Preprocessing to Reduce the Search Space for Odd Cycle Transversal

Bart M. P. Jansen
Eindhoven University of Technology, The Netherlands

Yosuke Mizutani
School of Computing, University of Utah, Salt Lake City, UT, USA

Blair D. Sullivan
School of Computing, University of Utah, Salt Lake City, UT, USA

Ruben F. A. Verhaegh
Eindhoven University of Technology, The Netherlands

Abstract

The NP-hard Odd Cycle Transversal problem asks for a minimum vertex set whose removal from an undirected input graph G breaks all odd cycles, and thereby yields a bipartite graph. The problem is well-known to be fixed-parameter tractable when parameterized by the size k of the desired solution. It also admits a randomized kernelization of polynomial size, using the celebrated matroid toolkit by Kratsch and Wahlström. The kernelization guarantees a reduction in the total *size* of an input graph, but does not guarantee any decrease in the size of the solution to be sought; the latter governs the size of the search space for FPT algorithms parameterized by k. We investigate under which conditions an efficient algorithm can detect one or more vertices that belong to an optimal solution to Odd Cycle Transversal. By drawing inspiration from the popular *crown reduction* rule for Vertex Cover, and the notion of *antler decompositions* that was recently proposed for Feedback Vertex Set, we introduce a graph decomposition called *tight odd cycle cut* that can be used to certify that a vertex set is part of an optimal odd cycle transversal. While it is NP-hard to compute such a graph decomposition, we develop parameterized algorithms to find a set of at least k vertices that belong to an optimal odd cycle transversal when the input contains a tight odd cycle cut certifying the membership of k vertices in an optimal solution. The resulting algorithm formalizes when the search space for the solution-size parameterization of Odd Cycle Transversal can be reduced by preprocessing. To obtain our results, we develop a graph reduction step that can be used to simplify the graph to the point that the odd cycle cut can be detected via color coding.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Theory of computation → Fixed parameter tractability

Keywords and phrases odd cycle transversal, parameterized complexity, graph decomposition, search-space reduction, witness of optimality

Digital Object Identifier 10.4230/LIPIcs.IPEC.2024.15

Related Version *Full Version*: https://arxiv.org/abs/2409.00245 [12]

Funding *Blair D. Sullivan*: Gordon & Betty Moore Foundation under grant GBMF4560.

1 Introduction

The NP-hard Odd Cycle Transversal problem asks for a minimum vertex set whose removal from an undirected input graph G breaks all odd cycles, and thereby yields a bipartite graph. Finding odd cycle transversals has important applications, for example in computational biology [8, 21] and adiabatic quantum computing [6, 7]. Odd Cycle Transversal parameterized by the desired solution size k has been studied intensively, leading to important advances such as *iterative compression* [19] and *matroid-based kernelization* [14, 15]. The randomized kernel due to Kratsch and Wahlström [15, Lemma 7.11] is a polynomial-time algorithm that reduces an n-vertex instance (G, k) of Odd Cycle Transversal to an instance (G', k') on $\mathcal{O}((k \log k \log\log k)^3)$ vertices, that is equivalent to the input instance with probability at least 2^{-n}. Experiments with this matroid-based kernelization, however, show disappointing preprocessing results in practice [18]. This formed one of the motivations for a recent line of research aimed at preprocessing that reduces the *search space* explored by algorithms solving the reduced instance, rather than preprocessing aimed at reducing the *encoding size* of the instance (which is captured by kernelization). To motivate our work, we present some background on this topic.

A *kernelization* of size $f\colon \mathbb{N} \to \mathbb{N}$ for a parameterized problem $\mathcal{P}$ is a polynomial-time algorithm that reduces any parameterized instance (x, k) to an instance (x', k') with the same YES/NO answer, such that $|x'|, k' \leq f(k)$. It therefore guarantees that the size of the instance is reduced in terms of the complexity parameter k. It does not directly ensure a reduction in the *search space* of the follow-up algorithm that is employed to solve the reduced instance. Since the running times of FPT algorithms for the natural parameterization of Odd Cycle Transversal [8, 19, 16] depend exponentially on the size of the sought solution, the size of the search space considered by such algorithms can be reduced significantly by a preprocessing step that finds some vertices S that belong to an optimal solution for the input graph G: the search for a solution of size k on G then reduces to the search for a solution of size $k - |S|$ on $G - S$. Researchers therefore started to investigate in which situations an efficient preprocessing phase can guarantee finding part of an optimal solution.

One line of inquiry in this direction aims at finding vertices that not only belong to an optimal solution, but are even required for building a c-approximate solution [2, 13]; such vertices are called *c-essential*. This has resulted in refined running time guarantees, showing that an optimal odd cycle transversal of size k can be found in time $2.3146^{k-\ell} \cdot n^{\mathcal{O}(1)}$, where ℓ is the number of vertices in the instance that are essential for making a 3-approximate solution [2]. Another line of research, more relevant to the subject of this paper, aims at finding vertices that belong to an optimal solution when there is a simple, locally verifiable certificate of the existence of an optimal solution containing them. So far, the latter direction has been explored for Vertex Cover (where a *crown decomposition* [1, 5] forms such a certificate), and for the (undirected) Feedback Vertex Set problem (where an *antler decomposition* [4]) forms such a certificate.

A *crown decomposition* (see Figure 1) of a graph G consists of a partition of its vertex set into three parts: the *crown* I (which is required to be a non-empty independent set), the *head* H (which is required to contain all neighbors of I), and the *remainder* $R = V(G) \setminus (I \cup H)$, such that the graph $G[I \cup H]$ contains a matching M of size $|H|$. Since I is an independent set, this matching partners each vertex of H with a private neighbor in I. The existence of a crown decomposition shows that there is an optimal vertex cover (a minimum-size vertex set intersecting all edges) that contains all vertices of H and none of I: any vertex cover contains at least $|M| = |H|$ vertices from $I \cup H$ to cover the matching M, while H covers all the edges of G that can be covered by selecting vertices from $I \cup H$. Hence a crown decomposition forms a polynomial-time verifiable certificate that there is an optimal vertex cover containing all vertices of H. It facilitates a reduction in search space for Vertex Cover: graph G has a vertex cover of size k if and only if $G - (I \cup H)$ has one of size $k - |H|$. A crown decomposition can be found in polynomial time if it exists, which yields a powerful reduction rule for Vertex Cover [1].

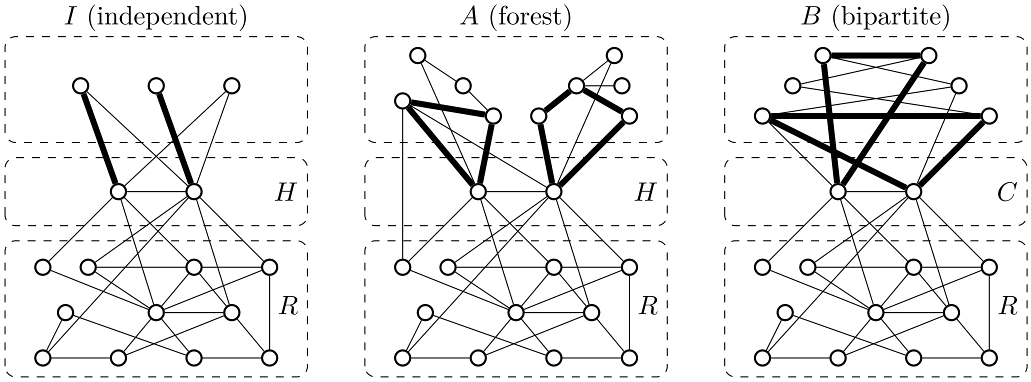

Figure 1 Examples of crown decomposition (left), antler decomposition for Feedback Vertex Set (middle) and a tight OCC for Odd Cycle Transversal (right). Packings of forbidden subgraphs are highlighted in bold.

Inspired by this decomposition for Vertex Cover, Donkers and Jansen [4] introduced the notion of an *antler decomposition* of a graph G. It is a partition of the vertex set into three parts: the *antler* A (which is required to induce a non-empty acyclic graph), the *head* H (which is required to contain *almost* all neighbors of A: for each tree T in the forest $G[A]$, there is at most one edge that connects T to a vertex outside H), and the *remainder* $R = V(G) \setminus (A \cup H)$, while satisfying an additional condition in terms of an integer z that represents the *order* of the antler decomposition. In its simplest form for $z = 1$ (we discuss $z > 1$ later), the additional condition says that the graph $G[A \cup H]$ should contain $|H|$ vertex-disjoint cycles. Since $G[A]$ is acyclic, each of these cycles contains exactly one vertex of H. They certify that any feedback vertex set of G contains at least $|H|$ vertices from $A \cup H$. Since A induces an acyclic graph, and all cycles in G that enter a tree T of $G[A]$ from R must leave A from H, the set H intersects all cycles of G that contain a vertex of $A \cup H$. Hence there is an optimal feedback vertex set containing H. By finding an antler decomposition we can therefore reduce the problem of finding a size-k solution in G to finding a size-$(k - |H|)$ solution in $G - (A \cup H)$, and therefore reduce the search space for algorithms parameterized by solution size.

Donkers and Jansen proved that, assuming $\mathsf{P} \neq \mathsf{NP}$, there unfortunately is no polynomial-time algorithm to find an antler decomposition if one exists [4, Theorem 3.4]. However, they gave a *fixed-parameter tractable* preprocessing algorithm, parameterized by the size of the head. There is an algorithm that, given a graph G and integer k such that G contains an antler decomposition (A, H, R) with $|H| = k$, runs in time $2^{\mathcal{O}(k^5)} \cdot n^{\mathcal{O}(1)}$ and outputs a set of at least k vertices that belong to an optimal feedback vertex set. For each fixed value of k, this yields a preprocessing algorithm to detect vertices that belong to an optimal solution if there is a simple certificate of their membership in an optimal solution.

In fact, Donkers and Jansen gave a more general algorithm; this is where z-antlers for $z > 1$ make an appearance. Recall that for a 1-antler decomposition (A, H, R) of a graph G, the graph $G[A \cup H]$ must contain a collection $\mathcal{C}$ of $|H|$ vertex-disjoint cycles. These cycles certify that the set H is an optimal feedback vertex set in the graph $G[A \cup H]$. In fact, the feedback vertex set H in $G[A \cup H]$ is already optimal for the subgraph $\mathcal{C} \subseteq G[A \cup H]$, and that subgraph $\mathcal{C}$ is structurally simple because each of its connected components (which is a cycle) has a feedback vertex set of size $z = 1$. This motivates the following definition of a z-antler decomposition for $z > 1$: the set H should be an optimal feedback vertex set for the subgraph $G[A \cup H]$, and moreover, there should be a subgraph $\mathcal{C}_z \subseteq G[A \cup H]$ such that (1) H is an optimal feedback vertex set in $\mathcal{C}_z$, and (2) each connected component of $\mathcal{C}_z$ has a feedback vertex set of size at most z. So for a z-antler decomposition (A, H, R) of a graph G, there is a certificate that H is part of an optimal solution in the overall graph G that consists of the decomposition together with the subgraph $\mathcal{C}_z \subseteq G[A \cup H]$ for which H is an optimal solution. The complexity of verifying this certificate scales with z: it comes down to verifying that $H \cap V(C)$ is indeed an optimal feedback vertex set of size at most z for each connected component of the subgraph $\mathcal{C}_z$. Donkers and Jansen presented an algorithm that, given integers $k \geq z \geq 0$ and a graph G that contains a z-antler decomposition whose head has size k, outputs a set of at least k vertices that belongs to an optimal feedback vertex set in time $2^{\mathcal{O}(k^5 z^2)} n^{\mathcal{O}(z)}$. For each fixed choice of k and z, this gives a reduction rule (that can potentially be applied numerous times on an instance) to reduce the search space if the preconditions are met.

Our contribution. We investigate search-space reduction for Odd Cycle Transversal, thereby continuing the line of research proposed by Donkers and Jansen [4]. We introduce the notion of *tight odd cycle cuts* to provide efficiently verifiable witnesses that a certain vertex set belongs to an optimal odd cycle transversal, and present algorithms to find vertices that belong to an optimal solution in inputs that admit such witnesses.

To be able to state our main result, we introduce the corresponding terminology. An *odd cycle cut* (OCC) in an undirected graph G is a partition of its vertex set into three parts: the bipartite part B (which is required to induce a bipartite subgraph of G), the cut part C (which is required to contain all neighbors of B), and the rest $R = V(G) \setminus (B \cup C)$. An odd cycle cut is called *tight* if the set C forms an optimal odd cycle transversal for the graph $G[B \cup C]$. In this case, it is easy to see that there is an optimal odd cycle transversal in G that contains all vertices of C, since all odd cycles through B are intersected by C. A tight OCC (B, C, R) has *order* z if there is a subgraph $\mathcal{C}_z$ of $G[B \cup C]$ for which C is an optimal odd cycle transversal, and for which each connected component of $\mathcal{C}_z$ has an odd cycle transversal of size at most z. This means that for $z = 1$, if there is such a subgraph $\mathcal{C}_z \subseteq G[B \cup C]$, then there is one consisting of $|C|$ vertex-disjoint odd cycles. We use the term z-tight OCC to refer to a tight OCC of order z. Our notion of z-tight OCCs forms an analogue of z-antler decompositions. Note that the requirement that C contains *all* neighbors of B is slightly more restrictive than in the Feedback Vertex Set case. We need this restriction for technical reasons, but discuss potential relaxations in Section 7.

Similarly to the setting of z-antlers for Feedback Vertex Set, assuming $\mathsf{P} \neq \mathsf{NP}$ there is no polynomial-time algorithm that always finds a tight OCC in a graph if one exists; not even in the case $z = 1$. We therefore develop algorithms that are efficient for small k and z. The following theorem captures our main result, which is an OCT-analogue of the antler-based preprocessing algorithm for FVS. The *width* of an OCC (B, C, R) is defined as $|C|$. Our theorem shows that for constant z we can efficiently find k vertices that belong to an optimal solution, if there is a z-tight OCC of width k.

▶ **Theorem 1.** *There is a deterministic algorithm that, given a graph G and integers $k \geq z \geq 0$, runs in $2^{\mathcal{O}(k^{33} z^2)} \cdot n^{\mathcal{O}(z)}$ time and either outputs at least k vertices that belong to an optimal solution for* Odd Cycle Transversal*, or concludes that G does not contain a z-tight OCC of width k.*

One may wonder whether it is feasible to have more control over the output, by having the algorithm output a z-tight OCC (B, C, R) of width k, if one exists. However, a small adaptation of a W[1]-hardness proof for antlers [4, Theorem 3.7] shows that the corresponding algorithmic task is W[1]-hard even for $z = 1$. This explains why the algorithm outputs a vertex set that belongs to an optimal solution, rather than a z-tight OCC.

In terms of techniques, our algorithm combines insights from the previous work on antlers [4] with ideas in the representative-set based kernelization [15] for Odd Cycle Transversal. The global idea behind the algorithm is to repeatedly simplify the graph, while preserving the structure of z-tight OCCs, to arrive at the following favorable situation: if there was a z-tight OCC of width k in the input, then the reduced graph has a z-tight OCC (B, C, R) of the same width that satisfies $|B| \in k^{\mathcal{O}(1)}$. At that point, we can use color coding with a set of $k^{\mathcal{O}(1)}$ colors to ensure that the structure $B \cup C$ gets colored in a way that makes it tractable to identify it. The simplification steps on the graph are inspired by the kernelization for Odd Cycle Transversal and involve the computation of a *cut covering set* of size $k^{\mathcal{O}(1)}$ that contains a minimum three-way $\{X, Y, Z\}$-separator for all possible choices of sets $\{X, Y, Z\}$ drawn from a terminal set T of size $k^{\mathcal{O}(1)}$. The existence of such sets follows from the matroid-based tools of Kratsch and Wahlström [15]. We can avoid the randomization incurred by their polynomial-time algorithm by computing a cut covering set in $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ time deterministically. Compared to the kernelization for Odd Cycle Transversal, a significant additional challenge we face in this setting is that the size of OCTs in the graph can be arbitrarily large in terms of the parameter k. Our algorithm is looking for a *small* region of the graph in which a vertex set exists with a simple certificate for its membership in an optimal solution; it cannot afford to learn the structure of global OCTs in the graph. This local perspective poses a challenge when repeatedly simplifying the graph: we not only have to be careful how these operations affect the total solution size in G, but also how these modifications affect the existence of simple certificates for membership in an optimal solution. This is why our reduction step works with three-way separators, rather than the two-way separators that suffice to solve or kernelize OCT.

Organization. The remainder of this work is organized as follows. The first twelve pages of the manuscript present the key statements and ideas. For statements marked (★), the proof can be found in the full version [12]. After presenting preliminaries on graphs in Section 2, we define (tight) OCCs in Section 3 and explore some of their properties. In Section 4 we show how color coding can be used to find an OCC whose bipartite part is connected and significantly larger than its cut. Given such an OCC, we show in Section 5 how to simplify the graph while preserving the essential structure of odd cycles in the graph. This leads to an algorithm that finds vertices belonging to an optimal solution the presence of a tight OCC in Section 6. Finally, we conclude in Section 7.

2 Preliminaries

Graphs. We only consider finite, undirected, simple graphs. Such a graph G consists of a set $V(G)$ of vertices and a set $E(G) \subseteq \binom{V(G)}{2}$ of edges. For ease of notation, we write uv for an undirected edge $\{u, v\} \in E(G)$; note that $uv = vu$. When it is clear which graph is referenced from context, we write n and m to denote the number of vertices and edges in this graph respectively. For a vertex $v \in V(G)$, its open neighborhood is $N_G(v) := \{u \in V(G) \mid uv \in E(G)\}$ and its closed neighborhood is $N_G[v] := N_G(v) \cup \{v\}$. For a vertex set $S \subseteq V(G)$ we define its open neighborhood as $N_G(S) := (\bigcup_{v \in S} N_G(v)) \setminus S$

and its closed neighborhood as $N_G[S] := \bigcup_{v \in S} N_G[v]$. The subgraph of G induced by a vertex set $S \subseteq V(G)$ is the graph $G[S]$ on vertex set S with edges $\{uv \in E(G) \mid \{u,v\} \subseteq S\}$. We use $G - S$ as a shorthand for $G[V(G) \setminus S]$ and write $G - v$ instead of $G - \{v\}$ for singletons. A *walk* is a sequence of (not necessarily distinct) vertices $(v_1, \ldots, v_k)$ such that $v_i, v_{i+1} \in E(G)$ for each $i \in [k-1]$. The walk is *closed* if we additionally have $v_k, v_1 \in E(G)$. A *cycle* is a closed walk whose vertices are all distinct. The *length* of a cycle $(v_1, \ldots, v_k)$ is k. A *path* is a walk whose vertices are all distinct. The *length* of a path $(v_1, \ldots, v_k)$ is $k - 1$. The vertices v_1, v_k are the *endpoints* of the path. For two (not necessarily disjoint) vertex sets S, T of a graph G, we say that a path $P = (v_1, \ldots, v_k)$ in G is an (S,T)-path if $v_1 \in S$ and $v_k \in T$. If one (or both) of S and T contains only one element, we may write this single element instead of the singleton set consisting of it.

The *parity* of a path or cycle refers to the parity of its length. For a walk $W = (v_1, \ldots, v_k)$, we refer to its vertex set as $V(W) = \{v_1, \ldots, v_k\}$. Observe that if W is a closed walk of odd parity (a *closed odd walk*), then the graph $G[V(W)]$ contains a cycle of odd length (an *odd cycle*): any edge connecting two vertices of $V(W)$ that are not consecutive on W splits the walk into two closed subwalks, one of which has odd length.

For a positive integer q, a *proper q-coloring* of a graph G is a function $f\colon V(G) \to \{0, \ldots, q-1\}$ such that $f(u) \neq f(v)$ for all $uv \in E(G)$. A graph G is *bipartite* if its vertex set can be partitioned into two *partite sets* $L \dot\cup R$ such that no edge has both of its endpoints in the same partite set. It is well-known that the following three conditions are equivalent for any graph G: (1) G is bipartite, (2) G admits a proper 2-coloring, and (3) there is no cycle of odd length in G. An *odd cycle transversal* (OCT) of a graph G is a set $S \subseteq V(G)$ such that $G - S$ is bipartite. An *independent set* is a vertex set S such that $G[S]$ is edgeless. We say that a vertex set X in a graph G *separates* two (not necessarily) disjoint vertex sets S and T if no connected component of $G - X$ simultaneously contains a vertex from S and a vertex from T. For a collection $\{T_1, \ldots, T_m\}$ of (not necessarily disjoint) vertex sets in a graph G, we say that a vertex set X is an *$\{T_1, \ldots, T_m\}$-separator* if X separates all pairs (T_i, T_j) for $i \neq j$. Note that X is allowed to intersect $\bigcup_{i \in [m]} T_i$.

The following lemma gives a simple sufficient condition for a graph to be bipartite.

▶ **Lemma 2.** *Let G be a graph and let $V_L \cup V_0 \cup V_R = V(G)$ be a partition of its vertices such that V_0 is a $\{V_L, V_R\}$-separator. If there exist proper 2-colorings $f_L : (V_0 \cup V_L) \to \{0,1\}$ and $f_R : (V_0 \cup V_R) \to \{0,1\}$ of $G[V_0 \cup V_L]$ and $G[V_0 \cup V_R]$ respectively such that $f_L(v_0) = f_R(v_0)$ for every $v_0 \in V_0$, then G is bipartite.*

Proof. To show that G is bipartite, we provide a proper 2-coloring of the graph. We define this coloring $f\colon V(G) \to \{0,1\}$ such that $f(v_0) = f_L(v_0) (= f_R(v_0))$ for every $v_0 \in V_0$, $f(v_L) = f_R(v_L)$ for every $v_L \in V_L$ and $f(v_R) = f_R(v_R)$ for every $v_R \in V_R$. To see that f is a proper 2-coloring, we show that no edge $e \in E(G)$ is monochromatic under f.

By the assumption that V_0 is a separator, each edge $e \in E(G)$ is contained in $G[V_0 \cup V_L]$ or $G[V_0 \cup V_R]$ (or both). If e is an edge in the former, its endpoints are colored the same as in f_L and are therefore bichromatic. The analogous argument for f_R holds when e is an edge of the latter. ◀

The next lemma captures the main idea behind the iterative compression algorithm [19] (cf. [3, §4.4]) for solving Odd Cycle Transversal. Given a (potentially suboptimal) odd cycle transversal W of a graph, it shows that the task of finding an odd cycle transversal disjoint from W whose removal leaves a bipartite graph with $W_0, W_1 \subseteq W$ in opposite partite sets of its bipartition is equivalent to separating two vertex sets derived from a baseline bipartition of $G - W$. Our statement below is implied by Claim 1 in the work of Jansen and de Kroon [9].

▶ **Lemma 3** ([9, Claim 1]). *Let W be an OCT in graph G. For each partition of $W = W_0 \cup W_1$ into two independent sets, for each proper 2-coloring c of $G - W$, we have the following equivalence for each $X \subseteq V(G) \setminus W$: the graph $G - X$ has a proper 2-coloring with W_0 color 0 and W_1 color 1* if and only if *the set X separates A from R in the graph $G - W$, with:*

$$A = (N_G(W_0) \cap c^{-1}(0)) \cup (N_G(W_1) \cap c^{-1}(1)),$$
$$R = (N_G(W_0) \cap c^{-1}(1)) \cup (N_G(W_1) \cap c^{-1}(0)).$$

Multiway cuts. Let $\mathcal{T} = (T_1, \ldots, T_s)$ be a partition of a set $T \subseteq V(G)$ of *terminal* vertices in an undirected graph G. A *multiway cut* of $\mathcal{T}$ in G is a vertex set $X \subseteq V(G)$ such that for each pair $t_i, t_j \in T \setminus X$ that belong to different parts of partition $\mathcal{T}$, the graph $G - X$ does not contain a path from t_i to t_j. A *restricted* multiway cut of $\mathcal{T}$ is a vertex set X that is a multiway cut for $\mathcal{T}$ such that $X \cap T = \emptyset$, i.e., it does not contain any terminals.

For a positive integer s, a *generalized s-partition* of a set T is a partition $\mathcal{T}^* = (T_0, T_1, \ldots, T_s, T_X)$ of T into $s + 2$ parts, some of which can be empty. The parts T_0 and T_X play a special role, which are the *free* and *deleted* part of $\mathcal{T}^*$, respectively. Let $T' = T_1 \cup \ldots \cup T_s$. A *multiway cut* of $\mathcal{T}^*$ is a (non-restricted) multiway cut in $G - T_X$ of the partition $\mathcal{T} = (T_1, \ldots, T_s)$ of T'. Hence the vertices of T_X are deleted from the graph, while no cut constraints are imposed on the vertices of T_0.

A *minimum multiway cut* of a generalized s-partition $\mathcal{T}^*$ in a graph G is a minimum-cardinality vertex set that satisfies the requirements of a multiway cut for $\mathcal{T}^*$. The following cut covering lemma by Kratsch and Wahlström will be useful for our algorithm.

▶ **Theorem 4** ([15, Theorem 5.14]). *Let G be an undirected graph on n vertices with a set $T \subseteq V(G)$ of terminal vertices, and let $s \in \mathbb{N}$ be a constant. There is a set $Z \subseteq V(G)$ with $|Z| = \mathcal{O}(|T|^{s+1})$ such that Z contains a minimum multiway cut of every generalized s-partition $\mathcal{T}^*$ of T, and we can compute such a set in randomized polynomial time with failure probability $\mathcal{O}(2^{-n})$.*

For a generalized s-partition $\mathcal{T} = (T_0, T_1, \ldots, T_s, T_X)$ of a terminal set $T \subseteq V(G)$ in an undirected graph G, we call a multiway cut X of $\mathcal{T}$ *restricted* if it satisfies $X \cap (\bigcup_{i=1}^{s} T_i) = \emptyset$. Hence a restricted multiway cut does not delete any vertex that is active as a terminal in the generalized partition. A minimum *restricted* multiway cut of $\mathcal{T}$ is a restricted multiway cut whose size is minimum among all restricted multiway cuts.

The following lemma shows that the randomization in the polynomial-time algorithm by Kratsch and Wahlström can be avoided by the use of a single-exponential FPT algorithm, and that the cut covering set can be adapted to work for *restricted* multiway cuts as long as we have a bound on their size.

▶ **Lemma 5** (★). *Let $s \in \mathbb{N}$ be a constant. There is a* deterministic *algorithm that, given an undirected n-vertex graph G and a set $T \subseteq V(G)$ of terminals, runs in time $2^{\mathcal{O}(|T|)} \cdot n^{\mathcal{O}(1)}$ and computes a set $Z \subseteq V(G)$ with $|Z| = \mathcal{O}(|T|^{2s+2})$ with the following guarantee: for each generalized s-partition $\mathcal{T}$ of T, if there is a restricted multiway cut for $\mathcal{T}$ of size at most $|T|$ in G, then the set Z contains a* minimum *restricted multiway cut of $\mathcal{T}$.*

3 Odd Cycle Cuts

In order to extend the "antler" framework of [4] to Odd Cycle Transversal (OCT), we define a problem-specific decomposition which we term *Odd Cycle Cuts* (OCCs). Our decompositions have three parts – a bipartite induced subgraph X_B, a vertex separator X_C (which we call the *head*), and a remainder X_R.

▶ **Definition 6** (Odd Cycle Cut). *Given a graph G, a partition (X_B, X_C, X_R) of $V(G)$ is an* Odd Cycle Cut (OCC) *if (1) $G[X_B]$ is bipartite, (2) there are no edges between X_B and X_R, and (3) $X_C \cup X_B \neq \emptyset$.*

We say $|X_C|$ is the *width* of an OCC, and observe that X_C hits all odd cycles in $G - X_R$. We denote the minimum size of an OCT in G by $\mathsf{oct}(G)$.

▶ **Observation 7.** *If (X_B, X_C, X_R) is an OCC in G, then $|X_C| \geq \mathsf{oct}(G[X_C \cup X_B])$.*

Analogous to z-antlers [4], here we define a *tight OCC* as a special case of an OCC. For a graph G, a set $X_C \subseteq V(G)$ and an integer z, an X_C*-certificate* of order z is a subgraph H of G such that X_C is an optimal OCT of H, and for each component H' of H we have $|X_C \cap V(H')| \leq z$. Throughout the paper, and starting with the following definition, we will use the convention of referring to a tight OCC as (A_B, A_C, A_R) to emphasize its stronger guarantees compared to an arbitrary OCC (X_B, X_C, X_R).

▶ **Definition 8** ((z-)tight OCC). *An OCC (A_B, A_C, A_R) of a graph G is* tight *when $|A_C| = \mathsf{oct}(G[A_C \cup A_B])$. Furthermore, (A_B, A_C, A_R) is a tight OCC of order z (equivalently, z-*tight OCC*) if $G[A_C \cup A_B]$ contains an A_C-certificate of order z.*

Note this definition naturally implies $\mathsf{oct}(G) = |A_C| + \mathsf{oct}(G[A_R])$: the union of A_C with a minimum OCT in $G[A_R]$ forms an OCT for G (since A_C separates A_B from A_R) for which the requirement $|A_C| = \mathsf{oct}(G[A_C \cup A_B])$ guarantees optimality. The main result of this section is that assuming a graph G has a z-tight OCC, there exists a z-tight OCC (A_B, A_C, A_R) such that the number of components in $G[A_B]$ is bounded in terms of z and $|A_C|$. This is an extension of [4, Lemma 4.6], and we defer its proof to the full version of this paper [12].

▶ **Lemma 9** (★). *Let (A_B, A_C, A_R) be a z-tight OCC in a graph G for some $z \geq 0$. There exists a set $A'_B \subseteq A_B$ such that $(A'_B, A_C, A_R \cup A_B \setminus A'_B)$ is a z-tight OCC in G and $G[A'_B]$ has at most $z^2|A_C|$ components.*

Finally, we introduce the notion of an *imposed separation problem* whose solutions naturally correspond to odd cycle transversals of specific subgraphs.

▶ **Definition 10.** *Let (X_B, X_C, X_R) be an OCC of G, and let $f_B \colon X_B \to \{0,1\}$ be a proper 2-coloring of $G[X_B]$. Let $C_1, C_2 \subseteq X_C$ be two disjoint subsets of X_C and let $f_C \colon C_1 \to \{0,1\}$ be a (not necessarily proper) 2-coloring of the vertices in C_1. Based on this 4-tuple of objects (C_1, C_2, f_C, f_B), we define three (potentially overlapping) subsets $A, R, N \subseteq X_B$.*

1. *Let A be the set of vertices $v_b \in X_B$ with a neighbor $v_c \in C_1$ such that $f_B(v_b) = f_C(v_c)$.*
2. *Let R be the set of vertices $v_b \in X_B$ with a neighbor $v_c \in C_1$ such that $f_B(v_b) \neq f_C(v_c)$.*
3. *Finally, let $N := N_G(C_2) \cap X_B$.*

We refer to the problem of finding a smallest $\{A, R, N\}$-separator in $G[X_B]$ as the $\{A, R, N\}$-separation problem imposed onto $G[X_B]$ by (C_1, C_2, f_C, f_B).

To see the connection between solutions and OCTs, one may let C_1 and f_B in this definition correspond to W and c respectively in Lemma 3, while the color classes of f_C correspond to the sets W_0 and W_1 respectively. As shown below in Lemma 11, we can recognize parts of tight OCCs as optimal solutions to specific imposed separation problems.

Although Definition 10 requires f_B and f_C to be colorings of X_B and C_1 respectively, we sometimes abuse the notation by providing colorings whose domains are supersets of these intended domains. In these cases, one may interpret the definition of the imposed separation problem as if given the restrictions of these colorings to their respective intended domains.

One important role of these separation problems is to allow us to characterize intersections of two OCCs when at least one is tight. Specifically, in Lemma 11, we show that the intersection of one OCC's head with the other OCC's bipartite part forms an optimal solution to a specific 3-way separation problem, which is even optimal for a corresponding 2-way problem.

▶ **Lemma 11** (★). *Let (X_B, X_C, X_R) be a (not necessarily tight) OCC in the graph G and let (A_B, A_C, A_R) be a tight OCC in G. Let $f_X : X_B \to \{0,1\}$ and $f_A : A_B \to \{0,1\}$ be proper 2-colorings of $G[X_B]$ and $G[A_B]$ respectively. Let A, R and N be the three sets to be separated in the separation problem imposed onto $G[X_B]$ by $(X_C \cap A_B, X_C \cap A_R, f_A, f_B)$ and let their names correspond to their roles as defined in Definition 10. Then, $A_C \cap X_B$ is both a minimum-size $\{A, R\}$-separator and a minimum-size $\{A, R, N\}$-separator in $G[X_B]$.*

This will prove to be a useful property in Section 5 by which we are able to recognize part of a tight OCC (A_B, A_C, A_R) in an arbitrary graph. We complement it with the statement below, indicating that the intersection $A_C \cap X_B$ is even bounded in size.

▶ **Lemma 12.** *Let (X_B, X_C, X_R) be a (not necessarily tight) OCC in the graph G and let (A_B, A_C, A_R) be a tight OCC in G. Then $|A_C \cap X_B| \le |X_C|$.*

Proof. Suppose for contradiction that $|A_C \cap X_B| > |S|$. Then, $A'_C := (A_C \setminus X_B) \cup (X_C \cap (A_B \cup A_C))$ is a subset of $A_B \cup A_C$ that is strictly smaller than A_C. Now, showing that A'_C is an OCT of $G[A_B \cup A_C]$ contradicts the assumption that A_C is a smallest such OCT by virtue of (A_B, A_C, A_R) being a tight OCC.

To show that A'_C is an OCT of $G[A_B \cup A_C]$, we let F be an arbitrary odd cycle in this graph and show that it intersects A'_C. First, if F intersects X_C, it intersects A'_C in particular, since $X_C \cap (A_B \cup A_C) \subseteq A'_C$.

Otherwise, since X_C separates X_B and X_R in G, F is completely contained in either $G[X_B]$ or $G[X_R]$. The former is not possible, since $G[X_B]$ is bipartite by assumption, so F lives in $G[X_R]$. Furthermore, since F was assumed to live in $G[A_B \cup A_C]$ and $G[A_B]$ is bipartite, F intersects A_C. In particular, as we found F to live in $G[X_R]$, it intersects $A_C \cap X_R$ which is a subset of A'_C by construction. Hence, F intersects A'_C in any case. ◀

4 Finding Odd Cycle Cuts

Our ultimate goal is to show that if the graph contains any tight OCC (X_B, X_C, X_R) with $|X_C| \le k$, then we can produce a tight OCC with $|X_C| \le k$ *and* $|X_B|$ upper-bounded by some function of k. To achieve this, we first show that we can efficiently find some OCC where $|X_B|$ is large enough, and then (in Section 5) that we can reduce any such cut so that $|X_B|$ is small without destroying any essential structure of the input graph.

Specifically, we say an OCC (X_B, X_C, X_R) is *reducible* with respect to some function g_r if $|X_B| > g_r(|X_C|)$. Our results all hold for a specific polynomial $g_r(x)$ in $\Theta(x^{16})$. Its definition relies on Lemma 5 in which sets Z and T are specified. Setting the value of s in this lemma to 3 yields the existence of a constant $c \in \mathbb{N}$ such that $|Z| \le c \cdot |T|^8$ for large enough $|T|$. Given this constant c, we define $g_r : \mathbb{N} \to \mathbb{N}$ as $g_r(x) = (6(2^8c+1)^2 + 2^8c) \cdot x^{16}$.

We say an OCC (X_B, X_C, X_R) is a *single-component* OCC if $G[X_B]$ is connected. Given a graph G, our goal is to output a reducible OCC efficiently assuming that G contains a single-component OCC (X_B, X_C, X_R) with $|X_B| > g_r(2|X_C|)$ and $|X_C| \le k$. We achieve this by color coding of the vertices in G (see the full version [12] for details). Consider a coloring $\chi : V(G) \to \{\dot{\mathsf{B}}, \dot{\mathsf{C}}\}$. For an integer ℓ, an OCC (X_B, X_C, X_R) with $|X_B| \ge \ell$ is *ℓ-properly* *colored* by χ if $X_C \subseteq \chi^{-1}(\dot{\mathsf{C}})$ and there is a set of ℓ vertices of X_B that are colored $\dot{\mathsf{B}}$ and induce a connected subgraph of G. First, we show how to construct an OCC with large X_B from a proper coloring.

▶ **Lemma 13** (★)**.** *Given a graph G, integers k, ℓ, and a coloring $\chi\colon V(G) \to \{\dot{\mathsf{B}}, \dot{\mathsf{C}}\}$ of $V(G)$ that ℓ-properly colors a single-component OCC (X_B, X_C, X_R) with $|X_C| \leq k$, an OCC (X'_B, X'_C, X'_R) such that $|X'_B| \geq \ell$ and $|X'_C| \leq 2k$ can be found in polynomial time.*

Proof sketch. We iterate over the connected components of $G[\chi^{-1}(\dot{\mathsf{B}})]$. For any component which is both large ($\geq \ell$) and bipartite, we try to find an OCC of small enough width where the component is contained in the X_B side of the cut. To do this, we use the machinery of bipartite separations introduced in Jansen et al. [11] (see the full version [12] for details). Intuitively, given a vertex set C which induces a connected bipartite subgraph, they either find a set of at most $2k$ vertices which separates $C' \supseteq C$ from the remainder of the graph so that $G[C']$ is bipartite, or certify that C is not part of X_B for any OCC with width $\leq k$. ◀

Now, we use this coloring scheme to find a reducible OCC, assuming that a graph G has a single-component OCC (X_B, X_C, X_R) with large X_B.

▶ **Lemma 14.** *There exists a $2^{\mathcal{O}(k^{16})}n^{\mathcal{O}(1)}$-time algorithm that, given a graph G and an integer k, either determines that G does not contain a single-component OCC (X_B, X_C, X_R) of width at most k with $|X_B| > g_r(2k)$ or outputs a reducible OCC in G.*

Proof. We will invoke the algorithm from Lemma 13 multiple times for $\ell = g_r(2k) + 1$. If we supply a coloring that ℓ-properly colors (X_B, X_C, X_R), then the algorithm is guaranteed to find an OCC (X'_B, X'_C, X'_R) such that $|X'_B| > g_r(2k)$ and $|X'_C| \leq 2k$, which is reducible as $|X'_B| > g_r(2k) \geq g_r(|X'_C|)$. If all relevant colorings fail to find such a reducible OCC, then we can conclude that G does not contain a single-component OCC (X_B, X_C, X_R) with $|X_C| \leq k$ and $|X_B| \geq \ell > g_r(2k)$.

Let $X'_B \subseteq X_B$ be an arbitrary vertex set of size ℓ that induces a connected subgraph of G. Since $G[X_B]$ is connected, such X'_B must exist. Observe that we obtain an ℓ-proper coloring if $X_C \cup X'_B$ are colored correctly. Let $s = |X_C \cup X'_B| = k + g_r(2k) + 1 = \mathcal{O}(k^{16})$.

Using an (n, k)-universal set, which is a well-known pseudorandom object [17, 3] used to derandomize applications of color coding (see [3, Theorem 5.20]), we can construct a family of $2^{\mathcal{O}(s)} \log n$ many subsets $A_1, \ldots, A_{2^{\mathcal{O}(s)} \log n}$ with the guarantee that for each set $S \subseteq V(G)$ of size s, for each subset S' of S, there exists a set in the family with $A_i \cap S = S'$. This can be done in $2^{\mathcal{O}(s)} n \log n = 2^{\mathcal{O}(k^{16})} n \log n$ time. From this family, we can construct a family of colorings that is guaranteed to include one that ℓ-properly colors a suitable OCC (X_B, X_C, X_R) if one exists. To derive a coloring χ_i from a member $A_i \subseteq V(G)$ of the (n, s)-universal set, it suffices to pick $\chi(a \in A) = \dot{\mathsf{R}}$ and $\chi(a \notin A) = \dot{\mathsf{B}}$.

We run the $n^{\mathcal{O}(1)}$-time algorithm from Lemma 13 for each coloring, which results in the overall runtime $2^{\mathcal{O}(k^{16})} n^{\mathcal{O}(1)}$. ◀

5 Reducing Odd Cycle Cuts

Given an OCC (X_B, X_C, X_R) of G with $|X_B| > g_r(|X_C|)$, the next step is to "shrink" X_B in a way that preserves some of the structure of the input graph. In this section, we give a reduction to do this and prove that it preserves the general structure of minimum-size OCTs and of tight OCCs in the graph. The reduction starts with a marking scheme that is discussed separately in Section 5.1. We give the full reduction, which includes this marking scheme as a subroutine, in Section 5.2. The reduction will only affect $G[X_B]$ and the edge set between X_B and X_C, which already ensures that an important part of the input graph is maintained.

5.1 A marking scheme for the reduction

The goal of the marking scheme is to mark a set $B^* \subseteq X_B$ of size $|X_C|^{\mathcal{O}(1)}$ as "interesting" vertices that the reduction should not remove or modify. Intuitively, we want this set to contain vertices which we expect might be part of the cut part of a tight OCC in G. More precisely, we guarantee that for every tight OCC in G there is a (possibly different) tight OCC (A_B, A_C, A_R) such that $A_C \cap X_B$ is contained in the marked set B^*.

As seen in Lemma 11, for every tight OCC (A_B, A_C, A_R) in the graph, the intersection $A_C \cap X_B$ forms an optimal solution to a specific imposed separation problem (Definition 10). As such, it suffices if B^* is a *cut covering set* for these imposed separation problems.

Indeed, the key ingredient of the algorithm presented below is the computation of such a cut covering set. Preceding this computation is a graph reduction ensuring that the computed set covers precisely the imposed separation problems. In Lemma 5, we will show that a cut covering set can be computed in deterministic FPT time parameterized by the size of the terminal set, which leads to a total running time of $2^{\mathcal{O}(|X_C|)} n^{\mathcal{O}(1)}$ time for the marking step.

▶ **Marking step.** *Consider the following algorithm.*
Input: A graph G and an OCC (X_B, X_C, X_R) of G.
Output: Marked vertices $B^* \subseteq X_B$.

1. *Find a proper 2-coloring $f_X : X_B \to \{0,1\}$ of $G[X_B]$.*
2. *Construct an auxiliary (undirected) graph G', initialized to a copy of $G[X_B]$. For each $v \in X_C$, do the following:*
   - *Add vertices $v^{(0)}$ and $v^{(1)}$ to G'.*
   - *For each neighbor $u \in N_G(v) \cap X_B$, add an edge $v^{(f_X(u))}u$.*

   Let T be the set $\{v^{(i)} \mid v \in X_C, i \in \{0,1\}\}$. Note that $|T| = 2|X_C|$.
3. *Compute a cut covering set $B^* \subseteq V(G')$ via Lemma 5 such that for every partition $\mathcal{T}^* = (T_0, T_1, T_2, T_3, T_X)$ of T, the set B^* contains a minimum-size solution to the following problem:*
   - *find a vertex set $S \subseteq V(G') \setminus (T_1 \cup T_2 \cup T_3)$ such that S separates T_i and T_j in the graph $G' - T_X$ for all $1 \le i < j \le 3$,*

   as long as this problem has a solution of size at most $|T|$.

▶ **Lemma 15** (★). *Let B^* be constructed as in the Marking step when given the graph G and an OCC (X_B, X_C, X_R) of G as input. If there exists a z-tight OCC (A_B, A_C, A_R) in G, then there exists a z-tight OCC (A^*_B, A^*_C, A^*_R) in G with $|A^*_C| = |A_C|$ and with $A^*_C \cap X_B \subseteq B^*$.*

Proof sketch. Let $f_X : X_B \to \{0,1\}$ be the 2-coloring obtained in step 1 of the Marking step, let $f_A : A_B \to \{0,1\}$ be a proper 2-coloring of $G[A_B]$ and consider the separation problem imposed onto $G[X_B]$ by $(X_C \cap A_B, X_C \cap A_R, f_A, f_B)$. Let A, R and N be the three sets to be separated in this problem with their names corresponding to their roles as in Definition 10.

By putting the correct copy of each vertex from $A_B \cap X_C$ into T_1 and T_2 respectively, putting both copies of vertices from $A_R \cap X_C$ into T_3 and putting both copies of vertices from $A_C \cap X_C$ into T_X, we obtain a partition $(\emptyset, T_1, T_2, T_3, T_X)$ of the set T defined in step 2, such that the corresponding separation problem has the same solution space as the $\{A, R, N\}$-separation problem imposed onto $G[X_B]$. By construction of B^* in step 3, there is a set $S \subseteq B^*$ (possibly different from $A_C \cap X_B$) that is an optimal $\{A, R, N\}$-separator in $G[X_B]$. To construct the tight OCC (A^*_B, A^*_C, A^*_R), we use this set S as replacement for $A_C \cap X_B$, which is also a minimum-size $\{A, R, N\}$-separator in $G[X_B]$ by Lemma 11.

As such, we define $A_C^* := (A_C \setminus X_B) \cup S$. To define A_B^*, let U be the set of vertices from $X_B \setminus S$ that are not reachable from N in $G[X_B] - S$. Now, we define $A_B^* := (A_B \setminus X_B) \cup U$. Finally, we define $A_R^* := V(G) \setminus (A_B^* \cup A_C^*)$. Clearly, this 3-partition of $V(G)$ satisfies the constraints $|A_C^*| = |A_C|$ and $A_C^* \cap X_B \subseteq B^*$. We proceed by showing that it satisfies the three additional properties required to be a z-tight OCC.

First, to see that $G[A_B^*]$ is bipartite, we note that A_B^* only contains vertices from A_B and $X_B \setminus S$. Both are vertex sets that induce a bipartite subgraph. Then, noting the correspondence between the sets A and R obtained from the separation problem and the sets A and R as in Lemma 3, we invoke this lemma on $G[(X_C \cap A_B) \cup X_B]$ with $c = f_A$ and with W_0 and W_1 being the two color classes of this coloring restricted to $X_C \cap A_B$. It follows that the vertices from A_B^* in $X_B \setminus S$ can be properly 2-colored by a coloring f that agrees with f_A on the vertex set $X_C \cap A_B$ that separates $A_B^* \cap X_R$ and $A_B^* \cap X_B$. As these two vertex sets are properly colored by f_A and f respectively, these colorings combine to a proper 2-coloring of the entire graph $G[A_B^*]$ (see Lemma 2).

Secondly, a case distinction shows that there are no edges between A_B^* and A_R^*. It combines the fact that $A_C \cap X_B$ is an $\{A, R, N\}$-separator in $G[X_B]$ – thereby in particular separating $A_B \cap X_B$ from N in $G[X_B]$ – and the fact that A_B^* only contains vertices that already belonged to A_B and vertices from X_B that are not reachable from N in $G[X_B] - A_C^*$.

Finally, it remains to show that (A_B^*, A_C^*, A_R^*) has an A_C^*-certificate of order z. To prove this, we show that the order-z certificate D of the original OCC (A_B, A_C, A_R) is also an order-z certificate in (A_B^*, A_C^*, A_R^*). The main effort here is to prove that D even lives in $A_B^* \cup A_C^*$, after which it is easy to see that it is also an order-z certificate for our new OCC.

As Lemma 11 guarantees that $A_C \cap X_B$ is not only an optimal $\{A, R, N\}$-separator in $G[X_B]$ but even an optimal $\{A, R\}$-separator in this graph, it contains exactly one vertex from every path of a maximum packing $\mathcal{P}$ of pairwise vertex-disjoint (A, R)-paths in $G[X_B]$, due to Menger's theorem [20, Theorem 9.1]. Likewise, $A_C^* \cap X_B = S$ is also an optimal $\{A, R\}$-separator in $G[X_B]$ and hence also contains exactly one vertex from every path of $\mathcal{P}$.

Intuitively, for any path $P \in \mathcal{P}$, a vertex on this path that stops being reachable from one endpoint of P when sliding the picked vertex along the path, starts becoming reachable from the other endpoint of P. As both endpoints of P belong to $A \cup R$ and S only differs from $A_C \cap X_B$ by which vertex is picked from each path in $\mathcal{P}$, it cannot drastically alter which vertices are reachable from $A \cup R$, which in turn are all vertices that end up in A_B^*.

Using the observation that A_B and A_B^* are separated from N by A_C and A_C^* respectively, we see that all vertices that *are* disconnected from $A \cup R$ by substituting $A_C \cap X_B$ for S are in particular also disconnected from N. Thereby, these vertices end up in A_B^*. This shows that $(A_B \cup A_C) \subseteq (A_B^* \cup A_C^*)$, which implies that the certificate D also lives in the latter. ◀

5.2 Simplifying the graph

Our eventual reduction starts with the Marking step from the previous section, after which the graph is modified in a way that leaves marked vertices untouched. We want the reduction to preserve the general structure of optimal OCTs and tight OCCs in the input graph. As this is governed by the locations and interactions of odd cycles in the graph, we encode this information in a more space-efficient manner using the following reduction.

▶ **Reduction step.** *Given a graph G and an OCC (X_B, X_C, X_R) of it, we construct a graph G' as follows.*

1. *Use the Marking step with input G and (X_B, X_C, X_R) to obtain the set $B^* \subseteq X_B$.*
2. *Initialize G' as a copy of $G - (X_B \setminus B^*)$.*

3. *For every $u, v \in X_C \cup B^*$ and for every parity $p \in \{even, odd\}$, check if the subgraph $G[X_B \setminus B^*]$ contains the internal vertex of a (u, v)-path with parity p. If so, then:*
- *if $p = even$, add two new vertices x and x' to G and connect both of them to u and v.*
- *if $p = odd$, add four new vertices x, y, x' and y' to G and add the edges $\{u, x\}$, $\{x, y\}$, $\{y, v\}$, $\{u, x'\}$, $\{x', y'\}$ and $\{y', v\}$.*

Note that we explicitly allow $u = v$ in this step.

Effectively, this reduction deletes the vertices $X_B \setminus B^*$ from the graph. For each pair of neighbors u, v from that set, if the deleted vertices provided an odd (resp. even) path between them, then we insert two vertex-disjoint odd (resp. even) paths between u and v. Hence we shrink the graph while preserving the parity of paths provided by the removed vertices.

As we prove in the full version of this paper [12], the reduction can be performed in $2^{\mathcal{O}(|X_C|)} \cdot n^{\mathcal{O}(1)}$ time and it is guaranteed to output a strictly smaller graph than its input graph whenever it receives an OCC that is reducible with respect to the function g_r as in Section 4. To show that the reduction also preserves OCT and OCC structures, we prove that it satisfies two safety properties formalized below in Lemmas 16 and 17.

▶ **Lemma 16** (★)**.** *Let G be a graph, let (X_B, X_C, X_R) be an OCC in G and let G' be the graph obtained by running the Reduction step with these input parameters. For all $z \geq 0$, if there exists a z-tight OCC (A_B, A_C, A_R) in G, then there exists a z-tight OCC (A'_B, A'_C, A'_R) in G' with $|A'_C| = |A_C|$.*

The proof of the lemma above uses Lemma 15 to infer that, for any z-tight OCC (A_B, A_C, A_R) of G, the graph G also contains a z-tight OCC (A^*_B, A^*_C, A^*_R) of the same width such that $A^*_C \subseteq V(G) \cap V(G')$. This allows for the construction of an OCC (A'_B, A'_C, A'_R) in G' with $A'_C = A^*_C$. Then, A'_B can be defined as the union of $A^*_B \cap V(G) \cap V(G')$ and the set of vertices that were added during the reduction to provide a replacement connection between any two vertices from $A^*_C \cup (A^*_B \cap V(G) \cap V(G'))$. Finally, $A'_R := V(G') \setminus (A'_B \cup A'_C)$.

The proof proceeds to show that the resulting partition (A'_B, A'_C, A'_R) is a z-tight OCC of G'. The two main insights used to prove this are the facts that:
- optimal OCTs of G' are disjoint from the set of newly added vertices $V(G') \setminus V(G)$, and
- odd cycles in G can be translated to very similar odd cycles in G' and vice versa.

These insights are also covered in the proof sketch of the second safety property below.

▶ **Lemma 17** (★)**.** *Let G be a graph, let (X_B, X_C, X_R) be an OCC in G and let G' be the graph obtained by running the Reduction step with these input parameters. If S' is a minimum-size OCT of G', then $S' \subseteq V(G) \cap V(G')$ and S' is a minimum-size OCT of G.*

Proof sketch. To see that $S' \subseteq V(G) \cap V(G')$, we show that S' contains none of the newly added vertices in $V(G') \setminus V(G)$. These newly added vertices come in pairs that form degree-2 paths connecting the same endpoints. Consider two such paths and let u and v be the endpoints of both of them. Suppose for contradiction that S' uses an internal vertex p_1 from one path to break an odd cycle F. Then it must also contain an internal vertex p_2 from the other path to break the odd cycle obtained by swapping one path for the other in F. As both these cycles also pass through u and v by construction, substituting p_1 and p_2 for one of u and v in S' yields a strictly smaller solution. This contradicts the optimality of S'.

To see that S' is an OCT of G, suppose for contradiction that $G - S'$ contains an odd cycle F. Every subpath of F that connects two vertices from $V(G) \cap V(G')$ via a path whose internal vertices lie in $G - V(G')$ can be replaced by one of the paths inserted during the construction of G', with the same endpoints and parity. Substituting every subpath of F that is absent in G' for such a replacement path yields a closed odd walk in $G' - S'$; but this contradicts the fact that S' is an OCT of G'. Hence $S' \subseteq V(G) \cap V(G')$ is an OCT in G.

It remains to show that S' is an OCT of minimum size. Suppose for contradiction that T is a strictly smaller OCT of G. We start by showing how to modify T into an OCT S of G that is at most as large and lives in $V(G) \cap V(G')$. To this end, let $f\colon V(G) \setminus T \to \{0,1\}$ and $f_X\colon X_B \to \{0,1\}$ be proper 2-colorings of $G - T$ and $G[X_B]$ respectively and consider the separation problem imposed onto $G[X_B]$ by $(X_C \setminus T, \emptyset, f, f_B)$. Let A, R, N be the three sets to be separated in this problem with their names corresponding to their roles as in Definition 10. Since the second argument C_2 in the 4-tuple is $\emptyset$, we obtain $N = \emptyset$.

The fact that $N = \emptyset$ ensures that the separation problem above is merely a 2-way separation problem between the sets A and R in $G[X_B]$. These sets are defined in such a way that, for a suitable choice of input parameters to Lemma 3, they coincide with the sets A and R in this lemma. Applying the lemma in one direction to $G[(X_C \setminus T) \cup X_B]$ with $c = f$ and with W_0 and W_1 being the two color classes of this coloring restricted to $X_C \setminus T$, yields that $T \cap X_B$ is an $\{A, R\}$-separator in $G[X_B]$. Applying it in the other direction yields that the removal of *any* $\{A, R\}$-separator T' from G allows for a proper 2-coloring f' of $G[(X_C \setminus T) \cup (X_B \setminus T')]$ that agrees with the coloring f on the vertex set $X_C \setminus T$. As this set separates the subgraphs $G[X_R \setminus T]$ and $G[X_B \setminus T']$ in $G - ((T \setminus X_B) \cup T')$ and these subgraphs are properly 2-colored by f and f' respectively, those two colorings combine to properly 2-color $G - ((T \setminus X_B) \cup T')$ (see Lemma 2). By construction of B^*, there is a minimum-size $\{A, R\}$-separator T^* in $G[X_B]$ with $T^* \subseteq B^*$. Hence, $S := (T \setminus X_B) \cup T^*$ is an OCT of G that lives in $V(G) \cap V(G')$. Furthermore, since T^* is a minimum-size $\{A, R\}$-separator in $G[X_B]$ and it replaces $T \cap X_B$, which is also an $\{A, R\}$-separator, we find that $|S| \leq |T| < |S'|$.

Since S' was assumed to be a minimum-size OCT of G', the smaller set S is not an OCT of G'. Therefore, $G' - S$ contains an odd cycle F'. The argument used before to convert an odd cycle in G to one in G' can also be used in the reverse direction to construct an odd cycle F in $G - S$ from F'. The existence of this cycle contradicts the assumption that S is an OCT of G, which concludes the proof by showing that S' is a minimum-size OCT of G. ◀

6 Finding and Removing Tight OCCs

Now we find tight OCCs by the same color coding technique used in previous work [4]. Consider a coloring $\chi\colon V(G) \cup E(G) \to \{\dot{\mathsf{B}}, \dot{\mathsf{C}}, \dot{\mathsf{R}}\}$ of the vertices and edges of a graph G. For every color $c \in \{\dot{\mathsf{B}}, \dot{\mathsf{C}}, \dot{\mathsf{R}}\}$, let $\chi_V^{-1}(c) = \chi^{-1}(c) \cap V(G)$. For any integer $z \geq 0$, a z-tight OCC (A_B, A_C, A_R) is *z-properly colored* by a coloring χ if all the following hold: (i) $A_C \subseteq \chi_V^{-1}(\dot{\mathsf{C}})$, (ii) $A_B \subseteq \chi_V^{-1}(\dot{\mathsf{B}})$, and (iii) for each component H of $G' = G[A_B \cup A_C] - \chi^{-1}(\dot{\mathsf{R}})$ we have $\mathsf{oct}(H) = |A_C \cap V(H)|$ and $|A_C \cap V(H)| \leq z$. Note that $\chi^{-1}(\dot{\mathsf{R}})$ may include both vertices and edges, so that the process of obtaining G' involves removing both the vertices and edges colored $\dot{\mathsf{R}}$. By a straight-forward adaptation of the color coding approach from previous work [4, Lemma 6.2], we can reconstruct a tight OCC from a proper coloring.

▶ **Lemma 18** (★). *There is an $n^{\mathcal{O}(z)}$ time algorithm taking as input an integer $z \geq 0$, a graph G, and a coloring $\chi\colon V(G) \cup E(G) \to \{\dot{\mathsf{B}}, \dot{\mathsf{C}}, \dot{\mathsf{R}}\}$ that either determines that χ does not z-properly color any z-tight OCC, or outputs a z-tight OCC (A_B, A_C, A_R) in G such that for each OCC $(\hat{A}_B, \hat{A}_C, \hat{A}_R)$ that is z-properly colored by χ, we have $\hat{A}_B \subseteq A_B$ and $\hat{A}_C \subseteq A_C$.*

Combining all ingredients in the previous sections leads to a proof of the main theorem.

▶ **Theorem 1.** *There is a deterministic algorithm that, given a graph G and integers $k \geq z \geq 0$, runs in $2^{\mathcal{O}(k^{33} z^2)} \cdot n^{\mathcal{O}(z)}$ time and either outputs at least k vertices that belong to an optimal solution for* Odd Cycle Transversal, *or concludes that G does not contain a z-tight OCC of width k.*

Proof sketch. Given an input graph G, we repeatedly invoke Lemma 14 to find a reducible OCC and use the Reduction step to shrink it. When we stabilize on a graph G', Lemma 16 guarantees that G' contains a z-tight OCC of width $|A'_C| = k$ if G had one. By Lemma 9, there is such a z-tight OCC (A'_B, A'_C, A'_R) in G' for which $G'[A'_B]$ has at most z^2k components. As each such component gives rise to a single-component OCC, none of them are large enough to be reducible. Hence $|A'_C \cup A'_B| \in (zk)^{\mathcal{O}(1)}$. Hence we can deterministically construct a family of $2^{(kz)^{\mathcal{O}(1)}} n^{\mathcal{O}(1)}$ colorings that includes one that properly colors (A'_B, A'_C, A'_R). Invoking Lemma 18 with such a coloring identifies a z-tight OCC in G' whose head A^*_C contains A'_C and therefore has size at least k. Then A^*_C is contained in an optimal OCT in G', so that Lemma 17 ensures A^*_C belongs to an optimal OCT in G. We output A^*_C. ◀

7 Conclusion

Inspired by crown decompositions for Vertex Cover and antler decompositions for Feedback Vertex Set, we introduced the notion of (tight) odd cycle cuts to capture local regions of a graph in which a simple certificate exists for the membership of certain vertices in an optimal solution to Odd Cycle Transversal. In addition, we developed a fixed-parameter tractable algorithm to find a non-empty subset of vertices that belong to an optimal odd cycle transversal in input graphs admitting a tight odd cycle cut; the parameter k we employed is the *width* of the tight OCC. Finding tight odd cycle cuts and removing the vertices certified to be in an optimal solution leads to search-space reduction for the natural parameterization of Odd Cycle Transversal. To obtain our results, one of the main technical ideas was to replace the use of minimum two-way separators that arise naturally when solving Odd Cycle Transversal, by minimum three-way separators that simultaneously handle breaking the odd cycles in a subgraph *and* separating the resulting local bipartite subgraph from the remainder of the graph.

Theoretical challenges. There are several interesting directions for follow-up work. We first discuss the theoretical challenges. The algorithm we presented runs in time $2^{k^{\mathcal{O}(1)}} n^{\mathcal{O}(z)}$, where z is the order of the tight odd cycle cut in the output guarantee of Theorem 1. The polynomial term in the exponent has a large degree, which is related to the size of the cut covering sets used to shrink the bipartite part of an odd cycle cut in terms of its width. While we expect that some improvements can be made by a more refined analysis, it would be more interesting to see whether an algorithmic approach that avoids color coding can lead to significantly faster algorithms.

An odd cycle cut (X_B, X_C, X_R) of width $|X_C| = k$ in a graph G gives rise to a k-secluded bipartite subgraph $G[X_B]$; recall that a subgraph is called k-secluded if its open neighborhood has size k. For enumerating inclusion-maximal *connected* k-secluded subgraphs that satisfy a property Π, a bounded-depth branching strategy was recently proposed [10] that generalizes the enumeration of important separators. Can such branching techniques be used to improve the running time for the search-space reduction problem considered in this paper to $2^{\mathcal{O}(k)} n^{\mathcal{O}(z)}$?

The dependence on the complexity z of the certificate is another topic for further investigation. The search-space reduction algorithm for Feedback Vertex Set by Donkers and Jansen [4] that inspired this work, also incurs a factor $n^{\mathcal{O}(z)}$ in its running time. For Feedback Vertex Set, it is conjectured but not proven that such a dependence on z is unavoidable. The situation is the same for Odd Cycle Transversal. Is there a way to rule out the existence of an algorithm for the task of Theorem 1 that runs in time $f(k, z) \cdot n^{\mathcal{O}(1)}$?

A last theoretical challenge concerns the definition of the substructures that are used to certify membership in an optimal odd cycle transversal. Our definition of an odd cycle cut (X_B, X_C, X_R) prohibits the existence of any edges between X_B and X_R. Together with the requirement that $G[X_B]$ is bipartite, this ensures that all odd cycles intersecting X_B are intersected by X_C. In principle, one could also obtain the latter conclusion from a slightly less restricted graph decomposition. Since any odd cycle enters a bipartite subgraph on one edge and leaves via another, knowing that each connected component H of $G[X_B]$ is connected to X_R by at most one edge is sufficient to guarantee that all odd cycles visiting X_B are intersected by X_C. The prior work on antler structures for Feedback Vertex Set allows the existence of one pendant edge per component, and manages to detect such antler structures efficiently. It would be interesting to see whether our approach can be generalized for *relaxed* odd cycle cuts in which each component of $G[X_B]$ has at most one edge to X_R. To adapt to this setting, one would have to refine the type of three-way separation problem that is used in the graph reduction step.

For Odd Cycle Transversal, one could relax the definition of the graph decomposition even further: to ensure that odd cycles visiting X_B are intersected by X_C, it would suffice for each connected component H of $G[X_B]$ to have at most one neighbor v_H in X_R, as long as all vertices of H adjacent to v_H belong to the *same* side of a bipartition of H.

Practical challenges. Since the investigation of search-space reduction is inspired by practical considerations, we should not neglect to discuss practical aspects of this research direction. While we do not expect the algorithm as presented here to be practical, it serves as a proof of concept that rigorous guarantees on efficient search-space reduction can be formulated. Our work also helps to identify the types of substructures that can be used to reason locally about membership in an optimal solution. Apart from finding faster algorithms in theory and experimenting with their results, one could also target the development of specialized algorithms for concrete values of k and z.

For $k = 1$, a tight odd cycle cut of width 1 effectively consists of a cutvertex c of the graph whose removal splits off a bipartite connected component B but for which the subgraph induced by $B \cup \{v\}$ contains an odd cycle. Preliminary investigations suggest that in this case, an algorithm that computes the block-cut tree, analyzes which blocks form non-bipartite subgraphs, and which cut vertices break all the odd cycles in their blocks, can be engineered to run in time $\mathcal{O}(|V(G)| + |E(G)|)$ to find a vertex v belonging to an optimal odd cycle transversal when given a graph that has a tight odd cycle cut of width $k = 1$. Do linear-time algorithms exist for $k > 1$? These would form valuable reduction steps in algorithms solving Odd Cycle Transversal exactly, such as the one developed by Wernicke [21].

The $k = 1$ case of the *relaxed* odd cycle cuts described above are in fact used as one of the reduction rules in Wernicke's algorithm [21, Rule 7]. His reduction applies whenever there is a triangle $\{u, v, w\}$ in which w has degree two and v has degree at most three. Under these circumstances, there is an optimal solution that contains u while avoiding v and w: since the removal of u decreases the degree of w to one, while w is one of the at most two remaining neighbors of v, the removal of u breaks all odd cycles intersecting $\{u, v, w\}$. This corresponds to the fact that the triple $(X_B = \{v, w\}, X_C = \{u\}, X_R = V(G) \setminus \{u, v, w\})$ forms a tight *relaxed* odd cycle cut. We interpret the fact that the $k = 1$ case was developed naturally in an existing algorithm as encouraging evidence that refined research into search-space reduction steps can eventually lead to impact in practice.

References

1 Faisal N. Abu-Khzam, Michael R. Fellows, Michael A. Langston, and W. Henry Suters. Crown structures for vertex cover kernelization. *Theory Comput. Syst.*, 41(3):411–430, 2007. doi:10.1007/s00224-007-1328-0.

2 Benjamin Merlin Bumpus, Bart M. P. Jansen, and Jari J. H. de Kroon. Search-space reduction via essential vertices. In Shiri Chechik, Gonzalo Navarro, Eva Rotenberg, and Grzegorz Herman, editors, *Proceedings of the 30th Annual European Symposium on Algorithms, ESA 2022*, volume 244 of *LIPIcs*, pages 30:1–30:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.ESA.2022.30.

3 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.

4 Huib Donkers and Bart M.P. Jansen. Preprocessing to reduce the search space: Antler structures for feedback vertex set. *Journal of Computer and System Sciences*, 144, 2024. doi:10.1016/j.jcss.2024.103532.

5 Michael R. Fellows. Blow-ups, win/win's, and crown rules: Some new directions in FPT. In Hans L. Bodlaender, editor, *Proceedings of the 29th International Workshop on Graph-theoretic Concepts in Computer Science, WG 2003*, volume 2880 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2003. doi:10.1007/978-3-540-39890-5_1.

6 Timothy D. Goodrich, Eric Horton, and Blair D. Sullivan. An updated experimental evaluation of graph bipartization methods. *ACM J. Exp. Algorithmics*, 26:12:1–12:24, 2021. doi:10.1145/3467968.

7 Timothy D. Goodrich, Blair D. Sullivan, and Travis S. Humble. Optimizing adiabatic quantum program compilation using a graph-theoretic framework. *Quantum Information Processing*, 17(5), April 2018. doi:10.1007/s11128-018-1863-4.

8 Falk Hüffner. Algorithm engineering for optimal graph bipartization. *J. Graph Algorithms Appl.*, 13(2):77–98, 2009. URL: http://jgaa.info/accepted/2009/Hueffner2009.13.2.pdf, doi:10.7155/JGAA.00177.

9 Bart M. P. Jansen and Jari J. H. de Kroon. FPT algorithms to compute the elimination distance to bipartite graphs and more. In Lukasz Kowalik, Michal Pilipczuk, and Pawel Rzazewski, editors, *Graph-Theoretic Concepts in Computer Science - 47th International Workshop, WG 2021, Warsaw, Poland, June 23-25, 2021, Revised Selected Papers*, volume 12911 of *Lecture Notes in Computer Science*, pages 80–93. Springer, 2021. doi:10.1007/978-3-030-86838-3_6.

10 Bart M. P. Jansen, Jari J. H. de Kroon, and Michal Wlodarczyk. Single-exponential FPT algorithms for enumerating secluded {-free subgraphs and deleting to scattered graph classes. In Satoru Iwata and Naonori Kakimura, editors, *Proceedings of the 34th International Symposium on Algorithms and Computation, ISAAC 2023*, volume 283 of *LIPIcs*, pages 42:1–42:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.ISAAC.2023.42.

11 Bart M. P. Jansen, Jari J. H. de Kroon, and Michał Włodarczyk. Vertex Deletion Parameterized by Elimination Distance and Even Less, 2022. arXiv:2103.09715 [cs]. doi:10.48550/arXiv.2103.09715.

12 Bart M. P. Jansen, Yosuke Mizutani, Blair D. Sullivan, and Ruben F. A. Verhaegh. Preprocessing to reduce the search space for odd cycle transversal, 2024. arXiv:2409.00245, doi:10.48550/arXiv.2409.00245.

13 Bart M. P. Jansen and Ruben F. A. Verhaegh. Search-space reduction via essential vertices revisited: Vertex multicut and cograph deletion. In Hans L. Bodlaender, editor, *Proceedings of the 19th Scandinavian Symposium on Algorithm Theory, SWAT 2024*, LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. In press. arXiv:2404.09769, doi:10.4230/LIPICS.SWAT.2024.28.

14 Stefan Kratsch and Magnus Wahlström. Compression via matroids: A randomized polynomial kernel for odd cycle transversal. *ACM Trans. Algorithms*, 10(4):20:1–20:15, 2014. doi:10.1145/2635810.

15 Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. *J. ACM*, 67(3):16:1–16:50, 2020. doi:10.1145/3390887.

16 Daniel Lokshtanov, N. S. Narayanaswamy, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. Faster parameterized algorithms using linear programming. *ACM Trans. Algorithms*, 11(2):15:1–15:31, 2014. doi:10.1145/2566616.

17 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.

18 Marcin Pilipczuk and Michal Ziobro. Experimental evaluation of parameterized algorithms for graph separation problems: Half-integral relaxations and matroid-based kernelization. *CoRR*, abs/1811.07779, 2018. arXiv:1811.07779.

19 Bruce A. Reed, Kaleigh Smith, and Adrian Vetta. Finding odd cycle transversals. *Oper. Res. Lett.*, 32(4):299–301, 2004. doi:10.1016/j.orl.2003.10.009.

20 Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer, 2003.

21 Sebastian Wernicke. *On the algorithmic tractability of single nucleotide polymorphism (SNP) analysis and related problems.* diplom.de, 2014.