Path-contractions, edge deletions and connectivity preservation ☆

Gregory Gutin[a], M. S. Ramanujan[b,1,*], Felix Reidl[a], Magnus Wahlström[a]

[a] *Royal Holloway University, University of London, Egham, UK*
[b] *Algorithms and Complexity Group, TU Wien, Vienna, Austria*

Abstract

We study several problems related to graph modification problems under connectivity constraints from the perspective of parameterized complexity: (a) (Weighted) Biconnectivity Deletion, where we are tasked with deleting k edges while preserving biconnectivity in an undirected graph, and its generalization, (b) Vertex-deletion Preserving Strong Connectivity, where we want to maintain strong connectivity of a digraph while deleting exactly k vertices, and (c) Path-contraction Preserving Strong Connectivity, in which the operation of path contraction on arcs is used instead. The parameterized tractability of this last problem was posed in [Bang-Jensen and Yeo, Discrete Applied Math 2008] as an open question and we answer it here in the negative: both variants of preserving strong connectivity are W[1]-hard. Preserving biconnectivity, on the other hand, turns out to be fixed-parameter tractable (FPT) and we provide an FPT algorithm that solves Weighted Biconnectivity Deletion. Moreover, we show that the unweighted case even admits a randomized polynomial kernel. Finally, we show that the most general case of the (unweighted) problem where one would like to preserve ρ-vertex connectivity for any ρ is (non-uniformly) FPT parameterized by k and ρ. This answers an open problem of Bang-Jensen et al. [1], who proved the fixed-parameter tractability of the *weighted, directed* version of the same problem. All our results provide further interesting data points for the systematic study of connectivity-preservation constraints in the parameterized setting.

Keywords: Parameterized Complexity, Graph Connectivity, vertex deletion, path contraction

1. Introduction

Some of the most well studied classes of network design problems involve starting with a given network and making modifications to it so that the resulting network satisfies certain

☆A preliminary version of this paper appeared in the proceedings of The 25th Annual European Symposium on Algorithms (ESA 2017). The research of Gregory Gutin was partially supported by Royal Society Wolfson Research Merit Award. M. S. Ramanujan acknowledges support from Austrian Science Fund (FWF, project P26696). Felix Reidl and Magnus Wahlström were supported by EPSRC grant EP/P007228/1.

*Corresponding author

Email addresses: G.Gutin@rhul.ac.uk (Gregory Gutin), ramanujan@ac.tuwien.ac.at (M. S. Ramanujan), felix.reidl@gmail.com (Felix Reidl), Magnus.Wahlstrom@rhul.ac.uk (Magnus Wahlström)

[1]This author has now moved to the University of Warwick, Coventry, UK.

connectivity requirements, for instance a prescribed edge- or vertex-connectivity. This class of problems has a long and rich history (see *e.g.* [2, 3]) and has recently started to be examined through the lens of parameterized complexity. Under this paradigm, we ask whether a (hard) problem admits an algorithm with a running time $f(k)n^{O(1)}$, where n is the size of the input, k the *parameter*, and f some computable function. A natural parameter to consider in this context is the number of editing operations allowed and we can reasonably assume that this number is small compared to the size of the graph.

To approach this line of research systematically, let us identify the 'moving parts' of the broader question of editing under connectivity-constraints: first and foremost, the network in question might best be modelled as either a directed or undirected graph, potentially with edge- or vertex-weights. This, in turn, informs the type of connectivity we restrict, *e.g.* strong connectivity or fixed value of edge-/vertex-connectivity. Additionally, the connectivity requirement might be non-uniform, *i.e.* it might be specified for individual vertex-pairs. The constraint one operates under might either be to *preserve*, to *augment*, or to *decrease* said connectivity. Finally, we need to fix a suitable editing operation; besides the obvious vertex- and edge-removal, more intricate operations like edge contractions are possible.

While not all possible combinations of these factors might result in a problem that currently has an immediate real-world application, they are nonetheless important data points in the systematic study of algorithmic tractability. For example, if we fix the editing operation to be the addition of edges (often called 'links' in this context) and our goal is to increase connectivity, then the resulting class of *connectivity augmentation problems* has been thoroughly researched. We refer to the monograph by Frank [3] for further results on polynomial-time solvable cases and approximation algorithms. Under the parameterized complexity paradigm, Nagamochi [4] and Guo and Uhlmann [5] studied the problem of augmenting a 1-edge- connected graph with k links to a 2-edge-connected graph. Nagamochi obtained an FPT algorithm for this problem while Guo and Uhlmann showed that this problem, alongside its vertex-connectivity variant, admits a quadratic kernel. Marx and Végh [6] studied the more general problem of augmenting the edge-connectivity of an undirected graph from $\lambda - 1$ to λ, via a minimum set of links that has a total *cost* of at most k, and obtained an FPT algorithm as well as a polynomial kernel for this problem. Basavaraju et al. [7] improved the running time of their algorithm and further showed the fixed-parameter tractability of a dual parameterization of this problem.

A second large body of work can be found in the antithetical class of problems, where we ask to *delete* edges from a network while *preserving* connectivity. Probably the most studied member of these *connectivity preservation problems* is the Minimum Strong Spanning Spanning Subgraph (MSSS) problem: given a strongly connected digraph we are asked to find a strongly connected subgraph with a minimum number of arcs. The problem is NP-complete (an easy reduction from the Hamiltonian Cycle problem) and there exist a number of approximation algorithms for it (see the monograph by Bang-Jensen and Gutin for details and references [2]). Bang-Jensen and Yeo [8] were the first to study MSSS from the parameterized complexity perspective. They presented an algorithm that runs in time $2^{O(k \log k)} n^{O(1)}$ and decides whether a given strongly connected digraph D on n vertices and m arcs has a strongly connected subgraph with at most $m - k$ arcs provided $m \geqslant 2n - 2$. Bang-Jensen et al. [1] extended this result not only to arbitrary number m of arcs but also to λ-arc-strong connectivity for an arbitrary integer λ, and they further extended it to

λ-edge-connected *undirected* graphs.

We consider the undirected variant of this problem, however, we aim to preserve the *vertex-connectivity* instead of edge-connectivity. As noted by Marx and Végh [6], vertex-connectivity variants of parameterized connectivity problems seem to be much harder to approach than their edge-connectivity counterparts.[2] Moreover, even the complexity of the problem of augmenting the vertex-connectivity of an undirected graph from 2 to 3, via a minimum set of up to k new links remains open [6]. Our first main result in this direction is the first FPT algorithm for the following problem[3]:

WEIGHTED BICONNECTIVITY DELETION parametrised by k

Input: A biconnected graph G, $k \in \mathbb{N}$, $w^* \in \mathbb{R}_{\geqslant 0}$ and a function $w : E(G) \to \mathbb{R}_{\geqslant 0}$.
Problem: Is there a set $S \subseteq E(G)$ of size at most k such that $G - S$ is biconnected and $w(S) \geqslant w^*$?

Theorem 1.1. WEIGHTED BICONNECTIVITY DELETION *can be solved in time* $2^{O(k \log k)} n^{O(1)}$.

We further show that this problem has a randomized polynomial kernelization when the edges are required to have only unit weights. To be precise, all inputs for the unweighted variant UNWEIGHTED BICONNECTIVITY DELETION are of the form (G, k, w^*, w), where $w^* = k$ and $w(e) = 1$ for every $e \in E(G)$.

Theorem 1.2. UNWEIGHTED BICONNECTIVITY DELETION *has a randomized kernel with* $O(k^9)$ *vertices.*

We then study a more general version of the (unweighted) problem and obtain a proof of non-uniform fixed-parameter tractability of this problem based on a recent meta-theorem of Lokshtanov et al. [13]. This is our second main tractability result.

VERTEX-CONNECTIVITY DELETION parametrised by k

Input: A undirected graph G, integers ρ, k such that G is ρ-vertex connected.
Problem: Is there a set $S \subseteq E(G)$ of size at most k such that $G - S$ is ρ-vertex connected?

Theorem 1.3. *For every fixed k and ρ, the* VERTEX-CONNECTIVITY DELETION *problem can be solved in time* $O(n^6)$.

Along with arc-additions and arc-deletions, a third interesting operation on digraphs is the *path-contraction* operation which has been used to obtain structural results on paths in digraphs [2]. To *path-contract* an arc (x, y) in a digraph D, we remove it from D, identify

[2]Marx and Végh [6] compare [9] and [10] to [11] and [12] with respect to polynomial-time exact and approximation algorithms.

[3]Note that since 1-vertex-connectivity is trivially equivalent to 1-edge-connectivity, the 1-vertex-connectivity case was proved to be FPT by Bang-Jensen *et al.* [1].

x and y and keep the in-arcs of x and the our-arcs of y for the combined vertex. The resulting digraph is denoted by $D /\!\!/ (x, y)$. It is useful to extend this notation to sequences of contractions: let $S = (a_1, a_2, \dots, a_p)$ be a sequence of arcs of a digraph D. Then $D /\!\!/ S$ is defined as $(\dots((D /\!\!/ a_1) /\!\!/ a_2) /\!\!/ \dots) /\!\!/ a_p$. Since the resulting digraph does not depend on the order of the arcs [2], this notation can equivalently be used for arc-sets.

Bang-Jensen and Yeo [8] asked whether the problem of path- contracting at least k arcs to maintain strong connectivity of a given digraph D is fixed-parameter tractable. Formally, the problem is stated as follows:

PATH-CONTRACTION PRESERVING STRONG CONNECTIVITY parametrised by k

Input: A strongly connected digraph D and an integer k.
Problem: Is there a sequence $S = (a_1, \dots, a_k)$ of arcs of D such that $D /\!\!/ S$ is also strongly connected?

Our first result is a negative answer to the question of Bang-Jensen and Yeo. That is, we show that this problem is unlikely to be FPT.

Theorem 1.4. PATH-CONTRACTION PRESERVING STRONG CONNECTIVITY *is* W[1]*-hard.*

We follow up this result by considering a natural vertex-deletion variant of the problem and extending our W[1]-hardness result to this problem as well. In this variant, the objective is to check for the existence of a set of *exactly* k vertices such that on deleting these vertices from the given digraph, the digraph stays strongly connected.

Theorem 1.5. VERTEX-DELETION PRESERVING STRONG CONNECTIVITY *is* W[1]*-hard.*

Our Methodology. In particular the p-λ-EDGE CONNECTED SUBGRAPH (p-λ-ECS) problem where the objective is to delete k edges while keeping the graph λ-edge connected. Call an edge *deletable* (we refer to it as *non-critical* in the case of vertex-connectivity) if deleting it keeps the given (di)graph λ-edge connected, *undeletable* (*critical*) otherwise, and call an edge *irrelevant* if there is a solution disjoint from the edge.

For an even value of λ and a λ-edge-connected undirected graph G, Bang-Jensen *et al.* [1] proved that unless the total number of deletable edges is bounded by $O(\lambda k^2)$, it is possible in polynomial time to obtain a set F of k edges such that $G - F$ is still λ-edge-connected. This result does not hold for odd values of λ as can be seen, *e.g.*, when $\lambda = 1$ and G is a cycle. In this much more involved case, unless the total number of deletable edges is bounded by $O(\lambda k^3)$, it is possible in polynomial time to obtain either a set F of k edges such that $G - F$ is still λ-edge-connected or to identify an irrelevant edge.

WEIGHTED BICONNECTIVITY DELETION is similar to the case of odd λ as we find either a solution or an irrelevant edge. The main difference between our FPT algorithm and the one presented by Bang-Jensen *et al.* is the deep structural analysis necessitated by the shift from edge-connectivity to vertex-connectivity in *undirected graphs*: While in the former case the failure to find a solution means that G can be decomposed into a 'cycle-like' structure, in our case no such simple structure arises. Instead, we perform a careful examination of mixed cuts in the graph, each of which comprise precisely of one critical edge e and a vertex w which we call the *partner* of e. We show that either a large number of critical edges share a common partner or there is a large number of critical edges with pairwise distinct partners. In the former case, we proof the existence of an irrelevant edge while in the latter case we are able to construct a solution. Our result is based on a non-trivial combination of several new structural properties of biconnected graphs and critical edges which we believe is of independent interest and useful in the study of other connectivity-constrained problems.

The kernel stated in Theorem 1.2 relies on the powerful *cut-covering lemma* of Kratsch and Wahlström [14] which has been central to the development of several recent kernelization algorithms [15]. While Bang-Jensen et al. obtained a randomized compression for the p-λ-ECS problem using sketching techniques from dynamic graph algorithms, we provide an alternative approach and show that when dealing with undirected biconnectivity it is also possible to obtain a (randomized) polynomial *kernel*. We believe that this approach could be applicable for higher values of vertex- connectivity and for other connectivity deletion problems, as long as one is able to bound the number of critical or undeletable edges in the given instance by an appropriate function of the parameter.

Further related work. In the Minimum Equivalent Digraph problem, given a digraph D, the aim is to find a spanning subgraph H of D with minimum number of arcs such that if there is an x-y directed path in D then there is such a path in H for every pair x, y of vertices of D. Since it is not hard to solve Minimum Equivalent Digraph for acyclic digraphs, Minimum Equivalent Digraph for general digraphs can be reduced to MSSS in polynomial time. Chapter 12 of the monograph of Bang- Jensen and Gutin [2] surveys pre-2009 results on Minimum Equivalent Digraph. The first exact algorithm for the Mnimum Equivalent Digraph problem, running in time $2^{O(m)}$, was given by Moyles and Thompson [16] in 1969, where m is the number of arcs in the graph. More recently, Fomin, Lokshtanov, and Saurabh [17] gave the first vertex-exponential algorithm for this problem, *i.e.* an algorithm with a running time of $2^{O(n)}$.

2. Preliminaries

Graphs. For an undirected graph G and vertex set $S \subseteq V(G)$, we denote by $E(S)$ the set of edges of G with both endpoints in S. For a pair of disjoint vertex sets $X, Y \subseteq V(G)$, we denote by $E(X, Y)$ the set of edges with one endpoint in X and the other in Y. For a vertex set $X \subseteq V(G)$, we denote by $N_G(X)$ the set of vertices of $V(G) \setminus X$ which are adjacent to a vertex in X. We denote by $\delta_G(X)$ the set $E(X, V(G) \setminus X)$. A vertex in a connected undirected graph is a *cut-vertex* if deleting this vertex disconnects the graph. A *biconnected graph* is a connected graph on two or more vertices having no cut-vertices.

For a directed or undirected path P, we denote by $V(P)$ and $E(P)$ the set of vertices and edges in P, respectively. We further denote by $V_{\mathsf{int}}(P)$ the set of internal vertices of P.

We say that two paths P_1 and P_2 are *internally vertex-disjoint* if $V_{\mathsf{int}}(P_1) \cap V_{\mathsf{int}}(P_2) = \emptyset$. Note that under this definition, a path consisting of a single vertex is internally vertex-disjoint to any other path.

For two internally vertex-disjoint paths $P_1 = v_1, \dots v_t$ and $P_2 = w_1, \dots, w_q$ such that $v_1 \neq w_1$ and $v_t = w_1$, we denote by $P_1 + P_2$ the concatenated path $v_1, \dots, v_{t-1}, v_t, w_2, \dots, w_q$. When we deal with undirected graphs, we will abuse this notation and also use $P_1 + P_2$ to refer to the concatenated path that arises when $v_1 = w_1$ and $v_t \neq w_q$ or $v_1 = w_q$ and

$w_1 \neq v_t$ or $w_1 = v_t$ and $v_1 \neq w_q$. In short, the two 'orientations' of any undirected path are used interchangeably and when we need to differentiate between the two orientations, we explicitly say that we are *traversing* the path from one specified endpoint to the other.

Definition 2.1. *Let G be a graph and $x, y \in V(G)$ two vertices. An x-y* separator *(an x-y* cut*) is a set $S \subseteq V(G) \setminus \{x, y\}$ (respectively $S \subseteq E(G)$) such that there is no x-y path in $G - S$. A* mixed x-y cut *is a set $S \subseteq V(G) \cup E(G)$ such that $|S \cap E(G)| = 1$ and there is no x-y path in $G - S$.*

Let $S \subseteq V(G) \cup E(G)$. We denote by $R_G(x, S)$ the set of vertices in the same connected component as x in the graph $G - S$. The reference to G is dropped if it is clear from the context.

Definition 2.2. *Let G be a graph and $x, y \in V(G)$. Let $\mathcal{P}$ be a set of internally vertex-disjoint x-y paths in G. Then, we call $\mathcal{P}$ an x-y* flow*. The* value *of this flow is $|\mathcal{P}|$. We say that an edge e* participates *in the x-y flow $\mathcal{P}$ if $e \in \bigcup_{P \in \mathcal{P}} P$.*

We denote by $\kappa_G(x, y)$ the value of the maximum x-y flow in G with the reference to G dropped when clear from the context.

Recall that Menger's theorem states that for distinct *non-adjacent* vertices x and y, the size of the smallest x-y separator is precisely $\kappa(x, y)$. We extend the definition of flows to vertex sets as follows. Let $x \in V(G)$ and $Y \subseteq V(G)$ be such that $x \notin Y$. Let $\mathcal{P}$ be a set of paths in G which have an endpoint in Y and intersect only in x. Then, we refer to $\mathcal{P}$ as an x-Y flow, with the value of this flow defined as $|\mathcal{P}|$.

Directed graphs. We will refer to edges in a digraph as *arcs*. For a vertex x in a digraph D we write $N^-_D(x)$ and $N^+_D(x)$ to denote its in- and out-neighbours, respectively. A *sink* is a vertex with no out-neighbours and a *source* is a vertex with no in-neighbours. While we will use path-contraction in digraphs only for single arcs, *i.e.* directed paths of length one, we restate the more general definition for context.

Definition 2.3 (Bang-Jensen and Gutin [2])**.** *Let P be an (x, y)-path in a directed multigraph D. Then, $D \mathbin{/\!\!/} P$ denotes the multigraph obtained from D by deleting all vertices of P and adding a new vertex z such that every arc with head x (tail y) and tail (respectively head) in $V \setminus V(P)$ becomes an arc with head (tail) z and the same tail (respectively head).*

The path-contraction of a single arc (x, y) is equivalent to identifying the vertices x and y as a new vertex z and then removing the resulting loop as well as all arcs from z to $N^+(x)$ and $N^-(y)$.

Parameterized Complexity. An instance of a parameterized problem Π is a pair (I, k) where I is the *main part* and k is the *parameter*; the latter is usually a non-negative integer. A parameterized problem is *fixed-parameter tractable* if there exists a computable function f such that instances (I, k) can be solved in time $O(f(k)|I|^c)$ where $|I|$ denotes the size of I. The class of all fixed-parameter tractable decision problems is called FPT and algorithms which run in the time specified above are called FPT algorithms.

To establish that a problem under a specific parameterization is not in FPT (under common complexity-theoretic assumptions) we provide *parameter-preserving reductions* from problems known to lie in intractable classes like $\mathsf{W}[1]$ or $\mathsf{W}[2]$. In such a reduction, an instance (I_1, k_1) is reduced in polynomial time to an instance (I_2, k_2) where $k_2 \leqslant f(k_1)$ for some function f. In the context of this paper we will use that Independent Set under its natural parameterization (the size of the independent set) is $\mathsf{W}[1]$-hard [18].

A *reduction rule* for a parameterized problem Π is an algorithm that given an instance (I, k) of a problem Π returns an instance (I', k') of the *same* problem. The reduction rule is said to be *sound* if it holds that $(I, k) \in \Pi$ if and only if $(I', k') \in \Pi$. A *kernelization* is a polynomial-time algorithm that given any instance (I, k) returns an instance (I', k') such that $(I, k) \in \Pi$ if and only if $(I', k') \in \Pi$ *and* $|I'| + k' \leqslant f(k)$ for some computable function f. The function f is called the *size* of the kernelization, and we have a polynomial kernelization if $f(k)$ is polynomially bounded in k. A *randomized kernelization* is an algorithm which is allowed to err with certain probability. That is, the returned instance will be equivalent to the input instance only with a certain probability.

2.1. Monadic Second Order Logic

The syntax of Monadic Second Order Logic (MSOL) of graphs includes the logical connectives $\vee$, $\wedge$, $\neg$, $\Leftrightarrow$, $\Rightarrow$, variables for vertices, edges, sets of vertices and sets of edges, the quantifiers $\forall$ and $\exists$, which can be applied to these variables, and five binary relations:

1. $u \in U$, where u is a vertex variable and U is a vertex set variable;
2. $d \in D$, where d is an edge variable and D is an edge set variable;
3. $\mathbf{inc}(d, u)$, where d is an edge variable, u is a vertex variable, and the interpretation is that the edge d is incident to u;
4. $\mathbf{adj}(u, v)$, where u and v are vertex variables, and the interpretation is that u and v are adjacent;
5. equality of variables representing vertices, edges, vertex sets and edge sets.

Counting Monadic Second Order Logic (CMSOL) extends MSOL by including atomic sentences testing whether the cardinality of a set is equal to q modulo r, where q and r are integers such that $0 \leqslant q < r$ and $r \geqslant 2$. That is, CMSOL is MSOL with the following atomic sentence: $\mathbf{card}_{q,r}(S) = \mathbf{true}$ if and only if $|S| \equiv q \pmod r$, where S is a set. We refer to [19, 20, 21] for a detailed introduction to CMSOL.

2.2. Unbreakable graphs

We will require the following definitions and results from [13].

Definition 2.4. [**Separation**] *A pair (X, Y) where $X \cup Y = V(G)$ is a* separation *if $E(X \setminus Y, Y \setminus X) = \emptyset$. The* order *of (X, Y) is $|X \cap Y|$. The* separator *of the separation (X, Y) is the set $X \cap Y$.*

Roughly speaking, a graph is unbreakable if it is not possible to "break" it into two large parts by removing only a small number of vertices. The formal definition follows.

Definition 2.5. [(s,c)-**unbreakable graph**] *Let G be a graph. If there exists a separation (X, Y) of order at most c such that $|X \setminus Y| > s$ and $|Y \setminus X| > s$, called an* (s, c)-witnessing separation*, then G is* (s, c)-breakable*. Otherwise, G is* (s, c)-unbreakable.

Finally, we need the following result which reduces the design of *non-uniform* FPT algorithms for problems expressible in CMSO to designing FPT algorithms for the *same* problems on a class of sufficiently unbreakable graphs.

Proposition 2.6. [13] *Let ψ be a CMSOL sentence. For all $c \in \mathbb{N}$, there exists $s \in \mathbb{N}$ such that if there exists an algorithm that solves* CMSOL$[\psi]$ *on (s, c)-unbreakable graphs in time $O(n^d)$ for some $d > 4$, then there exists an algorithm that solves* CMSOL$[\psi]$ *on general graphs in time $O(n^d)$.*

3. Preserving strong connectivity

In this section, we prove Theorem 1.4 and Theorem 1.5.

Theorem 1.4. Path-contraction Preserving Strong Connectivity *is* W[1]*-hard.*

Proof. We reduce Independent Set to Path-contraction Preserving Strong Connectivity.

Construction. Let (G, k) be an instance of Independent Set. We now define a digraph D as follows. We begin with the vertex set of D. For every vertex $v \in V(G)$, D has two vertices v^-, v^+. For every edge $e = (u, v) \in E(G)$, the digraph D has $k + 2$ vertices $\hat{e}, \hat{e}_1, \dots, \hat{e}_{k+1}$. Finally, there are $2k + 4$ special vertices $x, y, x^1, \dots, x^{k+1}, y^1, \dots, y^{k+1}$. This completes the definition of $V(D)$. We now define the arc set of D (see Figure 1).

- For every $v \in V(G)$, we add the arc (v^-, v^+) in D .
- For every $i \in [k + 1]$, we add the arcs $\{(x, x^i), (x^i, x), (y, y^i), (y^i, y), (y, x)\}$.
- For every edge $e = (u, v) \in E(G)$ and $i \in [k+1]$, we add the arcs $\{(\hat{e}, \hat{e}_i), (\hat{e}_i, \hat{e}), (v^-, \hat{e}), (\hat{e}, v^+), (u^-, \hat{e}), (\hat{e}, u^+)\}$ in D .
- For every $v \in V(G)$, we add the arc (x, v^-) and the arc (v^+, y).

This completes the construction of the digraph D. Clearly, D is strongly-connected.

For an edge $e = (u, v) \in E(G)$, we denote by $\mathcal{B}_e$ the set of arcs $\{(v^-, \hat{e}), (\hat{e}, v^+), (u^-, \hat{e}), (\hat{e}, u^+)\}$ and by $\mathcal{F}_e$, the set of arcs $\mathcal{B}_e \cup \{(\hat{e}, \hat{e}_i), (\hat{e}_i, \hat{e}) | i \in [k+1]\} \cup \{(u^-, u^+), (v^-, v^+), (x, v^-), (v^+, y), (x, u^-), (u^+, y), (y, x)\}$. We refer to the subgraph of D induced by $\mathcal{F}_e$ as the *edge-selection gadget* in D corresponding to e (see Figure 1). The intuition here is that, as we will prove formally, any solution in D will contain at most one of the two arcs $(u^-, u^+), (v^-, v^+)$.

Proof of correctness. We now argue that (G, k) is a yes-instance of Independent Set if and only if (D, k) is a yes-instance of Path-contraction Preserving Strong Connectivity. In the forward direction, suppose that (G, k) is a yes-instance of Independent Set and let $X \subseteq V(G)$ be a solution. Observe that $S = \{(v^-, v^+) \mid v \in X\}$ is a pairwise vertex-disjoint set of arcs. We claim that S is a solution for the instance (D, k). That is, $|S| \geqslant k$ and $D \mathbin{/\!\!/} S$ is strongly connected. The former is true by definition. We now argue the latter.

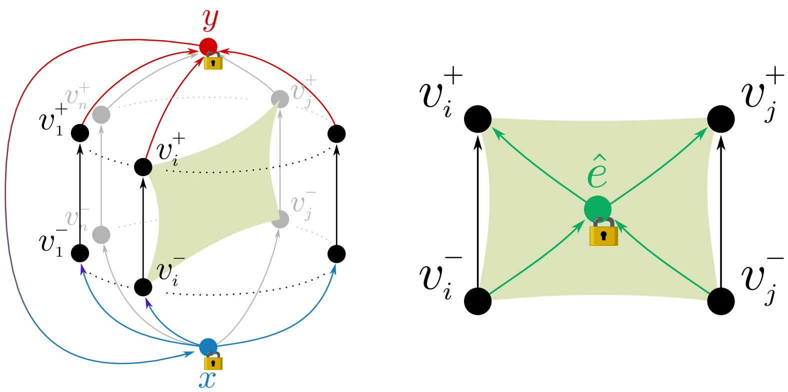

Figure 1: An illustration of the arcs in the reduced instance of Path-contraction Preserving Strong Connectivity. The second figure only contains the arcs of the edge-selection gadget corresponding to the edge $e = (v_i, v_j) \in E(G)$. Vertices with a padlock have additional $k+1$ pendant vertices with arcs in both directions.

Claim 3.1. *$D' = D \mathbin{/\!\!/} S$ is strongly connected.*

Proof. Observe that it is sufficient to prove that $D'' = D - Q$ is strongly connected, where

$$Q = \bigcup_{v \in X} (N^+(v^-) \cup N^-(v^+)) \setminus \{(v^-, v^+)\},$$

and $N^+(v)$ and $N^-(v)$ are the sets of out-neighbours and in-neighbours of v. In other words, for every arc $(v^-, v^+) \in S$, Q contains all the arcs that are lost when we path-contract this arc. We begin by observing that the set Q is disjoint from $\{(v^-, v^+) \mid v \in V(G)\}$. This follows from the definition of Q. Due to this observation and the presence of the arc (y, x), it follows that the vertices in

$$\mathcal{P} = \{(v^-, v^+) \mid v \in V(G)\} \cup \{x, y, x^1, \ldots, x^{k+1}, y^1, \ldots, y^{k+1}\}$$

occur in a single strongly connected component of D''. Hence, it suffices to argue that for every $e \in E(G)$, the vertex $\hat{e}$ is also in the same strongly connected component of D''.

Note that since X is an independent set in G, it must be the case that for any $e = (u, v) \in E(G)$, either $(u^-, u^+) \notin S$ or $(v^-, v^+) \notin S$. But this implies that either $\{(u^-, \hat{e}), (\hat{e}, u^+)\} \cap Q = \emptyset$ or $\{(v^-, \hat{e}), (\hat{e}, v^+)\} \cap Q = \emptyset$, implying that $\hat{e}$ is also in the same strongly connected component as the vertices in $\mathcal{P}$. Thus, D'' is strongly connected and so is D'. This completes the proof of the claim and hence proves the correctness of the forward direction of the reduction. □

We now consider the converse direction. Suppose that (D, k) is a yes-instance of Path-contraction Preserving Strong Connectivity and let $S = \{a_1, \ldots, a_k\}$ be a solution for this instance. We require the following claim.

Claim 3.2. *For every edge $e = (u, v) \in E(G)$, $|S \cap \{(u^-, u^+), (v^-, v^+)\}| \leqslant 1$. Furthermore, $S \subseteq \{(v^-, v^+) \mid v \in V(G)\}$.*

Proof. For the first statement, suppose to the contrary that S contains both the arcs (u^-, u^+) and (v^-, v^+) for some $e = (u, v) \in E(G)$. Then, observe that in the graph $D /\!\!/ S$, the arcs in the set $\mathcal{B}_e$ are absent. Since $\mathcal{B}_e$ contains all arcs incident to $\hat{e}$ except the ones incident to $\hat{e}_i$ for $i \in [k+1]$, this disconnects the undirected graph underlying $D /\!\!/ S$, implying that S is not a solution, a contradiction.

For the second statement, we argue that no arc incident to x, y or $\{\hat{e} \mid e \in E(G)\}$ can be in S. Suppose to the contrary that for some $i \in [k+1]$, the arc $(x, x^i) \in S$. Then, the arcs from x to $\{v^- \mid v \in V(G)\}$ are all absent from D', implying that D' is not strongly connected, a contradiction. On the other hand, if for some $v \in V(G)$, we path-contract the arc (x, v^-), the arc from x to x^i is absent in $D /\!\!/ S$ for every $i \in [k+1]$. Since $|S| \leqslant k$, there is at least one $i \in [k+1]$ such that the arc (x, x^i) is not in S. Since the arc (x, x^i) is absent from $D /\!\!/ S$, it follows that it is not strongly-connected, a contradiction. Finally, if S contains the arc (y, x), the arc (x^i, x) is not in $D /\!\!/ S$ for any $i \in [k+1]$, implying that it is not strongly-connected for the same reason as that in the previous case. Hence, we conclude that no arc incident on x is in S. The argument for y is analogous and hence we do not address it explicitly.

Suppose that for some $e = (u, v) \in E(G)$ and $i \in [k+1]$, there is an arc in $\{(\hat{e}, \hat{e}_i), (\hat{e}_i, \hat{e})\}$ which is in S. Observe that in the former case, the arcs $(\hat{e}, u^+)$ and $(\hat{e}, v^+)$ are absent in $D /\!\!/ S$, implying that the new vertex is a sink, a contradiction. In the latter case, the new vertex is a source, a contradiction. Now, suppose that S contains an arc in $\mathcal{B}_e$. Then, for some $i \in [k+1]$, the vertex $\hat{e}_i$ is left as a source or sink in $D /\!\!/ S$, a contradiction. This completes the proof of the claim. □

The claim above implies that if X is a solution for the reduced instance of Path-contraction Preserving Strong Connectivity, then the set S of arcs corresponds independent set in G. In other words, (G, k) is a yes-instance of Independent Set. This proves the correctness of the reduction and completes the proof of the theorem. □

We can prove a similar result for the Vertex-deletion Preserving Strong Connectivity problem. The problem is formally defined as follows.

Vertex-deletion Preserving Strong Connectivity parametrised by k

Input: A strongly connected digraph D and an integer k.
Problem: Is there a vertex set S of size (exactly) k such that the graph $D - S$ is strongly connected?

We have to require "exactly k" rather than "at least k" since otherwise we could delete all but one vertices of D and get a trivially strongly connected digraph.

Theorem 1.5. Vertex-deletion Preserving Strong Connectivity *is* W[1]*-hard.*

Proof. We will again use a reduction from Independent Set. Let G be a graph, an input of Independent Set with parameter k. We first reduce Independent Set to Vertex-deletion Preserving Connectivity with Undeletable Vertices: Given a connected graph H with some vertices marked and parameter k, is there k unmarked vertices in H whose deletion keeps H connected? To construct H, start from G with all vertices unmarked. Subdivide every edge of G with a marked vertex. Add another marked vertex x with edges to all unmarked vertices. It is easy to see that the reduction is correct since deleting two unmarked vertices in H which are adjacent in G leaves the corresponding subdivision vertex isolated.

Now we reduce Vertex-deletion Preserving Connectivity with Undeletable Vertices to VERTEX-DELETION PRESERVING STRONG CONNECTIVITY. Replace every edge uv of H by arcs uv and vu, unmark every marked vertex w of H and replace it by a directed cycle of length $k+2$ containing w (all other vertices of the cycle are new). Denote the resulting digraph by D; note that it is strongly connected. To see the correctness, it suffices to observe that we cannot delete less than $k+1$ vertices of any directed cycle of length $k+2$ and keep D strongly connected. This completes the proof of the theorem. □

4. Edge deletion to biconnected graphs

In this section, we present our FPT algorithm for the WEIGHTED BICONNECTIVITY DELETION problem on undirected graphs. Recall that the problem is defined as follows:

WEIGHTED BICONNECTIVITY DELETION parametrised by k

Input: A biconnected graph G, $k \in \mathbb{N}$, $w^* \in \mathbb{R}_{\geqslant 0}$ and a function $w : E(G) \to \mathbb{R}_{\geqslant 0}$.
Problem: Is there a set $S \subseteq E(G)$ of size at most k such that $G - S$ is biconnected and $w(S) \geqslant w^*$?

We refer to a set $S \subseteq E(G)$ such that $G - S$ is biconnected as a *biconnectivity deletion set* of G. For an instance (G, k, w^*, w) of WEIGHTED BICONNECTIVITY DELETION and a biconnectivity deletion set S of G, we say that S is a *solution* if $|S| \leqslant k$ and $w(S) \geqslant w^*$. The main result of this section is the following.

Theorem 1.1. WEIGHTED BICONNECTIVITY DELETION *can be solved in time* $2^{O(k \log k)} n^{O(1)}$.

We will first make a short digression in order to define the notion of critical edges and list certain structural properties that will be required in this and the following section.

4.1. Properties of critical edges

Definition 4.1. *We denote by $\kappa(G)$ the vertex-connectivity of a graph G. Let G be a ρ-vertex connected graph. An edge $e \in E(G)$ is called* ρ-critical *if $\kappa(G - e) < \rho$. We denote by $\mathsf{Critical}^\rho_\mathsf{G}(\mathsf{e})$ the subset of $E(G)$ comprising edges which are ρ-critical in $G - e$ but not in G. We denote by $\mathsf{Critical}^\rho_G(\emptyset)$ the set of edges which are already ρ-critical in G. In all notations, we ignore the explicit reference to G and ρ when these are clear from the context. We say that e is* ρ-critical for *a pair of vertices u, v in G if u and v are non-adjacent and e participates in every u-v flow of value ρ in G.*

The following lemma gives a useful structural characterization of edges which become ρ-critical upon the deletion of a particular edge of the graph.

Lemma 4.2. *Let G be a ρ-vertex connected graph. Let $e = (x, y)$ and $e' = (u, v)$ be distinct non-critical edges,* i.e. *they are not in $\mathsf{Critical}^\rho_G(\emptyset)$. Then the following are equivalent:*

1. $e' \in \mathsf{Critical}^\rho_G(e)$.

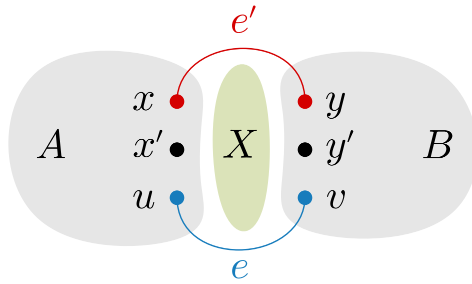

Figure 2: An illustration of the edges e, e' and the sets A and B in the proof of Lemma 4.2. Observe that there are no more edges with one endpoint each in A and B.

2. *There is a pair of vertices* $x', y' \in V(G)$ *and a mixed* x'-y' *cut* X *of size* ρ *in* $G - e$*, where* $e' \in X$.

3. e' *participates in every* x-y *flow of value* ρ *in* $G - e$.

Proof. We prove that (1) $\Rightarrow$ (2), (2) $\Rightarrow$ (3) and (3) $\Rightarrow$ (1). Consider the first implication. Since $e' \in \mathsf{Critical}_G^\rho(e)$, it must be the case that there are vertices x', y' such that e' is ρ-critical for the pair x', y'. That is, x' is non-adjacent to y' and e' is ρ-critical for the pair x', y' in $G' = G - e$. That is, e' participates in every x'-y' flow of value ρ in $G - e$. As a result, the maximum value of any x'-y' flow in $G - e - e'$ is precisely $\rho - 1$. By Menger's theorem, this implies the presence of a set $S \subseteq V(G)$ of size $\rho - 1$ which intersects all x'-y' paths in $G - e - e'$. Setting $X = S \cup \{e'\}$ completes the argument for the first implication.

Consider the second implication. Let $A = R_{G-e}(x', X)$ and let $B = V(G) \setminus (A \cup S)$ (see Figure 2), where $S = X \setminus \{e'\}$. Observe that if $u, v \in A$ or $u, v \in B$, then $X \setminus \{e'\}$ would be an x'-y' separator of size $\rho - 1$ in $G - e$, a contradiction to $G - e$ being ρ-vertex connected. Hence, it must be the case that either $u \in A$ and $v \in B$ or vice-versa. We assume without loss of generality that $u \in A$ and $v \in B$.

By a similar argument, since e' is not ρ-critical in G but is ρ-critical in $G - e$, it must be the case that the edge e also has one endpoint in A and one endpoint in B. Again, we assume without loss of generality that $x \in A$ and $y \in B$. But now, observe that in the graph $G - e - S$, every x-y path contains the edge e'. Since $|S| = \rho - 1$ and $G - e$ is ρ-connected, we conclude that e' participates in every x-y flow of value ρ in $G - e$. This completes the argument for the second implication.

For the final implication, observe that while $G - e$ is ρ-connected, the fact that e' participates in every x-y flow of value ρ in $G - e$ implies that e' is ρ-critical in $G - e$. Since e' is by definition, not ρ-critical in G, we conclude that $e' \in \mathsf{Critical}_G^\rho(e)$. This completes the proof of the lemma. $\square$

4.2. The FPT *algorithm for* Weighted Biconnectivity Deletion

In this section, we will prove Theorem 1.1 by giving an algorithm for a more general version of the Weighted Biconnectivity Deletion problem where the input also includes a set $E^\infty \subseteq E(G)$ and the objective is to decide whether there is a solution disjoint from this set. Henceforth, instances of Weighted Biconnectivity Deletion will be of the form (G, k, w^*, w, E^∞) and any solution S is required to be disjoint from E^∞. We will refer to edges of $E(G) \setminus E^\infty$ as *potential solution edges*. We say that a potential solution edge is *irrelevant* if either the instance has no solution, or has a solution that does not contain e. For an instance $I = (G, k, w^*, w, E^\infty)$ and $r \in \mathbb{N}$, we denote by $\mathsf{Heavy}_I(r)$ the *heaviest* r potential solution edges of G with respect to the function w. While this set is not necessarily unique (if multiple edges have the same weight, i.e., the same image under the function w), we will define $\mathsf{Heavy}_I(r)$ as the first r edges of *a fixed arbitrarily chosen ordering* of the edges of G in non-increasing order of their weights. If I is clear from the context, we simply write $\mathsf{Heavy}(r)$ when referring to $\mathsf{Heavy}_I(r)$.

Since we will only be dealing with biconnected graphs in this section, we will also drop the explicit reference to ρ in the notations from Definition 4.1. For instance, when we say that an edge is critical (non-critical), we imply that it is 2-critical (not 2-critical respectively). Observe that no edge from the set $\mathsf{Critical}_G(\emptyset)$ can be part of a solution. As a result, we assume without loss of generality that for any instance (G, k, w^*, w, E^∞), the set $\mathsf{Critical}_G(\emptyset)$ is contained in E^∞. Furthermore, since the edges in E^∞ can never be part of a solution, we assume without loss of generality that for every edge $e \in E^\infty$, $w(e) = 0$. The proof of Theorem 1.1 is based on the following lemma which states that either a) the number of potential solution edges in the instance is already bounded polynomially in k, or b) a 'small' set of the heaviest edges in the instance must intersect a solution, or c) there is an irrelevant edge which can be found in polynomial time. For ease of presentation, let use define the polynomial $\mu(x) := 20x^3 + 46x^2 + x$ for the rest of this section.

Lemma 4.3. *Let $I = (G, k, w^*, w, E^\infty)$ be an instance of* Weighted Biconnectivity Deletion*. If $|E(G) \setminus E^\infty| > \mu(k)$, then the set $\mathsf{Heavy}(\mu(k))$ contains either a solution edge or an irrelevant edge which can be computed in polynomial time.*

Given Lemma 4.3, Theorem 1.1 is proved as follows. Let $I = (G, k, w, w^*, E^\infty)$ be an instance of Weighted Biconnectivity Deletion. If the number of potential solution edges in this instance is already bounded by $\mu(k)$, then we simply enumerate all k-sized subsets of this set (there are $2^{O(k \log k)}$ choices) and check in polynomial time whether one of these subsets is a solution. Otherwise, we invoke Lemma 4.3 and either correctly conclude that the set $\mathsf{Heavy}(\mu(k))$ contains a solution edge, or we compute an irrelevant edge e in polynomial time. In the first case we branch on the set $\mathsf{Heavy}(\mu(k))$, reduce the budget k by 1 and the target weight w^* accordingly and recursively solve the resulting instance. In the second case, we add the edge e to the set E^∞ (thus decreasing the set of potential solution edges) and repeat.

Remark 4.4. *There is also an alternative strategy to the above, as follows. Let S be the set of all edges of weight at least w^*/k. Clearly S must be non-empty and any solution must intersect S. If $|S| \leqslant \mu(k)$, then we branch on S as above. Otherwise, we will be able to either find a biconnectivity deletion set $S' \subseteq S$ with $|S'| = k$ or an irrelevant edge in S as in Lemma 4.3. In the former case, S' is already a solution; in the latter case, we proceed according to the strategy above. Thus, this alternative strategy yields a slightly simpler proof, contains one less branching step and will be used in the kernelization algorithm in Subsection 4.3. On the other hand, the strategy above does not explicitly depend on w^*, and therefore always gives a maximum-weight solution. In either case, the main technical challenges in the* FPT *algorithm are exactly the same.*

The rest of this section is devoted to proving Lemma 4.3. In order to do so, we will present a greedy algorithm that runs in polynomial time and, assuming $|E(G) \setminus E^\infty| > \mu(k)$, will either produce a biconnectivity deletion set of size k contained strictly within $\mathsf{Heavy}(\mu(k))$, or it will identify an irrelevant edge. In the former case, we will argue that this implies that there is always a solution intersecting $\mathsf{Heavy}(\mu(k))$. More precisely, the algorithm will delete one potential solution edge from $\mathsf{Heavy}(\mu(k))$ at a time (while preserving biconnectivity), and will trace in each step the number of edges of $\mathsf{Heavy}(\mu(k))$ that become critical due to the removal of such an edge e, i.e., the size of the set $\mathsf{Critical}_{G'}(e) \cap \mathsf{Heavy}(\mu(k))$ where G' is the subgraph of G remaining after deleting the edges before e. We will then show that if $|\mathsf{Critical}_{G'}(e) \cap \mathsf{Heavy}(\mu(k))| \geqslant \frac{\mu(k)-k}{k}$, then G contains a special configuration from which we can either recover the required biconnectivity deletion set or identify an irrelevant edge.

4.2.1. Preliminary results

From now on, we assume that the given instance has more than $\mu(k)$ potential solution edges and begin by proving the following lemma which shows that if we find *some* biconnectivity deletion set of size k within $\mathsf{Heavy}(\mu(k))$, then there is a solution intersecting $\mathsf{Heavy}(\mu(k))$.

Lemma 4.5. *Let $I = (G, k, w^*, w, E^\infty)$ be an instance of* Weighted Biconnectivity Deletion *and let $S \subseteq \mathsf{Heavy}(\mu(k))$ be a biconnectivity deletion set of size k. If I is a yes-instance, then there is a solution for I intersecting the set $\mathsf{Heavy}(\mu(k))$.*

Proof. Suppose that this is not the case and let S' be a biconnectivity deletion set of size at most k such that $w(S') \geqslant w^*$. Note that S' is disjoint from $\mathsf{Heavy}(\mu(k))$ and S is contained in $\mathsf{Heavy}(\mu(k))$. Since $|S'| \leqslant |S|$, we infer that $w(S) \geqslant w(S')$, a contradiction to our assumption that there is no solution intersecting $\mathsf{Heavy}(\mu(k))$. This completes the proof of the lemma. □

Let $\hat{S} = \{f_1, \ldots, f_r\} \subseteq \mathsf{Heavy}(\mu(k))$ be a set greedily constructed as follows. The edge f_1 is the heaviest potential solution edge. That is, $w(f_1) \geqslant w(e)$ for every $e \in E(G) \setminus E^\infty$. For each $2 \leqslant i \leqslant r$, f_i is the heaviest edge of $\mathsf{Heavy}(\mu(k))$ which is *not* critical in $G - \{f_1, \ldots, f_{i-1}\}$. We terminate this procedure after k steps if we manage to find edges $\{f_1, \ldots, f_k\}$ or earlier if for some $r < k$, every edge of $\mathsf{Heavy}(\mu(k))$ is critical in $G - \{f_1, \ldots, f_r\}$.

Observe that by definition, $\hat{S}$ is a biconnectivity deletion set. Therefore, if $r = k$, then Lemma 4.5 implies that if there is a solution for the given instance, then there is one intersecting $\mathsf{Heavy}(\mu(k))$ (as required in Lemma 4.3). On the other hand, suppose that $r < k$. For each $i \in [r]$, we denote by $\hat{S}_i$, the set $\{f_1, \ldots, f_i\}$ and by $\hat{S}_0$, the empty set. Recall that we have already assumed that the number of potential solution edges is greater than $\mu(k) = 20k^3 + 46k^2 + k$. As a result, we have the following observation.

Observation 4.6. *There is an $i \in [r]$ such that $G - \hat{S}_i$ is biconnected and $|\mathsf{Critical}_{G-\hat{S}_{i-1}}(f_i) \cap \mathsf{Heavy}(\mu(k))| \geqslant \frac{\mu(k)-k}{k} = 20k^2 + 46k$.*

Let $i \in [r]$ be the index referred to in this observation. In the rest of the section, we let $\hat{F} = \hat{S}_{i-1}$, $e = f_i = (x, y)$ and $G' = G - \hat{F}$. The following observation is a straightforward consequence of Lemma 4.2 in our setting.

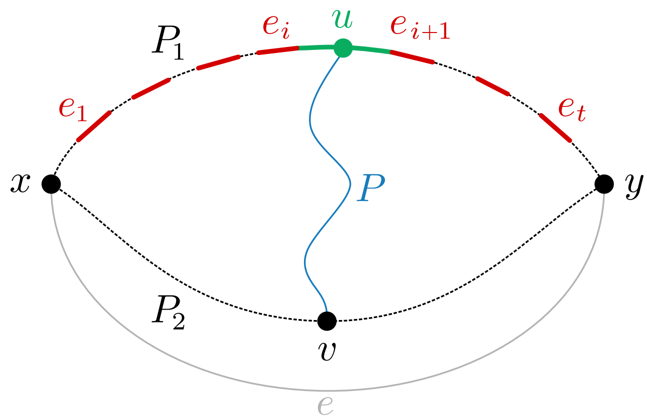

Figure 3: An illustration of $\mathsf{Segment}[i, i+1]$ (the subpath of P_1 between e_i and e_{i+1}) and P which is a nice path for this segment.

Observation 4.7. *Let G' and e be as above. Then, $\kappa_{G'-e}(x, y) = 2$ and for any x-y flow $\mathcal{H} = \{H_1, H_2\}$ of value 2 in $G' - e$, the following holds.*

1. *Every edge of $\mathsf{Critical}_{G'}(e)$ is critical for the pair x, y in $G' - e$ and hence lies on H_1 or H_2.*

2. *For every edge $e_1 \in \mathsf{Critical}_{G'}(e)$, say on H_1, there is at least one vertex v on H_2 such that $\{e_1, v\}$ is a mixed x-y cut in $G' - e$.*

We refer to the vertex v above as a *partner vertex* of e_1, and refer to the set of all partner vertices of e_1 as the *partner set* of e_1 and denote this set by $\mathsf{Partner}^e_{G'-e}(e_1)$. We do not explicitly refer to H_1 or H_2 in this notation because these will always be clear from the context. We will also drop the explicit reference to G' and e when these are clear from the context.

From Observation 4.6 and Obsevation 4.7, we now conclude the following:

Observation 4.8. *Let G' and e be as above. There is an x-y flow $\mathcal{P} = \{P_1, P_2\}$ of value 2 in $G' - e$ such that $|E(P_1) \cap \mathsf{Critical}_{G'}(e) \cap \mathsf{Heavy}(\mu(k))| \geqslant 10k^2 + 23k$.*

Henceforth, we work with this fixed x-y flow $\mathcal{P} = \{P_1, P_2\}$ in $G' - e$.

Definition 4.9. *Let G',$e = (x, y), P_1$ and P_2 be as above. Let $e_1, \ldots, e_t$ be some subset of $10k^2 + 23k$ edges in $\mathsf{Critical}_{G'}(e) \cap \mathsf{Heavy}(\mu(k)) \cap E(P_1)$ in the order in which they appear when traversing P_1 from x to y (see Figure 3), where $e_i = (u_i, v_i)$ and we may have $v_i = u_{i+1}$.*

- *For $i \in [t-1]$, we refer to the subpath of P_1 from v_i to u_{i+1} inclusively as $\mathsf{Segment}[i, i+1]$ of P_1 with the explicit reference to P_1 dropped when clear from the context.*

- *A path P with endpoints $u, v \in V(P_1) \cup V(P_2)$ but internally vertex-disjoint from P_1 and P_2 is said to be a* nice path *for $\mathsf{Segment}[i, i+1]$ if u is contained in $\mathsf{Segment}[i, i+1]$ and $v \in V_{\mathsf{int}}(P_2)$.*

When e, P_1, P_2 are as in the definition above, we write $w \leqslant w'$ for vertices $w, w' \in V(P_2)$ if either $w = w'$ or w is encountered before w' when traversing P_2 from x to y, and similarly for vertices on P_1. Furthermore, for a set $Q \subseteq V(P_2)$ and vertex $w \in V(P_2)$, we say that $w < Q$ ($w > Q$) if $w < q$ ($w > q$ respectively) for every $q \in Q$. We need the following crucial structural lemma regarding the structure of any path with endpoints in P_1 and P_2 but which is otherwise disjoint from these two paths.

Lemma 4.10. *Let $G', e = (x, y), P_1, P_2, e_1, \dots, e_t$ be as above. Let P be a path in G' with endpoints $u, v \in V(P_1) \cup V(P_2)$ but internally vertex-disjoint from P_1 and P_2. Then the following statements hold.*

1. *If $u, v \in V(P_1)$, then either $u, v \leqslant u_1$, or $u, v \geqslant v_t$, or there is a $j \in [t-1]$ such that u, v lie on* $\mathsf{Segment}[j, j+1]$.

2. *If $u, v \in V(P_2)$, then the subpath of P_2 from u to v is internally vertex-disjoint from the set $\bigcup_{i=1}^{t} \mathsf{Partner}^e_{G'}(e_i)$.*

3. *If $u \in V(P_1)$ and $v \in V(P_2)$, where u lies in $\mathsf{Segment}[i, i+1]$, then for every $j \leqslant i$ and every vertex $w \in \mathsf{Partner}(e_j)$ we have $w \leqslant v$, and for every $j \geqslant i+1$ and every vertex $w \in \mathsf{Partner}(e_j)$ we have $w \geqslant v$.*

4. *For every $i \in [t-1]$, there is at least one nice path for $\mathsf{Segment}[i, i+1]$, and for any $i \neq j \in [t-1]$, and paths P and P' which are nice paths for $\mathsf{Segment}[i, i+1]$ and $\mathsf{Segment}[j, j+1]$ respectively, P and P' are internally vertex-disjoint.*

Proof. For the first statement, suppose that there is an index $i \in [t]$ such that $u \leqslant u_i$ but $v \geqslant v_i$. Since e_i is critical in $G' - e$, Observation 4.7 implies that there is a mixed x-y cut $X = \{e_i, q\}$ where the vertex q lies on P_2. However, the graph induced on $V(P_1) \cup V(P)$ contains an x-y path disjoint from X, a contradiction. This completes the argument for the first statement.

The argument for the second statement is similar. Suppose to the contrary that there is an $i \in [t]$ and a vertex $w \in \mathsf{Partner}(e_i)$ such that the subpath of P_2 from u to v contains the vertex w. Recall that due to Observation 4.7, the set $X = \{e_i, w\}$ is a mixed x-y cut in $G' - e$. But then $u \in R_{G'-e}(x, X)$, $v \in R_{G'-e}(y, X)$, and the path P is disjoint from X, which contradicts X being an x-y cut.

For the third statement, suppose that there is a $j \leqslant i$ and $w \in \mathsf{Partner}(e_j)$ such that $w > v$. Let $X = \{e_j, w\}$. Due to Observation 4.7, we know that X is a mixed x-y cut in $G' - e$. Let $A = R_{G'-e}(x, X)$ and $B = R_{G'-e}(y, X)$. Since $u \in \mathsf{Segment}[i, i+1]$ and $j \leqslant i$, it follows that $u \in B$. Similarly, since $w > v$, it must be the case that $v \in A$. As above, we find that P is a path disjoint from X connecting A and B, which contradicts that X is an x-y cut. The argument for the case when there is a $j \geqslant i$ and $w \in \mathsf{Partner}(e_j)$ such that $w < v$ is analogous.

For the first part of the final statement, assume for a contradiction that for some $i \in [t-1]$, the path $\mathsf{Segment}[i, i+1]$ does not have a nice path. Recall that e_i is not critical in G' but is critical in $G' - e$. Therefore, there is a u_i-v_i flow of value 2 in the graph $G' - e_i$; let $\mathcal{H} = \{H_1, H_2\}$ be such a flow. If H_1 or H_2 intersects the internal vertices of P_2, then this implies the presence of a nice path for $\mathsf{Segment}[i, i+1]$. Hence, we assume that this does not happen. We also conclude that e must occur in H_1 or H_2. Indeed, observe that e is critical in $G' - e_i$, since $G' - e_i$ is biconnected but $G' - e - e_i$ is not. Hence $e \in \mathsf{Critical}_{G'}(e_i)$, and by Lemma 4.2, e must participate in $\mathcal{H}$. We may assume without loss of generality that H_1 contains the edge e. But now H_2 is a path from u_i to v_i in G', disjoint from both e, e', and $V_{\mathsf{int}}(P_2)$. Clearly, H_2 contains a subpath P in contradiction to the first statement of the present lemma. We conclude that for every $i \in [t-1]$, $\mathsf{Segment}[i, i+1]$ has a nice path.

For the second part of the last statement, let P and P' be paths as described which are not internally vertex-disjoint. Then $P \cup P'$ contains a walk, and therefore also a path, with endpoints in $V(P_1)$, internally vertex-disjoint from $V(P_1) \cup V(P_2)$, and with endpoints in distinct segments on P_1, in contradiction with the first statement of this lemma. This completes the argument for the last statement and hence the proof of the lemma. □

Let us now consider how partner sets can intersect.

Observation 4.11. *Let $G', e = (x, y), P_1, P_2, e_1, \ldots, e_t$ be as above. Let e_i, e_j be a pair of edges, $1 \leqslant i < j \leqslant t$, let $w_1, \ldots, w_r$ be the partner vertices of e_i in the order they appear on P_2, and let $w'_1, \ldots, w'_s$ be the partner vertices of e_j in the order they appear on P_2. Then $w_i \leqslant w'_j$ for every $i \in [r]$, $j \in [s]$. In particular, the set $\mathsf{Partner}(e_i) \cap \mathsf{Partner}(e_j)$ can consist of at most one vertex w, which* must *then be the last vertex of $\mathsf{Partner}(e_i)$ and the first vertex of partner(e_j) which is encountered when traversing P_2 from x to y.*

Proof. Let $w_i \in \mathsf{Partner}(e_i)$ and $w'_j \in \mathsf{Partner}(e_j)$. By Lemma 4.10 (4), $\mathsf{Segment}[i, i+1]$ has a nice path P with endpoints $u \in V(P_1)$ and $v \in V(P_2)$. By Lemma 4.10 (3), we have $w_i \leq v \leq w'_j$. Thus, $w_i \leq w'_j$ and the claim follows. □

Thus, there is a well-defined first and last element for each partner set and these two elements (they may coincide) define a subpath of P_2. Furthermore, the two subpaths corresponding to the partner sets of any two critical edges on P_1 do not have a 'strict' overlap and can only intersect in one vertex – their respective endpoints.

Having identified some of the structure in the graph, we now proceed to examine two cases. Recall that by Observation 4.8, the path P_1 contains at least $10k^2 + 23k$ edges of $\mathsf{Critical}_{G'}(e) \cap \mathsf{Heavy}(\mu(k))$. We will consider one of two cases: either there is a sufficiently large number of distinct partner sets, or there is a sufficiently large number of critical edges with identical partner sets. We show how to handle each case in turn.

4.2.2. Many distinct partner sets

We first handle the first case, by formally arguing that if there are sufficiently many distinct partner sets, then $\mathsf{Heavy}(\mu(k))$ contains a solution edge. We begin with an observation about connectivity.

Lemma 4.12. *Let $G', e = (x, y), P_1, P_2, e_1, \ldots, e_t$ be as above. For each $i \in [t]$, there is a pair of internally vertex-disjoint u_i-v_i paths P_a, P_b in $G' - e_i$ as follows.*

1. *P_a contains the edge e. Additionally, if $i > 1$, then P_a either contains e_{i-1} or intersects $\mathsf{Partner}(e_{i-1})$, and if $i < t$, then P_a either contains e_{i+1} or intersects $\mathsf{Partner}(e_{i+1})$.*

2. *P_b has an endpoint each in $\mathsf{Segment}[i-1, i]$ and $\mathsf{Segment}[i, i+1]$, but does not intersect P_1 anywhere else except in these segments,*

3. P_b *contains the set* $\mathsf{Partner}(e_i)$ *and is disjoint from the set* $\mathsf{Partner}(e_j) \setminus \mathsf{Partner}(e_i)$ *for any* $j \in [t]$, $j \neq i$, *except possibly the vertices of* $\bigcup_j \mathsf{Partner}(e_j)$ *immediately preceding and succeeding* $\mathsf{Partner}(e_i)$ *on* P_2.

Proof. Let (P_a, P_b) be a pair of internally vertex-disjoint u_i-v_i paths in $G' - e_i$. This exists since e_i is not critical in G'. Let $w \in \mathsf{Partner}(e_i)$. Since $\{w, e_i\}$ is a mixed x-y cut in $G' - e$, it follows that $\{w, e\}$ is a mixed u_i-v_i cut in $G' - e_i$. Hence one path, say P_a, must pass through e, and the other must pass through w. Since w was arbitrarily chosen, we find that P_b contains every vertex of $\mathsf{Partner}(e_i)$. Next, assume $i > 1$ and let w' be the largest vertex in $\mathsf{Partner}(e_{i-1})$ in the order $<$. Then $\{w', e_{i-1}\}$ is a mixed x-y cut in $G' - e$, thus P_a must pass through either e_{i-1} or w' on the way from u_i to x. The dual argument holds for e_{i+1} if $i < t$. This covers the first property. For the second and third properties, consider again the mixed cut $\{e_{i-1}, w'\}$. Since P_b contains u_i and v_i, both of which are on the same side in the above cut, P_b passes through the cut an even number of times; since P_a intersects the cut, P_b cannot pass through the cut and so cannot intersect P_1 in any segment before $\mathsf{Segment}[i-1, i]$, nor P_2 in any vertex before w'. (Note that P_b may intersect w', but it cannot intersect any vertex on the other side of the cut.) An analogous argument holds for e_{i+1} if $i < t$. □

We now state and prove the lemma which handles the first case, *i.e.* there are a sufficiently large number of distinct partner sets.

Lemma 4.13. *Let* $G', e = (x, y), P_1, P_2, e_1, \ldots, e_t$ *be as above. Assume that there are more than* $3k$ *distinct partner sets for the edges* $e_1, \ldots, e_t$. *Then the instance* (G, k, w^*, w, E^∞) *has a solution intersecting* $\mathsf{Heavy}(\mu(k))$.

Proof. Let $Z = \{e_1, \ldots, e_t\}$ and let $e_{i_1}, \ldots, e_{i_{3k+1}}$, be a subset of Z such that for every $1 \leqslant p < q \leqslant 3k+1$, (a) $i_q > i_p$ and (b) $\mathsf{Partner}(e_{i_p}) \neq \mathsf{Partner}(e_{i_q})$. Let $S = \{e_{i_1}, e_{i_4}, \ldots, e_{i_{3k-2}}\}$. Clearly $|S| = k$; we claim that S is a biconnectivity deletion set for G'.

To see this, let $e_{i_j} = (u_{i_j}, v_{i_j})$ be an arbitrary edge of S, and let P_a, P_b be (u_{i_j}, v_{i_j})-paths given by Lemma 4.12. Then the path P_b remains in $G' - S$; we will reconfigure P_a to be disjoint from S. We will create a path $P = P_x + (x, y) + P_y$, by separately providing a path P_x from u_{i_j} to x and a path P_y from v_{i_j} to y which are disjoint from P_b and neither of which contains the edge $e = (x, y)$. If $j = 1$, then e_{i_j} is the first edge of S along P_1 and we may simply use P_a from u_{i_j} to x as P_x, so assume $j > 1$. If P_a intersects $\mathsf{Partner}(e_{i_j - 1})$, then we may produce P_x by continuing along P_2 to x. Otherwise P_a uses the edge $e_{i_j - 1}$. In this case, produce P_x by continuing along P_1 to $\mathsf{Segment}[i_{j-3}, i_{j-3}+1]$, follow a nice path from this segment to P_2, and continue along P_2 to x.

We argue that the resulting path P_x is disjoint from P_b. If $j = 1$, then the claim is trivial. If P_a intersects $\mathsf{Partner}(e_{i_j - 1})$, then recall that P_a and P_b are internally disjoint, P_b intersects $\mathsf{Partner}(e_{i_j})$, and $\mathsf{Partner}(e_{i_j - 1}) \leqslant \mathsf{Partner}(e_{i_j})$. Thus P_x lies entirely before P_b on P_2. Otherwise, P_x uses a nice path P from $\mathsf{Segment}[i_{j-3}, i_{j-3}+1]$. The initial part of P_x follows P_a, which is disjoint from P_b by Lemma 4.12; the part between $u_{i_j - 1}$ and P is disjoint from P_b by Lemma 4.12(2); and $V_{\mathsf{int}}(P)$ is disjoint from P_b by Lemma 4.10(4). Let q be the endpoint of P on P_2, and let w be the first vertex of $\mathsf{Partner}(e_{i_{j-2}})$ on P_2. Then $q \leqslant w$ by Lemma 4.10(3), and we claim $w < V(P_b) \cap V(P_2)$. Note that $w \notin \mathsf{Partner}(e_{i_j})$ since the three sets $\mathsf{Partner}(e_{i_{j-2}})$, $\mathsf{Partner}(e_{i_{j-1}})$, $\mathsf{Partner}(e_{i_j})$ are distinct and by Observation 4.11, let w' be the first vertex of $\mathsf{Partner}(e_{i_j})$ on P_2. Assume for a contradiction that P_b intersects w. Then P_b provides a path from w to e_{i_j} that avoids $e_{i_{j-1}}$ and w'; hence $w' \notin \mathsf{Partner}(e_{i_{j-1}})$ and $\mathsf{Partner}(e_{i_{j-1}}) \cap \mathsf{Partner}(e_{i_j}) = \emptyset$. But since $\mathsf{Partner}(e_{i_{j-2}}) \neq \mathsf{Partner}(e_{i_{j-1}})$, there must be at least one further vertex $w'' \in \mathsf{Partner}(e_{i_{j-1}})$ such that $w < w'' < w'$; this contradicts that P_b intersects w by Lemma 4.10. Thus P_x and P_b are internally vertex-disjoint. The argument for P_y is analogous to that for P_x. Now $P_a = P_x + (x, y) + P_y$ and P_b form a pair of internally vertex-disjoint u_{i_j}-v_{i_j}-paths, and since $e_{i_j} \in S$ was chosen arbitrarily, we conclude that $G' - S$ is biconnected. Since G is a supergraph of G', $G - S$ is also biconnected.

Finally, it follows from Lemma 4.5 that since $S \subseteq \mathsf{Heavy}(\mu(k))$ is a biconnectivity deletion set of size k for G, there is a solution for the given instance intersecting $\mathsf{Heavy}(\mu(k))$. This completes the proof of the lemma. □

4.2.3. Identical partner sets

Due to Lemma 4.13, we assume that there are at most $3k$ distinct partner sets for the edges of $\mathsf{Critical}_{G'}(e) \cap \mathsf{Heavy}(\mu(k))$ which lie on P_1. Let $e_1, \ldots, e_t$ be the set of all edges of $\mathsf{Critical}_{G'}(e) \cap \mathsf{Heavy}(\mu(k)) \cap E(P_1)$, in the order they appear on P_1 from x to y. We define a set of exceptional edges; initially we set $\hat{I} = \{1 \leqslant i < t \mid \mathsf{Partner}(e_i) \neq \mathsf{Partner}(e_{i+1})\}$ (later we will define further exceptional edges). Then $|\hat{I}| \leqslant 3k$ by Observation 4.11; we study the structure of contiguous stretches of edges $e_i, \ldots, e_j$ with indices disjoint from $\hat{I}$. Note that all edges in such a stretch have identical partner sets. We make an observation about the structure.

Lemma 4.14. *Let $Z = \{e_i, \ldots, e_j\}$ be a set of edges of $\mathsf{Critical}_{G'}(e) \cap \mathsf{Heavy}(\mu(k)) \cap E(P_1)$ such that for every $i \leqslant i' < j' \leqslant j$, $e_{i'}$ occurs before $e_{j'}$ when traversing P_1 from x to y and $\mathsf{Partner}_{G'-e}(e_i) = \mathsf{Partner}_{G'-e}(e_j)$. Then the following hold:*

1. *$|\mathsf{Partner}_{G'-e}(e_{i'})| = 1$ and $\mathsf{Partner}_{G'-e}(e_{i'}) = \mathsf{Partner}_{G'-e}(e_i)$ for every $i \leqslant i' \leqslant j$, say $\mathsf{Partner}_{G'-e}(e_{i'}) = \{w\}$;*
2. *For every $i \leqslant i' < j$ and nice path P for $\mathsf{Segment}[i', i'+1]$, $V(P) \cap V(P_2) = \{w\}$.*

Proof. The first statement follows from Observation 4.11: since $\mathsf{Partner}_{G'-e}(e_i)$ and $\mathsf{Partner}_{G'-e}(e_j)$ can intersect in at most one vertex, we have $|\mathsf{Partner}_{G'-e}(e_i)| = 1$, and since partner sets appear in an "ordered" way on P_2, we have $\mathsf{Partner}_{G'-e}(e_{i'}) = \mathsf{Partner}_{G'-e}(e_i)$ for every $i < i' \leqslant j$. The second statement follows from the third statement of Lemma 4.10. □

In light of Lemma 4.14, for any edge e_i with $i \notin \hat{I}$, we let $w(i)$ denote the single partner vertex of e_i, i.e., $\mathsf{Partner}_{G'-e}(e_i) = \{w(i)\}$.

Definition 4.15. *For each $1 \leqslant i < t$ with $i \notin \hat{I}$, we define $\mathsf{Component}[i, i+1]$ as the set of vertices reachable from $V(\mathsf{Segment}[i, i+1])$ in the graph $G' - \{e_i, e_{i+1}, w(i)\}$ (see Figure 4). We let $\Gamma[i, i+1]$ denote the edge set $E(\mathsf{Component}[i, i+1]) \cup E(w(i), \mathsf{Component}[i, i+1])$.*

Observation 4.16. *Let $Z = [t] \setminus \hat{I}$ be the indices of non-exceptional edges e_i. The following hold.*

- *For every $i \in Z$, $\mathsf{Segment}[i, i+1]$ is contained in $\mathsf{Component}[i, i+1]$.*

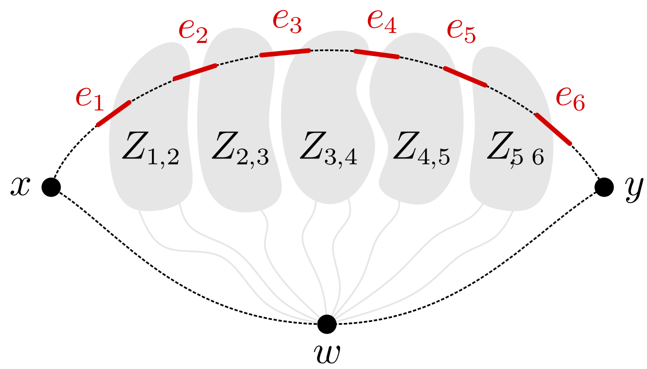

Figure 4: An illustration of the sets $\{\mathsf{Component}[i,i+1] \mid i \in [t-1]\}$ for $t = 6$, where the set $Z_{i,i+1}$ represents $\mathsf{Component}[i,i+1]$. Note that it is possible that $\mathsf{Component}[i,i+1]$ contains a single vertex, in which case this vertex must be the same as both v_i and u_{i+1}.

- *For every* $i \in Z$, $N_{G'}(\mathsf{Component}[i,i+1]) = \{u_i, v_{i+1}, w(i)\}$.
- *For every pair* $i, j \in Z$, $i \neq j$, *the sets* $\mathsf{Component}[i,i+1]$ *and* $\mathsf{Component}[j,j+1]$ *are vertex-disjoint and they are disjoint from* $V(P_2)$.
- *For every pair* $i, j \in Z$, $i \neq j$, $\Gamma[i,i+1]$ *and* $\Gamma[j,j+1]$ *are disjoint.*

Proof. The first and second statements follow from the definition of $\mathsf{Component}[i,i+1]$. For the third statement, the second statement of Lemma 4.14 implies that $\mathsf{Component}[i,i+1]$ is disjoint from $V(P_2)$ and the first statement of Lemma 4.10 implies that for every $i \neq j$, the sets $\mathsf{Component}[i,i+1]$ and $\mathsf{Component}[j,j+1]$ are vertex-disjoint. The final statement is a direct consequence of the third statement. This completes the proof. □

We need to consider one further complication. Recall that $\hat{F}$, as defined after Observation 4.6, denote the edges removed from the original graph G to create G'.

Definition 4.17. *For each* $1 \leqslant i \leqslant t-1$, $i \notin \hat{I}$, *we say that* $\mathsf{Component}[i,i+1]$ *is* affected *if the set* $\hat{F}$ *has an endpoint in* $\mathsf{Component}[i,i+1]$ *and* unaffected *otherwise.*

Since $|\hat{F}| < k$, it follows that fewer than $2k$ of these disjoint vertex-sets can be affected. We will treat these as a secondary set of exceptional indices; let $\hat{J} = \{i \in [t] \setminus \hat{I} \mid \mathsf{Component}[i,i+]$ is affected$\}$. We make a final observation.

Observation 4.18. *Let* $e_1, \ldots, e_t$ *be as above, with* $t \geqslant 10k^2+23k$. *Then there is a contiguous sequence* $a, \ldots, b \in [t]$ *of indices such that* $b \geqslant a + 2k + 3$ *and for every integral* $i \in [a,b]$, $\mathsf{Partner}(e_i) = \mathsf{Partner}(e_a)$ *and* $\mathsf{Component}[i,i+1]$ *is unaffected.*

Proof. We have $|\hat{I}| \leqslant 3k$ and $|\hat{J}| \leqslant 2k-2$, hence $[t] \setminus (\hat{I} \cup \hat{J})$ decomposes into at most $5k-1$ parts. With $t \geqslant (2k+4)(5k-1) + 5k - 2 = 10k^2 + 23k - 6$, one of these parts will contain at least $2k+4$ indices, hence its bounding indices a, b will satisfy $b \geqslant a + 2k + 3$. □

We refer to such a sequence $e_a, \ldots, e_b$ of edges as an *clean stretch* of P_1. The remaining task towards the FPT algorithm is to show that a sufficiently long clean stretch contains an irrelevant edge.

4.2.4. Reducing clean stretches

We will now restrict our attention to a single clean stretch $[a, b]$, and prove that it contains an irrelevant edge. To simplify the notation, let $w = w(a)$. We have the following lemma, where $\delta_H(Q)$ denotes the edges of H with one endpoint in Q.

Lemma 4.19. *Let $G' = G - \hat{F}$ be as above and let $Z = \{e_a, \ldots, e_b\}$ be a clean stretch. Then for every $a \leqslant i < b$, (a) $\mathsf{Component}[i, i+1]$ is unaffected, (b) $N_{G'}(\mathsf{Component}[i, i+1]) = N_G(\mathsf{Component}[i, i+1]) = \{u_i, v_{i+1}, w\}$ and (c) $\delta_{G'-w}(\mathsf{Component}[i, i+1]) = \delta_{G-w}(\mathsf{Component}[i, i+1]) = \{e_i, e_{i+1}\}$.*

Proof. Statement (a) holds by definition. For statements (b) and (c), the neighbourhoods and incident edges are the same in G as in G' since the components are unaffected, and it follows from the definition of $\mathsf{Component}[i, i+1]$ that $N_{G'}(\mathsf{Component}[i, i+1]) = \{u_i, v_{i+1}, w\}$ and $\delta_{G'-w}(\mathsf{Component}[i, i+1]) = \{e_i, e_{i+1}\}$. □

Lemma 4.20. *Let $G', Z = \{e_a, \ldots, e_b\}$ be as above. For any $i \in [a+1, b-1]$, the following hold:*

1. *There is a v_i-$\{w, u_{i+1}\}$ flow of value 2 in the graph $G[\mathsf{Component}[i, i+1] \cup \{w\}]$.*
2. *There is a u_i-$\{w, v_{i-1}\}$ flow of value 2 in the graph $G[\mathsf{Component}[i-1, i] \cup \{w\}]$.*

Proof. We show the first statement; the proof of the second is analogous. If $v_i = u_{i+1}$, then the statement follows by considering the single-vertex path v_i in combination with the nice path for $\mathsf{Segment}[i, i+1]$, hence assume that $v_i \neq u_{i+1}$. Since e_i is not critical in G' it follows that there is a u_i-v_i flow of value 2 in $G' - e_i$. However, observe that due to Lemma 4.19, $\{w, u_{i+1}\}$ is a u_i-v_i separator in $G' - e_i$. Hence of the two paths of the flow, one contains w and the other contains u_{i+1}. Truncating these paths at w and at u_{i+1} produces a flow in G'. By Lemma 4.19, this truncated flow must remain in $G'[\mathsf{Component}[i, i+1] \cup \{w\}]$, and since G is a supergraph of G' it also exists in G. This completes the proof of the statement. □

Lemma 4.21. *Let $G', Z = \{e_a, \ldots, e_b\}$ be as above. For any $i \in [a+1, b-1]$ and u_i-v_i path P in $G - w - e_i$, there exists paths P_1, P_2, P_3 such that $P = P_1 + P_2 + P_3$ and the following hold:*

1. *P_1 is a u_i-u_a path such that for every $a \leqslant j < i$, P_1 contains e_j and v_j occurs before u_j when traversing P_1 from u_i to u_a.*
2. *P_3 is a v_i-v_b path such that for every $i < j \leqslant b$, P_3 contains e_j and u_j occurs before v_j when traversing P_2 from v_i to v_b.*
3. *$E(P_2)$ disjoint from $\{e_j \mid j \in [a, b]\}$ and $V(P_2)$ is disjoint from $\bigcup_{j \in [a, b-1]} \mathsf{Component}[j, j+1]$.*

Proof. Observe that u_i lies in the set $\mathsf{Component}[i-1, i]$. Furthermore, by Lemma 4.19, we know that $\delta_{G-w-e_i}(\mathsf{Component}[i-1, i]) = \{e_{i-1}\}$. Since $\mathsf{Component}[i-1, i]$ does not contain v_i, it must be the case that P contains the edge e_{i-1} and furthermore, v_{i-1} is encountered before u_{i-1} when traversing P from u_i to v_i. We can then repeat the same argument for

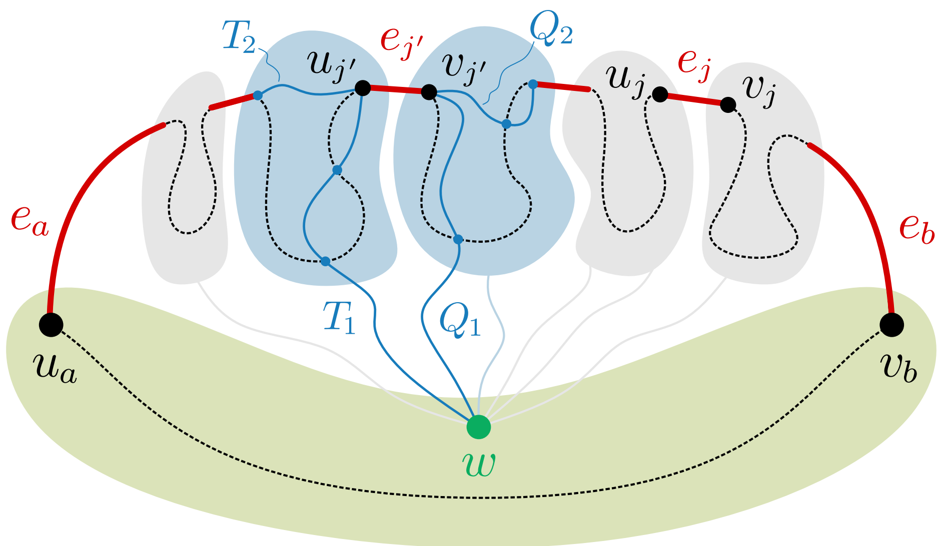

Figure 5: An illustration of the paths used in the proof of Lemma 4.22. The dashed lines represent the u_j-v_j path R.

e_{i-1}, e_{i-2} and so on until e_a. Hence, we conclude that u_a lies on P and that P_1, which is the subpath of P from u_i to u_a, contains every e_j such that $a \leqslant j < i$. Furthermore, for every $a \leqslant j < i$, v_j is encountered before u_j when traversing P_1 from u_i to u_a. This completes the argument for the first statement.

A symmetric argument implies that v_b lies on P and that P_3, which is the subpath of P from v_i to v_b, contains every e_j such that $i < j \leqslant b$. Furthermore, for every $i < j \leqslant b$, u_j is encountered before v_j when traversing P_2 from v_i to v_b.

For the final statement, observe that $E(P_1) \cup E(P_3)$ contains the set $\{e_a, e_{a+1}, \ldots, e_b\} \setminus e_i$. Therefore, the subpath of P from u_a to v_b, which we denote by P_2, is disjoint from the set $\{e_a, e_{a+1}, \ldots, e_b\}$. From Lemma 4.19, we infer that the only way P_2 can contain a vertex of $\mathsf{Component}[j, j+1]$ for some $j \in [a, b-1]$ is if it contains either e_j or e_{j+1}. Since we have already ruled this out, we conclude that P_2 is disjoint from the set $\bigcup_{j \in [a,b-1]} \mathsf{Component}[j, j+1]$. This completes the proof of the lemma. ☐

We are now ready to prove our lemma concerning irrelevant edges.

Lemma 4.22. *Let $G', Z = \{e_a, \ldots, e_b\}$ be as above where $b \geqslant a + 2k + 3$. Let $j \in [a+1, b-1]$ be such that for every $f \in \{e_{a+1}, \ldots, e_{b-1}\}$, $w(f) \geqslant w(e_j)$. Then, e_j is irrelevant.*

Proof. In order to prove the lemma, we need to argue that if there is a solution for the instance (G, k, w^*, w, E^∞), then there is one which does not contain e_j. Let S be a solution for this instance. If S is disjoint from e_j, then we are done. Suppose that this is not the case. We first argue that there is an edge which is not in S and has certain special properties. We will then argue that replacing e_j with this special edge also leads to a solution for the same instance.

Claim 4.23. *There exists $j' \in [a+1, b-1]$ such that S is disjoint from $\Gamma[j'-1, j'] \cup \{e_{j'}\} \cup \Gamma[j', j'+1]$.*

Proof. We first observe that S is disjoint from $\{e_a, \ldots, e_b\} \setminus \{e_j\}$ by Lemma 4.21. Next, for $\ell \in \{0, 1, \ldots, k\}$ let $\ell' = a + 2\ell$. Since there are $2k + 2$ edges in the set $\{e_{a+1}, \ldots, e_{b-1}\}$ and by Observation 4.16, it follows that there are $k + 1$ edge-disjoint sets $K_1, \ldots, K_{k+1}$ where

$$K_\ell = \Gamma[\ell' + 1, \ell' + 2] \quad \bigcup \quad \Gamma[\ell' + 2, \ell' + 3].$$

Since $|S| \leqslant k$, we conclude that there is an index j' such that S is disjoint from $\Gamma[j' - 1, j'] \cup \{e_{j'}\} \cup \Gamma[j', j' + 1]$. This completes the proof of the claim. □

Furthermore, by our choice of e_j, it follows that $w(e_{j'}) \geqslant w(e_j)$. Let $S' = S \setminus \{e_j\} \cup \{e_{j'}\}$. Clearly, $|S'| \leqslant k$ and $w(S') \geqslant w(S)$. We now argue that S' is also a biconnectivity deletion set.

Observe that in order to do so, it suffices to prove that there is a $u_{j'}$-$v_{j'}$ flow of value 2 in $G - S'$. Let $\mathcal{Q} = \{Q_1, Q_2\}$ be a $v_{j'}$-$\{w, u_{j'+1}\}$ flow of value 2 in the graph $G[\mathsf{Component}[j', j'+1] \cup \{w\}]$ where Q_1 is incident with w and Q_2 with $u_{j'+1}$ (see Figure 5). This flow exists by Lemma 4.20. Similarly, let $\mathcal{T} = \{T_1, T_2\}$ be a $u_{j'}$-$\{w, v_{j'-1}\}$ flow of value 2 in the graph $G[\mathsf{Component}[j' - 1, j'] \cup \{w\}]$ where T_1 is incident with w and T_2 with $v_{j'-1}$. By Observation 4.16, we know that $\mathsf{Component}[j', j' + 1]$ and $\mathsf{Component}[j' - 1, j']$ are vertex-disjoint. Therefore, $J_1 = Q_1 + T_1$ is a $u_{j'}$-$v_{j'}$ path in $G - S'$ and furthermore, J_1 is contained in $G[\mathsf{Component}[j' - 1, j'] \cup \mathsf{Component}[j', j' + 1] \cup \{w\}]$.

Since S is a solution containing e_j, it follows that there is a u_j-v_j flow of value 2 in the graph $G - S$. In particular, there is a u_j-v_j path R in the graph $G - S - w - e_j$. We assume without loss of generality that $j > j'$. The arguments in the other case are exactly the same. Due to Lemma 4.21, we know that $R = R_1 + R_2 + R_3$ where the paths R_1, R_2 and R_3 satisfy the stated properties.

Let R_1' be the subpath of R_1 from u_j to $u_{j'+1}$, R_1'' be the subpath of R_1 from $v_{j'-1}$ to u_a. Now, observe that $H = R_1' + Q_2 + e_{j'} + T_2 + R_1'' + R_2 + R_3$ is also a u_j-v_j path in $G - S - w - e_j$. Futhermore, H intersects J_1 only in $\{u_{j'}, v_{j'}\}$.

Now, $C = H + e_j$ is a cycle in $G - S - w$ such that $S \cap E(C) = e_j$ and $S' \cap E(C) = e_{j'}$. Therefore, $J_2 = C - e_{j'}$ is a $u_{j'}$-$v_{j'}$ path in $G - S - w$. Since C intersects J_1 only in $\{u_{j'}, v_{j'}\}$, we conclude that J_2 is internally vertex-disjoint from J_1 and contains no edges of S'. Thus, we have demonstrated a $u_{j'}$-$v_{j'}$ flow $\{J_1, J_2\}$ of value 2 in the graph $G - S'$, implying that S' is a biconnectivity deletion set and hence a solution for the instance (G, k, w, w^*, E^∞). Therefore, we conclude that the edge e_j is irrelevant, completing the proof of the lemma. □

Combining Lemma 4.13 and the fact that we can clearly locate the irrelevant edge e_j (in the statement of Lemma 4.22) in polynomial time, we obtain Lemma 4.3, our main objective. This concludes the description of our algorithm for Weighted Biconnectivity Deletion.

4.3. A randomized kernel for Unweighted Biconnectivity Deletion

We now present our randomized kernel for the Weighted Biconnectivity Deletion problem where instances are of the form (G, k, w^*, w, E^∞) where $w(e) = 1$ for every $e \in E(G) \setminus E^\infty$, $w(e) = 0$ for every $e \in E^\infty$, and $w^* = k$. This version of the problem will be referred to as Unweighted Biconnectivity Deletion and instances of this problem will henceforth be of the form (G, k, E^∞) where a solution is a biconnectivity deletion set of size

k contained in $E(G) \setminus E^\infty$. We continue to refer to the set $E(G) \setminus E^\infty$ as the set of potential solution edges and assume without loss of generality that at any point, any edge in the set $\mathsf{Critical}_G(\emptyset)$ is already part of E^∞. Finally, recall that a *linkage* from A to B in a digraph D, where A and B are vertex sets, is a collection of $|A| = |B|$ pairwise vertex-disjoint paths originating in A and terminating in B.

Our kernelization relies on a result of Kratsch and Wahlström [14]. Before we are able to state it formally, we need the following definitions. Let us define a *potentially overlapping A-B vertex cut* in a digraph D to be a set of vertices $C \subseteq V(D)$ such that $D - C$ contains no directed path from $A \setminus C$ to $B \setminus C$. For any digraph D and set $X \subseteq V(D)$, a set $Z \subseteq V(D)$ is called a *cut-covering set* for (D, X) if for any $A, B, R \subseteq X$, there is a minimum-cardinality potentially overlapping A-B vertex cut C in $D - R$ such that $C \subseteq Z$. We are now ready to state the result of Kratsch and Wahlström on which our kernelization is based.

Lemma 4.24 (Corollary 3, [14])**.** *Let D be a directed graph and let $X \subseteq V(D)$. We can identify a cut-covering set Z for (D, X) of size $O(|X|^3)$ in polynomial time with failure probability $O(2^{-|V(D)|})$.*

Armed with this lemma, we first give a randomized kernelization that outputs an instance whose size is bounded polynomially in the number of the potential solution edges in the input instance.

Lemma 4.25. Unweighted Biconnectivity Deletion *has a randomized kernel with number of vertices bounded by $O(|E(G) \setminus E^\infty|^3)$.*

Proof. Let $F = E(G) \setminus E^\infty$ be the set of potential solution edges. Now, the kernelization task essentially consists of retaining enough information from the input graph G to verify for any set $S \subseteq F$, whether S is a biconnectivity deletion set for G. We observe a simple equivalent statement of this property.

[Magnus] This entire claim + proof are newly added

Claim 4.26. *Let $G = (V, E)$ be a biconnected graph and $S \subseteq E$. Then S is a biconnectivity deletion set for G if and only if for every edge $e = (u, v) \in S$, the maximum $u - v$-flow in $G - S$ is at least 2.*

Proof. Indeed, on the one hand, if the condition is not fulfilled for some $(u, v) \in S$, then $G - S$ is clearly not biconnected. On the other hand, assume that $G - S$ is not biconnected, and let (U, W) be a separation of order 1, say $U \cap W = \{v\}$. Since G is biconnected there must exist an edge e between $U - v$ and $W - v$, say $e = (u, w)$. But then clearly $e \in S$ and the $u - w$-flow in $G - S$ is at most 1. □

We show an equivalent formulation of this as a question about the existence of linkages in an auxiliary digraph, followed by an appropriate invocation of Lemma 4.24.

For the formulation, we create a digraph $D_{G,F}$ from G and F. We refer to this digraph as D when G and F are clear from the context. In the first step, subdivide every edge $e \in F$ with a new vertex x_e. That is, for an edge $e = (u, v) \in F$, we create a new vertex x_e, remove the edge e and add edges (u, x_e) and (v, x_e). Let G_1 be the resulting undirected graph. In the second step, replace every edge (u, v) in $E(G_1)$ by a pair of arcs (u, v), (v, u). Finally, for every vertex v incident to any edge of F in G, add vertices v^+, v^- and add arcs from v^+ to all vertices in $N_{G_1}(v)$ and from all vertices in $N_{G_1}(v)$ to v^-. Let D be the resulting digraph. Note that $N_D^+(v^-) = \emptyset$ and $N_D^-(v^+) = \emptyset$. Let $X_E = \{x_e \mid e \in F\}$, $X_V = \{v^+, v^-, v \mid e \in F, e = (u,v)\}$ and $X = X_E \cup X_V$. We now relate solutions for the given instance and linkages in D.

Claim 4.27. *For any $S \subseteq F$, S is a biconnectivity deletion set for G if and only if for every edge $(u,v) \in S$ there is a linkage from $\{u^+, u\}$ to $\{v^-, v\}$ in $D - \{x_e \mid e \in S\}$.*

Proof. Consider an arbitrary edge $e = (u,v)$ in S. On the one hand, assume that there exists a u-v flow of value 2 in $G - S$. Then, by definition there exists a pair $\{P_1, P_2\}$ of internally vertex-disjoint u-v-paths in $G - S$. Observe that orienting both paths from u to v, replacing one copy of u by u^+ and one copy of v by v^-, and subdividing any edge $e' \in E(P_i) \cap F$ by the vertex $x_{e'}$ yields the required linkage.

On the other hand, let $\{P_1, P_2\}$ be a linkage from $\{u^+, u\}$ to $\{v^-, v\}$ in $D - \{x_e \mid e \in S\}$. For each $i \in \{1,2\}$, if P_i originates in u^+, then replace u^+ by u and if P_i terminates in v^-, then replace v^- by v. Call the paths resulting from P_1 and P_2 in this way, P_1' and P_2' respectively. Then P_1' and P_2' use only vertices of $V(G) \cup X_E$. Furthermore, these two paths use no edge of S since by definition, P_1 and P_2 are disjoint from every vertex $x_{e'}$ such that $e' \in S$. Thus the paths P_1', P_2' use only edges and vertices present in $G - S$, and form an internally vertex-disjoint pair of u-v-paths.

Hence linkages as described exist if and only if the $u - v$-flow in $G - S$ is at least 2 for every edge $(u,v) \in S$.G As argued above, this is equivaent to S being a biconnectivity deletion set in G.

□

Let $Z \subseteq V(D)$ be the cut-covering set for (D, X), as computed by the algorithm of Lemma 4.24. Having in hand the set Z, we define the set $Y = (Z \cap V(G)) \cup V(F)$. Note that Z could contain vertices from X_V, but we want Y to be a subset of $V(G)$. Therefore, we first add to Y those vertices in Z which are also vertices in G and then add the vertices of $V(F)$. Our objective now is to reduce G down to what is commonly known as the *torso* graph of G defined by Y (see [14]). We now make this precise in the form of reduction rules. In the rest of the proof of the lemma, we fix Z to be a set computed using Lemma 4.24 and let Y be as defined above. We now state three reduction rules which will be applied on the given instance in the order in which they are presented.

Reduction Rule 4.28. *If $k = 0$, then return an arbitrary yes-instance of constant size.*

Reduction Rule 4.29. *Suppose that Reduction Rule 4.28 has been applied on the given instance. If there is an edge $(u,v) \in F$ such that G contains a u-v path avoiding all edges of F and all vertices of $Y \setminus \{u,v\}$, then delete (u,v) from G and reduce the budget k by 1. That is, return the instance $(G - \{(u,v)\}, k-1, E^\infty)$.*

Reduction Rule 4.30. *Suppose that Reduction Rule 4.28 and Reduction Rule 4.29 have been applied exhaustively on the given instance. For every pair $u, v \in Y$ such that $(u,v) \notin E(G)$ and there is a u-v-path in G that is internally vertex-disjoint from Y, we add the edge (u,v). Finally, return the instance (G', k, E'^∞), where $G' = G[Y]$ and $E'^\infty = (E^\infty \cap E(G')) \cup (E(G') \setminus E(G))$.*

The soundness of Rule 4.28 is trivial and we move on to prove the soundness of the remaining two rules.

Claim 4.31. *Reduction Rules 4.29 and 4.30 are sound.*

Proof. Let $e = (p, q) \in F$ be an edge which is deleted in an application of Reduction Rule 4.29. Observe that in order to argue the soundness of this reduction rule, it suffices to argue that e is part of some solution for the given instance (if there exist any). Let S be an arbitrary subset of F containing e such that $S \setminus \{e\}$ is a solution. If S itself is a biconnectivity deletion set then we may correctly conclude that e is part of some solution for the given instance. Suppose that this is not the case.

Recall that by the previous claim, S is a biconnectivity deletion set for G if and only if there is a linkage from $\{u^+, u\}$ to $\{v^-, v\}$ in $D - \{x_e \mid e \in S\}$ for every $(u, v) \in S$. Since we are in the case that S is *not* a biconnectivity deletion set, there is a $(u, v) \in S$, with $A = \{u^+, u\}$, $B = \{v^-, v\}$, and $R = \{x_e \mid e \in S\}$ such that there is no linkage from A to B in $D - R$. Since $S \setminus \{e\}$ is a biconnectivity deletion set, $G - (S \setminus e)$ is biconnected and it follows from Claim 4.26 that we may choose $u = p$ and $v = q$, where furthermore $\kappa_{G-S}(p, q) = 1$. In addition, the fact that Z is a cut-covering set for (D, X) implies that Z contains a vertex w such that $C = \{w\}$ is a minimum-cardinality potentially overlapping A-B vertex cut in $D - R$. It is straightforward to see that $w \notin \{p, q, p^+, q^-\}$ since otherwise, there will be at least one path from A to B which is disjoint from w. Finally, since $\kappa_{G-S}(p, q) = 1$, it follows that every p-q path in $G - S$ intersects w. If $w \in X_E$ then we know that it corresponds to an edge in F. Otherwise, it corresponds to a vertex in Y. In either case, we obtain a contradiction to the applicability of Reduction Rule 4.29 on the edge (p, q), completing the proof of soundness for this rule.

We now argue the soundness of Reduction Rule 4.30. To do so, we prove that $S \subseteq F$ is a solution for (G, k, E^∞) if and only if it is a solution for (G', k, E'^∞). Let $D_1 = D_{G,F}$ and let $D_2 = D_{G',F}$.

In the forward direction, suppose that S is a solution for (G, k, E^∞). By Claim 4.27, it follows that for every edge $(u, v) \in S$, there is a linkage from $\{u^+, u\}$ to $\{v^-, v\}$ in $D_1 - \{x_e \mid e \in S\}$. Fix such an edge (u, v) and let the paths in the linkage be P_1, P_2. If we demonstrate such a linkage in D_2, then we are done. This can be achieved as follows. Let $i \in \{1, 2\}$ and consider a pair of vertices $x_i, y_i \in V(P_i) \cap Y$ such that the subpath of P_i from x_i to y_i has all its internal vertices disjoint from Y. Then, we know that the graph G' contains the edge (x_i, y_i) and hence the digraph D_2 contains the arc (x_i, y_i). We replace the subpath from x_i to y_i with the arc (x_i, y_i) and we do this for every such subpath of P_i. It is straightforward to see that what results is indeed a linkage from $\{u^+, u\}$ to $\{v^-, v\}$ in $D_2 - \{x_e \mid e \in S\}$. Hence, we conclude that S is a solution for (G', k, E'^∞).

The same argument can be reversed for the converse direction in order to convert, for any $(u, v) \in S$, a linkage from $\{u^+, u\}$ to $\{v^-, v\}$ in $D_2 - \{x_e \mid e \in S\}$ to a a linkage from $\{u^+, u\}$ to $\{v^-, v\}$ in $D_1 - \{x_e \mid e \in S\}$. This completes the proof of soundness of Reduction Rule 4.30. ◻

The above claim implies that if $(G', k', E(G') \setminus F')$ is the instance obtained by exhaustively applying the three reduction rules above, then $(G', k', E(G') \setminus F')$ is indeed equivalent to

(G, k, E^∞). Furthermore, the size $|V(G')| = O(|F|^3)$ and the randomized polynomial running time follow from Lemma 4.24. This completes the proof of the lemma. □

Theorem 1.2. Unweighted Biconnectivity Deletion *has a randomized kernel with $O(k^9)$ vertices.*

Proof. Let (G, k, E^∞) be the given instance and let $F = E(G) \setminus E^\infty$ be the set of potential solution edges in this instance. We present reduction rules which reduce F (while maintaining equivalence) to size $O(k^3)$; the result then follows from Lemma 4.25.

If $|F| = O(k^3)$, we are done. Otherwise, following the approach described in Section 4.2.1, we greedily construct a biconnectivity deletion set in G, at each step keeping track of the edges that become critical. That is, we let $\hat{S} = \{f_1, \ldots, f_r\} \subseteq F$ be a set greedily constructed as follows. The edge f_1 is an arbitrary edge in F and for each $2 \leqslant i \leqslant r$, f_i is an arbitrary edge which is *not* critical in $G - \{f_1, \ldots, f_{i-1}\}$. As earlier, we terminate this procedure after k steps if we manage to find edges $\{f_1, \ldots, f_k\}$ or earlier if for some $r < k$, every remaining edge of F is critical in $G - \{f_1, \ldots, f_r\}$.

If $r = k$, then we identify the instance as a yes-instance and return an arbitrary yes-instance of constant size. Otherwise, if there is an $i \in [r]$ such that $G - \{f_1, \ldots, f_i\}$ is biconnected and $|\mathsf{Critical}_{G-\{f_1,\ldots,f_{i-1}\}}(f_i)| \geqslant 20k^2 + 46k$, then we execute the case analysis in Section 4.2.2 and in polynomial time, either find $3k + 1$ distinct partner sets or an irrelevant edge. In the latter case, we simply remove this irrelevant edge from F (add it to the set E^∞). Finally, if we reach a case with at least $3k + 1$ distinct partner sets, then according to the proof of Lemma 4.13 we can find a biconnectivity deletion set $S \subseteq F$ with $|S| \geqslant k$ in polynomial time, and since we are dealing with the unweighted case, we can simply identify the instance as a yes-instance and return an arbitrary yes-instance of constant size.

The only remaining case is that this greedy algorithm fails to produce a large enough solution yet never marks too many edges as critical at once. That is, it terminates in $r < k$ steps and never marks more than $20k^2 + 46k$ edges as critical in step i for any $i \in [r]$. This implies that $|F| \leqslant 20k^3 + 46k^2 + k = O(k^3)$, completing the proof of the theorem. □

5. Edge deletion to highly connected graphs parameterized by connectivity and the size of the deletion set

In this section, we consider the most general parameterization of the unweighted version of the problem and give a *non-uniform* FPT algorithm. In this parameterization, we consider both k and the value of the vertex-connectivity we want to preserve in the input graph, as parameters. We formally define the problem below.

Vertex-connectivity Deletion	**Parameter:** $k + \rho$
Input: A undirected graph G, integers ρ, k such that G is ρ-vertex connected.	
Question: Is there a set $S \subseteq E(D)$ of size k such that $G - S$ is also ρ-vertex connected?	

Theorem 1.3. *For every fixed k and ρ, the* Vertex-connectivity Deletion *problem can be solved in time $O(n^6)$.*

We will prove this theorem using Proposition 2.6. Recall that in order to invoke Proposition 2.6, we need to first argue that this problem can be expressed in CMSOL.

Lemma 5.1. Vertex-connectivity Deletion *can be expressed in CMSOL.*

Proof. Let $G = (V, E)$ be a graph. The following sentence is not hard to express in CMSOL. For every set $U \subset V$ of size $\rho - 1$, there is a set $S \subseteq E$ of size k such that the following holds: for every non-empty $Y \subset V \setminus U$ there is an edge $e \in E \setminus S$ such that one end-vertex of e belongs to Y and the other to the complement of Y in $V \setminus U$. Indeed, it suffices to require that $\mathbf{card}_{0,k}(S) = \mathbf{true}$ and $\mathbf{card}_{0,\rho-1}(U) = \mathbf{true}$ since if the sentence holds for larger sets S and U, it holds for those of cardinality k and $\rho - 1$, too. This completes the proof of the lemma. □

In the rest of this section, whenever we say that k and ρ are fixed, we denote by c the value $2k + \rho$ and denote by s the constant given by Proposition 2.6 corresponding to this c. Lemma 5.1 and Proposition 2.6 together imply that in order to prove Theorem 1.3, it is sufficient to prove the following lemma.

Lemma 5.2. *For every fixed k and ρ,* Vertex-connectivity Deletion *can be solved in time $\mathcal{O}(n^5)$ on graphs which are (s, c)-unbreakable.*

The rest of this section is devoted to the proof of this lemma.

5.1. An algorithm for Vertex-connectivity Deletion *on (s, c)-unbreakable graphs*

We are now ready to prove our main structural lemma providing a bound on the number of 'non-critical' edges in any no-instance of Vertex-connectivity Deletion where the input graph is unbreakable.

Lemma 5.3. *For fixed k and ρ, let G be an (s, c)-unbreakable graph. Then, (G, ρ, k) is a yes-instance of* Vertex-connectivity Deletion *or $|E(G) \setminus \mathsf{Critical}^\rho_G(\emptyset)| < (2s+4k+9)\rho \cdot k$.*

Proof. Suppose that $|E(G) \setminus \mathsf{Critical}^\rho_G(\emptyset)| \geqslant (2s + 4k + 9)\rho \cdot k$. We first greedily select a maximal ordered set S of edges $\{e_1, \dots, e_\ell\}$ such that for every $i \in [\ell]$, the graph $G - S_i$ is ρ-connected, where $S_i = \{e_1, \dots, e_i\}$ and $S_0 = \emptyset$. Observe that by our assumption, the set $E(G) \setminus \mathsf{Critical}^\rho_G(\emptyset)$ is non-empty and hence there is at least one edge $e \in E(G) \setminus \mathsf{Critical}^\rho_G(\emptyset)$ such that $G - e$ is ρ-vertex connected. As a result, the set S_ℓ is non-empty. Furthermore, since S is maximal, it must be the case that $\mathsf{Critical}^\rho_{G-S_\ell}(\emptyset) = \emptyset$. That is, $G - S_\ell$ is ρ-connected and every edge in $G - S_\ell$ is ρ-critical.

If $\ell \geqslant k$, then it follows that (G, ρ, k) is a yes-instance of Vertex-connectivity Deletion. Suppose that this is not the case. Then, it follows that for some $i \in [\ell]$, the size of the set $\mathsf{Critical}_{G-S_{i-1}}(e_i)$ is at least $(2s + 4k + 9)\rho$. We now argue the following for such an i.

Claim 5.4. *There is an edge $e' \in \mathsf{Critical}_{G-S_{i-1}}(e_i)$ such that $G - S_{i-1}$ has a $(s + 2k, \rho + 3)$-separation (A, B) where $V(e_i) \cup V(e') \subseteq (A \cap B)$.*

Proof. Let $e_i = (x, y)$. We begin by selecting an arbitrary x-y flow $\mathcal{P}$ of value ρ in $G' = G - S_{i-1}$. The definition of S implies that G' is ρ-connected. Hence, such a flow exists. By the equivalence of statements (1) and (3) of Lemma 4.2, it follows that every edge in $\mathsf{Critical}_{G-S_{i-1}}(e_i)$ lies on a path in $\mathcal{P}$ (see Figure 6). Let $P_1, \dots, P_\rho$ be the paths in $\mathcal{P}$. Since

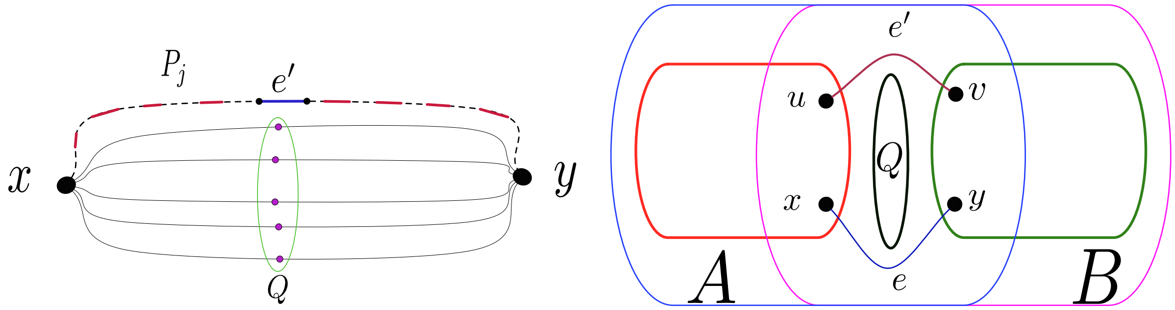

Figure 6: An illustration of the edge e' on the x-y path P_j in the graph $G - S_i$ and the separation (A, B) in $G - S_{i-1}$ defined based on e' and Q.

$|\mathsf{Critical}_{G-S_{i-1}}(e_i)| \geqslant (2s + 4k + 9)\rho$, it must be the case that there is a $j \in [\rho]$ such that P_j contains at least $(2s + 4k + 9)$ edges of $\mathsf{Critical}_{G-S_{i-1}}(e_i)$.

Let $r = 2s + 4k + 9$ and let $\{p_1, \dots, p_r\}$ denote an arbitrary subset of $\mathsf{Critical}_{G-S_{i-1}}(e_i)$ contained in P_j. Furthermore, assume without loss of generality that for every $1 \leqslant z < z' \leqslant r$, p_z is encountered before $p_{z'}$ when traversing P_j from x to y. Let $e' = (u, v) = p_{s+2k+5}$, where u is encountered before v when traversing P_j from x to y.

Since e' lies on *every* x-y flow of value ρ in G', there is an x-y separator Q of size $\rho - 1$ in $G' - e'$. Furthermore, Q must be disjoint from precisely one of the paths in $\mathcal{P}$. If it is disjoint from any path other than P_j, then it would imply the presence of an x-y path in $G' - e' - Q$, a contradiction. Hence, it must be the case that Q is disjoint from P_j.

Let $A = R_{G'-e'}(x, Q) \cup Q \cup \{x, y, u, v\}$ and let $B = (V(G) \setminus A) \cup Q \cup \{x, y, u, v\}$. Observe that (A, B) is a separation of G' with separator $Q \cup \{x, y, u, v\}$. Therefore, the order of this separation is $|Q| + 4 = \rho + 3$. Furthermore, since Q is disjoint from P_j, it follows that the vertices in the subpath of P_j from x to u are contained in A and the vertices in the subpath of P_j from v to y are all contained in B. Hence, $|A \setminus B|, |B \setminus A| > s + 2k$. This completes the proof of the claim. □

We will now transform the separation (A, B) in $G - S_{i-1}$ to a separation in G violating the premise of the lemma. Henceforth, we will let Z denote the set $A \cap B$ which we already know contains $V(e_i) \cup V(e')$. Let $Q' = Z \cup V(S_{i-1})$, $A' = A \cup (B \cap Q')$ and $B' = B \cup (A \cap Q')$. Observe that $A' \cap B' = Q'$. In other words, we construct another separation (A', B') where $A' \cap B' = Q'$ by taking the separation (A, B) and moving the endpoints of all the edges in S_{i-1} to the 'middle' of the separation. Since $\ell < k$, it follows that $i < k$ and hence $|V(S_{i-1})| \leqslant 2(k - 2)$. Therefore, we conclude that (A', B') is a separation in G of order at most $2k + \rho$ and furthermore, $|A'|, |B'| > s$, implying that G has a $(s, \rho + 2k)$-separation, which is in fact a (s, c)-separation, a contradiction. This completes the proof of the lemma. □

Since any solution for the instance (G, k, ρ) must be a subset of $E(G) \setminus \mathsf{Critical}^{\rho}_G(\emptyset)$, Lemma 5.3 implies that if, for a fixed k and ρ and instance (G, k, ρ), the set $E(G) \setminus \mathsf{Critical}_G(\emptyset)$ has size at least $(2s + 4k + 9)\rho k$, then the instance is a yes-instance and otherwise one can simply examine all subsets of $E(G) \setminus \mathsf{Critical}_G(\emptyset)$ of size at most k and decide whether the given instance is a yes-instance. Therefore, Lemma 5.2 follows from Lemma 5.3 and the existence of the $O(\rho n^3)$-time ρ-vertex-connectivity algorithm of Even [22].

6. Conclusions

Our results on Path-contraction Preserving Strong Connectivity, Weighted Biconnectivity Deletion, and Vertex-connectivity Deletion provide significant data points for the algorithmic landscape of graph editing problems under connectivity constraints and its application in network design.

Since we established that Path-contraction Preserving Strong Connectivity is W[1]-hard for general digraphs, we ask whether the problem becomes FPT when restricted to planar digraphs or other structurally sparse classes.

Concerning the parameterized algorithm for Weighted Biconnectivity Deletion, we ask whether the dependence of $2^{O(k \log k)}$ can be improved to single-exponential or proven to be optimal. Similarly, it remains open whether even a dependence of $2^{O(k \log k)}$ can be achieved for the Vertex-connectivity Deletion problem for a fixed value of ρ greater than 2.

Finally, regarding our polynomial kernel for Unweighted Biconnectivity Deletion, we ask whether it is possible to obtain a deterministic kernel. It is also left open whether the weighted case admits a polynomial kernel.

The results presented in this paper raise more questions than they answer, a clear indication that connectivity constraints are far from properly explored under the paradigm of parameterized complexity. As such, the topic offers exciting but challenging opportunities for further research.

References

[1] J. Bang-Jensen, M. Basavaraju, K. V. Klinkby, P. Misra, M. S. Ramanujan, S. Saurabh, M. Zehavi, Parameterized algorithms for survivable network design with uniform demands, in: Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018, 2018, pp. 2838–2850.

[2] J. Bang-Jensen, G. Z. Gutin, Digraphs: theory, algorithms and applications, Springer Science & Business Media, 2008.

[3] A. Frank, Connections in Combinatorial Optimization, Oxford Univ. Press, 2011.

[4] H. Nagamochi, An approximation for finding a smallest 2-edge-connected subgraph containing a specified spanning tree, Discrete Applied Mathematics 126 (1) (2003) 83 – 113, 5th Annual International Computing and combinatorics Conference.

[5] J. Guo, J. Uhlmann, Kernelization and complexity results for connectivity augmentation problems, Networks 56 (2) (2010) 131–142.

[6] D. Marx, L. A. Végh, Fixed-parameter algorithms for minimum-cost edge-connectivity augmentation, ACM Transactions on Algorithms (TALG) 11 (4) (2015) 27.

[7] M. Basavaraju, F. V. Fomin, P. Golovach, P. Misra, M. Ramanujan, S. Saurabh, Parameterized algorithms to preserve connectivity, in: Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP), Springer, 2014, pp. 800–811.

[8] J. Bang-Jensen, A. Yeo, The minimum spanning strong subdigraph problem is fixed parameter tractable, Discrete Applied Mathematics 156 (15) (2008) 2924–2929.

[9] T. Watanabe, A. Nakamura, Edge-connectivity augmentation problems, J. Comput. System Sci. 35 (1987) 96 – 144.

[10] A. Frank, Augmenting graphs to meet edge-connectivity requirements, SIAM J. Discrete Math. 5 (1) (1992) 25–53.

[11] A. Frank, T. Jordán, Minimal edge-coverings of pairs of sets, J. Comb. Theory, Ser. B 65 (1) (1995) 73–110.

[12] L. A. Végh, Augmenting undirected node-connectivity by one, SIAM J. Discrete Math. 25 (2) (2011) 695–718.

[13] D. Lokshtanov, M. S. Ramanujan, S. Saurabh, M. Zehavi, Reducing CMSO model checking to highly connected graphs, CoRR abs/1802.01453.

[14] S. Kratsch, M. Wahlström, Representative sets and irrelevant vertices: New tools for kernelization, in: 53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012, IEEE Computer Society, 2012, pp. 450–459.

[15] S. Kratsch, Recent developments in kernelization: A survey, Bulletin of the EATCS 113.

[16] D. M. Moyles, G. L. Thompson, An algorithm for finding a minimum equivalent graph of a digraph, Journal of the ACM (JACM) 16 (3) (1969) 455–460.

[17] F. V. Fomin, D. Lokshtanov, F. Panolan, S. Saurabh, Efficient computation of representative families with applications in parameterized and exact algorithms, Journal of the ACM (JACM) 63 (4) (2016) 29:1–29:60.

[18] R. G. Downey, M. R. Fellows, Fixed-parameter tractability and completeness II: on completeness for W[1], Theor. Comput. Sci. 141 (1&2) (1995) 109–131.

[19] S. Arnborg, J. Lagergren, D. Seese, Easy problems for tree-decomposable graphs, J. Algorithms 12 (2) (1991) 308–340.

[20] B. Courcelle, The monadic second-order logic of graphs. i. recognizable sets of finite graphs, Inf. Comput. 85 (1) (1990) 12–75.

[21] B. Courcelle, The expression of graph properties and graph transformations in monadic second-order logic, in: G. Rozenberg (Ed.), Handbook of Graph Grammars and Computing by Graph Transformations, Volume 1: Foundations, World Scientific, 1997, pp. 313–400.

[22] S. Even, R. E. Tarjan, Network flow and testing graph connectivity, SIAM journal on computing 4 (4) (1975) 507–518.