# ODD MULTIWAY CUT IN DIRECTED ACYCLIC GRAPHS

BY

SAHAND MOZAFFARI

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Adviser:

Assistant Professor Karthekeyan Chandrasekaran

**ABSTRACT**

We investigate the odd multiway node (edge) cut problem where the input is a graph with a specified collection of terminal nodes and the goal is to find a smallest subset of non-terminal nodes (edges) to delete so that the terminal nodes do not have an odd length path between them. In an earlier work, Lokshtanov and Ramanujan showed that both odd multiway node cut and odd multiway edge cut are fixed-parameter tractable (FPT) when parameterized by the size of the solution in *undirected graphs*. In this work, we focus on directed acyclic graphs (DAGs) and design a fixed-parameter algorithm. Our main contribution is a broadening of the shadow-removal framework to address *parity problems in DAGs*. We complement our FPT results with tight approximability as well as polyhedral results for 2 terminals in DAGs. Additionally, we show inapproximability results for odd multiway *edge* cut in undirected graphs even for 2 terminals.

## ACKNOWLEDGMENTS

The author would like to thank his advisor Professor Karthekeyan Chandrasekaran for many helpful conversations and his support throughout this work.

## TABLE OF CONTENTS

# CHAPTER 1: INTRODUCTION

In the classic $\{s,t\}$-cut problem, the goal is to delete the smallest number of edges so that the resulting graph has no path between $s$ and $t$. A natural generalization of this problem is the multiway cut problem, where the input is a graph with a specified set of terminal nodes and the goal is to delete the smallest number of non-terminal nodes/edges so that the terminals cannot reach each other in the resulting graph. In this work, we consider a parity variant of the multiway cut problem. A path[1] is an *odd-path* (*even-path*) if it has an odd (even) number of edges. In the OddMultiwayNodeCut (similarly, OddMultiwayEdgeCut), the input is a graph with a collection of terminal nodes and the goal is to delete the smallest number of non-terminal nodes (edges) so that the resulting graph has no odd-path between the terminals. This is a generalization of $\{s,t\}$-OddPathNodeBlocker (and similarly, $\{s,t\}$-OddPathEdgeBlocker), which is the problem of finding a minimum number of nodes (edges) that are disjoint from $s$ and $t$ that cover all $s-t$ odd-paths.

Covering and packing paths has been a topic of intensive investigation in graph theory as well as polyhedral theory. Menger's theorem gives a perfect duality relation for covering $s-t$ paths: the minimum number of nodes (edges) that cover all $s-t$ paths is equal to the maximum number of node-disjoint (edge-disjoint) $s-t$ paths. However, packing paths of restricted kinds has been observed to be a difficult problem in the literature. One special case is when the paths are required to be of odd-length for which many structural results exist [1, 2, 3]. In this work, we study the problem of covering $s-t$ odd-paths and more generally all odd-paths between a given collection of terminals.

Covering $s-t$ odd-paths in undirected graphs has been explored in the literature from the perspective of polyhedral theory—e.g., see Chapter 29 in Schrijver's book [3]. Given an undirected graph $G=(V,E)$ with distinct nodes $s,t \in V$ and non-negative edge lengths, we may find a shortest length $s-t$ odd-path in polynomial time. The shortest $s-t$ odd-path problem can be reduced to the shortest $s'-t$ even-path problem by adding a new source $s'$ and connecting it to the old source $s$. Edmonds gave a polynomial-time algorithm for the shortest length $s-t$ even-path problem by reducing it to the minimum-weight perfect matching problem [4, 5, 6]. However, as observed by Schrijver and Seymour [7], his approach of reducing to a matching problem does not extend to address other fundamental

[1] We emphasize that the term *paths* refers to simple paths and not walks. This distinction is particularly important in parity-constrained settings, because the existence of a walk with an odd number of edges between two nodes $s$ and $t$ does not imply the existence of an odd-path between $s$ and $t$. This is in contrast to the non-parity-constrained settings where the existence of a walk between $s$ and $t$ implies the existence of a path between $s$ and $t$.

problems about $s-t$ odd-paths. One such fundamental problem is the $\{s,t\}$-OddPathEdgeBlocker problem. Towards investigating $\{s,t\}$-OddPathEdgeBlocker, Schrijver and Seymour [7] considered the following polyhedron:

$$\mathcal{P}^{\text{odd-cover}} := \left\{ x \in \mathbb{R}_+^E : \sum_{e\in P} x_e \geq 1 \ \forall \ s-t \text{ odd-path } P \text{ in } G \right\} \ . \tag{1.1}$$

This leads to a natural integer programming formulation of $\{s,t\}$-OddPathEdgeBlocker: min $\left\{\sum_{e\in E} x_e : x \in \mathcal{P}^{\text{odd-cover}} \cap \mathbb{Z}^E\right\}$. By Edmonds' algorithm, we have an efficient separation oracle for $\mathcal{P}^{\text{odd-cover}}$ and hence there exists an efficient algorithm to optimize over $\mathcal{P}^{\text{odd-cover}}$ using the ellipsoid algorithm [8]. It was known that the extreme points of $\mathcal{P}^{\text{odd-cover}}$ are not integral. Schrijver and Seymour [7] proved a conjecture of Cook and Sebő that all extreme points of $\mathcal{P}^{\text{odd-cover}}$ are half-integral. Schrijver and Seymour's work also gave a min-max relation for the maximum fractional packing of $s-t$ odd-paths. However, their work does not provide algorithms to address $\{s,t\}$-OddPathEdgeBlocker. In fact, to the best of our knowledge, even the computational complexity of $\{s,t\}$-OddPathEdgeBlocker has not been addressed in the literature.

In this work, we undertake a comprehensive study of OddMultiwayNodeCut and OddMultiwayEdgeCut in directed acyclic graphs (DAGs). In addition to approximability, we focus on fixed-parameter tractability. Fixed-parameter algorithms have served as an alternative approach to address NP-hard problems [9]. A fixed-parameter algorithm for a problem decides all the problem's instances of size $n$ in time $f(k) \cdot n^{O(1)}$ for some computable function $f$, where $k$ is some integer parameter. Fixed-parameter algorithms for cut problems have provided novel insights into the connectivity structure of graphs [9]. The notion of *important separators* and the *shadow-removal technique* have served as the main ingredients in the design of fixed-parameter algorithms for numerous cut problems [10, 11, 12, 13, 14]. Our work also builds upon the shadow-removal technique to design fixed-parameter algorithms but differs from known applications substantially owing to the parity constraint. Parity-constrained cut problems have attracted much interest in the parameterized complexity community [15, 16, 17, 18] mainly due to their challenging nature: indeed, designing fixed-parameter algorithms for parity-constrained cut problems sparked the development of new and powerful techniques [15, 17, 18, 19, 20, 21, 22].

## 1.1 OUR CONTRIBUTIONS

The main focus of this work is the problem OddMultiwayNodeCut in directed acyclic graphs (DAGs). Before describing the reason for focusing on DAGs among directed graphs, we note that problems OddMultiwayNodeCut and OddMultiwayEdgeCut are equivalent in directed graphs by standard reductions (e.g., see Lemma A.1). The reason we focus on DAGs as opposed to all directed graphs is due to the following fact: it is $\mathsf{NP}$-complete to verify if a given directed graph has an $s \to t$ odd-path (e.g., see LaPaugh-Papadimitriou [6]). This fact already illustrates a stark contrast in the complexity between problem $\{s,t\}$-OddPathEdgeBlocker in undirected graphs and problem $(s \to t)$-OddPathEdgeBlocker in directed graphs: while in undirected graphs, verifying feasibility of a solution to $\{s,t\}$-OddPathEdgeBlocker can be done in polynomial-time, verifying feasibility of a solution to $(s \to t)$-OddPathEdgeBlocker in directed graphs is $\mathsf{NP}$-complete. However, there exists a polynomial time algorithm to verify if a given directed acyclic graph (DAG) has an $s \to t$ odd-path (e.g., see Lemma 3.1). For this reason, we restrict our focus to DAGs.

Our main contribution is the first fixed-parameter algorithm for problem OddMultiwayNodeCut in DAGs parameterized by solution size $k$. We complement the fixed-parameter algorithm by showing $\mathsf{NP}$-hardness and tight approximability results as well as polyhedral results for the two-terminal variant, namely problem$(s \to t)$-OddPathNodeBlocker, in DAGs.

In addition to the above results for DAGs, we also show $\mathsf{NP}$-hardness and an inapproximability result for $\{s,t\}$-OddPathEdgeBlocker in undirected graphs.

### 1.2 RELATED WORK

We are not aware of any work on this problem in directed graphs. We describe the known results in undirected graphs. A simple reduction[2] from Vertex Cover shows that $\{s,t\}$-OddPathNodeBlocker in undirected graphs is $\mathsf{NP}$-hard and does not admit a $(2-\varepsilon)$-approximation for $\varepsilon > 0$ assuming the Unique Games Conjecture [23]. These hardness results also hold for OddMultiwayNodeCut. The most relevant results to this work are that of Lokshtanov and Ramanujan [16, 24]. They studied an extension of OddMultiwayNodeCut and OddMultiwayEdgeCut that they termed as ParityMultiwayNodeCut (and ParityMultiwayEdgeCut)—the input is an undirected graph and two subsets of terminals $T_e$ and $T_o$ and the goal is to find the smallest number of non-terminal nodes (edges) so that every node $u \in T_o$ has no odd-path to any node in $T_e \cup T_o$ and every node $u \in T_e$ has no even-path to any node in $T_e \cup T_o$. Lokshtanov and Ramanujan designed

[2]Given an instance $G$ of Vertex Cover, introduce two new nodes $s$ and $t$ that are adjacent to all nodes in $G$ to obtain a graph $H$. Then a set $S \subseteq V(G)$ is a vertex cover in $G$ if and only if $S$ is a feasible solution to $\{s,t\}$-OddPathNodeBlocker in $H$.

a fixed-parameter algorithm for ParityMultiwayNodeCut by reducing the problem to OddMultiwayNodeCut and designing a fixed-parameter algorithm for OddMultiwayNodeCut. However, their algorithmic techniques work only for undirected graphs and do not extend to the case of OddMultiwayNodeCut in DAGs.

Lokshtanov and Ramanujan also showed that OddMultiwayEdgeCut is NP-hard in undirected graphs for three terminals. However, their reduction is not an approximation-preserving reduction. Hence the approximability of OddMultiwayEdgeCut in undirected graphs merits careful investigation. In particular, the complexity of OddMultiwayEdgeCut in undirected graphs even for the case of two terminals has not been addressed in the literature in spite of existing polyhedral work by Schrijver and Seymour [7] for this problem.

The subset odd cycle transversal problem (SubsetOCT) generalizes the OddMultiwayNodeCut problem in undirected graphs. Here, the input is an undirected graph $G$ with a subset of vertices $T$ and the goal is to determine a smallest subset of vertices that intersects every odd cycle containing a vertex from $T$. Fixed-parameter algorithms for SubsetOCT are also known in the literature [25].

## 1.3 RESULTS

**Directed acyclic graphs.** Recall that OddMultiwayNodeCut and OddMultiwayEdgeCut are equivalent in DAGs by standard reductions. Hence, all of the following results for DAGs hold for both problems.

The following is our main result.

**Theorem 1.1.** OddMultiwayNodeCut and OddMultiwayEdgeCut in DAGs can be solved in $2^{O(k^2)} \cdot n^{O(1)}$ time, where $k$ is the size of the optimal solution and $n$ is the number of nodes in the input graph.

In order to prove Theorem 1.1, we exploit the acyclic property of the input directed graph to reduce the instance to an instance of Odd Cycle Transversal. In Odd Cycle Transveral, the goal is to remove the smallest number of nodes to make an undirected graph bipartite. Odd Cycle Transversal is fixed-parameter tractable when parameterized by the number of removed nodes. Our algorithm builds upon the shadow-removal framework to accomplish the reduction to the Odd Cycle Transversal problem. We view our technique as an illustration of the broad applicability of the shadow-removal framework.

We complement our fixed-parameter algorithm in Theorem 1.1 with tight approximability results for the special case of 2 terminals. We refer the reader to Table 1.1 for a summary of the complexity and approximability results. Unlike the case of undirected graphs where there is still a gap in the approximability of both problems $\{s,t\}$-OddPathEdgeBlocker and $\{s,t\}$-OddPathNodeBlocker, we present tight approximability results for both problems $(s \to t)$-OddPathEdgeBlocker and $(s \to t)$-OddPathNodeBlocker in DAGs:

**Theorem 1.2.** The following inapproximability and approximability results hold:

(i) $(s \to t)$-OddPathNodeBlocker in DAGs is NP-hard, and has no efficient $(2-\varepsilon)$-approximation for any $\varepsilon > 0$ assuming the Unique Games Conjecture.

(ii) There exists an efficient 2-approximation algorithm for $(s \to t)$-OddPathNodeBlocker in DAGs.

We emphasize that our efficient 2-approximation algorithm for $(s \to t)$-OddPathEdgeBlocker mentioned in Theorem 1.2 is a combinatorial algorithm and not LP-based. We note that Schrijver and Seymour's result [7] that all extreme points of $\mathcal{P}^{\text{odd-cover}}$ are half-integral holds only in undirected graphs and fails in DAGs—see Theorem 1.3 below. Consequently, we are unable to design a 2-approximation algorithm using the extreme point structure of the natural LP-relaxation of the path-blocking integer programming. Instead, our approximation algorithm is combinatorial in nature. The correctness argument of our algorithm also shows that the integrality gap of the LP-relaxation of the path-blocking integer program is at most 2 in DAGs.

**Theorem 1.3.** The odd path cover polyhedron given by

$$\mathcal{P}^{\text{odd-cover-dir}} := \left\{ x \in \mathbb{R}_+^E : \sum_{e \in P} x_e \geq 1 \ \forall\ s \to t \text{ odd-path } P \text{ in } D \right\} \tag{1.2}$$

for directed acyclic graphs $D = (V, E)$ is not necessarily half-integral.

**Undirected graphs.** We next turn our attention to undirected graphs. As mentioned in Section 1.2, the problem $\{s,t\}$-OddPathNodeBlocker is NP-hard and does not admit a $(2-\varepsilon)$-approximation assuming the Unique Games Conjecture. We are unaware of a constant-factor approximation for $\{s,t\}$-OddPathNodeBlocker. For $\{s,t\}$-OddPathEdgeBlocker, the results of Schrijver and Seymour [7] show that the LP-relaxation of a natural integer linear programming formulation of $\{s,t\}$-OddPathEdgeBlocker is half-integral and thus leads to an efficient 2-approximation algorithm. However, the complexity of $\{s,t\}$-OddPathEdgeBlocker has not been addressed in the literature. We address this gap in complexity by showing the following NP-hardness and inapproximability results.

| Problem | Undirected graphs | DAGs |
|---|---|---|
| $\{s,t\}$-OddPathNodeBlocker | $(2-\varepsilon)$-inapprox | 2-approx (Thm. 1.2)<br>$(2-\varepsilon)$-inapprox (Thm 1.2) |
| $\{s,t\}$-OddPathEdgeBlocker | LP is half-integral [7]<br><br>2-approx [7]<br>$(\frac{6}{5}-\varepsilon)$-inapprox (Thm. 1.4) | LP is NOT half-integral (Thm. 1.3)<br>2-approx (Thm. 1.2)<br>$(2-\varepsilon)$-inapprox (Thm 1.2) |
| OddMultiwayEdgeCut | NP-hard for 3 terminals [16]<br>$(\frac{6}{5}-\varepsilon)$-inapprox for two terminals (Thm 1.4) | $2^{O(k^2)} \cdot \text{poly}(n)$ (Thm. 1.1) |
| OddMultiwayNodeCut | | $2^{O(k^2)} \cdot \text{poly}(n)$ (Thm. 1.1) |

Table 1.1: Complexity and approximability. Text in gray refers to known results while text in black refers to the results from this work.

**Theorem 1.4.** $\{s,t\}$-OddPathEdgeBlocker is NP-hard and has no efficient $(6/5-\varepsilon)$-approximation assuming the Unique Games Conjecture.

**Organization.** We summarize the preliminaries in Section 1.4. We devise the fixed-parameter algorithms for DAGs (Theorem 1.1) in Section 2. We complement with approximability results for DAGs (Theorems 1.2 and 1.3) in Section 3. Next, we focus on undirected graphs and present the inapproximability result (Theorem 1.4) in Section 4. We conclude by discussing a few open problems in Section 5.

### 1.4 PRELIMINARIES

Let $G$ be a (directed) graph with vertex set $V(G)$ and edge set $E(G)$. For single vertices $v \in V(G)$, we will frequently use $v$ instead of $\{v\}$. For a subset $W \subseteq V(G)$, *a $W$-path in $G$* is a path with both of its end nodes in $W$.

For a directed acyclic graph $G$ and node sets $T$, $V^\infty \subseteq V(G)$ where $T \subseteq V^\infty$, an *odd multiway cut* in $G$ is a set $M \subseteq V(G) \setminus V^\infty$ of nodes that intersects every odd $T$-path in $G$. We refer to elements of $T$ as *terminals*, elements of $V(G) \setminus T$ as *non-terminals*, and elements of $V^\infty$ as *protected nodes*.

We restate the problem of OddMultiwayNodeCut in DAGs in Figure 1.1.

For subsets $X$ and $Y$ of $V(G)$ we say that $M \subseteq V(G) \setminus V^\infty$ is an $X \to Y$ *separator* in $G$ when $G \setminus M$ has no path from $X$ to $Y$. The set of nodes that can be reached from a node set $X$ in $G$ is denoted by $\mathcal{R}_G(X)$. We note that $\mathcal{R}_G(X)$ always includes $X$.

| | |
|---|---|
| OddMultiwayNodeCut in DAGs | Parameter: $k$ |
| *Input:* A DAG $G$ with a set $V^\infty \subseteq V(G)$ of protected nodes, set $T \subseteq V^\infty$ of terminal nodes, and an integer $k \in \mathbb{Z}_+$. | |
| *Task:* Verify if there is an odd multiway node cut in $G$ of size at most $k$. | |

Figure 1.1: Statement of the OddMultiwayNodeCut problem in DAGs

We define the *forward shadow* of a node set $M$ to be $f_G(M) := V(G \setminus M) \setminus \mathcal{R}_{G\setminus M}(T)$, i.e., the set of nodes $v$ such that there is no $T \to v$ path in $G$ disjoint from $M$. Similarly, the *reverse shadow* of $M$, denoted $r_G(M)$, is the set of nodes $v$ from which there is no path to $T$ in $G \setminus M$. Equivalently, the reverse shadow is $f_{G^{\text{rev}}}(M)$, where $G^{\text{rev}}$ is the graph obtained from $G$ by reversing all the edge orientations. We refer to the union of the forward and the reverse shadow of $M$ in $G$, as *shadow* of $M$ in $G$ and denote it by $s_G(M)$. A set $M \subseteq V(G)$ is *thin*, if every node $v \in M$ is not in $r_G(M \setminus \{v\})$. See Fig. 1.2 for an example illustrating these concepts.

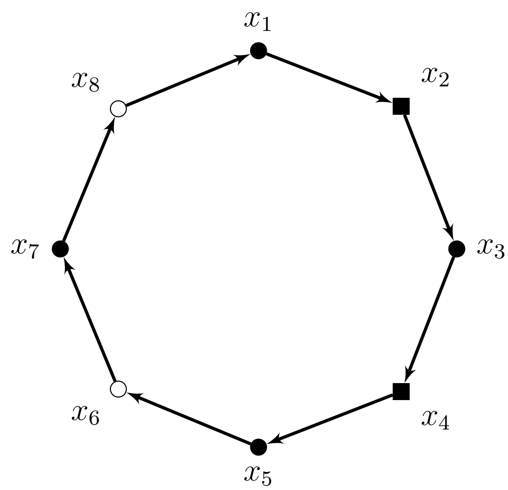



Figure 1.2: In this graph $G$, suppose $T = \{x_2, x_4\}$ and $M = \{x_6, x_8\}$. Then, we have the forward shadow $f_G(M) = \{x_1, x_7\}$, the reverse shadow $r_G(M) = \{x_5, x_7\}$ and the shadow $s_G(M) = \{x_1, x_5, x_7\}$.

We need the notion of important separators [12]. An $X \to Y$-separator $M'$ is said to *dominate* another $X \to Y$-separator $M$, if $|M'| \leq |M|$ and $\mathcal{R}_{G\setminus M}(X) \subsetneq \mathcal{R}_{G\setminus M'}(X)$. A minimal $X \to Y$-separator that is not dominated by any other separator is called an *important $X \to Y$-separator*.

For a directed graph $G$, its underlying undirected graph $\langle G \rangle$ is the undirected graph obtained from $G$ by dropping the edge orientations. In an undirected graph $H$ with protected nodes $V^\infty$, an *odd cycle transversal* is a set $U \subseteq V(H) \setminus V^\infty$ of nodes such that $H \setminus U$ is bipartite. The problem of finding a minimum odd cycle transversal in a given instance $(H, V^\infty)$ is the OddCycleTransversal problem. This problem is NP-hard, but admits fixed-parameter algorithms when parameterized by the size $k$ of an optimal solution. The asymptotically fastest fixed-parameter algorithm for OddCycleTransversal in terms of $k$ is due to Lokshtanov et al. [26]; it runs in time $2.32^k \cdot n^{O(1)}$, and is based on linear programming techniques. While their algorithm does not allow for protected nodes, the problem OddCycleTransversal with protected nodes can be reduced to OddCycleTransversal without protected nodes by iteratively replacing each protected node with $k+1$ nodes and connecting them to the same set of neighbors as the original node. We thus have:

**Proposition 1.1.** There is an algorithm that, given an $n$-node graph $H$, a set $V^\infty \subseteq V(H)$ of protected nodes and an integer $k$, in time $2.32^k \cdot n^{O(1)}$ decides if $H$ has an odd cycle transversal of size at most $k$ disjoint from $V^\infty$, and if so, returns one.

We will use OddCycleTransversal$(H, V^\infty, k)$ to denote the procedure that implements this fixed-parameter algorithm for the input graph $H$ with protected nodes in $V^\infty$ and parameter $k$.

# CHAPTER 2: FIXED-PARAMETER TRACTABILITY OF ODDMULTIWAYNODECUT IN DAGS

To solve OddMultiwayNodeCut in DAGs, we will use the shadow-removal technique introduced by Chitnis, Hajiaghayi and Marx [11]. We will reduce the problem to the OddCycleTransversal problem in undirected graphs, which is a fixed-parameter tractable problem when parameterized by the solution size. We introduce the following property:

**Definition 2.1** (Isolated Shadows Property)**.** Given an instance $(G, V^\infty, T, k)$, a set $M \subseteq V$ has the *isolated shadows property*, if every node $v \in s_G(M)$ has total degree at most one in $G \setminus M$.

We begin by arguing about "easy" instances, where we define an instance $(G, V^\infty, T, k)$ as easy if it has a solution $M$ (of size at most $k$) with the isolated shadows property provided that it has some solution (of size at most $k$) at all.

## 2.1 EASY INSTANCES

**Theorem 2.1.** There is an algorithm that, given any easy instance $(G, V^\infty, T, k)$ of OddMultiwayNodeCut where $G$ is a DAG, finds a solution of size at most $k$ in time $2.32^k \cdot n^{O(1)}$, where $n$ is the number of nodes in the input graph $G$.

*Proof.* Let $(G, V^\infty, T, k)$ be an instance of OddMultiwayNodeCut. Let $\langle G \rangle$ denote the undirected graph obtained from $G$ by dropping the orientations of the edges in $G$. We show the following equivalence: a set $M \subseteq V \setminus V^\infty$ with the isolated shadows property is a solution if and only if $\langle G \rangle \setminus M$ is bipartite with a bipartition $(A, B)$ such that $T \subseteq A$.

Suppose $\langle G \rangle \setminus M$ is bipartite with a bipartition $(A, B)$ such that $T \subseteq A$. In a bipartite graph, every two end-nodes of any odd path are necessarily in different parts. Hence, there is no odd $T$-path in $\langle G \rangle \setminus M$. Thus, there is no odd $T$-path in $G \setminus M$. Hence, the set $M$ is a solution for the OddMultiwayNodeCut instance $(G, V^\infty, T, k)$.

Suppose the solution $M$ has the isolated shadows property. Define the set $U := V(G \setminus M) \setminus s_G(M)$. Define

$$A := \{x \in U : \text{there is an even } T \to x\text{-path in } G \setminus M\} \text{ and} \tag{2.1}$$

$$B := \{x \in U : \text{there is an odd } T \to x\text{-path in } G \setminus M\}. \tag{2.2}$$

It follows from the definition of the shadow that every node in $U$ has a path $P_1$ from $T$ in $G \setminus M$. Therefore, every node of $U$ is in $A \cup B$. Also by definition, every node $v$ in $U$ has a path $P_2$ to $T$ in $G \setminus M$. The parity of every $T \to v$-path has to be the same as the parity of $P_2$, because the concatenation of a $T \to v$-path and a $v \to T$-path in $G \setminus M$ is a $T$-path in $G \setminus M$ and therefore must be even. We note that such a concatenation cannot create a cycle since $G$ is acyclic. In a general directed graph, this concatenation would result in a $T$-walk, but in a DAG the result will be a $T$-path. Thus, no node of $U$ is in both $A$ and $B$. Hence, we have that $(A, B)$ is a partition of $U$.

We observe that there cannot exist an edge from a node $v$ in $A$ to a node $u$ in $A$, as otherwise the concatenation of the even $T \to v$-path $Q_1$ with the edge $v \to u$ is an odd $T \to u$-path in $G \setminus M$ which means $u \in B$. This contradicts our conclusion about $A$ and $B$ being disjoint. By a similar argument, there is no edge between any pair of nodes in $B$. Thus, the sets $A$ and $B$ are independent sets in the the subgraph of $G$ induced by those two sets respectively. Hence $\langle G \rangle[A \cup B]$ is a bipartite graph. Furthermore, $(A, B)$ is a bipartition of $\langle G \rangle[A \cup B]$ with every node of $T$ in $A$. By assumption, the degree of every node $x \in s_G(M)$ is at most one. Therefore, $x$ has neighbors in at most one of $A$ and $B$. Thus, we can extend the bipartition $(A, B)$ of $\langle G \rangle[A \cup B]$ to a bipartition $(A', B')$ of $\langle G \rangle \setminus M$ as follows: denote $H := \langle G \rangle[A \cup B]$; repeatedly pick a node $x \in s_G(M) \setminus V(H)$ with a neighbor in $H$, include $x$ in a part ($A$ or $B$) in which $x$ has no neighbor and update $A$, $B$ and $H$.

Hence, if the given instance has a solution $M$ of size at most $k$ such that every node $v \in s_G(M)$ has total degree at most one, then such a solution can be found by the fixed-parameter algorithm for OddCycleTransversal. To ensure that the terminal nodes will be in the same part, we introduce a new protected node into the graph and connect it to every terminal node. This approach is described in Algorithm 2.1.

**Algorithm 2.1** `SolveEasyInstance`

1: **Input:** A DAG $G$ with a set $V^\infty \subseteq V(G)$ of protected nodes, a set $T \subseteq V^\infty$ of terminals, and an integer $k \in \mathbb{Z}_+$.
2: **Output:** A minimum odd multiway cut for $(G, V^\infty, T, k)$.
3: $G_1 \leftarrow$ the underlying undirected graph of $G$, i.e., $\langle G \rangle$.
4: Let $G_2$ be the graph obtained from $G_1$ by introducing a new node $x$ and connecting it to every node in $T$.
5: $N \leftarrow \textsc{OddCycleTransversal}(G_2, V^\infty \cup \{x\}, k)$
6: **return** $N$

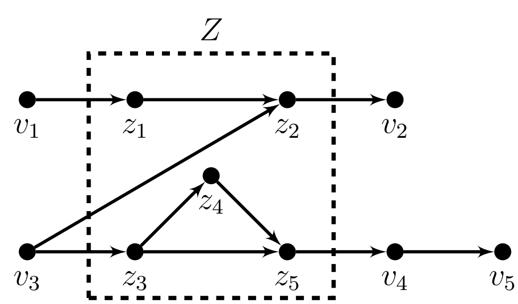
All steps in Algorithm 2.1 can be implemented to run in polynomial time except Step 5. By Proposition 1.1, Step 5 can run in time $2.32^k \cdot n^{O(1)}$. QED.

We will use the name `SolveEasyInstance` to refer to the algorithm of Theorem 2.1. Theorem 2.1 suggests that the existence of a solution $M$ of size at most $k$, such that every node $v \in s_G(M)$ has total degree at most one, is a useful property in an instance of Odd-MultiwayNodeCut. However, it is not necessarily the case that some solution of size at most $k$ always has this property. Our aim now is to reduce the given arbitrary instance $(G, V^\infty, T, k)$ to another instance that has such a solution or determine that no solution of size at most $k$ exists. For this purpose, we define the operation parity-preserving torso on DAGs, as follows.

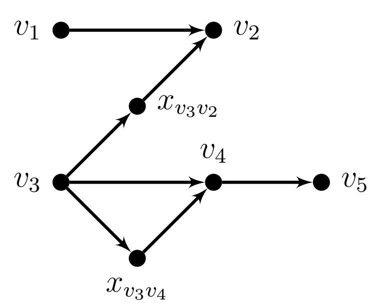
## 2.2 PARITY-PRESERVING TORSO

The parity-preserving torso operation was introduced by Lokshtanov and Ramanujan [16] for undirected graphs. We extend it in a natural fashion for DAGs.

**Definition 2.2** (Parity-preserving torso.)**.** Let $G$ be a DAG and $Z \subseteq V(G)$. Let $G'$ be the DAG obtained from $G \setminus Z$ by adding an edge from node $u$ to $v$, for every pair of nodes $u, v \in V(G) \setminus Z$ such that there is an odd-path from $u$ to $v$ in $G$ all of whose internal nodes are in $Z$. We obtain `ParityTorso`$(G, V^\infty, Z)$ from $(G', V'^\infty)$ by including a new node $x_{uv}$ and edges $u \to x_{uv}$ and $x_{uv} \to v$ for every pair of nodes $u, v \in V(G) \setminus Z$ such that there is an even path from $u$ to $v$ in $G$ all of whose internal nodes are in $Z$. The set $V'^\infty$ is defined to be the union of $V^\infty \setminus Z$ and all the new nodes $x_{uv}$ (see Fig. 2.1).



(a) The original graph $G$.



(b) `ParityTorso`$(G, Z)$.

Figure 2.1: An illustration of the parity-preserving torso operation.

We emphasize that the acyclic nature of the input directed graph allows us to implement the parity-preserving torso operation in polynomial time (e.g., using Lemma 3.1). Moreover, applying parity-preserving torso on a DAG results in a DAG as well. In what follows, we state the properties of the `ParityTorso` operation that are exploited by our algorithm. The parity-preserving torso operation, has the property that it maintains $u \to v$-paths along with their parities between any pair of nodes $u, v \in V(G) \setminus Z$. More precisely:

**Lemma 2.1.** Let $G$ be a DAG and $Z, V^\infty \subseteq V(G)$. Also define the pair $(G', V'^\infty) :=$ $\texttt{ParityTorso}(G, V^\infty, Z)$. Let $u, v$ be nodes in $V(G) \setminus Z$. There is a $u \to v$-path $P$ in $G$ if and only if there is a $u \to v$-path $Q$ of the same parity in $G'$. Moreover, the path $Q$ can be chosen so that the nodes of $P$ in $G \setminus Z$ are the same as the nodes of $Q$ in $G \setminus Z$, i.e., $V(P) \cap (V(G) \setminus Z) = V(Q) \cap (V(G) \setminus Z)$.

*Proof.* We prove the forward direction by induction on the length of $P$. For the base case of induction, consider paths of length zero in $G$ that are disjoint from $Z$. Such a path is not affected by the $\texttt{ParityTorso}$ operation. Suppose that the claim holds for all paths of length less than $\ell$, for some $\ell > 0$. Let $P$ be a $u \to v$-path of length $\ell$ in $G$, where $u, v \in V(G) \setminus Z$. If all internal nodes of $P$ are in $Z$, then by definition of $\texttt{ParityTorso}$, a path $Q$ of the same parity exists in $G'$ and $V(P) \cap (V(G) \setminus Z) = V(Q) \cap (V(G) \setminus Z)$. Otherwise, let $w \in V(G) \setminus Z$ be an internal node of $P$. Let $P_1$ and $P_2$ be the subpaths of $P$ from $u$ to $w$ and from $w$ to $v$. By induction hypothesis, there is a $u \to w$-path $Q_1$ in $G'$ of the same parity as $P_1$ where $V(P_1) \cap (V(G) \setminus Z) = V(Q_1) \cap (V(G) \setminus Z)$, and similarly, a $w \to v$-path $Q_2$ is found of the same parity as $P_2$ where $V(P_2) \cap (V(G) \setminus Z) = V(Q_2) \cap (V(G) \setminus Z)$. Since $G'$ is a DAG, the path $Q$ obtained by concatenating $Q_1$ and $Q_2$ has the same parity as $P$ and $V(P) \cap (V(G) \setminus Z) = V(Q) \cap (V(G) \setminus Z)$.

Conversely, suppose $Q$ is a path $u = x_0, x_1, x_2, \ldots, x_r = v$ in $G'$ from $u$ to $v$ where $u, v \in V(G)$. Then for every node $x_i$ of $Q$ in $V(G') \setminus V(G)$, replace the subpath $x_{i-1}, x_i, x_{i+1}$ with the even path in $G$ that connects $x_{i-1}$ to $x_{i+1}$. Also, for every pair $i$ where $x_i, x_{i+1} \in V(G)$ but $(x_i, x_{i+1})$ is not an edge in $G$, replace the subpath $x_i, x_{i+1}$ of $Q$ with the odd path that connects $x_i$ to $x_{i+1}$ in $G$. By construction, the resulting sequence is a path $P$ in $G$ and has the same parity as $Q$ and $V(P) \cap (V(G) \setminus Z) = V(Q) \cap (V(G) \setminus Z)$. QED.

**Corollary 2.1.** Let $\mathcal{I} = (G, V^\infty, T, k)$ be an input of OddMultiwayNodeCut and let $Z \subseteq V(G) \setminus T$. Let $(G', V'^\infty) := \texttt{ParityTorso}(G, V^\infty, Z)$ and denote the instance $(G', V'^\infty, T, k)$ by $\mathcal{I}'$. The instance $\mathcal{I}$ admits a solution $S$ of size at most $k$ that is disjoint from $Z$ if and only if the instance $\mathcal{I}'$ admits a solution of size at most $k$.

*Proof.* Let $M'$ be a solution to the instance $\mathcal{I}'$ of size at most $k$. As $V^\infty \setminus Z \subseteq V'^\infty$ and $M' \cap V'^\infty = \emptyset$ and $M' \cap Z = \emptyset$, we have that $M' \cap V^\infty = \emptyset$. By definition of the $\texttt{ParityTorso}$ operation, $V(G') \setminus V(G)$ is contained in $V'^\infty$ and therefore is disjoint from $M'$. Thus, $M' \subseteq V(G)$. Suppose $P$ is an odd $T$-path in $G$ and is disjoint from $M'$. By Lemma 2.1, there is an odd $T$-path in $G'$ that is also disjoint from $M'$, contradicting our assumption about $M'$.

Conversely, suppose $M$ is a solution for the instance $\mathcal{I}$ of size at most $k$ that is disjoint from $Z$. Suppose $P'$ is an odd $T$-path in $G'$ and is disjoint from $M$. By Lemma 2.1, there is an odd $T$-path in $G$ that is also disjoint from $M$, contradicting our assumption about $M$. QED.

Corollary 2.1 reveals that if there exists a solution $M$ in $G$ that is disjoint from $Z$ and $V^\infty$, then it also exists in the DAG obtained from $\texttt{ParityTorso}(G, V^\infty, Z)$ and hence it is sufficient to search for it in $\texttt{ParityTorso}(G, V^\infty, Z)$. Therefore, we are interested in finding a set $Z$ of nodes that is disjoint from some solution of size at most $k$, and moreover, the instance $(\texttt{ParityTorso}(G, V^\infty, Z), T, k)$ is an easy instance of the problem, i.e., satisfies the property mentioned in Theorem 2.1. The following lemma shows that it is sufficient to find a set $Z$ that contains the shadow of a solution.

**Lemma 2.2.** Let $G$ be a DAG and $M, Z, V^\infty \subseteq V(G)$. Suppose $M$ intersects every odd $T$-path in $G$ and $s_G(M) \subseteq Z \subseteq V(G) \setminus M$. Let $(G', V'^\infty) := \texttt{ParityTorso}(G, V^\infty, Z)$. Then every node in $s_{G'}(M)$ has total degree at most one in $G' \setminus M$.

*Proof.* We claim that $s_{G'}(M)$ is contained in $V(G') \setminus V(G)$. Suppose not. Then there is a node $v \in V(G) \setminus Z$ that is in $s_{G'}(M)$. Suppose $v \in r_{G'}(M)$. Thus, there is no path from $v$ to $T$ in $G'$ that is disjoint from $M$. By Lemma 2.1, every path in $G$ from $v$ to $T$ intersects $M$. Therefore, $v$ is in the shadow of $M$ in $G$ and is hence contained in $Z$. This is a contradiction. A similar contradiction arises if $v \in f_{G'}(M)$. Therefore, $s_{G'}(M)$ is disjoint from $V(G)$.

Let $x \in s_{G'}(M)$. We observe that by definition of the $\texttt{ParityTorso}$ operation, every node $x \in V(G') \setminus V(G)$ has in-degree and out-degree one. Let $u$ and $v$ be the in-neighbor and out-neighbor of $x$ in $G'$. Suppose $x$ has total degree two in $G' \setminus M$. This implies that $u, v \notin M$. Since $u$ is not in the shadow of $M$ in $G'$, there is a $T \to u$-path disjoint from $M$ in $G'$. Appending the $u \to x$ edge to that path, gives a $T \to x$-path in $G'$ disjoint from $M$. Thus, $x \notin f_{G'}(M)$. Similarly, $x \notin r_{G'}(M)$, because $v \notin M$. Thus, $x \notin s_{G'}(M)$, a contradiction. QED.

### 2.3 DIFFICULT INSTANCES

Corollary 2.1 and Lemma 2.2 show that if we find a set $Z$ such that for some solution $M$, the set $Z$ is disjoint from $M$ and contains the shadow of $M$ in $G$, then considering $\texttt{ParityTorso}(G, V^\infty, Z)$ will give a new instance that satisfies the conditions of Theorem 2.1. Our goal now is to obtain such a set $Z$. We will show the following lemma. We emphasize that the lemma holds for arbitrary digraphs.

**Lemma 2.3.** There is an algorithm $\texttt{ShadowContainer}$ that, given an instance $(G, V^\infty, T, k)$ of OddMultiwayNodeCut, in time $2^{O(k^2)} \operatorname{poly}(|V(G)|)$ returns a family $\mathcal{Z}$ of $2^{O(k^2)} \log |V(G)|$

subsets of $V(G)$, with the property that if the instance admits a solution of size at most $k$, then for some solution $M$ of size at most $k$ and there exists a set $Z \in \mathcal{Z}$ that is disjoint from $M$ and contains $s_G(M)$.

We defer the proof of Lemma 2.3 to first see its implications. We now show how the procedure `ShadowContainer` can be used to obtain a fixed-parameter algorithm for the OddMultiwayNodeCut problem in DAGs and thus prove Theorem 1.1.

**Algorithm 2.2** Minimum odd node multiway cut in DAGs

1: **Input:** A DAG $G$ with terminal set $T$, a set $V^\infty \supseteq T$ of protected nodes, and $k \in \mathbb{Z}_+$.
2: **Output:** An odd node multiway cut for $(G, T)$ of size at most $k$ and disjoint from $V^\infty$, or "no solution of size at most $k$" if such does not exist.
3: $\mathcal{Z} \leftarrow$ `ShadowContainer`$(G, T, V^\infty, k)$
4: **for** $Z \in \mathcal{Z}$ **do**
5:     $(G_1, V_1^\infty) \leftarrow$ `ParityTorso`$(G, V^\infty, Z)$
6:     $N \leftarrow \text{SolveEasyInstance}(G_1, V_1^\infty, T, k)$
7:     **if** $N$ is a solution in $G$ **then**
8:         **return** $N$
9: **return** "no solution of size at most $k$"

**Theorem 2.2.** There exists an algorithm that, given an instance $(G, V^\infty, T, k)$ of OddMultiwayNodeCut where $G$ is a DAG, in $2^{O(k^2)}\,\text{poly}(|V(G)|)$ time either finds a solution of size at most $k$ or determines that no such solution exists.

*Proof.* We use Algorithm 2.2. Let $(G, V^\infty, T, k)$ be an instance of OddMultiwayNodeCut, where $G$ is a DAG. Suppose there exists a solution of size at most $k$. By Lemma 2.3, the procedure `ShadowContainer`$(G, T, V^\infty, k)$ in Line 3 returns a family $\mathcal{Z}$ of subsets of $V(G)$ with $|\mathcal{Z}| = 2^{O(k^2)} \log |V(G)|$ containing a set $Z$ such that there is a solution $M$ of size at most $k$ that is disjoint from $Z$ and $Z$ contains $s_G(M)$. Let $(G_1, V_1^\infty)$ be the result of applying `ParityTorso` operation to the set $Z$ in $G$ (i.e., the result of Step 2 in Algorithm 2.2). By Lemma 2.2, every node in $s_{G_1}(M)$ has total degree at most one in $G_1 \setminus M$. Therefore, by Theorem 2.1, the set $N$ returned in Line 6 is a solution to the instance $(G_1, V_1^\infty, T, k)$. By Corollary 2.1, the set $N$ is also a solution to the original instance of the problem.

If there is no solution of size at most $k$, the algorithm will not find any. Therefore, the algorithm is correct. The runtime of the algorithm is dominated by Line 2 which can be implemented to run in $2^{O(k^2)}\text{poly}(|V(G)|)$ time by Lemma 2.3. QED.

To complete this proof, it remains to prove Lemma 2.3. In order to do so, we will use the following result.

**Theorem 2.3** (Chitnis, Hajiaghayi and Marx [27, Thm. 3.18])**.** There is an algorithm that, given a digraph $G$, a set of protected nodes $V^\infty \subseteq V(G)$, terminal nodes $T \subseteq V^\infty$ and an integer $k$, in time $2^{O(k^2)} \operatorname{poly}(|V(G)|)$ returns a family $\mathcal{Z}$ of subsets of $V(G) \setminus V^\infty$ with $|\mathcal{Z}| = 2^{O(k^2)} \log |V(G)|$ such that for every $S, Y \subseteq V(G)$ satisfying (i) and (ii) below, there is some $Z \in \mathcal{Z}$ for which $Y \subseteq Z \subseteq V(G) \setminus S$:

(i) $S$ is a thin set with $|S| \leq k$ and

(ii) for every $v \in Y$, there exists an important $v \to T$-separator contained in $S$.

To invoke Theorem 2.3, we need to guarantee that there exists a solution $S$ of size at most $k$ such that $S$ is thin and its reverse shadow $Y$ in $G$ has the property that for every $v \in Y$ there is an important $v \to T$-separator contained in $S$. Towards obtaining such a solution, we prove the following.

**Lemma 2.4.** Let $(G, V^\infty, T, k)$ be an instance of OddMultiwayNodeCut, where $G$ is a DAG. If there is a solution $M$ that does not contain an important $v \to T$-separator for some $v \in r_G(M)$, then there is another solution $M'$ of size at most $|M|$ such that $r_G(M) \cup f_G(M) \cup M \subseteq r_G(M') \cup f_G(M') \cup M'$, and $r_G(M) \subsetneq r_G(M')$.

*Proof.* Let $M_0$ be the set of nodes $u \in M$ for which there is a $v \to u$-path in $G$ that is internally disjoint from $M$. Since $v \in r_G(M)$, every $v \to T$-path intersects $M$. For a $v \to T$-path $P$, the first node $u \in P \cap M$ along $P$ is in $M_0$. Hence, every $v \to T$-path intersects $M_0$. Therefore, the set $M_0$ is a $v \to T$-separator in $G$. Therefore, it contains a minimal separator $M_1$. Since we assumed that there is no important $v \to T$-separator contained in $M$, the set $M_1$ is not an important $v \to T$-separator. Suppose $M_1$ is dominated by another $v \to T$-separator and let $M_2$ be an important $v \to T$-separator that dominates $M_1$. Define $M'$ as $(M \setminus M_1) \cup M_2$. We recall that a separator is by definition, disjoint from the protected node set. Therefore, $M' \cap V^\infty = \emptyset$. We will show that $M'$ contradicts the choice of $M$. We need the following claims.

**Claim 2.1.** $M \setminus M' \subseteq r_G(M')$.

*Proof of Claim 2.1.* We observe that $M \setminus M' = M_1 \setminus M_2$. Let $u$ be an arbitrary node in $M_1 \setminus M_2$. Since $u \in M_1$ and $M_1$ is a minimal $v \to T$-separator, there is a $v \to u$-path $P_1$ that is internally disjoint from $M_1$. Since $M_2$ dominates $M_1$, therefore, $R_{G \setminus M_1}(v) \subseteq R_{G \setminus M_2}(v)$. Thus, $V(P_1) \subseteq R_{G \setminus M_2}(v)$. Hence, $P_1$ is disjoint from $M_2$. Suppose $P_2$ is an arbitrary $u \to T$-path in $G$. Concatenation of $P_1$ and $P_2$ is a $v \to T$-path in $G$ and therefore, has to intersect

$M_2$. Since $P_1$ is disjoint from $M_2$, the path $P_2$ has to intersect $M_2$. Hence, every $u \to T$-path in $G$ intersects $M_2$ and in particular, intersects $M'$. Equivalently, $u \in r_G(M')$. This completes the proof of Claim 2.1. QED.

We next show that $M'$ is a feasible solution for the problem and is no larger than $M$.

**Claim 2.2.** The set $M'$ intersects every odd $T$-path in $G$ and $|M'| \leq |M|$.

*Proof of Claim 2.2.* By assumption, every odd $T$-path $P$ intersects $M$. If $P$ intersects $M \cap M'$, then it also intersects $M'$. If $P$ intersects $M \setminus M'$, then by Claim 2.1 it also intersects $M'$. Thus, every odd $T$-path in $G$ intersects $M'$. Furthermore, by definition of $M'$, we have

$$|M'| = |M| + (|M_2 \setminus M| - |M_1 \setminus M_2|) \leq |M| + (|M_2| - |M_1|) \leq |M| \quad . \tag{2.3}$$

This completes the proof of Claim 2.2. QED.

**Claim 2.3.** $r_G(M) \subseteq r_G(M')$.

*Proof of Claim 2.3.* Let $u$ be an arbitrary node in $r_G(M)$. The set $M$ is a $u \to T$-separator. Therefore, every $u \to T$-path intersects $M$. We need to show that every $u \to T$-path also intersects $M'$. Let $P$ be a $u \to T$-path. If $P$ intersects $M \cap M'$, then it also intersects $M'$. If $P$ does not intersect $M \cap M'$, then it has to intersect $M \setminus M'$. By Claim 2.1, every $M \setminus M' \to T$-path intersects $M'$. Therefore, $u \in r_G(M')$. This completes the proof of Claim 2.3. QED.

**Claim 2.4.** $r_G(M) \cup f_G(M) \cup M \subseteq r_G(M') \cup f_G(M') \cup M'$.

*Proof of Claim 2.4.* By Claim 2.1, $M \setminus M' \subseteq r_G(M')$; and by Claim 2.3, $r_G(M) \subseteq r_G(M')$. Thus, it remains to prove that $f_G(M) \subseteq r_G(M') \cup f_G(M') \cup M'$. Let $u$ be an arbitrary node in $f_G(M) \setminus (r_G(M') \cup f_G(M') \cup M')$. Since $u \notin f_G(M')$, there is a $T \to u$-path $P_1$ in $G$ that is disjoint from $M'$. But $u \in f_G(M)$. Thus $P_1$ has to intersect $M$, particularly it has to intersect $M \setminus M'$. Let $P_2$ be a subpath of $P_1$ from $M \setminus M'$ to $u$. Since $u \notin r_G(M')$, there is a $u \to T$-path $P_3$ in $G$ that is disjoint from $M'$. The concatenation of $P_2$ and $P_3$ is a path from $M \setminus M'$ to $T$ that is disjoint from $M'$. But by Claim 2.1, every $M \setminus M' \to T$-path in $G$ must intersect $M'$. This contradiction shows that $f_G(M) \subseteq (r_G(M') \cup f_G(M') \cup M')$. This completes the proof of Claim 2.4. QED.

**Claim 2.5.** $r_G(M) \subsetneq r_G(M')$.

*Proof of Claim 2.5.* By Claim 2.3, $r_G(M) \subseteq r_G(M')$. We need to prove that $r_G(M) \neq r_G(M')$. We recall that $M \setminus M' = M_1 \setminus M_2$. Since $M_2$ is an important $v \to T$-separator, it follows that the $v \to T$-separator $M_1$ is not contained in $M_2$. Therefore, $M \setminus M'$ is non-empty. Furthermore, by definition of reverse shadow, $M \setminus M'$ is not contained in $r_G(M)$, but by Claim 2.1, it is contained in $r_G(M')$. This completes the proof of Claim 2.5. QED.

By Claim 2.2, $M'$ is a solution of size not larger than $M$. Therefore, the set $M'$ has the properties claimed in Lemma 2.4. This completes the proof of Lemma 2.4. QED.

We recall that a set $M \subseteq V(G)$ is *thin*, if every node $v \in M$ is not in $r_G(M \setminus \{v\})$.

**Corollary 2.2.** Let $(G, V^\infty, T, k)$ be an input of ODDMULTIWAYNODECUT, where $G$ is a DAG. Let $M^*$ be an optimal solution that maximizes the size of $|r_G(S) \cup f_G(S) \cup S|$ among all optimal solutions $S$. If more than one optimal solution maximizes this quantity, choose the one with largest $|r_G(S)|$. The set $M^*$ is thin and for every node $v \in r_G(M^*)$ there is an important $v \to T$-separator in $M^*$.

*Proof.* The set $M^*$ is thin. If not, there is a node $v \in M^*$ that belongs to $r_G(M^* \setminus \{v\})$. Then $M^* \setminus \{v\}$ is a solution too, contradicting the optimality of $M^*$.

If there is a node $v \in r_G(M^*)$ for which there is no important $v \to T$-separator in $M^*$, then by Lemma 2.4, there exists a solution $M'$ such that $r_G(M) \cup f_G(M) \cup M \subseteq r_G(M') \cup f_G(M') \cup M'$, and $r_G(M) \subsetneq r_G(M')$. This contradicts the choice of $M^*$. Therefore, for every node $v \in r_G(M^*)$ there is an important $v \to T$-separator in $M^*$. QED.

We will use Corollary 2.2 to prove Lemma 2.3.

*Proof of Lemma 2.3.* Let us use `ReverseShadowContainer`$(G, V^\infty, k)$ to denote the algorithm from Theorem 2.3. We will show that Algorithm 2.3 generates the desired set.

By Theorem 2.3, the cardinality of $\mathcal{Z}$ returned by the algorithm is $2^{O(k^2)} \log |V(G)|$. The analysis of the runtime of the algorithm follows from the runtime analysis of the procedure `ReverseShadowContainer` in Theorem 2.3. To prove the correctness of this algorithm, we argue that at least one of the sets in the returned family $\mathcal{Z}$ has the desired properties.

Suppose there exists a solution of size at most $k$ and let $M^*$ be an optimal solution that maximizes the size of $|r_G(S) \cup f_G(S) \cup S|$ among all optimal solutions $S$. If more than one solution maximizes this quantity, choose the one with largest $|r_G(S)|$. By Corollary 2.2, the solution $M^*$ is thin and has the property that every node $v$ in the reverse shadow of $M^*$ has an important $v \to T$-separator contained in $M^*$. By Theorem 2.3, the procedure `ReverseShadowContainer`$(G, V^\infty, k)$ in Line 4 will return a family $\mathcal{Z}_1$ of sets containing a set $Z_1$ that is disjoint from $M^*$ and contains its reverse shadow. Let us fix such a $Z_1$.

**Algorithm 2.3** `ShadowContainer`

**Input:** A digraph $G$ with terminal set $T$, a set $V^\infty$ of protected nodes containing $T$, and $k \in \mathbb{Z}_+$.
**Output:** A set $\mathcal{Z}$ of at most $2^{O(k^2)} \log |V(G)|$ subsets of $V(G)$ with the property that if $(G, T, V^\infty, k)$ admits a solution of size at most $k$, then for some solution $M$ of size at most $k$, there exists a set $Z \in \mathcal{Z}$ that is disjoint from $M$ and contains $s_G(M)$.
Let $G^{\text{rev}}$ denote the graph obtained from $G$ by reversing the orientation of all edges
$\mathcal{Z}_1 \leftarrow$ `ReverseShadowContainer`$(G, V^\infty, k)$
**for** $Z_1 \in \mathcal{Z}_1$ **do**
    $\mathcal{Z}_2 \leftarrow$ `ReverseShadowContainer`$(G^{\text{rev}}, V^\infty \cup Z_1, k)$
    **for** $Z_2 \in \mathcal{Z}_2$ **do**
        $\mathcal{Z} \leftarrow \mathcal{Z} \cup \{Z_1 \cup Z_2\}$
**return** $\mathcal{Z}$

Note that $G^{\text{rev}}$ is a DAG on the same node set as $G$. What's more, any solution for the OddMultiwayNodeCut instance $(G^{\text{rev}}, V^\infty \cup Z_1, T, k)$ is also a solution for the instance $(G, V^\infty, T, k)$. Conversely, a solution for the instance $(G, V^\infty, T, k)$ that is disjoint from $Z_1$ is also a solution for the instance $(G^{\text{rev}}, V^\infty \cup Z_1, T, k)$. Therefore, the set $M^*$ is also an optimal solution to the instance $(G^{\text{rev}}, V^\infty \cup Z_1, T, k)$. We observe that $f_G(S) = r_{G^{\text{rev}}}(S)$ and $r_G(S) = f_{G^{\text{rev}}}(S)$ for all $S \subseteq V(G) \setminus V^\infty$. Therefore, $M^*$ maximizes the size of $r_{G^{\text{rev}}}(S) \cup f_{G^{\text{rev}}}(S) \cup S$ among all optimal solutions $S$ to $(G^{\text{rev}}, V^\infty \cup Z_1, T, k)$. We have the following claim.

**Claim 2.6.** If for an optimal solution $M'$ for the instance $(G^{\text{rev}}, V^\infty \cup Z_1, T, k)$ of OddMultiwayNodeCut we have the inclusions $r_{G^{\text{rev}}}(M^*) \subseteq r_{G^{\text{rev}}}(M')$ and $r_{G^{\text{rev}}}(M^*) \cup f_{G^{\text{rev}}}(M^*) \cup M^* \subseteq r_{G^{\text{rev}}}(M') \cup f_{G^{\text{rev}}}(M') \cup M'$, then $M' = M^*$.

*Proof of Claim 2.6.* On the one hand, $M^*$ maximizes $|r_{G^{\text{rev}}}(S) \cup f_{G^{\text{rev}}}(S) \cup S|$ among all optimal solutions for the instance $(G, V^\infty, T, k)$. On the other hand, $r_{G^{\text{rev}}}(M^*) \cup f_{G^{\text{rev}}}(M^*) \cup M^* \subseteq r_{G^{\text{rev}}}(M') \cup f_{G^{\text{rev}}}(M') \cup M'$. Consequently, the two sets $r_{G^{\text{rev}}}(M^*) \cup f_{G^{\text{rev}}}(M^*) \cup M^*$ and $r_{G^{\text{rev}}}(M') \cup f_{G^{\text{rev}}}(M') \cup M'$ must be equal. Therefore, the set $M' \setminus M^*$ is contained inside $r_{G^{\text{rev}}}(M^*) \cup f_{G^{\text{rev}}}(M^*) \cup M^*$. Since nodes in $f_{G^{\text{rev}}}(M^*)$ are protected in $G^{\text{rev}}$ by construction, the solution $M'$ cannot contain any node from $f_{G^{\text{rev}}}(M^*)$. Since $r_{G^{\text{rev}}}(M^*) \subseteq r_{G^{\text{rev}}}(M')$ and by definition of reverse shadow, $M'$ is disjoint from $r_{G^{\text{rev}}}(M^*)$. Thus, the set $M' \setminus M^*$ is disjoint from $M^*$ and $r_{G^{\text{rev}}}(M^*)$ and $f_{G^{\text{rev}}}(M^*)$, while being contained in $r_{G^{\text{rev}}}(M^*) \cup f_{G^{\text{rev}}}(M^*) \cup M^*$. Hence, $M' \setminus M^* = \emptyset$ or equivalently $M' \subseteq M^*$. Therefore, $M' = M^*$, because $|M'| = |M^*|$. This completes the proof of Claim 2.6. QED.

Suppose there is a node $v \in r_{G^{\text{rev}}}(M^*)$ such that no important $v \to T$-separator in $G^{\text{rev}}$ is contained in $M^*$. Then by Lemma 2.4, there is another optimal solution $M'$ such that

$r_{G^{\mathrm{rev}}}(M^*) \cup f_{G^{\mathrm{rev}}}(M^*) \cup M^* \subseteq r_{G^{\mathrm{rev}}}(M') \cup f_{G^{\mathrm{rev}}}(M') \cup M'$ and $r_{G^{\mathrm{rev}}}(M^*) \subsetneq r_{G^{\mathrm{rev}}}(M')$. By Claim 2.6, the set $M' = M^*$, which contradicts $r_{G^{\mathrm{rev}}}(M^*) \subsetneq r_{G^{\mathrm{rev}}}(M')$. This contradiction shows that for every node $v \in r_{G^{\mathrm{rev}}}(M^*)$, there is an important $v \to T$-separator in $G^{\mathrm{rev}}$ that is contained in $M^*$. Thus, by Theorem 2.3, the procedure $\texttt{ReverseShadowContainer}(G^{\mathrm{rev}}, V^{\infty} \cup Z_1, k)$ from Line 6 will return a family $\mathcal{Z}_2$ of sets containing a set $Z_2$ that is disjoint from $M^*$ and contains $r_{G^{\mathrm{rev}}}(M^*) = f_G(M^*)$. Hence $Z_1 \cup Z_2$ is disjoint from $M^*$ and contains $s_G(M^*)$. QED.

## CHAPTER 3: $(S \to T)$-ODDPATHNODEBLOCKER IN DAGS

In this chapter, we prove Theorem 1.2 by showing nearly-matching hardness of approximation (Theorem 3.1) and approximability results (Theorem 3.2). We also exhibit instances of DAGs for which $\mathcal{P}^{\text{odd-cover-dir}}$ is not half-integral (Theorem 1.3).

### 3.1 HARDNESS OF APPROXIMATION

The main result of this section is the following:

**Theorem 3.1.** $(s \to t)$-OddPathNodeBlocker in DAGs is $\mathsf{NP}$-complete, and has no efficient $(2 - \varepsilon)$-approximation for any $\varepsilon > 0$ assuming the Unique Games Conjecture.

As a first step, we show that $(s \to t)$-OddPathNodeBlocker is in $\mathsf{NP}$. While this is a folklore result, we present the proof for the sake of completeness.

**Lemma 3.1.** There exists a polynomial-time algorithm that, given a DAG $D$ and nodes $s$ and $t$ in $D$, decides whether there exists an odd-length $s \to t$-path in $D$.

*Proof.* We construct a directed bipartite graph $G$ as follows. For each node $v \in V(D)$, introduce nodes $v_L$ and $v_R$ in $G$. For each edge $uv \in E(D)$, add edges $u_L v_R$ and $u_R v_L$ to $G$. We claim that there is an odd-length $s \to t$ path in $D$ if and only if there is an $s_L \to t_R$ path in $G$. Since existence of an $s_L \to t_R$ path is decidable in polynomial time, this would prove the theorem.

We now prove the mentioned claim. Suppose $s = u^0, u^1, u^2, \ldots, u^\ell = t$ is an odd-length $s \to t$-path in $D$ with intermediate nodes $u^1, \ldots, u^{\ell-1}$. Then $s_L = u^0_L, u^1_R, u^2_L, \ldots, u^\ell_m = t_m$ is a path in $G$ and since $\ell$ is odd, we have $m = R$ and hence, the path in $G$ ends in $t_R$. Conversely, suppose $s_L = u^0_L, u^1_R, u^2_L, \ldots, u^\ell_R = t_R$ is a path in $G$. Since the path starts in one part and ends in the other, it must be of odd length. Therefore, $s = u^0, u^1, u^2, \ldots, u^l = t$ is an odd walk in $D$. Since every walk in a DAG is a path, we have an odd-length $s \to t$ path in $D$. QED.

With this result we are now ready to prove Theorem 3.1.

*Proof of Theorem 3.1.* We consider the decision version of problem $(s \to t)$-OddPath-NodeBlocker, where the input consists of a directed acyclic graph $D$ and a non-negative integer $w$ and the goal is to decide if there exists a feasible solution for $(s \to t)$-OddPath-NodeBlocker with at most $w$ nodes. Let $(D, w)$ be an instance of the decision version of $(s \to t)$-OddPathNodeBlocker. By Lemma 3.1, given a set of nodes $U \subseteq V(D)$ of cardinality at most $w$, we can verify in polynomial time whether $D \setminus U$ has no $s \to t$ odd-path. Therefore $(s \to t)$-OddPathNodeBlocker in DAGs is in NP.

To prove NP-hardness of $(s \to t)$-OddPathNodeBlocker in DAGs, we give a reduction from Vertex Cover. Recall that the input to the Vertex Cover problem is an undirected graph $G$ and $k \in \mathbb{Z}$ and the goal is to verify if there exists a vertex cover of size at most $k$. We construct a directed acyclic graph $H$ from $G$ as follows: pick an arbitrary ordering of the nodes and orient the edges $\{u, v\}$ of $G$ as $u \to v$ if $u < v$ in the ordering; we add two new nodes $s, t$ with directed edges $s \to u, u \to t$ for every $u \in V(G)$. The resulting graph $H$ is a directed acyclic graph. A subset $U \subseteq V(G)$ is a vertex cover in $G$ if and only if $U$ is a feasible solution to $(s \to t)$-OddPathNodeBlocker in $H$. QED.

## 3.2 APPROXIMATION AND INTEGRALITY GAP

In this section we present an approximation algorithm of factor 2 for $(s \to t)$-OddPathEdgeBlocker in DAGs. This factor matches the lower bound on the hardness of approximation shown in Section 3.1. We will use the following integer linear programming formulation of $(s \to t)$-OddPathEdgeBlocker and its LP-relaxation.

$$\begin{aligned} \min \quad & \sum_{e \in E(D)} c(e) x_e && \\ \text{subject to} \quad & \sum_{e \in P} x_e \geq 1 && \text{for all odd-length } s \to t \text{ path } P \text{ in } D, \\ & x_e \in \{0, 1\} && \forall\, e \in E(D). \end{aligned} \tag{3.1}$$

where each binary variable $x_e$ indicates whether $e$ is in the solution. This integer program can then be relaxed to a linear program by replacing the constraints $x_e \in \{0, 1\}$ with $x_e \geq 0$. We denote the resulting LP as *odd path blocker LP*.

**Theorem 3.2.** There exists a 2-approximation algorithm for $(s \to t)$-OddPathEdgeBlocker in DAGs.

*Proof.* Our algorithm uses a construction similar to what was described in the proof of Lemma 3.1. Let $D$ be an instance of $(s \to t)$-OddPathEdgeBlocker with edge costs $c : E \to \mathbb{R}_+$. Construct the directed bipartite graph $G$ with $V(G) := \{v_L, v_R : v \in V(D)\}$ and $E(G) := \{u_L v_R, u_R v_L : uv \in E(D)\}$. Define the cost of the edges $u_R v_L$ and $u_L v_R$ to be $c(uv)$. Let $X \subseteq E(G)$ be a minimum $s_L \to t_R$-cut in $G$. Let $F := \{uv : u_L v_R \in X \text{ or } u_R v_L \in X\}$

be the projection of the edges of $X$ onto the edges of $D$. Claims 3.1 and 3.2 below prove that $F$ is a 2-approximate solution for $(s \to t)$-OddPathEdgeBlocker in $D$. QED.

**Claim 3.1.** The set $F$ is a feasible solution for $(s \to t)$-OddPathEdgeBlocker in $D$.

*Proof.* For sake of contradiction, suppose not. Then there must exist an odd $s \to t$-path in $D \setminus F$. Let $s = u^0, u^1, \ldots, u^\ell = t$ be such a path. We note that the edge $u^i u^{i+1}$ is not in $F$ for $i = 0, 1, \ldots, \ell - 1$. Thus, $u_L^i u_R^{i+1}$ and $u_R^i u_L^{i+1}$ are not in $X$. Hence, $s_L = u_L^0, u_R^1, u_L^2, \ldots, u_R^\ell = t_R$ is a path in $G$. This contradicts the feasibility of $X$ as an $s_L \to t_R$-cut. Therefore $F$ is a feasible solution. QED.

Let $x^*$ be an optimal solution to the odd path blocker LP for $D$. Let $c(x)$ denote the objective value of a feasible solution $x$ to the odd path blocker LP for $D$. We use the same notation to denote the cost of an $s_L \to t_R$ cut in $G$.

**Claim 3.2.** The cost of $F$ is at most twice that of $x^*$.

*Proof.* We note that $c(F) \leq c(X)$, by the construction of $F$. It would suffice to show that $c(X) \leq 2c(x^*)$.

Define $Y : E(G) \to \mathbb{R}_+$ by setting $Y(u_L v_R) = Y(u_R v_L) = x^*(uv)$ and let $c(Y) := \sum_{e \in E(G)} c_e Y(e)$. We have that $c(Y) = 2c(x^*)$. We recall that any minimum $s_L \to t_R$-cut has the same value as an optimal solution to the following path blocking integer program, as well as its linear programming relaxation:

$$\min \sum_{e \in E(G)} w(e) y_e \tag{3.2}$$

$$\sum_{e \in P} y_e \geq 1 \ \forall \ s_L \to t_R \text{ path } P \text{ in } G,$$

$$y_e \in \{0, 1\} \ \forall \ e \in E(G).$$

Hence, it suffices to prove that $Y$ is a feasible solution to the LP-relaxation of the above integer program. For sake of contradiction, suppose it is not. It means that there is an $s_L \to t_R$ path $s_L = u_L^0, u_R^1, u_L^2, \ldots, u_R^\ell = t_R$ in $G$, such that the sum of the $Y$-values over its edges is less than one. Since $s_L$ and $t_R$ are in different parts of $G$, the length $\ell$ of this path must be odd. Now consider the path $s = u^0, u^1, \ldots, u^\ell = t$ in $D$. The sum of $x^*$ values on its edges is also less than one and since $\ell$ is odd, this contradicts the feasibility of $x^*$ to the odd path blocker LP for $D$. Therefore $Y$ must be feasible. QED.

The proof of Theorem 3.2 also yields the following corollary.

**Corollary 3.1.** The integrality gap of the odd path blocker LP in DAGs is at most 2.

## 3.3 EXTREME POINT STRUCTURE OF THE ODD PATH COVER POLYHEDRON

In this section, we examine the extreme point structure of the polyhedron $\mathcal{P}^{\text{odd-cover-dir}}$ defined in Section 1 but for the case of DAGs. Concretely, $\mathcal{P}^{\text{odd-cover-dir}}$ in a DAG is the set of feasible solutions to the odd path blocker LP defined in Section 3.2 and is given in the statement of Theorem 1.3. We say that a polyhedron is *half-integral* if each of its extreme points is a half-integral vector (i.e., each coordinate is an integer multiple of 1/2). Half-integrality is a desirable property in polyhedra associated with covering LPs, because it yields a simple rounding scheme that achieves an approximation factor of 2. Schrijver and Seymour [7] showed that $\mathcal{P}^{\text{odd-cover}}$ in undirected graphs is half-integral. In this section, we exhibit a DAG for which $\mathcal{P}^{\text{odd-cover-dir}}$ has a non-half-integral extreme point.

*Proof of Theorem 1.3.* Consider the DAG in Fig. 3.1. Subdivide every edge, except for the five thick edges in the top row, into two. All edges have unit cost. Let us denote the resulting DAG as $\Gamma = (V, E)$.

We first observe that in $\Gamma$, every odd-length path from $s$ to $t$ must use an odd number of the thick edges. Let $m$ be the number of edges in this network. We introduce a solution $x$ for this instance that is not half-integral. Set

$$\begin{aligned} &x(A) = 1/4, x(B) = 1/2, x(C) = 1/4, x(D) = 1/2, \\ &x(E) = 1/4, x(F) = 1/4, x(G) = 3/4, x(H) = 1/4, \end{aligned} \tag{3.3}$$

and let $x(e)$ be zero for every other edge $e$. The $x$ value of a subdivided edge is assigned to exactly one of the two resulting edges, with the value of the other edge being set to zero. It can be verified that $x \in \mathcal{P}^{\text{odd-cover-dir}}$ for this instance.

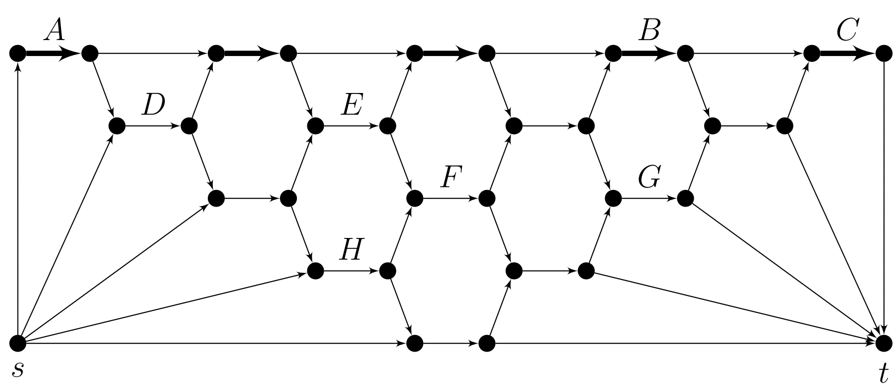



Figure 3.1: An instance of $(s \to t)$-OddPathEdgeBlocker. All edges are subdivided into two, except for the five thick edges in the top row.

Next, we show that this solution $x$ is an extreme point of $\mathcal{P}^{\text{odd-cover-dir}}$, i.e., $x$ is an optimal solution to the odd path blocker LP. For this, we find a solution to the dual linear program with the same objective value. Let $\mathcal{Q}_{s\to t}$ be the collection of edge-sets corresponding to odd-length paths from $s$ to $t$. As the dual of odd path blocker LP, we obtain:

$$\begin{aligned} \max \quad & \sum_{P\in\mathcal{Q}_{s\to t}} f_p && \\ \text{subject to} \quad & \sum_{P\in\mathcal{Q}_{s\to t}:e\in P} f_P \le c(e), && \text{for each edge } e \\ & f_P \ge 0, && \text{for all } P \in \mathcal{Q}_{s\to t} \end{aligned} \tag{3.4}$$

Let us call the dual LP as odd flow packing LP. The dual formulation describes the problem of sending the maximum flow along odd paths in the network, such that the amount of flow going through each edge does not exceed its capacity.

Consider the following paths in Fig. 3.2:

$$P_1 = (29, 1, 2, 11, 12, 19, 20, 25, 26, 30, 31, 32) \tag{3.5}$$
$$P_2 = (29, 11, 12, 3, 4, 13, 14, 21, 22, 27, 28, 32) \tag{3.6}$$
$$P_3 = (29, 19, 20, 13, 14, 5, 6, 15, 16, 23, 24, 32) \tag{3.7}$$
$$P_4 = (29, 25, 26, 21, 22, 15, 16, 7, 8, 17, 18, 32) \tag{3.8}$$
$$P_5 = (29, 30, 31, 27, 28, 23, 24, 17, 18, 9, 10, 32) \tag{3.9}$$
$$P_6 = (29, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 32) \tag{3.10}$$

The dual solution that we introduce sends a flow of value 1/2 along each of $P_1, P_2, \ldots, P_6$. The total flow, therefore would be 3. By strong duality condition, the feasible solutions of an LP and its dual match only at optimal solutions. Therefore, the primal solution $x$ is optimal.

Finally, we prove that the primal solution $x$ is an extreme point of the polyhedron $\mathcal{P}^{\text{odd-cover-dir}}$ for this instance. For this, we present $m$ linearly independent constraints of the odd path blocker LP that are satisfied as equations by $x$. For each of the $m - 8$ edges that are not in the support of $x$, the non-negativity constraint is tight. Consider $P_1, P_2, \ldots, P_6$ above, along with the following two odd-length $s \to t$ paths in $D$:

$$P_7 = (29, 1, 2, 3, 4, 5, 6, 15, 16, 23, 24, 32) \tag{3.11}$$
$$P_8 = (29, 11, 12, 3, 4, 5, 6, 7, 8, 17, 18, 32) \tag{3.12}$$

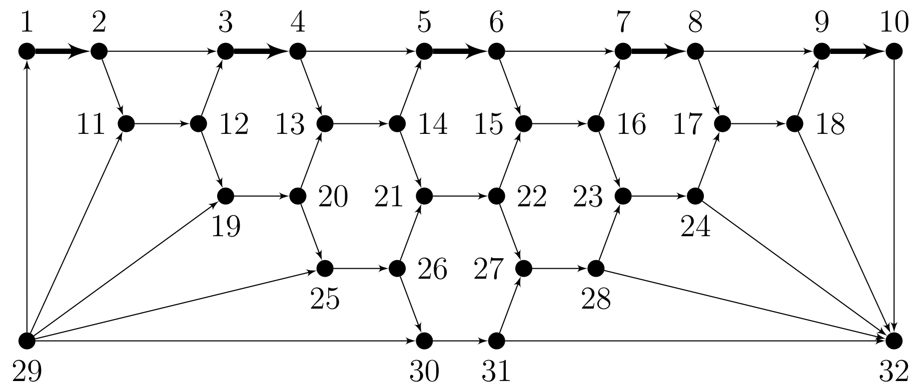



Figure 3.2: In this dual solution, flows of value 1/2 are being sent through $p_1, p_2, \ldots, p_6$.

We observe that the constraints corresponding to each of these paths are tight with respect to $x$. It remains to prove that these $m$ constraints are linearly independent. Since each of the non-negativity constraints have exactly one non-zero entry and no two of them have the same non-zero entry, they are linearly independent. To prove that all the constraints are linearly independent, it remains to show that the path constraints are linearly independent when restricted to the edges not in the support of $x$. This can be verified by computing the determinant of the corresponding constraint matrix. QED.

# CHAPTER 4: $\{S,T\}$-ODDPATHEDGEBLOCKER IN UNDIRECTED GRAPHS

In this chapter, we focus on the approximability of $\{s,t\}$-OddPathEdgeBlocker. We will show inapproximability results and an integrality gap instance which suggests that new techniques might be needed to improve on the approximation factor.

## 4.1 HARDNESS OF APPROXIMATION

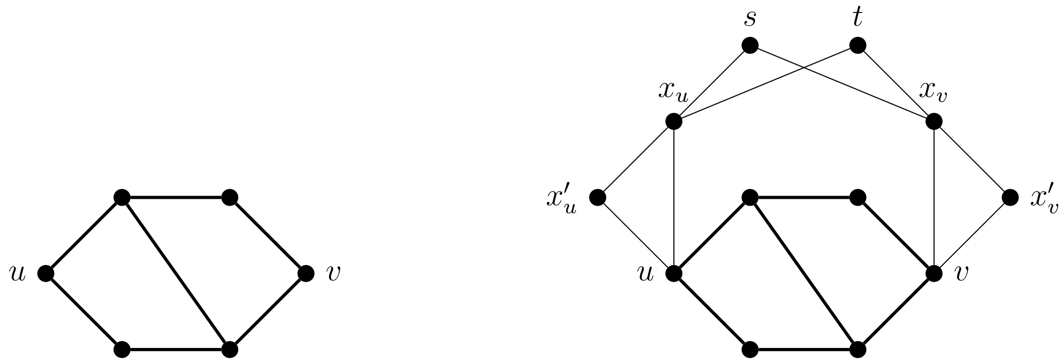
In this section we prove Theorem 1.4, i.e., $\mathsf{NP}$-hardness of $\{s,t\}$-OddPathEdgeBlocker.

**Theorem 1.4.** $\{s,t\}$-OddPathEdgeBlocker is $\mathsf{NP}$-hard and has no efficient $(6/5-\varepsilon)$-approximation assuming the Unique Games Conjecture.

*Proof.* Edmonds gave a polynomial time algorithm to decide whether a given undirected graph with nodes $s$ and $t$ has an odd-length $s-t$ path; cf. LaPaugh and Papadimitriou [6]. Therefore, given a candidate solution, one can verify its feasibility in polynomial time. Thus, $\{s,t\}$-OddPathEdgeBlocker is in $\mathsf{NP}$. We will show that the decision version of $\{s,t\}$-OddPathEdgeBlocker is $\mathsf{NP}$-complete by a polynomial-time reduction from MultiwayCut. We recall that the input to MultiwayCut is an undirected graph $G$, a collection $T$ of nodes in $G$ known as terminals and $k \in \mathbb{Z}_+$, and the goal is to verify if there exists a subset of at most $k$ edges of $G$ whose deletion ensures that no pair of terminals can reach each other.

Suppose $(G,T,k)$ is an instance of MultiwayCut where $G$ is an undirected graph with $n$ nodes and $m$ edges, and $T \subseteq V(G)$ is the set of terminals to be separated. We obtain a graph $H$ from $G$ as follows. Introduce nodes $s,t,x_v,x'_v$ and let $H$ have vertex set $V(H) := V(G) \cup \{s,t\} \cup \{x_v, x'_v : v \in T\}$. Further, let $H$ have edge set $E(H) := E(G) \cup \{\{x_v,v\},\{x'_v,v\},\{x_v,x'_v\},\{s,x_v\},\{t,x_v\} : v \in T\}$; replace every edge in $E(H)\setminus E(G)$ by $m+1$ parallel edges (see Fig. 4.1). For $k \leq m$, we claim that the MultiwayCut instance $(G,T,k)$ has a solution of size at most $k$ if and only if the $\{s,t\}$-OddPathEdgeBlocker instance $H$ has a solution of size at most $k$. Suppose $F \subseteq E(G)$ is a solution to MultiwayCut in $(G,T)$. If $F$ is not a solution to $\{s,t\}$-OddPathEdgeBlocker in $H$, then there is an odd path from $s$ to $t$ in $H \setminus F$. By construction, this path is of the form $sP_1vP_2uP_3t$, where $v,u \in T$. Hence, there is a $v-u$-path $P_2$ in $G \setminus F$, contradicting the feasibility of $F$ as a solution to MultiwayCut in $(G,T)$.

Conversely, suppose $F \subseteq E(H)$ is a solution to $\{s,t\}$-OddPathEdgeBlocker in $H$ of size at most $k$. Since $k \leq m$, and by construction, we may assume that $F \subseteq E(G)$. Suppose




(a) An instance of MultiwayCut with terminal set $\{u, v\}$.

(b) The reduced instance of $\{s, t\}$-OddPathEdgeBlocker

Figure 4.1: Illustration of the reduction from MultiwayCut to $\{s, t\}$-OddPathEdgeBlocker.

$F$ is not a solution to MultiwayCut in $(G, T)$. Then there is a $v - u$-path $P$ in $G \setminus F$ for some distinct $u, v \in T$. If $P$ is even, then let $Q$ be the path $sx_v vPux'_u x_u t$ in $G'$ and if $P$ is odd, then let $Q$ be the path $sx_v vPux_u t$ in $G'$. In both cases, the path $Q$ is an odd-length $s-t$ in $H - F$, contradicting the feasibility of $F$ as a solution to $\{s, t\}$-OddPathEdgeBlocker in $(G, E^\infty, s, t)$.

We note that the above reduction is an approximation factor preserving reduction. It is known that MultiwayCut is $\mathsf{NP}$-hard and does not admit a polynomial-time approximation scheme, unless $\mathsf{P} = \mathsf{NP}$ [28]. Moreover, there is no efficient $(6/5 - \varepsilon)$-approximation for MultiwayCut assuming the Unique Games Conjecture [29, 30]. Hence, the results follow. QED.

## 4.2 INTEGRALITY GAP

By the half-integrality of the extreme points of $P^{\text{odd-cover}}$ as established by Schrijver and Seymour [7], we have a 2-approximation algorithm for $\{s, t\}$-OddPathEdgeBlocker by solving the LP-relaxation of the odd path blocker LP. The following proposition shows that the integrality gap of the odd path blocker LP is indeed 2 and hence we cannot hope to improve on the 2-approximation using the odd path blocker LP.

**Lemma 4.1.** The integrality gap of the following odd path blocker LP for $\{s, t\}$-OddPath-

EdgeBlocker is at least 2:

$$\begin{aligned} \min \sum_{e \in E} c(e) x_e & & \quad (4.1) \\ \sum_{e \in P} x_e \geq 1, & \qquad \text{for all odd-length } s-t \text{ paths } P \text{ in } G \\ x_e \geq 0, & \qquad e \in E(G). \end{aligned}$$

*Proof.* For every $k \in \mathbb{Z}_+$, we construct a graph $G$ for which the integrality gap of the odd path blocker LP is at least $2(1-1/k)$. Let $S_k$ be the star graph on $k$ nodes (i.e., $V(S_k) := \{u_1, u_2, \ldots, u_k\}$ and $E(S_k) := \{u_i u_k : i \in \{1, \ldots, k-1\}\}$). Let $T$ be the set of leaves of $S_k$. Let $G$ be the graph obtained from $(S_k, T)$ by applying the construction in the proof of Theorem 1.4. An optimal solution to the odd path blocker LP for $G$, assigns the value $1/2$ to every edge of $S_k$, while an optimal integral solution removes all but one edges of $S_k$ that are in $G$. Therefore, the ratio of the integral solution to the fractional solution is $(k-1)/(k/2) = 2(1-1/k)$. QED.

The same arguments as in Lemma 4.1 also show that the integrality gap of the odd path node blocker LP for $\{s,t\}$-OddPathNodeBlocker is also at least 2.

# CHAPTER 5: CONCLUSIONS

In this work, we studied a natural cut problem with parity constraints in undirected and directed graphs. Our main results are fixed-parameter algorithms parameterized by the solution size, as well as constant-factor polynomial-time approximation algorithms and inapproximability reductions.

Several questions lend themselves for future work. Firstly, it would be interesting to determine the exact approximability bound achievable in polynomial time for OddMultiwayEdgeCut in undirected graphs, closing the gap between the lower bound of 6/5 and the upper bound of 2.

Secondly, an important line of investigation in parameterized complexity is the design of fixed-parameter algorithms that have the best possible asymptotic dependence on the parameters (modulo the Exponential-Time Hypothesis), with only linear time dependence on the instance size [13, 17, 19, 31, 32, 33]. Thus, it would be interesting to know whether the proposed fixed-parameter algorithms for OddMultiwayNodeCut in DAGs can be expedited to run in time $2^{O(k)} \cdot O(n)$.

Finally, we ask about the parameterized complexity of the common generalization of OddMultiwayNodeCut and Multicut known as Odd Multicut: given a graph $G$ with terminal pairs $\{s_1, t_1\}, \ldots, \{s_p, t_p\}$ and an integer $k$, decide if some set $S$ of at most $k$ nodes intersects all odd-paths between $s_i$ and $t_i$, for $i = 1, \ldots, p$. Does this problem admit a fixed-parameter algorithm parameterized by the solution size $k$?

## REFERENCES


[1] M. Chudnovsky, J. Geelen, B. Gerards, L. Goddyn, M. Lohman, and P. Seymour, "Packing non-zero $A$-paths in group-labelled graphs," *Combinatorica*, vol. 26, no. 5, pp. 521–532, 2006.

[2] S. Ibrahimpur and C. Swamy, "Min-max theorems for packing and covering odd $(u, v)$-trails," in *International Conference on Integer Programming and Combinatorial Optimization*. Springer, 2017, pp. 279–291.

[3] A. Schrijver, *Combinatorial optimization. Polyhedra and efficiency. Vol. A*, ser. Algorithms and Combinatorics. Springer-Verlag, Berlin, 2003, vol. 24, paths, flows, matchings, Chapters 1–38.

[4] J. Edmonds, "Maximum matching and a polyhedron with $0, 1$-vertices," *J. Res. Nat. Bur. Standards Sect. B*, vol. 69B, pp. 125–130, 1965.

[5] J. Edmonds, "Paths, trees, and flowers," *Canad. J. Math.*, vol. 17, pp. 449–467, 1965.

[6] A. S. LaPaugh and C. H. Papadimitriou, "The even-path problem for graphs and digraphs," *Networks*, vol. 14, no. 4, pp. 507–513, 1984.

[7] A. Schrijver and P. D. Seymour, "Packing odd paths," *J. Combin. Theory Ser. B*, vol. 62, no. 2, pp. 280–288, 1994.

[8] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric algorithms and combinatorial optimization*, 2nd ed., ser. Algorithms and Combinatorics. Springer-Verlag, Berlin, 1993, vol. 2.

[9] M. Cygan, F. V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh, *Parameterized algorithms*. Springer, Cham, 2015.

[10] K. Bringmann, D. Hermelin, M. Mnich, and E. J. van Leeuwen, "Parameterized complexity dichotomy for Steiner Multicut," *J. Comput. System Sci.*, vol. 82, no. 6, pp. 1020–1043, 2016.

[11] R. Chitnis, M. Hajiaghayi, and D. Marx, "Fixed-parameter tractability of directed multiway cut parameterized by the size of the cutset," *SIAM J. Comput.*, vol. 42, no. 4, pp. 1674–1696, 2013.

[12] D. Marx, "Parameterized graph separation problems," *Theoret. Comput. Sci.*, vol. 351, no. 3, pp. 394–406, 2006.

[13] D. Marx, B. O'Sullivan, and I. Razgon, "Finding small separators in linear time via treewidth reduction," *ACM Trans. Algorithms*, vol. 9, no. 4, pp. Art. 30, 35, 2013.

[14] D. Marx and I. Razgon, "Fixed-parameter tractability of multicut parameterized by the size of the cutset," *SIAM J. Comput.*, vol. 43, no. 2, pp. 355–388, 2014.

[15] S. Kratsch and M. Wahlström, "Compression via matroids: a randomized polynomial kernel for odd cycle transversal," *ACM Trans. Algorithms*, vol. 10, no. 4, pp. Art. 20, 15, 2014.

[16] D. Lokshtanov and M. S. Ramanujan, "Parameterized tractability of multiway cut with parity constraints," in *Proc. ICALP 2012*, ser. Lecture Notes Comput. Sci., 2012, vol. 7391, pp. 750–761.

[17] D. Lokshtanov, M. S. Ramanujan, S. Saurabh, and M. Zehavi, "Parameterized complexity and approximability of directed odd cycle transversal," 2017, https://arxiv.org/abs/1704.04249.

[18] B. Reed, K. Smith, and A. Vetta, "Finding odd cycle transversals," *Oper. Res. Lett.*, vol. 32, no. 4, pp. 299–301, 2004.

[19] M. Etscheid and M. Mnich, "Linear kernels and linear-time algorithms for finding large cuts," *Algorithmica*, vol. 80, pp. 2574–2615, 2018.

[20] B. M. P. Jansen, M. Pilipczuk, and E. J. van Leeuwen, "A deterministic polynomial kernel for odd cycle transversal and vertex multiway cut in planar graphs," in *Proc. STACS 2019*, ser. Leibniz Int. Proc. Informatics, vol. 126, 2019, pp. 39:1–39:18.

[21] N. Kakimura and K. ichi Kawarabayashi, "Fixed-parameter tractability for subset feedback set problems with parity constraints," *Theoret. Comput. Sci.*, vol. 576, pp. 61–76, 2015.

[22] S. Kolay, D. Lokshtanov, F. Panolan, and S. Saurabh, "Quick but odd growth of cacti," *Algorithmica*, vol. 79, pp. 271–290, 2017.

[23] S. Khot and O. Regev, "Vertex cover might be hard to approximate to within $2 - \epsilon$," *J. Comput. System Sci.*, vol. 74, no. 3, pp. 335–349, 2008.

[24] S. Ramanujan, "Parameterized graph separation problems: New techniques and algorithms," Ph.D. dissertation, The Institute of Mathematical Sciences, Chennai, 2013.

[25] D. Lokshtanov, P. Misra, M. S. Ramanujan, and S. Saurabh, "Hitting selected (odd) cycles," *SIAM J. Discrete Math.*, vol. 31, no. 3, pp. 1581–1615, 2017.

[26] D. Lokshtanov, N. S. Narayanaswamy, V. Raman, M. S. Ramanujan, and S. Saurabh, "Faster parameterized algorithms using linear programming," *ACM Trans. Algorithms*, vol. 11, no. 2, pp. Art. 15, 31, 2014.

[27] R. Chitnis, M. Cygan, M. Hajiaghayi, and D. Marx, "Directed subset feedback vertex set is fixed-parameter tractable," *ACM Trans. Algorithms*, vol. 11, no. 4, pp. Art. 28, 28, 2015.

[28] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis, “The complexity of multiterminal cuts,” *SIAM J. Comput.*, vol. 23, no. 4, pp. 864–894, 1994.

[29] R. Manokaran, J. Naor, P. Raghavendra, and R. Schwartz, “SDP gaps and UGC hardness for multiway cut, 0-extension, and metric labeling [extended abstract],” in *Proc. STOC 2008*, 2008, pp. 11–20.

[30] H. Angelidakis, Y. Makarychev, and P. Manurangsi, “An improved integrality gap for the Călinescu-Karloff-Rabani relaxation for multiway cut,” in *Proc. IPCO 2017*, ser. Lecture Notes Comput. Sci., vol. 10328, 2017, pp. 39–50.

[31] Y. Iwata, K. Oka, and Y. Yoshida, “Linear-time FPT algorithms via network flow,” in *Proc. SODA 2014*, 2014, pp. 1749–1761.

[32] D. Lokshtanov, M. S. Ramanujan, and S. Saurabh, “A linear time parameterized algorithm for directed feedback vertex set,” 2016, https://arxiv.org/abs/1609.04347.

[33] M. S. Ramanujan and S. Saurabh, “Linear time parameterized algorithms via skew-symmetric multicuts,” in *Proc. SODA 2014*, 2014, pp. 1739–1748.

# APPENDIX A: EQUIVALENCE OF ODDMULTIWAYEDGECUT AND ODDMULTIWAYNODECUT

In this chapter, we show an approximation-preserving and parameter-preserving equivalence between OddMultiwayEdgeCut and OddMultiwayNodeCut in directed graphs. To establish the notation, we restate the definitions of the two problems below.

| OddMultiwayEdgeCut | Parameter: $k$ |
|---|---|
| *Input:* A directed acyclic graph $G$ with a set $T \subseteq V(G)$ of terminal nodes and a set $E^\infty \subseteq E(G)$ of protected edges, and an integer $k \in \mathbb{Z}_+$. | |
| *Task:* Verify if there exists an odd multiway edge cut of $T$ in $G$ of size at most $k$ and disjoint from $E^\infty$, that is, a set $M \subseteq E(G) \setminus E^\infty$ of edges that intersects every odd $T$-path in $G$. | |

Figure A.1: Statement of the OddMultiwayEdgeCut problem in DAGs

| OddMultiwayNodeCut | Parameter: $k$ |
|---|---|
| *Input:* A directed acyclic graph $G$ with a set $T \subseteq V$ of terminal nodes and a set $V^\infty \subseteq V(G)$ of protected nodes, and an integer $k \in \mathbb{Z}_+$. | |
| *Task:* Verify if there exists an odd multiway node cut in $G$ of size at most $k$ and disjoint from $V^\infty$, that is, a set $M \subseteq V(G) \setminus V^\infty$ of nodes that intersects every odd $T$-path in $G$. | |

Figure A.2: Statement of the OddMultiwayNodeCut problem in DAGs

**Lemma A.1.** There exist approximation-preserving and parameter-preserving reductions between OddMultiwayEdgeCut in directed graphs and OddMultiwayNodeCut in directed graphs.

*Proof.* We first show an approximation preserving and parameter preserving reduction from OddMultiwayEdgeCut to OddMultiwayNodeCut. Let $\mathcal{I} = (G, E^\infty, T, k)$ be an instance of the OddMultiwayEdgeCut problem in directed graphs. We create a new graph $G'$ by subdividing every edge $e \in E(G) \setminus E^\infty$ into three edges, by introducing two new nodes $x_e$ and $y_e$. By construction, $G'$ is a directed graph too. Define $V^\infty := V(G)$. We claim that the instance $\mathcal{I}$ of OddMultiwayEdgeCut has a solution of size, say $r$, if and only if the instance $\mathcal{I}' = (G', V^\infty, T, k)$ of OddMultiwayNodeCut has some solution of size $r$.

Let $M \subseteq E(G) \setminus E^\infty$ be an odd edge multiway cut in $G$. We claim that $M' := \{x_e : e \in M\}$ is an odd multiway cut in $G'$. Let $P'$ be an odd $T$-path in $G'$. Let $P$ be the corresponding path in $G$. By construction, $P$ has the same parity as $P'$ and therefore, is odd. Since $M$ is an odd multiway cut in $G$, it must contain an edge $e \in E(P)$. Therefore, $M'$ contains $x_e$. Thus, every odd $T$-path in $G'$ intersects $M'$. Hence, $M'$ is a solution for the instance $\mathcal{I}'$ of the OddMultiwayNodeCut problem. Moreover, $|M'| = |M|$.

Conversely, let $N'$ be a solution to the instance $\mathcal{I}'$ of the OddMultiwayNodeCut problem. Define $N := \{e : x_e \in N' \text{ or } y_e \in N'\}$. By construction, $N$ is disjoint from $E^\infty$. Let $P$ be an odd $T$-path in $G$ and let $P'$ be the corresponding path in $G'$. The path $P'$ is an odd $T$-path in $G'$. Since $N'$ is an odd multiway node cut in $G'$, it must contain a node $v \in V(P')$. Since $V(G) \subseteq V^\infty$, it must be the case that $v \in \bigcup_{e \in E(G)} \{x_e, y_e\}$. Suppose $v \in \{x_e, y_e\}$ for some $e \in E(G)$. By construction, we have $e \in N$. Therefore, $N$ includes an edge from every odd $T$-path in $G$. Thus, $N$ is a solution to the instance $\mathcal{I}$ of the OddMultiwayEdgeCut problem. Moreover, $|N| = |N'|$. We next show an approximation preserving and parameter preserving reduction from OddMultiwayNodeCut to OddMultiwayEdgeCut. Let $\mathcal{I} = (G, V^\infty, T, k)$ be an instance of the OddMultiwayNodeCut problem in directed graphs. We create a new graph $G'$ as follows. For every node $v \in V(G)$, we create three nodes $v_{\text{in}}$, $v_{\text{mid}}$ and $v_{\text{out}}$ and put an edge from $v_{\text{in}}$ to $v_{\text{mid}}$ and an edge from $v_{\text{mid}}$ to $v_{\text{out}}$. For every edge $uv \in E(G)$, we put an edge from $u_{\text{out}}$ to $v_{\text{in}}$ in $G'$. Define $E^\infty := \{u_{\text{out}}v_{\text{in}} : uv \in E(G)\} \cup \{v_{\text{in}}v_{\text{mid}}, v_{\text{mid}}v_{\text{out}} : v \in V^\infty\}$ and define $T'$ as $\{t_{\text{in}} : t \in T\}$. We claim that the instance $\mathcal{I}$ of the OddMultiwayNodeCut problem has a solution of size, say $r$, if and only if the instance $\mathcal{I}' := (G', E^\infty, T', k)$ of OddMultiwayEdgeCut has a solution of size $r$.

Let $M \subseteq V(G)$ be a solution to the instance $\mathcal{I}$ of the OddMultiwayEdgeCut. We claim that $M' := \{v_{\text{mid}}v_{\text{out}} : v \in M\}$ is an odd multiway cut in $G'$. By construction, $M'$ is disjoint from $E^\infty$. We show that it intersects every odd $T'$-path in $G'$. Let $P'$ be an odd $T'$-path in $G'$. Let $P$ be the corresponding path in $G$. By construction, $P$ has the same parity as $P'$ and therefore, is odd. Since $M$ is an odd multiway cut in $G$, it must contain a node $v \in V(P)$. Therefore, $M'$ contains $v_{\text{mid}}v_{\text{out}}$. Thus, every odd $T'$-path in $G'$ intersects $M'$. Hence, $M'$ is a solution for the instance $\mathcal{I}'$ of OddMultiwayEdgeCut. Moreover $|M'| = |M|$.

Conversely, suppose $N'$ is a solution to the instance $\mathcal{I}'$ of the OddMultiwayNode-Cut problem. Define $N := \{v : v_{\text{in}}v_{\text{mid}} \in N' \text{ or } v_{\text{mid}}v_{\text{out}} \in N'\}$. By construction, $N$ is disjoint from $V^\infty$. Let $P$ be an odd $T$-path in $G$ and let $P'$ be the corresponding $T'$-path in $G'$. The path $P'$ is an odd $T'$-path in $G'$. Since $N'$ is an odd multiway edge cut in $G'$, it must contain an edge $e \in E(P')$. By choice of $E^\infty$, the edge $e$ has to be in $\{v_{\text{in}}v_{\text{mid}}, v_{\text{mid}}v_{\text{out}} : v \in V(G) \setminus V^\infty\}$. Suppose $e \in \{v_{\text{in}}v_{\text{mid}}\}$ for some $v \in V(G) \setminus V^\infty$. By construction, $v \in N$. Therefore, $N$ includes a node from every odd $T$-path in $G$. Thus, $N$ is a solution to the instance $\mathcal{I}$ of the OddMultiwayNodeCut problem. Moreover, $|N| = |N'|$. QED.