satisfying that (1) S' contains the vertices in X' and no vertex in X; (2) $|S'| = |S|$.

The proof of Lemma 8 is similar to the one of Lemma 5 and we omit it. By Lemma 8, we get the following rule.

Rule 6. For the reduced instance (T, k) of the k-TP-ST problem and based on the bipartite graph $G = (Y \uplus B, E)$, delete the vertices in $B \backslash V(H)$ from T and B.

Assume that (T, k) is the reduced instance obtained by applying Rules 5 and 6, and let F be a matched feedback arc set in T.

Lemma 9. For the reduced instance (T, k), the number of vertices in T is bounded by $2|F| + k$.

Proof. Since F is a matched feedback arc set, the number of vertices in $V(F)$ is $2|F|$. By Lemma 8, and because Rule 6 is not applicable on T, the number of vertices in B is bounded by k. Since each vertex of T is either contained in $V(F)$ or B, the number of vertices in T is bounded by $2|F| + k$.

A topological sort σ can be obtained in $T \backslash F$, denoted by $\sigma = (v_1, v_2, \ldots, v_n)$, where v_i is a vertex in $V(T)$ and n is the number of vertices in T. Let $\sigma(v)$ be the index of v in T. For any arc (v_i, v_j) $(1 \leqslant i, j \leqslant n, i \neq j)$ in T, if $i < j$, then arc (v_i, v_j) is called a forward arc under σ; if $i > j$, then arc (v_i, v_j) is called a backward arc under σ; if $i = j + 1$, then arc (v_i, v_j) is called a consecutive backward arc under σ. For any triangle (v_i, v_j, v_h) $(1 \leqslant i, j, h \leqslant n)$, if there exists an ordering of indexes in $\{i, j, h\}$ that are consecutive, then triangle (v_i, v_j, v_h) is called a consecutive triangle in T.

Let $\beta = (t_1, \ldots, t_h)$ be an ordering of the vertex-disjoint triangles in T, and let T_i be the subgraph obtained by deleting the vertices of triangles in $\bigcup_{r=1}^{i}\{t_r\}$ from T. If β satisfies the following properties: (1) the triangles in $\{t_1, \ldots, t_h\}$ construct a maximal vertex-disjoint triangle packing; (2) for each t_i $(1 \leqslant i \leqslant h)$, if t_i is not a consecutive triangle, no triangle in T_{i-1} is a consecutive triangle, then we call that β is a special ordering of the triangles.

For any triangle $t_i = (u, v, w)$ in β, if an arc e in the matched feedback arc set F satisfies one of the following conditions: (1) e is contained in t_i; (2) one endpoint of e is contained in t_i; (3) e is contained in a triangle in subgraph T_{i-1} and not contained in any triangle in subgraph T_i, then we say that e is influenced by triangle t_i. For a backward arc $f = (v_i, v_j)$ $(1 \leqslant j < i \leqslant n)$, we call f spans the vertex v if $j < \sigma(v) < i$. For a minimal matched feedback arc set F and a topological sort σ in $T \backslash F$, the following three lemmas give some properties between F and σ.

Lemma 10. Let F be a minimal matched feedback arc set in T and σ a topological sort in $T \backslash F$. Then, all the arcs in F are backward arcs under σ.

Proof. Assume that there exists an arc f contained in F that is not a backward arc. Then, all the arcs in $T \backslash (F \backslash \{f\})$ are forward arcs. Therefore, $F \backslash \{f\}$ is also a matched feedback arc set, contradicting that F is a minimal matched feedback arc set. Thus, all the arcs in F are backward arcs.

Lemma 11. Let F be a minimal matched feedback arc set in T and σ a topological sort in $T \backslash F$. For any arc $f = (x, y)$ in F, if there exists a vertex v in T such that $\sigma(y) < \sigma(v) < \sigma(x)$, then (y, v, x) is a triangle in T.

Proof. Since F is a minimal matched feedback arc set and f is an arc in F, all the arcs with u or v as an endpoint are not contained in $F \backslash \{f\}$. By Lemma 10, f is a backward arc. Then, all the arcs except f with u or v as an endpoint are forward arcs. It is not hard to see that (y, v, x) is a triangle under σ.

Lemma 12. For any arc $f = (x, y)$ in F, f is not contained in any triangle in T if and only if f is a consecutive backward arc under σ.

Proof. We first prove that if f is a consecutive backward arc in T, then f is not contained in any triangle in T. Assume that f is contained in a triangle t, then t needs two backward arcs in F, contradicting that F is a matching. We now prove the other direction. Assume that f is not a consecutive backward arc in T. By Lemma 11, there exists a vertex v in T such that (x, v, y) is a triangle.

Given a reduced instance (T, k) of the k-TP-ST, let F be a minimal matched feedback arc set of T, and let P be a maximal vertex-disjoint triangle packing with a special ordering $\beta = (t_1, \ldots, t_{|P|})$ of T. Recall that T_i $(1 \leqslant i \leqslant |P|)$ is the subgraph obtained by deleting the vertices of triangles in $\bigcup_{r=1}^{i}\{t_r\}$ from T. If $|P| \geqslant k$, then (T, k) is a yes-instance. In the following, we assume that the size of P is less than k.

Lemma 13. There exist at most three arcs in F that are influenced by a consecutive triangle in β.

Proof. Assume that $t_i = (u, v, w)$ is a consecutive triangle in β. Then, at least one arc in t_i is contained in F. Without loss of generality, assume that (w, u) is contained in F and influenced by t_i. It is possible that an arc in F may have v as an endpoint. Assume that after deleting the triangle t_i, there exists an arc f in F not contained in other triangles in T_{i-1}. Then, f is influenced by triangle t_i. We first prove that arc f spans vertices u, v, w. Let $f = (x, y)$, where $\sigma(y) < \sigma(x)$. Assume that f does not span u, v, w. Then, we get that either $\sigma(x) < \sigma(u)$ or $\sigma(y) > \sigma(w)$. Since F is a minimal matched feedback arc set,

f must be contained in a cycle and be a backward arc. Then, there must exist at least one vertex in T that is spanned by f, implying that f is not a consecutive arc. By Lemma 11, f is contained in a triangle without containing any vertex of t_i, contradicting that f is influenced by t_i.

Since after deleting the triangle t_i, arc f is not contained in any triangle in T_i, by Lemma 12, f is a consecutive arc in T_i. Thus, f spans only u, v, w in T. Therefore, after deleting the triangle t_i, there exists at most one non-consecutive backward arc in F that is not contained in any triangle in T_i. Thus, there exist at most three arcs in F that are influenced by consecutive triangle t_i.

We are ready to prove that $|F| \leqslant 3k$, except there are some unknown relationships between F and the influenced arcs in F. We first prove that the number of arcs in F influenced by a consecutive triangle is at most three.

Lemma 14. There exist at most three arcs in F that are influenced by a non-consecutive triangle in β.

Proof. Assume that $t_i = (u, v, w)$ is a non-consecutive triangle in β. Then, there is no consecutive triangle in T_{i-1}. At least one arc in t_i is contained in F. Without loss of generality, assume that (w, u) is contained in F. Since (w, u) is a backward arc in T_{i-1}, then $\sigma(u) < \sigma(v) < \sigma(w)$. For the triangle t_i, the backward arc (w, u) is contained in F and influenced by t_i. It is possible that an arc in F may have v as endpoint. Assume that after deleting the triangle t_i, there exists an arc f in F not contained in other triangles in T_{i-1}. Then, f is influenced by triangle t_i. We first prove that arc f spans exactly two vertices in $\{u, v, w\}$. Let $f = (x, y)$, where $\sigma(y) < \sigma(x)$. We consider the following cases.

(1) Arc f does not span u, v, w. The proof of this case is similar to the one of Lemma 13.

(2) Arc f spans u, v, w. Then, we get that $\sigma(y) < \sigma(u) < \sigma(w) < \sigma(x)$. Since t_i is not a consecutive triangle, there exists a vertex z such that $\sigma(u) < \sigma(z) < \sigma(w)$. By Lemma 11, f is contained in triangle (y, z, x) without containing any vertex in t_i, contradicting that f is influenced by t_i.

(3) Arc f only spans one vertex of $\{u, v, w\}$. Without loss of generality, assume that f spans u. By Lemma 11, (x, u, y) is a triangle. Since T_i contains no consecutive triangles, there exists a vertex z such that $\sigma(y) < \sigma(z) < \sigma(x)$. By Lemma 11, f is contained in triangle (y, z, x) without containing any vertex in t_i, contradicting that f is influenced by t_i.

Since after deleting the triangle t_i, arc f is not contained in any triangle in T_i, by Lemma 12, f is a consecutive arc in T_i. We have that f spans only two vertices in $\{u, v, w\}$ that are consecutive in σ. Therefore, after deleting the triangle t_i, there exists at most one non-consecutive backward arc (in F) that is not contained in any triangle in T_i. Thus, there exist at most three arcs in F that are influenced by triangle t_i.

By Lemmas 13 and 14, we get the following lemma.

Lemma 15. For a maximal vertex-disjoint triangle packing P with a special ordering β of T and a minimal matched feedback arc set F of T, if $|P| \leqslant k$, then $|F| \leqslant 3k$.

Proof. For any triangle t_i in β, if t_i is a consecutive triangle in P, by Lemma 13, at most three backward arcs in F are influenced by t_i. If t_i is not a consecutive triangle, by Lemma 14, at most three backward arcs in F are influenced by t_i. The number of triangles in β is $|P|$. Therefore, at most $3|P|$ arcs in F are influenced by the triangles in β.

We now prove that each arc in F must be influenced by the triangles in β. Assume that there exists an arc f in F that is not influenced by the triangles in β. Then, in the subgraph $T_{|P|}$, f is contained in a triangle with three vertices from $T \backslash V(P)$, contradicting that P is a maximal vertex-disjoint triangle packing. Thus, we get that if $|P| \leqslant k$, then $|F| \leqslant 3k$.

We are ready to prove the final conclusion of this section.

Theorem 3. The k-TP-ST problem admits a kernel of size $7k$.

Proof. For a given instance (T, k) of the k-TP-ST problem, by Lemma 6, a minimal matched feedback arc set F can be obtained in polynomial time. We first apply Rules 5 and 6 exhaustively on T. By Lemmas 9, the number of vertices in T is bounded by $2|F| + k$, and by Lemma 15, we get that $|F|$ is bounded by $3k$. Thus, the number of vertices in T is bounded by $2 \cdot 3k + k = 7k$.

5 A $7k$ vertex kernel for k-FVS-ST

In this section, we provide a kernel with $7k$ vertices for k-FVS-ST. Similarly to Section 4, we have that $|F| \leqslant 3k$; otherwise, there are more than k vertex-disjoint triangles, implying that the instance is 'no'—breaking this more than k vertex-disjoint triangles needs more than k vertices. Thus, the only thing we need is to prove that the kernel size is bounded by $2|F|+k$, where F is a matched feedback arc set.

Before designing the reduction rules, we first construct a bipartite graph as in Section 4. For each arc $e \in F$, add a vertex v_e to X. Add the vertices in $V(T)\backslash V(F)$ to Y. We construct a bipartite graph $G=(X \uplus Y, E)$ as follows. For two vertices $u \in X$ and $v \in Y$, if there is a triangle constructed by the corresponding arc of u and vertex v, then add uv to E. An expansion $B'=(X' \uplus Y', E', M)$ of B is a subgraph in B where $X' \subseteq X$, $Y' \subseteq Y$, E' is induced by $X' \cup Y'$, M is matching satisfying X and $N(Y') \subseteq X'$. If we cannot find an expansion structure, then we can directly bound the size of the kernel, which is proved later. One can see that $m \leqslant 3k$ by Lemma 15. Then, the main idea is removing the redundant vertices in $V(Y)$ and bounding the size of $V(Y)$. For a vertex $x \in X$, let $\operatorname{arc}(x)$ denote the corresponding arc in F of u. We first give a lemma which can help us to study the structure.

Lemma 16. For a k-FVS-ST instance (T,k), let $B'=(X' \uplus Y', E, M)$ be an expansion structure (if exist) of B. Then (T,k) is a yes-instance if and only if (T',k) is a yes-instance, where T' is from T by removing $Y'\backslash V(M)$.

Proof. As T' is the subgraph of T, if (T,k) is yes, then (T',k) is also yes. We prove the other direction. Suppose that there is a feasible solution S for (T',k). The main idea is that we can replace S by S' such that S' is a solution for (T,k) and $|S'| \leqslant |S|$. Let $Y_1 = Y' \cap V(M)$.

Let $R=\emptyset$. For each vertex $x \in X'$, if $S \cap V(\operatorname{arc}(x))=\emptyset$, then we randomly add one vertex from $V(\operatorname{arc}(x))$ to R. We obtain $S'=S\backslash Y_1 \cup R$. We have $|R| \leqslant |S \cap Y_1|$ because for each x $V(\operatorname{arc}(x)) \cap S=\emptyset$ there needs a vertex $y \in Y_1$ to break the triangle corresponding to the matching edge xy. Since $|R| \leqslant |S \cap Y_1|$, we have $|S'| \leqslant |S|$.

Now we prove that S' is a solution of (T,k). Suppose for contradiction that there is a triangle t in $T-S'$. Then t must contain some vertex in $Y'\backslash V(M)$. Since Y' is a vertex set in which each vertex is not incident to F, then t contains exactly one vertex $y \in Y'\backslash V(M)$. Noting that Y' is not in the minimum vertex cover, we have that any vertex in Y' is only adjacent to some vertex in X'. Moreover, t must be formed by $\operatorname{arc}(x)$ and y where $x \in X'$. However, we choose at least one vertex from $\operatorname{arc}(x)$, which implies that t cannot exist.

The following reduction is from a folklore that in a tournament a vertex is in a cycle if and only if it is in a triangle.

Rule 7. Delete all the vertices that are not in the triangles.

We apply Rule 7 exhaustively and find an expansion structure. For the bipartite graph $B=(X \uplus Y, E)$ corresponding to T, we find a minimum vertex cover. In the bipartite graph the minimum vertex cover V_α is equal to the maximum matching. If we have a matching M whose size is larger than k, then we get a no-instance because there are at least $k+1$ vertex-disjoint triangles. Then, the vertex cover of B is at most k. Letting $X'=V_\alpha \cap X$ and $Y'=Y\backslash V_\alpha$, we have a bipartite graph $B'=(X' \uplus Y', E')$ where E is the edges induced by $X' \cup Y'$. As X' is a subset of minimum vertex cover, there is a matching M satisfying X' into Y'. For a vertex $y \in Y'$, if there exists an edge e incident to y, then e is incident to some vertexes in X'; otherwise, e is not covered. So, we have $N(Y') \subseteq X'$. As a result, B' is an expansion structure. It is possible that $X'=\emptyset$. In this case, we prove that k-FVS-ST admits a $7k$ vertex kernel. If $X'=\emptyset$, then there is no edge e' incident to both a vertex in X and a vertex in $Y\backslash Y'$. By Rule 7, we have $Y=Y'$, and the number of the vertex is at most $2m+k$. By Lemma 15, $m \leqslant 3k$, we have a $7k$ vertex kernel. In the following, we assume that X' is not empty.

We provide a reduction rule which is related to the expansion structure.

Rule 8. For a k-FVS-ST instance (T,k), let $B'=(X' \uplus Y', E, M)$ be an expansion structure of B. Deleting all the vertices in $Y'\backslash V(M)$.

Rule 8 can be applied safely. Let (T',k) be the instance after applying Rule 8, and by Lemma 16, the instance (T',k) is yes if and only if (T,k) is yes. Thus, Rule 8 is safe. To use the above reduction rules exhaustively, we can get a $7k$ vertex kernel. Now, we prove the kernel of k-FVS-ST.

Theorem 4. The k-FVS-ST problem admits a kernel of size $7k$.

Proof. If we cannot find an expansion structure in B corresponding T, we can get a $7k$ vertex kernel, which has been discussed above. We now assume that there is an expansion structure $B=(X' \uplus Y', E', M)$. Then by Rule 8, $|Y'|=|X'|$. Since the minimum vertex cover is at most k, $|X'|+|Y\backslash Y'| \leqslant k$. we have $|Y| \leqslant k$ and the vertex of the resulting graph is at most $2m+k$, which implies that k-FVS-ST admits a $7k$ vertex kernel as $m \leqslant 3k$ by Lemma 15.

6 Conclusion

In this paper, we give some improved linear kernels in tournaments or sparse tournaments. The main result is the $3.5k$ kernel for the k-ATP-T problem. Since there is a $(3+\epsilon)k$ vertex kernel [29] for the edge-disjoint triangle packing problem in undirected graphs, there may be a $(3+\epsilon)k$ vertex kernel for the k-ATP-T problem. Actually, the crucial work is to bound the number of vertices in T_2. In tournaments, a vertex set that destroys all triangles is a feedback vertex set. Another interesting problem is whether there exist linear kernels for the vertex-disjoint triangle packing problem and the feedback vertex set problem in tournaments.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 62172446, 61872450).

References

1 Bafna V, Pevzner P A. Genome rearrangements and sorting by reversals. SIAM J Comput, 1996, 25: 272–289

2 El-Mabrouk N. Genome rearrangement by reversals and insertions/deletions of contiguous segments. In: Combinatorial Pattern Matching. Berlin: Springer, 2000. 222–234

3 Shao M F, Lin Y, Moret B. An exact algorithm to compute the DCJ distance for genomes with duplicate genes. In: Research in Computational Molecular Biology. Cham: Springer, 2014. 280–292

4 Bodlaender H L, Thomassé S, Yeo A. Kernel bounds for disjoint cycles and disjoint paths. Theor Comput Sci, 2011, 412: 4570–4578

5 Caprara A, Panconesi A, Rizzi R. Packing cycles in undirected graphs. J Algorithms, 2003, 48: 239–256

6 Fomin F V, Lokshtanov D, Misra N, et al. Quadratic upper bounds on the Erdős-Pósa property for a generalization of packing and covering cycles. J Graph Theor, 2013, 74: 417–424

7 Krivelevich M, Nutov Z, Salavatipour M R, et al. Approximation algorithms and hardness results for cycle packing problems. ACM Trans Algorithms, 2007, 3: 48

8 Slivkins A. Parameterized tractability of edge-disjoint paths on directed acyclic graphs. SIAM J Discrete Math, 2010, 24: 146–157

9 Holyer I. The NP-completeness of some edge-partition problems. SIAM J Comput, 1981, 10: 713–717

10 Caprara A, Rizzi R. Packing triangles in bounded degree graphs. Inf Processing Lett, 2002, 84: 175–180

11 Kann V. Maximum bounded 3-dimensional matching is MAX SNP-complete. Inf Processing Lett, 1991, 37: 27–35

12 Chlebík M, Chlebíková J. Approximation hardness for small occurrence instances of np-hard problems. In: Proceedings of the 5th Italian Conference on Algorithms and Complexity. Berlin: Springer, 2003. 152–164

13 Chen J, Kanj I A. Constrained minimum vertex cover in bipartite graphs: complexity and parameterized algorithms. J Comput Syst Sci, 2003, 67: 833–847

14 Liu Y L, Chen J, Huang J G. On book thickness parameterized by the vertex cover number. Sci China Inf Sci, 2022, 65: 140603

15 Wang J X, Li W J, Li S H, et al. On the parameterized vertex cover problem for graphs with perfect matching. Sci China Inf Sci, 2014, 57: 072107

16 Shi F, Wang J X, Yang Y F, et al. A fixed-parameter algorithm for the maximum agreement forest problem on multifurcating trees. Sci China Inf Sci, 2016, 59: 012102

17 Cygan M, Fomin F V, Kowalik L, et al. Parameterized Algorithms. Cham: Springer, 2015

18 Fomin F V, Lokshtanov D, Saurabh S, et al. Kernelization: Theory of Parameterized Preprocessing. Cambridge: Cambridge University Press, 2019

19 Demaine E D, Fomin F V, Hajiaghayi M, et al. Subexponential parameterized algorithms on bounded-genus graphs and H-minor-free graphs. J ACM, 2005, 52: 866–893

20 Dorn F, Fomin F V, Thilikos D M. Subexponential parameterized algorithms. Comput Sci Rev, 2008, 2: 29–39

21 Fomin F V, Villanger Y. Subexponential parameterized algorithm for minimum fill-in. SIAM J Comput, 2013, 42: 2197–2216

22 Dorn F, Fomin F V, Lokshtanov D, et al. Beyond bidimensionality: Parameterized subexponential algorithms on directed graphs. Inf Computation, 2013, 233: 60–70

23 Bandyapadhyay S, Lochet W, Lokshtanov D, et al. Subexponential parameterized algorithms for cut and cycle hitting problems on H-minor-free graphs. In: Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2022. 2063–2084

24 Lokshtanov D, Panolan F, Saurabh S, et al. Subexponential parameterized algorithms on disk graphs. In: Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2022. 2005–2031

25 Marx D, Misra P, Neuen D, et al. A framework for parameterized subexponential algorithms for generalized cycle hitting problems on planar graphs. In: Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2022. 2085–2127

26 Fan W F, He K, Li Q, et al. Graph algorithms: parallelization and scalability. Sci China Inf Sci, 2020, 63: 203101

27 Mathieson L, Prieto-Rodriguez E, Shaw P. Packing edge disjoint triangles: A parameterized view. In: Proceedings of the 1st International Workshop on Parameterized and Exact Computation. Bergen: Springer, 2004. 127–137

28 Yang Y J. Towards optimal kernel for edge-disjoint triangle packing. Inf Processing Lett, 2014, 114: 344–348

29 Lin W B, Xiao M Y. A $(3+\varepsilon)k$-vertex kernel for edge-disjoint triangle packing. Inf Processing Lett, 2019, 142: 20–26

30 Fisher D C, Lundgren J R, Merz S K, et al. The domination and competition graphs of a tournament. J Graph Theor, 1998, 29: 103–110

31 Butz M V, Sastry K, Goldberg D E. Strong, stable, and reliable fitness pressure in XCS due to tournament selection. Genet Program Evolvable Mach, 2005, 6: 53–77

32 Cui G, Wong M L, Lui H K. Machine learning for direct marketing response models: Bayesian networks with evolutionary programming. Manage Sci, 2006, 52: 597–612

33 de Jong K A, Schultz A C. Using experience-based learning in game playing. In: Proceedings of the 5th International Conference on Machine Learning, Ann Arbor, 1988. 284–290

34 Dudek G. Tournament searching method to feature selection problem. In: Proceedings of the 10th International Conference on Artifical Intelligence and Soft Computing, Zakopane, 2010. 437–444

35 Bessy S, Bougeret M, Krithika R, et al. Packing arc-disjoint cycles in tournaments. Algorithmica, 2021, 83: 1393–1420

36 Bessy S, Bougeret M, Thiebaut J. Triangle packing in (sparse) tournaments: Approximation and kernelization. In: Proceedings of the 25th Annual European Symposium on Algorithms, Vienna, 2017. 1–13

37 Fomin F V, Le T N, Lokshtanov D, et al. Subquadratic kernels for implicit 3-hitting set and 3-set packing problems. ACM Trans Algorithms, 2019, 15: 1–44