Triangle Packing in (Sparse) Tournaments: Approximation and Kernelization*

Stéphane Bessy[1], Marin Bougeret[2], and Jocelyn Thiebaut[3]

1 **Université de Montpellier – CNRS, LIRMM, Montpellier, France**
`bessy@lirmm.fr`
2 **Université de Montpellier – CNRS, LIRMM, Montpellier, France**
`bougeret@lirmm.fr`
3 **Université de Montpellier – CNRS, LIRMM, Montpellier, France**
`thiebaut@lirmm.fr`

Abstract

Given a tournament $\mathcal{T}$ and a positive integer k, the C_3-Packing-T problem asks if there exists a least k (vertex-)disjoint directed 3-cycles in $\mathcal{T}$. This is the dual problem in tournaments of the classical minimal feedback vertex set problem. Surprisingly C_3-Packing-T did not receive a lot of attention in the literature. We show that it does not admit a PTAS unless P=NP, even if we restrict the considered instances to sparse tournaments, that is tournaments with a feedback arc set (FAS) being a matching. Focusing on sparse tournaments we provide a $(1+\frac{6}{c-1})$ approximation algorithm for sparse tournaments having a linear representation where all the backward arcs have "length" at least c. Concerning kernelization, we show that C_3-Packing-T admits a kernel with $\mathcal{O}(m)$ vertices, where m is the size of a given feedback arc set. In particular, we derive a $\mathcal{O}(k)$ vertices kernel for C_3-Packing-T when restricted to sparse instances. On the negative size, we show that C_3-Packing-T does not admit a kernel of (total bit) size $\mathcal{O}(k^{2-\epsilon})$ unless NP $\subseteq$ coNP / Poly. The existence of a kernel in $\mathcal{O}(k)$ vertices for C_3-Packing-T remains an open question.

1998 ACM Subject Classification G.2.2 [Graph Theory] Graph Algorithms

Keywords and phrases Tournament, triangle packing, feedback arc set, approximation and parameterized algorithms

Digital Object Identifier 10.4230/LIPIcs.ESA.2017.14

1 Introduction and related work

Tournament

A tournament $\mathcal{T}$ on n vertices is an orientation of the edges of the complete undirected graph K_n. Thus, given a tournament $\mathcal{T} = (V, A)$, where $V = \{v_i, i \in [n]\}$, for each $i, j \in [n]$, either $v_iv_j \in A$ or $v_jv_i \in A$. A tournament $\mathcal{T}$ can alternatively be defined by an ordering $\sigma(\mathcal{T}) = (v_1, \ldots, v_n)$ of its vertices and a set of *backward arcs* $\overleftarrow{A}_\sigma(\mathcal{T})$ (which will be denoted $\overleftarrow{A}(\mathcal{T})$ as the considered ordering is not ambiguous), where each arc $a \in \overleftarrow{A}(\mathcal{T})$ is of the form $v_{i_1}v_{i_2}$ with $i_2 < i_1$. Indeed, given $\sigma(\mathcal{T})$ and $\overleftarrow{A}(\mathcal{T})$, we can define $V = \{v_i, i \in [n]\}$ and $A = \overleftarrow{A}(\mathcal{T}) \cup \overrightarrow{A}(\mathcal{T})$ where $\overrightarrow{A}(\mathcal{T}) = \{v_{i_1}v_{i_2} : (i_1 < i_2) \text{ and } v_{i_2}v_{i_1} \notin \overleftarrow{A}(\mathcal{T})\}$ is the set of forward arcs of $\mathcal{T}$ in the given ordering $\sigma(\mathcal{T})$. In the following, $(\sigma(\mathcal{T}), \overleftarrow{A}(\mathcal{T}))$ is called a *linear*

* An extended version of this paper is available at [4], `https://hal-lirmm.ccsd.cnrs.fr/lirmm-01550313`.

25th Annual European Symposium on Algorithms (ESA 2017).
Editors: Kirk Pruhs and Christian Sohler; Article No. 14; pp. 14:1–14:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

representation of the tournament $\mathcal{T}$. For a backward arc $e = v_jv_i$ of $\sigma(\mathcal{T})$ the *span value* of e is $j-i-1$. Then $\texttt{minspan}(\sigma(\mathcal{T}))$ (resp. $\texttt{maxspan}(\sigma(\mathcal{T}))$) is simply the minimum (resp. maximum) of the span values of the backward arcs of $\sigma(\mathcal{T})$.

A set $A' \subseteq A$ of arcs of $\mathcal{T}$ is a *feedback arc set* (or *FAS*) of $\mathcal{T}$ if every directed cycle of $\mathcal{T}$ contains at least one arc of A'. It is clear that for any linear representation $(\sigma(\mathcal{T}), \overleftarrow{A}(\mathcal{T}))$ of $\mathcal{T}$ the set $\overleftarrow{A}(\mathcal{T})$ is a FAS of $\mathcal{T}$. A tournament is *sparse* if it admits a FAS which is a matching. We denote by C_3-Packing-T the problem of packing the maximum number of vertex disjoint triangles in a given tournament, where a triangle is a directed 3-cycle. More formally, an input of C_3-Packing-T is a tournament $\mathcal{T}$, an output is a set (called a *triangle packing*) $S = \{t_i, i \in [|S|]\}$ where each t_i is a triangle and for any $i \neq j$ we have $V(t_i) \cap V(t_j) = \emptyset$, and the objective is to maximize $|S|$. We denote by $opt(\mathcal{T})$ the optimal value of $\mathcal{T}$. We denote by C_3-Perfect-Packing-T the decision problem associated to C_3-Packing-T where an input $\mathcal{T}$ is positive iff there is a triangle packing S such that $V(S) = V(\mathcal{T})$ (which is called a *perfect triangle packing*).

Related work

We refer the reader to the extended version of the paper [4] where we recall the definitions of the problems mentioned bellow as well as the standard definitions about parameterized complexity and approximation. A first natural related problem is 3-Set-Packing as we can reduce C_3-Packing-T to 3-Set-Packing by creating an hyperedge for each triangle.

Classical complexity / approximation. It is known that C_3-Packing-T is polynomial if the tournament does not contain the three forbidden sub-tournaments described in [5]. From the point of view of approximability, the best approximation algorithm is the $\frac{4}{3}+\epsilon$ approximation of [7] for 3-Set-Packing, implying the same result for K_3-packing and C_3-Packing-T. Concerning negative results, it is known [9] that even K_3-packing is MAX SNP-hard on graphs with maximum degree four. The related "dual" problems FAST and FVST received a lot of attention with for example the NP-hardness and PTAS for FAS in [6] and [12] respectively, and the $\frac{7}{3}$ approximation and inapproximability results for FVST in [13].

Kernelization. We precise that the implicitly considered parameter here is the size of the solution. There is a $\mathcal{O}(k^2)$ vertex kernel for K_3-packing in [14], and even a $\mathcal{O}(k^2)$ vertex kernel for 3-Set-Packing in [1], which is obtained by only removing vertices of the ground set. This remark is important as it directly implies a $\mathcal{O}(k^2)$ vertex kernel for C_3-Packing-T (see Section 4). Let us now turn to negative results. There is a whole line of research dedicated to finding lower bounds on the size of polynomial kernels. The main tool involved in these bounds is the weak composition introduced in [10] (whose definition is recalled in [4]). Weak composition allowed several almost tight lower bounds, with for examples the $\mathcal{O}(k^{d-\epsilon})$ for d-Set-Packing and $\mathcal{O}(k^{d-4-\epsilon})$ for K_d-packing of [10]. These results where improved in [8] to $\mathcal{O}(k^{d-\epsilon})$ for perfect d-Set-Packing, $\mathcal{O}(k^{d-1-\epsilon})$ for K_d-packing, and leading to $\mathcal{O}(k^{2-\epsilon})$ for perfect K_3-packing. Notice that negative results for the "perfect" version of problems (where parameter $k = \frac{n}{d}$, where d is the number of vertices of the packed structure) are stronger than for the classical version where k is arbitrary. Kernel lower bound for these "perfect" versions is sometimes referred as *sparsification lower bounds.*

Our contributions

Our objective is to study the approximability and kernelization of C_3-PACKING-T. On the approximation side, a natural question is a possible improvement of the $\frac{4}{3}+\epsilon$ approximation implied by 3-SET-PACKING. We show that, unlike FAST, C_3-PACKING-T does not admit a PTAS unless P=NP, even if the tournament is sparse. We point out that, surprisingly, we were not able to find any reference establishing a negative result for C_3-PACKING-T, even for the NP-hardness. As these results show that there is not much room for improving the approximation ratio, we focus on sparse tournaments and followed a different approach by looking for a condition that would allow ratio arbitrarily close to 1. In that spirit, we provide a $(1+\frac{6}{c-1})$ approximation algorithm for sparse tournaments having a linear representation with `minspan` at least c. Concerning kernelization, we complete the panorama of sparsification lower bounds of [11] by proving that C_3-PERFECT-PACKING-T does not admit a kernel of (total bit) size $\mathcal{O}(n^{2-\epsilon})$ unless NP $\subseteq$ coNP / Poly. This implies that C_3-PACKING-T does not admit a kernel of (total bit) size $\mathcal{O}(k^{2-\epsilon})$ unless NP $\subseteq$ coNP / Poly. We also prove that C_3-PACKING-T admits a kernel of $\mathcal{O}(m)$ vertices, where m is the size of a given FAS of the instance, and that C_3-PACKING-T restricted to sparse instances has a kernel in $\mathcal{O}(k)$ vertices (and so of total size bit $\mathcal{O}(k\log(k))$). The existence of a kernel in $\mathcal{O}(k)$ vertices for the general C_3-PACKING-T remains our main open question.

2 Specific notations and observations

Given a linear representation $(\sigma(\mathcal{T}),\overleftarrow{A}(\mathcal{T}))$ of a tournament $\mathcal{T}$, a triangle t in $\mathcal{T}$ is a triple $t=(v_{i_1},v_{i_2},v_{i_3})$ with $i_l<i_{l+1}$ such that either $v_{i_3}v_{i_1}\in\overleftarrow{A}(\mathcal{T})$, $v_{i_3}v_{i_2}\notin\overleftarrow{A}(\mathcal{T})$ and $v_{i_2}v_{i_1}\notin\overleftarrow{A}(\mathcal{T})$ (in this case we call t a *triangle with backward arc* $v_{i_3}v_{i_1}$), or $v_{i_3}v_{i_1}\notin\overleftarrow{A}(\mathcal{T})$, $v_{i_3}v_{i_2}\in\overleftarrow{A}(\mathcal{T})$ and $v_{i_2}v_{i_1}\in\overleftarrow{A}(\mathcal{T})$ (in this case we call t a *triangle with two backward arcs* $v_{i_3}v_{i_2}$ and $v_{i_2}v_{i_1}$). Given two tournaments $\mathcal{T}_1,\mathcal{T}_2$ defined by $\sigma(\mathcal{T}_l)$ and $\overleftarrow{A}(\mathcal{T}_l)$ we denote by $\mathcal{T}=\mathcal{T}_1\mathcal{T}_2$ the tournament called the concatenation of $\mathcal{T}_1$ and $\mathcal{T}_2$, where $\sigma(\mathcal{T})=\sigma(\mathcal{T}_1)\sigma(\mathcal{T}_2)$ is the concatenation of the two sequences, and $\overleftarrow{A}(\mathcal{T})=\overleftarrow{A}(\mathcal{T}_1)\cup\overleftarrow{A}(\mathcal{T}_2)$. Given a tournament $\mathcal{T}$ and a subset of vertices X, we denote by $\mathcal{T}\setminus X$ the tournament $\mathcal{T}[V(\mathcal{T})\setminus X]$ induced by vertices $V(\mathcal{T})\setminus X$, and we call this operation *removing* X *from* $\mathcal{T}$. Given an arc $a=uv$ we define $h(a)=v$ as the head of a and $t(a)=u$ as the tail of a. Given a linear representation $(V(\mathcal{T}),\overleftarrow{A}(\mathcal{T}))$ and an arc $a\in\overleftarrow{A}(\mathcal{T})$, we define $s(a)=\{v: h(a)<v<t(a)\}$ as the *span* of a. Notice that the span value of a is then exactly $|s(a)|$. Given a linear representation $(V(\mathcal{T}),\overleftarrow{A}(\mathcal{T}))$ and a vertex $v\in V(\mathcal{T})$, we define the degree of v by $d(v)=(a,b)$, where $a=|\{vu\in\overleftarrow{A}(\mathcal{T}):u<v\}|$ is called the *left degree* of v and $b=|\{uv\in\overleftarrow{A}(\mathcal{T}):u>v\}|$ is called the *right degree* of v. We also define $V_{(a,b)}=\{v\in V(\mathcal{T})|d(v)=(a,b)\}$. Given a set of pairwise distinct pairs D, we denote by C_3-PACKING-T^D the problem C_3-PACKING-T restricted to tournaments such that there exists a linear representation where $d(v)\in D$ for all v. Notice that when $D_M=\{(0,1),(1,0),(0,0)\}$, instances of C_3-PACKING-T^{D_M} are the sparse tournaments. Finally let us point out that it is easy to decide in polynomial time if a tournament is sparse or not, and if so, to give a linear representation whose FAS is a matching. The corresponding algorithm is detailed in [4]. Thus, in the following, when considering a sparse tournament we will assume that a linear ordering of it where backward arcs form a matching is also given. Finally, due to space limitations, the proofs of the results marked with '($\star$)' have been removed and are available in [4].

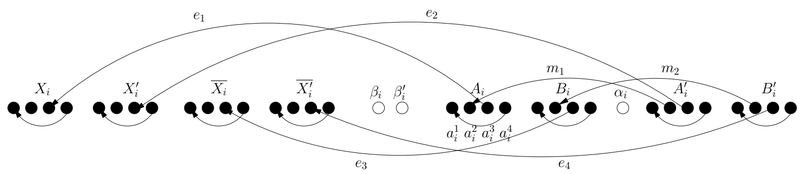

Figure 1 Example of a variable gadget L_i.

3 Approximation for sparse tournaments

3.1 APX-hardness for sparse tournaments

In this subsection we prove that C_3-PACKING-T^{D_M} is APX-hard by providing a L-reduction (see Definition in [4]) from Max 2-SAT(3), which is known to be APX-hard [2, 3]. Recall that in the MAX 2-SAT(3) problem each clause contains exactly 2 variables and each variable appears in at most 3 clauses (and at most twice positively and once negatively).

Definition of the reduction. Let $\mathcal{F}$ be an instance of MAX 2-SAT(3). In the following, we will denote by n the number of variables in $\mathcal{F}$ and m the number of clauses. Let $\{x_i, 1 \in [n]\}$ be the set of variables of $\mathcal{F}$ and $\{C_j, j \in [m]\}$ its set of clauses.

We now define a reduction f which maps an instance $\mathcal{F}$ of MAX 2-SAT(3) to an instance $\mathcal{T}$ of C_3-PACKING-T^{D_M}. For each variable x_i with $i \in [n]$, we create a tournament L_i as follows and we call it *variable gadget*. We refer the reader to Figure 1 where an example of variable gadget is depicted. Let $\sigma(L_i) = (X_i, X_i', \overline{X_i}, \overline{X_i}', \{\beta_i\}, \{\beta_i'\}\ , A_i, B_i, \{\alpha_i\}, A_i', B_i')$. We define $C = \{X_i, X_i', \overline{X_i}, \overline{X_i}', A_i, B_i, A_i', B_i'\}$. All sets of C have size 4. We denote $X_i = (x_i^1, x_i^2, x_i^3, x_i^4)$, and we extend the notation in a straightforward manner to the other others sets of C. Let us now define $\overleftarrow{A}(L_i)$. For each set of C, we add a backward arc whose head is the first element and the tail is the last element (for example for X_i we add the arc $x_i^4 x_i^1$). Then, we add to $\overleftarrow{A}(L_i)$ the set $\{e_1, e_2, e_3, e_4\}$ where $e_1 = x_i^3 a_i^3$, $e_2 = x_i'^3 a_i'^3$, $e_3 = \overline{x_i^3} b_i^3$, $e_4 = \overline{x_i'^3} b_i'^3$ and the set $\{m_1, m_2\}$ where $m_1 = a_i'^2 a_i^2$, $m_2 = b_i'^2 b_i^2$ called the two *medium arcs* of the variable gadget. This completes the description of tournament L_i. Let $L = L_1 \dots L_n$ be the concatenation of the L_i.

For each clause C_j with $j \in [1, m]$, we create a tournament K_j with ordering $\sigma(K_i) = (\theta_j, d_j^1, c_j^1, c_j^2, d_j^2)$ and $\overleftarrow{A}(K_i) = \{d_j^2 d_j^1\}$. We also define $K = K_1 \dots K_m$. Let us now define $\mathcal{T} = LK$. We add to $\overleftarrow{A}(\mathcal{T})$ the following backward arcs from $V(K)$ to $V(L)$. If $C_j = l_{i_1} \vee l_{i_2}$ is a clause in $\mathcal{F}$ then we add the arcs $c_j^1 v_{i_1}, c_j^2 v_{i_2}$ where v_{i_c} is the vertex in $\{x_{i_c}^2, x_{i_c}'^2, \overline{x_{i_c}^2}\}$ corresponding to l_{i_c}: if l_{i_c} is a positive occurrence of variable i_c we chose $v_{i_c} \in \{x_{i_c}^2, x_{i_c}'^2\}$, otherwise we chose $v_{i_c} = \overline{x_{i_c}^2}$. Moreover, we chose vertices v_{i_c} in such a way that for any $i \in [n]$, for each $v \in \{x_i^2, x_i'^2, \overline{x_i^2}\}$ there exists a unique arc $a \in \overleftarrow{A}(\mathcal{T})$ such that $h(a) = v$. This is always possible as each variable has at most two positive occurrences and one negative occurrence. Thus, x_i^2 represent the first positive occurrence of variable i, and $x_i'^2$ the second one. We refer the reader to Figure 2 where an example of the connection between variable and clause gadget is depicted.

Notice that vertices of $\overline{X_i'}$ are never linked to the clauses gadget. However, we need this set to keep the variable gadget symmetric so that setting x_i to true or false leads to the same number of triangles inside L_i. This completes the description of $\mathcal{T}$. Notice that the degree of any vertex is in $\{(0,1), (1,0), (0,0)\}$, and thus $\mathcal{T}$ is an instance of C_3-PACKING-T^{D_M}.

Let us now distinguish three different types of triangles in $\mathcal{T}$. A triangle $t = (v_1, v_2, v_3)$ of $\mathcal{T}$ is called an *outer* triangle iff $\exists j \in [m]$ such that $v_2 = \theta_j$ and $v_3 = c_j^l$ (implying that $v_1 \in V(L)$),

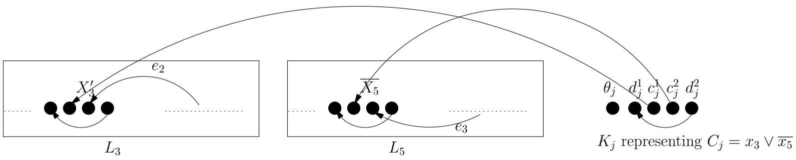

Figure 2 Example showing how a clause gadget is attached to variable gadgets.

variable inner iff $\exists i \in [n]$ such that $V(t) \subseteq V(L_i)$, and *clause inner* iff $\exists j \in [m]$ such that $V(t) \subseteq V(K_j)$. Notice that a triangle $t = (v_1, v_2, v_3)$ of $\mathcal{T}$ which is neither outer, variable or clause inner has necessarily $v_3 = c_j^l$ for some j, and $v_2 \neq \theta_j$ (v_2 could be in $V(L)$ or $V(K)$). In the following definition, for any $Y \in C$ (where $C = \{X_i, X'_i, \overline{X_i}, \overline{X_i}', A_i, B_i, A'_i, B'_i\}$) with $Y = (y^1, y^2, y^3, y^4)$, we define $t_Y^2 = (y^1, y^2, y^4)$ and $t_Y^3 = (y^1, y^3, y^4)$. For example, $t^2_{X'_i} = (x_i'^1, x_i'^2, x_i'^4)$. For any $i \in [n]$, we define P_i and $\overline{P_i}$, two sets of vertex disjoint variable inner triangles of $V(L_i)$, by:

- $P_i = \{t^3_{X_i}, t^3_{X'_i}, t^2_{\overline{X_i}}, t^2_{\overline{X'_i}}, t^3_{A_i}, t^2_{B_i}, t^3_{A'_i}, t^2_{B'_i}, (h(e_3), \beta_i, t(e_3)), (h(e_4), \beta'_i, t(e_4)), (h(m_1), \alpha_i, t(m_1))\}$
- $\overline{P_i} = \{t^2_{X_i}, t^2_{X'_i}, t^3_{\overline{X_i}}, t^3_{\overline{X'_i}}, t^2_{A_i}, t^3_{B_i}, t^2_{A'_i}, t^3_{B'_i}, (h(e_1), \beta_i, t(e_1)), (h(e_2), \beta'_i, t(e_2)), (h(m_2), \alpha_i, t(m_2))\}$

Notice that P_i (resp. $\overline{P_i}$) uses all vertices of L_i except $\{x_i^2, x_i'^2\}$ (resp. $\{\overline{x_i^2}, \overline{x_i'^2}\}$). For any $j \in [m]$, and $x \in [2]$ we define the set of clause inner triangle of K_j, that is $Q_j^x = \{(d_j^1, c_j^x, d_j^2)\}$.

Informally, setting variable x_i to true corresponds to create the 11 triangles of P_i in L_i (as leaving vertices $\{x_i^2, x_i^{2'}\}$ available allows to create outer triangles corresponding to satisfied clauses), and setting it to false corresponds to create the 11 triangles of $\overline{P_i}$. Satisfying a clause j using its x^{th} literal (represented by a vertex $v \in V(L)$) corresponds to create triangle in Q_j^{3-x} as it leaves c_j^x available to create the triangle (v, θ_j, c_j^x). Our final objective (in Lemma 4) is to prove that satisfying k clauses is equivalent to find $11n+m+k$ vertex disjoint triangles.

Restructuration lemmas. Given a solution S, let $I_i^L = \{t \in S : V(t) \subseteq V(L_i)\}$, $I_j^K = \{t \in S : V(t) \subseteq V(K_j)\}$, $I^L = \cup_{i \in [n]} I_i^L$ be the set of variable inner triangles of S, $I^K = \cup_{j \in [m]} I_j^K$ be the set of clause inner triangles of S, and $O = \{t \in S$ t is an outer triangle $\}$ be the set of outer triangles of S. Notice that *a priori* I^L, I^K, O does not necessarily form a partition of S. However, we will show in the next lemmas how to restructure S such that I^L, I^K, O becomes a partition.

▶ **Lemma 1** ($\star$). *For any S we can compute in polynomial time a solution $S' = \{t'_l, l \in [k]\}$ such that $|S'| \geq |S|$ and for all $j \in [m]$ there exists $x \in [2]$ such that $I_j'^K = Q_j^x$ and*

- *either S' does not use any other vertex of K_j ($V(S') \cap V(K_j) = V(Q_j^x)$)*
- *either S' contains an outer triangle $t'_l = (v, \theta_j, c_j^{3-x})$ with $v \in V(L)$ (implying $V(S') \cap V(K_j) = V(K_j)$)*

▶ **Corollary 2.** *For any S we can compute in polynomial time a solution S' such that $|S'| \geq |S|$, and S' only contains outer, variable inner, and clause inner triangles. Indeed, in the solution S' of Lemma 1, given any $t \in S'$, either $V(t)$ intersects $V(K_j)$ for some j and then t is an outer or a clause inner triangle, or $V(t) \subseteq V(L_i)$ for $i \in [n]$ as there is no backward arc uv with $u \in V(L_{i_1})$ and $v \in V(L_{i_2})$ with $i_1 \neq i_2$.*

▶ **Lemma 3** ($\star$). *For any S we can compute in polynomial time a solution S' such that $|S'| \geq |S|$, S' satisfies Lemma 1, and for every $i \in [n]$, $I_i'^L = P_i$ or $I_i'^L = \overline{P_i}$.*

Proof of the L-reduction. We are now ready to prove the main lemma (recall that f is the reduction from Max 2-SAT(3) to C_3-Packing-T^{D_M} described in Section 3.1), and also the main theorem of the section.

▶ **Lemma 4.** *Let $\mathcal{F}$ be an instance of* Max 2-SAT(3). *For any k, there exists an assignment a of $\mathcal{F}$ satisfying at least k clauses if and only if there exists a solution S of $f(\mathcal{F})$ with $|S| \geq 11n + m + k$, where n and m are respectively is the number of variables and clauses in $\mathcal{F}$. Moreover, in the $\Leftarrow$ direction, assignment a can be computed from S in polynomial time.*

Proof. For any $i \in [n]$, let $A_i = P_i$ if x_i is set to true in a, and $A_i = \overline{P_i}$ otherwise. We first add to S the set $\cup_{i\in[n]}A_i$. Then, let $\{C_{j_l}, l \in [k]\}$ be k clauses satisfied by a. For any $l \in [k]$, let i_l be the index of a literal satisfying C_{j_l}, let $x \in [2]$ such that $c^x_{j_l}$ corresponds to this literal, and let $Z_l = \{x^2_{i_l}, x'^2_{i_l}\}$ if literal i_l is positive, and $Z_l = \{\overline{x^2_{i_l}}\}$ otherwise. For any $j \in [m]$, if $j = i_l$ for some l (meaning that j corresponds to a satisfied clause), we add to S the triangle in Q^{3-x}_j, and otherwise we arbitrarily add the triangle Q^1_j. Finally, for any $l \in [k]$ we add to S triangle $t_l = (y_l, \theta_{j_l}, c^x_{j_l})$ where $y_l \in Z_l$ is such that y_l is not already used in another triangle. Notice that such an y_l always exists as triangles of $A_i, i \in [n]$ do not intersect Z_l (by definition of the A_i), and as there are at most two clauses that are true due to positive literal, and one clause that is true due to a negative literal. Thus, S has $11n + m + k$ vertex disjoint triangles.

Conversely, let S a solution of $f(\mathcal{F})$ with $|S| \geq 11n + m + k$. By Lemma 3 we can construct in polynomial time a solution S' from S such that $|S'| \geq |S|$, S' only contains outer, variable or clause inner triangles, for each $j \in [m]$ there exists $x \in [2]$ such that $I'^K_j = Q^x_j$, and for each $i \in [n], I'^L_i = P_i$ or $I'^L_i = \overline{P_i}$. This implies that the $k' \geq k$ remaining triangles must be outer triangles. Let $\{t'_l, l \in [k']\}$ be these k' outer triangles with $t'_l = (y_l, \theta_{j_l}, c^{x_l}_{j_l})$ Let us define the following assignation a: for each $i \in [n]$, we set x_i to true if $I'^L_i = P_i$, and false otherwise. This implies that a satisfies at least clauses $\{C_{j_l}, l \in [k']\}$. ◀

▶ **Theorem 5.** C_3-Packing-T^{D_M} *is APX-hard, and thus does not admit a PTAS unless P = NP.*

Proof. Let us check that Lemma 4 implies a L-reduction (whose definition is recalled in [4]). Let OPT_1 (resp. OPT_2) be the optimal value of $\mathcal{F}$ (resp. $f(\mathcal{F})$). Notice that Lemma 4 implies that $OPT_2 = OPT_1 + 11n + m$. It is known that $OPT_1 \geq \frac{3}{4}m$ (where m is the number of clauses of $\mathcal{F}$). As $n \leq m$ (each variable has at least one positive and one negative occurrence), we get $OPT_2 = OPT_1 + 11n + m \leq \alpha OPT_1$ for an appropriate constant α, and thus point (a) of the definition is verified. Then, given a solution S' of $f(\mathcal{F})$, according to Lemma 4 we can construct in polynomial time an assignment a satisfying $c(a)$ clauses with $c(a) \geq S' - 11n - m$. Thus, the inequality (b) of the Definition of a L-reduction with $\beta = 1$ becomes $OPT_1 - c(a) \leq OPT_2 - S' = OPT_1 + 11n + m - S'$, which is true. ◀

Reduction of Theorem 5 does not imply the NP-hardness of C_3-Perfect-Packing-T as there remain some unused vertices. However, it is straightforward to adapt the reduction by adding backward arcs whose head (resp. tail) are before (resp. after) $\mathcal{T}$ to consume the remaining vertices. This leads to the following result.

▶ **Theorem 6** ($\star$). C_3-Perfect-Packing-T^{D_M} *is NP-hard.*

To establish the kernel lower bound of Section 4, we also need the NP-hardness of C_3-Perfect-Packing-T where instances have a slightly simpler structure (to the price of losing the property that there exists a FAS which is a matching).

▶ **Theorem 7** ($\star$). *C_3-Perfect-Packing-T remains NP-hard even restricted to tournaments $\mathcal{T}$ admitting the following linear ordering.*

- *$\mathcal{T} = LK$ where L and K are two tournaments*
- *tournaments L and K are "fixed":*
  - *$K = K_1 \dots K_m$ for some m, where for each $j \in [m]$ we have $V(K_j) = (\theta_j, c_j)$*
  - *$L = R_1 L_1 \dots L_n R_2$, where each L_i has is a copy of the variable gadget of Section 3.1, $R_i = \{r_i^l, l \in [n']\}$ where $n' = 2n - m$, and in addition $\overleftarrow{L}$ also contains $R = \{(r_2^l r_1^l), l \in [n']\}$ which are called the dummy arcs.*

3.2 $(1 + \frac{6}{c-1})$-approximation when backward arcs have large minspan

Given a set of pairwise distinct pairs D and an integer c, we denote by C_3-Packing-$\text{T}^D_{\geq c}$ the problem C_3-Packing-T^D restricted to tournaments such that there exists a linear representation of minspan at least c and where $d(v) \in D$ for all v. In all this section we consider an instance $\mathcal{T}$ of C_3-Packing-$\text{T}^{D_M}_{\geq c}$ with a given linear ordering $(V(\mathcal{T}), \overleftarrow{A}(\mathcal{T}))$ of minspan at least c and whose degrees belong to D_M. The motivation for studying the approximability of this special case comes from the situation of MAX-SAT(c) where the approximability becomes easier as c grows, as the derandomized uniform assignment provides a $\frac{2^c}{2^c-1}$ approximation algorithm. Somehow, one could claim that MAX-SAT(c) becomes easy to approximate for large c as there are many ways to satisfy a given clause. As the same intuition applies for tournaments admitting an ordering with large minspan (as there are $c-1$ different ways to use a given backward in a triangle), our objective was to find a polynomial approximation algorithm whose ratio tends to 1 when c increases.

Let us now define algorithm Φ. We define a bipartite graph $G = (V_1, V_2, E)$ with $V_1 = \{v_a^1 : a \in \overleftarrow{A}(\mathcal{T})\}$ and $V_2 = \{v_l^2 : v_l \in V_{(0,0)}\}$. Thus to each backward arc we associate a vertex in V_1 and to each vertex v_l with $d(v_l) = (0,0)$ we associate a vertex in V_2. Then $\{v_a^1, v_l^2\} \in E$ iff $(h(a), v_l, t(a))$ is a triangle in $\mathcal{T}$.

In phase 1, Φ computes a maximum matching $M = \{e_l, l \in [|M|]\}$ in G. For every $e_l = \{v_{ij}^1, v_l^2\} \in M$ create a triangle $t_l^1 = (v_j, v_l, v_i)$. Let $S^1 = \{t_l^1, l \in [|M|]\}$. Notice that triangles of S^1 are vertex disjoint. Let us now turn to phase 2. Let $\mathcal{T}^2$ be the tournament $\mathcal{T}$ where we removed all vertices $V(S^1)$. Let $(V(\mathcal{T}^2), \overleftarrow{A}(\mathcal{T}^2))$ be the linear ordering of $\mathcal{T}^2$ obtained by removing $V(S^1)$ in $(V(\mathcal{T}), \overleftarrow{A}(\mathcal{T}))$. We say that three distinct backward edges $\{a_1, a_2, a_3\} \subseteq \overleftarrow{A}(\mathcal{T}^2)$ can be packed into triangles t_1 and t_2 iff $V(\{t_1, t_2\}) = V(\{a_1, a_2, a_3\})$ and the t_i are vertex disjoint. For example, if $h(a_1) < h(a_2) < t(a_1) < h(a_3) < t(a_2) < t(a_3)$, then $\{a_1, a_2, a_3\}$ can be packed into $(h(a_1), h(a_2), t(a_1))$ and $(h(a_3), t(a_2), t(a_3))$ (recall that when $\overleftarrow{A}(\mathcal{T})$ form a matching, (u, v, w) is triangle iff $wu \in \overleftarrow{A}(\mathcal{T})$ and $u < v < w$), and if $h(a_1) < h(a_2) < t(a_2) < h(a_3) < t(a_3) < t(a_1)$, then $\{a_1, a_2, a_3\}$ cannot be packed into two triangles. In phase 2, while it is possible, Φ finds a triplet of arcs of $Y \subseteq \overleftarrow{A}(\mathcal{T}^2)$ that can be packed into triangles, create the two corresponding triangles, and remove $V(Y)$. Let S^2 be the triangle created in phase 2 and let $S = S^1 \cup S^2$.

▶ **Observation 8.** *For any $a \in \overleftarrow{A}(\mathcal{T})$, either $V(a) \subseteq V(S)$ or $V(a) \cap V(S) = \emptyset$. Equivalently, no backward arc has one endpoint in $V(S)$ and the other outside $V(S)$.*

According to Observation 8, we can partition $\overleftarrow{A}(\mathcal{T}) = \overleftarrow{A}_0 \cup \overleftarrow{A}_1 \cup \overleftarrow{A}_2$, where for $i \in \{1, 2\}$, $\overleftarrow{A}^i = \{a \in \overleftarrow{A}(\mathcal{T}) : V(a) \subseteq V(S^i)$ is the set of arcs used in phase i, and $\overleftarrow{A}_0 =_{def} \{b_i, i \in [x]\}$ are the remaining unused arcs. Let $\overleftarrow{A}_\Phi = \overleftarrow{A}_1 \cup \overleftarrow{A}_2$, $m_i = |\overleftarrow{A}_i|$, $m = m_0 + m_1 + m_2$ and $m_\Phi = m_1 + m_2$ the number of arcs (entirely) consumed by Φ. To prove the $1 + \frac{6}{c-1}$ desired approximation ratio, we will first prove in Lemma 9 that Φ uses at most all the arcs

($m_A \geq (1-\epsilon(c))m$), and in Theorem 10 that the number of triangles made with these arcs is "optimal". Notice that the latter condition is mandatory as if Φ used its m_Φ arcs to only create $\frac{2}{3}(m_\Phi)$ triangles in phase 2 instead of creating $m' \approx m_\Phi$ triangle with m' backward arcs and m' vertices of degree $(0,0)$, we would have a $\frac{3}{2}$ approximation ratio.

▶ **Lemma 9** ($\star$). *For any $c \geq 2$, $m_\Phi \geq (1-\frac{6}{c+5})m$*

▶ **Theorem 10.** *For any $c \geq 2$, Φ is a polynomial $(1+\frac{6}{c-1})$ approximation algorithm for* C_3-Packing-T$^{D_M}_{\geq c}$.

Proof. Let OPT be an optimal solution. Let us define $OPT_i \subseteq OPT$ and $\overleftarrow{A}^*_i \subseteq \overleftarrow{A}(\mathcal{T})$ as follows. Let $t=(u,v,w) \in OPT$. As the FAS of the instance is a matching, we know that $wu \in \overleftarrow{A}(\mathcal{T})$ as we cannot have a triangle with two backward arcs. If $d(v)=(0,0)$ then we add t to OPT_1 and wu to $\overleftarrow{A}^*_1$. Otherwise, let v' be the other endpoint of the unique arc a containing v. If $v' \notin V(OPT)$, then we add t to OPT_3 and $\{wu, a\}$ to $\overleftarrow{A}^*_3$. Otherwise, let $t' \in OPT$ such that $v' \in V(t')$. As the FAS of the instance is a matching we know that v' is the middle point of t', or more formally that $t'=(u',v',w')$ with $u'w' \in \overleftarrow{A}(\mathcal{T})$. We add $\{t,t'\}$ to OPT_2 and $\{wu, a, w'u'\}$ to $\overleftarrow{A}^*_2$. Notice that the OPT_i form a partition of OPT, and that the $\overleftarrow{A}^*_i$ have pairwise empty intersection, implying $|\overleftarrow{A}^*_1|+|\overleftarrow{A}^*_2|+|\overleftarrow{A}^*_3| \leq m$. Notice also that as triangles of OPT_1 correspond to a matching of size $|OPT_1|$ in the bipartite graph defined in phase 1 of algorithm Φ, we have $|OPT_1| = |\overleftarrow{A}^*_1| \leq |\overleftarrow{A}_1|$.

Putting pieces together we get (recall that S is the solution computed by Φ) $|OPT| = |OPT_1|+|OPT_2|+|OPT_3| = |\overleftarrow{A}^*_1| + \frac{2}{3}|\overleftarrow{A}^*_2| + \frac{1}{2}|\overleftarrow{A}^*_3| \leq |\overleftarrow{A}^*_1| + \frac{2}{3}(|\overleftarrow{A}^*_2|+|\overleftarrow{A}^*_3|) \leq |\overleftarrow{A}^*_1| + \frac{2}{3}(m - |\overleftarrow{A}^*_1|) \leq \frac{1}{3}|\overleftarrow{A}_1| + \frac{2}{3}m$ and $|S| = |S^1|+|S^2| = |\overleftarrow{A}_1| + \frac{2}{3}|\overleftarrow{A}_2| \geq |\overleftarrow{A}_1| + \frac{2}{3}((1-\frac{6}{c+5})m - |\overleftarrow{A}_1|) = \frac{1}{3}|\overleftarrow{A}_1| + \frac{2}{3}(1-\frac{6}{c+5})m$ which implies the desired ratio. ◀

4 Kernelization

In all this section we consider the decision problem C_3-Packing-T parameterized by the size of the solution. Thus, an input is a pair $I=(\mathcal{T},k)$ and we say that I is positive iff there exists a set of k vertex disjoint triangles in $\mathcal{T}$.

4.1 Positive results for sparse instances

Observe first that the kernel in $\mathcal{O}(k^2)$ vertices for 3-Set Packing of [1] directly implies a kernel in $\mathcal{O}(k^2)$ vertices for C_3-Packing-T. Indeed, given an instance $(\mathcal{T}=(V,A),k)$ of C_3-Packing-T, we create an instance $(I'=(V,C),k)$ of 3-Set Packing by creating an hyperedge $c \in C$ for each triangle of $\mathcal{T}$. Then, as the kernel of [1] only removes vertices, it outputs an induced instance $(\overline{I'} = I'[V'],k')$ of I with $V' \subseteq V$, and thus this induced instance can be interpreted as a subtournament, and the corresponding instance $(\mathcal{T}[V'],k')$ is an equivalent tournament with $\mathcal{O}(k^2)$ vertices.

As shown in the next theorem, as we could expect it is also possible to have kernel bounded by the number of backward arcs.

▶ **Theorem 11.** C_3-Packing-T *admits a polynomial kernel with $\mathcal{O}(m)$ vertices, where m is the number of arcs in a given FAS of the input.*

Proof. Let $I=(\mathcal{T},k)$ be an input of the decision problem associated to C_3-Packing-T. Observe first that we can build in polynomial time a linear ordering $\sigma(\mathcal{T})$ whose backward arcs $\overleftarrow{A}(\mathcal{T})$ correspond to the given FAS. We will obtain the kernel by removing useless vertices of degree $(0,0)$. Let us define a bipartite graph $G = (V_1, V_2, E)$ with $V_1 = \{v_a^1 : a \in \overleftarrow{A}(\mathcal{T})\}$ and $V_2 = \{v_l^2 : v_l \in V_{(0,0)}\}$. Thus, to each backward arc we associate a vertex in V_1 and to each vertex v_l with $d(v_l) = (0,0)$ we associate a vertex in V_2. Then, $\{v_a^1, v_l^2\} \in E$ iff $(h(a), v_l, t(a))$ is a triangle in $\mathcal{T}$. By Hall's theorem, we can in polynomial time partition V_1 and V_2 into $V_1 = A_1 \cup A_2$, $V_2 = B_0 \cup B_1 \cup B_2$ such that $N(A_2) \subseteq B_2$, $|B_2| \leq |A_2|$, and there is a perfect matching between vertices of A_1 and B_1 (B_0 is simply defined by $B_0 = V_2 \setminus (B_1 \cup B_2)$).

For any $i, 0 \leq i \leq 2$, let $X_i = \{v_l \in V_{(0,0)} : v_l^2 \in B_i\}$ be the vertices of $\mathcal{T}$ corresponding to B_i. Let $V_{\neq(0,0)} = V(\mathcal{T}) \setminus V_{(0,0)}$. Notice that $|V_{\neq(0,0)}| \leq 2m$. We define $\mathcal{T}' = \mathcal{T}[V_{\neq(0,0)} \cup X_1 \cup X_2]$ the sub-tournament obtained from $\mathcal{T}$ by removing vertices of X_0, and $I' = (\mathcal{T}', k)$. We point out that this definition of $\mathcal{T}'$ is similar to the final step of the kernel of [1] as our partition of V_1 and V_2 (more precisely $(A_1, B_0 \cup B_1)$) corresponds in fact to the crown decomposition of [1]. Observe that $|V(\mathcal{T}')| \leq 2m + |A_1| + |A_2| \leq 3m$, implying the desired bound of the number of vertices of the kernel.

It remains to prove that I and I' are equivalent. Let $k \in \mathbb{N}$, and let us prove that there exists a solution S of $\mathcal{T}$ with $|S| \geq k$ iff there exists a solution S' of $\mathcal{T}'$ with $|S'| \geq k$. Observe that the $\Leftarrow$ direction is obvious as $\mathcal{T}'$ is a subtournament of $\mathcal{T}$. Let us now prove the $\Rightarrow$ direction. Let S be a solution of $\mathcal{T}$ with $|S| \geq k$. Let $S = S_{(0,0)} \cup S_1$ with $S_{(0,0)} = \{t \in S : t = (h(a), v, t(a))$ with $v \in V_{(0,0)}, a \in \overleftarrow{A}(\mathcal{T})\}$ and $S_1 = S \setminus S_{(0,0)}$. Observe that $V(S_1) \cap V_{(0,0)} = \emptyset$, implying $V(S_1) \subseteq V_{\neq(0,0)}$. For any $i \in [2]$, let $S^i_{(0,0)} = \{t \in S_{(0,0)} : t = (h(a), v, t(a))$ with $v \in V_{(0,0)}, v_a^1 \in A_i\}$ be a partition of $S_{(0,0)}$. We define $S' = S_1 \cup S^2_{(0,0)} \cup S'^1_{(0,0)}$, where $S'^1_{(0,0)}$ is defined as follows. For any $v_a^1 \in A_1$, let $v^2_{\mu(a)} \in B_1$ be the vertex associated to v_a^1 in the (A_1, B_1) matching. To any triangle $t = (h(a), v, t(a)) \in S^1_{(0,0)}$ we associate a triangle $f(t) = (h(a), v_{\mu(a)}, t(a)) \in S'^1_{(0,0)}$, where by definition $v_{\mu(a)} \in X_1$. For the sake of uniformity we also say that any $t \in S_1 \cup S^2_{(0,0)}$ is associated to $f(t) = t$.

Let us now verify that triangles of S' are vertex disjoint by verifying that triangles of $S'^1_{(0,0)}$ do not intersect another triangle of S'. Let $f(t) = (h(a), v_{\mu(a)}, t(a)) \in S'^1_{(0,0)}$. Observe that $h(a)$ and $t(a)$ cannot belong to any other triangle $f(t')$ of S' as for any $f(t'') \in S'$, $V(f(t'')) \cap V_{\neq(0,0)} = V(t'') \cap V_{\neq(0,0)}$ (remember that we use the same notation $V_{\neq(0,0)}$ to denote vertices of degree $(0,0)$ in $\mathcal{T}$ and $\mathcal{T}'$). Let us now consider $v_{\mu(a)}$. For any $f(t') \in S_1$, as $V(f(t')) \cap V_{(0,0)} = \emptyset$ we have $v_{\mu(a)} \notin V(f(t'))$. For any $f(t') = (h(a'), v_l, t(a')) \in S^2_{(0,0)}$, we know by definition that $v^1_{a'} \in A_2$, implying that $v_l^2 \in B_2$ (and $v_l \in X_2$) as $N(A_2) \subseteq B_2$ and thus that $v_l \neq v_{\mu(a)}$. Finally, for any $f(t') = (h(a'), v_l, t(a')) \in S'^1_{(0,0)}$, we know that $v_l = v_{\mu(a')}$, where $a \neq a'$, leading to $v_l \neq v_{\mu(a)}$ as μ is a matching. ◀

Using the previous result we can provide a $\mathcal{O}(k)$ vertices kernel for C_3-Packing-T restricted to sparse tournaments.

▶ **Theorem 12** (⋆). *C_3-Packing-T restricted to sparse tournaments admits a polynomial kernel with $\mathcal{O}(k)$ vertices, where k is the size of the solution.*

4.2 No (generalised) kernel in $\mathcal{O}(k^{2-\epsilon})$

In this section we provide an OR-cross composition (see [4] where we recall the definition) from C_3-Perfect-Packing-T restricted to instances of Theorem 7 to C_3-Perfect-Packing-T parameterized by the total number of vertices.

Definition of the instance selector. The next lemma build a special tournament, called an *instance selector* that will be useful to design the cross composition.

▶ **Lemma 13** ($\star$). *For any $\gamma = 2^{\gamma'}$ and ω we can construct in polynomial time (in γ and ω) a tournament $P_{\omega,\gamma}$ such that*

- *there exists γ subsets of ω vertices $X^i = \{x^i_j : j \in [\omega]\}$, that we call the distinguished set of vertices, such that*
  - *the X^i have pairwise empty intersection*
  - *for any $i \in [\gamma]$, there exists a packing S of triangles of $P_{\omega,\gamma}$ such that $V(P_{\omega,\gamma}) \setminus V(S) = X^i$ (using this packing of $P_{\omega,\gamma}$ corresponds to select instance i)*
  - *for any packing S of triangles of $P_{\omega,\gamma}$ with $|V(S)| = |V(P_{\omega,\gamma})| - \omega$ there exists $i \in [\gamma]$ such that $V(P_{\omega,\gamma}) \setminus V(S) \subseteq X^i$*
- *$|V(P_{\omega,\gamma})| = \mathcal{O}(\omega\gamma)$.*

Definition of the reduction. We suppose given a family of t instances $F = \{\mathcal{I}_l, l \in [t]\}$ of C_3-Perfect-Packing-T restricted to instances of Theorem 7 where $\mathcal{I}_l$ asks if there is a perfect packing in $\mathcal{T}_l = L_l K_l$. We chose our equivalence relation of the cross-composition such that there exist n and m such that for any $l \in [t]$ we have $|V(L_l)| = n$ and $|V(K_l)| = m$. We can also copy some of the t instances such that t is a square number and $g = \sqrt{t}$ is a power of two. We reorganize our instances into $F = \{\mathcal{I}_{(p,q)} : 1 \leq p, q \leq g\}$ where $\mathcal{I}_{(p,q)}$ asks if there is a perfect packing in $\mathcal{T}_{(p,q)} = L_p K_q$. Remember that according to Theorem 7, all the L_p are equals, and all the K_q are equals. We point out that the idea of using a problem on "bipartite" instances to allow encoding t instances on a "meta" bipartite graph $G = (A, B)$ (with $A = \{A_i, i \in \sqrt{t}\}$, $B = \{B_i, i \in \sqrt{t}\}$) such that each instance p, q is encoded in the graph induced by $G[A_i \cup B_i]$ comes from [8]. We refer the reader to Figure 3 which represents the different parts of the tournament. We define a tournament $G = LM_G\tilde{L}\tilde{M}_G P_{(n,g)}$, where $L = L_1 \dots L_g$, $\tilde{M}_G$ is a set of n vertices of degree $(0, 0)$, M_G is a set of $(g-1)n$ vertices of degree $(0, 0)$, $\tilde{L} = \tilde{L}_1 \dots \tilde{L}_g$ where each $\tilde{L}_p$ is a set of n vertices, and $P_{(n,g)}$ is a copy of the instance selector of Lemma 13. Then, for every $p \in [g]$ we add to G all the possible n^2 backward arcs going from $\tilde{L}_p$ to L_p. Finally, for every distinguished set X^p of $P_{(n,g)}$ (see in Lemma 13), we add all the possible n^2 backward arcs from X^p to $\tilde{L}_p$.

Now, in a symmetric way we define a tournament $D = KM_D\tilde{K}\tilde{M}_D P'_{(m,g)}$, where $K = K_1 \dots K_g$, $\tilde{M}_D$ is a set of m vertices of degree $(0, 0)$, M_D is a set of $(g-1)m$ vertices of degree $(0, 0)$, $\tilde{K} = \tilde{K}_1 \dots \tilde{K}_g$ where each $\tilde{K}_q$ is a set of m vertices, and $P'_{(m,g)}$ is a copy of the instance selector of Lemma 13. Then, for every $q \in [g]$ we add to G all the m^2 possible backward arcs going from $\tilde{K}_p$ to K_p. For every distinguished set X'^q of $P'_{(m,g)}$ we also add all the possible m^2 backward arcs from X'^q to $\tilde{K}_q$. Finally, we define $\mathcal{T} = GD$. Let us add some backward arcs from D to G. For any p and q with $1 \leq p, q \leq g$, we add backward arcs from K_q to L_p such that $\mathcal{T}[K_q L_p]$ corresponds to $\mathcal{T}_{(p,q)}$. Notice that this is possible as for any fixed p, all the $\mathcal{T}_{(p,q)}, q \in [g]$ have the same left part L_p, and the same goes for any fixed right part.

Restructuration lemmas. Given a set of triangles S we define $S_{\subseteq P'} = \{t \in S | V(t) \subseteq P'_{(m,g)}\}$, $S_{\subseteq P} = \{t \in S : V(t) \subseteq P_{(n,g)}\}$, $S_{\tilde{M}_D} = \{t \in S : V(t)$ intersects $\tilde{K}$, $\tilde{M}_D$ and $P'_{m,g}\}$, $S_{M_D} = \{t \in S : V(t)$ intersects K, M_D and $\tilde{K}\}$, $S_{\tilde{M}_G} = \{t \in S : V(t)$ intersects $\tilde{L}$, $\tilde{M}_G$ and $P_{n,g}\}$, $S_{M_G} = \{t \in S : V(t)$ intersects L, M_G and $\tilde{L}\}$, $S_D = \{t \in S : V(t) \subseteq V(D)\}$, $S_G = \{t \in S : V(t) \subseteq V(G)\}$, and $S_{GD} = \{t \in S : V(t)$ intersects $V(G)$ and $V(D)\}$. Notice that S_G, S_{GD}, S_D is a partition of S.

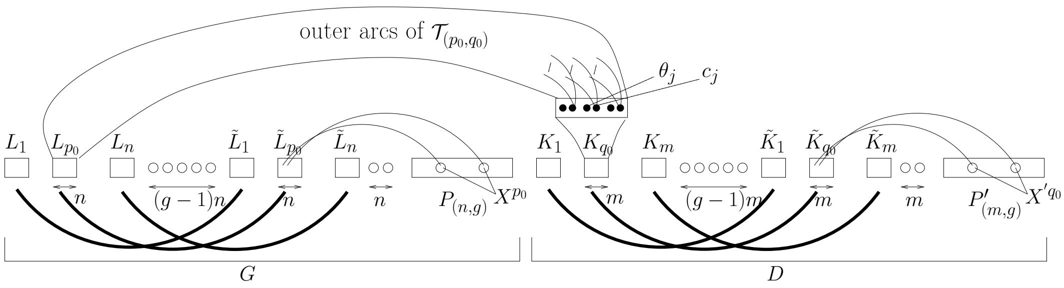

Figure 3 A example of the weak composition. All depicted arcs are backward arcs. Bold arcs represents the n^2 (or m^2) possible arcs between the two groups.

▶ **Claim 14.** *If there exists a perfect packing S of $\mathcal{T}$, then $|S_{\tilde{M}_D}| = m$ and $|S_{M_D}| = (g-1)m$. This implies that $V(S_{\tilde{M}_D} \cup S_{M_D}) \cap V(\tilde{K}) = V(\tilde{K})$, meaning that the vertices of $\tilde{K}$ are entirely used by $S_{\tilde{M}_D} \cup S_{M_D}$.*

Proof. We have $|S_{\tilde{M}_D}| \leq m$ since $|\tilde{M}_D| = m$. We obtain the equality since the vertices of $\tilde{M}_D$ only lie in the span of backward arcs from $P'_{m,g}$ to $\tilde{K}$, and they are not the head or the tail of a backward arc in $\mathcal{T}$. Thus, the only way to use vertices of $\tilde{M}_D$ is to create triangles in $S_{\tilde{M}_D}$, implying $|S_{\tilde{M}_D}| \geq m$. Using the same kind of arguments we also get $|S_{M_D}| = (g-1)m$. As $|V(\tilde{K})| = gm$ we get the last part of the claim. ◀

▶ **Claim 15.** *If there exists a perfect packing S of $\mathcal{T}$, then there exists $q_0 \in [g]$ such that $\tilde{K}_S = \tilde{K}_{q_0}$, where $\tilde{K}_S = \tilde{K} \cap V(S_{\tilde{M}_D})$.*

Proof. Let $S_{P'}$ be the triangles of S with at least one vertex in $P'_{m,g}$. As according to Claim 14 vertices of $\tilde{K}$ are entirely used by $S_{\tilde{M}_D} \cup S_{M_D}$, the only way to consume vertices of $P'_{m,g}$ is by creating local triangles in $P'_{m,g}$ or triangles in $S_{\tilde{M}_D}$. In particular, we cannot have a triangle (u,v,w) with $\{u,v\} \subseteq \tilde{K}$ and $w \in P'_{m,g}$, or with $u \in \tilde{K}$ and $\{v,w\} \subseteq P'_{m,g}$. More formally, we get the partition $S_{P'} = S_{\subseteq P'} \cup S_{\tilde{M}_D}$. As S is a perfect packing and uses in particular all vertices of $P'_{m,g}$ we get $|V(S_{P'})| = |V(P'_{m,g})|$, implying $|V(S_{\subseteq P'})| = |V(P'_{m,g})| - m$ by Claim 14. By Lemma 13, this implies that there exists $q_0 \in [g]$ such that $X' \subseteq X'^{q_0}$ where $X' = V(P'_{m,g}) \setminus V(S_{\subseteq P'})$. As X' are the only remaining vertices that can be used by triangles of $S_{\tilde{M}_D}$, we get that the m triangles of $S_{\tilde{M}_D}$ are of the form (u,v,w) with $u \in \tilde{K}_{q_0}$, $v \in \tilde{M}_D$, and $w \in X'$. ◀

▶ **Claim 16.** *If there exists a perfect packing S of $\mathcal{T}$, then there exists $q_0 \in [g]$ such that $V(S_{P'} \cup S_{\tilde{M}_D} \cup S_{M_D}) = V(D) \setminus K_{q_0}$.*

Proof. By Claim 14 we know that $|S_{M_D}| = (g-1)m$. As by Claim 15 there exists $q_0 \in [g]$ such that $\tilde{K}_S = \tilde{K}_{q_0}$, we get that the $(g-1)m$ triangles of S_{M_D} are of the form (u,v,w) with $u \in K \setminus K_{q_0}$, $v \in M_D$, and $w \in \tilde{K} \setminus \tilde{K}_{q_0}$. ◀

▶ **Lemma 17** ($\star$). *If there exists a perfect packing S of $\mathcal{T}$, then $V(S_{GD}) \cap V(G) \subseteq V(L)$. Informally, triangles of S_{GD} do not use any vertex of $M_G, \tilde{L}, \tilde{M}_T$ and $P_{n,g}$.*

Lemma 17 will allow us to prove Claims 18, 19 and 20 using the same arguments as in the right part (D) of the tournament as all vertices of $M_G, \tilde{L}, \tilde{M}_T$ and $P_{n,g}$ must be used by triangles in S_G.

▶ **Claim 18** ($\star$). *If there exists a perfect packing S of $\mathcal{T}$, then $|S_{\tilde{M}_G}| = n$ and $|S_{M_G}| = (g-1)n$. This implies that $V(S_{\tilde{M}_G} \cup S_{M_G}) \cap V(\tilde{L}) = V(\tilde{L})$, meaning that vertices of $\tilde{L}$ are entirely used by $S_{\tilde{M}_G} \cup S_{M_G}$.*

▶ **Claim 19** (⋆). *If there exists a perfect packing S of $\mathcal{T}$, then there exists $p_0 \in [g]$ such that $\tilde{L}_S = \tilde{L}_{p_0}$, where $\tilde{L}_S = \tilde{L} \cap V(S_{\tilde{M}_G})$.*

▶ **Claim 20** (⋆). *If there exists a perfect packing S of $\mathcal{T}$, then there exists $p_0 \in [g]$ such that $V(S_P \cup S_{\tilde{M}_G} \cup S_{M_G}) = V(G) \setminus L_{p_0}$.*

We are now ready to state our final claim is now straightforward as according Claim 16 and 20 we can define $S_{(p_0,q_0)} = S \setminus ((S_{P'} \cup S_{\tilde{M}_D} \cup S_{M_D}) \cup (S_P \cup S_{\tilde{M}_G} \cup S_{M_G}))$.

▶ **Claim 21.** *If there exists a perfect packing S of $\mathcal{T}$, there exists $p_0, q_0 \in [g]$ and $S_{(p_0,q_0)} \subseteq S$ such that $V(S_{(p_0,q_0)}) = V(\mathcal{T}_{(p_0,q_0)})$ (or equivalently such that $S_{(p_0,q_0)}$ is a perfect packing of $\mathcal{T}_{(p_0,q_0)}$).*

Proof of the weak composition

▶ **Theorem 22.** *For any $\epsilon > 0$, C_3-Perfect-Packing-T (parameterized by the total number of vertices N) does not admit a polynomial (generalized) kernelization with size bound $\mathcal{O}(N^{2-\epsilon})$ unless NP $\subseteq$ coNP / Poly.*

Proof. Given t instances $\{\mathcal{I}_l\}$ of C_3-Perfect-Packing-T restricted to instances of Theorem 7, we define an instance $\mathcal{T}$ of C_3-Perfect-Packing-T as defined in Section 4. We recall that $g = \sqrt{t}$, and that for any $l \in [t]$, $|V(L_l)| = n$ and $|V(K_l)| = m$. Let $N = |V(\mathcal{T})|$. As $N = |V(P'_{(m,g)})| + m + (g-1)m + 2mg + |V(P_{(n,g)})| + n + (g-1)n + 2ng$ and $|V(P_{(\omega,\gamma)})| = O(\omega\gamma)$ by Lemma 13, we get $N = \mathcal{O}(g(n+m)) = \mathcal{O}(t^{\frac{1}{2+o(1)}} \max(|\mathcal{I}_l|))$. Let us now verify that there exists $l \in [t]$ such that $\mathcal{I}_l$ admits a perfect packing iff $\mathcal{T}$ admits a perfect packing. First assume that there exist $p_0, q_0 \in [g]$ such that $\mathcal{I}_{(p_0,q_0)}$ admits a perfect packing. By Lemma 21, there is a packing $S_{P'}$ of $P'_{(m,g)}$ such that $V(S_{p'}) = V(P'_{(m,g)}) \setminus {X'}^{q_0}$. We define a set $S_{\tilde{M}_D}$ of m vertex disjoint triangles of the form (u,v,w) with $u \in \tilde{L}_{q_0}, v \in \tilde{M}_D, w \in {X'}^{q_0}$. Then, we define a set S_{M_D} of $(g-1)m$ vertex disjoint triangles of the form (u,v,w) with $u \in L \setminus L_{q_0}, v \in M_D, w \in \tilde{L} \setminus \tilde{L}_{q_0}$. In the same way we define S_P, $S_{\tilde{M}_G}$ and S_{M_G}. Observe that $V(\mathcal{T}) \setminus ((S_{P'} \cup S_{\tilde{M}_D} \cup S_{M_D}) \cup (S_P \cup S_{\tilde{M}_G} \cup S_{M_G})) = K_{q_0} \cup L_{p_0}$, and thus we can complete our packing into a perfect packing of $\mathcal{T}$ as $\mathcal{I}_{(p_0,q_0)}$ admits a perfect packing. Conversely if there exists a perfect packing S of $\mathcal{T}$, then by Claim 21 there exists $p_0, q_0 \in [g]$ and $S_{(p_0,q_0)} \subseteq S$ such that $V(S_{(p_0,q_0)}) = V(\mathcal{T}_{(p_0,q_0)})$, implying that $\mathcal{I}_{(p_0,q_0)}$ admits a perfect packing. ◀

▶ **Corollary 23.** *For any $\epsilon > 0$, C_3-Packing-T (parameterized by the size k of the solution) does not admit a polynomial kernel with size $\mathcal{O}(k^{2-\epsilon})$ unless NP $\subseteq$ coNP / Poly.*

5 Conclusion and open questions

Concerning approximation algorithms for C_3-Packing-T restricted to sparse instances, we have provided a $(1 + \frac{6}{c+5})$-approximation algorithm where c is a lower bound of the minspan of the instance. On the other hand, it is not hard to solve by dynamic programming C_3-Packing-T for instances where maxspan is bounded above. Using these two opposite approaches it could be interesting to derive an approximation algorithm for C_3-Packing-T with factor better than 4/3 even for sparse tournaments.

Concerning FPT algorithms, the approach we used for sparse tournament (reducing to the case where $m = \mathcal{O}(k)$ and apply the $\mathcal{O}(m)$ vertices kernel) cannot work for the general case. Indeed, if we were able to sparsify the initial input such that $m' = \mathcal{O}(k^{2-\epsilon})$, applying the kernel in $\mathcal{O}(m')$ would lead to a tournament of total bit size (by encoding the two endpoint of each arc) $\mathcal{O}(m' log(m')) = \mathcal{O}(k^{2-\epsilon})$, contradicting Corollary 23. Thus the situation for C_3-Packing-T could be as in vertex cover where there exists a kernel in $\mathcal{O}(k)$ vertices, derived from [15], but the resulting instance cannot have $\mathcal{O}(k^{2-\epsilon})$ edges [8]. So it is challenging question to provide a kernel in $\mathcal{O}(k)$ vertices for the general C_3-Packing-T problem.

References

1 Faisal N. Abu-Khzam. A quadratic kernel for 3-set packing. In *International Conference on Theory and Applications of Models of Computation*, pages 81–87. Springer, 2009.

2 Giorgio Ausiello, Pierluigi Crescenzi, Giorgio Gambosi, Viggo Kann, Alberto Marchetti-Spaccamela, and Marco Protasi. *Complexity and approximation: Combinatorial optimization problems and their approximability properties.* Springer Science & Business Media, 2012.

3 Piotr Berman and Marek Karpinski. On some tighter inapproximability results. In *International Colloquium on Automata, Languages, and Programming*, pages 200–209. Springer, 1999.

4 Stephane Bessy, Marin Bougeret, and Jocelyn Thiebaut. Triangle packing in (sparse) tournaments: approximation and kernelization. Technical report, HAL LIRMM, lirmm-01550313, v1, 2017. URL: `https://hal-lirmm.ccsd.cnrs.fr/lirmm-01550313`.

5 Mao-Cheng Cai, Xiaotie Deng, and Wenan Zang. A min-max theorem on feedback vertex sets. *Mathematics of Operations Research*, 27(2):361–371, 2002.

6 Pierre Charbit, Stéphan Thomassé, and Anders Yeo. The minimum feedback arc set problem is NP-hard for tournaments. *Combinatorics, Probability and Computing*, 16(01):1–4, 2007.

7 Marek Cygan. Improved approximation for 3-dimensional matching via bounded pathwidth local search. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 509–518. IEEE, 2013.

8 Holger Dell and Dániel Marx. Kernelization of packing problems. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete algorithms, SODA'12*, 2012.

9 Venkatesan Guruswami, C. Pandu Rangan, Maw-Shang Chang, Gerard J. Chang, and C. K. Wong. The vertex-disjoint triangles problem. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 26–37. Springer, 1998.

10 Danny Hermelin and Xi Wu. Weak compositions and their applications to polynomial lower bounds for kernelization. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 104–113. Society for Industrial and Applied Mathematics, 2012.

11 Bart M. P. Jansen and Astrid Pieterse. Sparsification upper and lower bounds for graph problems and Not-All-Equal SAT. *Algorithmica*, pages 1–26, 2015.

12 Claire Kenyon-Mathieu and Warren Schudy. How to rank with few errors. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 95–103. ACM, 2007.

13 Matthias Mnich, Virginia Vassilevska Williams, and László A. Végh. A 7/3-Approximation for Feedback Vertex Sets in Tournaments. In *24th Annual European Symposium on Algorithms, ESA 2016*, pages 67:1–67:14, 2016.

14 Hannes Moser. A problem kernelization for graph packing. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pages 401–412. Springer, 2009.

15 George L. Nemhauser and Leslie E. Trotter Jr. Properties of vertex packing and independence system polyhedra. *Mathematical Programming*, 6(1):48–61, 1974.