Packing Arc-Disjoint 4-Cycles in Oriented Graphs

Jasine Babu ✉
Indian Institute of Technology Palakkad, India

R. Krithika ✉
Indian Institute of Technology Palakkad, India

Deepak Rajendraprasad ✉
Indian Institute of Technology Palakkad, India

Abstract

Given a directed graph G and a positive integer k, the Arc Disjoint r-Cycle Packing problem asks whether G has k arc-disjoint r-cycles. We show that, for each integer $r \geq 3$, Arc Disjoint r-Cycle Packing is NP-complete on oriented graphs with girth r. When r is even, the same result holds even when the input class is further restricted to be bipartite. On the positive side, focusing on $r = 4$ in oriented graphs, we study the complexity of the problem with respect to two parameterizations: solution size and vertex cover size. For the former, we give a cubic kernel with quadratic number of vertices. This is smaller than the compression size guaranteed by a reduction to the well-known 4-Set Packing. For the latter, we show fixed-parameter tractability using an unapparent integer linear programming formulation of an equivalent problem.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms; Mathematics of computing → Graph algorithms

Keywords and phrases arc-disjoint cycles, bipartite digraphs, oriented graphs, parameterized complexity

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2022.5

1 Introduction

Disjoint Cycle Packing is a fundamental problem in graph theory and combinatorial optimization. Given a (directed or undirected) graph G and a positive integer k, the objective of the Disjoint Cycle Packing problem is to determine whether G has k (vertex or arc/edge) disjoint cycles. All variants of Disjoint Cycle Packing are NP-complete [3, 13, 23] and therefore have been studied in various algorithmic realms. The fixed-parameter tractable (FPT) algorithm (with respect to the number k of cycles as the parameter) [7] given by Bodlaender in 1994 for Vertex Disjoint Cycle Packing is one of the earliest results in the parameterized complexity framework. This problem does not admit polynomial kernels [9] but has a lossy kernel [28]. However, Edge Disjoint Cycle Packing has a polynomial kernel (and hence is also FPT) [9].

While Disjoint Cycle Packing in undirected graphs is amenable to parameterized algorithms, their directed-analogues are not. On directed graphs, Vertex Disjoint Cycle Packing and Arc Disjoint Cycle Packing are both W[1]-hard [3, 31]. Therefore, studying this problem on a subclass of directed graphs and studying Disjoint r-Cycle Packing (where the length of each cycle in the solution set is required to be r) are natural directions of research. Both Vertex Disjoint Cycle Packing and Arc Disjoint Cycle Packing are NP-complete but FPT on tournaments [5, 6]. However, these problems are W[1]-hard on bipartite digraphs [3, 31]. In this paper, we focus on Arc Disjoint r-Cycle Packing in oriented graphs. Bessy et. al., have shown that Arc Disjoint 3-Cycle Packing is NP-Complete on tournaments [5]. From this, it follows easily (by reduction via subdivision of arcs) that, for each $q \geq 1$, Arc Disjoint $3q$-Cycle Packing is NP-complete

© Jasine Babu, R. Krithika, and Deepak Rajendraprasad;
licensed under Creative Commons License CC-BY 4.0
42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2022).
Editors: Anuj Dawar and Venkatesan Guruswami; Article No. 5; pp. 5:1–5:16
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

on oriented graphs with girth $3q$, and that the input class can be restricted to be bipartite when q is even. This leaves open the complexity of Arc Disjoint r-Cycle Packing in digraphs for $r \not\equiv 0 \mod 3$. Our first set of results close this gap.

We show that Arc Disjoint r-Cycle Packing is NP-complete on oriented graphs for each integer $r \geq 3$, by a reduction from a variant of Satisfiability [23, LO1].

- (Theorem 1) For each integer $r \geq 3$, Arc-Disjoint r-Cycle Packing on oriented graphs of girth r is NP-complete. Further, for each even integer $r \geq 4$, this result holds even when the input graph is restricted to be bipartite.

It is easy to verify that Arc Disjoint r-Cycle Packing reduces to r-Set Packing. In r-Set Packing, given a family $\mathcal{F}$ of sets over a universe U, where each set in the family has cardinality at most r, and a positive integer k, the objective is to decide whether there are sets $S_1, \ldots, S_k \in \mathcal{F}$ that are pairwise disjoint. Note that r is fixed. Given an instance (G, k) of Arc Disjoint r-Cycle Packing, the instance $(E(G), \mathcal{C}, k)$ of r-Set Packing where $\mathcal{C}$ is the set of r-cycles of G is equivalent to it. It is well-known that r-Set Packing admits a kernel with $\mathcal{O}(k^r)$ sets [17] and $\mathcal{O}(k^{r-1})$ elements [1, 29] leading to a straight-forward $\mathcal{O}^\star(2^{\mathcal{O}(k \log k)})$-time algorithm. Further, using the standard color-coding technique [2, 12, 30], r-Set Packing admits an $\mathcal{O}^\star(2^{\mathcal{O}(k)})$-time algorithm. These results imply that Arc Disjoint r-Cycle Packing in general digraphs admits an FPT algorithm with running time $\mathcal{O}^\star(2^{\mathcal{O}(k)})$ and a polynomial kernel. However, the kernel with $\mathcal{O}(k^r)$ sets and $\mathcal{O}(k^{r-1})$ elements for r-Set Packing does not straightaway give a kernel of same size for Arc Disjoint r-Cycle Packing. Restricting our attention to Arc Disjoint 4-Cycle Packing, we obtain a cubic kernel with quadratic number of vertices.

- (Theorem 2) Arc Disjoint 4-Cycle Packing in oriented graphs has a kernel with $\mathcal{O}(k^3)$ edges and $\mathcal{O}(k^2)$ vertices.

In parameterized complexity, solution size is one of the most natural and almost always the first parameter considered for an optimization problem. Though this parameterization has proven to be fruitful, solution size does not reflect the input structure. Therefore, structural parameters like treewidth, the size of a vertex cover, the size of a feedback vertex set and in general the size of a modulator to a family of graphs have been considered in the literature [11, 12, 16, 18, 24]. In the context of Disjoint Cycle Packing, the parameters that have been studied are treewidth, the size of a vertex cover, the size of a feedback vertex set and a modulator to a cluster graph and max leaf number [8, 24]. The importance in these structural parameters is partly due to their practical relevance and partly due to their role in identifying parameterizations that yield FPT algorithms. In this spirit, we study the complexity of Arc Disjoint 4-Cycle Packing parameterized by the size of a vertex cover (of the underlying undirected graph). We first reduce this problem to Arc Disjoint 4-Cycle Packing in oriented bipartite graphs parameterized by the size ℓ of one of the parts of the bipartition. Then we define a new equivalent problem of building a multidigraph with certain decomposition properties and give an integer linear programming formulation for it where the number of variables is a function of ℓ. Finally, by invoking the FPT algorithm for Integer Linear Programming parameterized by the number of variables [12, 22, 25, 26], we obtain the following result.

- (Theorem 3) Arc Disjoint 4-Cycle Packing in oriented graphs is FPT with respect to size of a vertex cover as the parameter.

Road Map. The paper is organized as follows. In Section 2, we give the necessary definitions related to directed graphs. In Section 3, we show the NP-completeness of Arc Disjoint r-Cycle Packing in oriented (bipartite) graphs of girth r. In Section 4, we describe FPT algorithms and polynomial kernels for Arc Disjoint 4-Cycle Packing. Finally, we conclude with some remarks and open problems in Section 5.

2 Preliminaries

The set $\{1, 2, \ldots, n\}$ is denoted by $[n]$. A *multidigraph* is a pair (V, A) consisting of a set V of *vertices* and a multiset A of ordered pairs of vertices (called *arcs*) in V. A *directed graph* (or *digraph*) is a multidigraph (V, A) where A is a set of ordered pairs of distinct vertices in V. Note that a digraph is a multidigraph with no self-loops or multiple/parallel arcs. An arc is specified as an ordered pair of vertices and this pair of vertices are called as its *endpoints*. A *matching* is a collection of arcs that do not share any endpoint. For a digraph G, $V(G)$ and $A(G)$ denote the set of its vertices and the set of its arcs, respectively. An *oriented graph* is a digraph G having no pair of vertices $u, v \in V(G)$ with $(u, v), (v, u) \in A(G)$. A *bipartite (di)graph* is a (di)graph G whose vertex set can be partitioned into two sets X and Y such that every arc/edge in G has one endpoint in X and the other endpoint in Y. We denote such a digraph as $G[X, Y]$ and say that (X, Y) is a bipartition of G.

Two vertices u, v are said to be *adjacent* in G if $(u, v) \in A(G)$ or $(v, u) \in A(G)$. For a set of arcs F, $V(F)$ denotes the union of the sets of endpoints of arcs in F. A vertex u is said to be an *in-neighbour* of v if $(u, v) \in A(G)$. Similarly, a vertex u is said to be an *out-neighbour* of v if $(v, u) \in A(G)$. For a vertex $v \in V(G)$, its *out-neighborhood*, denoted by $N^+(v)$, is the set $\{u \in V(G) \mid (v, u) \in A(G)\}$ and its *in-neighborhood*, denoted by $N^-(v)$, is the set $\{u \in V(G) \mid (u, v) \in A(G)\}$. The *out-degree* and *in-degree* of a vertex v are the sizes of its out-neighborhood and in-neighborhood, respectively. For a set $X \subseteq V(G) \cup A(G)$, $G - X$ denotes the digraph obtained from G by deleting X.

A *path* P in G is a sequence $(v_1, \ldots, v_k)$ of distinct vertices such that for each $i \in [k-1]$, $(v_i, v_{i+1}) \in A(G)$. We say that P *starts at* v_1 and *ends at* v_k and also refer to v_1 and v_k as the endpoints of P. The set $\{v_1, \ldots, v_k\}$ is denoted by $V(P)$ and the set $\{(v_i, v_{i+1}) \mid i \in [k-1]\}$ is denoted by $A(P)$. A *cycle* C in G is a sequence $(v_1, \ldots, v_k)$ of distinct vertices such that $(v_1, \ldots, v_k)$ is a path and $(v_k, v_1) \in A(G)$. The set $\{v_1, \ldots, v_k\}$ is denoted by $V(C)$ and the set $\{(v_i, v_{i+1}) \mid i \in [k-1]\} \cup \{(v_k, v_1)\}$ is denoted by $A(C)$. The length of a path or cycle X is the number of vertices in it. A cycle (path) of length q is called a q-cycle (q-path) and a cycle on three vertices is also called a *triangle*. A collection of q-cycles is called a C_q*-packing* or a q*-cycle packing*.

For details on parameterized algorithms, we refer to standard books in the area [12, 14, 19, 21].

3 NP-Completeness

In this section, we show that for each even integer $r \geq 4$, Arc-Disjoint r-Cycle Packing is NP-complete on oriented bipartite graphs of girth r and for each integer $r \geq 3$, Arc-Disjoint r-Cycle Packing is NP-complete on oriented graphs of girth r. It is easy to verify that Arc Disjoint r-Cycle Packing is in NP. To prove NP-hardness, we give a polynomial-time reduction from a variant of the Satisfiability problem. Let SAT(1, 2) denote Satisfiability restricted to formulas with at most 3 variables per clause and each variable occurring exactly once negatively and once or twice positively in the formula.

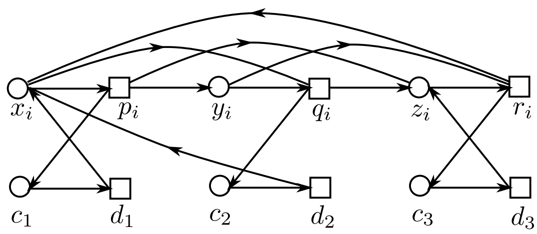

Figure 1 Gadget corresponding to the variable X_i that appears negatively in C_1 and positively in C_2 and C_3. Here, circles and squares denote a bipartition.

It is well-known that Satisfiability is NP-complete when restricted to instances with 2 or 3 variables per clause and at most 3 occurrences per variable [32, Theorem 2.1]. By a straightforward transformation of instances of this problem into instances of SAT(1, 2), it follows that SAT(1, 2) is also NP-complete.

Now, we proceed to the NP-hardness of Arc-Disjoint r-Cycle Packing. First, we show the hardness of Arc-Disjoint 4-Cycle Packing and then move on to the general case. Consider an instance ψ of SAT(1,2) with n variables and m clauses. From ψ, we construct an oriented bipartite graph G such that ψ is satisfiable if and only if G has an arc-disjoint 4-cycle packing of size $m + n$. Let $\{X_1, X_2, \ldots, X_n\}$ and $\{C_1, C_2, \ldots, C_m\}$ be the sets of variables and clauses, respectively, in ψ. We consider the ordering of the variables (and clauses) given by the increasing order of their indices. The construction of G from ψ is as follows.

- For every $i \in [n]$, add a set of six vertices $\{x_i, y_i, z_i, p_i, q_i, r_i\}$ where $(x_i, p_i, y_i, q_i, z_i, r_i)$ is a directed path and $(r_i, x_i), (x_i, q_i), (y_i, r_i), (p_i, z_i) \in A(G)$.
- For every $j \in [m]$, add two vertices c_j and d_j along with the arc (c_j, d_j).
- For every $i \in [n]$ and $j \in [m]$ such that X_i appears in C_j,
  - if the appearance is as a negative literal, then add arcs (d_j, x_i) and (p_i, c_j).
  - if the appearance is as a positive literal and this is the first such appearance, then add arcs (d_j, x_i) and (q_i, c_j).
  - if the appearance is as a positive literal and this is the second such appearance, then add arcs $(d_j, z_i), (r_i, c_j)$.

Refer to Figure 1 for an illustration. In order to prove the correctness of the reduction, we first make some observations about the type of 4-cycles in G. Notice that for each clause C_j, the vertices c_j and d_j in G respectively have out-degree one and in-degree one. Hence, any 4-cycle containing one of these vertices must contain the arc (c_j, d_j). Further, observe that if a 4-cycle contains the arc (c_j, d_j), then the other two vertices of that cycle must belong to the the same variable gadget X_i for some $i \in [n]$. We will refer to such a cycle as a *clause-variable cycle* involving C_j and X_i. Notice that if there is a clause-variable cycle involving C_j and X_i, then it must be one of the following: (c_j, d_j, x_i, p_i), (c_j, d_j, x_i, q_i) or (c_j, d_j, z_i, r_i). The first one will be referred to as the *negative clause-variable cycle* and the other two will be referred to as *positive clause-variable cycles.*

▶ **Observation 3.1.** *Let $j \in [m]$. If a 4-cycle in G contains a vertex c_j or d_j, then it must be a clause-variable cycle. In any arc-disjoint 4-cycle packing of G, there is at most one clause-variable cycle involving C_j.*

Now, consider a 4-cycle of G that does not contain c_j or d_j vertices for any $j \in [m]$. In any such cycle, all its four vertices must belong to the same variable gadget X_i for some $i \in [n]$. This is because there are no edges between vertices that belong to two distinct variable

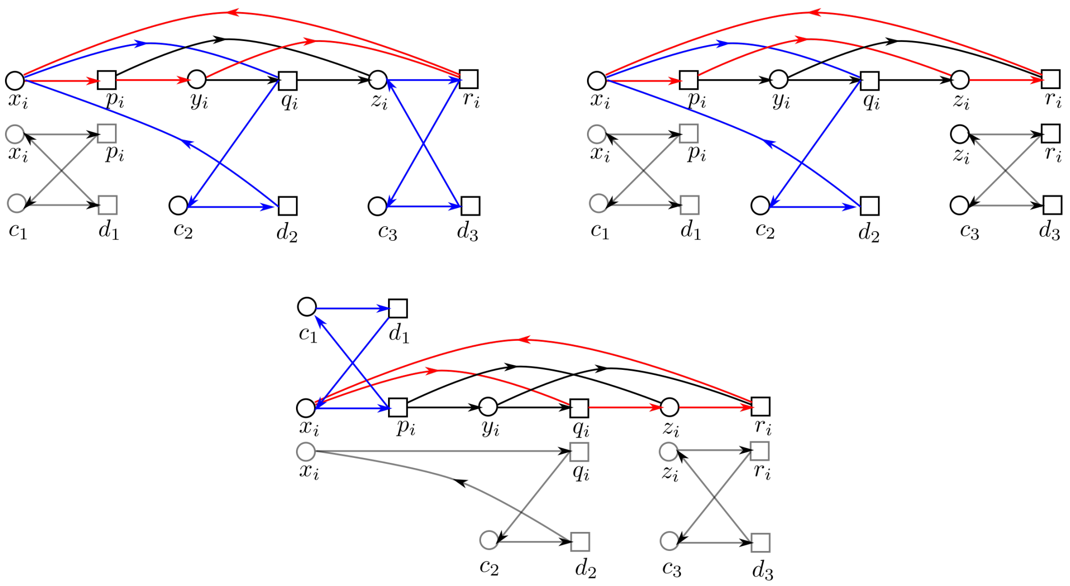

Figure 2 The 3 different variable cycles highlighted in red with the clause-variable cycles that are arc-disjoint to it highlighted in blue and the clause-variable cycles that are not arc-disjoint to it shown separately in grey.

gadgets. Any such 4-cycle whose vertex set is contained in the variable gadget of X_i would be referred to as a *variable cycle* of X_i. Notice that the variable cycles of X_i possible are (x_i, p_i, y_i, r_i), (x_i, p_i, z_i, r_i) and (x_i, q_i, z_i, r_i). We will call (x_i, p_i, y_i, r_i) as the *first positive variable cycle* of X_i, (x_i, p_i, z_i, r_i) as the *second positive variable cycle* of X_i and (x_i, q_i, z_i, r_i) as the *negative variable cycle* of X_i. Note that all these three cycles contain the arc (r_i, x_i). Hence, we can make the following observation.

▶ **Observation 3.2.** *Any 4-cycle in G which is not a clause-variable cycle must be a variable cycle. Let $i \in [n]$. In any arc-disjoint 4-cycle packing of G, there is at most one variable cycle of X_i.*

▶ **Observation 3.3.** *Let $i \in [n]$.*

- *The first positive variable cycle of X_i is arc-disjoint from positive clause-variable cycles involving any clause C_j and X_i.*
- *The negative variable cycle of X_i is arc-disjoint from negative clause-variable cycles involving any clause C_j and X_i.*
- *If X_i appears positively in clauses C_j and C_k, then the positive clause-variable cycle involving C_j and X_i is arc-disjoint from the positive clause-variable cycle involving C_k and X_i.*

The following is another crucial observation.

▶ **Observation 3.4.** *Suppose $\mathcal{F}$ is a family of arc-disjoint 4-cycles in G. Let $i \in [n]$.*

- *If $\mathcal{F}$ contains a negative clause-variable cycle involving clause C_j and X_i for some $j \in [m]$, then $\mathcal{F}$ cannot contain a (first or second) positive variable cycle of X_i.*
- *If $\mathcal{F}$ contains a positive clause-variable cycle involving clause C_j and X_i for some $j \in [m]$, then $\mathcal{F}$ cannot contain a negative variable cycle of X_i.*

Refer to Figure 2 for an illustration.

▶ **Lemma 3.1.** *ψ is a* Yes*-instance of SAT(1,2) if and only if $(G, m+n)$ is a* Yes*-instance of* Arc-disjoint 4-Cycles.

Proof. From Observations 3.1 and 3.2, it follows that any arc-disjoint 4-cycle packing of G can contain at most $m+n$ cycles.

Suppose ψ is a Yes-instance of SAT(1,2). We will produce a family $\mathcal{F}$ consisting of $m+n$ arc-disjoint 4-cycles in G. Let τ be a satisfying truth assignment of ψ. When ψ is evaluated on τ, in each clause C_j of ψ, at least one literal gets the truth value True. From each clause C_j, choose any one literal l_j which gets the truth value True and do the following.

- if l_j is a positive occurrence of a variable X_i, then add a positive clause-variable cycle involving C_j and X_i to $\mathcal{F}$.
- if l_j is a negative occurrence of a variable X_i, then add the negative clause-variable cycle involving C_j and X_i to $\mathcal{F}$.

For each variable X_i do the following.

- if X_i is assigned to be True, add the first positive variable cycle of X_i to $\mathcal{F}$.
- otherwise, add the negative variable cycle of X_i to $\mathcal{F}$.

From Observation 3.3, it follows that the cycles added to $\mathcal{F}$ are arc-disjoint. Further, we have added $m+n$ cycles to $\mathcal{F}$.

To prove the converse, suppose there exist a family $\mathcal{F}$ consisting of $m+n$ arc-disjoint 4-cycles in G. We will produce a truth assignment τ that satisfies ψ. By Observations 3.1 and 3.2, among the cycles in $\mathcal{F}$, exactly m are clause-variable cycles and exactly n are variable cycles, one corresponding to each variable. If the variable cycle of X_i is a positive variable cycle, assign X_i to be True. If the variable cycle of X_i is a negative variable cycle, assign X_i to be False. This defines the truth assignment τ. Now, it remains to show that τ satisfies ψ. Consider any clause C_j of ψ. Since $\mathcal{F}$ contains exactly m clause-variable cycles, there must be a clause-variable cycle in $\mathcal{F}$ that involves C_j and some variable X_i. By Observation 3.4, if the variable cycle of X_i in $\mathcal{F}$ is a positive variable cycle, then the clause-variable cycle in $\mathcal{F}$ involving C_j and X_i is a positive clause variable cycle. If this happens, then X_i appears positively in C_j. As per our truth assignment τ, we have assigned X_i to be True and hence, the clause C_j will be satisfied. Similarly, from Observation 3.4, we can also show that if variable cycle of X_i in $\mathcal{F}$ is a negative variable cycle, then we would have assigned X_i to False, in which case as well, the clause C_j will be satisfied by τ. Since the same argument holds for every clause C_j, we can see that τ satisfies all clauses of ψ simultaneously. ◀

Lemma 3.1 along with the fact that the reduction described runs in polynomial time leads to the following result.

▶ **Lemma 3.2.** *Arc-Disjoint 4-Cycle Packing in oriented bipartite graphs is* NP*-complete.*

Now, we give a generalization of the above reduction by constructing a graph G_r from ψ, for each integer $r \geq 4$. If $r = 4$, then $G_r = G$. If $r > 4$, then we make the following two modifications in G to obtain G_r.

- For every $i \in [n]$, replace the arc (r_i, x_i) in G by a path $P_i = (r_i, a_{i,1}, a_{i,2} \ldots, a_{i,r-4}, x_i)$ of length $r-2$ from r_i to x_i in G_r.
- For every $j \in [m]$, replace the arc (c_j, d_j) in G by a path $Q_j = (c_j, b_{j,1}, b_{j,2} \ldots, b_{j,r-4}, d_j)$ of length $r-2$ from c_j to d_j in G_r.

By extending similar arguments as in the case of G, it can be seen that for any $j \in [m]$ if a cycle in G_r contains either vertex c_j or vertex d_j, then that cycle must contain the entire path Q_j. Further, similar to G, for any cycle of G_r that does not contain c_j or d_j vertices for any $j \in [m]$, all vertices of the cycle must belong to the same variable gadget X_i for some $i \in [n]$ and the cycle must necessarily contain the entire path P_i. From these arguments, it can be seen that G_r has no cycles of length less than r.

By extending our earlier definitions, r-cycles in G_r that contain Q_j for some $j \in [m]$ will be called as clause-variable cycles and r-cycles in G_r that contain P_i for some $i \in [n]$ will be called as variable cycles. The terminology of negative and positive clause variable cycles can also be generalized in the same manner. It can be seen that a variable cycle that contains P_i has exactly two vertices from outside P_i and they both belong to the gadget of variable X_i itself and a clause-variable cycle that contains Q_j has exactly two vertices from outside Q_j and they both belong to the same variable gadget.

Our remaining arguments for the case when $r = 4$ also extend easily so that the statements of Observation 3.1, Observation 3.2, Observation 3.3, Observation 3.4 and Lemma 3.1 can be generalized by replacing G with G_r and 4 with r. It may be noted that the graph G_r obtained has girth r and when r is even, G_r is bipartite. Further, it is known from the literature that Arc-Disjoint 3-Cycle Packing is NP-complete for tournaments [5]. Hence, we have the following theorem, obtained as a generalization of Lemma 3.2.

▶ **Theorem 1.** *For each integer $r \geq 3$, Arc-Disjoint r-Cycle Packing on oriented graphs of girth r is NP-complete. Further, for each even integer $r \geq 4$, this result holds even when the input graph is restricted to be bipartite.*

Arc Disjoint $r_{\leq}$ Cycle Packing is a related problem, where the cycles in the packing are required to be of length at most r. For an odd integer $r > 4$, the answer to Arc Disjoint $r_{\leq}$ Cycle Packing on any input oriented bipartite graph is the same as when the cycles in the packing are restricted to be of length at most $r - 1$. This yields the following corollary.

▶ **Corollary 1.** *For each integer $r \geq 4$, Arc Disjoint $r_{\leq}$ Cycle Packing on oriented bipartite graphs is NP-complete.*

4 FPT Algorithms and Kernels for Packing Arc-Disjoint 4-Cycles

Now, we move on to describing FPT algorithms and polynomial kernels for Arc Disjoint 4-Cycle Packing in oriented graphs. First we consider the standard parameter (solution size) and then proceed to vertex cover size as the parameter.

4.1 Solution Size as Parameter

As mentioned before, Arc Disjoint 4-Cycle Packing reduces to 4-Set Packing. While the current fastest FPT algorithm for 4-Set Packing solves Arc Disjoint 4-Cycle Packing in the same time, the kernel for 4-Set Packing with $\mathcal{O}(k^3)$ elements and $\mathcal{O}(k^4)$ sets does not straight away give a kernel of same size for Arc Disjoint 4-Cycle Packing. Here, we describe an $\mathcal{O}(k^3)$ sized kernel with $\mathcal{O}(k^2)$ vertices. Towards this, we give two kernelization procedures - one that produces a cubic kernel and an other that gives a quadratic vertex kernel. By combining these two procedures, we get the desired kernel. Our algorithm crucially uses reduction rules based on the *new expansion lemma* [20] and *sunflowers* [15, 19, 21].

Consider an instance (G, k) of Arc Disjoint 4-Cycle Packing. The first reduction rule is a standard preprocessing rule.

▶ **Reduction Rule 4.1.** *If there is a vertex or an edge in G that is not in any 4-cycle, then delete it.*

The correctness of this rule is immediate as no C_4-packing can contain a vertex or an edge that is not a part of any 4-cycle. The next reduction rule uses the notion of *sunflowers* [15, 19, 21].

▶ **Definition 4.1.** (ℓ-sunflower) *An ℓ-sunflower in G with core $C \subseteq A(G)$ is a set $\mathcal{S}$ of ℓ 4-cycles such that for any two elements S and S' in $\mathcal{S}$, we have $A(S) \cap A(S') = C$.*

Now, we describe a rule that identifies edges that may be assumed to be in some solution (if the instance is a YES-instance) using sunflowers.

▶ **Reduction Rule 4.2.** *If there is an edge e in G such that there is a $(4k-3)$-sunflower with $\{e\}$ as the core, then delete e from G and decrease k by 1.*

Note that finding a sunflower with core $\{(u, v)\}$ reduces to finding a maximum matching in the auxiliary bipartite graph G_{uv} with bipartition (A, B) defined as follows: $A = N^-(u)$, $B = N^+(v)$ and a vertex $x \in A$ is adjacent to a vertex $y \in B$ if and only if $(y, x) \in A(G)$. Observe that A and B are not necessarily disjoint and we rename B so that $A \cap B = \emptyset$ before finding the matching. Thus, Reduction Rule 4.2 can be applied in polynomial time. The correctness of the rule is justified by the following lemma.

▶ **Lemma 4.1.** *If Reduction Rule 4.2 is applicable on (G, k), then (G, k) is a YES-instance if and only if $(G - e, k - 1)$ is a YES-instance.*

Proof. Let $\mathcal{S}$ denote a $(4k-3)$-sunflower in G with core $\{e\}$. Since e is in at most one cycle of any 4-cycle packing of G, it is clear that $(G - e, k - 1)$ is a YES-instance, whenever (G, k) is. Conversely, consider a $(k-1)$-sized 4-cycle packing $\mathcal{F}$ of $G - e$. Then, $|A(\mathcal{F})| = 4(k-1)$ and there are at most $4(k-1)$ cycles in $\mathcal{S}$ that have an edge from $A(\mathcal{F})$. Therefore, there exists a cycle C in the sunflower $\mathcal{S}$ such that $A(C) \cap A(\mathcal{F}) = \emptyset$. Then, $\mathcal{F} \cup \{C\}$ is a k-sized 4-cycle packing of G. ◀

Subsequently, we assume that (G, k) is an instance on which Reductions Rules 4.1 and 4.2 are not applicable. Let $\mathcal{X}$ be a maximal set of 4-cycles in G such that for any two elements X and Y in $\mathcal{X}$, we have $|A(X) \cap A(Y)| \leq 1$. The following lemma shows that if $\mathcal{X}$ is sufficiently large, then (G, k) is a YES-instance.

▶ **Lemma 4.2.** *If $|\mathcal{X}| > 16k^2$, then there is a set $\mathcal{C} \subseteq \mathcal{X}$ of k arc-disjoint 4-cycles that can be obtained in polynomial time.*

Proof. Obtain a sequence $\mathcal{F}_1, \ldots, \mathcal{F}_r$ of disjoint subsets of $\mathcal{X}$ as follows. For $i \geq 1$, consider an arbitrary 4-cycle $C_i \in \mathcal{X} \setminus \bigcup_{j=1}^{i-1} \mathcal{F}_j$ and let $\mathcal{F}_i$ be the subset of cycles in $\mathcal{X} \setminus \bigcup_{j=1}^{i-1} \mathcal{F}_j$ that share an edge with C_i. Observe that for each $i \in [r]$, $|\mathcal{F}_i| \leq 4(4(k-1))$ as Reduction Rule 4.2 is not applicable. Further, $r \geq k$ as $|\mathcal{X}| > 16k^2$. Then, $C_1, \ldots, C_k$ is the required 4-cycle packing. ◀

Lemma 4.2 lets us apply the following reduction rule.

▶ **Reduction Rule 4.3.** *If $|\mathcal{X}| > 16k^2$, then replace the instance (G, k) by a constant-sized YES-instance.*

Subsequently, we assume that $|\mathcal{X}| \leq 16k^2$. Define $\mathcal{P}$ to be the set of all 3-paths of G that are in some 4-cycle in $\mathcal{X}$. Note that $\mathcal{P}$ is $\mathcal{O}(k^2)$. The next lemma says that $\mathcal{P}$ "hits" all 4-cycles in G.

▶ **Lemma 4.3.** *For every 4-cycle C in G, there is a 3-path $P \in \mathcal{P}$ such that $A(P) \subseteq A(C)$.*

Proof. Consider a 4-cycle C in G. If $C \in \mathcal{X}$, then the claim trivially holds. Otherwise, $C \notin \mathcal{X}$ and by the maximality of $\mathcal{X}$, there is a 4-cycle X in $\mathcal{X}$ such that $|A(X) \cap A(C)| \geq 2$. Let e and e' be two common arcs of X and C. If e and e' share an endpoint, then the 3-path formed by e and e' that is present in X and C is also in $\mathcal{P}$ and the claim holds. Otherwise, e and e' form a matching implying that $X = C$ leading to a contradiction. ◀

The next rule is one that uses the notion of *expansion* and *new expansion lemma* [20].

▶ **Definition 4.2** (ℓ-expansion)**.** *Let ℓ be a positive integer and H be a bipartite graph with bipartition (A, B). For $\hat{A} \subseteq A$ and $\hat{B} \subseteq B$, a set $M \subseteq E(H)$ of edges is called an ℓ-*expansion *of $\hat{A}$ onto $\hat{B}$ if the following properties hold.*

- *every vertex of $\hat{A}$ is incident to exactly ℓ edges in M.*
- *exactly $\ell|\hat{A}|$ vertices in $\hat{B}$ are incident to edges in M.*

For an ℓ-expansion M of $\hat{A}$ onto $\hat{B}$, we call the vertices of $\hat{B}$ that are endpoints of edges in M as *saturated* and the remaining vertices of $\hat{B}$ as *unsaturated*. Observe that a 1-expansion is simply a matching of $\hat{A}$ to $\hat{B}$ that saturates $\hat{A}$.

▶ **Proposition 4.1** (New ℓ-Expansion Lemma, [20])**.** *Let ℓ be a positive integer and H be a bipartite graph with bipartition (A, B). Then there exists $\hat{A} \subseteq A$ and $\hat{B} \subseteq B$ such that there is an ℓ-expansion M of $\hat{A}$ onto $\hat{B}$ in H, $N(\hat{B}) \subseteq \hat{A}$ and $|B \setminus \hat{B}| \leq \ell|A \setminus \hat{A}|$. Moreover, the sets $\hat{A}$ and $\hat{B}$ (and M) can be computed in polynomial time.*

Note that $\hat{B}$ (and $\hat{A}$) may be empty. In that case, since $|B \setminus \hat{B}| \leq \ell|A \setminus \hat{A}|$, we have $|B| \leq \ell|A|$. Therefore, if $|B| > \ell|A|$, then $\hat{B} \neq \emptyset$. Now, we are ready to state the next reduction rule.

We will apply Proposition 4.1 on an auxiliary bipartite graph $\hat{G}$ with bipartition (A, B) where $A = \mathcal{P}$ and $B = V(G) \setminus V(\mathcal{P})$, and a vertex (x, y, z) in A is adjacent to $v \in B$ if and only if (x, y, z, v) is a 4-cycle in G.

▶ **Reduction Rule 4.4.** *Let $\hat{A} \subseteq A$ and $\hat{B} \subseteq B$ be the sets and M be the ℓ-expansion computed by Proposition 4.1 on $\hat{G}$ for $\ell = 1$. Let $U \subseteq \hat{B}$ be the set of unsaturated vertices of M. Delete U from G.*

Observe that after the application of this rule, if $\hat{B}$ is empty, then $|V(G)| \leq 3|A| + |B| \leq 3|A|+|A|$. Otherwise, $\hat{B}$ is non-empty and $|V(G)| \leq 3|A|+|\hat{B} \setminus U|+|B \setminus \hat{B}| \leq 3|A|+|\hat{A}|+|A \setminus \hat{A}|$. In both cases, $|V(G)|$ is $\mathcal{O}(k^2)$. The correctness of the rule is justified by the following lemma.

▶ **Lemma 4.4.** *(G, k) is a yes-instance if and only if $(G - U, k)$ is a yes-instance.*

Proof. The "if" part of the claim follows from the fact that $G - U$ is a subgraph of G. Now, suppose (G, k) is a yes-instance. Let $\mathcal{C}$ be a k-sized 4-cycle packing of G. Note that by Lemma 4.3, for each $C \in \mathcal{C}$, there exists a 3-path P_C in $\mathcal{P}$ that is contained in C. Let $\widehat{\mathcal{C}}$ be the set of cycles in $\mathcal{C}$ such that for each $C \in \widehat{\mathcal{C}}$, there is a 3-path P_C in $\hat{A}$ such that P_C is contained in C. Note that each 4-cycle in $\mathcal{C} \setminus \widehat{\mathcal{C}}$ is also a 4-cycle in $G - U$ as $N(\widehat{B}) \subseteq \hat{A}$. Let $\widehat{\mathcal{C}} = \{C_1, \ldots, C_r\}$ and for each $i \in [r]$, let $P_i = (x_i, y_i, z_i)$ be a 3-path in $\hat{A}$ that is contained in C_i. As $\hat{A}$ is saturated by a matching M in $\hat{G}$, there is a vertex $v_i \in B$ matched to the vertex P_i in $\hat{A}$. By replacing each C_i by (x_i, y_i, z_i, v_i) in $\mathcal{C}$, we get a k-sized 4-cycle packing of $G - U$. ◀

Now, we have the following property on an instance on which none of the reduction rules described so far is applicable.

▶ **Proposition 4.2.** *If (G, k) is an instance of* Arc Disjoint 4-Cycle Packing *on which none of the Reduction Rules 4.1, 4.2, 4.3, 4.4 is applicable, then G has $\mathcal{O}(k^2)$ vertices.*

Note that since all the reduction rules described can be applied in polynomial time, Proposition 4.2 gives us a quadratic vertex kernel for Arc Disjoint 4-Cycle Packing. However, this kernel may have $\mathcal{O}(k^4)$ edges.

Next, we describe a procedure that reduces the number of edges to $\mathcal{O}(k^3)$. This procedure is a stand-alone algorithm that produces a cubic kernel. We begin with the definition of *C_4-partners.*

▶ **Definition 4.3.** *Two arcs (x, y) and (z, v) are called C_4-partners in G if (x, y, z, v) is a C_4 in G.*

For an arc (x, y) in G, E_{xy} denotes the set of all C_4-partners of (x, y). We will rightfully treat E_{xy} as a subdigraph of G. For a collection of subgraphs $\mathcal{H}$ of G, $A(\mathcal{H})$ denotes $\bigcup_{H\in\mathcal{H}} A(H)$. For a set S and a natural number q, $\lfloor S \rfloor_q$ denotes S itself if $|S| \leq q$ and an arbitrary q-sized subset of S otherwise.

We now describe a procedure that marks a certain number of 4-cycles and takes the subgraph spanned by the arcs of the marked cycles as the kerenel. This marking procedure is described as Algorithm 1.

Algorithm 1 MarkingProcedure.

Require: A digraph G and a positive integer k
Ensure: A subgraph H of G on $\mathcal{O}(k^3)$ edges such that H has an arc-disjoint C_4-packing of size k if and only if G has an arc-disjoint C_4-packing of size k.

```
Find a maximal arc-disjoint C_4-packing P in G
if |P| ≥ k then
    A(H) ← A(P') and V(H) ← V(P'), where P' is any subset of k distinct cycles from P.
    return H
for each arc (x, y) ∈ A(P) do
    if E_xy has a matching M of size 4k then
        F_xy = M
    else
        Let S be the set of endpoints of a maximum matching in E_xy
        F_xy ← ∅
        for each vertex s ∈ S do
            Let A_s be the set of arcs in E_xy outgoing from s
            Let B_s be the set of arcs in E_xy incoming to s
            Add ⌊A_s⌋_4k ∪ ⌊B_s⌋_4k to F_xy
    for each arc (z, v) ∈ F_xy do
        mark the 4-cycle (x, y, z, v)
Let H be the subgraph of G whose arc set is the union of the arcs of all the marked 4-cycles and vertex set is the set of endpoints of its arcs.
return H
```

We show the correctness of **MarkingProcedure** in the following lemma.

▶ **Lemma 4.5.** *Given a digraph G and a positive integer k,* ***MarkingProcedure*** *(Algorithm 1) returns a digraph H on $O(k^3)$ edges and $\mathcal{O}(|V(G)|)$ vertices such that H has an arc-disjoint C_4-packing of size k if and only if G has an arc-disjoint C_4-packing of size k.*

Proof. Since H is a subgraph of G, the "only if" direction is immediate. To prove the other direction, suppose (G, k) is an yes-instance and let $\mathcal{Q} = \{Q_1, \ldots Q_k\}$ be an arc-disjoint C_4-packing in G which has the maximum number of edges from H. If all the cycles in $\mathcal{Q}$ are present in H, then it has a k-sized packing as required. Hence for the rest of the proof we assume Q_1 is not contained in H and show that we can replace Q_1 with another 4-cycle Q'_1 which is contained in H and arc-disjoint from $Q_2, \ldots, Q_k$. Then the C_4-packing $\{Q'_1, Q_2, \ldots, Q_k\}$ contradicts the choice of $\mathcal{Q}$.

Since Q_1 is a 4-cycle in G, and $\mathcal{P}$ is a maximal C_4-packing in G, there is an arc (x, y) of Q_1 present $A(\mathcal{P})$. Let $Q_1 = (x, y, z, v)$. If (z, v) was added to F_{xy} by the algorithm, Q_1 would have been marked and hence Q_1 would be present in H. Hence we assume that $(z, v) \notin F_{xy}$. If E_{xy} had a matching of size $4k$, then there are $4k$ arc-disjoint 3-length paths from y to x in in H. At most $4(k-1)$ of them can be hit by other cycles $Q_2, \ldots, Q_k$ of the packing $\mathcal{Q}$. Let (y, z', v', x) be a 3-length path in H that is arc-disjoint from $Q_2, \ldots, Q_k$. We can replace $Q_1 = (x, y, z, v)$ with $Q'_1 = (x, y, z', v')$ in $\mathcal{Q}$.

Suppose E_{xy} did not have a matching of size $4k$. Then either z or v (or both) belong to S. We will argue the case when $z \in S$. The case $v \in S$ is similar. Since (z, v) is not in F_{xy}, the set F_{xy} contains $4k$ other edges of E_{xy} outgoing from z. Hence there are $4k$ arc-disjoint 2-length paths from z to x in in H. At most $4(k-1)$ of them can be hit by other cycles $Q_2 \ldots Q_k$ of the packing $\mathcal{Q}$. Let (z, v', x) be a 2-length path in H that is arc-disjoint from $Q_2, \ldots, Q_k$. We can replace $Q_1 = (x, y, z, v)$ with $Q'_1 = (x, y, z, v')$ in $\mathcal{Q}$.

We now prove a bound on the size of H. Observe that for each arc $(x, y) \in A(\mathcal{P})$, $|F_{xy}|$ is bounded above by $4k$ in the first case and $8k|S|$ in the second case. Since $|S| \le 2(4k-1)$, we have $|F_{xy}| \le 64k^2$. Since $|A(\mathcal{P})| < 4k$, the total number of 4-cycles that are marked is less than $256k^3$ and hence H has at most $1024k^3$ arcs. ◀

Applying Algorithm 1 on the instance guaranteed by Proposition 4.2, leads to the following result.

▶ **Theorem 2.** *Arc Disjoint 4-Cycle Packing admits an $\mathcal{O}(k^3)$ sized kernel with $\mathcal{O}(k^2)$ vertices.*

We remark that in Theorem 2, our focus was on only getting a cubic kernel with quadratic number of vertices and we have not attempted to optimize the constants involved.

4.2 Vertex Cover Size as Parameter

In this section, we study the time complexity of Arc-Disjoint 4-Cycle Packing in oriented graphs parameterized by the size of a vertex-cover. The first step is to reduce this problem to Arc Disjoint 4-Cycle Packing in oriented bipartite graphs parameterized by the size of the smaller part. Then we invent a new problem which is in bijective correspondence with arc-disjoint 4-cycle packing in oriented bipartite graphs. Given an oriented bipartite graph G, we consider the problem of building an auxiliary multidigraph H with vertex set as the smaller part of G which can be decomposed into a disjoint union of a collection of transitive bipartite multidigraphs, each with an upper bound on its chromatic index. We formulate this problem as an Integer Linear Programming problem with its number of variables and constraints, though exponential in the size of the smaller part of G, is independent of the size of the larger part. Hence solving this integer linear program gives an FPT algorithm for the original problem.

▶ **Lemma 4.6.** *Let G be an oriented graph with a vertex-cover X. G has a collection of k pairwise arc-disjoint 4-cycles if and only if there exists a collection $\mathcal{P}$ of pairwise arc-disjoint 3-paths in $G[X]$ such that the bipartite digraph $G_{\mathcal{P}}$ obtained from G by deleting all the edges inside X and adding for each $P = (a, b, c)$ in $\mathcal{P}$ a new 3-path (a, b_P, c) (where b_P is a new two-degree vertex) contains k pairwise arc-disjoint 4-cycles.*

Proof. Suppose G has a collection $\mathcal{C}$ of k pairwise arc-disjoint 4-cycles. Since X is a vertex-cover of G, for each $C = (a, b, c, d) \in \mathcal{C}$, the intersection of C with $G[X]$ is either the entire C, a 3-path in C, or two vertices of C. In the first case, add (a, b, c) and (c, d, a) to $\mathcal{P}$. In the second case, add the 3-path in $C \cap G[X]$ to $\mathcal{P}$. In the third case, the two vertices of C from X are non-adjacent and no path is added to $\mathcal{P}$. By the arc-disjointness of $\mathcal{C}$ and the above construction, $\mathcal{P}$ is a collection of pairwise arc-disjoint 3-paths in $G[X]$. In this case it is easy to see that the bipartite digraph $G_{\mathcal{P}}$ constructed for $\mathcal{P}$ also contains k pairwise arc-disjoint 4-cycles.

In the other direction, let $\mathcal{P}$ be a collection of pairwise arc-disjoint 3-paths in $G[X]$ such that the bipartite digraph $G_{\mathcal{P}}$ constructed based on it has a family $\mathcal{C}$ of k pairwise arc-disjoint 4-cycles. Since $G_{\mathcal{P}}$ is bipartite with X as one part, every cycle $C = (a, b, c, d)$ in $\mathcal{C}$ has two non-adjacent vertices, say a and c in X and the other two vertices outside X. This splits C into two 3-length paths (a, b, c) and (c, d, a). Let $\mathcal{P}_{\mathcal{C}}$ denote this collection of $2k$ pairwise arc-disjoint paths obtained from the k cycles in $\mathcal{C}$, each starting and ending in X. We will show that each 3-path in $\mathcal{P}_{\mathcal{C}}$ is either present in G, or it can be substituted by a 3-path in G with the same endpoints, such that the resulting collection $\mathcal{P}'_{\mathcal{C}}$ is pairwise arc-disjoint in G. This would suffice to construct k pairwise arc-disjoint 4-cycles in G. Let (a, b, c) be a 3-path in $\mathcal{P}_{\mathcal{C}}$. If b is a vertex in G, then the same path (a, b, c) is available in G as well and is added to $\mathcal{P}'_{\mathcal{C}}$. If b is a new vertex in $G_{\mathcal{P}}$ but not in G, then b represented a 3-path (a, b', c) in $G[X]$, Add (a, b', c) to $\mathcal{P}'_{\mathcal{C}}$ instead of (a, b, c). Note that the arc-disjointness among the newly added 3-paths is due to the arc-disjointness in $\mathcal{P}$ and their arc-disjointness with the 3-paths from $\mathcal{P}_{\mathcal{C}}$ is since the former is made up of arcs from $G[X]$, while the latter is made up of arcs across X and $V(G) \setminus X$. ◀

If X has ℓ vertices, then there are at most ℓ^3 3-paths in $G[X]$ and any collection $\mathcal{P}$ of pairwise arc-disjoint 3-paths has at most $\ell^2/2$ paths in it. Hence the number of choices for $\mathcal{P}$ is upper bounded by $\binom{\ell^3}{\ell^2/2} \leq (2e\ell)^{\ell^2/2}$. Hence one can use an algorithm to solve Arc Disjoint 4-Cycle Packing in bipartite graphs to solve the same problem for G with a blow-up of $(2e\ell)^{\ell^2/2}$ in running time.

This leads us to the study of parameterized complexity of Arc Disjoint 4-Cycle Packing in bipartite digraphs G with bipartition $L \cup R$ with respect to $\min\{|L|, |R|\}$ as the parameter. Consider an instance $\mathcal{I} = (G, k)$ with $|L| = \min\{|L|, |R|\} = \ell$. Let $L = \{v_1, \ldots, v_\ell\}$ and $R = \{u_1, \ldots, u_n\}$.

▶ **Definition 4.4. (Signature)** *For a vertex $u \in R$, the signature of u is a string $\pi_u = (\pi_u(1), \ldots, \pi_u(\ell))$ in $\{1, -1, 0\}^\ell$ defined as follows.*

$$\pi_u(i) = \begin{cases} 1, & \textit{if } u \in N^+(v_i) \\ -1, & \textit{if } u \in N^-(v_i) \\ 0, & \textit{otherwise} \end{cases}$$

Observe that the signature of vertices in R can be determined in $O(n)$ time and there are at most 3^ℓ distinct signatures. Let $\mathcal{S}$ denote the set of possible signatures. For each $\sigma \in \mathcal{S}$, let R_σ denote the set of vertices of R that have signature equal to σ. For each $\sigma \in \mathcal{S}$, let $\sigma^{-1}(+1)$ denote the set $\{v_i \in L \mid \sigma(i) = 1\}$ and $\sigma^{-1}(-1)$ denote the set $\{v_i \in L \mid \sigma(i) = -1\}$.

Given two multidigraphs $G_1 = (V, A_1)$ and $G_2 = (V, A_2)$ on the same vertex set, we denote by $G_1 \uplus G_2$ the multidigraph obtained by taking the union of G_1 and G_2 after renaming the arcs in A_2 so that A_2 is disjoint from A_1. We call $G_1 \uplus G_2$ as the *disjoint union* of G_1 and G_2 and extend the terminology to any finite collection of graphs on a common vertex set. A multidigraph in which for each pair of vertices u, v the number of arcs directed from u to v is the same as the number of arcs directed from v to u is called *balanced*.

▶ **Lemma 4.7.** *G has a collection of k pairwise arc-disjoint 4-cycles if and only if there exists a collection of bipartite multidigraphs H_σ, $\sigma \in S$, on the vertex set L such that*

(i) *each arc of H_σ is directed from $\sigma^{-1}(+1)$ to $\sigma^{-1}(-1)$,*
(ii) *the edges of H_σ can be properly coloured using $|R_\sigma|$ colours, and*
(iii) *the disjoint union $H = \biguplus_{\sigma \in S} H_\sigma$ is a balanced multidigraph with $2k$ arcs.*

Proof. Suppose G contains an arc-disjoint collection $\mathcal{C}$ of 4-cycles, $|\mathcal{C}| = k$. For each $\sigma \in S$, we set $H_\sigma = (L, A_\sigma)$ where A_σ contains an arc from u to w for each $v \in R_\sigma$ such that (u, v, w) is a path in one of the cycles in $\mathcal{C}$. The first condition in the lemma follows from the definition of signature. The second condition follows since for each $v \in R_\sigma$ the collection of arcs (u, w) in H_σ where (u, v, w) is a path in one of the cycles in $\mathcal{C}$ forms a matching in H_σ. The third condition follows since each 4-cycle (u, v, w, x) in $\mathcal{C}$ (with $u, w \in L$) contributes two opposite arcs, specifically (u, w) and (w, u), to H.

In the opposite direction, suppose we have a collection of bipartite multigraphs H_σ, $\sigma \in S$, on the vertex set L, satisfying the three conditions of the lemma. We first construct a collection $\mathcal{P}$ of pairwise arc-disjoint P_3's in G starting and ending in L as follows. Condition 2 assures that we can decompose the arc-set of each H_σ into at most $|R_\sigma|$ matchings H_v, $v \in R_\sigma$. For each arc (u, w) in H_v, add the directed path (u, v, w) to $\mathcal{P}$. The first condition in the lemma ensures that (u, v, w) is indeed a path in G. Two paths resulting from the same matching H_v, for each $v \in R$, are arc-disjoint because of the disjointness of the end-vertices of the two paths. Two paths resulting from H_v and $H_{v'}$, for each $v, v' \in R, v \neq v'$, are arc-disjoint since their middle vertices are two distinct vertices in R. Hence $\mathcal{P}$ is a collection of arc-disjoint P_3's from G. Condition 3 guarantees that $|\mathcal{P}| = 2k$ and that the P_3's in $\mathcal{P}$ can be perfectly paired to form a collection $\mathcal{C}$ of pairwise arc-disjoint 4-cycles in G, with $|\mathcal{C}| = k$. ◀

The feasibility of finding a collection of bipartite multidigraphs satisfying the conditions of Lemma 4.7 can be checked by the following integer linear program.

Set of Variables: $X = \{x_{\sigma,u,w} \mid \sigma \in S, u \in \sigma^{-1}(-1), w \in \sigma^{-1}(+1)\}$
Feasible Solution: An integral assignment to the variables satisfying the following properties.

- For each signature $\sigma \in \mathcal{S}$ and $u, w \in L$

$$\sum_{w' \colon x_{\sigma,u,w'} \in X} x_{\sigma,u,w'} \leq |R_\sigma| \text{ and } \sum_{u' \colon x_{\sigma,u',w} \in X} x_{\sigma,u',w} \leq |R_\sigma|$$

- For each pair $u, w \in L$

$$\sum_{\sigma \colon x_{\sigma,u,w} \in X} x_{\sigma,u,w} = \sum_{\sigma \colon x_{\sigma,w,u} \in X} x_{\sigma,w,u}.$$

Optimum Solution: A feasible solution that maximizes $\sum_{x_{\sigma,u,w} \in X} x_{\sigma,u,w}$.

Since the edge-chromatic number of bipartite multigraphs is equal to their maximum degree [4], the second condition of Lemma 4.7 is equivalent to ensuring that H_σ has a maximum degree at most $|R_\sigma|$. This is ensured by the first group of constraints. The second group of constraints ensure that the disjoint union H is balanced. The edge-count of H is the cost function to be maximized.

Integer Linear Programming is FPT when parameterized by the number of variables by the following result.

▶ **Proposition 4.3** ([12, 22, 25, 26]). *An integer linear program of size L with p variables can be solved in $\mathcal{O}(p^{2.5p+o(p)}(L + \log M_x)\log(M_x M_c))$ time where M_x is an upper bound on the absolute values a variable can take in a solution and M_c is the largest absolute value of a coefficient in the vector c corresponding to the objective function.*

In our integer linear program, the number of variables p is equal to $|X|$ which is $O(3^\ell \ell^2)$ and the number of constraints is $O(3^\ell \ell)$. So the size L of the integer linear program is $O(3^{2\ell}\ell^3)$. Moreover the maximum value M_x that a variable can take is bounded by $|R_\sigma|$ which is $O(n)$ and all the coefficients are 1 or 0. Hence, Proposition 4.3 gives the following result.

▶ **Theorem 3.** *Arc Disjoint 4-Cycle Packing parameterized by the size of a vertex cover is* FPT.

5 Concluding Remarks

In this work, we studied Arc Disjoint r-Cycle Packing and showed that it is NP-complete on oriented graphs with girth r and remains so for even r when the input is further restricted to be bipartite. For $r = 4$, we gave a cubic kernel (containing a quadratic number of vertices) with respect to the number of cycles as the parameter and showed fixed-parameter tractability with respect to the size of a vertex cover as the parameter. Improving the size of this kernel or showing tightness and giving an FPT algorithm for the problem parameterized by treewidth are interesting future directions. Note that treewidth of a graph is at most the size of its vertex cover. Also, coming up with the best possible polynomial kernels possible for $r > 4$ is a natural next question.

Tournaments and bipartite tournaments are well-studied special classes of digraphs with interesting structural and algorithmic properties. While Arc Disjoint Cycle Packing is known to be NP-complete in tournaments [5], the complexity of this problem in bipartite tournaments is still open. Further, Arc Disjoint 4-Cycle Packing is also open in bipartite tournaments. The NP-completeness of Arc Disjoint Cycle Packing in tournaments was established by a reduction from SAT(1,2) to Arc Disjoint 3-Cycle Packing in tournaments. The construction of the graph in the reduced instance may be viewed as consisting of two phases where in the first phase, an oriented graph is constructed and in the second phase, this graph is completed into a tournament using a decomposition of the edges of a complete (undirected) graph into triangles [27]. The second phase of this approach does not seem to extend to bipartite tournaments. Further, Arc Disjoint 4-Cycle Packing in bipartite tournaments is closely related to a conjecture by Brualdi and Shen [10] that asserts that every Eulerian bipartite tournament has a decomposition of its arcs into 4-cycles. We believe that resolving the complexity of this problem would shed some light on this conjecture.

References

1 Faisal N. Abu-Khzam. An improved kernelization algorithm for r-set packing. *Inf. Process. Lett.*, 110(16):621–624, 2010. doi:10.1016/j.ipl.2010.04.020.

2 Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *J. ACM*, 42(4):844–856, 1995. doi:10.1145/210332.210337.

3 Jørgen Bang-Jensen and Gregory Z. Gutin. *Digraphs – Theory, Algorithms and Applications, Second Edition*. Springer Monographs in Mathematics. Springer, 2009.

4 Claude Berge. *Graphs and hypergraphs*. North-Holland Pub. Co., 1973.

5 Stéphane Bessy, Marin Bougeret, R. Krithika, Abhishek Sahu, Saket Saurabh, Jocelyn Thiebaut, and Meirav Zehavi. Packing arc-disjoint cycles in tournaments. *Algorithmica*, 83(5):1393–1420, 2021. doi:10.1007/s00453-020-00788-2.

6 Stéphane Bessy, Marin Bougeret, and Jocelyn Thiebaut. Triangle packing in (sparse) tournaments: Approximation and kernelization. In Kirk Pruhs and Christian Sohler, editors, *25th Annual European Symposium on Algorithms, ESA 2017, September 4-6, 2017, Vienna, Austria*, volume 87 of *LIPIcs*, pages 14:1–14:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.ESA.2017.14.

7 Hans L. Bodlaender. On disjoint cycles. *Int. J. Found. Comput. Sci.*, 5(1):59–68, 1994. doi:10.1142/S0129054194000049.

8 Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Kernel bounds for path and cycle problems. *Theor. Comput. Sci.*, 511:117–136, 2013. doi:10.1016/j.tcs.2012.09.006.

9 Hans L. Bodlaender, Stéphan Thomassé, and Anders Yeo. Kernel bounds for disjoint cycles and disjoint paths. *Theor. Comput. Sci.*, 412(35):4570–4578, 2011. doi:10.1016/j.tcs.2011.04.039.

10 Richard A. Brualdi and Jian Shen. Disjoint cycles in eulerian digraphs and the diameter of interchange graphs. *J. Comb. Theory, Ser. B*, 85(2):189–196, 2002. doi:10.1006/jctb.2001.2094.

11 Leizhen Cai. Parameterized complexity of vertex colouring. *Discret. Appl. Math.*, 127(3):415–429, 2003. doi:10.1016/S0166-218X(02)00242-1.

12 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

13 Dorit Dor and Michael Tarsi. Graph decomposition is NPC – A complete proof of Holyer's conjecture. In S. Rao Kosaraju, Mike Fellows, Avi Wigderson, and John A. Ellis, editors, *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, May 4-6, 1992, Victoria, British Columbia, Canada*, pages 252–263. ACM, 1992. doi:10.1145/129712.129737.

14 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

15 Paul Erdős and Richard Rado. Intersection theorems for systems of sets. *Journal of the London Mathematical Society*, s1-35(1):85–90, 1960. doi:10.1112/jlms/s1-35.1.85.

16 Michael R. Fellows, Bart M. P. Jansen, and Frances A. Rosamond. Towards fully multivariate algorithmics: Parameter ecology and the deconstruction of computational complexity. *Eur. J. Comb.*, 34(3):541–566, 2013. doi:10.1016/j.ejc.2012.04.008.

17 Michael R. Fellows, Christian Knauer, Naomi Nishimura, Prabhakar Ragde, Frances A. Rosamond, Ulrike Stege, Dimitrios M. Thilikos, and Sue Whitesides. Faster fixed-parameter tractable algorithms for matching and packing problems. *Algorithmica*, 52(2):167–176, 2008. doi:10.1007/s00453-007-9146-y.

18 Michael R. Fellows, Daniel Lokshtanov, Neeldhara Misra, Matthias Mnich, Frances A. Rosamond, and Saket Saurabh. The complexity ecology of parameters: An illustration using bounded max leaf number. *Theory Comput. Syst.*, 45(4):822–848, 2009. doi:10.1007/s00224-009-9167-9.

19 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006. doi:10.1007/3-540-29953-X.

20 Fedor V. Fomin, Tien-Nam Le, Daniel Lokshtanov, Saket Saurabh, Stéphan Thomassé, and Meirav Zehavi. Subquadratic kernels for implicit 3-hitting set and 3-set packing problems. *ACM Trans. Algorithms*, 15(1):13:1–13:44, 2019. doi:10.1145/3293466.

21 Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. *Kernelization: Theory of Parameterized Preprocessing*. Cambridge University Press, 2019.

22 András Frank and Éva Tardos. An application of simultaneous diophantine approximation in combinatorial optimization. *Comb.*, 7(1):49–65, 1987. doi:10.1007/BF02579200.

23 Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

24 Bart M. P. Jansen. *The Power of Data Reduction: Kernels for Fundamental Graph Problems*. PhD thesis, Utrecht University, The Netherlands, 2013.

25 Hendrik W. Lenstra Jr. Integer programming with a fixed number of variables. *Math. Oper. Res.*, 8(4):538–548, 1983. doi:10.1287/moor.8.4.538.

26 Ravi Kannan. Minkowski's convex body theorem and integer programming. *Math. Oper. Res.*, 12(3):415–440, 1987. doi:10.1287/moor.12.3.415.

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

28 Daniel Lokshtanov, Fahad Panolan, M. S. Ramanujan, and Saket Saurabh. Lossy kernelization. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 224–237. ACM, 2017. doi:10.1145/3055399.3055456.

29 Hannes Moser. *Finding optimal solutions for covering and matching problems*. PhD thesis, Friedrich Schiller University of Jena, 2010. URL: https://d-nb.info/999819399.

30 Moni Naor, Leonard J. Schulman, and Aravind Srinivasan. Splitters and near-optimal derandomization. In *36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, USA, 23-25 October 1995*, pages 182–191. IEEE Computer Society, 1995. doi:10.1109/SFCS.1995.492475.

31 Aleksandrs Slivkins. Parameterized tractability of edge-disjoint paths on directed acyclic graphs. *SIAM J. Discret. Math.*, 24(1):146–157, 2010. doi:10.1137/070697781.

32 Craig A. Tovey. A simplified NP-complete satisfiability problem. *Discret. Appl. Math.*, 8(1):85–89, 1984. doi:10.1016/0166-218X(84)90081-7.