Parameterized Tractability of Multiway Cut with Parity Constraints

Daniel Lokshtanov[1] and M. S. Ramanujan[2]

[1] University of California, San Diego, USA.
daniello@ii.uib.no
[2] The Institute of Mathematical Sciences, Chennai, India.
msramanujan@imsc.res.in

Abstract. In this paper, we study a parity based generalization of the classical MULTIWAY CUT problem. Formally, we study the PARITY MULTIWAY CUT problem, where the input is a graph G, vertex subsets T_e and T_o ($T = T_e \cup T_o$) called terminals, a positive integer k and the objective is to test whether there exists a k-sized vertex subset S such that S intersects all odd paths from $v \in T_o$ to $T \setminus \{v\}$ and all even paths from $v \in T_e$ to $T \setminus \{v\}$. When $T_e = T_o$, this is precisely the classical MULTIWAY CUT problem. If $T_o = \emptyset$ then this is the EVEN MULTIWAY CUT problem and if $T_e = \emptyset$ then this is the ODD MULTIWAY CUT problem. We remark that even the problem of deciding whether there is a set of at most k vertices that intersects all odd paths between a pair of vertices s and t is NP-complete. Our primary motivation for studying this problem is the recently initiated parameterized study of parity versions of graphs minors (Kawarabayashi, Reed and Wollan, FOCS 2011) and separation problems similar to MULTIWAY CUT. The area of design of parameterized algorithms for graph separation problems has seen a lot of recent activity, which includes algorithms for MULTI-CUT on undirected graphs (Marx and Razgon, STOC 2011, Bousquet, Daligault and Thomassé, STOC 2011), k-WAY CUT (Kawarabayashi and Thorup, FOCS 2011), and MULTIWAY CUT on directed graphs (Chitnis, Hajiaghayi and Marx, SODA 2012). A second motivation is that this problem serves as a good example to illustrate the application of a generalization of important separators which we introduce, and can be applied even when most of the recently develped tools fail to apply. We believe that this could be a useful tool for several other separation problems as well. We obtain this generalization by dividing the graph into slices with small boundaries and applying a divide and conquer paradigm over these slices. We show that PARITY MULTIWAY CUT is fixed parameter tractable (FPT) by giving an algorithm that runs in time $f(k)n^{\mathcal{O}(1)}$. More precisely, we show that instances of this problem with solutions of size $\mathcal{O}(\log \log n)$ can be solved in polynomial time. Along with this new notion of generalized important separators, our algorithm also combines several ideas used in previous parameterized algorithms for graph separation problems including the notion of important separators and randomized selection of important sets to simplify the input instance.

1 Introduction

A fundamental min-max theorem about connectivity in graphs is Menger's Theorem, which states that the maximum number of vertex disjoint paths between two vertices s

and t, is equal to the minimum number of vertices whose removal separates these two vertices. Indeed, a maximum set of vertex disjoint paths between s, t and a minimum size set of vertices separating these two vertices can be computed in polynomial time. A known generalization of this theorem, commonly known as Mader's T-path Theorem [21] states that, given a graph G and a subset T of vertices, there are either k vertex disjoint paths with only the end points in T (such paths are called *T-paths* and if their length is odd (even), then *odd (even)* T-paths), or there is a vertex set of size at most $2k$ which intersects every T-path. Although computing a maximum set of vertex disjoint T-paths can be done in polynomial time by using matching techniques, the decision version of the dual problem of finding a minimum set of vertices that intersects every T-path is NP-complete for $|T| > 2$. Formally, this problem is the classical MULTIWAY CUT problem, where the input is a graph G, a subset of vertices T called terminals, a positive integer k and the objective is to test whether there exists a k-sized vertex subset that intersects every T-path. This is a very well studied problem in terms of approximation, as well as parameterized algorithms [2, 9, 22]. In this paper we study a generalization of this classical MULTIWAY CUT problem to a parity version. Formally, we study the PARITY MULTIWAY CUT problem which is defined as follows.

PARITY MULTIWAY CUT (PMWC)
Instance: A graph $G = (V, E)$, vertex subsets T_e and T_o ($T = T_e \cup T_o$), integer k.
Parameter: k
Question: Is there a vertex set S of size at most k which intersects
1. all odd paths from a vertex $v \in T_o$ to some other vertex $u \in T \setminus \{v\}$,
2. all even paths from a vertex $v \in T_e$ to some other vertex $u \in T \setminus \{v\}$?

When $T_e = T_o$, this is precisely the classical MULTIWAY CUT problem. If $T_o = \emptyset$ then this is the EVEN MULTIWAY CUT (EMWC) problem and if $T_e = \emptyset$ then this is the ODD MULTIWAY CUT (OMWC) problem.

Our main motivation for studying this particular generalization is the recently initiated parameterized study of parity versions of graphs minors by Kawarabayashi, Reed and Wollan [15] and separation problems similar to MULTIWAY CUT [1, 4, 23]. The area of design of parameterized algorithms for graph separation problems has seen a lot of recent activity, which includes algorithms for MULTI-CUT on undirected graphs [23, 1], k-WAY CUT [16] and MULTIWAY CUT on directed graphs [4]. Furthermore, recently, Geelen, Gerards, Reed, Seymour and Vetta [10] proved an odd variant of Mader's T-path Theorem. They showed that, given a graph G and a subset T of vertices, there are either k vertex disjoint odd T-paths, or there is a vertex set of size at most $2k$ which intersects every odd T-path. This result has already turned out to be useful in graph theory [10, 18], as well as in the design of parameterized algorithms [11, 13, 14]. This result was crucial in settling the parameterized complexity of finding k vertex disjoint odd length cycles in a graph [14]. Observe that, this odd variant of Mader's T-path Theorem naturally gives rise to the OMWC problem, a special case of PMWC.

The goal of parameterized complexity is to find ways of solving NP-hard problems more efficiently than by brute force. Here, the aim is to restrict the combinatorial explosion of computational difficulty to a parameter that is hopefully much smaller than the input size. Formally, a *parameterization* of a problem is the assignment of an integer k to each input instance and we say that a parameterized problem is *fixed-parameter tractable* (FPT) if there is an algorithm that solves the problem in time $f(k) \cdot |I|^{\mathcal{O}(1)}$, where $|I|$ is the size of the input instance and f is an arbitrary computable function depending only on the parameter k. For more background, the reader is referred to the monographs [7, 8, 25].

Unlike MULTIWAY CUT, the PMWC is already NP-complete for the case when $|T| = 2$. Indeed, consider the following reduction from VERTEX COVER to PMWC. Given an instance $(G = (V, E), k)$ of VERTEX COVER, add two new vertices t_1 and t_2, make them both adjacent to every vertex in V, and set $T_o = \{t_1, t_2\}$ and $T_e = \emptyset$. Call this new graph G'. It is easy to see that G has a vertex cover of size at most k if and only if G' has k-sized vertex subset that intersects every odd T_o-path. In fact, our argument shows that OMWC is NP-complete for the case when $|T| = 2$. One can similarly show that EMWC is NP-complete for the case when $|T| = 2$.

Marx [22] was the first to consider cut problems in the context of parameterized complexity. He gave an algorithm for MULTIWAY CUT with running time $\mathcal{O}(4^{k^3} n^{\mathcal{O}(1)})$ with the current fastest algorithm running in time $\mathcal{O}(2^k n^{\mathcal{O}(1)})$ [5]. Even the recent developments in techniques to solve graph separation problems [23] and parity based graph problems [17], do not seem to apply to PMWC, a natural companion of these problems. In this paper, we introduce a new notion of *generalized important separators*, which along with the tools used to solve parameterized cut problems like MULTIWAY CUT and MULTI-CUT, allows us to design an FPT algorithm for PMWC. In general, this notion seems to allow us to bring a number of problems under a single umbrella, and in particular we demonstrate its application to PMWC. The main result of this paper is the following.

Theorem 1. PMWC *can be solved in time* $2^{2^{\mathcal{O}(k)}} n^{\mathcal{O}(1)}$ *time.*

Our algorithm combines several ideas used in previous parameterized algorithms for graph separation problems including the notion of important separators and randomized selection of important sets to simplify the input instance. Furthermore, we introduce a generalization of important separators, which we believe could be a useful tool for several other separation problems. The algorithm for PMWC has three phases, in the first phase using a well-known technique of iterative compression, we bound the number of even terminals by a linear function of k. In the second phase we remove even terminals using the notion of generalized important separators that we define in this paper and obtain $f(k)$ instances of OMWC. We obtain the generalized important separators by dividing the graph into slices with small boundaries and applying a divide and conquer paradigm over these slices. In the final phase we solve these instances of OMWC by designing an FPT algorithm for OMWC. More precisely we obtain the following result.

Lemma 1. OMWC *can be solved in time* $2^{2^{\mathcal{O}(k)}} n^{\mathcal{O}(1)}$ *time.*

We note that OMWC can be shown to be FPT be a simple reduction to the SUBSET OCT problem which was shown to be FPT in [17]. However, such an algorithm for OMWC would have a much worse dependence on the parameter k when compared to the algorithm we present in this paper. We also point out that in the case of the EMWC problem with two terminals, we may subdivide the edges incident on one of them, thus converting all even paths between these terminals into odd paths and vice versa. This reduction shows that OMWC is equivalent to EMWC in the case of two terminals and hence Lemma 1 immediately gives an FPT algorithm for EMWC in the case of two terminals. We also consider the edge version of PMWC, the EDGE PARITY MULTIWAY CUT (EPMWC) problem, where the input is a graph G, a subset of vertices $T = T_e \cup T_o$, a positive integer k and the objective is to determine whether there exists a k-sized edge subset that intersects every even path from a vertex $v \in T_e$ to $T \setminus \{v\}$ and every odd path from a vertex $v \in T_o$ to $T \setminus \{v\}$. We show that this problem is also FPT by establishing a parameter preserving reduction from EPMWC to PMWC.

Related Work. Parity problems hold a lot of promise and remain hitherto unexplored from the perspective of parameterized complexity, with exceptions that are few and far between. The first parameterized algorithm for ODD CYCLE TRANSVERSAL, finding a k sized vertex subset that intersect all odd cycles only appeared in 2004 [28]. Recently, Kawarabayashi and Reed [13] obtained an almost linear time parameterized algorithm for ODD CYCLE TRANSVERSAL, albeit with a much worse dependence on solution size than in [28]. Kawarabayashi and Reed [14] settled the parameterized complexity of ODD CYCLE PACKING, finding k vertex disjoint odd cycles in a graph, by showing it to be FPT. The Parameterized Complexity of ODD CYCLE PACKING was a long standing open problem and is much more general problem than the famous DISJOINT PATHS problem, finding vertex disjoint paths between given pairs of vertices. Recently, Kawarabayashi, Reed and Wollan [15] initiated the parameterized study of parity versions of graphs minors and gave an algorithm to find odd minors. Other studies include finding odd subdivision, parity paths passing through specific vertices [12, 11]. On the cut side, as we mentioned before, the area was initiated by the paper of Marx [22]. The notions used in this paper has been useful in settling parameterized complexity of variety of problems including DIRECTED FEEDBACK vertex SET [3], ALMOST 2 SAT [27] and ABOVE GUARANTEE VERTEX COVER [27, 26]. Recently, Marx and Razgon [23] and Bousquet, Daligault and Thomassé [1] independently showed that MULTI-CUT, finding k vertices to disconnect given pairs of terminals is FPT. Continuing this line of study, Chitnis, Hajiaghayi and Marx studied MULTIWAY CUT on directed graphs and showed it to be FPT [4].

2 Preliminaries

Given a path P, we refer to the number of edges in P as the length of P and denote it by $|P|$. We call a path an odd (even) path if the length of the path is odd (respectively even). We refer to the parity of $|P|$ as the parity of the path P. If P is a path from some vertex in a set X to some vertex in a set Y, we say that P is an X-Y path. If A contains a single vertex x, we say that P is an x-Y path. Given a graph $G = (V, E)$ and $T \subseteq V$, paths with only the end points in T are called *T-paths* and if their length is odd (even), then *odd (even)* T-paths. Given a vertex set S, we denote by $P \cap S$, the set of vertices in S which intersect P. Given two paths $P_1 = v_1, \ldots, v_l$ and $P_2 = u_1, \ldots, u_r$, such that one end point of P_1 (say v_l) is the same vertex as some endpoint of P_2 (say u_1) and no other vertices of the two paths coincide, we define the concatenated path $P_1 + P_2$ as $v_1, \ldots, v_l, u_2, \ldots, u_r$. Given a walk $W = v_1, \ldots, v_l$ from $t_1 = v_1$ to $t_2 = v_l$, we define a *closed loop* (or *loop*) of W, as a closed walk $W' = v_i, v_{i+1}, \ldots, v_j, v_p, v_{p+1}, \ldots, v_r$ where $v_i = v_r$ and $v_j = v_p$, $v_{i+1}, \ldots, v_{j-1}$ and $v_{p+1}, \ldots, v_{r-1}$ are either vertex disjoint simple paths which are subpaths of W, or are the same exact path which is a subpath of W. By *deleting* a closed loop $W' = v_i, \ldots, v_r$ from W, we obtain the walk $W'' = v_1, \ldots, v_i, v_{r+1}, \ldots, v_l$. Note that deleting an even closed loop does not affect the parity of the walk, while deleting an odd closed loop flips the parity of the walk. Hence, loops of the second type (for eg. v_1, v_2, v_1), where the two internal subpaths coincide exactly can be removed without changing the parity of the walk since such loops are by definition, even. Hence, in our discussions, we will assume that such loops do not occur, that is, if $W' = v_i, v_{i+1}, \ldots, v_j, v_p, v_{p+1}, \ldots, v_r$ is a loop, then it is a cycle. Given a set S_c of clauses and a set S_v of variables of a 2 SAT formula F, we denote by $F \setminus S_c$ the formula obtained from F by deleting the clauses in S_c, we denote by $F \setminus S_v$ the formula obtained from F be deleting the clauses which involve a variable in S_v. We denote by F_{S_c}, the restriction of F to the clauses in S_c. In the ALMOST 2 SAT(VARIABLE) problem, we are given a 2 SAT formula F, and a postive integer k and the problem is to check if there is a k-sized set of variables of F, whose deletion makes F satisfiable.

In an instance $(G, T_e \cup T_o, k)$ of PMWC, the vertices in T_e are called *even* terminals and those in T_o are called *odd* terminals. Vertices in $T_e \setminus T_o$ are called *purely even* terminals and those in $T_o \setminus T_e$ are called *purely odd* terminals.

3 Structural Claims

In this section we first prove some general claims that will be used in several proofs later. Following this, we define the notion of isolated and semi-isolated components which we use to impose a certain useful structure on the input instance.

Lemma 2. *Let $G = (V, E)$ be a graph and let t_1 and t_2 be two vertices in G. Let W be a walk from t_1 to t_2. If there is no odd closed loop in W, then there is a path from t_1 to t_2 which has the same parity as W. If there is an odd closed loop in W, then there is a vertex u in the closed loop which has paths of both parities to t_1 and paths of both parities to t_2.*

Proof. Let $W = v_1, \ldots, v_l$ where $v_1 = t_1$ and $v_l = t_2$. Suppose that every closed loop in W is even. We prove the first statement of the lemma by induction on the number of closed loops in the walk W. In the base case, the number of closed loops is 0. In this case, the lemma is vacuously true. Now, suppose that the number of closed loops in W is some $s > 0$ and assume that the statement of the lemma holds for all walks with less than s closed loops. Consider a closed loop W' in W. Since this is even, deleting it will result in another walk W'' which has the same parity as W, and has less than s closed loops. By the induction hypothesis, there is a path from t_1 to t_2 with the same parity as W''. This proves the first statement of the lemma.

Suppose $W' = v_i, v_{i+1}, \ldots, v_j, v_p, v_{p+1}, \ldots, v_r$ is an odd closed loop of W where $P_1 = v_{i+1}, \ldots, v_{j-1}$ and $P_2 = v_{p+1}, \ldots, v_{r-1}$ are vertex disjoint simple paths which are subpaths of W, and $v_i = v_r$ and $v_j = v_p$. We assume without loss of generality that there is no other closed loop in walk, since we can just delete such loops if they did exist. Since W' is an odd loop, at least one of P_1 or P_2 must have length at least 2. Without loss of generality, we assume that P_1 has length at least 2. Consider the vertex v_{i+1}. We claim that it has paths of both parity to t_1 and t_2. Indeed, consider the path $P_3 = v_{i+1}, v_i, \ldots, v_1$ and the path $P_4 = v_{i+1}, \ldots, v_j, v_{p-1}, \ldots, v_1$. Clearly, they are paths of opposite parities from v_{i+1} to t_1. Similarly, we have paths of both parities to t_2. This completes the proof of the second statement of the lemma.

Lemma 3. *Let $G = (V, E)$ be a graph, and let t_1 and t_2 be two distinct vertices of G. Suppose that every t_1-t_2 path in G has the same parity, and every vertex of G lies on a t_1-t_2 path. Then, any vertex of G cannot have paths of both parity to t_1 and it cannot have paths of both parity to t_2.*

Proof. Let v be a vertex lying on an t_1-t_2 path P. Let P_1 be the subpath of P from t_1 to v and let P_2 be the subpath of P from v to t_2. We prove by double induction on $|P_1|$ and $|P_2|$ that v cannot have paths of both parity to t_1. It is analogous to show that v cannot have paths of both parity to t_2.

In the base case, first, let $|P_2| = 0$. This implies that $v = t_2$ and the statement of the lemma holds. Now, consider the base case for $|P_1|$, that is $|P_1| = 0$. This implies that $v = t_1$ and the statement of the lemma holds. Now, assume that $|P_2|, |P_1| > 0$ and that the claim holds for all smaller values of $|P_2|$ and it also holds for all smaller values of $|P_1|$. Let P_3 be a path from v to t_1 which has parity opposite to that of P_1. If P_3 is disjoint from P_1 and P_2, $P_3 + P_2$ is an odd path from t_1 to t_2, which is a contradiction. Hence, assume that P_3 intersects P_1 or P_2 . Suppose P_3 intersects P_2 first when traversing from v to t_1 and let v' be the first vertex along P_3 which occurs in P_2 when traversing P_3 from v to t_1. Let P_3' be the subpath of P_3 from v to v', let P_3'' be the subpath of P_3 from v' to t_1, let P_2' be the subpath of P_2 from v to v' and let P_2'' be the subpath of P_2 from v' to t_2. If P_3' and P_2' have differing parities, the paths $P_1 + P_3' + P_2''$ and $P_1 + P_2' + P_2''$ are t_1-t_2 paths with differing parities, a contradiction. Hence, P_3' and P_2' must have the same parity. But now, v' has two paths $P_2' + P_1$ and P_3'', to t_1, which have different parities, and the subpath of P_2 from v' to t_2, which is P_2'', has length strictly less than $|P_2|$. But, by induction hypothesis, v' cannot have paths of both parity to t_1, which is a contradiction. Now, suppose that P_3 intersects P_1 first and let v' be the first vertex along P_3 which occurs in P_2 when traversing P_3 from v to t_1. Let P_3' be the subpath of P_3 from v' to v, let P_3'' be the subpath of P_3 from t_1 to v', let P_1' be the subpath of P_1 from v' to v and let P_1'' be the subpath of P_1 from t_1 to v'. If P_3' and P_1' have differing parities, the paths $P_1'' + P_3' + P_2$ and $P_1'' + P_1' + P_2$ are t_1-t_2 paths with different parities, a contradiction. Hence, P_3' and P_1' must have the same parity. But now, v' has two paths P_3'' and $P_1' + P_2$ to t_1, which have different parities and the subpath of P_1 from v' to t_1, which is P_1', has length strictly less than $|P_1|$. But, by induction hypothesis, v' cannot have paths of both parity to t_1, which is a contradiction. This completes the proof of the lemma.

We also prove the following lemma.

Lemma 4. *Let $G = (V, E)$ be a graph and T be a vertex set such that every T-path has even parity and for every $w \in V(G)$ there is a T-path containing w. Then, for every $w \in V$, all $w - T$ paths have the same parity.*

Proof. The lemma follows by applying arguments similar to those in Lemma 3 for every vertex w in G and for every pair t_1, t_2 of terminals in T such that v lies on a t_1-t_2 path in the subgraph.

3.1 Isolated and semi-isolated components

Definition 1. *Consider an instance $(G, T = T_e \cup T_o, k)$ of* PMWC *and let S be a solution to this instance (see Fig. 1). The set of vertices not reachable from T in $G \setminus S$ is called the* isolated *part of S, and the set of vertices reachable from T in $G \setminus S$ is called the* non-isolated *part of S. For any connected component C in the subgraph induced on the non-isolated part, the* semi-isolated *part of C is the set $C' \subset C$ of vertices which do not lie on a T'-path in this subgraph, where $T' = T \cap C$. The semi-isolated part of S is the union of the semi-isolated part of every component in $G \setminus S$. Any connected component in the graph induced on isolated (respectively semi-isolated) part of S is called an isolated (similarly semi-isolated) component of S, with the reference to S dropped if it is clear from the context.*

Observation 2 *Let $(G, T = T_e \cup T_o, k)$ be an instance of* PMWC *and let S be a solution to this instance. Consider any connected component C in the graph induced on the non-isolated part of S, let $T' = T \cap C$ and let C' be the semi-isolated part of C. Then, the graph $G[C \setminus C']$ is connected.*

Proof. Consider a Steiner tree in $G[C]$ connecting the vertices in T'. Since the vertices of T' are connected in $G[C]$, such a tree exists. This tree H will be a subgraph of $G[C \setminus C']$ since every vertex on this tree lies on some $T'-$path. Now, any vertex u in $G[C \setminus C']$ is be reachable from T' (in $G[C \setminus C']$). Since any two vertices of T' are connected in H, $G[C \setminus C']$ is connected as well.

Definition 2. *Let $(G, T = T_e \cup T_o, k)$ be an instance of* PMWC *and let S be a solution to this instance. Consider any connected component C in the non-isolated part of S and let C' be the semi-isolated part of C. Then, the subgraph $G[C \setminus C']$ is called a* main *component of S.*

Observation 3 *Let $(G, T = T_e \cup T_o, k)$ be an instance of* PMWC *and let S be a solution to this instance.*
(a) *Any connected component of $G \setminus S$ with at least two terminals contains terminals from exactly one of $T_o \setminus T_e$ or $T_e \setminus T_o$.*
(b) *Any connected component of $G \setminus S$ contains at most 2 vertices from T_e.*

Proof. **(a)** Let C be a component of $G \setminus S$ containing two terminals t_1 and t_2 such that $t_1 \in T_o$ and $t_2 \in T_e$. Then, there is a path between t_1 and t_2 in $G \setminus S$. If this path is odd, then it contradicts the intersection of S with every t_1-$T \setminus \{t_1\}$ path and if it is even, then it contradicts the intersection of S with every t_2-$T \setminus \{t_2\}$ path.

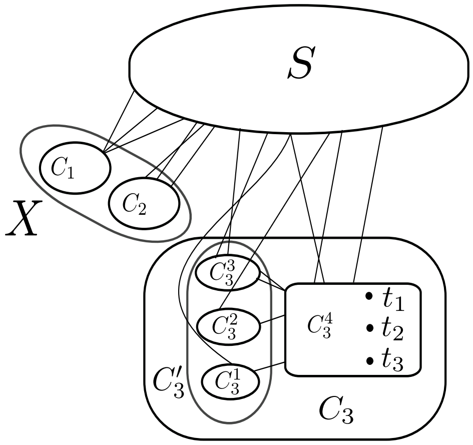

Fig. 1. An illustration of a solution S where X is the isolated part of S, C_3 is a non-isolated component, C'_3 is the semi-isolated part of C_3 and C_3^4 is a main component.

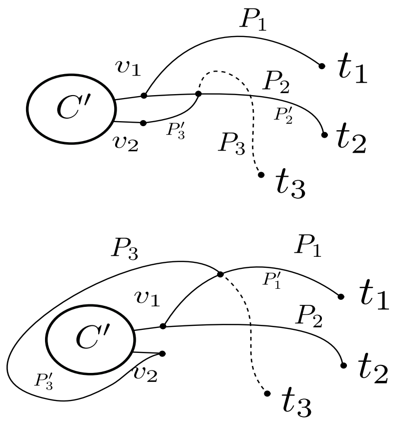

Fig. 2. An illustration of the two subcases of case (b) in Lemma 5.

(b) Let C be a component of $G \setminus S$ containing a set $T' \subseteq T_e$. Consider a steiner tree on the set T' in $G \setminus S$. Since the steiner tree is a connected bipartite graph, if $|T'| > 2$, there must be two vertices of T' which lie in the same partition and hence have an even path between them in this tree. But this is a contradiction since this even path is disjoint from S.

Lemma 5. *Let $(G, T = T_e \cup T_o, k)$ be an instance of* PMWC *and let S be a solution to this instance. Then, any component in the semi-isolated part of S has a single neighbor in the corresponding main component.*

Proof. Consider a semi-isolated component C' and let C be it's corresponding main component. Since the vertices of C' are in the same connected component as those of C in the graph $G \setminus S$, C' has at least one neighbor in C. We will now show that C' cannot have more than one neighbor in C.

Suppose that this is not the case and let v_1 and v_2 be two distinct vertices in C which are adjacent to vertices in C'. We note that in the cases when both v_1 and v_2 are terminals, there is a path between two terminals which intersects the semi-isolated component C', which is not possible by definition. Now, consider the case when exactly one of the two vertices, say v_1 is a terminal. But, v_2 lies on a path between two terminals, say w_1 and w_2 where w_1 or w_2 could be v_1. Consider this path and conside the two subpaths of this path from v_2 to w_1 and from v_2 to w_2. Atleast one of these two subpaths is disjoint from v_1. Hence, subpath, along with the edges from C' to v_1 and v_2 results in a T-path which intersects a semi-isolated component, a contradiction.

Hence, we consider the remaining case where v_1,$v_2 \notin T$. Let P be a T-path from t_1 to t_2 which contains v_1. We know that such a path exists since v_1 lies in a main component. Let P_1 be the subpath of P from t_1 to v_1 and let P_2 be the subpath of P from v_1 to t_2. Let P_3 be a path from v_2 to $t_3 \in T$ (t_3 can be the same as t_1 or t_2). We know that such a path exists since v_2 is in a main component. We now consider the following two cases.

(a) P_3 does not intersect P_2 or P_1. Then clearly, there are paths from t_1 to t_3 and t_2 to t_3 which intersect the semi-isolated component C', and are disjoint from the solution, which is a contradiction since no vertex in C' can lie on a T-path disjoint from the solution.

(b) P_3 intersects P_2 or P_1. Suppose P_3 intersects P_2 first (see Fig. 2) when traversing from v_2 to t_3 and let the vertex at which this intersection occurs be u. Let P_3' be the subpath of P_3 from v_2 to u and let P_2' be the subpath of P_2 from u to t_2. Additionally, let P be a path from v_1 to v_2 such that the internal vertices of P lie in C'. Since C' is a connected component, we know that such a path exists. But now, $P_1 + P + P_3' + P_2'$ is a t_1-t_2 path disjoint from S and intersecting C'. This is a contradiction since no vertex in C' can lie on a T-path in $G \setminus S$.
Now, suppose that P_3 intersects P_1 first at the vertex u. Let P_3' be the subpath of P_3 from v_2 to u, let P_1' be the subpath of P_1 from u to t_1 and let P be a path from v_1 to v_2 such that the internal vertices of P lie in C'. But now, $P_2 + P + P_3' + P_1'$ is a t_2-t_1 path disjoint from S and intersecting C'. This is a contradiction since no vertex in C' can lie on a T-path in $G \setminus S$.

This concludes the proof of the lemma.

Definition 3. *Let C' be a semi-isolated component of S. We refer to the neighbor of C' in the corresponding main component as the* pivot *of C' and we denote it by $\chi(C')$.*

4 PMWC **parameterized by the solution size**

The algorithm for PMWC has three phases, in the first phase using the well-known technique of iterative compression, we bound the number of even terminals by $7k$. In the second phase we remove even terminals using the notion of generalized important separators that we define in this section and obtain $f(k)$ instances of OMWC. In this section we outline the first two phases of the algorithm and in the next section we give the details of the final phase – an FPT algorithm for OMWC.

4.1 Bounding the number of even terminals

We now describe a way to separate and remove the even terminals from the instance. We will first describe a way to reduce the given instance of PMWC to multiple (but a bounded number of) instances, each with a bounded number of even terminals, such that solving these instances will lead to a solution for the input instance. To this end we will use the technique of iterative compression. In this technique, we assume that a solution of size $k + 1$ is part of the input, and attempt to compress it to a solution of size k. The method adopted usually is to begin with a subgraph that trivially admits a $(k + 1)$-sized solution and then expand it iteratively.

Given an instance $(G = (V, E), T = T_e \cup T_o, k)$ of PMWC, where $V = \{v_1, \ldots, v_n\}$, we define a graph $G_i = G[V_i]$ where $V_i = \{v_1, \ldots, v_i\}$. We iterate through the instances $(G_i, T_i = (T_e \cap V_i) \cup (T_o \cap V_i), k)$ starting from $i = k + 1$ and for the i^{th} instance, with the help of a *known* solution S_i of size at most $k + 1$ we try to find a solution $\hat{S}_i$ of size at most k. Formally, the *compression* problem we address is following.

PMWC COMPRESSION

Instance: $(G = (V, E), T = T_e \cup T_o, k, S)$ where G is an undirected graph, T_e, T_o are vertex sets, k a postive integer and S, a PMWC of size at most $k + 1$.
Parameter: k
Question: Does there exist a PMWC of size at most k for this instance?

We will reduce the PMWC problem to $n - k$ instances of the PMWC COMPRESSION problem as follows. Let $I_i = (G_i, (T_e \cap V_i) \cup (T_o \cap V_i), S_i, k)$ be the i^{th} instance of PMWC COMPRESSION. Clearly, the set V_{k+1} is a solution of size at most $k + 1$ for the instance I_{k+1}. It is also easy to see that if $\hat{S}_{i-1}$ is a solution of size at most k for instance I_{i-1}, then the set $\hat{S}_{i-1} \cup \{v_i\}$ is a solution of size at most $k + 1$ for the instance I_i. We use these two observations to start off the iteration with the instance $(G_{k+1}, (T_e \cap V_{k+1}) \cup (T_o \cap V_{k+1}), k, S_{k+1} = V_{k+1})$ and try to compute a solution of size at most k for this instance. If there is such a solution $\hat{S}_{k+1}$, we set $S_{k+2} = \hat{S}_{k+1} \cup \{v_{k+1}\}$ and try to compute a solution of size at most k for the instance I_{k+2} and so on. If, during any iteration, the corresponding instance does not have a solution of the required size, it implies that the original instance is also a NO instance. Finally the solution for the original input instance will be $\hat{S}_n$. Since there can be at most n iterations, the total time taken is bounded by n times the time required to solve the PMWC COMPRESSION problem.

We will now describe a way to bound the number of even terminals in an instance of PMWC COMPRESSION. Let $(G, T = T_e \cup T_o, k, S)$ be an instance of PMWC COMPRESSION. Fix a hypothetical solution $\hat{S}$ for this instance. We first guess the set $Y = S \cap \hat{S}$. There are 2^{k+1} such possibilities. For each guess of Y, we delete it from the instance, and also delete vertices which are no longer relevant for the instance. This results in an instance of PMWC which has a solution $N = S \setminus Y$ and we are required to find a solution of size at most $k - |Y|$ which is disjoint from N. Now, we show that if the resulting instance has such a solution, then it must be the case that the number of even terminals in this instance is bounded.

Suppose that the resulting instance indeed has such a solution. Fix such a solution S'. We call a component of $G \setminus N$ *affected* if it contains some vertex of S' and *unaffected* otherwise. Clearly, there can be at most k affected components. Now, consider the unaffected components which contain even terminals. We claim that the number of such components cannot be more that $2k$. Suppose this was not the case. Then there must exist three unaffected components which contain even terminals and share a neighbor in N. But this implies that there will be an even path between atleast two of these terminals which is disjoint from the new solution S', a contradiction. Hence, the number of components of N which contain even terminals is at most $3k$. By Observation 3, any component can contain at most 2 even terminals. Since N contains at most k even terminals, the number of even terminals in the instance is bounded by $7k$. Hence, if the number of terminals in the instance after removing the guess Y exceeds $7k$, we can reject this guess right away. Note that, even if we compute a solution of the required size which is not disjoint from the set N, we can use it to continue the iteration. Hence, once we have an instance with a bounded number of even terminals, we ignore the fact that there is a solution disjoint from N and just compute any solution of the required size for the corresponding PMWC instance. Since we only need to deal with PMWC instances arising from instances of PMWC COMPRESSION, henceforth we will assume that the given instance of PMWC contains at most $7k$ even terminals.

4.2 Removing even terminals

We initially perform the following preprocessing step on the given instance $(G, T = T_e \cup T_o, k)$ of PMWC. For every purely odd terminal $t_i \in T \setminus T_e$, we add $2(k+1)$ new vertices $T_i = \{t_i^1, \ldots, t_i^{k+1}\}$ and $\hat{T}_i = \{\hat{t}_i^1, \ldots, \hat{t}_i^{k+1}\}$ and make t_i adjacent to every vertex in T_i. Finally, we add all possible edges between the sets T_i and $\hat{T}_i$. We now define a new set of purely odd terminals $T'_o = \bigcup_{t_i \in T \setminus T_e} \hat{T}_i$. That is, for every t_i in $T \setminus T_e$, we replace t_i with the $k+1$ vertices $\hat{t}_i^j$ in the set of purely odd terminals. We will now show that the resulting instance is indeed equivalent to the input instance.

Lemma 6. *Given an instance $(G, T = T_e \cup T_o, k)$ of* PMWC*, let $(G', T' = T_e \cup T'_o, k)$ be the instance obtained as a result of the terminal transformation described above (see Fig. 3). Then, (G, T, k) is a* YES *instance if and only if (G', T', k) is a* YES *instance.*

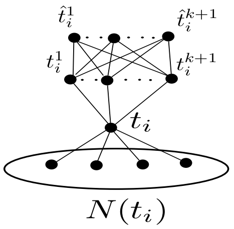

Fig. 3. Illustration of the terminal transformation.

Proof. Suppose that S is a solution for the instance $(G, T = T_e \cup T_o, k)$. We claim that S is also a solution for the instance $(G', T' = T_e \cup T'_o, k)$. Any T'-path P in $G' \setminus S$ can be converted to a T-path of the same parity and between the corresponding terminals in $G \setminus S$ by removing (if necessary) the first two and/or the last two edges of P. Hence, a path of forbidden parity in $G' \setminus S$ implies a path of forbidden parity in $G \setminus S$, a contradiction.

Conversely, suppose that S' is a solution for the instance (G', T', k). Observe that, due to our construction, we can assume without loss of generality that the solution S'

comprises only vertices of G. We now claim that S' is also a solution for the instance (G, T, k). Suppose this is not the case, and let P be a t_1-t_2 path of forbidden parity in $G \setminus S'$. If this path was between two even terminals, then this path would exist in G' as well, and hence would intersect S'. Hence, one of the end points of this path must be a purely odd terminal t_i. But in this case, we also have a path of the same parity from $\hat{t}_i^j$ to t_2 in the graph $G' \setminus S'$ for every j. Similarly, if necessary, we can extend the other end point by two edges to obtain a path of forbidden parity in $G' \setminus S'$, which is a contradiction. This completes the proof of the lemma.

Due to Lemma 6, henceforth, we will assume that the given input instance is already of the form described above. This also allows us to assume that the solution will be disjoint from the set of purely odd terminals. We will now describe a procedure to reduce this instance of PMWC to an instance with no even terminals, thereby resulting in an instance of OMWC.

We fix a hypothetical solution for the PMWC instance and work with this solution. We first guess the intersection of the hypothetical solution with the set of even terminals and delete these vertices from the graph. Let S be the subset of the solution left after this step, that is S is a solution for the remaining instance. We then guess the way S partitions the even terminals into different connected components in the graph $G \setminus S$. There are at most $2^{|T_e|}$ possible intersections and $|T_e|^{|T_e|} = 2^{\mathcal{O}(k \log k)}$ partitions for the even terminals. Hence, the total number of possible guesses is $2^{\mathcal{O}(k \log k)}$. We say that S *conflicts* with a partition if there is a component of $G \setminus S$ containing terminals from two distinct sets of the partition. For each guess of the partition, we attempt to find a solution which partitions the even terminals in a way which does not conflict with the guess. We fix one such guess of the partition, say $\mathcal{P}$ and work with this partition for the rest of the section. In addition, note that, we can now assume that the solution S is disjoint from the entire set of terminals. This is because we have already guessed (and deleted) the intersection with even terminals, and it is already disjoint from the purely odd terminals.

4.3 Important separators

The notion of important separators was formally introduced in [22] to handle the MULTIWAY CUT problem and the same concept was used implicitly in [2] to give an improved algorithm for the same problem. In this subsection, we recall some definitions related to important separators and a few lemmas which will be required for our algorithm.

Definition 4. *Let $G = (V, E)$ be an undirected graph, let $X, S \subseteq V$ be vertex subsets. We denote by $R_G(X, S)$ the set of vertices of G reachable from X in the graph $G \setminus S$ and we denote by $NR_G(X, S)$ the set of vertices of $G \setminus S$ which are not reachable from X in the graph $G \setminus S$. We drop the subscript G if it is clear from the context.*

Definition 5. *Let $G = (V, E)$ be an undirected graph and let $X, Y \subset V$ be two disjoint vertex sets. A subset $S \subseteq V \setminus (X \cup Y)$ is called an X-Y separator in G if $R_G(X, S) \cap Y = \emptyset$ or in other words there is no path from X to Y in the graph $G \setminus S$. We denote by $\lambda_G(X, Y)$ the size of the smallest X-Y separator in G. An X-Y separator S_1 is said* *to **cover** an X-Y separator S with respect to X if $R(X, S_1) \supset R(X, S)$ and S_1 is said to **dominate** S if it covers S and $|S_1| \leq |S|$. If the set X is clear from the context, we just say that S_1 dominates S. An X-Y separator is said to be inclusion wise minimal if none of its proper subsets is an X-Y separator.*

Definition 6. *Two X-Y separators S and S_1 are said to be **incomparable** if neither covers the other.*

Observation 4 *Let S_1 and S_2 be two incomparable X-Y separators. Then, $R(X, S_1) \cap S_2 \neq \emptyset$ and $R(X, S_2) \cap S_1 \neq \emptyset$. That is, there is a vertex of S_1 reachable from X in the graph $G \setminus S_2$ and a vertex of S_2 reachable from X in the graph $G \setminus S_1$. Also, $NR(X, S_1) \cap S_2 \neq \emptyset$ and $NR(X, S_2) \cap S_1 \neq \emptyset$. That is, there is a vertex of S_1 separated from X in the graph $G \setminus S_2$ and a vertex of S_2 separated from X in the graph $G \setminus S_1$.*

Definition 7. *Let $G = (V, E)$ be an undirected graph, $X, Y \subset V$ be vertex sets and $S \subseteq V$ be an X-Y separator in G. We say that S is an **important** X-Y separator if it is inclusionwise minimal and there does not exist another X-Y separator S_1 such that S_1 dominates S with respect to X.*

Lemma 7. ([22]) *Let $G = (V, E)$ be an undirected graph, $X, Y \subset V$ be disjoint vertex sets. There exists a unique important X-Y separator S^* of size $\lambda_G(X, Y)$ and it can be computed in polynomial time.*

Lemma 8. ([2]) *The number of important X-Y separators of size at most k is at most 4^k and these can be enumerated in time $\mathcal{O}(4^k n^{\mathcal{O}(1)})$.*

4.4 Important separator sequences and a generalization of important separators

In this subsection we will define the notion of an important separator sequence and use it in the context of PMWC to define generalized important separators with the properties we require.

Definition 8. *Let $G = (V, E)$ be a graph and let $X, Y \subseteq V$ be disjoint vertex sets. We define an important X-Y separator of order i, S^i to be the unique smallest important X-S^{i-1} separator in G, where $S^0 = Y$.*

By Lemma 7, for every i, an important X-Y separator of order i is unique and can be computed in polynomial time.

Definition 9. *We define a smallest X-Y separator sequence $\mathcal{I}$ to be a set $\mathcal{I} = \{S^i | 1 \leq i \leq l\}$, where S^i is an important X-Y separator of order i, for every $1 \leq i, j \leq l$, $|S^i| = |S^j|$, and $\lambda(X, S^l) > \lambda(X, Y)$, that is there is no X-S^l cut of size $|S^l|$ (see Fig. 4).*

Observation 5 *Given two X-Y separators S_1 and S_2, we say that $S_1 \preceq S_2$ if S_2 covers S_1 with respect to X. Then, $(\mathcal{I}, \preceq)$ forms a total order where $\mathcal{I}$ is a smallest X-Y separator sequence.*

Note that a smallest X-Y separator sequence is unique and can be computed in polynomial time. Observation 5 is the reason we refer to the set $\mathcal{I}$ as a *sequence*.

Lemma 9. *Let P_1 and P_2 be two separators in $\mathcal{I}$ such that $P_1 \preceq P_2$ and there is no P_3 in $\mathcal{I}$ such that $P_1 \preceq P_3 \preceq P_2$. Then, the size of a minimum X-Y separator which lies in the set $NR(X, P_1) \cap R(X, P_2)$ is at least $|P_1| + 1$.*

Proof. Suppose that this is not the case and let S be an X-Y separator of size $|P_1|$ which lies in the set $NR(X, P_1) \cap R(X, P_2)$. By the statement of the lemma, $S \notin \mathcal{I}$. Let i be such that $P_2 = S^{i-1}$ and $P_1 = S^i$, that is, P_1 is the unique smallest important X-P_2 separator. But, since $P_1 \preceq S \preceq P_2$, and S lies in the set $NR(X, P_1) \cap R(X, P_2)$ it contradicts the fact that P_1 is an important X-P_2 separator.

The key consequence of the definition of the smallest separator sequence is that it defines a natural partition of the graph into slices with *small boundaries*. Using this, we may restrict our search to local parts of the graph, in which case finding separators with certain properties becomes easier. We will now describe how this concept is applied in the context the PMWC problem.

Definition 10. *Given sets X and Y and a minimal X-Y separator S, let l be the size of a minimum* EMWC *of the set X in the graph $G \setminus S$. We say that a minimal X-Y separator S' **well dominates** S (with respect to X) if S' dominates S with respect to X and the size of a minimum* EMWC *of X in the graph $G \setminus S'$ is at most l.*

Note that any X-Y separator well dominates itself. Now, let T_1 be any set in the partition $\mathcal{P}$, and let $\hat{S}$ be a minimal part of S separating T_1 from $T \setminus T_1$. Recall that T_1 is a set of even terminals and by Observation 3, we can assume that T_1 contains at most 2 even terminals. We will first show that for any separator which well dominates $\hat{S}$, there is a solution for the PMWC instance containing this separator. Following that, we will describe an algorithm to compute a T_1-$T \setminus T_1$ separator that well dominates $\hat{S}$.

Lemma 10. *Let $(G, T = T_e \cup T_o, k)$ be an instance of* PMWC*, let S be a solution for this instance, and T_1 be a set in $\mathcal{P}$, with $T_2 = T \setminus T_1$. Let $\hat{S}$ be a minimal part of S separating T_1 and T_2. Let $\hat{S}_1$ be a T_1-T_2 separator which well dominates $\hat{S}$. Then, there is also a solution for the instance which contains $\hat{S}_1$.*

Proof. Let $\hat{K}$ be a minimum EMWC of T_1 in the graph $G \setminus \hat{S}$ and let $\hat{K}_1$ be a minimum EMWC of T_1 in the graph $G \setminus \hat{S}_1$. We know that $|\hat{S}| \geq |\hat{S}_1|$ and $|\hat{K}| \geq |\hat{K}_1|$. Now, consider the set $S' = (S \setminus (\hat{S} \cup \hat{K})) \cup (\hat{S}_1 \cup \hat{K}_1)$. We claim that S' is also a solution for the given instance. It is clear that the size of S' is at most that of S. Hence, it remains to show that S' is indeed a PMWC for the given instance.

Suppose that this is not the case and let t_i and t_j be two terminals such that there is a path P of forbidden parity between t_i and t_j in the graph $G \setminus S'$. Then, there must be a vertex $v \in \hat{S} \cup \hat{K}$ such that the path P intersects v. Since $\hat{S}_1$ dominates $\hat{S}$, this vertex must be reachable from T_1 in the graph $G \setminus \hat{S}$. But, $\hat{S}_1$ is a T_1-T_2 separator. Hence, if t_i or t_j is in T_2, then the path P must intersect $\hat{S}_1$ and hence intersects S', a contradiction. Therefore, it must be the case that t_i and t_j are precisely the vertices in T_1. Now, since this path P lies entirely inside the component containing T_1 in the graph $G \setminus \hat{S}_1$, it must

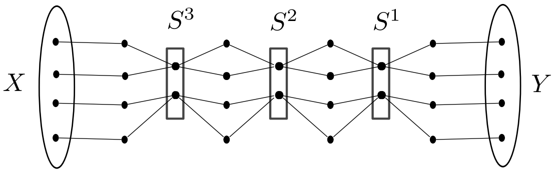

Fig. 4. Illustration of a smallest separator sequence

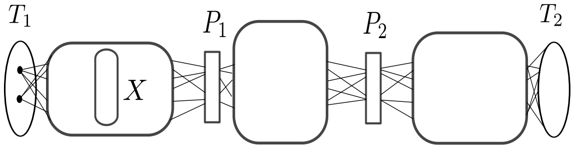

Fig. 5. Illustration of case 1 of the branching.

be the case that this path intersects $\hat{K}_1$ and this in turn implies that the path P intersects S', a contradiction. This completes the proof of the lemma.

Lemma 11. *Let $(G, T = T_e \cup T_o, k)$ be an instance of* PMWC *with a solution S, $\mathcal{P}$ be a partition of T_e such that S does not conflict with $\mathcal{P}$. Let T_1 be a set in $\mathcal{P}$ and T_2 be the set $T \setminus T_1$. Let X be a minimal part of S separating T_1 from T_2. Then, there is an algorithm which runs in time $2^{2^{\mathcal{O}(k)}} n^{\mathcal{O}(1)}$ and returns a set of at most $2^{\mathcal{O}(k^3)}$ T_1-T_2 separators of which at least one separator well dominates X (X is also called the target separator for this instance).*

Proof. For a given subset of vertices, the algorithm computes (if there is one) a T_1-T_2 separator of size at most k, which is contained in the given subset, and well dominates X. Initially, and also when the subset is not explicitly given, we allow this subset to be the entire vertex set of the current graph, and as we prune our search, we will define the subset accordingly. We first fix a hypothetical minimum EMWC of T_1 in the graph $G \setminus X$, say K and guess the size of this set, say l.

Description of algorithm. We first check if there is a T_1-T_2 separator of size at most k within the given subset Z. If not, we return NO. If there is no path from T_1 to T_2, then we return $\emptyset$. Furthermore, if $k \leq 10$, then we find a T_1-T_2 separator well dominating X by enumerating all T_1-T_2 separators of size at most k which lie inside Z. Otherwise, we compute the smallest T_1-T_2 separator sequence, $\mathcal{I}$ comprising only of the vertices of Z. We call a T_1-T_2 separator S' *good* if the size of the minimum EMWC of T_1 in the graph $G \setminus S'$ is at most l and we call it *bad* otherwise. The following observation plays a crucial role in allowing us to ignore (potentially) large parts of the graph during our search.

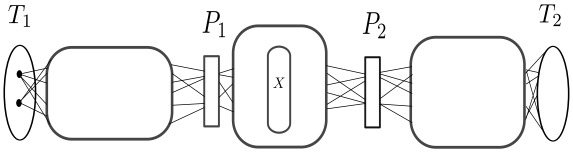

Fig. 6. Illustration of case 2 of the branching.

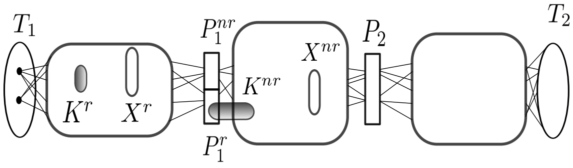

Fig. 7. Illustration of case 3 of the branching.

Observation 6 *If a T_1-T_2 separator is* good*, all T_1-T_2 separators covered by this separator are also* good *and if a T_1-T_2 separator is* bad*, all T_1-T_2 separators which cover this separator are* bad*.*

For each T_1-T_2 separator in $\mathcal{I}$, we now determine whether the separator is good or bad. Since $|T_1| \leq 2$, by Lemma 1, this step takes $2^{2^{O(k)}} n^{\mathcal{O}(1)}$ time. Let P_1 be the maximal element of $\mathcal{I}$ which is good and let P_2 be the minimal element of $\mathcal{I}$, which is bad. That is, P_1 is good and every separator in $\mathcal{I} \setminus \{P_1\}$ which covers P_1 is bad, P_2 is bad and every separator in $\mathcal{I} \setminus \{P_2\}$ covered by P_2 is good. If all the separators in $\mathcal{I}$ are good, then P_2 is defined as T_2 and if all separators in $\mathcal{I}$ are bad, then P_1 is defined as T_1. We will create a number of sub-instances, recurse on each of these instances and finally return the *union* of the sets returned by these recursive calls. The sub-instances are created by exhaustive branching according to the following case analysis on the "relative position" of the target separator with P_1 and P_2.

1. P_1 covers the target separator X (see Fig. 5) or $P_1 = X$. In this case, P_1 itself is a separator which well dominates the target separator and we have indeed found a separator of the required kind. Hence, we return P_1.

2. The target separator X covers P_1, but is itself covered by P_2 (see Fig. 6). Let $\tilde{S}_1$ be the intersection of X with P_1 and $\tilde{S}_2$ be the intersection of X with P_2. We first guess the set $\tilde{S}_1$. If this set is non empty, then we delete it from the graph G, and recursively compute a T_1-T_2 separator of size at most $k - |\tilde{S}_1|$ in the graph $G \setminus \tilde{S}_1$, which lies in the set $NR_G(T_1, P_1)$, and well dominates $X \setminus \tilde{S}_1$ in the graph $G \setminus \tilde{S}_1$. If the set $\tilde{S}_1$ is empty and $P_2 \neq T_2$, then we guess the set $\tilde{S}_2$. If this set is non empty, then we delete it from the graph and recursively compute a set containing a T_1-T_2 separator of size at most $k - |\tilde{S}_2|$ in the graph $G \setminus \tilde{S}_2$, which lies in the set $NR_G(T_1, P_1) \cap R_G(T_1, P_2)$, which also well dominates $X \setminus \tilde{S}_2$ in the graph $G \setminus \tilde{S}_2$. Finally, if the set $\tilde{S}_2$ is also empty, then we recursively compute a set containing a T_1-T_2 separator of size at most k which is contained in the set $NR_G(T_1, P_1) \cap R_G(T_1, P_2)$ and well dominates X in the graph G, and return this set.

3. The target separator X is incomparable with P_1 (see Fig. 7). Let $\tilde{S}_1$ be the intersection of X with P_1, P_1^r be the intersection of P_1 with $R_G(T_1, X)$, P_1^{nr} be the rest of P_1. Also, let X^r be the intersection of X with $R_G(T_1, P_1)$ and let X^{nr} be the rest of X. Since X is incomparable with P_1, by Observation 4, P_1^r, X^r, P_1^{nr} and X^{nr} are non empty. We first guess the set $\tilde{S}_1$. If it is non empty, then we delete it from the graph and recursively compute a set containing a T_1-T_2 separator of size at most $k - |\tilde{S}_1|$ in the graph $G \setminus \tilde{S}_1$ which well dominates $X \setminus \tilde{S}_1$. If it is empty, then we guess the sets P_1^r and P_1^{nr} and also the sizes of the sets X^r and X^{nr}.

We now construct a graph G' as follows. Initially, we set G' as the subgraph of G induced on the set $R_G(T_1, P_1)$. For every vertex in P_1^r, we guess if it is in the set K, in which case, we delete it from the graph G'. From the remaining vertices of P_1^r, for every pair of vertices, we guess if there is an odd (respectively even) path between them in the graph $G \setminus S$, with the internal vertices disjoint from the vertices of G' and add an edge (respectively subdivided edge) between these vertices. We note that it is possible to add both an edge and a subdivided edge between a pair of vertices. This completes the construction of G'. Now, we recursively compute a set containing a T_1-P_1^{nr} separator in the graph G', which well dominates X^r in this graph. Once we compute this set, for each separator X' in the set, we delete it from the graph G and in the resulting graph, recursively compute a set containing a T_1-T_2 separator which lies in the set $NR_G(T_1, P_1)$ and well dominates X^{nr} in $G \setminus X'$. Finally, we construct a new set by pairing up each separator from the first set, with the corresponding separators in the second set, and return this new set.

4. The target separator is incomparable with P_2. This case is analogous to case 3.

We note that the target separator is distinct from P_2 and cannot cover P_2, due to Observation 6 and hence this case need not be taken into consideration.

Correctness. For each instance I for which the algorithm is called, we define a measure $\mu(I) = 2k - \lambda$ where k is an upper bound on the size of the separator we are searching for in I, and λ is the size of the smallest such separator. We prove the correctness of the algorithm by induction on the measure $\mu(I)$.

In the base case, if $\lambda > k$, then algorithm returns NO, which is clearly correct. Additionally, if $\lambda = 0$, then the algorithm simply returns $\emptyset$ as the separator, which is also correct. Finally, in the case when $k \leq 10$, the required separator is found by a brute force search and hence is correct as well. We now assume that the algorithm is correct on all instances with measure less than μ. Now, consider an instance I such that $\mu(I) = \mu$.

1. If case 1 is true, then the algorithm is clearly correct in returning P_1.

2. Suppose case 2 is true. Consider the instance I' obtained by deleting non empty $\tilde{S}_1$. We now search for a T_1-T_2 separator of size at most $k - |\tilde{S}_1|$ in the graph $G \setminus \tilde{S}_1$. Since P_1 was a minimum T_1-T_2 cut, by Menger's theorem, the graph $G \setminus \tilde{S}_1$ will have $|P_1| - |\tilde{S}_1|$ vertex disjoint T_1-T_2 paths and hence the size of the smallest T_1-T_2 separator in the graph $G \setminus \tilde{S}_1$ is $|P_1| - |\tilde{S}_1|$. Hence, $\mu(I') = \mu - |\tilde{S}_1|$. Since $\tilde{S}_1$ is non empty, by induction hypothesis, the algorithm is correct on the instance I' and hence indeed

computes a T_1-T_2 separator which well dominates $X \setminus \tilde{S}_1$. Hence, adding the set $\tilde{S}_1$ to this separator gives us a T_1-T_2 separator of size at most k which well dominates X. The case when $\tilde{S}_2$ is non empty is analgous.
Now, consider the case when both $\tilde{S}_1$ and $\tilde{S}_2$ are empty. By Lemma 9, any T_1-T_2 separator which lies in the set $NR(T_1, P_1) \cap R(T_1, P_2)$ must have size at least $|P_1| + 1$. Hence, $\mu(I') \leq \mu - 1$. By induction hypothesis, the algorithm returns a T_1-T_2 separator of size at most k which well dominates X, which proves the correctness of this case.
3. Suppose case 3 is true. The case when a non empty $\tilde{S}_1$ is deleted from the graph is correct by the same arguments as that seen in the previous case. Now, consider the case when $\tilde{S}_1$ is empty. Consider the graph G' constructed as described in the algorithm and assume that all the guesses made while constructing it were correct. Also, let K^r be the intersection of the set K with $R(T_1, P_1)$ and K^{nr} be the rest of K.

Lemma 12. ***(a)*** *The set X^r is a T_1-P_1^{nr} separator in the graph G' such that K^r is an* EMWC *of T_1 in the graph $G'_1 \setminus X^r$.*
(b) *Let X' be a T_1-P_1^{nr} separator which well dominates X^r in the graph G'. Then, the set $X' \cup X^{nr}$ well dominates X in the graph G.*

Proof. **(a)** Suppose that the set K^r is not an EMWC of T_1 in the graph $G' \setminus X^r$ and let P be an even T_1-path disjoint from $X^r \cup K^r$. Replace any edge (respectively subdivided edge) in P which was not present in G, with the corresponding path of the same parity in $G \setminus S$. It must be the case that the resulting walk, say W, is an even T_1-walk in the graph $G \setminus S$. If W was a path, then it would contradict our assumption that S was a solution for the given instance of PMWC. Hence, it must be the case that W contains some vertex twice. Observe that any vertex in W which occurs more that once must be a vertex in the set $R(T_1, X) \setminus R(T_1, P_1)$. We now claim that any vertex $u \in P_1$ which lies on W, must also lie in the main component of S containing T_1.

If this was not the case and u was in a semi-isolated component, then, W must contain the pivot of u twice since it is a T_1-walk disjoint from S. But, since u is reachable from T_1 in the graph $G \setminus S$, the pivot of u must lie in the set $R(T_1, P_1)$, in which case it cannot appear twice in W, a contradiction. Hence, every vertex of P_1 which lies on W also lies in the main component containing T_1. Now, we also know that any path between two vertices in the main component such that it is disjoint from S, also lies in the main component (see proof of Lemma 5). Hence, the even walk W lies entirely inside the main component containing T_1, in the graph $G \setminus S$. But in this case, we know that there cannot be an odd closed loop in W (see Lemma 2 and Lemma 3), and hence there exists an even T_1-path disjoint from S, which is a contradiction.
(b) Let K' be a minimum EMWC of T_1 in the graph $G' \setminus X'$. Since X' well dominates X^r in G', $|K'| \leq |K^r|$. Hence, $|X' \cup X^{nr}| \leq |X|$ and $|K' \cup K^{nr}| \leq |K|$ and $X' \cup X^{nr}$ is a T_1-T_2 separator since it separates T_1 from P_1^{nr} and T_2 from P_1^r. Therefore, it is sufficient to show that in the graph $G \setminus (X' \cup X^{nr})$, the set $K' \cup K^{nr}$ is indeed an EMWC of T_1. Suppose that this is not the case and consider an even T_1-path P in the graph $G \setminus (X' \cup X^{nr})$. By our construction of G', there must exist an even T_1-path P' in the graph $G' \setminus X'$. But this contradicts the assumption that K' is an EMWC of T_1 in $G' \setminus X'$. This completes the proof of the lemma.

Now, let X' be a T_1-P_1^{nr} separator which well dominates X^r in the graph G'. Then, due to Lemma 12, $X' \cup X^{nr}$ is a T_1-T_2 separator which well dominates X in the graph G_1. Hence, any T_1-T_2 separator which well dominates $X' \cup X^{nr}$ in the graph G also well dominates X. Therefore, we may assume that the target separator for the current instance I is in fact $X' \cup X^{nr}$. Let X'' be a P_1^r-T_2 separator in the graph $G \setminus X'$ which well dominates X^{nr} and let K'' be a minimum EMWC of T_1 in the graph $G \setminus (X' \cup X'')$. Since $K' \cup K^{nr}$ is an EMWC of T_1 in the graph $G \setminus (X' \cup X^{nr})$, it must be the case that $|K''| \leq |K' \cup K^{nr}|$. Therefore, the set $X' \cup X''$ well dominates $X' \cup X^{nr}$ and hence well dominates X in the graph G.

We first search for a T_1-P_1^{nr} separator of size at most $|X^r|$ in the graph G'. By Menger's theorem, we know that there are P_1^{nr} vertex disjoint paths from T_1 to P_1^{nr}, which is also the size of the smallest T_1-P_1^{nr} separator. Now, $\mu(I) = 2(|X^r|+|X^{nr}|)-(|P_1^{nr}|+|P_1^r|)$ and $\mu(I') = 2|X^r|-|P_1^{nr}|$, which implies that $\mu(I') = \mu(I)-(2|X^{nr}|-|P_1^r|)$. Since $|X^{nr}| \geq |P_1^r|$, $\mu(I) - \mu(I') \geq |X^{nr}|$. Since X^{nr} is non empty, by induction hypothesis, the algorithm is correct on I' and returns a set X' well dominating X^r. Now, consider the instance I'' which is a result of deleting X^r from G. Now, $\mu(I'') = 2|X^{nr}| - |P_1^r|$, which implies that $\mu(I) - \mu(I'') \geq |X^r|$. By induction hypothesis, the algorithm returns a set X'' well dominating X^{nr} in the graph $G \setminus X'$ and hence the set $X' \cup X''$ indeed well dominates the set X in the graph G. This proves the correctness of this case and case 4 as well. We also make the following observation, which will be used in bounding the running time of the algorithm. For the instances I' and I'' described as above, $\mu = \mu(I') + \mu(I'')$. This completes the proof of correctness of the algorithm.

Running time. We will show by induction on $\mu(I)$ that the number of separators returned by an execution of the algorithm on instance I, $N(\mu(I))$, is bounded by $2^{\mu(I)^3}$. In each of the base cases of the algorithm, we either return a single separator of the required kind or say NO. Since there are at most k choices for l, the number of separators returned in the base case is at most k, and the claim clearly holds. We now assume that the claimed bound is true for all instances with $\mu(I) < \mu$. Now, consider an instance I such that $\mu(I) = \mu$.

1. The number of separators returned due to case 1 is one for every choice of l, and hence at most k.

2. Consider case 2. There are at most k choices for l, 4^k choices for the sets $\tilde{S}_1$ and $\tilde{S}_2$ and for each choice, we return at most $N(\mu - 1)$ separators. Hence, by induction hypothesis, the number of separators returned due to case 2 is bounded by $k \cdot 4^k \cdot 2^{(\mu-1)^3}$.

3. Consider case 3. There are at most k choices for l, 2^k choices for the set P_1^r, at most k choices for $|X^{nr}|$, at most 2^k choices for the set $P_1^r \cap S$, at most 4^{k^2} possible graphs we construct and for each of these choices, we return at most $N(\mu_1) \cdot N(\mu_2)$ separators where $\mu_1+\mu_2 = \mu$, which, by the induction hypothesis is at most $2^{(\mu-1)^3} \cdot 2^1$. Hence, the number of separators returned due to case 3 is bounded by $k \cdot 2^k \cdot k \cdot 2^k \cdot 4^{k^2} \cdot 2^{(\mu-1)^3} \cdot 2^1$.

4. Similarly, the number of separators returned due to case 4 is bounded by $k \cdot 2^k \cdot k \cdot 2^k \cdot 4^{k^2} \cdot 2^{(\mu-1)^3} \cdot 2^1$.

Using the fact that $k \leq \mu \leq 2k$ and $k > 10$, we note that the number of separators returned by each case is at most $\frac{1}{4} \cdot 2^{\mu^3}$, which yields the required bound. The number of separators returned is hence $2^{\mathcal{O}(k^3)}$. The algorithm spends $2^{2^{\mathcal{O}(k)}}n^{\mathcal{O}(1)}$ time at each node of the search tree and hence, the total time taken by the algorithm is $2^{2^{\mathcal{O}(k)}}n^{\mathcal{O}(1)}$. This completes the proof of the Lemma.

4.5 Proof of Theorem 1

Given Lemma 10 and Lemma 11, we do the following. Pick a set T_1 in $\mathcal{P}$ and guess a T_1-$T \setminus T_1$ separator well dominating the minimal part of the solution separating T_1 and $T \setminus T_1$. Once T_1 has been separated from the rest of the terminals, we use Lemma 1 to compute a minimum EMWC of T_1 in the component containing T_1. Following this, we pick another set from $\mathcal{P}$, and repeat. At the end of this procedure, we will be left with an instance of PMWC with no even terminals, resulting in an instance of the OMWC problem. In each step, we either pick a vertex in the solution or discard an even terminal. Hence the number of steps is bounded by $8k$ and by Lemma 11, in $2^{2^{\mathcal{O}(k)}}n^{\mathcal{O}(1)}$ time, we obtain $2^{\mathcal{O}(k^4)}$ instances of OMWC such that the given instance of PMWC is a YES instance if and only if one of these instances of OMWC is a YES instance. This, combined with Lemma 1 proves Theorem 1. □

5 Odd Multiway Cut parameterized by the solution size

In this section, we prove Lemma 1 by describing an FPT algorithm for OMWC. The main idea of this algorithm is that we reduce the instance of OMWC to an instance with certain properties, which ensure that we can solve it in FPT time.

Lemma 13. *Given an instance (G, T, k) of* OMWC*, let S be a solution and let C' be some semi-isolated component in $G \setminus S$. Then, no vertex of T occurs as a pivot of C'.*

Proof. We first recall that we have assumed some structural properties regarding the terminals of the input instance (see Lemma 6). We have already shown that if there is a solution, then there is one which is disjoint from the set of terminals. Our construction also ensures that for any solution, for every i, the terminals $\hat{T}_i$ occur in the same main component.

We now proceed to the proof of the lemma. Suppose that for some i, the vertex $\hat{t}_i^j$ is the pivot of C' for some j. Hence, there is a vertex $v \in C'$ which is adjacent to $\hat{t}_i^j$. But this vertex is also adjacent to the terminals $\hat{t}_i^l$, for every $l \neq j$ and these terminals are also contained in the same main component as $\hat{t}_i^j$. Now, we have a T-path $\hat{t}_i^j, v, \hat{t}_i^l$ intersecting C' and disjoint from S, which contradicts the assumption that C' is a semi-isolated component.

The properties we require the instance of OMWC to possess, have been formalized in the following definition.

Definition 11. *An instance (G, T, k) of* OMWC *is said to admit a* special *solution if there is a solution S such that each component of the isolated part and semi-isolated part of S is a single vertex. An instance which admits a special solution is called a* special instance.

Lemma 14. *Let (G, T, k) be an instance of* OMWC *which admits a special solution. Then we can find a solution for this instance in time $2.32^k n^{\mathcal{O}(1)}$.*

Proof. The proof is by a parameter preserving reduction to the variable version of ALMOST 2 SAT, called the ALMOST 2 SAT(VARIABLE) problem, which can then be solved in $2.32^k n^{\mathcal{O}(1)}$ time [20]. The reduction is as follows. For every vertex u of the graph, we have a variable x_u. The variable x_u is intended to represent the parity of T-u paths. The 2 SAT formula is constructed as follows. For every edge (t, u) where $t \in T$, add a clause (u). For every edge (u, v) in the graph, add two clauses $(x_u \vee x_v)$ and $(\bar{x}_u \vee \bar{x}_v)$ to the 2 SAT formula F. This completes the construction of F. We remark that both these clauses will be satisfied if and only if x_u and x_v are assigned different values. In addition, we also remark that the subformula F_P of F induced by the clauses corresponding to the edges of some odd T-path P is unsatisfiable, that is, we cannot find a satisfying assignment for it unless we delete some variables. We claim that if (G, T, k) admits a special solution, then (F, k) is a YES instance and if (F, k) is a YES instance of ALMOST 2 SAT(VARIABLE), then (G, T, k) is also a YES instance of OMWC.

Suppose (G, T, k) has a special solution S. Let S_v be the set of the variables corresponding to the vertices in S. We claim that the formula $F' = F \setminus S_v$ is satisfiable. Consider the following assignment for F'. Assign arbitrary values to the variables corresponding to vertices isolated from T in $G \setminus S$. For any vertex u in any main component, any T-u path must have the same parity (by Lemma 4). Assign 0 to x_u if all T-u paths are even and 1 otherwise. Now, for any vertex u in the semi-isolated part, due to Lemma 5, the corresponding variable remains in a single clause (whose other variable corresponds to a vertex in a main component and already has an assignment) and hence we assign the appropriate value to x_u so as to satisfy this clause. We call clauses thus satisfied, *explicitly* satisfied clauses. We claim that this assignment satisfies F'. Consider any clause $(l_1 \vee l_2)$ of F' and suppose that it is not one of the explicitly satisfied clauses. Since this clause remains after the deletion of some variables, it must be the case that neither of the vertices corresponding to the variables involved in the clause are deleted. Hence, both these vertices must lie in some main component of S. Since there is an edge between these two vertices, they must have paths of opposite parity to T and which implies that the corresponding variables are assigned different values. Hence, this clause is satisfied.

Conversely, consider a solution S_v for the instance (F, k), let $F' = F \setminus S_v$, and let S be the set of vertices corresponding to the variables in S_v. We claim that S is a solution for the given instance of OMWC. Suppose that this is not the case, and consider any odd T-path P in the graph $G \setminus S$ and let F_P be the subformula of F induced by the clauses corresponding to the edges in P. Since none of the vertices intersecting the path P have been deleted, it must be the case that none of the variables corresponding to the vertices along this path are in S_v. But this implies that F_P remains as a subformula of F' and since F_P is not satisfiable, F' is also not satisfiable, which is a contradiction.

In the rest of the section, we will show how, given an instance of OMWC, one can reduce it to equivalent instance(s) with a special solution.

5.1 Parity Preserving Torsos

In this section, we consider a fixed hypothetical solution S for the input instance (G, T, k), and examine the structure of the instance with respect to this solution. Using our observations, we will prove some additional structural claims and introduce definitions we need for the description of the randomized transformation, which will help us to construct a special instance from the given input instance. Following that, we will describe a randomized algorithm for the OMWC problem when parameterized by the solution size, and describe a derandomization procedure as well. We begin by defining the notion of *Parity Preserving Torsos*, which we will use to construct a set of equivalent instances of the problem.

Definition 12. *Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a vertex set. The graph $Parity\text{-}Torso(G, C)$ is constructed as follows. Consider the induced subgraph $G[C]$ and perform the following operations on it. For every pair $u, v \in C$, if there is an odd path in G from u to v whose internal vertices are not in C, then add an edge (u, v). Furthermore, for every pair $u, v \in C$, if there is an even path in G from u to v whose internal vertices are not in C, then add a subdivided edge between u and v. The resulting graph is referred to as $Parity\text{-}Torso(G, C)$ or $PT(G, C)$ and we call a newly added vertex which is part of a subdivided edge, a subdivision vertex.*

Lemma 15. *Given a graph $G = (V, E)$ and a vertex set C, the graph $PT(G, C)$ can be constructed in polynomial time.*

Proof. Testing if there is an even path between two vertices s and t in a graph can be done in polynomial time [19]. Testing if there is an odd path between s and t can be done by subdividing every edge incident on s and checking if there is an even s-t path in this modified graph. Hence, given C, for every component X of $G \setminus C$, for every pair of vertices of C in the neigborhood of this component, we can test in polynomial time if there is an even (respectively odd) path between these vertices, whose internal vertices lie in X. Since the number of components of $G \setminus C$ is at most $|V|$ and for each component there are at most $|V|^2$ pairs of neighbors in C, the graph $PT(G, C)$ can be constructed in polynomial time.

Note that intuitively, the operation $Parity\text{-}Torso$ is designed to preserve the parity of paths between vertices in C. The following lemma formalizes this intuition.

Lemma 16. *Let $C \subseteq V$ and let $u, v \in C$. If a set $S \subset C$ is such that all paths of some fixed parity from u to v in $PT(G, C)$ intersect S, then all paths of the same parity from u to v in G also intersect S.*

Proof. We first define the notion of *external subpaths* as follows. Consider some u-v path P and let $p, q \in C$ be two vertices on this path such that the subpath of P between p and q contains no other vertex in C. We call this subpath of P between p and q, an external subpath. We now prove the following claim which we will then use to prove the statement of the lemma.

Claim. Given an r-s path P in G, there is a path in $PT(G,C)$ which has the same parity as P, contains all the vertices in $P \cap C$, is disjoint from $C \setminus P$. That is, the only vertices on this path other than those in $P \cap C$ can be subdivision vertices between vertices in $P \cap C$.

Proof. We prove the claim by induction on the number of external subpaths of P. In the base case, the number of external subpaths of P is 0, which implies that P contains only vertices of C and hence lies in $PT(G,C)$. Thus, the claim is true for the case when the number of external subpaths of P is 0.

Now, suppose that the number of external subpaths of P is some $t > 0$ and assume that our claim holds true for all paths with less that t external subpaths. When traversing P from r to s, let x and y be the first two vertices in $C \cap P$ such that the subpath of P between x and y is an external subpath and call this subpath $\hat{P}$. If $\hat{P}$ has odd length, then, there is an edge between x and y in the graph $PT(G,C)$ and if $\hat{P}$ has even length, then there is a subdivided edge between x and y in the graph $PT(G,C)$. Let P_2 be the path (edge or subdivided edge) in $PT(G,C)$ between x and y which has the same parity as $\hat{P}$. Now, let P' be the subpath of P from r to x and let P'' be the subpath of P from y to s. The path P' lies entirely inside C and the path P'' is a path from y to s with less than t external subpaths. By the induction hypothesis, there is a path P_1 from y to s in C which has the same parity as P'', contains the vertices of $P'' \cap C$ and is disjoint from $C \setminus P''$. But now, $P' + P_2 + P''$ is a path from r to s in $PT(G,C)$ which has the same parity as P, contains the set of vertices in $P \cap C$ and is disjoint from $C \setminus P$. This completes the proof of the claim.

We now prove the statement of the lemma as follows. Suppose that S does not intersect all paths of said parity from u to v in G, and let P be a u-v path of this parity in $G \setminus S$. By the above claim, there is a path P_1 from u to v of the same parity in $PT(G,C)$ which contains $P \cap C$ and is disjoint from $C \setminus P$. Since P is disjoint from S, it must be the case that $S \subseteq C \setminus P$ and hence P_1 is also disjoint from S, which is a contradiction to our assumption that S intersects all u-v paths of that parity in $PT(G,C)$.

Definition 13. *Given an instance $I = (G = (V,E), T, k)$ of* OMWC*, and a set $Z \subseteq V \setminus T$, we define an instance I/Z of* OMWC *as $(PT(G, V \setminus Z), T, k)$.*

Observe that, while constructing the instance I/Z, the $Parity\text{-}Torso$ operation is applied on the set $V \setminus Z$. The following lemma shows that if we compute a set Z with certain properties, an application of the $Parity\text{-}Torso$ operation depending on this set will lead us to an equivalent special instance of the problem.

Lemma 17. *Let $I = (G = (V,E), T, k)$ be an instance of* OMWC *and let $Z \subseteq V \setminus T$. If I is a* NO *instance, then I/Z is also a* NO *instance. If I has a solution S such that Z contains the isolated and semi-isolated parts of S and $Z \cap (S \cup T) = \emptyset$ (see Fig. 8), then S is a special solution for the instance I/Z.*

Proof. Suppose that I/Z is a YES instance and suppose that S is a solution for this instance. By Lemma 16, any even T-path in I intersects S as well and hence S is a

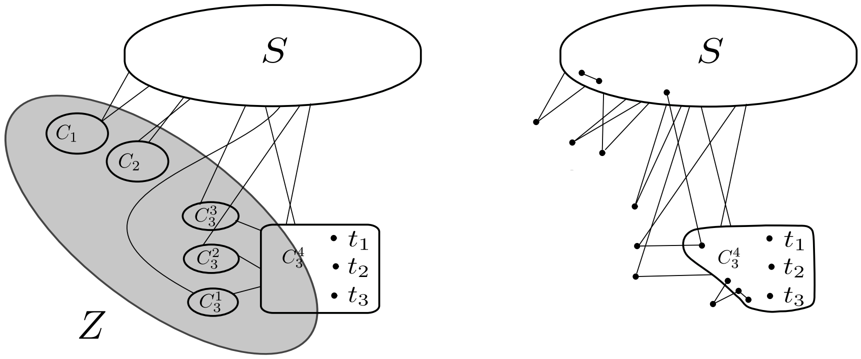

Fig. 8. Illustration of an application of the *parity-torso* operation.

solution for I, which makes it a YES instance. This proves the first statement of the lemma.

We now prove the second statement of the lemma. Suppose that I has a solution S such that Z contains the isolated and semi isolated parts of S and $Z \cap (S \cup T) = \emptyset$. We will first show that S is indeed a solution for I/Z. Before we prove this, we make the following claim.

Claim. Any new edge (or subdivided edge) added between two vertices of some main component during the construction of the graph $PT(G, V \setminus Z)$, is due to a path which lies entirely in the same main component.

Proof. Consider two vertices x and y in some main component and let P_2 be a newly added edge (respectively subdivided edge) between these two vertices. Let P be a path from x to y which has the same parity as P_2 and has its internal vertices in Z. Since Z is disjoint from S, the path P also is disjoint from S. If the path P intersects some isolated or semi-isolated component, it must also intersect S, which is not possible. This implies that P lies entirely in the same main component as x and y. This completes the proof of the claim.

We will now prove that S is a solution for the instance I/Z as follows. Suppose that this is not the case, and let P be an odd path from t_1 to t_2 in the graph $PT(G, V \setminus Z) \setminus S$. Since the set Z contains all the isolated and semi-isolated components of S, the only original vertices of the graph which occur in P are from a single main component of S. By the claim above, any edge (or subdivided edge) between these vertices in $PT(G, V \setminus Z)$ is due to a path lying in the main component. Hence, we can replace the newly added edges (respectively subdivided edges) in P with the original paths in G, to get an even walk W from t_1 to t_2 which lies entirely inside this main component. We now claim that every closed loop in W is of even length and hence removing these loops enables us to obtain an even path from t_1 to t_2. Suppose that there is an odd closed loop in W. Then, by Lemma 2, for some vertex on this odd loop, there are both odd and even length paths to T, which is a contradiction to Lemma 4. Hence, every loop of W is even and we can remove these loops to obtain an odd T-path disjoint from S (see proof of Lemma 2), which is a contradiction to our original assumption that S is a solution for the instance I.

We will now show that S is indeed a special solution for the instance I/Z. We will show this by considering each vertex of $PT(G, V \setminus Z) \setminus S$ and examining the component it belongs to. Consider a vertex v in $PT(G, V \setminus Z) \setminus S$. The vertex v must either be a vertex in G or a subdivision vertex added as a result of a subdivided edge. If v is a vertex in G, it must occur in a main component of S in I. But this implies that v also occurs in a main component of S in I/Z since the $Parity\text{-}Torso$ operation maintains connectivity along with parity for this vertex with respect to the other vertices disjoint from Z. We can also argue along the same lines as in the proof of Lemma 5 that, if v is a subdivision vertex between two vertices in the main component of S in I, then it also lies on a T-path and hence occurs in the same main component of S as these two vertices in I/Z. Now, suppose v is a subdivision vertex which occurs as a result of a subdivided edge added between a vertex of S and a vertex u in some some main component of S in I, which is disjoint from Z. Since u remains in some main component of S in I/Z, v will be a singleton semi-isolated component of S in I/Z. Finally, suppose that v is a subdivision vertex which occurs as a result of a subdivided edge added between two vertices of S. Then, v is a singleton isolated component. Hence, S is a special solution for the instance I/Z and this proves the second statement of the lemma.

In the following subsections, we examine the structure of such a set Z and using our observations, describe a method of constructing such a set.

5.2 Important components and clusters

Definition 14. *We call J an* important component *if $G[J]$ is connected and $N(J)$ is an important J-T separator of size at most $k + 1$.*

The motivation for the above definition is due to the following lemma, which will allow us to consider the required set Z as the union of a set of important components.

Lemma 18. *Let (G, T, k) be a* YES *instance of* OMWC. *Then, there is a solution S such that every component of the isolated and semi-isolated part of S is an important component.*

Proof. Let S be a smallest solution for the given instance which maximizes the sum of the sizes of the isolated part and the semi-isolated part. We will prove that every component of the isolated and semi-isolated part of this solution is an important component. Suppose this is not so. We have the following two cases.

Case 1. Suppose that some component C_1 in the semi-isolated part is not an important component. Since the size of the neighborhood of C_1 is at most $k + 1$, it must be the case that there is an important separator K which dominates the C_1-T separator $N(C_1)$. Define the set $S' = (S \setminus N(C_1)) \cup (K \setminus \{w\})$ where w is an arbitrary vertex of K. Clearly, $|S'| \leq |S|$. We claim that S' is a solution for the given instance as well. Suppose this is not the case and let P be an odd T-path in the graph $G \setminus S'$. It must be the case that this path contains some vertex v in $S \setminus S'$, which implies that v is in $(N(C_1) \setminus K) \cup \{w\}$. Since v lies on a T-path, it has two vertex disjoint paths to T. But K is an important C_1-T separator dominating $N(C_1)$ and hence is a v-T separator as well. But then, K must contain two vertices of the path P. Hence, $K \setminus \{w\}$ contains some vertex of this T-path, which is a contradiction to our assumption that S' is not a solution. This proves that S' is indeed a solution. In addition, we may assume that S' is inclusionwise minimal, that is, for every vertex of S', there is an odd T-path which intersects this vertex and is disjoint from the rest of S'. This is because if this were not the case, and we have a vertex u in S' such that every odd T-path it intersects also intersects some vertex in $S' \setminus \{u\}$, we can remove u from S', and still have a solution.

We now claim that the sum of the sizes of the isolated and semi-isolated parts of S' is strictly greater than that of S, which will contradict our assumption about the maximality of S with respect to the size of the isolated and semi-isolated part.

We first observe that S' is disjoint from the isolated part of S. If this were not the case, then there is a vertex u of S' in some component in the isolated part of S. Since S' is minimal, there is an odd T-path P which intersects u and is disjoint from $S' \setminus \{u\}$. Since any T-path intersecting u must intersect S in at least two vertices, P must contain two vertices of $S \setminus S'$ and hence this implies the existence of two vertex disjoint $\{v_1, v_2\}$-T paths for some $v_1, v_2 \in S \setminus S'$ and these paths are disjoint from u as well. But K intersects all $(S \setminus S')$-T paths and hence $K \setminus \{w\}$ intersects at least one of the two vertex disjoint $\{v_1, v_2\}$-T paths. This implies that $S' \setminus \{u\}$ is also a solution, which contradicts the minimality of S'. Hence, S' is disjoint from the isolated part of S.

We now observe that S' is also disjoint from the semi-isolated part of S. To prove this, it is sufficient to show that K does not intersect the semi-isolated part of S. Suppose this is not the case and let u be a vertex of K which intersects some semi-isolated component of S, say C. Then, in K, we replace u with the pivot of C to get a set K' and claim that K' is a C_1-T separator which dominates K. To prove this, it is enough to show that K' is a C_1-T separator. If this were not the case, consider a C_1-T path P which intersects u and is disjoint from $\chi(C)$. Since the path P must enter and leave the semi-isolated component without intersecting the pivot, this implies that P contains two vertices x and y of $S \cap N(C_1)$ such that u is contained in the subpath of P between x and y. Let P_1 be the subpath of P from y to T. Now, consider an edge $e = (a, y)$ where $a \in C_1$. We know that such an edge exists since y is in $N(C_1)$. But now, we have a C_1-T path $e + P_1$ disjoint from K, which is a contradiction to K being a C_1-T separator. We have thus proved that S' is also disjoint from the semi-isolated part of S.

We will now show that the vertices in $S \setminus S'$ are either in the isolated or semi-isolated part of S'. Following that, we will show that every vertex in the isolated or semi-isolated part of S is also in either the isolated or semi-isolated part of S'. Since $S \setminus S'$ is non empty, these two statements imply that the sum of the sizes of the isolated and semi-isolated components of S' is strictly larger than that of S.

Consider a vertex v in $S \setminus S'$. Note that, in order to show that v is in an isolated of semi-isolated component of S', it is sufficient for us to show that there is no T-path intersecting v in the graph $G \setminus S'$. Suppose that this is not the case, and let P be a T-path intersecting v in the graph $G \setminus S'$. Then, there are two vertex disjoint paths from v to T.

But we know that K is an $(S \setminus S')$-T separator and hence there can be at most one v-T path in the graph $G \setminus (K \setminus \{w\})$ which implies that there can be at most one path in the graph $G \setminus S'$, which is a contradiction. Thus, we have proved that there is no T-path intersecting v in the graph $G \setminus S'$ and hence proved that the vertices in $S \setminus S'$ are in either the isolated or semi-isolated part of S'.

Consider a vertex v in the isolated or semi-isolated part of S. In order to show that v is in an isolated or semi-isolated component of S', it is sufficient for us to show that there is no T-path intersecting v in the graph $G \setminus S'$. Suppose that this is not the case, and let P be a T-path containing v in the graph $G \setminus S'$. Since v is in the isolated or semi-isolated part of S, the path intersects S and hence intersects $S \setminus S'$. This implies that there is a vertex $u \in S \setminus S'$ which lies on a T-path in the graph $G \setminus S'$. But, we have already shown that no vertex in $S \setminus S'$ can lie on a T-path in the graph $G \setminus S'$ and thus we have reached a contradiction. Hence, we have proved that every vertex in the isolated or semi-isolated part of S stays in the isolated or semi-isolated part of S' as well. This completes the proof for case 1.

Case 2. Suppose that some component C_2 in the isolated part is not an important component. Since the size of the neighborhood of C_2 is at most k, it must be the case that there is an important separator K which dominates $N(C_2)$. Define $S' = (S \setminus N(C_2)) \cup K$. Clearly, S' is no larger than S. We claim that S' is also a solution for the given instance.

Suppose this is not the case and let P be an odd T-path in the graph $G \setminus S'$. It must be the case that this path contains some vertex v in $S \setminus S'$, which implies that v is in $(N(C_2) \setminus K)$. But K is an important C_2-T separator dominating $N(C_2)$ and hence is a v-T separator as well. Hence, K intersects the path P, which is a contradiction to our assumption that S' is not a solution. This proves that S' is indeed a solution. In addition, we may assume that S' is inclusionwise minimal.

We now claim that the sum of the sizes of the isolated and semi-isolated parts of S' is strictly greater than that of S, which will contradict our assumption about the maximality of S with respect to the size of the isolated and semi-isolated part.

We first observe that S' is disjoint from the isolated part of S. If this were not the case, then there is a vertex u of S' in some component in the isolated part of S. Since S' is minimal, there is an odd T-path P which intersects u and is disjoint from $S' \setminus \{u\}$. Since any T-path intersecting u must intersect S in at least two vertices, P must contain two vertices of $S \setminus S'$ and hence this implies the existence of two vertex disjoint $\{v_1, v_2\}$-T paths for some $v_1, v_2 \in S \setminus S'$ and these paths are disjoint from u as well. But K intersects all $(S \setminus S')$-T paths and this implies that $S' \setminus \{u\}$ is also a solution, which contradicts the minimality of S'. Hence, S' is disjoint from the isolated part of S.

We now observe that S' is also disjoint from the semi-isolated part of S. To prove this, it is sufficient to show that K does not intersect the semi-isolated part of S. Suppose this is not the case and let u be a vertex of K which intersects some semi-isolated component of S, say C. Then, in K, we replace u with the pivot of C to get a set K' and claim that K' is a C_2-T separator which dominates K. To prove this, it is enough to show that K' is a C_2-T separator. If this were not the case, consider a C_2-T path P which intersects u and is disjoint from $\chi(C)$. Since the path P must enter and leave the

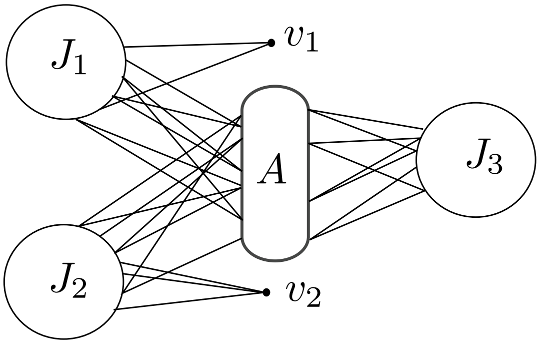

Fig. 9. Illustration of an important cluster L_A with important components J_1, J_2, and J_3, where $N(J_1) = A \cup \{v_1\}$, $N(J_2) = A \cup \{v_2\}$ and $N(J_3) = A$.

semi-isolated component without intersecting the pivot, this implies that P contains two vertices x and y of $S \cap N(C_2)$ such that u is contained in the subpath of P between x and y. Let P_1 be the subpath of P from y to T. Now, consider an edge $e = (a, y)$ where $a \in C_2$. We now that such an edge exists since $y \in N(C_2)$. But now, we have a C_2-T path $e + P_1$ disjoint from K, which is a contradiction to K being a C_2-T separator. We have thus proved that S' is also disjoint from the semi-isolated part of S.

We will now show that any vertex which lies in $S \setminus S'$, or in the isolated or semi-isolated part of S lies in the isolated or semi-isolated part of S'. Since $S \setminus S'$ is non empty, this implies that the sum of the sizes of the isolated and semi-isolated components of S' is strictly larger than that of S.

Consider a vertex v which is in $S \setminus S'$ or in the isolated or semi-isolated part of S. Note that, in order to show that v is in an isolated or semi-isolated component of S', it is sufficient for us to show that there is no T-path intersecting v in the graph $G \setminus S'$. We know that any T-path in the graph $G \setminus S'$ intersecting $S \setminus S'$ or the isolated or semi-isolated part of S must intersect $S \setminus S'$. But K is an $(S \setminus S')$-T separator, which implies that there cannot be a T-path intersecting v in the graph $G \setminus S'$. This completes the proof for case 2.

In either case, we have arrived at a contradiction and this completes the proof of the lemma.

Definition 15. *For every $A \subseteq V$, the important cluster L_A is the disjoint union of every important component with $N(C) = A \cup P$ where the set P has size at most 1 and could be different for different important components (see Fig 9).*

The motivation behind the definition of important clusters is that, while the number of important components we are interested in may not be bounded by a function of k, there is a function $f : \mathbb{N} \to \mathbb{N}$, such that they can be partitioned into at most $f(k)$ important clusters.

Observation 7 *Any important component can be a part of at most $k + 2$ important clusters.*

Proof. Consider an important component J and let $D = N(J)$. The only clusters which J can occur in correspond to subsets of D of size at least $|D| - 1$. Since $|D| \leq k + 1$, the number of clusters which J can occur in is at most $k + 2$.

Lemma 19. *Every vertex $v \in V$ is contained in at most 4^{k+1} important components and at most $(k+2)4^{k+1}$ important clusters. Furthermore, all the important components and clusters can be enumerated in time $\mathcal{O}(4^k n^{\mathcal{O}(1)})$.*

Proof. Consider an important component J. Then, $N(J)$ is an important v-T separator for any $v \in J$. Since there are only at most 4^{k+1} important v-T separators, v can occur in at most 4^{k+1} important components and by Observation 7, v can occur in at most $4^{k+1}(k + 2)$ important clusters. The set of important components can be computed by computing the set of important v-T separators for every vertex v in the graph (Lemma 8). Once the set of important components is computed, we can compute the set of important clusters by adding each important component to the appropriate clusters. By Lemma 8, this requires time $\mathcal{O}(4^k n^{\mathcal{O}(1)})$.

5.3 Randomized selection of important components

We recall that our objective is to find a set Z which contains the isolated and semi-isolated part of some solution S, and is disjoint from S. By Lemma 18, we know that there exists such a set Z which contains only important components. Note that, if we take the important clusters corresponding to all the subsets of the solution S and take their union, then this set will contain Z. Hence, Z is contained in the union of at most 2^k important clusters. Now, we will select important clusters randomly and compute the probability with which we will have selected a set of vertices which contains all the isolated and semi-isolated components of some solution and is disjoint from the solution. We first enumerate all important clusters and pick each one independently, with probability $\frac{1}{2}$. Call the union of all important clusters thus picked, $\mathcal{H}$. We will now compute the probability that the set $\mathcal{H}$ satisfies the required properties.
1. The probability that $\mathcal{H}$ contains the isolated and semi-isolated part of S is equal to the probability that we pick all the at most 2^k clusters whose disjoint union is the isolated and semi-isolated part of S. This probability is at least $(\frac{1}{2})^{2^k}$.
2. The probability that $\mathcal{H}$ is disjoint from S, is at least $(1 - \frac{1}{2})^{k(k+2)4^{k+1}}$, since there are at most $k(k + 2)4^{k+1}$ important clusters intersecting S.
Finally, since these two events are independent, the probability that they both occur is at least $(\frac{1}{2})^{2^k}(1 - \frac{1}{2})^{k(k+2)4^{k+1}} = 2^{-2^{\mathcal{O}(k)}}$. Hence, by Lemma 17, with probability at least $2^{-2^{\mathcal{O}(k)}}$, the instance $I \setminus \mathcal{H}$ is a special instance, which can solved by applying Lemma 14. Thus, we have a randomized algorithm (see Algorithm 5.1) for OMWC which has a success probability of $2^{-2^{\mathcal{O}(k)}}$.

5.4 Derandomization

We derandomize the selection of important clusters by using the standard technique of splitters. An (n, r, r^2)-*splitter* is a family of functions from $[n]$ to $[r^2]$ such that for any

```
Input   : An instance $(G, T, k)$ of OMWC
Output: A solution of size at most $k$ for the instance $(G, T, k)$ if it exists and No
        otherwise
Enumerate all important components.
Enumerate all important clusters.
Select each cluster independently, with probability $\frac{1}{2}$. Call the union of the selected
  clusters Z.
Construct the instance $I/Z$.
Compute a solution $S$ for this instance by reduction to an instance of Almost 2
  SAT(Variable) and solving this instance.
if $S$ is not No and $S$ is a solution to the input instance then
    return $S$.
end
return No
```

Algorithm 5.1: Randomized algorithm for OMWC

$r-$sized subset X of $[n]$, there is a function in this family which is injective on X. There exist explicit constructions of an (n, r, r^2)-splitter of size $\mathcal{O}(r^6 \log r \log n)$ [24].

Let $\mathcal{C}$ be the set of all important clusters in G. We let $\mathcal{P}$ be the set of important clusters whose disjoint union is the union of the isolated and semi-isolated parts of S and let $\mathcal{Q}$ be the set of all important clusters which intersect S. We know that $|\mathcal{C}| = c \leq n(k+2)4^{k+1}$, $|\mathcal{P}| = p \leq 2^k$, and $|\mathcal{Q}| = q \leq k(k+2)4^{k+1}$. For the deterministic selection of important clusters, we go through each function in a $(c, p+q, (p+q)^2)$-splitter family $\mathcal{F}$ and every subset $F \subseteq [(p+q)^2]$. For some $f \in \mathcal{F}$ and F, we select those sets of $\mathcal{C}$ which f maps to an element in F and call the union of these sets $\mathcal{H}$.

We now count the number of instances returned by this algorithm. The size of the splitter family we use is bounded by $2^{\mathcal{O}(k)} \log n$ and for each function in the splitter family, we have $\binom{(p+q)^2}{p+q} = 2^{2^{\mathcal{O}(k)}}$ subsets F. Hence, the number of instances returned by this algorithm is $2^{2^{\mathcal{O}(k)}} \log n$ and we have the following lemma.

Lemma 20. *There is an algorithm that, given an instance (G, T, k) of* OMWC*, runs in time $2^{2^{\mathcal{O}(k)}} n^{\mathcal{O}(1)}$ and returns $2^{2^{\mathcal{O}(k)}} \log n$ instances such that if I is a* Yes *instance, then at least one of the returned instances has a special solution and if I is a* No *instance, then all of the returned instances are* No *instances.*

5.5 Proof of Lemma 1

We are now ready to prove Lemma 1. In our algorithm for OMWC, we first compute the set of all important clusters and use Lemma 20 to obtain a set of instances of OMWC in $2^{2^{\mathcal{O}(k)}} n^{\mathcal{O}(1)}$ time such that if the given instance is a Yes instance, then at least one of the returned instances has a special solution. We then apply the reduction of Lemma 14 to each of these instances. Each such reduction only requires polynomial time. We then solve the resulting Almost 2 SAT(Variable) instances and return Yes if and only if at least one of the Almost 2 SAT(Variable) instances was found to be a Yes instance. The time required to solve ALMOST 2 SAT(VARIABLE) for each instance is $2^{\mathcal{O}(k)}n^{\mathcal{O}(1)}$. Hence, our algorithm for OMWC runs in time $2^{2^{\mathcal{O}(k)}}n^{\mathcal{O}(1)}$. □

6 EDGE PARITY MULTIWAY CUT parameterized by the solution size

Lemma 21. EDGE PARITY MULTIWAY CUT *(*EPMWC*) is* NP*-complete for* $|T| \geq 3$.

Proof. It is clear that this problem is in NP. We prove the NP-hardness of EPMWC by showing that even the edge variant of OMWC is NP-hard for $|T| \geq 3$. Since OMWC is a special case of PMWC, the lemma follows. We give a polynomial time many-one reduction from the EDGE MULTIWAY CUT problem (EDGE MWC) to EDGE ODD MULTIWAY CUT (EOMWC). In the EDGE MWC problem we are given a graph G, a set of terminals T and a positive integer k and the objective is to determine if there is a k-sized subset of edges, whose removal from the graph disconnects every pair of terminals. EDGE MWC is NP-complete for $|T| \geq 3$ [6].

Given an instance $(G = (V, E), T, k)$ of EDGE MWC, for each terminal $t_i \in T$, we add $3k+4$ vertices $t^1_{i_1}, \dots, t^1_{i_{k+1}}, t^2_{i_1}, \dots, t^2_{i_{k+1}}, t^3_{i_1}, \dots, t^3_{i_{k+1}}, t'_i$. Add an edge between t_i and $t^l_{i_j}$ for every $1 \leq j \leq k+1$ and $l \in \{1, 3\}$ (see Fig. 10). Add an edge between t'_i and $t^l_{i_j}$ for every $1 \leq j \leq k+1$ and $l \in \{2, 3\}$. Add an edge between $t^1_{i_j}$ and $t^2_{i_j}$ for every $1 \leq j \leq k+1$. Let the resulting graph be called G'. Define a new set of terminals $T' = \{t'_i | t_i \in T\}$. We claim that the instance (G, T, k) is a YES instance of EDGE MWC if and only if the instance (G', T', k) is a YES instance of EOMWC.

Let S be a solution for the instance (G, T, k). We claim that S is also a solution for the instance (G', T', k). Suppose that this is not the case and consider some odd T'-path P in $G' \setminus S$. Suppose the path P is from t'_i to t'_j. Clearly, this path has to intersect t_i and t_j, which implies that there is a path from t_i to t_j in the graph $G' \setminus S$. Since the newly added edges and vertices do not create any new T-paths, this path also exists in the graph $G \setminus S$, which is a contradiction to our assumption.

Now, let S' be a solution for the instance (G', T', k). Note that S' will not contain any newly added edges. We claim that S' is a solution for the instance (G, T, k). Suppose this is not the case and let P be a T-path in the graph $G \setminus S'$ from t_i to t_j. Since S' is disjoint from the newly added edges, we can extend this path P on both sides using the appropriate newly added edges, to get an odd T'-path in $G' \setminus S'$, which is a contradiction. This completes the proof.

Theorem 8. EPMWC *can be solved in time* $2^{2^{\mathcal{O}(k)}}n^{\mathcal{O}(1)}$.

Proof. The proof is by a polynomial time parameter preserving reduction to PMWC where the solution is required to be disjoint from the terminals. Theorem 1 also holds for this case (since we assume that the solution is disjoint from the terminals from Lemma 11 onwards). Let $(G = (V, E), T = T_e \cup T_o, k)$ be an instance of EPMWC. We construct a graph G' as follows. Take the set T and for each vertex $v \in V \setminus T$, make $k+1$ copies $v^1, \dots, v^{k+1}$ of this vertex. For each edge $e = (u, v) \in E$, add two

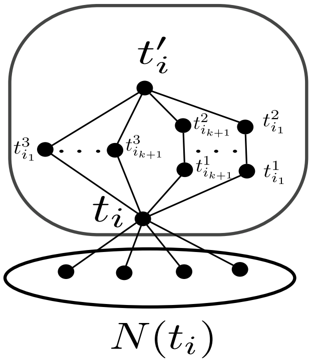

Fig. 10. An illustration of the terminal gadget in Lemma 21.

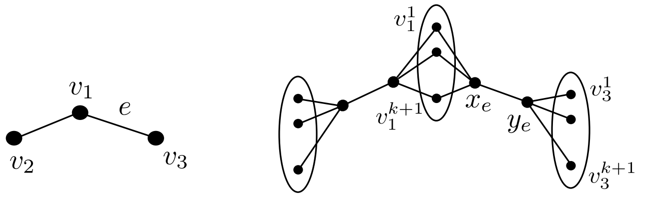

Fig. 11. An illustration of the reduction from EPMWC to PMWC.

vertices x_e and y_e and add an edge between them (see Fig. 11). Furthermore, add an edge from each copy of u (if $u \in T$, there is a single copy) to x_e and from each copy of v to y_e. We call the resulting graph G' and keep the set of terminals the same. This completes the construction. We claim that (G, T, k) is a YES instance of EPMWC if and only if (G', T, k) is a YES instance of PMWC where the solution is disjoint from T.

Suppose that S is a solution of size at most k for the instance (G, T, k). For each edge e in S, pick one of the vertices x_e or y_e and we claim that this set S' forms a solution of the required kind for the instance (G', T, k). Clearly $S' \cap T = \emptyset$. Now, suppose that there is a t_1-t_2 path P of forbidden parity in $G' \setminus S'$. The existence of this path implies that no vertex of the form x_e or y_e lying on this path has been deleted, which implies that none of the corresponding edges in the original graph were deleted. But these edges together constitute a t_1-t_2 path of the same parity in G disjoint from S, which is a contradiction.

Conversely, let S' be a solution for the instance (G', T, k) such that $S' \cap T = \emptyset$. Note that the construction of G' then ensures that S' only contains vertices of the form x_e or y_e. For each vertex x_e or y_e in S', pick the corresponding edge in G and call this set S. We claim that S is a solution for the instance (G, T, k). Clearly, $|S| \leq k$. Suppose that there is a t_1-t_2 path P_1 of forbidden parity in $G \setminus S$. Since no edge along this path is in S, neither the x_e nor the y_e vertex corresponding to any edge e in this path is in S'. But this implies that existence of a t_1-t_2 path of the same parity in $G' \setminus S'$, which is a contradiction.

7 Conclusion

In this paper, we introduce a notion of generalized important separators and by supplementing this idea with randomized selection of important components, along with the iterative compression technique, we give an FPT algorithm for a parity based generalization of the classical MULTIWAY CUT problem, the PARITY MULTIWAY CUT problem. The design of improved FPT algorithms for this problem, as well as FPT algorithms for other parity based separation problems like the parity version of MULTICUT remains an interesting open problems, as does the kernelization complexity of these problems.

Acknowledgements

We would like to thank Saket Saurabh for the insightful discussions on graph separation problems.

References

1. N. Bousquet, J. Daligault, and S. Thomassé. Multicut is fpt. In *STOC*, pages 459–468, 2011.
2. J. Chen, Y. Liu, and S. Lu. An improved parameterized algorithm for the minimum node multiway cut problem. *Algorithmica*, 55(1):1–13, 2009.
3. J. Chen, Y. Liu, S. Lu, B. O'Sullivan, and I. Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. *J. ACM*, 55(5), 2008.
4. R. H. Chitnis, M. Hajiaghayi, and D. Marx. Fixed-parameter tractability of directed multiway cut parameterized by the size of the cutset. In *SODA*, pages 1713–1725, 2012.
5. M. Cygan, M. Pilipczuk, M. Pilipczuk, and J. O. Wojtaszczyk. On multiway cut parameterized above lower bounds. IPEC, 2011.
6. E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *Siam Journal on Computing*, 23:864–894, 1994.
7. R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer-Verlag, New York, 1999.
8. J. Flum and M. Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2006.
9. N. Garg, V. V. Vazirani, and M. Yannakakis. Multiway cuts in directed and node weighted graphs. In *ICALP*, volume 820 of *Lecture Notes in Computer Science*, pages 487–498, 1994.
10. J. Geelen, B. Gerards, B. A. Reed, P. D. Seymour, and A. Vetta. On the odd-minor variant of hadwiger's conjecture. *J. Comb. Theory, Ser. B*, 99(1):20–29, 2009.

11. K. ichi Kawarabayashi, Z. Li, and B. A. Reed. Recognizing a totally odd k_4-subdivision, parity 2-disjoint rooted paths and a parity cycle through specified elements. In *SODA*, pages 318–328, 2010.
12. K. ichi Kawarabayashi and B. A. Reed. A nearly linear time algorithm for the half integral parity disjoint paths packing problem. In *SODA*, pages 1183–1192, 2009.
13. K. ichi Kawarabayashi and B. A. Reed. An (almost) linear time algorithm for odd cyles transversal. In *SODA*, pages 365–378, 2010.
14. K. ichi Kawarabayashi and B. A. Reed. Odd cycle packing. In *STOC*, pages 695–704, 2010.
15. K. ichi Kawarabayashi, B. A. Reed, and P. Wollan. The graph minor algorithm with parity conditions. In *FOCS*, pages 27–36, 2011.
16. K. ichi Kawarabayashi and M. Thorup. The minimum k-way cut of bounded size is fixed-parameter tractable. In *FOCS*, pages 160–169, 2011.
17. N. Kakimura, K. ichi Kawarabayashi, and Y. Kobayashi. Erdös-pósa property and its algorithmic applications: parity constraints, subset feedback set, and subset packing. In *SODA*, pages 1726–1736, 2012.
18. N. Kakimura, K. ichi Kawarabayashi, and D. Marx. Packing cycles through prescribed vertices. *J. Comb. Theory, Ser. B*, 101(5):378–381, 2011.
19. A. S. Lapaugh and C. H. Papadimitriou. The even-path problem for graphs and digraphs. *Networks*, 14(4):507–513, 1984.
20. D. Lokshtanov, N. S. Narayanaswamy, V. Raman, M. S. Ramanujan, and S. Saurabh. Faster parameterized algorithms using linear programming. *Manuscript*, 2012.
21. W. Mader. Über die Maximalzahl kreuzungsfreier H-Wege. *Arch. Math. (Basel)*, 31(4):387–402, 1978/79.
22. D. Marx. Parameterized graph separation problems. *Theoret. Comput. Sci.*, 351(3):394–406, 2006.
23. D. Marx and I. Razgon. Fixed-parameter tractability of multicut parameterized by the size of the cutset. In *STOC*, pages 469–478, 2011.
24. M. Naor, L. J. Schulman, and A. Srinivasan. Splitters and near-optimal derandomization. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, FOCS '95, pages 182–191, Washington, DC, USA, 1995. IEEE Computer Society.
25. R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*, volume 31 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, Oxford, 2006.
26. V. Raman, M. S. Ramanujan, and S. Saurabh. Paths, flowers and vertex cover. In *ESA*, volume 6942 of *Lecture Notes in Computer Science*, pages 382–393, 2011.
27. I. Razgon and B. O'Sullivan. Almost 2-sat is fixed-parameter tractable. *J. Comput. Syst. Sci.*, 75(8):435–450, 2009.
28. B. A. Reed, K. Smith, and A. Vetta. Finding odd cycle transversals. *Oper. Res. Lett.*, 32(4):299–301, 2004.