HOMOLOGICAL REPRESENTATIVES IN TOPOLOGICAL PERSISTENCE

by

Tao Hou

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

Department of Computer Science

West Lafayette, Indiana

May 2022

THE PURDUE UNIVERSITY GRADUATE SCHOOL STATEMENT OF COMMITTEE APPROVAL

Dr. Tamal Krishna Dey, Chair

Department of Computer Science

Dr. Saugata Basu

Department of Mathematics

Dr. Erin Wolf Chambers

Department of Computer Science, Saint Louis University

Dr. Kent Richard Quanrud

Department of Computer Science

Approved by:

Dr. Kihong Park

To *my wife*, Lu Wang

ACKNOWLEDGMENTS

Finishing the work for a PhD degree is never easy. Without the supports of others, it is hard to imagine all the things I have done while pursuing my PhD degree. Hence, I would like to give thanks to those who have helped me come thus far.

First and foremost, I would like to thank my PhD advisor Tamal K. Dey. It is Dr. Dey who brought me into the fanciful world of computational topology and geometry and taught me everything about research. I still could remember the first several months being his student, unfledged, a little ambitious, a little nervous, but knowing little about research. If there was any growth or progress I have made since then, Dr. Dey would always be a part in it. I also want to thank him for all the encouragements he gave me without any hesitation – they meant a lot to me. Finally, I want to thank him and his wife for all the great food they cooked for treating me and my colleagues – it was super delicious and made me homesick.

I want to thank all the other members in my committee (preliminary and final): Dr. Saugata Basu, Dr. Erin W. Chambers, Dr. Alex Pothen, and Dr. Kent R. Quanrud. Thanks for serving in my committee and giving me so many helpful advices during my presentation.

I want to thank all the collaborators who worked with me on research: Dr. Dunbar P. Birnie, Anand V. Patel, Juan D. Beltran Rodriguez, and Sayan Mandal. Research work is rarely accomplished alone and I want to thank them for helping me and making me part of a great research team.

I want to thank my parents, for all the love, care, and support they have given me.

Lastly, and very importantly, I would like to give my thanks to my wife. She has done so much for me to pursue a PhD degree overseas. She has always been supportive for everything I set up my mind to and every dream I intend to pursue. I feel fortunate to have her as a companion along the road. Without her, I would not have the courage and strength to navigate through all these.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Harnessing the power of data has been a driving force for computing in recently years. However, the non-vectorized or even non-Euclidean nature of certain data with complex structures also poses new challenges to the data science community. Topological data analysis (TDA) has proven effective in several scenarios for alleviating the challenges, by providing techniques that can reveal hidden structures and high-order connectivity for data. A central technique in TDA is called *persistent homology*, which provides intervals tracking the birth and death of topological features in a growing sequence of topological spaces. In this dissertation, we study the *representative* problem for persistent homology, motivated by the observation that persistent homology does not pinpoint a specific homology class or cycle born and dying with the persistence intervals. Furthermore, studying the representatives also leads us to new findings for related problems such as persistence computation.

First, we look into the representative problem for (standard) persistence homology and term the representatives as *persistent cycles*. We define persistent cycles as cycles born and dying with given persistence intervals and connect the definition to interval decomposition for persistence modules. We also look into the computation of optimal (minimum) persistent cycles which have guaranteed quality. We prove that it is NP-hard to compute minimum persistent p-cycles for the two types of intervals in persistent homology in general dimensions ($p > 1$). In view of the NP-hardness results, we then identify a special but important class of inputs called *weak* $(p + 1)$*-pseudomanifolds* whose minimum persistent p-cycles can be computed in polynomial time. The algorithms are based on a reduction to minimum (s, t)-cuts on dual graphs.

Second, we propose alternative persistent cycles capturing the dynamic changes of homological features born and dying with persistence intervals, which the previous persistent cycles do not reveal. We focus on persistent homology generated by piecewise linear (PL) functions and base our definition on an extension of persistence called the *levelset zigzag* persistence. We define a *sequence* of cycles called *levelset persistent cycles* containing a cycle between each consecutive critical points within the persistence interval. Due to the NP-harness results proven previously, we propose polynomial-time algorithms computing

optimal sequences of levelset persistent p-cycles for weak $(p+1)$-pseudomanifolds. Our algorithms draw upon the idea of relating optimal cycles to min-cuts in a graph that we exploited earlier for standard persistent cycles. Note that levelset zigzag poses non-trivial challenges for the approach because a sequence of optimal cycles instead of a single one needs to be computed in this case.

Third, we investigate the computation of *zigzag* persistence on graph inputs, motivated by the fact that graphs model real-world circumstances in many applications where they may constantly change to capture dynamic behaviors of phenomena. Zigzag persistence, an extension of the standard persistence incorporating both insertions and deletions of simplices, is one appropriate instrument for analyzing such changing graph data. However, unlike standard persistence which admits nearly linear-time algorithms for graphs, such results for the zigzag version improving the general $O(m^\omega)$ time complexity are not known, where $\omega < 2.37286$ is the matrix multiplication exponent. We propose algorithms for zigzag persistence on graphs which run in near-linear time. Specifically, given a filtration of length m on a graph of size n, the algorithm for 0-dimension runs in $O(m \log^2 n + m \log m)$ time and the algorithm for 1-dimension runs in $O(m \log^4 n)$ time. The algorithm for 0-dimension draws upon another algorithm designed originally for pairing critical points of Morse functions on 2-manifolds. The correctness proof of the algorithm, which is a major contribution, is achieved with the help of representatives. The algorithm for 1-dimension pairs a negative edge with the *earliest* positive edge so that a representative 1-cycle containing both edges resides in all intermediate graphs.

1. INTRODUCTION

Recently, data science has emerged as one of the central topics in computer science, thanks to algorithms newly discovered and computing powers unleashed by parallel processing. The critical techniques leading to a booming data-centric era are mostly *statistics-based*, meaning that large amount of vectorized, Euclidean training data must be made available in order for algorithms to discover (or learn) patterns that can be generalized to unseen inputs. However, this also poses challenges to users of these techniques. First, while in some cases large amount of data (such as user behavior data collected from the internet) are easily accessible, it is considerably expensive (if not impossible) to collect datasets suitable for statistical training for some tasks. Second, some data are not Euclidean in nature, e.g., point clouds, graphs, or triangular meshes, making them not the ideal input for statistical training algorithms.

Topological data analysis (TDA), as an emerging area, has proven effective in several scenarios for alleviating challenges described above [1]. Broadly speaking, TDA aims at providing methods for revealing structures in data, mostly hidden and in high dimensions, with the help of algebraic topology [2]. A central technique in TDA is called *persistent homology*, whose advent is resulted from an algorithm proposed by Edelsbrunner et al. [3]. Since its advent, persistent homology has been more extensively studied in different fronts, such as its underlying algebraic structure [4], [5], its stability [6], and generalizations [7], [8]. The idea of persistent homology can be briefly summarized as follows: given a growing sequence of topological spaces, persistent homology returns a multiscale and stable topological summary called *persistence diagram* or *barcode*, which consists of intervals tracking the birth and death of topological features in the sequence. Some of its nice properties, such as stability and ability to capture global structures of data, have made persistent homology successful in performing various tasks, such as providing topological descriptors for data not easily vectorized [9]–[11], processing images while preserving global shape [12], [13], or regularization in statistical learning [14].

In this dissertation, we study a mathematical object inherent in persistent homology, i.e., its homological *representatives*. While such representatives were implicitly mentioned in previous works [3], [15], [16], we place them in the center stage and study the various aspects concerning the representatives and their implications. One reason leading to our study is that representatives for persistent homology are natural extensions of *generators* for a fixed topological space, which is a classical problem in computational topology and is extensively studied [17]–[19]. There are also other significances for studying the representative problem for persistent homology:

- First and foremost, the representatives provide valuable augmented information in addition to the persistence barcodes. It is known that persistent homology can track the birth and death of homology classes in a growing sequence of topological spaces (i.e., a *filtration*). However, persistent homology does not pinpoint a specific homology class (or a cycle in it) born and died with an interval, which represents the interval. Such a representative for persistent homology can provide important geometrical information or visualizations for the topological spaces, which may be critical in certain tasks (see [20] for a usage of such representatives for image segmentation).

- These representatives may also have important implications for research on other topics in TDA, leading to new findings in the field. For example, Chapter 4 presents near-linear algorithms for computing *zigzag* persistence (an extension of the usual persistent homology) on graphs, which improves the previously known time complexity on graph inputs. Such improvements are made by looking into representatives that persistence algorithms explicitly or implicitly computes [3], [21]. There has also been a recent work [22] on updating persistence barcodes for zigzag filtrations over local changes, which is achieved by a coherent maintenance of representatives. We also see a potential use of the representative maintenance for computing generalized rank invariants [23], [24] for 2-parameter persistence modules, which may help compute homological structures advocated recently [25].

Summary of contributions.

Evolving around the theme of representatives for persistent homology, we study the following problems with various achievements:

- In Chapter 2, we look into the representative problem for (standard) persistence homology and term the representatives as *persistent cycles*. We define persistent cycles as cycles born and dying with given intervals in persistence barcodes, and justify our definition by showing that persistent cycles generate interval decomposition for persistence modules [26]. Besides the definition, we also look into the computation of optimal (minimum) persistent cycles which have guaranteed quality. We first show that it is NP-hard to compute minimum persistent p-cycles for the two types of intervals (i.e., *finite* and *infinite*) in persistent homology in general dimensions ($p > 1$). In view of the NP-hardness results, we then identify a special but important class of inputs called *weak* $(p+1)$*-pseudomanifolds* (Definition 2.2.1) whose minimum persistent p-cycles can be computed in polynomial time. For finite intervals from the p-th persistence barcode of a weak $(p+1)$-pseudomanifold, we utilize the fact that persistent cycles of such intervals are null-homologous and reduce the problem to a minimal cut problem. Since the same problem for infinite intervals on weak $(p+1)$-pseudomanifolds is NP-hard, we further assume the weak $(p+1)$-pseudomanifold to be embedded in $\mathbb{R}^{p+1}$ so that the complex has a natural dual graph structure and the problem reduces to a minimal cut problem.

- One drawback of the standard persistent cycles proposed in Chapter 2 is that only a single cycle born at the start is used, while homological features may vary continuously inside an interval (see Figure 3.1 for an example). In Chapter 3, we propose alternative persistent cycles capturing the dynamic changes of homological features born and dying with persistence intervals. We focus on a special but important type of persistent homology – those generated by piecewise linear (PL) functions [1]. We also base our definition on an extension of standard persistence called the *levelset zigzag* persistence [27], which tracks survival of homological features at and in between the critical points. Given a persistence interval from levelset zigzag, we define a *sequence* of cycles called *levelset persistent cycles* so that there is a cycle between each consecutive critical points within the interval. Thus, a sequence of levelset persistent cycles can capture all variations of homological features in the lifecycle of an interval. Also note that levelset persistent cycles are oriented toward richer types of intervals [27] (see also *extended* persistence [28]).

Due to the NP-hardness of computing minimum persistent cycles presented in Chapter 2, we again look into the computation of an optimal sequence of levelset persistent cycles (one that has the minimum *sum* of weight) for weak pseudomanifolds. Our approaches which also utilize minimum cuts differ from approaches presented in Chapter 2 to account for the fact that a sequence of optimal cycles instead of a single one need to be computed.

- In Chapter 4, we look into the computation of an extension of standard persistent homology called *zigzag* persistence with the help of representatives. Zigzag persistence empowers TDA to deal with filtrations allowing both insertion and deletion of simplices. In practice, allowing deletion of simplices does make the topological tool more powerful (e.g., dynamic networks [29]). Specifically, we focus on the computation of zigzag persistence on a special but important class of inputs, i.e., graphs, because graphs model real-world circumstances in many applications where they may constantly change to capture the dynamic behavior of the phenomena.

  Unlike standard persistence which admits nearly linear-time algorithms for graphs, such results for the zigzag version improving the general $O(m^\omega)$ time complexity [30] are not previously known, where $\omega < 2.37286$ is the matrix multiplication exponent [31]. Our main contributions are algorithms for computing zigzag persistence on graphs which run in near-linear time. Specifically, given a filtration with m additions and deletions on a graph with n vertices and edges, our algorithm for 0-dimension runs in $O(m \log^2 n + m \log m)$ time and our algorithm for 1-dimension runs in $O(m \log^4 n)$ time. The algorithm for 0-dimension draws upon another algorithm [32] designed originally for pairing critical points of Morse functions on 2-manifolds. Besides making connections of the two problems, one achievement for our algorithm for 0-dimension is the correctness proof, which is accomplished with the help of representatives (Definition 4.3.2). The algorithm for 1-dimension finds a pairing of the *positive* and *negative* edges such that representative cycles for all pairs exist. We then utilize a sequence of observations to reduce the pairing to finding the maximum weight of edges on a path in a minimum spanning forest.

2. OPTIMAL REPRESENTATIVES FOR PERSISTENT HOMOLOGY

In this chapter, we study the *optimal representative* problem for persistent homology. We first propose the definition for the representatives, termed as *persistent cycles*, and justify the definition by making a connection to the *interval decomposition* of persistence modules (see Section 2.3). Based on the definition, we then propose to compute the *optimal* persistent cycles, where the optimality means having *minimum* weight so that the optimal cycles provide tightest representations. We also study the hardness for computing optimal persistent cycles by proving that the problem is NP-hard in general dimensions for both the two types of intervals in persistent homology (see Section 2.4 and 2.5). In view of the NP-hardness results, we then identify an important class of simplicial complexes, which are generalizations of $(p+1)$*-manifolds* called *weak* $(p+1)$*-pseudomanifolds* (see Definition 2.2.1), whose minimum persistent p-cycles can be computed in polynomial time (see Section 2.6 and 2.7).

Table 2.1 summarizes findings in this chapter, where PCYC-FIN$_p$ denotes the problem of computing minimum persistent p-cycles for finite intervals given arbitrary simplicial complexes and PCYC-INF$_p$ denotes the same problem for infinite intervals. We also let WPCYC-FIN$_p$ denote a subproblem[1] of PCYC-FIN$_p$ and let WPCYC-INF$_p$, WEPCYC-INF$_p$ denote two subproblems of PCYC-INF$_p$, with the subproblems requiring additional constraints on the given simplicial complex as specified in Table 2.1. Note that the polynomial-time algorithm for PCYC-INF$_1$ was proposed in [20].

Table 2.1. Summary of findings in Chapter 2.

Problem	**Restriction on K**	d	**Hardness**
PCYC-FIN$_p$	–	≥ 1	NP-hard
WPCYC-FIN$_p$	K a weak $(p+1)$-pseudomanifold	≥ 1	Polynomial
PCYC-INF$_p$	–	$= 1$	Polynomial
WPCYC-INF$_p$	K a weak $(p+1)$-pseudomanifold	≥ 2	NP-hard
WEPCYC-INF$_p$	K a weak $(p+1)$-pseudomanifold in $\mathbb{R}^{p+1}$	≥ 2	Polynomial

[1]↑ For two problems P_1 and P_2, P_2 is a *subproblem* of P_1 if any instance of P_2 is an instance of P_1 and P_2 asks for computing the same solutions as P_1.

2.1 Literature review

In the context of computing optimal cycles in computational topology, most works have been done in the non-persistence setting. These works compute minimum cycles for homology groups of a given simplicial complex. Only very few works address the problem while taking into account the persistence. We review some of the relevant works below.

2.1.1 Optimal cycles for homology groups

In terms of computing minimum cycles for homology groups, two problems are of most interest: the *localization* problem and the *minimum basis* problem. The localization problem asks for computing a minimum cycle in a homology class and the minimum basis problem asks for computing a set of generating cycles for a homology group whose sum of weights is minimum. With $\mathbb{Z}_2$ coefficients, these two problems are in general hard. Specifically, Chambers et al. [17] proved that the localization problem over dimension one is NP-hard when the given simplicial complex is a 2-manifold. Chen and Freedman [18] proved that the localization problem is NP-hard to approximate with fixed ratio over arbitrary dimension. They also showed that the minimum basis problem is NP-hard to approximate with fixed ratio over dimension greater than one. For one-dimensional homology, Dey et al. [19] proposed a polynomial time algorithm for the minimum basis problem. Several other works [33]–[36] address variants of the two problems while considering special input classes, alternative cycle measures, or coefficients for homology other than $\mathbb{Z}_2$.

In this work, we use graph cuts and their duality extensively. The duality of cuts on a planar graph and separating cycles on the dual graph has long been utilized to efficiently compute maximum flows and minimum cuts on planar graphs, a topic for which Chambers et al. [17] provide a comprehensive review. In their paper [17], Chambers et al. discover the duality between minimum cuts of a surface-embedded graph and minimum homologous cycles in a dual complex, and then devise $O(n \log n)$ algorithms for both problems assuming the genus of the surface to be fixed. Chen and Freedman [18] proposed an algorithm which computes a minimum non-bounding p-cycle given a $(p+1)$-complex embedded in $\mathbb{R}^{p+1}$, utilizing a natural duality of p-cycles in the complex and cuts in the dual graph. The minimum non-bounding cycle algorithm can be further extended to solve the localization problem and the minimum basis problem over dimension p given a $(p+1)$-complex embedded in $\mathbb{R}^{p+1}$.

2.1.2 Optimal persistent cycles

As pointed out earlier, our main focus is the optimality of representative cycles in the persistence framework. Some early works ([37], [38]) address the representative cycle problem for persistence by computing minimum cycles at the birth points of intervals without considering what actually die at the death points. Wu et al. [16] proposed an algorithm computing minimum persistent 1-cycles for finite intervals using an annotation technique and heuristic search. However, the time complexity of the algorithm is exponential in the worst-case. Obayashi [39] casts the minimum persistent cycle problem for finite intervals into an integer program, but the rounded result of the relaxed linear program is not guaranteed to be optimal.

2.2 Preliminaries

2.2.1 Simplicial complex

A *simplicial complex* K is a collection of *simplices* which are abstractly defined as subsets of a ground set called the *vertex set* of K. If a simplex σ is in K, then all its subsets called its *faces* are also in K. The simplex σ is also referred to as a q-simplex if the cardinality of the vertex set of σ is $q+1$. A *q-face* of σ is a q-simplex being a face of σ and a *q-coface* of σ is a q-simplex having σ as a face. We call a q-simplex of K a *boundary q-simplex* if it has less than two $(q+1)$-cofaces in K. A *simplicial set* is a set of simplices and the *closure* of a simplicial set Σ is the simplicial complex consisting of all the faces of the simplices in Σ. A simplicial complex is *finite* if it contains finitely many simplices. In this chapter, we only consider finite simplicial complexes. We have the following special type of simplicial complexes which are useful to the computation in this chapter.

Definition 2.2.1 (Weak pseudomanifold)**.** *A simplicial complex K is a* weak $(p+1)$-pseudomanifold *if each p-simplex is a face of no more than two $(p+1)$-simplices in K.*

If each vertex of a simplicial complex K is a point in a Euclidean space, then each simplex of K can be interpreted as the convex hull of its vertices. The simplicial complex K is said to be *embedded in* the Euclidean space if the interiors of all its simplices are disjoint. The *underlying space* of K, denoted by $|K|$, is the point-wise union of all the simplices of K.

Definition 2.2.2 (Oriented simplex [40])**.** *A q-simplex with an ordering of its vertices is an* oriented q-simplex*. For each q-simplex σ ($q > 0$), there are exactly two equivalent classes of vertex orderings, resulting in two oriented q-simplices of σ. We refer to them as the* oppositely oriented q-simplices*.*

Remark 2.2.1. Any simplex by default is unoriented. We denote an unoriented q-simplex σ spanned by vertices $v_0, \dots, v_q$ as $\sigma = \{v_0, \dots, v_q\}$ and an oriented q-simplex $\vec{\sigma}$ as $\vec{\sigma} = [v_0, \dots, v_q]$, where $v_0, \dots, v_q$ specify the ordering of the spanning vertices.

2.2.2 Simplicial homology

We provide a brief overview of simplicial homology used in this chapter. See any standard book on the topic, e.g., [40]. Let $q \geq 0$, K be a simplicial complex, and $\mathbb{G}$ be an abelian group. The *q-th chain group* $\mathsf{C}_q(K;\mathbb{G})$ is defined to be the abelian group containing all finite sums of the form $\sum_{\mathrm{i}} n_{\mathrm{i}}\vec{\sigma}_{\mathrm{i}}$, where $n_{\mathrm{i}} \in \mathbb{G}$ and $\vec{\sigma}_{\mathrm{i}}$ is an oriented q-simplex of K. Each element in $\mathsf{C}_q(K;\mathbb{G})$ is called a *q-chain* of K. Note that for two oppositely oriented q-simplices $\vec{\sigma}$ and $\vec{\sigma}'$, we have that $n\vec{\sigma} = (-n)\vec{\sigma}'$ for any $n \in \mathbb{G}$. Therefore, $\mathsf{C}_q(K;\mathbb{G})$ can be interpreted as a direct sum of N_q copies of $\mathbb{G}$ where N_q is the number of q-simplices of K and each copy of $\mathbb{G}$ corresponds to a q-simplex of K. The *q-th boundary operator* $\partial_q : \mathsf{C}_q(K;\mathbb{G}) \to \mathsf{C}_{q-1}(K;\mathbb{G})$ is a group homomorphism such that for any oriented q-simplex $[v_0, \dots, v_q]$

$$\partial_q\Big([v_0, \dots, v_q]\Big) = \sum_{\mathrm{i}=0}^{q}(-1)^{\mathrm{i}}[v_0, \dots, \widehat{v}_{\mathrm{i}}, \dots, v_q]$$

where the notation $[v_0, \dots, \widehat{v}_{\mathrm{i}}, \dots, v_q]$ means that $\widehat{v}_{\mathrm{i}}$ is deleted from the simplex. For brevity, we often omit the subscript of the boundary operator ∂_q and denote it as ∂ when this does not cause any confusion. The kernel of ∂_q is called the *q-th cycle group* of K and is denoted as $\mathsf{Z}_q(K;\mathbb{G})$. The image of ∂_{q+1} is called the *q-th boundary group* of K and is denoted as

$\mathsf{B}_q(K;\mathbb{G})$. A q-chain in $\mathsf{Z}_q(K;\mathbb{G})$ is called a *q-cycle* and a q-chain in $\mathsf{B}_q(K;\mathbb{G})$ is called a *q-boundary*. For a q-chain A, the $(q-1)$-chain $\partial(A)$ is also called the *boundary* of A.

A fundamental fact in homology theory is that $\partial_q\partial_{q+1} = 0$ for any q. This implies that $\mathsf{B}_q(K;\mathbb{G}) \subseteq \mathsf{Z}_q(K;\mathbb{G})$. The *$q$-th homology group* of K denoted by $\mathsf{H}_q(K;\mathbb{G})$ is defined as the quotient $\mathsf{Z}_q(K;\mathbb{G})/\mathsf{B}_q(K;\mathbb{G})$. Each coset in $\mathsf{H}_q(K;\mathbb{G})$ is called a *homology class* and a cycle is said to be *homologous to* another cycle if they belong to the same homology class. As any boundary cycle represents the homology class 0 in $\mathsf{H}_q(K;\mathbb{G})$, a boundary is also said to be *null-homologous*.

The abelian group $\mathbb{G}$ in the above definitions is called the *coefficient group* for the homology groups. Sometimes, when the coefficient group $\mathbb{G}$ is clear, we simply drop it and denote a chain group as $\mathsf{C}_q(K)$. This applies to other groups defined in simplicial homology. In this chapter, two coefficient groups $\mathbb{Z}_2$ and $\mathbb{Z}$ are used for simplicial homology. When not explicitly stated, the coefficients are assumed to be in $\mathbb{Z}_2$. With $\mathbb{Z}_2$ coefficients, the orientations of simplices no longer matter and a q-chain can be interpreted as a set of q-simplices with summation of two q-chains being the *symmetric difference*. A q-cycle is then a set of q-simplices where every $(q-1)$-face of these simplices adjoins an even number of q-simplices. Also note that because $\mathbb{Z}_2$ is a field, all groups defined in simplicial homology with $\mathbb{Z}_2$ coefficients become vector spaces and homomorphisms between these groups (such as ∂) become linear maps.

Definition 2.2.3 (q-weighted)**.** *A simplicial complex K is* q-weighted *if each q-simplex σ of K has a non-negative finite weight $w(\sigma)$. The weight of a q-chain A of K is then defined as $w(A) = \sum_{\sigma\in A} w(\sigma)$.*

Definition 2.2.4 (q-connected)**.** *Let Σ be a set of simplices, and let σ, σ' be two q-simplices of Σ for $q \geq 1$. A q-path from σ to σ' in Σ is a sequence of q-simplices of Σ, $\tau_1, \ldots, \tau_\ell$, such that $\tau_1 = \sigma$, $\tau_\ell = \sigma'$, and each consecutive τ_{i}, $\tau_{\mathrm{i}+1}$ share a $(q-1)$-face in Σ. A maximal set of q-simplices of Σ, in which each pair is connected by a q-path, constitutes a q-connected component of Σ. We also say that Σ is q-connected if it has only one q-connected component.*

Remark 2.2.2. See Figure 2.3a for an example of 1-connected components and 2-connected components.

Definition 2.2.5 (q-connected cycle)**.** *A q-cycle ζ (with $\mathbb{Z}_2$ coefficients) is* q-connected *if the complex derived by taking the closure of the simplicial set ζ is q-connected.*

2.2.3 Filtration

A *filtration* $\mathcal{F}$ of a simplicial complex K is a filtered sequence of subcomplexes of K,

$$\mathcal{F} : \varnothing = K_0 \subseteq K_1 \subseteq \ldots \subseteq K_n = K,$$

such that K_i and $K_{\mathrm{i}-1}$ differ by one simplex denoted by $\sigma^{\mathcal{F}}_\mathrm{i}$. We let i be the *index* of $\sigma^{\mathcal{F}}_\mathrm{i}$ in $\mathcal{F}$ and denote it as $\mathrm{ind}(\sigma^{\mathcal{F}}_\mathrm{i}) = \mathrm{i}$. A subcomplex K_i in the filtered sequence of $\mathcal{F}$ is also referred to as a *partial complex.*

2.2.4 Persistent homology

We will provide a brief description of persistent homology. We recommend the book by Edelsbrunner and Harer [1] for a detailed explanation of this topic. Note that persistent homology in this chapter is always assumed to be with $\mathbb{Z}_2$ coefficients. The persistence algorithm starts with a filtration $\mathcal{F} : \varnothing = K_0 \subseteq K_1 \subseteq \ldots \subseteq K_n = K$ of a simplicial complex K, and for each simplex $\sigma^{\mathcal{F}}_\mathrm{i}$, inspects whether $\partial(\sigma^{\mathcal{F}}_\mathrm{i})$ is a boundary in $K_{\mathrm{i}-1}$. If $\partial(\sigma^{\mathcal{F}}_\mathrm{i})$ is a boundary in $K_{\mathrm{i}-1}$, $\sigma^{\mathcal{F}}_\mathrm{i}$ is called *positive*; otherwise, it is called *negative.* The d-chains (or d-cycles) in K_i that are not in $K_{\mathrm{i}-1}$ are said to be *born in* K_i or *created by* $\sigma^{\mathcal{F}}_\mathrm{i}$. A positive d-simplex creates some d-cycles and a negative d-simplex makes some $(d-1)$-cycles become boundaries. In the latter case, we also say that the negative d-simplex *kills* or *destroys* those $(d-1)$-cycles. What is central to the persistence algorithm is a notion called *pairing*: A positive simplex is initially *unpaired* when introduced; when a negative d-simplex $\sigma^{\mathcal{F}}_\mathrm{i}$ comes, the algorithm finds a $(d-1)$-cycle created by an unpaired positive $(d-1)$-simplex $\sigma^{\mathcal{F}}_\mathrm{j}$ which is homologous to $\partial(\sigma^{\mathcal{F}}_\mathrm{i})$ and *pair* $\sigma^{\mathcal{F}}_\mathrm{j}$ *with* $\sigma^{\mathcal{F}}_\mathrm{i}$. Alongside the pairing, a *finite interval* $[\mathrm{j}, \mathrm{i})$ is added to the $(d-1)$-th *persistence diagram* (also called *barcode*), which is denoted by $\mathsf{Pers}_{d-1}(\mathcal{F})$. After all simplices are processed, some positive simplices may still be unpaired.

For each $\sigma_{\mathrm{i}}^{\mathcal{F}}$ of these unpaired simplices, an *infinite interval* $[\mathrm{i}, +\infty)$ is added to $\mathsf{Pers}_d(\mathcal{F})$, where d is the dimension of $\sigma_{\mathrm{i}}^{\mathcal{F}}$.

Note that the pairing in the persistence algorithm for a given filtration is unique. Also note that in this chapter, we assume a filtration of a complex is given and the persistence intervals start and end with indices of the paired simplices. However, in real-life applications, one is often given a function on a simplicial complex. To produce the persistence intervals, a filtration needs to be derived and the endpoints of the intervals are taken as function values on the paired simplices. In such cases, we can associate a given interval to its simplex pair, take the indices of the paired simplices, and get an interval which can serve as an input to our algorithms.

2.2.5 Persistence module

In this chapter, we adopt the *categorical* definition of persistence module [41]. A *category* C consists of *objects* and *morphisms* from an object to another object. A *functor* $F : C \to B$ from C to another category B is a mapping such that any object c of C is mapped to an object $F(c)$ of B and any morphism $f : c \to c'$ of C is mapped to a morphism $F[f] : F(c) \to F(c')$ of B. We recommend [42] for detailed definitions of categories and functors. The definition of persistence module relies on some common categories: The category $\mathbb{Z}^+$ (the category $\{1, \dots, n\}$ alike) consists of objects from $\mathbb{Z}^+$ and a unique morphism from i to j if $\mathrm{i} \leq \mathrm{j}$. We also denote the morphism from i to j as $\mathrm{i} \leq \mathrm{j}$. The category **Simp** consists of objects which are all the simplicial complexes and morphisms which are simplicial maps. The category **Vec** consists of objects which are all the vector spaces over $\mathbb{Z}_2$ and morphisms which are linear maps. A *persistence module* $\mathcal{P}$ is then defined as a functor $\mathcal{P} : \mathbb{Z}^+ \to \mathbf{Vec}$[2].

A persistence module is usually induced by a filtration $\mathcal{F} : \varnothing = K_0 \subseteq K_1 \subseteq \ldots \subseteq K_m = K$ of a simplicial complex K. We can also interpret a filtration $\mathcal{F}$ as a functor $\mathcal{F} : \mathbb{Z}^+ \to \mathbf{Simp}$, where $\mathcal{F}(\mathrm{i}) = K_{\mathrm{i}}$ for $\mathrm{i} \leq m$, $\mathcal{F}(\mathrm{i}) = K$ for $\mathrm{i} > m$, and a morphism $\mathcal{F}[\mathrm{i} \leq \mathrm{j}] : \mathcal{F}(\mathrm{i}) \to \mathcal{F}(\mathrm{j})$ is the inclusion. Denoting $\mathsf{H}_q : \mathbf{Simp} \to \mathbf{Vec}$ as the q-th homology functor with $\mathbb{Z}_2$ coefficients, the *q-th persistence module* $\mathcal{P}_q^{\mathcal{F}}$ of $\mathcal{F}$ is obtained by composing

[2]↑Sometimes we also call a functor $\mathcal{P} : \{1, \dots, n\} \to \mathbf{Vec}$ as a persistence module.

the two functors H_q and $\mathcal{F}$, that is, $\mathcal{P}_q^{\mathcal{F}} = \mathsf{H}_q\mathcal{F}$. Specifically, $\mathcal{P}_q^{\mathcal{F}}(\mathrm{i}) = \mathsf{H}_q(K_\mathrm{i})$ for $\mathrm{i} \leq m$, $\mathcal{P}_q^{\mathcal{F}}(\mathrm{i}) = \mathsf{H}_q(K)$ for $\mathrm{i} > m$, and the morphism $\mathcal{P}_q^{\mathcal{F}}[\mathrm{i} \leq \mathrm{j}] : \mathsf{H}_q(K_\mathrm{i}) \to \mathsf{H}_q(K_\mathrm{j})$[3] is the linear map induced by the inclusion.

A special class of persistence modules are the *interval modules*. Given an interval $[b, d) \subset \mathbb{Z}^+$, an interval module $\mathcal{I}^{[b,d)}$ is defined as: $\mathcal{I}^{[b,d)}(\mathrm{i}) = \mathbb{Z}_2$ for $\mathrm{i} \in [b, d)$ and $\mathcal{I}^{[b,d)}(\mathrm{i}) = 0$ otherwise; $\mathcal{I}^{[b,d)}[\mathrm{i} \leq \mathrm{j}]$ is the identity map for $\mathrm{i}, \mathrm{j} \in [b, d)$ and $\mathcal{I}^{[b,d)}[\mathrm{i} \leq \mathrm{j}]$ is the zero map otherwise. By quiver theory, a q-th persistence module obtained from a finite complex K has a unique decomposition $\mathcal{P}_q^{\mathcal{F}} \approx \bigoplus_{\mathrm{j}\in J} \mathcal{I}^{[b_\mathrm{j},d_\mathrm{j})}$ in terms of interval modules, where J is a finite index set [26]. Then, the *persistence diagram* (or *barcode*) of $\mathcal{P}_q^{\mathcal{F}}$ is $\mathsf{Pers}(\mathcal{P}_q^{\mathcal{F}}) = \{[b_\mathrm{j}, d_\mathrm{j}) \mid \mathrm{j} \in J\}$, which is the set of intervals for the interval modules which $\mathcal{P}_q^{\mathcal{F}}$ decomposes into. Note that $\mathsf{Pers}(\mathcal{P}_q^{\mathcal{F}})$ is indeed the $\mathsf{Pers}_q(\mathcal{F})$ defined in Section 2.2.4.

2.2.6 Undirected flow network

An *undirected flow network* (G, s_1, s_2) consists of an undirected graph G with vertex set $V(G)$ and edge set $E(G)$, a capacity function $c : E(G) \to [0, +\infty]$, and two non-empty disjoint subsets s_1 and s_2 of $V(G)$. Vertices in s_1 are referred to as *sources* and vertices in s_2 are referred to as *sinks*. A *cut* (S, T) of (G, s_1, s_2) consists of two disjoint subsets S and T of $V(G)$ such that $S \cup T = V(G)$, $s_1 \subseteq S$, and $s_2 \subseteq T$. The set of edges that connect a vertex in S and a vertex in T are referred as the edges *across* the cut (S, T) and is denoted as $\xi(S, T)$. The *capacity* of a cut (S, T) is defined as $c(S, T) = \sum_{\mathrm{e}\in\xi(S,T)} c(\mathrm{e})$. A *minimum cut* of (G, s_1, s_2) is a cut with the minimum capacity. Note that we allow parallel edges in G (see Figure 2.3a) to ease the presentation. These parallel edges can be merged into one edge during computation.

2.3 Definition of persistent cycles

In this section, we first provide a definition of *persistent basis*, which is then utilized to introduce and justify the definition of persistent cycles.

[3]↑$K_\mathrm{j} = K$ when $\mathrm{j} > m$.

Definition 2.3.1 (Persistent basis). *An indexed set of q-cycles $\{c_{\mathrm{j}} \mid \mathrm{j} \in J\}$ is called a* persistent q-basis *for a filtration $\mathcal{F}$ if $\mathcal{P}_q^{\mathcal{F}} = \bigoplus_{\mathrm{j}\in J} \mathcal{I}^{[b_{\mathrm{j}},d_{\mathrm{j}})}$ and for each $\mathrm{j} \in J$ and $b_{\mathrm{j}} \leq k < d_{\mathrm{j}}$, $\mathcal{I}^{[b_{\mathrm{j}},d_{\mathrm{j}})}(k) = \{0, [c_{\mathrm{j}}]\}$.*

Definition 2.3.2 (Persistent cycle). *Let $\mathcal{F}$ be a filtration. For an interval $[b,d) \in \mathsf{Pers}_q(\mathcal{F})$, a q-cycle c is called a* persistent q-cycle *for the interval, if one of the following holds:*

- *$d \neq +\infty$, c is a cycle in K_b containing $\sigma_b^{\mathcal{F}}$, and c is not a boundary in K_{d-1} but becomes a boundary in K_d;*
- *$d = +\infty$ and c is a cycle in K_b containing $\sigma_b^{\mathcal{F}}$.*

Remark 2.3.1. Note that the definition of persistent cycles for finite intervals is identical to that of [16], [39].

The following theorem characterizes each cycle in a persistent basis:

Theorem 2.3.1. *An indexed set of q-cycles $\{c_{\mathrm{j}} \mid \mathrm{j} \in J\}$ is a persistent q-basis for a filtration $\mathcal{F}$ if and only if $\mathcal{P}_q^{\mathcal{F}} \approx \bigoplus_{\mathrm{j}\in J} \mathcal{I}^{[b_{\mathrm{j}},d_{\mathrm{j}})}$ and c_{j} is a persistent q-cycle for every interval $[b_{\mathrm{j}}, d_{\mathrm{j}}) \in \mathsf{Pers}(\mathcal{P}_q^{\mathcal{F}})$.*

Proof. Suppose $\{c_{\mathrm{j}} \mid \mathrm{j} \in J\}$ is an indexed set of q-cycles satisfying the above conditions. For each $\mathrm{j} \in J$, we construct an interval module $\mathcal{I}_{\mathrm{j}}$, such that $\mathcal{I}_{\mathrm{j}}(\mathrm{i}) = \{0, [c_{\mathrm{j}}]\}$ for $b_{\mathrm{j}} \leq \mathrm{i} < d_{\mathrm{j}}$ and $\mathcal{I}_{\mathrm{j}}(\mathrm{i}) = 0$ otherwise. We claim that $\mathcal{P}_q^{\mathcal{F}} = \bigoplus_{\mathrm{j}\in J} \mathcal{I}_{\mathrm{j}}$. We first prove that $\mathcal{P}_q^{\mathcal{F}}(\mathrm{i}) = \bigoplus_{\mathrm{j}\in J} \mathcal{I}_{\mathrm{j}}(\mathrm{i})$ for each $\mathrm{i} \in \mathbb{Z}^+$, by proving that $\{[c_{\mathrm{j}}] \mid \mathrm{j} \in J, \mathrm{i} \in [b_{\mathrm{j}}, d_{\mathrm{j}})\}$ forms a basis of $\mathcal{P}_q^{\mathcal{F}}(\mathrm{i})$. Using mathematical induction, since $\sigma_1^{\mathcal{F}}$ is a vertex, this is trivially true. Suppose for $\mathrm{i} - 1$ this is true. If $\sigma_{\mathrm{i}}^{\mathcal{F}}$ is neither positive nor negative, i.e., $\mathsf{H}_q(K_{\mathrm{i}-1}) \approx \mathsf{H}_q(K_{\mathrm{i}})$ by the isomorphism induced from the inclusion, this is also trivially true for i. If $\sigma_{\mathrm{i}}^{\mathcal{F}}$ is positive, suppose the corresponding interval of $\sigma_{\mathrm{i}}^{\mathcal{F}}$ is $[b_{\mathrm{j}'}, d_{\mathrm{j}'})$ (note that $b_{\mathrm{j}'} = \mathrm{i}$ and $d_{\mathrm{j}'}$ could possibly be $+\infty$). Since $\{[c_{\mathrm{j}}] \mid \mathrm{j} \in J, \mathrm{i} - 1 \in [b_{\mathrm{j}}, d_{\mathrm{j}})\}$ are still independent in $\mathcal{P}_q^{\mathcal{F}}(\mathrm{i})$ and $[c_{\mathrm{j}'}]$ is not in the span of them, then $\{[c_{\mathrm{j}}] \mid \mathrm{j} \in J, \mathrm{i} - 1 \in [b_{\mathrm{j}}, d_{\mathrm{j}})\} \cup [c_{\mathrm{j}'}] = \{[c_{\mathrm{j}}] \mid \mathrm{j} \in J, \mathrm{i} \in [b_{\mathrm{j}}, d_{\mathrm{j}})\}$ are independent in $\mathcal{P}_q^{\mathcal{F}}(\mathrm{i})$. Since the cardinality of $\{[c_{\mathrm{j}}] \mid \mathrm{j} \in J, \mathrm{i} \in [b_{\mathrm{j}}, d_{\mathrm{j}})\}$ equals the dimension of $\mathcal{P}_q^{\mathcal{F}}(\mathrm{i})$, it must form a basis of $\mathcal{P}_q^{\mathcal{F}}(\mathrm{i})$. If $\sigma_{\mathrm{i}}^{\mathcal{F}}$ is negative, then there must be a $[c_{\mathrm{j}'}]$ for a $\mathrm{j}' \in J$ such that $d_{\mathrm{j}'} = \mathrm{i}$. For any $[c] \in \mathcal{P}_q^{\mathcal{F}}(\mathrm{i}) = \mathsf{H}_q(K_{\mathrm{i}})$, $[c] = \sum_{\mathrm{j}\in J'} [c_{\mathrm{j}}]$, where $J' \subseteq \{\mathrm{j} \in J \mid \mathrm{i} - 1 \in [b_{\mathrm{j}}, d_{\mathrm{j}})\}$.

If $\mathrm{j}' \in J'$, then $[c] = \sum_{\mathrm{j} \in J' - \{\mathrm{j}'\}} [c_\mathrm{j}]$, because $[c_{\mathrm{j}'}] = 0$ in $\mathsf{H}_q(K_\mathrm{i})$. Then $\{[c_\mathrm{j}] \mid \mathrm{j} \in J, \mathrm{i} - 1 \in [b_\mathrm{j}, d_\mathrm{j})\} - \{c_{\mathrm{j}'}\} = \{[c_\mathrm{j}] \mid \mathrm{j} \in J, \mathrm{i} \in [b_\mathrm{j}, d_\mathrm{j})\}$ spans $\mathsf{H}_q(K_\mathrm{i})$. This means that it also forms a basis of $\mathsf{H}_q(K_\mathrm{i})$. It is then obvious that the direct sums of the maps of the interval modules are actually the maps of $\mathcal{P}_q^\mathcal{F}$, so $\{c_\mathrm{j} \mid \mathrm{j} \in J\}$ is a persistent q-basis for $\mathcal{F}$.

Suppose $\{c_\mathrm{j} \mid \mathrm{j} \in J\}$ is a persistent q-basis for $\mathcal{F}$. For each $\mathrm{j} \in J$, c_j must not be in $K_{b_\mathrm{j}-1}$, because otherwise $[c_\mathrm{j}]$ would be in the image of $\mathcal{P}_q^\mathcal{F}[b_\mathrm{j} - 1 \leq b_\mathrm{j}]$. It is obvious that c_j must contain $\sigma_\mathrm{j}^\mathcal{F}$. Note that for each $\mathrm{j} \in J$ and each $\mathrm{i} \in [b_\mathrm{j}, d_\mathrm{j})$, $\mathcal{P}_q^\mathcal{F}[\mathrm{i} \leq \mathrm{i}+1]([c_\mathrm{j}]) = \mathcal{I}^{[b_\mathrm{j}, d_\mathrm{j})}[\mathrm{i} \leq \mathrm{i}+1]([c_\mathrm{j}])$. Then for each $\mathrm{j} \in J$ such that $d_\mathrm{j} \neq +\infty$, $[c_\mathrm{j}] \neq 0$ in $K_{d_\mathrm{j}-1}$ and $[c_\mathrm{j}] = 0$ in K_{d_j}. □

Remark 2.3.2. With Definition 2.3.2 and Theorem 2.3.1, it is true that for a persistent q-cycle c of an interval $[b, d) \in \mathsf{Pers}_q(\mathcal{F})$, we can always form an interval module decomposition of $\mathcal{P}_q^\mathcal{F}$, where c is a representative cycle for the interval module of $[b, d)$.

2.3.1 Minimum persistent cycle problems

We can now formally define the minimum persistent cycle problems:

Problem 2.3.1 (PCYC-FIN$_p$)**.** *Given a finite p-weighted simplicial complex K, a filtration $\mathcal{F} : \varnothing = K_0 \subseteq K_1 \subseteq \ldots \subseteq K_n = K$, and a finite interval $[b, d) \in \mathsf{Pers}_p(\mathcal{F})$, this problem asks for computing a p-cycle with the minimum weight which is born in K_b and becomes a boundary in K_d.*

Problem 2.3.2 (PCYC-INF$_p$)**.** *Given a finite p-weighted simplicial complex K, a filtration $\mathcal{F} : \varnothing = K_0 \subseteq K_1 \subseteq \ldots \subseteq K_n = K$, and an infinite interval $[b, +\infty) \in \mathsf{Pers}_p(\mathcal{F})$, this problem asks for computing a p-cycle with the minimum weight which is born in K_b.*

2.4 NP-hardness for finite intervals

2.4.1 NP-hardness in dimension one

We first prove that the PCYC-FIN$_p$ problem is NP-hard when $p = 1$. The proof is done by showing a special version (see definition below) of PCYC-FIN$_1$ is NP-hard. This special version reduces to the general version straightforwardly in polynomial time by assigning every edge a weight of 1.

Problem 2.4.1 (LST-PERS-CYC)**.** *Given a filtration $\mathcal{F} : \varnothing = K_0 \subseteq K_1 \subseteq \ldots \subseteq K_m = K$, and a finite interval $[b, d) \in \mathsf{Pers}_1(\mathcal{F})$, this problem asks for finding a 1-cycle with the least number of edges which is born in K_b and becomes a boundary in K_d.*

Theorem 2.4.1. *The problem LST-PERS-CYC is NP-hard.*

To prove Theorem 2.4.1, we reduce the NP-hard MAX-2SAT [43] problem to LST-PERS-CYC, which is similar as in [44]. The MAX-2SAT problem is defined as:

Problem 2.4.2 (MAX-2SAT)**.** *Given N variables $x_1, \ldots, x_N$ and M clauses $c_1, \ldots, c_M$, with the clauses being the disjunction of at most two variables. Find an assignment of Boolean values to all the variables such that the maximal number of clauses are satisfied.*

Proof of Theorem 2.4.1. We will reduce MAX-2SAT to LST-PERS-CYC. Given an instance of MAX-2SAT, we first construct a simplicial complex K as in [44], by forming a triangulated cylinder C_i for each variable x_i and a cycle w_j for each clause c_j, such that the two ends z_i and z'_i of C_i correspond to x_i and $\neg x_\mathrm{i}$ and are the only two cycles with the least number of edges of their homology class in C_i. To make the process clearer, our construction of the cycles z_i, z'_i, and w_j are a little different from [44]. Each z_i or z'_i has $3M$ edges and M of them are bijectively assigned to the M clauses, such that in between each two consecutive edges assigned to some clauses, there are two edges which are not assigned to any clause. For a clause cycle $w_\mathrm{j} = (z'_\mathrm{i}, z_k)$ (do the similar for other cases), we assign three edges to w_j and pick one edge to be shared with the edges in z'_i and z_k assigned to w_j. Let $\overline{z} = \sum_{\mathrm{i}=1}^{N} z_\mathrm{i} + \sum_{\mathrm{j}=1}^{M} w_\mathrm{j}$, then our construction will make it true that, there is an assignment of Boolean values making k clauses satisfied if and only if there is a cycle in $[\overline{z}]$ with $3MN + 3M - 2k$ edges.

Next we are going to construct a filtration $\mathcal{F}'$ of a complex K', where $K \subseteq K'$. We first construct a filtration $\mathcal{F}$ of K, with the only restriction: Pick an edge e of a clause cycle, which is not shared with any end cycle of the variable cylinders, and take e as the last simplex added to the filtration. To construct $\mathcal{F}'$ and K', we need to find all simple cycles of $\overline{z}$. A simple cycle is defined as a cycle such that, each vertex has degree 2 and there is only one connected component in the cycle. We can use a DFS-based algorithm to find all simple cycles for $\overline{z}$: Treat $\overline{z}$ as graph and run DFS on the graph. Find a non-DFS-tree edge

(v_1, v_2) of $\overline{z}$, then find the lowest common ancestor w of v_1 and v_2 in the DFS tree. The paths in the DFS tree from w to v_1 and w to v_2, plus the edge (v_1, v_2), form a simple cycle of $\overline{z}$. Delete the simple cycle from the graph and repeat the above process until the graph becomes empty.

For each simple cycle $\overline{c}$ of $\overline{z}$, we attach a cylinder $\overline{C}$ to $\overline{c}$ such that, one end of $\overline{C}$ is $\overline{c}$, the other end of $\overline{C}$ is a quadrilateral, and all the other edges of $\overline{C}$ connect $\overline{c}$ to the quadrilateral. An example of such a cylinder connecting a dodecagon and a quadrilateral is illustrated in Figure 2.1a. After all the cylinders are attached to the simple cycles, we get a simplicial complex K_1. We can append the simplices of $K_1 \smallsetminus K$ to $\mathcal{F}'$, to get a filtration $\mathcal{F}_1$ of K_1. Since K_1 deformation retracts onto K, all negative triangles of $K_1 \smallsetminus K$ are paired with an edge of $K_1 \smallsetminus K$. We then construct a simplicial complex whose boundary is the sum of all the quadrilaterals and an outer cycle c', as in Figure 2.1b, and attach this simplicial complex to K_1 by gluing the quadrilaterals, to get a simplicial complex K_2. To form a filtration $\mathcal{F}_2$ of K_2, we first append the red edges in Figure 2.1b to $\mathcal{F}_1$, then append all the other simplices of $K_2 \smallsetminus K_1$. Finally, we form a cone around c' to get K' and append the missing simplices to get the filtration $\mathcal{F}'$.

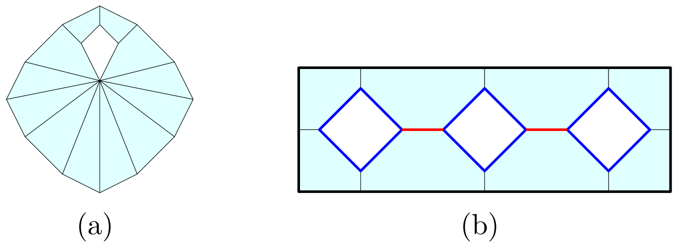


Figure 2.1. (a) A cylinder connecting a dodecagon and a quadrilateral. (b) A simplicial complex whose boundary is the sum of three quadrilaterals (blue) and an outer cycle (bold). Some polygons in the figure are not triangulated.

Let t be the last triangle in $\mathcal{F}'$, then it is true that $K' \smallsetminus t$ deformation retracts to the union of K_1 and the red edges. This indicates that all negative triangles of $K' \smallsetminus K_1$, other than t, are paired with edges of $K' \smallsetminus K_1$. Let the index of e in $\mathcal{F}'$ be b and the index of t in $\mathcal{F}'$ be d, we claim that $[b, d)$ is an interval of $\mathsf{Pers}_1(\mathcal{F}')$. To prove this, first note that $\overline{z}$ is born in K_b and becomes a boundary in K_d. By the time t is added, e is unpaired. So by the persistence algorithm [3], t must be paired with e.

Now we have constructed an instance of LST-PERS-CYC, from an instance of MAX-2SAT: Given the filtration $\mathcal{F}'$ and the interval $[b, d) \in \mathsf{Pers}_1(\mathcal{F}')$, find a persistent 1-cycle with the least number of edges. We then prove that the answer to LST-PERS-CYC is also the answer to MAX-2SAT. First note that the map $\mathsf{H}_1(K_b) \to \mathsf{H}_1(K_{d-1})$ is injective. This means that any persistent 1-cycle for $[b, d)$ must be homologous to $\overline{z}$ in K_b, as they are homologous in K_{d-1}. It follows that computing the minimum persistent 1-cycle of $[b, d)$ is equivalent to computing the minimum cycle of the homology class $[\overline{z}]$ in K_b, which is in turn equivalent to computing the answer for the original MAX-2SAT problem. Then we have had a reduction from MAX-2SAT to LST-PERS-CYC. Furthermore, the reduction is in polynomial time and the size of the constructed instance of LST-PERS-CYC is a polynomial function of that of MAX-2SAT, so LST-PERS-CYC is NP-hard. $\square$

2.4.2 Suspension operator

Similar to the work [18], the NP-hardness proofs for general dimensions accomplish the reduction with the help of a suspension operator. While Hatcher [2] defines this operator for general topological spaces, we need a definition of the operator for simplicial complexes and observe some of its properties that are useful for the proofs.

Definition 2.4.1 (Suspension [45])**.** *The* suspension $\mathcal{S}K$ *of a simplicial complex* K *is defined as a simplicial complex*

$$\mathcal{S}K = \Big\{\{\omega_1\}, \{\omega_2\}\Big\} \cup K \cup \Big(\bigcup_{\sigma\in K} \Big\{\sigma \cup \{\omega_1\}, \sigma \cup \{\omega_2\}\Big\}\Big)$$

where ω_1*,* ω_2 *are two extra vertices.*

Remark 2.4.1. In the above definition, we denote a simplex by its set of vertices.

In the rest of this subsection, we let K be an arbitrary simplicial complex. Any simplex of the form $\sigma \cup \{\omega_i\}$ in $\mathcal{S}K$ is called a *suspended simplex.* The symbol $\mathcal{S}$ is also used to denote a linear map $\mathcal{S} : \mathsf{C}_q(K) \to \mathsf{C}_{q+1}(\mathcal{S}K)$, where $\mathcal{S}\sigma = \sigma \cup \{\omega_1\} + \sigma \cup \{\omega_2\}$ for any q-simplex σ of K. Note that since $\mathcal{S}$ is injective, the map $\mathcal{S}$ defines an isomorphism from $\mathsf{C}_q(K)$ to the image $\mathcal{S}(\mathsf{C}_q(K))$. For any chain $A \in \mathcal{S}(\mathsf{C}_q(K))$, we abuse the notation slightly by letting $\mathcal{S}^{-1}A$ denote the chain in $\mathsf{C}_q(K)$ mapped to A under $\mathcal{S}$.

Proposition 2.4.1. *For any $q \geq 1$, the following diagram commutes:*

$$\begin{tikzcd}
\mathsf{C}_q(K) \arrow[r, "\partial"] \arrow[d, "\mathcal{S}"', "\approx"] & \mathsf{C}_{q-1}(K) \arrow[d, "\approx"', "\mathcal{S}"] \\
\mathcal{S}(\mathsf{C}_q(K)) \arrow[r, "\partial"] & \mathcal{S}(\mathsf{C}_{q-1}(K))
\end{tikzcd}$$

Proof. For any q-simplex $\sigma = \{v_0, \dots, v_q\}$ of K, we have

$$\begin{aligned}
\partial(\mathcal{S}\sigma) &= \partial\Big(\{v_0, \dots, v_q, \omega_1\} + \{v_0, \dots, v_q, \omega_2\}\Big) \\
&= \sum_{\mathrm{i}=0}^{q} \{v_0, \dots, \widehat{v_\mathrm{i}}, \dots, v_q, \omega_1\} + \{v_0, \dots, v_q\} + \sum_{\mathrm{i}=0}^{q} \{v_0, \dots, \widehat{v_\mathrm{i}}, \dots, v_q, \omega_2\} + \{v_0, \dots, v_q\} \\
&= \sum_{\mathrm{i}=0}^{q} \Big(\{v_0, \dots, \widehat{v_\mathrm{i}}, \dots, v_q, \omega_1\} + \{v_0, \dots, \widehat{v_\mathrm{i}}, \dots, v_q, \omega_2\}\Big) \\
&= \sum_{\mathrm{i}=0}^{q} \mathcal{S}\Big(\{v_0, \dots, \widehat{v_\mathrm{i}}, \dots, v_q\}\Big) = \mathcal{S}\left(\sum_{\mathrm{i}=0}^{q} \{v_0, \dots, \widehat{v_\mathrm{i}}, \dots, v_q\}\right) = \mathcal{S}\partial(\sigma)
\end{aligned}$$

In the above equations, the notation $\widehat{v_\mathrm{i}}$ means that v_i is deleted from the simplex. ☐

Proposition 2.4.2. *For $q \geq 1$ and any q-cycle ζ of $\mathcal{S}K$ containing only suspended simplices, one has $\zeta \in \mathcal{S}(\mathsf{C}_{q-1}(K))$.*

Proof. For any suspended q-simplex $\sigma \cup \{\omega_\mathrm{i}\}$ of ζ, if $\omega_\mathrm{i} = \omega_1$, then $\sigma \cup \{\omega_2\}$ must also belong to ζ because no other suspended q-simplices of $\mathcal{S}K$ have σ in the boundary. If $\omega_\mathrm{i} = \omega_2$, the same argument follows. ☐

Proposition 2.4.3. *If q is the top dimension of K and $q \geq 1$, then for any $A \in \mathsf{C}_{q+1}(\mathcal{S}K)$ such that $\partial(A)$ contains only suspended simplices, one has $A \in \mathcal{S}(\mathsf{C}_q(K))$.*

Proof. Because q is the top dimension of K, A contains only suspended simplices. For any $\sigma \cup \{\omega_\mathrm{i}\} \in A$, we have $\sigma \in \partial\Big(\sigma \cup \{\omega_\mathrm{i}\}\Big)$. If $\omega_\mathrm{i} = \omega_1$, to make σ cancelled in $\partial(A)$, $\sigma \cup \{\omega_2\}$ must also belong to A because no other $(q+1)$-simplices in $\mathcal{S}K$ have σ in the boundary. If $\omega_\mathrm{i} = \omega_2$, the same argument follows. ☐

2.4.3 NP-hardness in general dimensions

The following proposition helps to prove the NP-hardness in general dimensions:

Proposition 2.4.4. *PCYC-FIN$_{p-1}$ reduces to PCYC-FIN$_p$ for $p \geq 2$.*

Proof. Given an instance $(K, \mathcal{F}, [b, d))$ of PCYC-FIN$_{p-1}$, where the i_{th} complex of $\mathcal{F}$ is denoted as K_{i}, we can assume the top dimension of K to be p. The reason is that if it were not, we can restrict $\mathcal{F}$ to the p-skeleton of K without affecting $\mathsf{Pers}_{p-1}(\mathcal{F})$ and the persistent $(p-1)$-cycles. Then, we let $\mathcal{S}K$ be the simplicial complex for the instance of PCYC-FIN$_p$ we are going to construct. For any suspended p-simplex $\sigma \cup \{\omega_{\mathrm{i}}\}$ of $\mathcal{S}K$, let the weight of $\sigma \cup \{\omega_{\mathrm{i}}\}$ be half of the weight of σ in K. Furthermore, let the weight of any non-suspended p-simplex of $\mathcal{S}K$ be the sum of all the weights of $(p-1)$-simplices in K plus 1. We endow $\mathcal{S}K$ with a filtration $\mathcal{S}\mathcal{F} : \varnothing = \widehat{K}_0 \subseteq \widehat{K}_1 \subseteq \ldots \subseteq \widehat{K}_{3n+2} = \mathcal{S}K$, where n is the number of simplices of K. Denoting the i_{th} simplex added in $\mathcal{F}$ as σ_{i} and the i_{th} simplex added in $\mathcal{S}\mathcal{F}$ as $\widehat{\sigma}_{\mathrm{i}}$, we let $\widehat{\sigma}_1 = \{\omega_1\}$, $\widehat{\sigma}_2 = \{\omega_2\}$, and for any $1 \leq \mathrm{i} \leq n$, $\widehat{\sigma}_{3\mathrm{i}} = \sigma_{\mathrm{i}}$, $\widehat{\sigma}_{3\mathrm{i}+1} = \sigma_{\mathrm{i}} \cup \{\omega_1\}$, $\widehat{\sigma}_{3\mathrm{i}+2} = \sigma_{\mathrm{i}} \cup \{\omega_2\}$.

We observe the following facts:

(i). For any i, $\widehat{\sigma}_{3\mathrm{i}}$ is positive and pairs with $\widehat{\sigma}_{3\mathrm{i}+1}$ in $\mathcal{S}\mathcal{F}$.

(ii). For any i and j, if there is a $(p-1)$-cycle created by σ_{i} which is a boundary in K_{j}, then there is a p-cycle created by $\widehat{\sigma}_{3\mathrm{i}+2}$ which is a boundary in $\widehat{K}_{3\mathrm{j}+2}$.

(iii). For any i and j, if there is a p-cycle created by $\widehat{\sigma}_{3\mathrm{i}+2}$ which is a boundary in $\widehat{K}_{3\mathrm{j}+2}$, then there is a $(p-1)$-cycle created by σ_{i} which is a boundary in K_{j}.

The correctness of (i) is not hard to verify. To verify (ii), we can suspend the $(p-1)$-cycle and use Proposition 2.4.1 to reach the claim. The argument for (iii) is as follows: Consider a p-cycle $\widehat{\zeta}_0$ created by $\widehat{\sigma}_{3\mathrm{i}+2}$ which is a boundary in $\widehat{K}_{3\mathrm{j}+2}$. For any non-suspended p-simplex σ of $\widehat{\zeta}_0$, we add $\partial\big(\sigma \cup \{\omega_1\}\big)$ to the cycle $\widehat{\zeta}_0$ so that σ is canceled and only suspended simplices are added. Note that the adding process only adds p-simplices in $\widehat{K}_{3\mathrm{i}+2}$ and never cancels $\widehat{\sigma}_{3\mathrm{i}+2}$. After all non-suspended simplices of $\widehat{\zeta}_0$ are canceled, we derive a p-cycle $\widehat{\zeta}$ which is created by $\widehat{\sigma}_{3\mathrm{i}+2}$ and contains only suspended simplices. By Proposition 2.4.2, $\mathcal{S}^{-1}\widehat{\zeta}$ is well defined. Since $\widehat{\zeta}$ is homologous to $\widehat{\zeta}_0$ in $\widehat{K}_{3\mathrm{i}+2}$, $\widehat{\zeta}$ is also a boundary in $\widehat{K}_{3\mathrm{j}+2}$. Let $\widehat{\zeta}$ be the boundary of a $(p+1)$-chain $\widehat{A}$ in $\widehat{K}_{3\mathrm{j}+2}$. Because $\mathcal{S}K_{\mathrm{j}} = \widehat{K}_{3\mathrm{j}+2}$, by Proposition 2.4.3, $\widehat{A} \in \mathcal{S}(\mathsf{C}_p(K_{\mathrm{j}}))$. Furthermore, by Proposition 2.4.1, we have $\mathcal{S}^{-1}\widehat{\zeta} = \mathcal{S}^{-1}\partial(\widehat{A}) = \partial(\mathcal{S}^{-1}\widehat{A})$. So $\mathcal{S}^{-1}\widehat{\zeta}$ is a $(p-1)$-cycle created by σ_{i} which is a boundary in K_{j}.

From the above facts, it is immediate that $\widehat{\sigma}_{3b+2}$ is a positive simplex in $\mathcal{SF}$ and pairs with $\widehat{\sigma}_{3\delta+2}$ so that $[3b+2, 3d+2)$ is an interval in $\mathsf{Pers}_p(\mathcal{SF})$. It is also true that there is a bijection from the persistent $(p-1)$-cycles of $[b,d)$ to the persistent p-cycles of $[3b+2, 3d+2)$ containing only suspended simplices. Furthermore, the bijection preserves the weights of the cycles. From the weight assigning policy, the minimum persistent p-cycle of $[3b+2, 3d+2)$ must contain only suspended simplices, so this minimum persistent p-cycle of $[3b+2, 3d+2)$ induces a minimum persistent $(p-1)$-cycle of $[b,d)$. Now we have reduced PCYC-FIN$_{p-1}$ to PCYC-FIN$_p$. Furthermore, the reduction is in polynomial time and the size of $(\mathcal{S}K, \mathcal{SF}, [3b+2, 3d+2))$ is a polynomial function of the size of $(K, \mathcal{F}, [b,d))$. $\square$

Combining Proposition 2.4.4 and the hardness result in Section 2.4.1, we obtain the following theorem:

Theorem 2.4.2. *PCYC-FIN$_p$ is NP-hard for $p \geq 1$.*

2.5 NP-hardness for infinite intervals

In this subsection, we prove that it is NP-hard to approximate WPCYC-INF$_p$ with any fixed ratio. Let PROB be a minimization problem with solutions having positive costs. Given an instance $\mathcal{I}$ of PROB, let C^* be the cost of the minimum solution of $\mathcal{I}$. For $r \geq 1$, a solution of $\mathcal{I}$ with cost C is said to have an *approximation ratio* r if $C/C^* \leq r$ [46]. We let PROB$[r]$ denote the problem that asks for an approximate solution with ratio r given an instance of PROB. Moreover, in order to make approximation ratios well-defined for WPCYC-INF$_p$, we let WPCYC-INF$_p^+$ denote a subproblem of WPCYC-INF$_p$ where all p-simplices are positively weighted.

Before proving the hardness result, we first recall the definition of the nearest codeword problem, which is NP-hard to approximate with any fixed ratio [18]:

Problem 2.5.1 (NR-CODE)**.** *Given an $l \times k$ full-rank matrix $\mathcal{A}$ over $\mathbb{Z}_2$ for $k < l$ and a vector $y_0 \in (\mathbb{Z}_2)^l \smallsetminus \mathsf{img}(\mathcal{A})$, find a vector in $y_0 + \mathsf{img}(\mathcal{A})$ with the minimum Hamming weight.*

Remark 2.5.1. The Hamming weight of a vector y, denoted as $\|y\|_H$, is the number of non-zero components in y.

Theorem 2.5.1. *WPCYC-INF$_2^+$ is NP-hard to approximate with any fixed ratio.*

Similar to the NP-hardness proof of homology localization in [18], our proof of Theorem 2.5.1 conducts the reduction from the NR-CODE problem. One may think that a direct reduction from homology localization may be more straightforward. However, such a reduction is not immediately evident. The two problems appear to be of different nature: While the homology localization problem asks for a minimum cycle in a given homology class, WPCYC-INF$_2^+$ asks for a minimum cycle in a complex containing a given simplex without referring to any particular homology class.

Proof. For any $r > 1$, we reduce the NP-hard problem NR-CODE$[2r]$ to WPCYC-INF$_2^+[r]$. Given an instance $(\mathcal{A}, y_0)$ of NR-CODE$[2r]$, we first compute the $(l-k)\times l$ parity check matrix $\mathcal{A}^\perp$ [18], which is a matrix such that $\mathsf{ker}(\mathcal{A}^\perp) = \mathsf{img}(\mathcal{A})$. Similar to the proof of Lemma 4.3.1 in [18], we then build a "tube complex" T_1 with $(l-k)$ 1-cells each of which is a 1-sphere and l 2-cells each of which is a 2-sphere with holes. The 2-cells of T_1 are attached to the 1-cells along the holes such that the boundary matrix ∂_2 of this tube complex equals $\mathcal{A}^\perp$. The "q-chains" and "q-cycles" for a tube complex are analogously defined as for a simplicial complex. We also assign a weight of 1 to each 2-cell of T_1. By this construction, there is a straightforward bijection $\phi : (\mathbb{Z}_2)^l \to \mathsf{C}_2(T_1)$, such that the Hamming weight of a vector equals the weight of the corresponding 2-chain. Note that $\mathsf{Z}_2(T_1) = \mathsf{ker}(\partial_2) = \phi(\mathsf{ker}(\mathcal{A}^\perp)) = \phi(\mathsf{img}(\mathcal{A}))$. Let $\tilde{y}_0 = \phi(y_0)$, we then add a 2-cell $\widehat{t}$ whose boundary equals $\partial_2(\tilde{y}_0)$ to T_1 and get a new tube complex T_2. We call the 2-cycles in T_2 which are not in T_1 as the new 2-cycles in T_2. Then $\widehat{t} + \tilde{y}_0$ is a new 2-cycle in T_2 and the set of new 2-cycles in T_2 is $\widehat{t} + \tilde{y}_0 + \mathsf{Z}_2(T_1)$. We let the weight of $\widehat{t}$ also be 1. Note that there is a bijection $\psi : y_0 + \mathsf{img}(\mathcal{A}) \to \widehat{t} + \tilde{y}_0 + \mathsf{Z}_2(T_1)$, where $\psi(y_0 + z) = \widehat{t} + \tilde{y}_0 + \phi(z)$ for any $z \in \mathsf{img}(\mathcal{A})$, such that $w(\psi(y_0 + z)) = \|y_0 + z\|_H + w(\widehat{t})$.

We then construct an instance of WPCYC-INF$_2^+[r]$ by first triangulating T_2 to get a simplicial complex K. We make K 2-weighted such that the sum of the weights of all triangles in any 2-cell of T_2 equals the weight of the 2-cell. It is not hard to make the size of K a polynomial function of the number of cells of T_2. Let σ be a 2-simplex in the triangulation of the 2-cell $\widehat{t}$. We build a filtration $\mathcal{F}$ of K with σ being the last simplex added. Let the index of σ in $\mathcal{F}$ be b. Then, $[b,+\infty)$ is an infinite interval of $\mathsf{Pers}_2(\mathcal{F})$. Note that there is a bijection between the new 2-cycles in T_2 and the persistent 2-cycles of $[b,+\infty)$, where the weights of the cycles are preserved. Therefore, from the solution of WPCYC-INF$_2^+[r]$ with the input $(K,\mathcal{F},[b,+\infty))$, we can derive a new 2-cycle $\widehat{t}+\tilde{y}_0+\zeta$ of T_2, where $\zeta\in\mathsf{Z}_2(T_1)$ and $\widehat{t}+\tilde{y}_0+\zeta$ is an r-approximation of the minimum new 2-cycle. Let $\widehat{t}+\tilde{y}_0+\zeta^*$ be a minimum new 2-cycle of T_2, we have

$$\frac{w(\widehat{t}+\tilde{y}_0+\zeta)}{w(\widehat{t}+\tilde{y}_0+\zeta^*)}\le r \implies \frac{w(\widehat{t})+w(\tilde{y}_0+\zeta)}{w(\widehat{t})+w(\tilde{y}_0+\zeta^*)}\le r \implies w(\tilde{y}_0+\zeta)\le r-1+rw(\tilde{y}_0+\zeta^*)$$

We also have

$$1\le\frac{r}{r-1}w(\tilde{y}_0+\zeta^*) \implies r-1\le rw(\tilde{y}_0+\zeta^*)$$

Therefore

$$w(\tilde{y}_0+\zeta)\le 2rw(\tilde{y}_0+\zeta^*) \implies \|y_0+\phi^{-1}(\zeta)\|_H\le 2r\|y_0+\phi^{-1}(\zeta^*)\|_H$$

Since $y_0+\phi^{-1}(\zeta^*)$ is a minimum solution of $(\mathcal{A},y_0)$, then $y_0+\phi^{-1}(\zeta)$ is a $2r$-approximation of the minimum solution of $(\mathcal{A},y_0)$. Hence, we have reduced NR-CODE$[2r]$ to WPCYC-INF$_2^+[r]$. Furthermore, the reduction is in polynomial time and the sizes of the instances are related by a polynomial function, so WPCYC-INF$_2^+[r]$ is NP-hard. □

Theorem 2.5.2. *WPCYC-INF$_p^+$ is NP-hard to approximate with any fixed ratio for $p\ge 2$.*

Proof. For any $p\ge 3$ and $r\ge 1$, we reduce WPCYC-INF$_{p-1}^+[r]$ to WPCYC-INF$_p^+[r]$. Given an instance $(K,\mathcal{F},[b,+\infty))$ of WPCYC-INF$_{p-1}^+[r]$, where the i_{th} complex of $\mathcal{F}$ is denoted as K_i, let $K'=\mathcal{S}K_b^{p-1}$ where K_b^{p-1} is the $(p-1)$-skeleton of K_b. We make K' p-weighted such that any p-simplex $\sigma\cup\{\omega_\mathrm{i}\}$ of K' has half of the weight of σ in K. The complex K' is endowed with a filtration $\mathcal{F}'$ such that $\sigma_b^{\mathcal{F}}\cup\{\omega_2\}$ is the last simplex added to $\mathcal{F}'$. Let b' be the index of $\sigma_b^{\mathcal{F}}\cup\{\omega_2\}$ in $\mathcal{F}'$, then $[b',+\infty)\in\mathsf{Pers}_p(\mathcal{F}')$. It is true that $\mathcal{S}$ restricts to a bijection from

$\mathsf{Z}_{p-1}(K_b)$ to $\mathsf{Z}_p(K')$ preserving the weights of the cycles. Furthermore, for any $\zeta \in \mathsf{Z}_{p-1}(K_b)$, ζ is a persistent $(p-1)$-cycle of $[b, +\infty) \in \mathsf{Pers}_{p-1}(\mathcal{F})$ if and only if $\mathcal{S}\zeta$ is a persistent p-cycle of $[b', +\infty) \in \mathsf{Pers}_p(\mathcal{F}')$. Suppose that ζ' is a solution for the instance $(K', \mathcal{F}', [b', +\infty))$ of WPCYC-INF$_p^+[r]$, i.e., ζ' is an r-approximation of the minimum solution. Then, $\mathcal{S}^{-1}\zeta'$ is an r-approximation for the instance $(K, \mathcal{F}, [b, +\infty))$ of WPCYC-INF$_{p-1}^+[r]$. Therefore, the reduction is done. $\square$

2.6 Minimum persistent cycles for finite intervals given weak pseudomanifolds

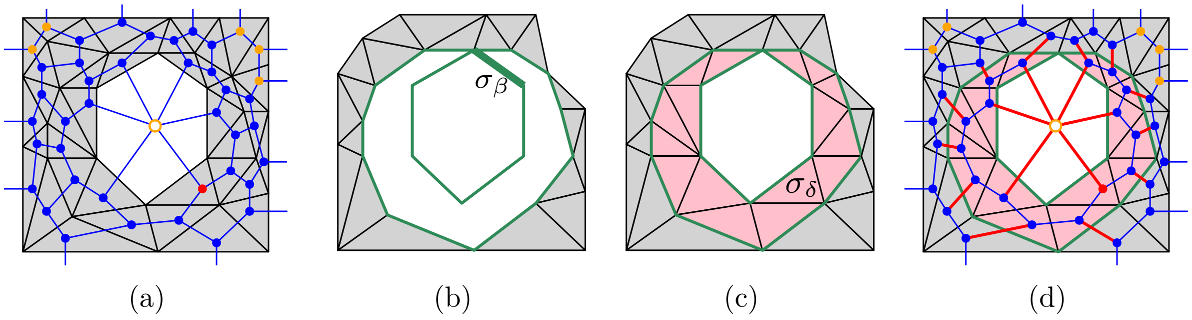


Figure 2.2. An example of the constructions in our algorithm showing the duality between persistent cycles and cuts having finite capacity for $p = 1$. (a) The input weak 2-pseudomanifold K with its dual flow network drawn in blue, where the central hollow vertex denotes the dummy vertex, the red vertex denotes the source, and all the orange vertices (including the dummy one) denote the sinks. All "dangled" graph edges dual to the outer boundary 1-simplices actually connect to the dummy vertex and these connections are not drawn. (b) The partial complex K_b in the input filtration $\mathcal{F}$, where the bold green 1-simplex denotes $\sigma_b^{\mathcal{F}}$ which creates the green 1-cycle. (c) The partial complex K_d in $\mathcal{F}$, where the 2-simplex $\sigma_d^{\mathcal{F}}$ creates the pink 2-chain killing the green 1-cycle. (d) The green persistent 1-cycle of the interval $[b, d)$ is dual to a cut (S, T) having finite capacity, where S contains all the vertices inside the pink 2-chain and T contains all the other vertices. The red graph edges denote those edges across (S, T) and their dual 1-chain is the green persistent 1-cycle.

In this section, we present an algorithm which computes minimum persistent p-cycles for finite intervals given a filtration of a weak $(p+1)$-pseudomanifold when $p \geq 1$. The general process is as follows: Suppose that the input weak $(p+1)$-pseudomanifold is K associated with a filtration $\mathcal{F} : K_0 \subseteq K_1 \subseteq \ldots \subseteq K_n$ and the task is to compute the minimum persistent cycle of a finite interval $[b, d) \in \mathsf{Pers}_p(\mathcal{F})$. We first construct an undirected dual graph G

for K where vertices of G are dual to $(p+1)$-simplices of K and edges of G are dual to p-simplices of K. One dummy vertex termed as *infinite vertex* which does not correspond to any $(p+1)$-simplices is added to G for graph edges dual to those boundary p-simplices. We then build an undirected flow network on top of G where the source is the vertex dual to $\sigma_d^{\mathcal{F}}$ and the sink is the infinite vertex along with the set of vertices dual to those $(p+1)$-simplices which are added to $\mathcal{F}$ after $\sigma_d^{\mathcal{F}}$. If a p-simplex is $\sigma_b^{\mathcal{F}}$ or added to $\mathcal{F}$ before $\sigma_b^{\mathcal{F}}$, we let the capacity of its dual graph edge be its weight; otherwise, we let the capacity of its dual graph edge be $+\infty$. Finally, we calculate a minimum cut of this flow network and return the p-chain dual to the edges across the minimum cut as a minimum persistent cycle of the interval.

The intuition of the above algorithm is best explained by an example in Figure 2.2, where $p=1$. The key to the algorithm is the duality between persistent cycles of the input interval and cuts of the dual flow network having finite capacity. To see this duality, first consider a persistent p-cycle ζ of the input interval $[b,d)$. There exists a $(p+1)$-chain A in K_d created by $\sigma_d^{\mathcal{F}}$ whose boundary equals ζ, making ζ killed. We can let S be the set of graph vertices dual to the simplices in A and let T be the set of the remaining graph vertices, then (S,T) is a cut. Furthermore, (S,T) must have finite capacity as the edges across it are exactly dual to the p-simplices in ζ and the p-simplices in ζ have indices in $\mathcal{F}$ less than or equal to b. On the other hand, let (S,T) be a cut with finite capacity, then the $(p+1)$-chain whose simplices are dual to the vertices in S is created by $\sigma_d^{\mathcal{F}}$. Taking the boundary of this $(p+1)$-chain, we get a p-cycle ζ. Because p-simplices of ζ are exactly dual to the edges across (S,T) and each edge across (S,T) has finite capacity, ζ must reside in K_b. We only need to ensure that ζ contains $\sigma_b^{\mathcal{F}}$ in order to show that ζ is a persistent cycle of $[b,d)$. In Section 2.6.2, we argue that ζ actually contains $\sigma_b^{\mathcal{F}}$, so ζ is indeed a persistent cycle. Note that while the above explanation introduces the general idea, the rigorous statement and proof of the duality are articulated by Proposition 2.6.2 and 2.6.3.

We list the pseudocode in Algorithm 2.6.1 and it works as follows: Line 2 and 3 set up a complex $\tilde{K}$ that the algorithm mainly works on, where $\tilde{K}$ is taken as the closure of the $(p+1)$-connected component of K containing $\sigma_d^{\mathcal{F}}$. The reason for working on $\tilde{K}$ instead of the entire complex is explained later in this section. Line 4 constructs the dual graph G from

Algorithm 2.6.1 Computing minimum persistent p-cycles for finite intervals for weak $(p+1)$-pseudomanifolds

Input:
K: finite p-weighted weak $(p+1)$-pseudomanifold
p: integer ≥ 1
$\mathcal{F}$: filtration $K_0 \subseteq K_1 \subseteq \ldots \subseteq K_n$ of K
$[b,d)$: finite interval of $\mathsf{Pers}_p(\mathcal{F})$
Output:
minimum persistent p-cycle for $[b,d)$

```
 1: procedure MinPersCycFin($K, p, \mathcal{F}, [b,d)$)
        ▷ set up the complex $\tilde{K}$ being worked on
 2:     $C^{p+1} \leftarrow (p+1)$-connected component of $K$ containing $\sigma_d^{\mathcal{F}}$
 3:     $\tilde{K} \leftarrow$ closure of the simplicial set $C^{p+1}$
        ▷ construct dual graph
 4:     $(G, \theta) \leftarrow$ DualGraphFin($\tilde{K}, p$)
        ▷ assign capacity to $G$
 5:     for each $\mathrm{e} \in E(G)$ do
 6:         if $\mathrm{ind}(\theta^{-1}(\mathrm{e})) \leq b$ then
 7:             $c(\mathrm{e}) \leftarrow w(\theta^{-1}(\mathrm{e}))$
 8:         else
 9:             $c(\mathrm{e}) \leftarrow +\infty$
        ▷ set the source
10:     $s_1 \leftarrow \{\theta(\sigma_d^{\mathcal{F}})\}$
        ▷ set the sink
11:     $s_2 \leftarrow \{v \in V(G) \mid v \neq \phi,\ \mathrm{ind}(\theta^{-1}(v)) > d\}$
12:     if $\phi \in V(G)$ then
13:         $s_2 \leftarrow s_2 \cup \{\phi\}$
14:     $(S^*, T^*) \leftarrow$ min-cut of $(G, s_1, s_2)$
15:     return $\theta^{-1}(\xi(S^*, T^*))$
```

$\tilde{K}$ and line 5–13 builds the flow network on top of G. Note that we denote the infinite vertex by ϕ. Line 14 computes a minimum cut for the flow network and line 15 returns the p-chain dual to the edges across the minimum cut. In the pseudocodes of this chapter, to ease the exposition, we treat a mathematical function as a computer program object. For example, the function θ returned by DualGraphFin in Algorithm 2.6.1 denotes the bijection between the simplices of $\tilde{K}$ and their dual vertices or edges (see Section 2.6.1 for details). In practice, these constructs can be easily implemented in any computer programming language.

To see the reason why we work on $\tilde{K}$, we first note that the dual graph constructed directly from K may be disconnected[4]. While cuts are still well-defined for a disconnected flow network, one may prefer a connected one as the minimum cut computation only concerns the graph component containing the source. By constructing the dual graph from $\tilde{K}$, it can be ensured that the graph is connected. In order for Algorithm 2.6.1 to work, one has to further show that the sink is non-empty so that the computed persistent cycle is non-empty. This is verified in Proposition 2.6.1. An intuitive reason why the computation from $\tilde{K}$ is still correct is as follows: Each persistent p-cycle ζ of the given interval corresponds to a $(p+1)$-chain A which kills ζ, i.e., $\partial(A) = \zeta$. Suppose that A is not entirely contained in $\tilde{K}$. Notice that $A \cap \tilde{K} \neq \varnothing$ and contains at least the killer simplex $\sigma_d^{\mathcal{F}}$. Then $\partial(A \cap \tilde{K})$ must be a persistent cycle of the interval residing in $\tilde{K}$ which has a smaller weight. Hence, a minimum persistent cycle must reside in $\tilde{K}$. In Section 2.6.2, we formally verify the construction.

The time complexity of Algorithm 2.6.1 depends on the encoding scheme of the input and the data structure used for representing a simplicial complex. For encodings of the input, we assume K and $\mathcal{F}$ to be represented by a sequence of all the simplices of K ordered by their indices in $\mathcal{F}$, where each simplex is denoted by its set of vertices. We also assume a simple yet reasonable simplicial complex data structure as follows: In each dimension, simplices are mapped to integral identifiers ranging from 0 to the number of simplices in that dimension minus 1; each q-simplex has an array (or linked list) storing all the id's of its $(q+1)$-cofaces; a hash map for each dimension is maintained for the query of the integral id of each simplex in that dimension based on the spanning vertices of the simplex. We further assume p to be constant. By the above assumptions, let n be the size (number of bits) of the encoded input, then there are no more than n elementary $O(1)$ operations in line 2 and 3. So, the time complexity of line 2 and 3 is $O(n)$. It is not hard to verify that the flow network construction also takes $O(n)$ time so the time complexity of Algorithm 2.6.1 is determined by the minimum cut algorithm. Using the max-flow algorithm by Orlin [47], the time complexity of Algorithm 2.6.1 becomes $O(n^2)$.

[4]↑ For an example in $d = 1$, take K as two disconnected triangulated 2-spheres. Its dual graph consists of two connected components.

In the rest of this section, we first explain the bijection θ returned by DualGraphFin, then prove the correctness of the algorithm.

2.6.1 The bijection θ

The vertex set $V(G)$ of G contains vertices which correspond to the $(p+1)$-simplices of $\tilde{K}$. The set $V(G)$ may also contain an infinite vertex ϕ if $\tilde{K}$ contains any boundary p-simplex. We define a bijection

$$\theta : \{(p+1)\text{-simplices of } \tilde{K}\} \to V(G) \smallsetminus \{\phi\}$$

such that for any $(p+1)$-simplex σ^{p+1} of $\tilde{K}$, $\theta(\sigma^{p+1})$ is the vertex that σ^{p+1} is dual to. Similarly, we define another bijection

$$\theta : \{p\text{-simplices of } \tilde{K}\} \to E(G)$$

using the same notation θ.

Note that we can take the image of a subset of the domain under a function. Therefore, if (S,T) is a cut for a flow network built on G, then $\theta^{-1}(\xi(S,T))$ denotes the set of p-simplices dual to the edges across the cut. Also note that since simplicial chains with $\mathbb{Z}_2$ coefficients can be interpreted as sets, $\theta^{-1}(\xi(S,T))$ is also a p-chain.

2.6.2 Algorithm correctness

In this subsection, we prove the correctness of Algorithm 2.6.1. Some of the symbols we use refer to Algorithm 2.6.1.

Proposition 2.6.1. *In Algorithm 2.6.1, the sink s_2 is not an empty set.*

Proof. For contradiction, suppose that s_2 is an empty set. Then, $\phi \notin V(G)$ and $\sigma_d^{\mathcal{F}}$ is the $(p+1)$-simplex of $\tilde{K}$ with the greatest index in $\mathcal{F}$. Because $\phi \notin V(G)$, any p-simplex of $\tilde{K}$ must be a face of two $(p+1)$-simplices of $\tilde{K}$, so the set of $(p+1)$-simplices of $\tilde{K}$

forms a $(p+1)$-cycle created by $\sigma_d^{\mathcal{F}}$. Then $\sigma_d^{\mathcal{F}}$ must be a positive simplex in $\mathcal{F}$, which is a contradiction. $\square$

The following two propositions specify the duality mentioned at the beginning of this section:

Proposition 2.6.2. *For any cut (S,T) of (G,s_1,s_2) with finite capacity, the p-chain $\zeta = \theta^{-1}(\xi(S,T))$ is a persistent p-cycle for $[b,d)$ and $w(\zeta) = c(S,T)$.*

Proof. Let $A = \theta^{-1}(S)$, we first want to prove $\zeta = \partial(A)$, so that ζ is a cycle. Let σ^p be any p-simplex of ζ, then $\theta(\sigma^p)$ connects a vertex $u \in S$ and a vertex $v \in T$. If $v = \phi$, then σ^p cannot be a face of another $(p+1)$-simplex in K other than $\theta^{-1}(u)$. So, σ^p is a face of exactly one $(p+1)$-simplex of A. If $v \neq \phi$, then σ^p is also a face of exactly one $(p+1)$-simplex of A. Therefore, $\sigma^p \in \partial(A)$. On the other hand, let σ^p be any p-simplex of $\partial(A)$, then σ^p is a face of exactly one $(p+1)$-simplex σ_0^{p+1} of A. If σ^p is a face of another $(p+1)$-simplex σ_1^{p+1} in K, then $\sigma_1^{p+1} \in \tilde{K}$ and $\sigma_1^{p+1} \notin A$. So, $\theta(\sigma^p)$ connects the vertex $\theta(\sigma_0^{p+1}) \in S$ and the vertex $\theta(\sigma_1^{p+1}) \in T$ in the graph G. If σ^p is a face of exactly one $(p+1)$-simplex in K, $\theta(\sigma^p)$ must connect $\theta(\sigma_0^{p+1}) \in S$ and $\phi \in T$ in G. So we have $\theta(\sigma^p) \in \xi(S,T)$, i.e., $\sigma^p \in \theta^{-1}(\xi(S,T))$.

We then show that ζ is created by $\sigma_b^{\mathcal{F}}$. By Proposition 2.6.1, ζ cannot be empty. Therefore, for contradiction, we can suppose that ζ is created by a p-simplex $\sigma^p \neq \sigma_b^{\mathcal{F}}$. Because $c(S,T)$ has finite capacity, we have that $\mathrm{ind}(\sigma^p) < b$. We can let ζ' be a persistent cycle of $[b,d)$ and $\zeta' = \partial(A')$ where A' is a $(p+1)$-chain of K_d. Then we have $\zeta + \zeta' = \partial(A + A')$. Since A and A' are both created by $\sigma_d^{\mathcal{F}}$, then $A + A'$ is created by a $(p+1)$-simplex with an index less than d in $\mathcal{F}$. So $\zeta + \zeta'$ is a p-cycle created by $\sigma_b^{\mathcal{F}}$ which becomes a boundary before $\sigma_d^{\mathcal{F}}$ is added. This means that $\sigma_b^{\mathcal{F}}$ is already paired when $\sigma_d^{\mathcal{F}}$ is added, contradicting the fact that $\sigma_b^{\mathcal{F}}$ is paired with $\sigma_d^{\mathcal{F}}$. Similarly, we can prove that ζ is not a boundary until $\sigma_d^{\mathcal{F}}$ is added, so ζ is a persistent cycle of $[b,d)$. Since (S,T) has finite capacity, we must have

$$c(S,T) = \sum_{\mathrm{e} \in \theta(\zeta)} c(\mathrm{e}) = \sum_{\theta^{-1}(\mathrm{e}) \in \zeta} w(\theta^{-1}(\mathrm{e})) = w(\zeta).$$

$\square$

Proposition 2.6.3. *For any persistent p-cycle ζ for $[b,d)$, there exists a cut (S,T) of (G,s_1,s_2) such that $c(S,T) \leq w(\zeta)$.*

Proof. Let A be a $(p+1)$-chain in K_d such that $\zeta = \partial(A)$. Note that A is created by $\sigma_d^{\mathcal{F}}$ and ζ is the set of p-simplices which are face of exactly one $(p+1)$-simplex of A. Let $\zeta' = \zeta \cap \tilde{K}$ and $A' = A \cap \tilde{K}$, we claim that $\zeta' = \partial(A')$. To prove this, first let σ^p be any p-simplex of ζ', then σ^p is a face of exactly one $(p+1)$-simplex σ^{p+1} of A. Since $\sigma^p \in \tilde{K}$, it is also true that $\sigma^{p+1} \in \tilde{K}$, so $\sigma^{p+1} \in A'$. Then σ^p is a face of exactly one $(p+1)$-simplex of A', so $\sigma^p \in \partial(A')$. On the other hand, let σ^p be any p-simplex of $\partial(A')$, then σ^p is a face of exactly one $(p+1)$-simplex σ_0^{p+1} of A'. Note that $\sigma_0^{p+1} \in A$ and we then want to prove that σ^p is a face of exactly one $(p+1)$-simplex σ_0^{p+1} of A. Suppose that σ^p is a face of another $(p+1)$-simplex σ_1^{p+1} of A, then $\sigma_1^{p+1} \in \tilde{K}$ because $\sigma_0^{p+1} \in \tilde{K}$. So we have $\sigma_1^{p+1} \in A \cap \tilde{K} = A'$, contradicting the fact that σ^p is a face of exactly one $(p+1)$-simplex of A'. Then we have $\sigma^p \in \partial(A)$. Since $\sigma_0^{p+1} \in \tilde{K}$, we have $\sigma^p \in \tilde{K}$, which means that $\sigma^p \in \zeta'$.

Let $S = \theta(A')$ and $T = V(G) \smallsetminus S$, then it is true that (S, T) is a cut of (G, s_1, s_2) because A' is created by $\sigma_d^{\mathcal{F}}$. We claim that $\theta^{-1}(\xi(S,T)) = \partial(A')$. The proof of the equality is similar to the one in the proof of Proposition 2.6.2. It follows that $\xi(S,T) = \theta(\zeta')$. We then have that

$$c(S,T) = \sum_{\mathrm{e} \in \theta(\zeta')} c(\mathrm{e}) = \sum_{\theta^{-1}(\mathrm{e}) \in \zeta'} w(\theta^{-1}(\mathrm{e})) = w(\zeta')$$

because each p-simplex of ζ' has an index less than or equal to b in $\mathcal{F}$.

Finally, because ζ' is a subchain of ζ, we must have $c(S,T) = w(\zeta') \leq w(\zeta)$. □

Combining the above facts, we can conclude:

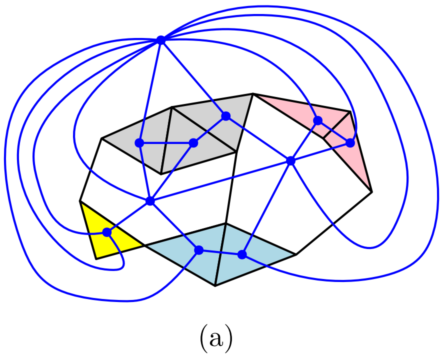
Theorem 2.6.1. *Algorithm 2.6.1 computes a minimum persistent p-cycle for the given interval $[b, d)$.*

Proof. First, the flow network (G, s_1, s_2) constructed by Algorithm 2.6.1 must be valid by Proposition 2.6.1. Next, because the interval $[b, d)$ must have a persistent cycle, by Proposition 2.6.3, the flow network (G, s_1, s_2) has a cut with finite capacity. This means that $c(S^*, T^*)$ is finite. By Proposition 2.6.2, the chain $\zeta^* = \theta^{-1}(\xi(S^*, T^*))$ is a persistent cycle of $[b, d)$. Assume that ζ^* is not a minimum persistent cycle of $[b, d)$ and instead let ζ' be a minimum persistent cycle of $[b, d)$. Then there exists a cut (S', T') such that

$c(S', T') \leq w(\zeta') < w(\zeta^*) = c(S^*, T^*)$ by Proposition 2.6.2 and 2.6.3, contradicting the fact that (S^*, T^*) is a minimum cut. □

2.7 Minimum persistent cycles for infinite intervals given weak pseudomanifolds

In Section 2.5, we proved that computing minimum persistent p-cycles ($p \geq 2$) for infinite intervals is NP-hard even if we restrict to weak $(p+1)$-pseudomanifolds. However, when the complex is embedded in $\mathbb{R}^{p+1}$, the problem becomes polynomially tractable. In this section, we present an algorithm for this problem in $p \geq 1$[5]. The algorithm uses a similar duality described in Section 2.6. However, a direct use of the approach in Section 2.6 does not work. For example, in Figure 2.3a, 1-simplices that do not have any 2-cofaces cannot reside in any 2-connected component of the given complex. Hence, no cut in the flow network may correspond to a persistent cycle of the infinite interval created by such a 1-simplex. Furthermore, unlike the finite interval case, we do not have a negative simplex whose dual can act as a source in the flow network.



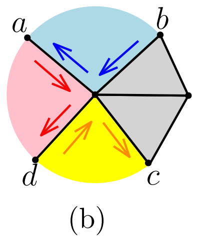


Figure 2.3. (a) A weak 2-pseudomanifold $\tilde{K}$ embedded in $\mathbb{R}^2$ with three voids. Its dual graph is drawn in blue. The complex has one 1-connected component and four 2-connected components with the 2-simplices in different 2-connected components colored differently. (b) An example illustrating the pairing of boundary p-simplices in the neighborhood of a $(p-1)$-simplex for $p = 1$. The four boundary 1-simplices produce six oriented boundary 1-simplices and the paired oriented 1-simplices are colored the same.

Let $(K, \mathcal{F}, [b, +\infty))$ be an input to the problem where K is a weak $(p+1)$-pseudomanifold embedded in $\mathbb{R}^{p+1}$, $\mathcal{F} : K_0 \subseteq K_1 \subseteq \ldots \subseteq K_n$ is a filtration of K, and $[b, +\infty)$ is an infinite

[5]↑As mentioned earlier, when $p = 1$, this problem is polynomially tractable for arbitrary complexes.

interval of $\mathsf{Pers}_p(\mathcal{F})$. By the definition of the problem, the task boils down to computing a minimum p-cycle containing $\sigma_b^{\mathcal{F}}$ in K_b. Note that K_b is also a weak $(p+1)$-pseudomanifold embedded in $\mathbb{R}^{p+1}$.

Generically, assume $\tilde{K}$ is an arbitrary weak $(p+1)$-pseudomanifold embedded in $\mathbb{R}^{p+1}$ and we want to compute a minimum p-cycle containing a p-simplex $\tilde{\sigma}$ for $\tilde{K}$. By the embedding assumption, the connected components of $\mathbb{R}^{p+1} \smallsetminus |\tilde{K}|$ are well defined and we call them the *voids* of $\mathbb{R}^{p+1} \smallsetminus |\tilde{K}|$. The complex $\tilde{K}$ has a natural (undirected) dual graph structure as exemplified by Figure 2.3a for $p = 1$, where the graph vertices are dual to the $(p+1)$-simplices as well as the voids and the graph edges are dual to the p-simplices. The duality between cycles and cuts is as follows: Since the ambient space $\mathbb{R}^{p+1}$ is contractible (homotopy equivalent to a point), every p-cycle in $\tilde{K}$ is the boundary of a $(p+1)$-dimensional region obtained by point-wise union of certain $(p+1)$-simplices and/or voids. We can derive a cut[6] of the dual graph by putting all vertices contained in the $(p+1)$-dimensional region into one vertex set and putting the rest into the other vertex set. On the other hand, for every cut of the graph, we can take the point-wise union of all the $(p+1)$-simplices and voids dual to the graph vertices in one set of the cut and derive a $(p+1)$-dimensional region. The boundary of the derived $(p+1)$-dimensional region is then a p-cycle in $\tilde{K}$. We observe that by making the source and sink dual to the two $(p+1)$-simplices or voids that $\tilde{\sigma}$ adjoins, we can build a flow network where a minimum cut produces a minimum p-cycle in $\tilde{K}$ containing $\tilde{\sigma}$.

The efficiency of the above algorithm is in part determined by the efficiency of the dual graph construction. This step requires identifying the voids that the boundary p-simplices are incident on. A straightforward approach would be to first group the boundary p-simplices into p-cycles by local geometry, and then build the nesting structure of these p-cycles to correctly reconstruct the boundaries of the voids. This approach has a quadratic worst-case complexity. To make the void boundary reconstruction faster, we assume that the simplicial complex being worked on is p-connected so that building the nesting structure is not needed. Our reconstruction then runs in almost linear time. To satisfy the p-connected assumption, we begin our algorithm by taking $\tilde{K}$ as a p-connected subcomplex of K_b containing $\sigma_b^{\mathcal{F}}$ and

[6]↑ The cut here is defined on a graph without sources and sinks, so the cut is simply a partition of the vertex set into two sets.

continue only with this $\tilde{K}$. The computed output is still correct because the minimum cycle in $\tilde{K}$ is again a minimum cycle in K_b as shown in Section 2.7.2.

Algorithm 2.7.1 Computing minimum persistent p-cycles for infinite intervals for weak $(p+1)$-pseudomanifolds embedded in $\mathbb{R}^{p+1}$

Input:
K: finite p-weighted weak $(p+1)$-pseudomanifold embedded in $\mathbb{R}^{p+1}$
p: integer ≥ 1
$\mathcal{F}$: filtration $K_0 \subseteq K_1 \subseteq \ldots \subseteq K_n$ of K
$[b, +\infty)$: infinite interval of $\mathsf{Pers}_p(\mathcal{F})$
Output:
minimum persistent p-cycle for $[b, +\infty)$

```
1:  procedure MinPersCycInf($K, p, \mathcal{F}, [b, +\infty)$)
        ▷ set up the complex $\tilde{K}$ being worked on
2:      $K'_b \leftarrow$ Prune($K_b, p$)
3:      $C_b \leftarrow$ $p$-connected component of $K'_b$ containing $\sigma_b^{\mathcal{F}}$
4:      $\Sigma^{p+1} \leftarrow \{\sigma \in K'_b \mid \sigma$ is a $(p+1)$-simplex and all $p$-faces of $\sigma$ are in $C_b\}$
5:      $\tilde{K} \leftarrow$ (closure of the simplicial set $C_b$) $\cup\, \Sigma^{p+1}$
        ▷ construct dual graph
6:      $(\vec{\zeta}_1, \ldots, \vec{\zeta}_k) \leftarrow$ VoidBoundary($\tilde{K}, p$)
7:      $(G, \theta) \leftarrow$ DualGraphInf($\tilde{K}, p, \vec{\zeta}_1, \ldots, \vec{\zeta}_k$)
        ▷ assign capacity to $G$
8:      for each $\mathrm{e} \in E(G)$ do
9:          $c(\mathrm{e}) \leftarrow w(\theta^{-1}(\mathrm{e}))$
10:     $(v_1, v_2) \leftarrow$ end vertices of edge $\theta(\sigma_b^{\mathcal{F}})$ in $G$
        ▷ set the source
11:     $s_1 \leftarrow \{v_1\}$
        ▷ set the sink
12:     $s_2 \leftarrow \{v_2\}$
13:     $(S^*, T^*) \leftarrow$ min-cut of $(G, s_1, s_2)$
14:     return $\theta^{-1}(\xi(S^*, T^*))$
```

We list the pseudocode in Algorithm 2.7.1 and it works as follows: Line 2–5 set up the complex $\tilde{K}$ that the algorithm works on. Line 2 prunes K_b to produce a complex K'_b. Given (K_b, p), the Prune subroutine iteratively deletes a p-simplex σ^p of K_b such that there is a $(p-1)$-face of σ^p having σ^p as the only p-coface (i.e., σ^p is a dangled p-simplex), until no such p-simplex can be found. It is not hard to verify that Prune only deletes p-simplices not residing in any p-cycles, so a minimum d-cycle containing $\sigma_b^{\mathcal{F}}$ is never deleted. We perform the pruning because it can reduce the graph size for the minimum cut computation which is more time consuming. In line 3−5, we take the p-connected component C_b of K'_b containing $\sigma_b^{\mathcal{F}}$ and add a set Σ^{p+1} of $(p+1)$-simplices to the closure of C_b to form $\tilde{K}$. The set Σ^{p+1} contains all $(p+1)$-simplices of K'_b whose p-faces reside in C_b. The reason of adding the set Σ^{p+1} is to reduce the number of voids for the complex $\tilde{K}$ and in turn reduce the running time of the subsequent void boundary reconstruction. For example, in Figure 2.4b, we could treat the entire complex as K'_b, all 1-simplices as C_b, and all 2-simplices as Σ^{p+1}. If we do not add Σ^{p+1} to the closure of C_b, there will be seven more voids corresponding to the seven 2-simplices. Line 6 reconstructs the void boundaries for $\tilde{K}$. Each returned $\vec{\zeta_j}$ denotes a set of p-simplices forming the boundary of a void. As indicated in Section 2.7.1, the p-simplices in a void boundary are oriented. Line 7 constructs the dual graph G based on the reconstructed void boundaries. Similar to Algorithm 2.6.1, the function θ returned by DualGraphInf denotes the bijection from p-simplices of $\tilde{K}$ to $E(G)$. Line 8−12 build the flow network on top of G. The capacity of each edge is equal to the weight of its dual p-simplex and the source and sink are selected as previously described. Line 13 computes a minimum cut for the flow network and line 14 returns the p-chain dual to the edges across the minimum cut.

We make the same assumptions as in the complexity analysis for Algorithm 2.6.1. Since the void boundary reconstruction needs to sort the p-cofaces of certain $(p-1)$-simplices, its worst-case time complexity is $O(n \log n)$. Then, all operations other than the minimum cut computation take $O(n \log n)$ time. Therefore, similar to Algorithm 2.6.1, Algorithm 2.7.1 achieves a complexity of $O(n^2)$ by using Orlin's max-flow algorithm [47].

In the rest of this section, we first describe the subroutine VoidBoundary invoked by Algorithm 2.7.1 and then prove the correctness of the algorithm.

2.7.1 Void boundary reconstruction

As previously stated, the object of the reconstruction is to identify which voids a boundary p-simplex of $\tilde{K}$ is incident on. The task becomes complicated because a void may have disconnected boundaries and a p-simplex may bound more than one void. This is exemplified in Figure 2.4a. To address this issue, we orient the boundary p-simplices and determine the

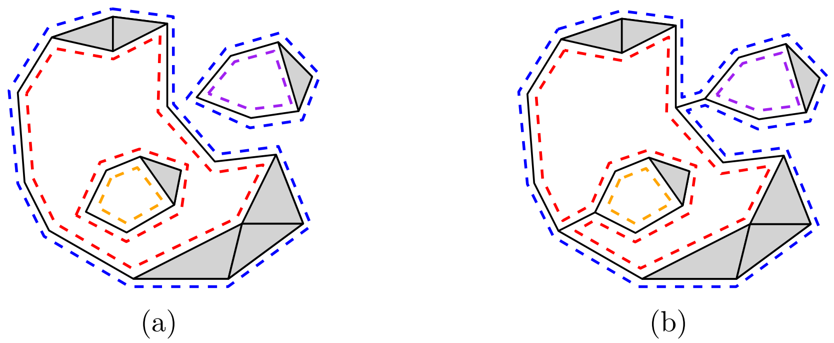


Figure 2.4. Examples showing how the void boundaries are reconstructed for $p = 1$. (a) Oriented boundary 1-simplices (drawn as dashed edges) of a simplicial complex are grouped into six 1-cycles and these six 1-cycles are further grouped into four void boundaries with each void boundary identically colored. (b) With the complex being 1-connected, the four grouped 1-cycles are exactly the boundaries of the four voids.

orientations consistently from the voids they bound. This is possible because an orientation of a p-simplex in $\mathbb{R}^{p+1}$ associates exactly one of its two sides to the p-simplex. To reconstruct the boundaries, we first inspect the neighborhood of each $(p-1)$-simplex being a face of a boundary p-simplex and pair the oriented boundary p-simplices in the neighborhood which locally bound the same void. Figure 2.3b gives an example of the oriented boundary p-simplices pairing for $p = 1$. In Figure 2.3b, there are three local voids each colored differently. The oriented 1-simplices with the same color bound the same void and are paired.

After pairing the oriented boundary p-simplices, we group them by putting paired ones into the same group. Each group then forms a p-cycle (with $\mathbb{Z}$ coefficients). This is exemplified by Figure 2.4 for $p = 1$. Note that in general, the above grouping does not fully reconstruct the void boundaries. This can be seen from Figure 2.4a where the complex has four voids but the grouping produces six 1-cycles. In order to fully reconstruct the boundaries, one has to retrieve the nesting structure of these p-cycles, which may take $\Omega(n^2)$ time in the worst-case. However, as we work on a complex $\tilde{K}$ that is p-connected, we cannot have voids with disconnected boundaries. Therefore, the grouping of oriented p-simplices can fully recover the void boundaries. Figure 2.4b gives an example for this when $p = 1$, where we add two 1-simplices to make the complex 1-connected. The four 1-cycles produced by the grouping are exactly the boundaries of the four voids.

In the rest of this subsection, we formalize the above ideas for reconstructing void boundaries and provide a proof for the correctness. Throughout this subsection, $\tilde{K}$ and p are as defined in Algorithm 2.7.1. We first introduce the definition of the natural orientation of a q-simplex in $\mathbb{R}^q$. We use its induced orientation to canonically orient the boundary simplices.

Definition 2.7.1 (Natural orientation [48])**.** *Let $q > 1$ and $\sigma = \{v_0, \ldots, v_q\}$ be a q-simplex in $\mathbb{R}^q$, an oriented simplex $\vec{\sigma} = [v'_0, \ldots, v'_q]$ of σ is* naturally oriented *if* $\det(v'_1 - v'_0, \ldots, v'_q - v'_0) > 0$*. For each face σ' of σ, the natural orientation of σ induces an orientation of σ' which we term as the* induced orientation*.*

We now formally define the boundary of a void as follows:

Definition 2.7.2 (Boundary of void)**.** *Let K be a simplicial complex embedded in $\mathbb{R}^q$ where $q \geq 2$, an oriented $(q-1)$-simplex $\vec{\sigma}^{q-1} = [v_0, \ldots, v_{q-1}]$ of K is said to* bound *a void $\mathcal{V}$ of $\mathbb{R}^q \smallsetminus |K|$ if the following conditions are satisfied:*

- *The simplex $\sigma^{q-1} = \{v_0, \ldots, v_{q-1}\}$ is contained in the closure of $\mathcal{V}$.*
- *Let u be an interior point of $\sigma^{q-1} = \{v_0, \ldots, v_{q-1}\}$, v be a point in $\mathcal{V}$ such that the line segment $\overline{uv}$ is contained in $\mathcal{V}$ and $\overline{uv}$ is orthogonal to the hyperplane spanned by σ^{q-1}. Furthermore, let $\vec{\sigma}^q$ be the naturally oriented simplex of $\{v, v_0, \ldots, v_{q-1}\}$. Then, $\vec{\sigma}^{q-1}$ has the induced orientation from $\vec{\sigma}^q$.*

The boundary *of a void $\mathcal{V}$ is then defined as the set of oriented $(q-1)$-simplices of K bounding $\mathcal{V}$.*

Remark 2.7.1. We can also interpret the boundary of a void as a sum of oriented $(q-1)$-simplices, then the boundary defines a $(q-1)$-cycle (with $\mathbb{Z}$ coefficients).

We now describe the pairing algorithm of the oriented boundary p-simplices for $\tilde{K}$. From now on, we denote the set of boundary p-simplices of $\tilde{K}$ as $\mathsf{bd}(\tilde{K})$. Let σ^{p-1} be a $(p-1)$-simplex which is a face of a p-simplex in $\mathsf{bd}(\tilde{K})$, we first take a 2D plane Δ which contains an interior point of σ^{p-1} and is orthogonal to the hyperplane spanned by σ^{p-1}. We then take the intersection of the plane Δ with each boundary p-simplex in the neighborhood of σ^{p-1} to get a set of line segments that we order circularly starting from an arbitrary one. For each two consecutive line segments in this order which enclose a void, we pick a point p on the plane Δ which resides in the void. Suppose that one of the two line segments is derived from a boundary p-simplex $\sigma_0^p = \{v_0, \ldots, v_p\}$. We take the $(p+1)$-simplex $\sigma^{p+1} = \{p, v_0, \ldots, v_p\}$ and the induced oriented simplex $\vec{\sigma}_0^p$ of σ_0^p derived from the naturally oriented simplex of σ^{p+1}. For the other line segment, we similarly derive an induced oriented simplex $\vec{\sigma}_1^p$ and pair the two oriented p-simplices $\vec{\sigma}_0^p$ and $\vec{\sigma}_1^p$. Figure 2.3b can be reused to exemplify the pairing. The union of the shaded regions in the figure is the plane Δ and a, b, c, and d are the line segments derived from intersecting the plane with four boundary p-simplices. Taking the circular order a, b, c, d, we see that the consecutive ones which enclose a void are (a, b), (c, d), and (d, a). For (a, b), we can pick p as an interior point in the blue region and the two oriented p-simplices corresponding to a and b can be induced and paired.

In summary, the steps of the VoidBoundary subroutine are the following:

1. For each $(p-1)$-simplex σ^{p-1} being a face of a p-simplex in $\mathsf{bd}(\tilde{K})$, pair all oriented boundary p-simplices in the neighborhood.

2. After gathering all the pairing, group the oriented boundary p-simplices by putting all paired ones into a group.

3. Return $(\vec{\zeta}_1, \ldots, \vec{\zeta}_k)$, each of which is a group of the oriented boundary p-simplices.

The following theorem concludes the correctness of the reconstruction:

Theorem 2.7.1. *Any $\vec{\zeta}_j$ returned by* VoidBoundary *is the boundary of a void of* $\mathbb{R}^{p+1} \setminus |\tilde{K}|$.

We first define some symbols used for proving Theorem 2.7.1. The interior of a set U is denoted by $\mathsf{Int}(U)$. The boundary of a topological ball $\mathbb{B}$ is denoted by $\mathsf{bd}(\mathbb{B})$. The set of q-cofaces of a simplex σ in a Δ-complex [2] K is denoted by $\mathsf{cof}_q^K(\sigma)$.

The proof of Theorem 2.7.1 is based on the extended Jordan–Brouwer separation theorem (Theorem 2.7.2) by Alexander [49]. The statement of the theorem depends on the following definition:

Definition 2.7.3 (Pseudomanifold)**.** *A simplicial complex K is a q-*pseudomanifold *if K is a pure q-complex and each $(q-1)$-simplex is a face of exactly two q-simplices in K.*

Remark 2.7.2. Note that definitions for q-pseudomanifolds, such as in [50], typically assume the complex to be q-connected.

Theorem 2.7.2. *Let $q > 1$ and $\mathcal{M}$ be a finite $(q-1)$-connected $(q-1)$-pseudomanifold embedded in $\mathbb{R}^q$, then $\mathbb{R}^q \smallsetminus |\mathcal{M}|$ has exactly 2 connected components.*

Now we can finish our proof:

Proof of Theorem 2.7.1. The general idea of the proof is as follows: Using a trick which we call the "de-contracting", we first create a Δ-complex $\tilde{K}'$ where each oriented simplex of $\vec{\zeta_{\mathrm{j}}}$ uniquely corresponds to an unoriented simplex. Then, using a trick which we call the "de-pinching", we show that $\vec{\zeta_{\mathrm{j}}}$ is the boundary of a region $\mathcal{A}$. Finally, from the above fact, we use proof by contradiction to reach the conclusion. Figure 2.5b gives an example of the "de-contracting" and "de-pinching".

First, let Σ' be the set of p-simplices of $\tilde{K}$ whose both oriented simplices are in $\vec{\zeta_{\mathrm{j}}}$. For a p-simplex σ^p of Σ', we can let $\mathbb{B}'$ be a topological $(p+1)$-ball residing in $\mathbb{R}^{p+1}$ such that $\mathsf{bd}(\mathbb{B}')$ equals two p-simplices with boundaries glued together. We then homeomorphically map points of $\mathbb{R}^{p+1} \smallsetminus \sigma^p$ to $\mathbb{R}^{p+1} \smallsetminus \mathbb{B}'$. By taking care of the mapping near the boundary of $\mathbb{B}'$, we can get a new ambient $\mathbb{R}^{p+1}$ and a new Δ-complex where all simplices of $\tilde{K}$ are untouched except that σ^p now corresponds to the two p-simplices bounding $\mathbb{B}'$. We can also think of the above process as "de-contracting" the topological p-ball σ^p into the topological $(p+1)$-ball $\mathbb{B}'$ so that σ^p turns into two separate p-simplices with identical $(p-1)$-faces (see Figure 2.5a for an example). After doing the "de-contraction" for all p-simplices in Σ', we get a Δ-complex $\tilde{K}'$. It is true that an oriented boundary p-simplex in $\tilde{K}$ can be naturally identified as an oriented boundary p-simplex in $\tilde{K}'$. It is also true that the groups of oriented boundary p-simplices in $\tilde{K}$ are still groups of oriented boundary p-simplices in $\tilde{K}'$ under the natural identification. So we can let $\vec{\zeta_{\mathrm{j}}}$ denote the same group of oriented p-simplices in $\tilde{K}'$. The construction guarantees that if $\vec{\zeta_{\mathrm{j}}}$ is the boundary of a void of $\mathbb{R}^{p+1} \smallsetminus |\tilde{K}'|$, then $\vec{\zeta_{\mathrm{j}}}$ is also the boundary of a void of $\mathbb{R}^{p+1} \smallsetminus |\tilde{K}|$. So we only need to show that $\vec{\zeta_{\mathrm{j}}}$ is the boundary of a void of $\mathbb{R}^{p+1} \smallsetminus |\tilde{K}'|$ (see Figure 2.5b for an example). From now on, we always treat $\vec{\zeta_{\mathrm{j}}}$ as a set of oriented p-simplices as well as a p-cycle (with $\mathbb{Z}$ coefficients) in $\tilde{K}'$.

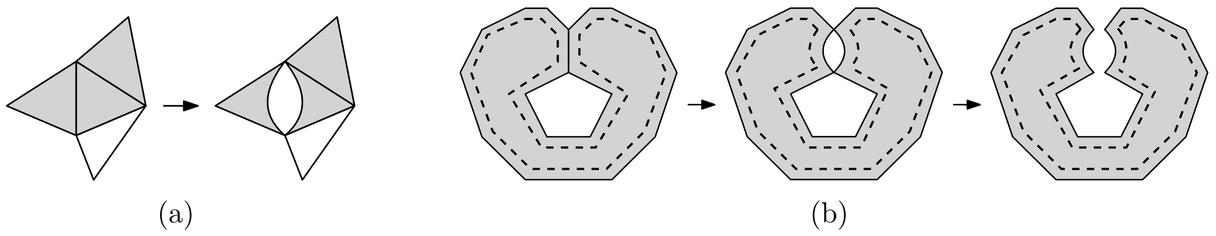


Figure 2.5. (a) An example of the "de-contraction" of σ^p for $p = 1$, where a 1-simplex in the left simplicial complex turns into two curved 1-simplices with identical boundary in the right Δ-complex. The topological 2-ball $\mathbb{B}'$ is the one bounded by the two curved 1-simplices. (b) Left to middle: An example demonstrating the void boundary correspondence from $\tilde{K}$ to $\tilde{K}'$ for $p = 1$. After a 1-simplex is de-contracted, the shaded void for $\tilde{K}$ corresponds to the shaded void for $\tilde{K}'$ and their boundaries (dashed line) can be identified. Middle to right: The "de-pinching" properly separates apart incident edges (1-simplices) for the two vertices (0-simplices) having more than two 1-cofaces. The complex $\mathcal{M}_h$ (on the right) then becomes a pseudomanifold. Te deform $\mathcal{M}_h$ back to $\mathcal{M}$ (in this example $\mathcal{M} = \tilde{K}'$), only points in $\mathcal{B}$ (unshaded region) are contracted.

Since different oriented simplices of $\vec{\zeta}_{\mathrm{j}}$ correspond to different unoriented simplices in $\tilde{K}'$, we define a bijection $\psi : \vec{\zeta}_{\mathrm{j}} \to \zeta$. The bijection ψ maps each oriented simplex of $\vec{\zeta}_{\mathrm{j}}$ to its corresponding unoriented simplex and ζ is the image of this mapping. We then let $\mathcal{M}$ be the closure of the simplicial set ζ. Note that ζ is a p-cycle (with $\mathbb{Z}_2$ coefficients) of $\tilde{K}'$ and $\mathcal{M}$ is a subcomplex of $\tilde{K}'$. Therefore, each $(p-1)$-simplex is a face of an even number of p-simplices in $\mathcal{M}$. We first pick a $(p-1)$-simplex σ^{p-1} of $\mathcal{M}$ such that $\left|\mathsf{cof}_p^{\mathcal{M}}(\sigma^{p-1})\right| > 2$, then pick two p-simplices σ_0^p and σ_1^p from $\mathsf{cof}_p^{\mathcal{M}}(\sigma^{p-1})$ such that $\psi^{-1}(\sigma_0^p)$ and $\psi^{-1}(\sigma_1^p)$ are paired in the void boundary reconstruction for $\tilde{K}'$. It is then true that $\sigma_0^p \cup \sigma_1^p$ forms a topological p-ball $\mathbb{B}_1^p$ containing σ^{p-1}. Forming the topological p-balls for all such pairs of p-simplices in $\mathsf{cof}_p^{\mathcal{M}}(\sigma^{p-1})$, we get a set of p-balls $\{\mathbb{B}_1^p, \dots, \mathbb{B}_\kappa^p\}$ for $\kappa = \left|\mathsf{cof}_p^{\mathcal{M}}(\sigma^{p-1})\right|\big/2$. For each i, we slightly move $\mathbb{B}_{\mathrm{i}}^p \setminus \mathsf{Int}(\sigma^{p-1})$ while keeping $\mathsf{bd}(\mathbb{B}_{\mathrm{i}}^p)$ untouched. We then take the closure of each $\mathbb{B}_{\mathrm{i}}^p \setminus \mathsf{Int}(\sigma^{p-1})$ to get a new Δ-complex $\mathcal{M}_1$ in which the $\mathbb{B}_{\mathrm{i}}^p$'s have their interiors disjoint. Note that in $\mathcal{M}_1$, σ^{p-1} now corresponds to κ different $(p-1)$-simplices sharing the boundary. We can repeat the above "de-pinching" process for each $(p-1)$-simplex having more than two p-cofaces in $\mathcal{M}$ and then get a sequence of Δ-complexes $(\mathcal{M}_0, \mathcal{M}_1, \dots, \mathcal{M}_h)$.

In the sequence, $\mathcal{M}_0 = \mathcal{M}$ and $\mathcal{M}_\mathrm{i}$ is derived from $\mathcal{M}_{\mathrm{i}-1}$ by doing the "de-pinching" on a $(p-1)$-simplex. It is then true that $\mathcal{M}_h$ is a pure p-dimensional p-connected Δ-complex where each $(p-1)$-simplex is a face of exactly two p-simplices. Since we can subdivide $\mathcal{M}_h$ to make it a simplicial complex, by Theorem 2.7.2, $|\mathcal{M}_h|$ must separate $\mathbb{R}^{p+1}$ into two connected components. Note that for each i, we can treat $\mathbb{R}^{p+1} \smallsetminus |\mathcal{M}_\mathrm{i}|$ as a subset of $\mathbb{R}^{p+1} \smallsetminus |\mathcal{M}_{\mathrm{i}+1}|$ because to deform $\mathcal{M}_{\mathrm{i}+1}$ back to $\mathcal{M}_\mathrm{i}$, we only need to contract some points in $\mathbb{R}^{p+1} \smallsetminus |\mathcal{M}_{\mathrm{i}+1}|$ to points in $|\mathcal{M}_{\mathrm{i}+1}|$. Then the connected components of $\mathbb{R}^{p+1} \smallsetminus |\mathcal{M}|$ are still connected in $\mathbb{R}^{p+1} \smallsetminus |\mathcal{M}_h|$. Since all oriented p-simplices of $\vec{\zeta}_\mathrm{j}$ bound the same void of $\mathbb{R}^{p+1} \smallsetminus |\tilde{K}'|$, we can let this void be $\mathcal{V}$. The void $\mathcal{V}$ is still connected in $\mathbb{R}^{p+1} \smallsetminus |\mathcal{M}|$ because $\mathbb{R}^{p+1} \smallsetminus |\tilde{K}'| \subseteq \mathbb{R}^{p+1} \smallsetminus |\mathcal{M}|$. Therefore, $\mathcal{V}$ is still connected in $\mathbb{R}^{p+1} \smallsetminus |\mathcal{M}_h|$. We can let $\mathcal{A}$ be the connected component of $\mathbb{R}^{p+1} \smallsetminus |\mathcal{M}_h|$ containing $\mathcal{V}$ and let $\mathcal{B}$ be the other connected component. The p-simplices in $\mathcal{M}$ and $\mathcal{M}_h$ can be identified because going from each $\mathcal{M}_\mathrm{i}$ to $\mathcal{M}_{\mathrm{i}+1}$ the interior of each p-simplex is never touched. Therefore, ζ is still a p-cycle (with $\mathbb{Z}_2$ coefficients) in $\mathcal{M}_h$. We then have that the two p-cycles (with $\mathbb{Z}$ coefficients) in $\mathcal{M}_h$, which are derived from the two consistent orientations of simplices of ζ, bound $\mathcal{A}$ and $\mathcal{B}$. Then, as one of the two p-cycles (with $\mathbb{Z}$ coefficients) derived from ζ, $\vec{\zeta}_\mathrm{j}$ must be the boundary of $\mathcal{A}$ or $\mathcal{B}$ in $\mathcal{M}_h$. We have that $\vec{\zeta}_\mathrm{j}$ bounds $\mathcal{A}$ because $\mathcal{B}$ does not contain points from $\mathcal{V}$. A fact about our construction is that to deform each $\mathcal{M}_\mathrm{i}$ back into $\mathcal{M}_{\mathrm{i}-1}$, we only need to contract points in $\mathcal{B}$. This implies that $\mathcal{A}$ is still a void of $\mathbb{R}^{p+1} \smallsetminus |\mathcal{M}|$ with boundary $\vec{\zeta}_\mathrm{j}$ (see Figure 2.5b for an example).

To prove that $\vec{\zeta}_\mathrm{j}$ is the boundary of a void of $\mathbb{R}^{p+1} \smallsetminus |\tilde{K}'|$, we only need to show that there are no oriented p-simplices which are in the boundary of $\mathcal{V}$ but do not belong to $\vec{\zeta}_\mathrm{j}$. For contradiction, suppose that there is such an oriented p-simplex $\vec{\sigma}^p$. Then $\vec{\sigma}^p$ must not be oppositely oriented to any oriented simplex of $\vec{\zeta}_\mathrm{j}$ because otherwise $\vec{\sigma}^p$ would bound another connected component of $\mathbb{R}^{p+1} \smallsetminus |\mathcal{M}|$ and thus bound another connected component of $\mathbb{R}^{p+1} \smallsetminus |\tilde{K}'|$. Let σ^p be the unoriented p-simplex of $\vec{\sigma}^p$, then $\sigma^p \notin \mathcal{M}$ because otherwise $\vec{\sigma}^p$ would be oppositely oriented to an oriented simplex of $\vec{\zeta}_\mathrm{j}$. Since $\sigma^p \notin \mathcal{M}$, the interior of σ^p must reside in $\mathbb{R}^{p+1} \smallsetminus |\mathcal{M}|$. From now on, we always treat $\mathcal{A}$ as a void of $\mathbb{R}^{p+1} \smallsetminus |\mathcal{M}|$. Then among all voids of $\mathbb{R}^{p+1} \smallsetminus |\mathcal{M}|$, the interior of σ^p resides in $\mathcal{A}$. This is because $\mathcal{A}$ is the void of $\mathbb{R}^{p+1} \smallsetminus |\mathcal{M}|$ containing $\mathcal{V}$. If σ^p resides in a void other than $\mathcal{A}$, points to either side of σ^p cannot be from $\mathcal{V}$. Since $\tilde{K}'$ is p-connected, there must be a sequence of p-simplices

$(\sigma_0^p, \ldots, \sigma_l^p)$ of $\tilde{K}'$ such that $\sigma_0^p = \sigma^p$, $\sigma_l^p \in \mathcal{M}$, and σ_i^p, $\sigma_{\mathrm{i}+1}^p$ share a $(p-1)$-face for each i such that $0 \leq \mathrm{i} < l$. Because the interior of σ_l^p is not in $\mathcal{A}$, we can let $\sigma_{l'}^p$ be the first p-simplex in the sequence whose interior is not in $\mathcal{A}$, then $l' \neq 0$ and the interior of $\sigma_{l'-1}^p$ is in $\mathcal{A}$. Let $\sigma_{l'-1}^{p-1}$ be the $(p-1)$-face shared by $\sigma_{l'-1}^p$ and $\sigma_{l'}^p$, we claim that $\sigma_{l'-1}^{p-1} \in \mathcal{M}$. If $\sigma_{l'}^p \in \mathcal{M}$, then it is obvious that $\sigma_{l'-1}^{p-1} \in \mathcal{M}$. If $\sigma_{l'}^p \notin \mathcal{M}$, then it is also true that $\sigma_{l'-1}^{p-1} \in \mathcal{M}$ because otherwise the interiors of $\sigma_{l'-1}^p$ and $\sigma_{l'}^p$ would be connected in $\mathbb{R}^{p+1} \smallsetminus |\mathcal{M}|$. Around the neighborhood of $\sigma_{l'-1}^{p-1}$ during the void boundary reconstruction for $\tilde{K}'$, any two paired oriented simplices from $\vec{\zeta}_\mathrm{j}$ enclose a region residing in $\mathcal{A}$. Because of the nature of the pairing, $\sigma_{l'-1}^p$ cannot be contained in any of the regions enclosed by the paired oriented simplices from $\vec{\zeta}_\mathrm{j}$. Since $\vec{\zeta}_\mathrm{j}$ is the boundary of the void $\mathcal{A}$ of $\mathbb{R}^{p+1} \smallsetminus |\mathcal{M}|$, all other regions in the neighborhood of $\sigma_{l'-1}^{p-1}$ must not be in $\mathcal{A}$. This implies that $\sigma_{l'-1}^p$ is not in $\mathcal{A}$, which is a contradiction. □

2.7.2 Algorithm correctness

To prove the correctness of Algorithm 2.7.1, we need two conclusions about cycles with $\mathbb{Z}_2$ coefficients. Specifically, Proposition 2.7.1 says that an embedded $(q-1)$-cycle in $\mathbb{R}^q$ separates the space and hence the two oriented simplices of a $(q-1)$-simplex in the cycle bound different voids. Proposition 2.7.2 says that a q-simplex in a q-cycle belongs to a q-connected sub-cycle of the q-cycle.

Proposition 2.7.1. *Let $q \geq 2$, ζ be a $(q-1)$-cycle (with $\mathbb{Z}_2$ coefficients) of a simplicial complex embedded in $\mathbb{R}^q$, and $\mathcal{Z}$ be the closure of the simplicial set ζ. Then for any $(q-1)$-simplex σ of ζ, the two oriented simplices of σ must bound different voids of $\mathbb{R}^q \smallsetminus |\mathcal{Z}|$.*

Proof. Consider a closed topological q-ball $\mathbb{B}$ such that $\sigma \subseteq \mathbb{B}$ and $\mathbb{B} \cap |\mathcal{Z} \smallsetminus \sigma|$ equals the boundary of σ. Let $\mathbb{B}_1$ and $\mathbb{B}_2$ be the two open half balls of $\mathbb{B}$ separated by σ. Then it is true that the two oriented simplices of σ bound different voids of $\mathbb{R}^q \smallsetminus |\mathcal{Z}|$ if and only if $\mathbb{B}_1$ and $\mathbb{B}_2$ are not connected in $\mathbb{R}^q \smallsetminus |\mathcal{Z}|$. So we only need to show that $\mathbb{B}_1$ and $\mathbb{B}_2$ are not connected in $\mathbb{R}^q \smallsetminus |\mathcal{Z}|$. Consider a filtration of $\mathcal{Z}$ where σ is the last simplex added. Because σ is a positive simplex in the filtration, by adding σ, the dimension of H_{q-1} must increase by 1. By Alexander duality, the dimension of H_0 of the complement space also increases by 1. Then $\mathbb{B}_1$ and $\mathbb{B}_2$ cannot be connected in $\mathbb{R}^q \smallsetminus |\mathcal{Z}|$. □

Proposition 2.7.2. *Let ζ be a q-cycle (with $\mathbb{Z}_2$ coefficients) of a simplicial complex where $q > 0$, then for any q-simplex σ of ζ, there must be a q-cycle ζ' (with $\mathbb{Z}_2$ coefficients) containing σ such that $\zeta' \subseteq \zeta$ and ζ' is q-connected.*

Proof. We can construct an undirected graph L for ζ, with vertices of L corresponding to the q-simplices in ζ. For each $(q-1)$-simplex σ^{q-1} which is a face of a q-simplex of ζ, let $\mathcal{N}$ be the set of q-simplices in ζ having σ^{q-1} as a face, then $|\mathcal{N}|$ must be even. We can pair q-simplices of $\mathcal{N}$ arbitrarily, and make each pair of q-simplices form an edge in L. Let C be the connected component of L containing the corresponding vertex of σ and ζ' be the q-chain corresponding to C, then ζ' must be a cycle. This is because we can pair the $(q-1)$-faces of all q-simplices in ζ' according to the edges in L, so $\partial(\zeta') = 0$. Furthermore, ζ' contains σ, $\zeta' \subseteq \zeta$, and ζ' is q-connected. □

Throughout the rest of this subsection, some of the symbols we use refer to Algorithm 2.7.1. We endow the ambient space $\mathbb{R}^{p+1}$ with a "cellular complex" structure by treating voids of $\mathbb{R}^{p+1} \smallsetminus |\tilde{K}|$ as $(p+1)$-dimensional "cells". This cellular complex of $\mathbb{R}^{p+1}$ is denoted as $\mathcal{R}^{p+1}$ and $\mathcal{R}^{p+1} = \tilde{K} \cup \{\text{voids of } \mathbb{R}^{p+1} \smallsetminus |\tilde{K}|\}$. For $\mathcal{R}^{p+1}$, most terminologies from algebraic topology for simplicial complexes are inherited with the exception that $(p+1)$-dimensional elements of $\mathcal{R}^{p+1}$ are called $(p+1)$-*cells*. Then, we can also let θ denote the bijection from $(p+1)$-cells of $\mathcal{R}^{p+1}$ to $V(G)$. To derive $\partial(\mathcal{V})$ for a void $\mathcal{V}$ of $\mathbb{R}^{p+1} \smallsetminus |\tilde{K}|$, we map oriented p-simplices in the boundary of $\mathcal{V}$ (Definition 2.7.2) to their corresponding unoriented p-simplices. Then $\partial(\mathcal{V})$ is defined as the sum (with $\mathbb{Z}_2$ coefficients) of these unoriented p-simplices. It is not hard to see that $\partial(\mathcal{V})$ is a p-cycle (with $\mathbb{Z}_2$ coefficients) because each void boundary is a p-cycle (with $\mathbb{Z}$ coefficients).

Proposition 2.7.3. *For any cut (S,T) of (G, s_1, s_2), the p-chain $\zeta = \theta^{-1}(\xi(S,T))$ is a persistent p-cycle for $[b, +\infty)$ and $w(\zeta) = c(S,T)$.*

Proof. We have three things to show: (i) ζ contains $\sigma_b^{\mathcal{F}}$; (ii) $w(\zeta) = c(S,T)$; (iii) ζ is a cycle. Claim (i) and (ii) are not hard to verify and we prove claim (iii) by showing that $\zeta = \sum_{\alpha \in \theta^{-1}(S)} \partial(\alpha)$, so that as a sum of cycles, ζ is a cycle. The detail for the equality of the two chains is omitted as it is similar to the one in the proof of Proposition 2.6.2. □

Proposition 2.7.4. *For any persistent p-cycle ζ for $[b, +\infty)$, there exists a cut (S, T) of (G, s_1, s_2) such that $c(S, T) \leq w(\zeta)$.*

Proof. Because of the nature of the pruning, ζ must reside in K'_b. By Proposition 2.7.2, there must be a p-cycle $\zeta' \subseteq \zeta$ such that ζ' is p-connected and contains $\sigma^{\mathcal{F}}_b$. Hence, ζ' resides in $\tilde{K}$. Let $\mathcal{Z}'$ be the closure of the simplicial set ζ', we can run the void boundary reconstruction algorithm of Section 2.7.1 on $\mathcal{Z}'$ and take a void boundary $\vec{\zeta}$ containing an oriented simplex $\vec{\sigma}^{\mathcal{F}}_b$ of $\sigma^{\mathcal{F}}_b$. We can map each oriented simplex of $\vec{\zeta}$ to its unoriented simplex and let ζ_0 be the sum of these unoriented simplices, then ζ_0 is a p-cycle (with $\mathbb{Z}_2$ coefficients) and $\zeta_0 \subseteq \zeta'$. By Proposition 2.7.1, the oppositely oriented simplex of $\vec{\sigma}^{\mathcal{F}}_b$ must not be in $\vec{\zeta}$, so ζ_0 contains $\sigma^{\mathcal{F}}_b$. Let $\vec{\zeta}$ bound a void $\mathcal{V}$ of $\mathbb{R}^{p+1} \smallsetminus |\mathcal{Z}'|$, we can let $\mathcal{A}$ be the $(p+1)$-chain of $\mathcal{R}^{p+1}$ consisting of all the $(p+1)$-cells residing in $\mathcal{V}$ and let $\mathcal{B}$ be the $(p+1)$-chain consisting of all the other $(p+1)$-cells, then $\partial(\mathcal{A}) = \partial(\mathcal{B}) = \zeta_0$. Let v_1, v_2 be the two end vertices of $\theta(\sigma^{\mathcal{F}}_b)$. Because the oppositely oriented simplex of $\vec{\sigma}^{\mathcal{F}}_b$ does not bound $\mathcal{V}$ in $\mathcal{Z}'$, it must be true that one of v_1, v_2 is in $\theta(\mathcal{A})$ and the other is in $\theta(\mathcal{B})$. We can let $(S, T) = (\theta(\mathcal{A}), \theta(\mathcal{B}))$ or $(\theta(\mathcal{B}), \theta(\mathcal{A}))$ based on which set contains the source of the flow network, then (S, T) is a cut of the flow network constructed in Algorithm 2.7.1. Furthermore, we have $\zeta_0 = \theta^{-1}(\xi(S, T))$ and $c(S, T) = w(\zeta_0) \leq w(\zeta)$. □

The following theorem concludes the correctness of Algorithm 2.7.1:

Theorem 2.7.3. *Algorithm 2.7.1 computes a minimum persistent p-cycle for the given interval $[b, +\infty)$.*

Proof. First, the flow network (G, s_1, s_2) constructed by Algorithm 2.7.1 is valid. The reason is that, by Proposition 2.7.1, it cannot happen that the two oriented simplices of $\sigma^{\mathcal{F}}_b$ bound the same void of $\mathbb{R}^{p+1} \smallsetminus |\tilde{K}|$. So $\sigma^{\mathcal{F}}_b$ must correspond to an edge of G. Then by Proposition 2.7.3 and 2.7.4, we can reach the conclusion. □

3. OPTIMAL SEQUENCES OF REPRESENTATIVES FOR LEVELSET ZIGZAG PERSISTENCE

In Chapter 2, we propose persistent cycles as concrete representatives for standard (i.e., non-zigzag) persistent homology, which also enable one to navigate back to the topological space from a barcode. We also discuss the computation of optimal persistent cycles, which are of special interest due to having guaranteed quality. However, one drawback of standard persistent cycles is that only a single cycle born at the start is used, while homological features may vary continuously inside an interval. For example, in Figure 3.1, let the growing space be the *sub-levelset filtration* of a function f, in which $\alpha_1, \ldots, \alpha_4$ are consecutive critical values and $s_0, \ldots, s_3$ are regular values in between. If we consider the changes of homology after each critical point, then a non-trivial 1-cycle z_1 is first born in $f^{-1}(-\infty, \alpha_1]$ and splits into two in $f^{-1}(-\infty, s_2]$. The two separate cycles eventually shrink and die independently, generating a (standard) persistence interval $[\alpha_1, \alpha_4)$. Using standard persistent cycles, only z_1 would be picked as a representative for $[\alpha_1, \alpha_4)$, which fails to depict the subsequent behaviors.

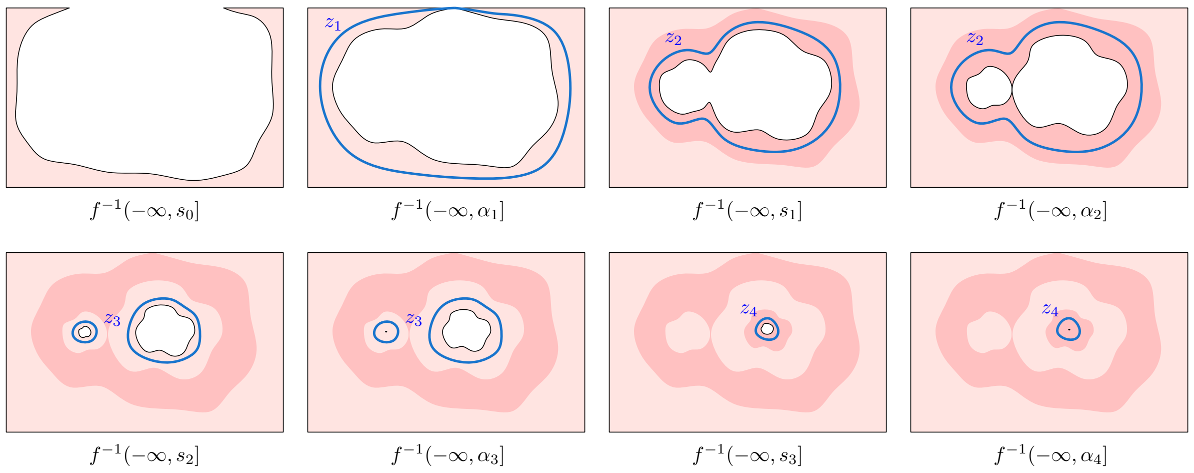


Figure 3.1. Evolution of a homological feature across different critical points.

In this chapter, we propose alternative persistent cycles capturing the dynamic behavior shown in Figure 3.1. We focus on a special but important type of persistent homology – those generated by piecewise linear (PL) functions [1]. We also base our definition on an extension of standard persistence called the *levelset zigzag* persistence [27], which tracks the survival of homological features at and in between the critical points. Given a persistence interval from levelset zigzag, we define a *sequence* of cycles called *levelset persistent cycles* so that there is a cycle between each consecutive critical points within the interval. For example, in Figure 3.1, $[\alpha_1, \alpha_4)$ is also a persistence interval (i.e., a *closed-open* interval) in the levelset zigzag of f. The cycles z_1, z_2, z_3, z_4 forming a sequence of levelset persistent 1-cycles for $[\alpha_1, \alpha_4)$ capture all the variations across the critical points. Section 3.2 details the definition.

Levelset zigzag on a PL function relates to the standard sub-levelset version in the following way: *finite* intervals from the standard version on the original function and its negation produce *closed-open* and *open-closed* intervals in levelset zigzag, while levelset zigzag additionally provides *closed-closed* and *open-open* intervals [27]. Thus, levelset persistent cycles are oriented toward richer types of intervals (see also *extended* persistence [28]).

Computationally, optimal cycle problems for homology in both persistence and non-persistence settings are NP-hard in general (see Chapter 2). Other than the optimal homology basis algorithms in dimension one [19], [51], [52], to our knowledge, all polynomial-time algorithms for such problems aim at manifold-like complexes [17], [18], [33], [36], [53]. In particular, the existing algorithms for general dimensions (see Chapter 2 and [18]) exploit the dual graph structure of given complexes and reduce the optimal cycle problem in codimension one to a minimum cut problem. In this chapter, we find a way of applying this technique to computing an optimal sequence of levelset persistent cycles – one that has the minimum *sum* of weight. Our approach which also works for general dimensions differs from algorithms proposed in Chapter 2 to account for the fact that a sequence of optimal cycles instead of a single one need to be computed.

As in Chapter 2, we also assume the input to be weak $(p + 1)$-pseudomanifold (Definition 2.2.1). Given an arbitrary PL function on a weak $(p + 1)$-pseudomanifold $(p \geq 1)$, we show that an optimal sequence of levelset persistent p-cycles can be computed in polynomial time for *any* type of levelset zigzag interval of dimension p. This is in contrast to the standard persistence setting described in Chapter 2, where computing optimal persistent p-cycles for one type of intervals (the *infinite* intervals) is NP-hard even for weak $(p+1)$-pseudomanifolds.

Note that among the four mentioned types of intervals in levelset zigzag, closed-open and open-closed intervals are symmetric so that everything concerning open-closed intervals can be derived directly from the closed-open case. Hence, for these two types of intervals, we address everything only for the closed-open case.

We propose three algorithms for the three types of intervals by utilizing minimum (s,t)-cuts on the dual graphs. Specifically, levelset persistent p-cycles for an open-open interval have direct correspondence to (s,t)-cuts on a dual graph, and so the optimal ones can be computed directly from the minimum (s,t)-cut. For the remaining cases, the crux is to deal with monkey saddles and the computation spans two phases. The first phase computes minimum p-cycles in certain components of the complex; then, using minimum cuts, the second phase determines the optimal combination of the components by introducing some *augmenting* edges.

3.1 Preliminaries

Zigzag modules, barcodes, and filtrations.

A *zigzag module* [7] (or *module* for short) is a sequence of vector spaces

$$\mathcal{M} : V_0 \leftrightarrow V_1 \leftrightarrow \cdots \leftrightarrow V_m$$

in which each $V_\mathrm{i} \leftrightarrow V_{\mathrm{i}+1}$ is a linear map and is either forward, i.e., $V_\mathrm{i} \to V_{\mathrm{i}+1}$, or backward, i.e., $V_\mathrm{i} \leftarrow V_{\mathrm{i}+1}$. In this chapter, vector spaces are taken over $\mathbb{Z}_2$. A module $\mathcal{S} : W_0 \leftrightarrow W_1 \leftrightarrow \cdots \leftrightarrow W_m$ is called a *submodule* of $\mathcal{M}$ if each W_i is a subspace of V_i and each map $W_\mathrm{i} \leftrightarrow W_{\mathrm{i}+1}$ is the restriction of $V_\mathrm{i} \leftrightarrow V_{\mathrm{i}+1}$. For an interval $[b,d] \subseteq [0,m]$, $\mathcal{S}$ is called an *interval submodule* of $\mathcal{M}$ over $[b,d]$ if W_i is one-dimensional for $\mathrm{i} \in [b,d]$ and is trivial for $\mathrm{i} \notin [b,d]$, and $W_\mathrm{i} \leftrightarrow W_{\mathrm{i}+1}$ is an isomorphism for $\mathrm{i} \in [b, d-1]$. By the Krull-Schmidt principle and Gabriel's theorem [7], $\mathcal{M}$ admits an *interval decomposition*, $\mathcal{M} = \bigoplus_{k\in\Lambda} \mathcal{I}^{[b_k,d_k]}$, in which each $\mathcal{I}^{[b_k,d_k]}$ is an interval submodule of $\mathcal{M}$ over $[b_k,d_k]$. We call the (multi-)set of intervals $\{[b_k,d_k] \mid k \in \Lambda\}$ as the *zigzag barcode* (or *barcode* for short) of $\mathcal{M}$, and denote it as $\mathsf{Pers}(\mathcal{M})$. Each interval in a zigzag barcode is called a *persistence interval*.

A *zigzag filtration* (or *filtration* for short) is a sequence of simplicial complexes or general topological spaces

$$\mathcal{X} : X_0 \leftrightarrow X_1 \leftrightarrow \cdots \leftrightarrow X_m$$

in which each $X_\mathrm{i} \leftrightarrow X_{\mathrm{i}+1}$ is either a forward inclusion $X_\mathrm{i} \hookrightarrow X_{\mathrm{i}+1}$ or a backward inclusion $X_\mathrm{i} \hookleftarrow X_{\mathrm{i}+1}$. If not mentioned otherwise, a zigzag filtration is always assumed to be a sequence of simplicial complexes. Applying the p-th homology functor with $\mathbb{Z}_2$ coefficients, the *p-th zigzag module* of $\mathcal{X}$ is induced:

$$\mathsf{H}_p(\mathcal{X}) : \mathsf{H}_p(X_0) \leftrightarrow \mathsf{H}_p(X_1) \leftrightarrow \cdots \leftrightarrow \mathsf{H}_p(X_m)$$

in which each $\mathsf{H}_p(X_\mathrm{i}) \leftrightarrow \mathsf{H}_p(X_{\mathrm{i}+1})$ is the linear map induced by inclusion. The barcode of $\mathsf{H}_p(\mathcal{X})$ is also called the *p-th zigzag barcode* of $\mathcal{X}$ and is alternatively denoted as $\mathsf{Pers}_p(\mathcal{X})$, where each interval in $\mathsf{Pers}_p(\mathcal{X})$ is called a *p-th persistence interval*. For an interval $[b, d] \in \mathsf{Pers}_p(\mathcal{X})$, we also conveniently denote the interval as $[X_b, X_d] \in \mathsf{Pers}_p(\mathcal{X})$, i.e., by its starting and ending spaces. This is specially helpful when a filtration is not naturally indexed by consecutive integers, as can be seen in Section 3.2. In this case, an element $X_\mathrm{i} \in [X_b, X_d]$ is just a space in $\mathcal{X}$ with $b \leq \mathrm{i} \leq d$.

A special type of filtration called *simplex-wise* filtration is frequently used in this chapter, in which each forward (resp. backward) inclusion is an addition (resp. deletion) of a single simplex. Any p-th zigzag module induced by a simplex-wise filtration has the property of being *elementary*, meaning that all linear maps in the module are of the three forms: (i) an isomorphism; (ii) an injection with rank 1 cokernel; (iii) a surjection with rank 1 kernel. This property is useful for the definitions and computations.

Graphs and (s,t)-cuts.

Given a graph $G = (V(G), E(G))$ and a weight function $w : E(G) \to [0, \infty]$, a *cut* (S, T) of G consists of two sets such that $S \cap T = \varnothing$ and $S \cup T = V(G)$. We define $E(S, T)$ as the set of all edges of G connecting a vertex in S and a vertex in T, in which each edge is said to *cross* the cut. The weight of the cut is defined as $w(S, T) = \sum_{\mathrm{e} \in E(S,T)} w(\mathrm{e})$. Let $\mathfrak{s}$ and $\mathfrak{t}$ be two disjoint non-empty subsets of $V(G)$; the tuple $(G, \mathscr{s}, \mathscr{t})$ is called a *weighted* (s,t)*-graph*, where $\mathscr{s}$ is the set of *sources* and $\mathscr{t}$ is the set of *sinks*. An (s,t)*-cut* (S,T) of $(G, \mathscr{s}, \mathscr{t})$ is a cut of G such that $\mathscr{s} \subseteq S$ and $\mathscr{t} \subseteq T$. The *minimum* (s,t)*-cut* of $(G, \mathscr{s}, \mathscr{t})$ is an (s,t)-cut with the minimum weight.

Dual graphs for manifolds.

A manifold-like complex (e.g., a weak pseudomanifold) often has an undirected dual graph structure, which is utilized extensively in this chapter. Let the complex be $(p+1)$-dimensional. Then, each $(p+1)$-simplex is dual to a vertex and each p-simplex is dual to an edge in the dual graph. For a p-simplex with two $(p+1)$-cofaces τ_1 and τ_2, its dual edge connects the vertex dual to τ_1 and the vertex dual to τ_2. For a p-simplex of other cases, its dual edge is problem-specific and is explained in the corresponding paragraphs.

3.2 Problem statement

In this section, we develop the definitions for levelset persistent cycles and the optimal ones. Levelset persistent cycles are sometimes simply called *persistent cycles* for brevity, and this should cause no confusion. We begin the section by defining levelset zigzag persistence in Section 3.2.1, where we present an alternative version of the classical one proposed by Carlsson et al. [27]. Adopting this alternative version enables us to focus on critical values (and the changes incurred) in a specific dimension. Section 3.2.1 also defines a simplex-wise levelset filtration, which provides an elementary view of levelset zigzag and is helpful to our definition and computation.

Section 3.2.2 details the definition of levelset persistent cycles. The cycles in the middle of the sequence are the same for all types of intervals, while the cycles for the endpoints differ according to the types of ends.

Finally, in Section 3.2.3, we address an issue left over from Section 3.2.1, which is the validity of the discrete levelset filtration. The validity is found to be relying on the triangulation representing the underlying shape. We also argue that the triangulation has to be fine enough in order to obtain accurate depictions of persistence intervals by levelset persistent cycles.

3.2.1 p-th levelset zigzag persistence

Throughout the section, let $p \geq 1$, K be a finite simplicial complex with underlying space $X = |K|$, and $f : X \to \mathbb{R}$ be a PL function [1] derived by interpolating values on vertices. We consider PL functions that are *generic*, i.e., having distinct values on the vertices. Note that the function values can be slightly perturbed to satisfy this if they are not initially. An open interval $I \subseteq \mathbb{R}$ is called *regular* if there exist a topological space Y and a homeomorphism

$$\Phi : Y \times I \to f^{-1}(I)$$

such that $f \circ \Phi$ is the projection onto I and Φ extends to a continuous function $\overline{\Phi} : Y \times \overline{I} \to f^{-1}\big(\overline{I}\big)$ with $\overline{I}$ being the closure of I [27]. It is known that f is of *Morse type* [27], meaning that each levelset $f^{-1}(s)$ has finitely generated homology, and there are finitely many *critical values*

$$\alpha_0 = -\infty < \alpha_1 < \cdots < \alpha_n < \alpha_{n+1} = \infty$$

such that each interval (α_i, α_{i+1}) is regular. Note that critical values of f can only be function values of K's vertices.

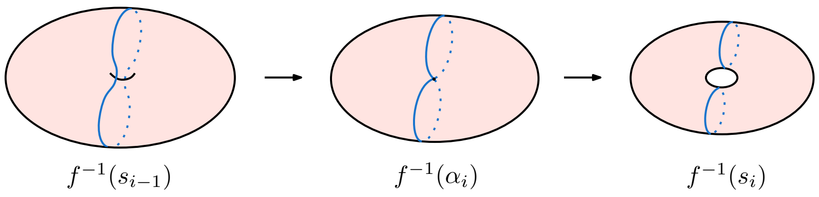


Figure 3.2. A critical value α_i across which the 2nd homology stays the same; f is defined on a 3D domain and s_{i-1}, s_i are two regular values with $s_{i-1} < \alpha_i < s_i$. The levelset $f^{-1}(s_{i-1})$ is a 2-sphere where two antipodal points are getting close and eventually pinch in $f^{-1}(\alpha_i)$. Crossing the critical value, $f^{-1}(s_i)$ becomes a torus.

As mentioned, levelset persistent cycles for a p-th interval should capture the changes of p-th homology across different critical values. However, some critical values may cause no change to the p-th homology. Figure 3.2 illustrates such a critical value for $p = 2$ around which only the 1st homology changes and the 2nd homology stays the same. Thus, to capture the most essential variation, the persistent p-cycles should stay the same across such critical values. The following definition characterizes those critical values that we are interested in:

Definition 3.2.1 (p-th homologically critical value)**.** *A critical value $\alpha_{\mathrm{i}} \neq -\infty, \infty$ of f is called a p-th homologically critical value (or p-th critical value for short) if one of the two linear maps induced by inclusion is not an isomorphism:*

$$\begin{cases} \mathsf{H}_p\big(f^{-1}(\alpha_{\mathrm{i}-1}, \alpha_{\mathrm{i}})\big) \to \mathsf{H}_p\big(f^{-1}(\alpha_{\mathrm{i}-1}, \alpha_{\mathrm{i}+1})\big) \\ \mathsf{H}_p\big(f^{-1}(\alpha_{\mathrm{i}-1}, \alpha_{\mathrm{i}+1})\big) \leftarrow \mathsf{H}_p\big(f^{-1}(\alpha_{\mathrm{i}}, \alpha_{\mathrm{i}+1})\big) \end{cases}$$

For convenience, we also let $-\infty, \infty$ be p-th critical values. Moreover, a vertex v of K is a p-th critical vertex if $f(v)$ is a p-th critical value.

Remark 3.2.1. By inspecting the (classical) levelset barcode [27] of f (also see Section 3.4.1), it can be easily determined whether a critical value is p-th critical.

Throughout this section, we let

$$\alpha_0^p = -\infty < \alpha_1^p < \cdots < \alpha_m^p < \alpha_{m+1}^p = \infty$$

denote all the p-th homologically critical values of f, and $v_1^p, \ldots, v_m^p$ denote the corresponding p-th critical vertices.

Definition 3.2.2 (p-th levelset zigzag)**.** *Denote $f^{-1}(\alpha_{\mathrm{i}}^p, \alpha_{\mathrm{j}}^p)$ as $\mathbb{X}_{(\mathrm{i},\mathrm{j})}^p$ for any $\mathrm{i} < \mathrm{j}$. The continuous version of p-th levelset filtration of f, denoted $\mathcal{L}_p^{\mathsf{c}}(f)$, is defined as*

$$\mathcal{L}_p^{\mathsf{c}}(f) : \mathbb{X}_{(0,1)}^p \hookrightarrow \mathbb{X}_{(0,2)}^p \hookleftarrow \mathbb{X}_{(1,2)}^p \hookrightarrow \mathbb{X}_{(1,3)}^p \hookleftarrow \cdots \hookrightarrow \mathbb{X}_{(m-1,m+1)}^p \hookleftarrow \mathbb{X}_{(m,m+1)}^p$$

The barcode $\mathsf{Pers}_p(\mathcal{L}_p^{\mathsf{c}}(f))$ is called the p-th levelset barcode of f, in which each interval is called a p-th levelset persistence interval of f.

Remark 3.2.2. See Figure 3.3 for an example of $\mathcal{L}_1^{\mathsf{c}}(f)$ and its 1st levelset barcode.

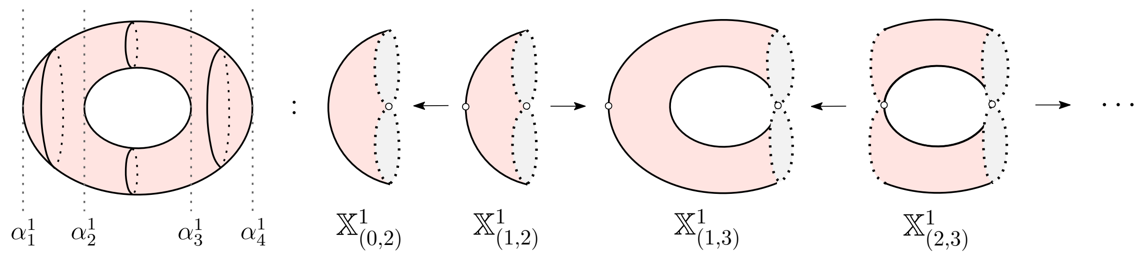


Figure 3.3. A torus with the height function f taken over the horizontal line. The 1st levelset barcode is $\left\{\left(\alpha_1^1, \alpha_4^1\right), \left[\alpha_2^1, \alpha_3^1\right]\right\}$. We list the first half of $\mathcal{L}_1^{\mathsf{c}}(f)$ but excluding $\mathbb{X}^1_{(0,1)} = \varnothing$; the remaining half is symmetric. An empty dot indicates the point is not included in the space.

In $\mathcal{L}_p^{\mathsf{c}}(f)$, $\mathbb{X}^p_{(\mathrm{i},\mathrm{i}+1)}$ is called a *p-th regular subspace*, and a homological feature in $\mathsf{H}_p(\mathbb{X}^p_{(\mathrm{i},\mathrm{i}+1)})$ is alive in the entire real-value interval $\left(\alpha^p_{\mathrm{i}}, \alpha^p_{\mathrm{i}+1}\right)$; $\mathbb{X}^p_{(\mathrm{i}-1,\mathrm{i}+1)}$ is called a *p-th critical subspace*, and a homological feature in $\mathsf{H}_p(\mathbb{X}^p_{(\mathrm{i}-1,\mathrm{i}+1)})$ is alive at the critical value α^p_{i}. Intervals in $\mathsf{Pers}_p(\mathcal{L}_p^{\mathsf{c}}(f))$ can then be mapped to real-value intervals in which the homological features persist, and are classified into four types based on the open and closeness of the ends; see Table 3.1. From now on, levelset persistence intervals can be of the two forms shown in Table 3.1, which we consider as interchangeable. We postpone the justification of Definition 3.2.2 to Section 3.4, where we prove that the p-th levelset barcode in Definition 3.2.2 is equivalent to the classical one defined in [27].

Table 3.1. Four types of intervals in $\mathsf{Pers}_p(\mathcal{L}_p^{\mathsf{c}}(f))$ and their mapping to real-value intervals.

closed-open:	$\left[\mathbb{X}^p_{(b-1,b+1)}, \mathbb{X}^p_{(d-1,d)}\right]$	$\Leftrightarrow$	$\left[\alpha^p_b, \alpha^p_d\right)$
open-closed:	$\left[\mathbb{X}^p_{(b,b+1)}, \mathbb{X}^p_{(d-1,d+1)}\right]$	$\Leftrightarrow$	$\left(\alpha^p_b, \alpha^p_d\right]$
closed-closed:	$\left[\mathbb{X}^p_{(b-1,b+1)}, \mathbb{X}^p_{(d-1,d+1)}\right]$	$\Leftrightarrow$	$\left[\alpha^p_b, \alpha^p_d\right]$
open-open:	$\left[\mathbb{X}^p_{(b,b+1)}, \mathbb{X}^p_{(d-1,d)}\right]$	$\Leftrightarrow$	$\left(\alpha^p_b, \alpha^p_d\right)$

Discrete version.

Since the optimal persistent cycles can only be computed on the discrete domain K, we provide a discrete version of our construction. First, let the subcomplex $\mathbb{K}^p_{(\mathrm{i},\mathrm{j})}$ of K denote the discrete version of $\mathbb{X}^p_{(\mathrm{i},\mathrm{j})}$:

$$\mathbb{K}^p_{(\mathrm{i},\mathrm{j})} := \Big\{\sigma \in K \mid \forall\, v \in \sigma, f(v) \in \Big(\alpha^p_{\mathrm{i}}, \alpha^p_{\mathrm{j}}\Big)\Big\} \tag{3.1}$$

We also define $\mathbb{K}^p_{[\mathrm{i},\mathrm{j})}$ and $\mathbb{K}^p_{(\mathrm{i},\mathrm{j}]}$ similarly, in which $f(v)$ in Equation (3.1) belongs to $\Big[\alpha^p_{\mathrm{i}}, \alpha^p_{\mathrm{j}}\Big)$ and $\Big(\alpha^p_{\mathrm{i}}, \alpha^p_{\mathrm{j}}\Big]$ respectively. Then, the *discrete version* of $\mathcal{L}^{\mathsf{c}}_p(f)$, denoted $\mathcal{L}_p(f)$, is defined as

$$\mathcal{L}_p(f) : \mathbb{K}^p_{(0,1)} \hookrightarrow \mathbb{K}^p_{(0,2)} \hookleftarrow \mathbb{K}^p_{(1,2)} \hookrightarrow \mathbb{K}^p_{(1,3)} \hookleftarrow \cdots \hookrightarrow \mathbb{K}^p_{(m-1,m+1)} \hookleftarrow \mathbb{K}^p_{(m,m+1)}$$

In $\mathcal{L}_p(f)$, $\mathbb{K}^p_{(\mathrm{i},\mathrm{i}+1)}$ is called a *p-th regular complex* and $\mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i}+1)}$ is called a *p-th critical complex*. At this moment, we assume that $\mathbb{X}^p_{(\mathrm{i},\mathrm{j})}$ deformation retracts to $\mathbb{K}^p_{(\mathrm{i},\mathrm{j})}$ whenever $\mathrm{i} < \mathrm{j}$, and hence $\mathcal{L}^{\mathsf{c}}_p(f)$ and $\mathcal{L}_p(f)$ are equivalent. We discuss this assumption in detail in Section 3.2.3.

Simplex-wise levelset filtration.

For defining and computing levelset persistent cycles, besides the filtration $\mathcal{L}_p(f)$, we also work on a simplex-wise version expanding $\mathcal{L}_p(f)$. We do this to harness the property that a simplex-wise filtration induces an elementary p-th module, which eliminates ambiguities in our definitions and computations.

Definition 3.2.3 (Simplex-wise levelset filtration)**.** *For the PL function f, the p-th simplex-wise levelset filtration of f, denoted $\mathcal{F}_p(f)$, is derived from $\mathcal{L}_p(f)$ by expanding each forward (resp. backward) inclusion in $\mathcal{L}_p(f)$ into a sequence of additions (resp. deletions) of a single simplex. We also let the additions and deletions follow the order of the function values:*

- *For the forward inclusion $\mathbb{K}^p_{(\mathrm{i},\mathrm{i}+1)} \hookrightarrow \mathbb{K}^p_{(\mathrm{i},\mathrm{i}+2)}$ in $\mathcal{L}_p(f)$, let $u_1 = v^p_{\mathrm{i}+1}, u_2, \ldots, u_k$ be all the vertices with function values in $\Big[\alpha^p_{\mathrm{i}+1}, \alpha^p_{\mathrm{i}+2}\Big)$ such that $f(u_1) < f(u_2) < \cdots < f(u_k)$. Then, the lower stars [1] of $u_1, \ldots, u_k$ are added by $\mathcal{F}_p(f)$ following the order.*

- *Symmetrically, for the backward inclusion* $\mathbb{K}^p_{(\mathrm{i},\mathrm{i}+2)} \hookleftarrow \mathbb{K}^p_{(\mathrm{i}+1,\mathrm{i}+2)}$ *in* $\mathcal{L}_p(f)$*, let* $u_1, u_2, \ldots, u_k = v^p_{\mathrm{i}+1}$ *be all the vertices with function values in* $\left(\alpha^p_{\mathrm{i}}, \alpha^p_{\mathrm{i}+1}\right]$ *such that* $f(u_1) < f(u_2) < \cdots < f(u_k)$*. Then, the upper stars of* $u_1, \ldots, u_k$ *are deleted by* $\mathcal{F}_p(f)$ *following the order.*

Note that for each $u_{\mathrm{j}} \in \{u_1, \ldots, u_k\}$*, we add (resp. delete) simplices inside the lower (resp. upper) star of* u_{j} *in any order maintaining the condition of a filtration.*

In this chapter, we always assume a *fixed* $\mathcal{F}_p(f)$ derived from $\mathcal{L}_p(f)$. It is always of the form

$$\mathcal{F}_p(f) : K_0 \leftrightarrow K_1 \leftrightarrow \cdots \leftrightarrow K_r$$

where each K_{i}, $K_{\mathrm{i}+1}$ differ by a simplex denoted σ_{i} and each linear map is denoted as $\varphi_{\mathrm{i}} : \mathsf{H}_p(K_{\mathrm{i}}) \leftrightarrow \mathsf{H}_p(K_{\mathrm{i}+1})$. Note that each complex in $\mathcal{L}_p(f)$ equals a K_{j} in $\mathcal{F}_p(f)$, and specifically, $K_0 = \mathbb{K}^p_{(0,1)}$, $K_r = \mathbb{K}^p_{(m,m+1)}$.

Simplex-wise intervals.

The property of zigzag persistence indicates that any interval J in $\mathsf{Pers}_p(\mathcal{L}_p(f))$ can be considered as produced by an interval J' in $\mathsf{Pers}_p(\mathcal{F}_p(f))$, and we call J' the *simplex-wise interval* of J. The mapping of intervals of $\mathsf{Pers}_p(\mathcal{F}_p(f))$ to those of $\mathsf{Pers}_p(\mathcal{L}_p(f))$ has the following rule:

For any $[K_\beta, K_\delta] \in \mathsf{Pers}_p(\mathcal{F}_p(f))$*, let* $\mathsf{F}^{[\beta,\delta]} : K_\beta \leftrightarrow K_{\beta+1} \leftrightarrow \cdots \leftrightarrow K_\delta$ *be the part of* $\mathcal{F}_p(f)$ *between* K_β *and* K_δ*, and let* $\mathbb{K}^p_{(b,b')}$ *and* $\mathbb{K}^p_{(d,d')}$ *respectively be the first and last complex from* $\mathcal{L}_p(f)$ *which appear in* $\mathsf{F}^{[\beta,\delta]}$*. Then,* $[K_\beta, K_\delta]$ *produces an interval* $\left[\mathbb{K}^p_{(b,b')}, \mathbb{K}^p_{(d,d')}\right]$ *for* $\mathsf{Pers}_p(\mathcal{L}_p(f))$*. However, if* $\mathsf{F}^{[\beta,\delta]}$ *contains no complexes from* $\mathcal{L}_p(f)$*, then* $[K_\beta, K_\delta]$ *does not produce any levelset persistence interval; such an interval in* $\mathsf{Pers}_p(\mathcal{F}_p(f))$ *is called* trivial.

As can be seen later, any levelset persistent cycles in this chapter are defined on *both* a levelset persistence interval and its simplex-wise interval. We further note that persistent cycles for trivial intervals in $\mathsf{Pers}_p(\mathcal{F}_p(f))$ are exactly the same as standard persistent cycles, and we refer to [53] for their definition and computation.

3.2.2 Definition of levelset persistent cycles

Consider a levelset persistence interval in $\mathsf{Pers}_p(\mathcal{L}_p(f))$ with endpoints α^p_b, α^p_d produced by a simplex-wise interval $[K_\beta, K_\delta] \in \mathsf{Pers}_p(\mathcal{F}_p(f))$. The levelset persistence interval can also be denoted as $\left[\mathbb{K}^p_{(b',b+1)}, \mathbb{K}^p_{(d-1,d')}\right]$, where $b' = b$ or $b-1$, and $d' = d$ or $d+1$ (see Table 3.1). A sequence of levelset persistent cycles should achieve the following for the goal:

1. Reflect the changes of homological features across all p-th critical values between α^p_b and α^p_d.

2. Capture the critical events at the birth and death points.

For the first requirement, we add to the sequence the following p-cycles:

$$z_{\mathrm{i}} \subseteq \mathbb{K}^p_{(\mathrm{i},\mathrm{i}+1)} \quad \text{for each } b \leq \mathrm{i} < d$$

because $\mathbb{K}^p_{(\mathrm{i},\mathrm{i}+1)}$ is the complex between α^p_{i} and $\alpha^p_{\mathrm{i}+1}$. This is the same for all types of intervals. However, for the second requirement, we have to separately address the differently types of ends, and there are the following cases:

Open birth: The starting complex of the levelset persistence interval is $\mathbb{K}^p_{(b,b+1)}$. We require the corresponding p-cycle z_b in $\mathbb{K}^p_{(b,b+1)}$ to become a boundary when included back into $\mathbb{K}^p_{(b-1,b+1)}$, so that it represents a new-born class in $\mathsf{H}_p(\mathbb{K}^p_{(b,b+1)})$. In $\mathcal{F}_p(f)$, the inclusion is further expanded as follows, where the birth happens at $K_{\beta-1} \hookleftarrow K_\beta$:

$$\mathbb{K}^p_{(b-1,b+1)} \hookleftarrow \cdots \hookleftarrow K_{\beta-1} \hookleftarrow K_\beta \hookleftarrow \cdots \hookleftarrow \mathbb{K}^p_{(b,b+1)}$$

We also consider z_b as a p-cycle in K_β because $\mathbb{K}^p_{(b,b+1)} \subseteq K_\beta$; then, in $\mathcal{F}_p(f)$, $[z_b] \in \mathsf{H}_p(K_\beta)$ should be the non-zero class in the kernel of $\varphi_{\beta-1} : \mathsf{H}_p(K_{\beta-1}) \leftarrow \mathsf{H}_p(K_\beta)$ in order to the capture the birth event.

Open death: Symmetrically to open birth, the corresponding p-cycle z_{d-1} in the ending complex $\mathbb{K}^p_{(d-1,d)}$ should become a boundary (i.e., die) entering into $\mathbb{K}^p_{(d-1,d+1)}$. The inclusion is further expanded as follows in the simplex-wise filtration, where the death happens at $K_\delta \hookrightarrow K_{\delta+1}$:

$$\mathbb{K}^p_{(d-1,d)} \hookrightarrow \cdots \hookrightarrow K_\delta \hookrightarrow K_{\delta+1} \hookrightarrow \cdots \hookrightarrow \mathbb{K}^p_{(d-1,d+1)}$$

To capture the death event, $[z_{d-1}] \in \mathsf{H}_p(K_\delta)$ should be the non-zero class in the kernel of φ_δ, where we also consider z_{d-1} as a p-cycle in K_δ.

Closed birth: The starting complex of the levelset persistence interval is $\mathbb{K}^p_{(b-1,b+1)}$, and the birth event happens when $\mathbb{K}^p_{(b-1,b)}$ is included into $\mathbb{K}^p_{(b-1,b+1)}$. The inclusion is further expanded as follows:

$$\mathbb{K}^p_{(b-1,b)} \hookrightarrow \cdots \hookrightarrow K_{\beta-1} \hookrightarrow K_\beta \hookrightarrow \cdots \hookrightarrow \mathbb{K}^p_{(b-1,b+1)}$$

In the simplex-wise filtration, the birth happens at the inclusion $K_{\beta-1} \hookrightarrow K_\beta$. Since no $z_\mathrm{i} \subseteq \mathbb{K}^p_{(\mathrm{i},\mathrm{i}+1)}$ for $b \le \mathrm{i} < d$ can be considered as a p-cycle in K_β (see Proposition 3.2.1), we add to the sequence a new-born p-cycle z_{b-1} in K_β to capture the birth, which is equivalent to saying that z_{b-1} contains the simplex $\sigma_{\beta-1}$ (note that $\sigma_{\beta-1}$ is a p-simplex; see [27]).

Closed death: Symmetrically to closed birth, the death happens when the last complex $\mathbb{K}^p_{(d-1,d+1)}$ turns into $\mathbb{K}^p_{(d,d+1)}$ because of the deletion, which is at $K_\delta \hookleftarrow K_{\delta+1}$ in $\mathcal{F}_p(f)$:

$$\mathbb{K}^p_{(d-1,d+1)} \hookleftarrow \cdots \hookleftarrow K_\delta \hookleftarrow K_{\delta+1} \hookleftarrow \cdots \hookleftarrow \mathbb{K}^p_{(d,d+1)}$$

Since no p-cycles defined above are considered to come from K_δ (Proposition 3.2.1), we add to the sequence a p-cycle z_d in $K_\delta \subseteq \mathbb{K}^p_{(d-1,d+1)}$ containing σ_δ, so that it represents a class disappearing in $K_{\delta+1}$ (and hence in $\mathbb{K}^p_{(d,d+1)}$). Note that σ_δ is a p-simplex [27].

Proposition 3.2.1. *If the given levelset persistence interval is closed at birth end, then $K_\beta \subseteq \mathbb{K}^p_{(b-1,b]}$ so that each $\mathbb{K}^p_{(\mathrm{i},\mathrm{i}+1)}$ for $b \le \mathrm{i} < d$ is disjoint with K_β. Similarly, if the persistence interval is closed at death end, then $K_\delta \subseteq \mathbb{K}^p_{[d,d+1)}$ so that each $\mathbb{K}^p_{(\mathrm{i},\mathrm{i}+1)}$ for $b \le \mathrm{i} < d$ is disjoint with K_δ.*

Remark 3.2.3. Note that the disjointness of these complexes also makes computation of the optimal persistent cycles feasible; see Section 3.3.

Proof. See Appendix A.2.1. □

One final thing left for the definition is to relate two consecutive p-cycles z_{i}, $z_{\mathrm{i}+1}$ in the sequence. It can be verified that both z_{i}, $z_{\mathrm{i}+1}$ reside in $\mathbb{K}^p_{(\mathrm{i},\mathrm{i}+2)}$, and hence we require them to be homologous in $\mathbb{K}^p_{(\mathrm{i},\mathrm{i}+2)}$. In this way, we have

$$[z_{\mathrm{i}}] \mapsto [z_{\mathrm{i}}] = [z_{\mathrm{i}+1}] \hookleftarrow [z_{\mathrm{i}+1}]$$

under the linear maps

$$\mathsf{H}_p\left(\mathbb{K}^p_{(\mathrm{i},\mathrm{i}+1)}\right) \to \mathsf{H}_p\left(\mathbb{K}^p_{(\mathrm{i},\mathrm{i}+2)}\right) \leftarrow \mathsf{H}_p\left(\mathbb{K}^p_{(\mathrm{i}+1,\mathrm{i}+2)}\right)$$

so that all p-cycles in the sequence represent corresponding homology classes.

For easy reference, we formally present the definitions individually for the three types of intervals:

Definition 3.2.4 (Open-open case). *For an open-open $\left(\alpha^p_b, \alpha^p_d\right) \in \mathsf{Pers}_p(\mathcal{L}_p(f))$ produced by a simplex-wise interval $[K_\beta, K_\delta]$, the levelset persistent p-cycles is a sequence $z_b, z_{b+1}, \ldots, z_{d-1}$ such that: (i) each $z_{\mathrm{i}} \subseteq \mathbb{K}^p_{(\mathrm{i},\mathrm{i}+1)}$; (ii) $[z_b] \in \mathsf{H}_p(K_\beta)$ is the non-zero class in the kernel of $\varphi_{\beta-1} : \mathsf{H}_p(K_{\beta-1}) \leftarrow \mathsf{H}_p(K_\beta)$; (iii) $[z_{d-1}] \in \mathsf{H}_p(K_\delta)$ is the non-zero class in the kernel of $\varphi_\delta : \mathsf{H}_p(K_\delta) \to \mathsf{H}_p(K_{\delta+1})$; (iv) each consecutive z_{i}, $z_{\mathrm{i}+1}$ are homologous in $\mathbb{K}^p_{(\mathrm{i},\mathrm{i}+2)}$.*

Definition 3.2.5 (Closed-open case). *For a closed-open $\left[\alpha^p_b, \alpha^p_d\right) \in \mathsf{Pers}_p(\mathcal{L}_p(f))$ produced by a simplex-wise interval $[K_\beta, K_\delta]$, the levelset persistent p-cycles is a sequence $z_{b-1}, z_b, \ldots, z_{d-1}$ such that: (i) $z_{b-1} \subseteq K_\beta$ and $\sigma_{\beta-1} \in z_{b-1}$; (ii) $z_{\mathrm{i}} \subseteq \mathbb{K}^p_{(\mathrm{i},\mathrm{i}+1)}$ for each $\mathrm{i} \geq b$; (iii) $[z_{d-1}] \in \mathsf{H}_p(K_\delta)$ is the non-zero class in the kernel of $\varphi_\delta : \mathsf{H}_p(K_\delta) \to \mathsf{H}_p(K_{\delta+1})$; (iv) each consecutive z_{i}, $z_{\mathrm{i}+1}$ are homologous in $\mathbb{K}^p_{(\mathrm{i},\mathrm{i}+2)}$.*

Definition 3.2.6 (Closed-closed case). *For a closed-closed $\left[\alpha^p_b, \alpha^p_d\right] \in \mathsf{Pers}_p(\mathcal{L}_p(f)$ produced by a simplex-wise interval $[K_\beta, K_\delta]$, the levelset persistent p-cycles is a sequence $z_{b-1}, z_b, \ldots, z_d$*

such that: (i) $z_{b-1} \subseteq K_\beta$ and $\sigma_{\beta-1} \in z_{b-1}$; (ii) $z_d \subseteq K_\delta$ and $\sigma_\delta \in z_d$; (iii) $z_\mathrm{i} \subseteq \mathbb{K}^p_{(\mathrm{i},\mathrm{i}+1)}$ for each $b \leq \mathrm{i} < d$; (iv) each consecutive z_i, $z_{\mathrm{i}+1}$ are homologous in $\mathbb{K}^p_{(\mathrm{i},\mathrm{i}+2)}$.

Figure 3.1 illustrates a sequence of levelset persistent 1-cycles for a closed-open interval, where z_1 captures the birth event (created by the corresponding 1st critical vertex[1]) and z_2, z_3, z_4 are the ones in the 1st regular complexes. The cycle z_4, which becomes a boundary when the last critical vertex is added, captures the death event. See Figure 3.5 and 3.7 in Section 3.3 for examples of other types of intervals.

To see that levelset persistent cycles actually "represent" an interval, we show that each such sequence induces an interval submodule so that all these interval submodules form an interval decomposition for $\mathsf{H}_p(\mathcal{L}_p(f))$ and $\mathsf{H}_p(\mathcal{F}_p(f))$. The details are provided in Section 3.5.

Optimal levelset persistent cycles.

To define optimal cycles, we assign weights to p-cycles of K as follows: let each p-simplex σ of K have a non-negative finite weight $w(\sigma)$; then, a p-cycle z of K has the weight $w(z) := \sum_{\sigma \in z} w(\sigma)$.

Definition 3.2.7. *For an interval of $\mathsf{Pers}_p(\mathcal{L}_p(f))$, an optimal sequence of levelset persistent p-cycles is one with the minimum sum of weight.*

3.2.3 Validity of discrete levelset filtrations

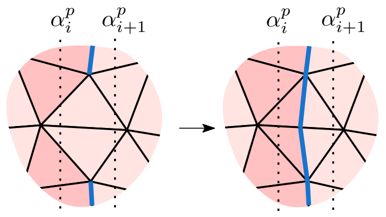


Figure 3.4. Finer triangulation makes the discrete levelset filtration equivalent with the continuous one.

One thing left over from Section 3.2.1 is to justify the validity of the discrete version of p-th levelset filtration. It turns out that the validity depends on the triangulation of K. For

[1]In the discrete setting, z_1 is indeed created by an edge incident to the critical vertex.

example, let K be the left complex in Figure 3.4; then, $\mathbb{K}^p_{(\mathrm{i},\mathrm{i}+1)}$ (the blue part) is not homotopy equivalent to $\mathbb{X}^p_{(\mathrm{i},\mathrm{i}+1)}$ (the part between the dashed lines), and hence $\mathcal{L}_p(f)$ is not equivalent to $\mathcal{L}^{\mathsf{c}}_p(f)$. We observe that the non-equivalence is caused by the two central triangles which contain more than one critical value. A subdivision of the two central triangles on the right (so that no triangles contain more than one critical value) renders $\mathbb{X}^p_{(\mathrm{i},\mathrm{i}+1)}$ deformation retracting to $\mathbb{K}^p_{(\mathrm{i},\mathrm{i}+1)}$. Based on the above observation, we formulate the following property, which guarantees the equivalence of modules induced by $\mathcal{L}_p(f)$ and $\mathcal{L}^{\mathsf{c}}_p(f)$:

Definition 3.2.8. *The complex K is said to be compatible with the p-th levelsets of the PL function f if for any simplex σ of K and its convex hull $|\sigma|$, function values of points in $|\sigma|$ include at most one p-th critical value of f.*

Proposition 3.2.2. *If K is compatible with the p-th levelsets of f, then $\mathbb{X}^p_{(\mathrm{i},\mathrm{j})}$ deformation retracts to $\mathbb{K}^p_{(\mathrm{i},\mathrm{j})}$ for any $\mathrm{i} < \mathrm{j}$, which implies that $\mathsf{H}_p(\mathcal{L}_p(f))$ and $\mathsf{H}_p(\mathcal{L}^{\mathsf{c}}_p(f))$ are isomorphic.*

Proof. See Appendix A.2.2. $\square$

In this chapter, we only consider the situation where a complex is compatible with the p-th levelsets of its PL function. We regard this assumption reasonable because in the discrete optimization, the quality of computed output depends on the triangulation of underlying space. When the assumption is violated, it becomes impossible to depict certain changes of homological features on the discrete domain. Note that a complex can be refined to become compatible if it is not already.

3.3 Computation

In this section, given a weak $(p+1)$-pseudomanifold with $p \geq 1$, we present algorithms that compute an optimal sequence of levelset persistent p-cycles for a p-th interval. Though the computation for all types of intervals is based on minimum cuts, we address the algorithm for each type separately in each subsection. The reasons are as follows. First, one has to choose a subcomplex to work on in order to build a dual graph for the minimum cut computation. In the open-open case, the subcomplex is always a $(p+1)$-*pseudomanifold* without boundary (see Section 3.3.1) whose dual graph is obvious; in the other cases, however, we do not have such convenience and the dual graph construction is more involved. Also, the closed-open case has to deal with the so-called "monkey saddles" and the solution adopts a two-phase approach (see Section 3.3.2); in the open-open case, however, no such issues occur and the algorithm is much simpler. We also note that even for standard persistent cycles which have simpler definitions, the hardness results and the algorithms for the *finite* and *infinite* intervals are still different; see [53]. With all being said, we observe that the computation for the closed-closed case does exhibit resemblance to the closed-open case and is only described briefly; see Section 3.3.3.

Other than the type of persistence interval, all subsections make the same assumptions on input as the following:

- $p \geq 1$ is the dimension of interest.
- K is a finite weak $(p+1)$-pseudomanifold with a finite weight $w(\sigma) \geq 0$ for each p-simplex σ.
- $f : |K| \to \mathbb{R}$ is a generic PL function with p-th critical values $\alpha_0^p = -\infty < \alpha_1^p < \cdots < \alpha_m^p < \alpha_{m+1}^p = \infty$ and corresponding p-th critical vertices $v_1^p, \ldots, v_m^p$. We also assume that K is compatible with the p-th levelsets of f.
- $\mathcal{F}_p(f) : K_0 \leftrightarrow K_1 \leftrightarrow \cdots \leftrightarrow K_r$ is a fixed simplex-wise levelset filtration. Each K_i, $K_{\mathrm{i}+1}$ in $\mathcal{F}_p(f)$ differ by a simplex σ_i, and each linear map in $\mathsf{H}_p(\mathcal{F}_p(f))$ is denoted as $\varphi_\mathrm{i} : \mathsf{H}_p(K_\mathrm{i}) \leftrightarrow \mathsf{H}_p(K_{\mathrm{i}+1})$.

3.3.1 Open-open case

Throughout this subsection, assume that we aim to compute the optimal persistent p-cycles for an *open-open* interval $\left(\alpha_b^p, \alpha_d^p\right)$ from $\mathsf{Pers}_p(\mathcal{L}_p(f))$, which is produced by a simplex-wise interval $[K_\beta, K_\delta]$ from $\mathsf{Pers}_p(\mathcal{F}_p(f))$. Figure 3.5 illustrates a sequence of persistent 1-cycles z_1, z_2, z_3 for an open-open interval $\left(\alpha_1^1, \alpha_4^1\right)$.

As seen from Section 3.2.2, the following portion of $\mathcal{F}_p(f)$ is relevant to the definition (and hence the computation):

$$\begin{tikzcd}
\mathbb{K}^p_{(b-1,b+1)} & K_{\beta-1} \arrow[l, hook', dashed] & K_\beta \arrow[l, hook', "\sigma_{\beta-1}"'] & \mathbb{K}^p_{(b,b+1)} \arrow[l, hook', dashed] \arrow[r, hook, dashed] & \cdots \\
{} & \mathbb{K}^p_{(d-1,d)} \arrow[l, hook', dashed] \arrow[r, hook, dashed] & K_\delta \arrow[r, hook, "\sigma_\delta"] & K_{\delta+1} \arrow[r, hook, dashed] & \mathbb{K}^p_{(d-1,d+1)}
\end{tikzcd} \tag{3.2}$$

In the above sequence, the non-dashed hooked arrows indicate the addition or deletion of one simplex, while the dashed arrows indicate the addition or deletion of zero or more simplices. The simplices $\sigma_{\beta-1}$, σ_δ are the ones creating and destroying the simplex-wise interval, which are both $(p+1)$-simplices [27]. We further restrict the computation to (a $(p+1)$-connected component of) $\mathbb{K}^p_{(b-1,d+1)}$ considering that each complex in Sequence (3.2) is a subcomplex of $\mathbb{K}^p_{(b-1,d+1)}$.

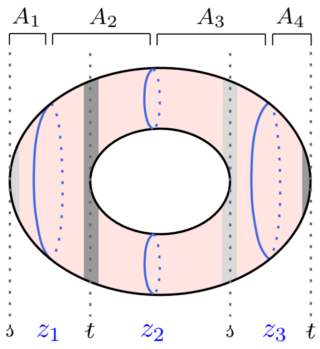


Figure 3.5. A sequence of levelset persistent 1-cycles for an open-open interval $\left(\alpha_1^1, \alpha_4^1\right)$; the complex (assume the torus to be finely triangulated), the function, and the 1st critical values are the same as in Figure 3.3.

We now describe the algorithm. Since the deletion of the $(p+1)$-simplex $\sigma_{\beta-1}$ gives birth to the interval $[K_\beta, K_\delta]$, $\sigma_{\beta-1}$ must be relevant to our computation. So we let the complex being worked on, denoted K', be the closure of the $(p+1)$-connected component of K containing $\sigma_{\beta-1}$. (Note that the closure of a set of simplices consists of all faces of the simplices in the set.) Based on the property of open-open intervals, K' must be a $(p+1)$-pseudomanifold without boundary (see Proposition 3.3.1, Claim 3). Letting G be the dual graph of K', we build a weighted (s,t)-graph $(G,\mathfrak{s},t)$. To set up the sources and sinks, we define the set $\mathbb{K}^p_{(\mathrm{i})}$ of simplices as follows:

$$\mathbb{K}^p_{(\mathrm{i})} := \mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i}+1)} \setminus \left(\mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i})} \cup \mathbb{K}^p_{(\mathrm{i},\mathrm{i}+1)}\right)$$

Roughly speaking, $\mathbb{K}^p_{(\mathrm{i})}$ consists of simplices containing the critical value α^p_{i} (for example, the darker triangles in Figure 3.4 belong to $\mathbb{K}^p_{(\mathrm{i})}$), and note that $\mathbb{K}^p_{(\mathrm{i})}$ may not be a simplicial complex. Based on the above definition, we alternately put vertices dual to the $(p+1)$-simplices in $\mathbb{K}^p_{(b)}, \dots, \mathbb{K}^p_{(d)}$ into sources and sinks. For our example in Figure 3.5 where K' is the entire torus, the source $\mathfrak{s}$ contains vertices dual to 2-simplices in $\mathbb{K}^1_{(1)} \cup \mathbb{K}^1_{(3)}$, and the sink t contains vertices dual to 2-simplices in $\mathbb{K}^1_{(2)} \cup \mathbb{K}^1_{(4)}$. Note that $\mathbb{K}^1_{(1)}, \dots, \mathbb{K}^1_{(4)}$ are alternately shaded with light and dark gray in Figure 3.5.

The correctness of the above construction is based on the duality of the levelset persistent p-cycles and (s,t)-cuts on $(G,\mathfrak{s},t)$. To see the duality, first consider the sequence of persistent 1-cycles z_1, z_2, z_3 in Figure 3.5. By Definition 3.2.4, there exist 2-chains $A_1 \subseteq \mathbb{K}^1_{(0,2)}$, $A_2 \subseteq \mathbb{K}^1_{(1,3)}$, $A_3 \subseteq \mathbb{K}^1_{(2,4)}$, and $A_4 \subseteq \mathbb{K}^1_{(3,5)}$ as shown in the figure such that $z_1 = \partial(A_1)$, $z_1 + z_2 = \partial(A_2)$, $z_2 + z_3 = \partial(A_3)$, and $z_3 = \partial(A_4)$. Let S contain the vertices dual to $A_1 + A_3$ and T contain the vertices dual to $A_2 + A_4$. Then, (S,T) is an (s,t)-cut of $(G,\mathfrak{s},t)$. Since edges in $E(S,T)$ are dual to 1-simplices in $z_1 + z_2 + z_3$, we have that $w(S,T) = w(z_1) + w(z_2) + w(z_3)$. So we derive a cut (S,T) dual to the persistent 1-cycles z_1, z_2, z_3. On the other hand, an (s,t)-cut of $(G,\mathfrak{s},t)$ produces a sequence of persistent p-cycles for the given interval. For example, let (S,T) be a cut where S contains the graph vertices in $A_1 + A_3$ and T contains the graph vertices in $A_2 + A_4$, as in Figure 3.5. We then take the intersection of the dual 1-simplices of $E(S,T)$ with $\mathbb{K}^1_{(1,2)}, \mathbb{K}^1_{(2,3)}, \mathbb{K}^1_{(3,4)}$. The resulting 1-chains z_1, z_2, z_3 is a sequence of persistent 1-cycles for the interval $\left(\alpha^1_1, \alpha^1_4\right)$. Hence, by the duality, a minimum (s,t)-cut of $(G,\mathfrak{s},t)$ always produces an optimal sequence of levelset persistent p-cycles.

We formally list the pseudocode as follows:

Algorithm 3.3.1. *Given the input as specified, the algorithm does the following:*

1. *Let K' be the closure of the $(p+1)$-connected component of K containing $\sigma_{\beta-1}$. Note that K' is a $(p+1)$-pseudomanifold without boundary (see Proposition 3.3.1, Claim 3).*

2. *Build a weighted dual graph G of K', where $V(G)$ corresponds to $(p+1)$-simplices of K' and $E(G)$ corresponds to p-simplices of K'. Let θ denote both the bijection from the $(p+1)$-simplices to $V(G)$ and the bijection from the p-simplices to $E(G)$. For each edge e of G, if $\theta^{-1}(\mathrm{e}) \in \mathbb{K}^p_{(\mathrm{i},\mathrm{i}+1)}$ for i s.t. $b \leq \mathrm{i} < d$, then set $w(\mathrm{e})$, the weight of e, as $w(\theta^{-1}(\mathrm{e}))$; otherwise, set $w(\mathrm{e}) = \infty$.*

3. *For each i s.t. $b \leq \mathrm{i} \leq d$, let Δ_i denote the set of $(p+1)$-simplices in $K' \cap \mathbb{K}^p_{(\mathrm{i})}$. Also, let L_e be the set of even integers in $[0, d-b]$ and L_o be the set of odd ones. Then, let $\mathcal{s} = \theta\Big(\bigcup_{\mathrm{i}\in L_\mathsf{e}} \Delta_{b+\mathrm{i}}\Big)$, $\mathcal{t} = \theta\Big(\bigcup_{\mathrm{i}\in L_\mathsf{o}} \Delta_{b+\mathrm{i}}\Big)$, and compute the minimum (s,t)-cut (S^*, T^*) of $(G, \mathcal{s}, \mathcal{t})$.*

4. *For each i s.t. $b \leq \mathrm{i} < d$, let $z^*_\mathrm{i} = \mathbb{K}^p_{(\mathrm{i},\mathrm{i}+1)} \cap \theta^{-1}(E(S^*, T^*))$. Return $z^*_b, \ldots, z^*_{d-1}$ as an optimal sequence of levelset persistent p-cycles for the interval $\Big(\alpha^p_b, \alpha^p_d\Big)$.*

Justification.

For the correctness of Algorithm 3.3.1, we first present Proposition 3.3.1 stating several facts about the algorithm which are used to prove the two propositions (3.3.2 and 3.3.3) on the duality. Then, Proposition 3.3.2 and 3.3.3 lead to Theorem 3.3.1, which draws the conclusion.

Proposition 3.3.1. *The following hold for Algorithm 3.3.1:*

1. *The simplex σ_δ resides in K'.*

2. *Let $z_b, \ldots, z_{d-1}$ be any sequence of persistent p-cycles for $\Big(\alpha^p_b, \alpha^p_d\Big)$; then, there exist $(p+1)$-chains $A_b \subseteq K_{\beta-1}, A_{b+1} \subseteq \mathbb{K}^p_{(b,b+2)}, \ldots, A_{d-1} \subseteq \mathbb{K}^p_{(d-2,d)}, A_d \subseteq K_{\delta+1}$ such that $\sigma_{\beta-1} \in A_b$, $\sigma_\delta \in A_d$, $z_b = \partial(A_b)$, $z_{d-1} = \partial(A_d)$, and $z_{\mathrm{i}-1} + z_\mathrm{i} = \partial(A_\mathrm{i})$ for each $b < \mathrm{i} < d$. Furthermore, let $z'_\mathrm{i} = K' \cap z_\mathrm{i}$, $A'_\mathrm{i} = K' \cap A_\mathrm{i}$ for each i; then, $\sigma_{\beta-1} \in A'_b$, $\sigma_\delta \in A'_d$, $z'_b = \partial\Big(A'_b\Big)$, $z'_{d-1} = \partial\Big(A'_d\Big)$, and $z'_{\mathrm{i}-1} + z'_\mathrm{i} = \partial\Big(A'_\mathrm{i}\Big)$ for each $b < \mathrm{i} < d$. Finally,*

one has that $A'_b + \cdots + A'_d$ equals the set of $(p+1)$-simplices of K' and $A'_b, \ldots, A'_d$ are disjoint.

3. *The complex K' is a $(p+1)$-connected $(p+1)$-pseudomanifold without boundary, i.e., each p-simplex has exactly two $(p+1)$-cofaces in K'.*

Proof. See Appendix A.2.3. $\square$

Proposition 3.3.2. *Let $z_b, \ldots, z_{d-1}$ be any sequence of levelset persistent p-cycles for $\left(\alpha^p_b, \alpha^p_d\right)$; then, there exists an (s,t)-cut (S,T) of $(G, \mathfrak{s}, \mathfrak{t})$ such that $w(S,T) \leq \sum_{\mathrm{i}=b}^{d-1} w(z_\mathrm{i})$.*

Proof. Let $A'_b, \ldots, A'_d$ and $z'_b, \ldots, z'_{d-1}$ be as specified in Claim 2 of Proposition 3.3.1 for the given $z_b, \ldots, z_{d-1}$, and let $S = \theta\Big(\sum_{\mathrm{j}\in L_\mathsf{e}} A'_{b+\mathrm{j}}\Big)$, $T = \theta\Big(\sum_{\mathrm{j}\in L_\mathsf{o}} A'_{b+\mathrm{j}}\Big)$. We first show that for a Δ_i such that $\mathrm{i}-b$ is even, Δ_i does not intersect $\sum_{\mathrm{j}\in L_\mathsf{o}} A'_{b+\mathrm{j}}$. For contradiction, suppose instead that there is a σ in both of them. Then, since $\Delta_\mathrm{i} \subseteq \mathbb{K}^p_{(\mathrm{i})} \subseteq \mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i}+1)}$ and $A'_{b+\mathrm{j}} \subseteq \mathbb{K}^p_{(b+\mathrm{j}-1,b+\mathrm{j}+1)}$ for each $\mathrm{j} \in L_\mathsf{o}$, σ must be in $A'_{\mathrm{i}-1} \subseteq \mathbb{K}^p_{(\mathrm{i}-2,\mathrm{i})}$ or $A'_{\mathrm{i}+1} \subseteq \mathbb{K}^p_{(\mathrm{i},\mathrm{i}+2)}$ because other chains in $\{A'_{b+\mathrm{j}} \mid \mathrm{j} \in L_\mathsf{o}\}$ do not intersect $\mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i}+1)}$. So we have that σ is in $\mathbb{K}^p_{(\mathrm{i}-2,\mathrm{i})}$ or $\mathbb{K}^p_{(\mathrm{i},\mathrm{i}+2)}$. The fact that $\sigma \in \Delta_\mathrm{i} \subseteq \mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i}+1)}$ implies that σ is in $\mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i})}$ or $\mathbb{K}^p_{(\mathrm{i},\mathrm{i}+1)}$, a contradiction to $\sigma \in \Delta_\mathrm{i} \subseteq \mathbb{K}^p_{(\mathrm{i})} = \mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i}+1)} \setminus \Big(\mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i})} \cup \mathbb{K}^p_{(\mathrm{i},\mathrm{i}+1)}\Big)$. So Δ_i does not intersect $\sum_{\mathrm{j}\in L_\mathsf{o}} A'_{b+\mathrm{j}}$. Then, since $\sum_{\mathrm{j}=b}^{d} A'_\mathrm{j}$ equals the set of $(p+1)$-simplices of K' by Claim 2 of Proposition 3.3.1, we have that $\Delta_\mathrm{i} \subseteq \sum_{\mathrm{j}\in L_\mathsf{e}} A'_{b+\mathrm{j}}$, i.e., $\theta(\Delta_\mathrm{i}) \subseteq S$. This means that $\mathfrak{s} \subseteq S$. Similarly, we have $\mathfrak{t} \subseteq T$. Claim 2 of Proposition 3.3.1 implies that $S \cup T = V(G)$ and $S \cap T = \varnothing$, and so (S,T) is an (s,t)-cut of $(G, \mathfrak{s}, \mathfrak{t})$. The fact that $\sum_{\mathrm{i}=b}^{d-1} z'_\mathrm{i} = \partial\Big(\sum_{\mathrm{j}\in L_\mathsf{e}} A'_{b+\mathrm{j}}\Big) = \partial\Big(\sum_{\mathrm{j}\in L_\mathsf{o}} A'_{b+\mathrm{j}}\Big)$ implies that $\sum_{\mathrm{i}=b}^{d-1} z'_\mathrm{i} = \theta^{-1}(E(S,T))$. So we have $w(S,T) = \sum_{\mathrm{i}=b}^{d-1} w(z'_\mathrm{i}) \leq \sum_{\mathrm{i}=b}^{d-1} w(z_\mathrm{i})$. $\square$

Proposition 3.3.3. *For any (s,t)-cut (S,T) of $(G, \mathfrak{s}, \mathfrak{t})$ with finite weight, let $z_\mathrm{i} = \mathbb{K}^p_{(\mathrm{i},\mathrm{i}+1)} \cap \theta^{-1}(E(S,T))$ for each i s.t. $b \leq \mathrm{i} < d$. Then, $z_b, \ldots, z_{d-1}$ is a sequence of levelset persistent p-cycles for $\left(\alpha^p_b, \alpha^p_d\right)$ with $\sum_{\mathrm{i}=b}^{d-1} w(z_\mathrm{i}) = w(S,T)$.*

Proof. We first prove that, for any i s.t. $b < \mathrm{i} < d$ and $\mathrm{i} - b$ is even, $\partial\Big(\theta^{-1}(S) \cap \mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i}+1)}\Big) = z_{\mathrm{i}-1} + z_\mathrm{i}$. To prove this, first consider any $\sigma \in \partial\Big(\theta^{-1}(S) \cap \mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i}+1)}\Big)$. We have that σ is a face of only one $(p+1)$-simplex τ_1 in $\theta^{-1}(S) \cap \mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i}+1)}$. Note that $\tau_1 \in \theta^{-1}(S) \subseteq K'$. Since K' is a $(p+1)$-pseudomanifold without boundary (Claim 3 of Proposition 3.3.1), σ

has another $(p+1)$-coface τ_2 in K'. Then, it must be true that $\tau_2 \in \theta^{-1}(T)$. To see this, suppose instead that $\tau_2 \in \theta^{-1}(S)$. Note that $\tau_2 \notin \mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i}+1)}$ because otherwise τ_2 would be in $\theta^{-1}(S) \cap \mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i}+1)}$, contradicting the fact that σ has only one $(p+1)$-coface in $\theta^{-1}(S) \cap \mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i}+1)}$. Also note that τ_2 is not in $\mathbb{K}^p_{(\mathrm{i}-1)}$ or $\mathbb{K}^p_{(\mathrm{i}+1)}$ because if τ_2 is in one of them, combining the fact that $\mathrm{i}-1-b$ and $\mathrm{i}+1-b$ are odd, we would have that τ_2 is in $\Delta_{\mathrm{i}-1}$ or $\Delta_{\mathrm{i}+1}$ and thus $\theta(\tau_2) \in \mathscr{t} \subseteq T$, which is a contradiction. Since $K' \subseteq \mathbb{K}^p_{(b-1,d+1)}$ and $\left\{\mathbb{K}^p_{(b-1,\mathrm{i}-1)}, \mathbb{K}^p_{(\mathrm{i}-1)}, \mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i}+1)}, \mathbb{K}^p_{(\mathrm{i}+1)}, \mathbb{K}^p_{(\mathrm{i}+1,d+1)}\right\}$ covers $\mathbb{K}^p_{(b-1,d+1)}$, we have that τ_2 is in $\mathbb{K}^p_{(b-1,\mathrm{i}-1)}$ or $\mathbb{K}^p_{(\mathrm{i}+1,d+1)}$. This implies that $\sigma \subseteq \tau_2$ is in $\mathbb{K}^p_{(b-1,\mathrm{i}-1)}$ or $\mathbb{K}^p_{(\mathrm{i}+1,d+1)}$, contradicting that $\sigma \subseteq \tau_1 \in \mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i}+1)}$. It is now true that $\sigma \in \theta^{-1}(E(S,T))$ because $\tau_1 \in \theta^{-1}(S)$ and $\tau_2 \in \theta^{-1}(T)$. Since (S,T) has finite weight, σ must come from a $\mathbb{K}^p_{(\mathrm{j},\mathrm{j}+1)}$ for $b \leq \mathrm{j} < d$ and thus must come from $\mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i})}$ or $\mathbb{K}^p_{(\mathrm{i},\mathrm{i}+1)}$. Then, σ is in $z_{\mathrm{i}-1}$ or z_{i}. Moreover, since $z_{\mathrm{i}-1}$ and z_{i} are disjoint, we have $\sigma \in z_{\mathrm{i}-1} + z_{\mathrm{i}}$.

On the other hand, for any $\sigma \in z_{\mathrm{i}-1} + z_{\mathrm{i}}$, first assume that $\sigma \in z_{\mathrm{i}-1} = \mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i})} \cap \theta^{-1}(E(S,T))$. Since $\sigma \in \theta^{-1}(E(S,T))$, σ must be a face of a $(p+1)$-simplex τ in $\theta^{-1}(S)$ and another $(p+1)$-simplex in $\theta^{-1}(T)$. We then show that $\tau \in \mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i}+1)}$. Suppose instead that $\tau \notin \mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i}+1)}$, and let v be the vertex belonging to τ but not σ. We have that $f(v) \notin (\alpha^p_{\mathrm{i}-1}, \alpha^p_{\mathrm{i}+1})$ because if $f(v) \in (\alpha^p_{\mathrm{i}-1}, \alpha^p_{\mathrm{i}+1})$, the fact that $\sigma \in \mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i})}$ would imply that τ is in $\mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i}+1)}$. Note that $f(v)$ cannot be greater than or equal to $\alpha^p_{\mathrm{i}+1}$ because otherwise K would not be compatible with the p-th levelsets of f. Therefore, $f(v) \leq \alpha^p_{\mathrm{i}-1}$, and it must be true that $\tau \in \mathbb{K}^p_{(\mathrm{i}-2,\mathrm{i})}$. This implies that $\tau \in \mathbb{K}^p_{(\mathrm{i}-1)}$. We now have that $\tau \in \Delta_{\mathrm{i}-1}$, where $\mathrm{i}-1-b$ is odd. Then, $\theta(\tau) \in \mathscr{t} \subseteq T$, a contradiction to $\tau \in \theta^{-1}(S)$. Combining the fact that $\tau \in \mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i}+1)}$ and τ is the only $(p+1)$-coface of σ in $\theta^{-1}(S)$, we have that τ is the only $(p+1)$-coface of σ in $\theta^{-1}(S) \cap \mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i}+1)}$. If $\sigma \in z_{\mathrm{i}}$, we can have the same result. Therefore, $\sigma \in \partial\Big(\theta^{-1}(S) \cap \mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i}+1)}\Big)$, and we have proved that $\partial\Big(\theta^{-1}(S) \cap \mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i}+1)}\Big) = z_{\mathrm{i}-1} + z_{\mathrm{i}}$.

Similarly, we can prove that $\partial\Big(\theta^{-1}(T) \cap \mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i}+1)}\Big) = z_{\mathrm{i}-1} + z_{\mathrm{i}}$ for i s.t. $b < \mathrm{i} < d$ and $\mathrm{i}-b$ is odd, $\partial\Big(\theta^{-1}(S) \cap K_{\beta-1}\Big) = z_b$, and $\partial\Big(\theta^{-1}(S) \cap K_{\delta+1}\Big) = z_{d-1}$ or $\partial\Big(\theta^{-1}(T) \cap K_{\delta+1}\Big) = z_{d-1}$ based on the parity of $d-b$. Since $\sigma_{\beta-1} \in K_{\beta-1} \subseteq \mathbb{K}^p_{[b,b+1)}$ and $\sigma_{\beta-1} \notin \mathbb{K}^p_{(b,b+1)}$, we have that $\sigma_{\beta-1} \in \mathbb{K}^p_{(b)}$, which means that $\theta(\sigma_{\beta-1}) \in \mathscr{s} \subseteq S$. Therefore, $\sigma_{\beta-1} \in \theta^{-1}(S) \cap K_{\beta-1}$. Since $\partial\Big(\theta^{-1}(S) \cap K_{\beta-1}\Big) = z_b$, we have that $z_b \sim \partial(\sigma_{\beta-1})$ in K_β, i.e., $[z_b] \in \mathsf{H}_p(K_\beta)$ is the non-zero class in $\mathsf{ker}(\varphi_{\beta-1})$. Analogously, $[z_{d-1}] \in \mathsf{H}_p(K_\delta)$ is the non-zero class in $\mathsf{ker}(\varphi_\delta)$. The above facts imply that $z_b, \ldots, z_{d-1}$ is a sequence of levelset persistent p-cycles for $\left(\alpha_b^p, \alpha_d^p\right)$. The equality of the weight follows from the disjointness of $z_b, \ldots, z_{d-1}$ and the fact that $w(S,T)$ is finite. $\square$

Theorem 3.3.1. *Algorithm 3.3.1 computes an optimal sequence of levelset persistent p-cycles for a given* open-open *interval.*

3.3.2 Closed-open case

Throughout the subsection, assume that we aim to compute the optimal persistent p-cycles for a *closed-open* interval $\left[\alpha_b^p, \alpha_d^p\right)$ from $\mathsf{Pers}_p(\mathcal{L}_p(f))$, which is produced by a simplex-wise interval $[K_\beta, K_\delta]$ from $\mathsf{Pers}_p(\mathcal{F}_p(f))$. Figure 3.6a and 3.6b provide examples for $p = 1$, where z_1', z_2', z_3' and z_1'', z_2'', z_3'' are two sequences of levelset persistent 1-cycles for the interval $\left[\alpha_2^1, \alpha_4^1\right)$.

Similar to the previous case, we have the following portion of $\mathcal{F}_p(f)$ relevant to the definition and computation:

$$\begin{tikzcd}
\mathbb{K}^p_{(b-1,b)} \arrow[r, hook, dashed] & K_{\beta-1} \arrow[r, hook, "\sigma_{\beta-1}"] & K_\beta \arrow[r, hook, dashed] & \mathbb{K}^p_{(b-1,b+1)} & \mathbb{K}^p_{(b,b+1)} \arrow[l, hook', dashed] \arrow[r, hook, dashed] & \cdots \\
& \cdots & \mathbb{K}^p_{(d-1,d)} \arrow[l, hook', dashed] \arrow[r, hook, dashed] & K_\delta \arrow[r, hook, "\sigma_\delta"] & K_{\delta+1} \arrow[r, hook, dashed] & \mathbb{K}^p_{(d-1,d]}
\end{tikzcd} \tag{3.3}$$

The creator $\sigma_{\beta-1}$ is a p-simplex and the destroyer σ_δ is a $(p+1)$-simplex [27]. Note that we end the sequence with $\mathbb{K}^p_{(d-1,d]}$ instead of $\mathbb{K}^p_{(d-1,d+1)}$ as in the case "open death" in Section 3.2.2. This is valid due to the following reasons: (i) $\mathbb{K}^p_{(d-1,d]}$ is derived from $\mathbb{K}^p_{(d-1,d)}$ by adding the lower star of v_d^p and hence must appear in $\mathcal{F}_p(f)$ based on Definition 3.2.3; (ii) $K_{\delta+1}$ is a subcomplex of $\mathbb{K}^p_{(d-1,d]}$ and the proof is similar to that of Proposition 3.2.1. Therefore, the computation can be restricted to $\mathbb{K}^p_{(b-1,d]}$ because each complex in Sequence (3.3) is a subcomplex of $\mathbb{K}^p_{(b-1,d]}$.

Overview.

To give a high-level view of our algorithm, we first use an example to illustrate several important observations. These observations provide insights into the solution and lead to the key issue. We then discuss the key issue in detail. Finally, we describe our solution in words, and present the formal pseudocode after that.

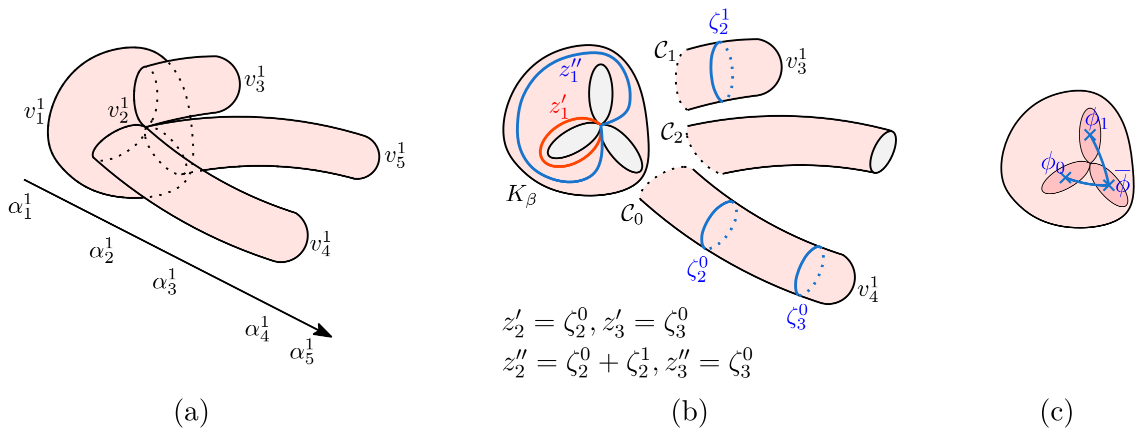


Figure 3.6. (a) A complex K with all 1st critical vertices listed, in which v_2^1 is a monkey saddle; the direction of the height function is indicated. (b) The relevant subcomplex $\mathbb{K}^p_{(b-1,d]} = \mathbb{K}^1_{(1,4]}$ with K_β broken from the remaining parts for a better illustration. (c) The complex K_β with boundaries filled by 2-dimensional "cells" drawn as darker regions. The blue edges are augmenting edges. Note that K_β also contains boundary 1-simplices around the critical vertex v_1^1, which are not drawn.

Now consider the example in Figure 3.6, and let z_1, z_2, z_3 be a general sequence of persistent 1-cycles for $\left[\alpha_2^1, \alpha_4^1\right)$. By definition, there exist 2-chains

$$A_2 \subseteq \mathbb{K}^1_{(1,3)},\ A_3 \subseteq \mathbb{K}^1_{(2,4)},\ \text{and } A_4 \subseteq \mathbb{K}^1_{(3,4]}$$

such that

$$z_1 + z_2 = \partial(A_2),\ z_2 + z_3 = \partial(A_3),\ \text{and } z_3 = \partial(A_4)$$

Letting $A = A_2 + A_3 + A_4$, we have $\partial(A) = z_1 \subseteq K_\beta$. (We still assume that $\left[\alpha_2^1, \alpha_4^1\right)$ is produced by a simplex-wise interval denoted $[K_\beta, K_\delta]$.) One strategy we adopt is to separate K_β from $\mathbb{K}^p_{(b-1,d]}$ and tackle K_β, $\mathbb{K}^p_{(b-1,d]} \setminus K_\beta$ independently. Note that $\mathbb{K}^p_{(b-1,d]} = \mathbb{K}^1_{(1,4]}$ in our example. Then we separate A into the part that is in K_β and the part that is not. Obviously, the part of A *not* in K_β comes from different 2-connected components of $\mathbb{K}^1_{(1,4]} \setminus K_\beta$, which are $\mathcal{C}_0$, $\mathcal{C}_1$, and $\mathcal{C}_2$ shown in Figure 3.6b. We then observe that any such component intersecting A must be totally included in A, because a 2-simplex of the component not in A would cause

$\partial(A)$ to contain 1-simplices not in K_β, contradicting $\partial(A) \subseteq K_\beta$. For the same reason, any component intersecting A must have its boundary in K_β. For example, in Figure 3.6b, no 2-simplices in $\mathcal{C}_2$ can fall in A, while $\mathcal{C}_1$ can either be totally in or disjoint with A. The proof of Proposition 3.3.7 formally justifies this observation. We also note that there can be only one 2-connected component of $\mathbb{K}^1_{(1,4]} \setminus K_\beta$ (i.e., $\mathcal{C}_0$ in Figure 3.6b) whose boundary resides in K_β and contains $\sigma_{\beta-1}$ (see Proposition 3.3.5). (While this is not drawn in Figure 3.6, we assume that K is triangulated in a way that $\sigma_{\beta-1}$ is shared by the boundaries of $\mathcal{C}_0$ and $\mathcal{C}_2$.) A fact about $\mathcal{C}_0$ is that it is always included in A (see the proof of Proposition 3.3.7). For the other components with boundaries in K_β (i.e., $\mathcal{C}_1$ in Figure 3.6b), any subset of them can contribute to a certain A and take part in forming the persistent cycles. For example, in Figure 3.6b, only $\mathcal{C}_0$ contributes to the persistent 1-cycles z'_1, z'_2, z'_3, and both $\mathcal{C}_0, \mathcal{C}_1$ contribute to z''_1, z''_2, z''_3.

The crux of the algorithm, therefore, is to determine a subset of the components along with $\mathcal{C}_0$ contributing to the *optimal* persistent cycles (a complicated monkey saddle with multiple forks may result in many such components), because we can compute the optimal persistent cycles under a fixed choice of subset. To see this, suppose that z''_1, z''_2, z''_3 in Figure 3.6 are the optimal persistent 1-cycles for $\left[\alpha^1_2, \alpha^1_4\right)$ under the choice of the subset $\left\{\mathcal{C}_0, \mathcal{C}_1\right\}$, i.e, z''_1, z''_2, z''_3 have the minimum sum of weight among all persistent 1-cycles coming from *both* $\mathcal{C}_0$ and $\mathcal{C}_1$. We first observe that z''_1 must be the minimum 1-cycle homologous to $\partial(\mathcal{C}_0) + \partial(\mathcal{C}_1)$ in K_β. Such a cycle z''_1 can be computed from a minimum (s,t)-cut on a dual graph of K_β. Also, the set of 1-cycles $\left\{\zeta^0_2 \subseteq \mathbb{K}^1_{(2,3)}, \zeta^0_3 \subseteq \mathbb{K}^1_{(3,4)}\right\}$ must be the one in $\mathcal{C}_0$ with the minimum sum of weight such that

$$\zeta^0_2 \sim \partial(\mathcal{C}_0) \text{ in } \mathbb{K}^1_{(1,3)},\ \zeta^0_2 \sim \zeta^0_3 \text{ in } \mathbb{K}^1_{(2,4)}, \text{ and } \zeta^0_3 \text{ null-homologous in } \mathbb{K}^1_{(3,4]}$$

Additionally, $\zeta^1_2 \subseteq \mathbb{K}^1_{(2,3)}$ must be the minimum 1-cycle in $\mathcal{C}_1$ which is homologous to $\partial(\mathcal{C}_1)$ in $\mathbb{K}^1_{(1,3)}$ and is null-homologous in $\mathbb{K}^1_{(2,3]}$. (See Step 2 of Algorithm 3.3.2 for a formal description.) To compute the minimum cycles $\left\{\zeta^0_2, \zeta^0_3\right\}$, ζ^1_2, we utilize an algorithm similar to Algorithm 3.3.1.

Note that a priori optimal selection of the components is not obvious; while introducing more components increases weights for cycles in the p-th regular complexes (because the components are disjoint), the cycle in K_β corresponding to this choice may have a smaller weight due to belonging to a different homology class (e.g. $z_1'' \sim \partial(\mathcal{C}_0) + \partial(\mathcal{C}_1)$ may have much smaller weight than $z_1' \sim \partial(\mathcal{C}_0)$ in Figure 3.6b).

Our solution is as follows: generically, suppose that $\mathcal{C}_0, \dots, \mathcal{C}_k$ are all the $(p+1)$-connected components of $\mathbb{K}^p_{(b-1,d]} \setminus K_\beta$ with boundaries in K_β, where $\mathcal{C}_0$ is the one whose boundary contains $\sigma_{\beta-1}$. We do the following:

1. For each $\mathrm{j} \in [0, k]$, compute the minimum (possibly empty) p-cycles $\left\{\zeta^{\mathrm{j}}_{\mathrm{i}} \mid b \leq \mathrm{i} < d\right\}$ in $\mathcal{C}_{\mathrm{j}}$ as described in Step 2 of Algorithm 3.3.2. Note that for $\mathcal{C}_1$ in Figure 3.6b, ζ^1_3 is empty, which makes ζ^1_2 null-homologous in $\mathbb{K}^1_{(2,3]}$.

2. Build a dual graph G for K_β, where dummy vertices $\phi_0, \dots, \phi_k$ correspond to the boundaries $\partial(\mathcal{C}_0), \dots, \partial(\mathcal{C}_k)$ and a single dummy vertex $\overline{\phi}$ corresponds to the remaining boundary portion of K_β. Roughly speaking, when a dummy vertex ϕ_{j} is said to "correspond to" $\partial(\mathcal{C}_{\mathrm{j}})$, one can imagine that a $(p+1)$-dimensional "cell" with boundary $\partial(\mathcal{C}_{\mathrm{j}})$ is added to K_β and ϕ_{j} is the vertex dual to this cell. In addition to the regular dual edges, for each ϕ_{j}, we add to G an *augmenting edge* connecting ϕ_{j} to $\overline{\phi}$ and let its weight be $\sum_{\mathrm{i}=b}^{d-1} w\left(\zeta^{\mathrm{j}}_{\mathrm{i}}\right)$. See Figure 3.6c for an example of the dummy vertices and augmenting edges.

3. Compute the minimum (s,t)-cut (S^*, T^*) of $\left(G, \phi_0, \overline{\phi}\right)$, which produces an optimal sequence of levelset persistent p-cycles for $\left[\alpha^p_b, \alpha^p_d\right)$.

To see the correctness of the algorithm, consider a general (s,t)-cut (S,T) of $\left(G, \phi_0, \overline{\phi}\right)$. Whenever a ϕ_{j} is in S, it means that the component $\mathcal{C}_{\mathrm{j}}$ is chosen to form the persistent cycles. Also, since the augmenting edge $\left\{\phi_{\mathrm{j}}, \overline{\phi}\right\}$ is crossing the cut, its weight records the cost of introducing $\mathcal{C}_{\mathrm{j}}$ in forming the persistent cycles. Let $\phi_{\nu_0}, \dots, \phi_{\nu_\ell}$ be all the dummy vertices in S. We have that the non-augmenting edges in $E(S,T)$ produce a dual p-cycle z_{b-1} in K_β homologous to $\partial(\mathcal{C}_{\nu_0}) + \cdots + \partial(\mathcal{C}_{\nu_\ell})$. Then, the p-cycle z_{b-1}, along with all $\left\{\zeta^{\nu_{\mathrm{j}}}_{\mathrm{i}} \mid b \leq \mathrm{i} < d\right\}$

from $\mathcal{C}_{\nu_0}, \ldots, \mathcal{C}_{\nu_\ell}$, form a sequence of persistent p-cycles for $\left[\alpha_b^p, \alpha_d^p\right)$ whose sum of weight equals $w(S,T)$. Later in the subsection, we formally justify the algorithm.

Pseudocode.

We now provide the full details of our algorithm. For ease of exposition, so far we have let $\mathbb{K}^p_{(b-1,d]}$ be the complex on which we compute the optimal persistent cycles. However, this has a flaw, which can be illustrated by the example in Figure 3.6. Imagine that v_4^1 and v_5^1 in the figure are pinched together, so that K is not a 2-manifold anymore (but still a weak 2-pseudomanifold). The simplex-wise filtration $\mathcal{F}_p(f)$ can be constructed in a way that the disc around v_4^1 is formed before the disc around v_5^1; such an $\mathcal{F}_p(f)$ is essentially the same as the one before pinching. However, $\mathbb{K}^p_{(b-1,d]} = \mathbb{K}^1_{(1,4]}$ now contains both v_4^1, v_5^1. As a consequence, Proposition 3.3.5 which is a major observation for our solution does not hold because the component containing v_5^1 (which is $\mathcal{C}_2$ in Figure 3.6b with the right hole filled) also has its boundary containing $\sigma_{\beta-1}$. To solve this, we make an adjustment to work on a complex $\tilde{K}$ instead of $\mathbb{K}^p_{(b-1,d]}$, so that Proposition 3.3.5 is still true; see Step 1 of Algorithm 3.3.2 for the definition of $\tilde{K}$. It can be easily verified that each complex in Sequence (3.3) $\big($possibly excluding $\mathbb{K}^p_{(d-1,d]}$ which is indeed irrelevant$\big)$ is a subcomplex of $\tilde{K}$.

Our previous exposition also frequently deals with the complex K_β. However, in the pseudocode (Algorithm 3.3.2), K_β takes another form: we add to K_β some missing $(p+1)$-simplices and denote the new complex as $\overline{K}_\beta$; see Step 1 of the pseudocode for definition. Doing this makes our description of the components in Step 2 neater.

Algorithm 3.3.2. *Given the input as specified, the algorithm does the following:*

1. *Set the following:*
    - $\tilde{K} = \mathbb{K}^p_{(b-1,d)} \cup K_{\delta+1}$
    - $\overline{K}_\beta = K_\beta \cup \Big\{(p+1)\text{-simplices with all } p\text{-faces in } K_\beta\Big\}$
2. *Let $\mathcal{C}_0, \ldots, \mathcal{C}_k$ be all the $(p+1)$-connected components of $\tilde{K} \setminus \overline{K}_\beta$ such that $\partial(\mathcal{C}_\mathrm{j}) \subseteq \overline{K}_\beta$ for each j, where $\mathcal{C}_0$ is the unique one whose boundary contains $\sigma_{\beta-1}$. (Note that the* boundary *$\partial(\mathcal{C}_\mathrm{j})$ here means the boundary of the $(p+1)$-chain $\mathcal{C}_\mathrm{j}$.)*

For each $\mathcal{C}_{\mathrm{j}}$, let M_{j} be the closure of $\mathcal{C}_{\mathrm{j}}$, and among all sets of p-cycles of the form

$$\left\{z_{\mathrm{i}} \subseteq M_{\mathrm{j}} \cap \mathbb{K}^{p}_{(\mathrm{i},\mathrm{i}+1)} \mid b \leq \mathrm{i} < d\right\}$$

such that

- $z_b \sim \partial(\mathcal{C}_{\mathrm{j}})$ *in* $M_{\mathrm{j}} \cap \mathbb{K}^{p}_{(b-1,b+1)}$
- $z_{\mathrm{i}-1} \sim z_{\mathrm{i}}$ *in* $M_{\mathrm{j}} \cap \mathbb{K}^{p}_{(\mathrm{i}-1,\mathrm{i}+1)}$ *for each* $b < \mathrm{i} < d$
- z_{d-1} *is null-homologous in* $M_{\mathrm{j}} \cap K_{\delta+1}$

compute the set $\left\{\zeta^{\mathrm{j}}_{\mathrm{i}} \mid b \leq \mathrm{i} < d\right\}$ *with the minimum sum of weight.*

3. *Build a weighted dual graph G from $\overline{K}_{\beta}$ as follows:*

   Let each $(p+1)$-simplex of $\overline{K}_{\beta}$ correspond to a vertex in G, and add the dummy vertices $\overline{\phi}, \phi_0, \ldots, \phi_k$ to G. Let θ denote the bijection from the $(p+1)$-simplices to $V(G) \setminus \left\{\overline{\phi}, \phi_0, \ldots, \phi_k\right\}$.

   Let each p-simplex σ of $\overline{K}_{\beta}$ correspond to an edge e *in G, where the weight of* e*, $w(\mathrm{e})$, equals the weight of σ. There are the following cases:*

   σ has two $(p+1)$-cofaces in $\overline{K}_{\beta}$: e *is the usual one.*

   σ has one $(p+1)$-coface τ in $\overline{K}_{\beta}$: *If $\sigma \in \partial(\mathcal{C}_{\mathrm{j}})$ for a $\mathcal{C}_{\mathrm{j}}$, let* e *connect $\theta(\tau)$ and ϕ_{j} in G; otherwise, let* e *connect $\theta(\tau)$ and $\overline{\phi}$.*

   σ has no $(p+1)$-cofaces in $\overline{K}_{\beta}$: *If σ is in the boundaries of two components $\mathcal{C}_{\mathrm{i}}$ and $\mathcal{C}_{\mathrm{j}}$, let* e *connect ϕ_{i} and ϕ_{j}; if σ is in the boundary of only one component $\mathcal{C}_{\mathrm{j}}$, let* e *connect ϕ_{j} and $\overline{\phi}$; otherwise, let* e *connect $\overline{\phi}$ on both ends.*

   In addition to the above edges, add the augmenting edges with weights as described. Let θ also denote the bijection from the p-simplices to the non-augmenting edges and let $E'(S,T)$ denote the set of non-augmenting edges crossing a cut (S,T).

4. *Compute the minimum (s,t)-cut (S^*, T^*) of $\left(G, \phi_0, \overline{\phi}\right)$. Let $\phi_{\mu_0}, \ldots, \phi_{\mu_l}$ be all the dummy vertices in S^*. Then, set*

   $z^*_{b-1} = \theta^{-1}(E'(S^*, T^*))$ *and* $z^*_{\mathrm{i}} = \sum_{\mathrm{j}=0}^{l} \zeta^{\mu_{\mathrm{j}}}_{\mathrm{i}}$ *for each* $b \leq \mathrm{i} < d$

Return $z^*_{b-1}, z^*_b, \ldots, z^*_{d-1}$ as an optimal sequence of levelset persistent p-cycles for $\left[\alpha^p_b, \alpha^p_d\right)$.

As mentioned, the minimum cycles in Step 2 can be computed using a similar approach of Algorithm 3.3.1, with a difference that Algorithm 3.3.1 works on a complex "closed at both ends" while M_j is "closed only at the right". Therefore, we need to add a dummy vertex to the dual graph for the boundary, which is put into the source. Note that we can build a single dual graph for all the M_j's and share the dummy vertex, so that we only need to invoke one minimum cut computation.

Justification.

We now prove the correctness of Algorithm 3.3.2. We first present the following proposition stating a basic fact about $\sigma_{\beta-1}$:

Proposition 3.3.4. *The p-simplex $\sigma_{\beta-1}$ has no $(p+1)$-cofaces in $\overline{K}_\beta$.*

Proof. Supposing instead that $\sigma_{\beta-1}$ has a $(p+1)$-coface τ in $\overline{K}_\beta$, then $\partial(\tau) \subseteq K_\beta$. Since $\overline{K}_\beta \subseteq \mathbb{K}^p_{(b-1,b]}$, the p-cycle $\partial(\tau)$ created by $\sigma_{\beta-1}$ is a boundary in $\mathbb{K}^p_{(b-1,b]}$. Simulating a run of Algorithm A.1.1 (presented in Appendix A.1) with input $\mathcal{F}_p(f)$, at the $(\beta-1)$-th iteration, we can let $\partial(\tau)$ be the representative p-cycle at index β for the new interval $[\beta, \beta]$. However, since $\partial(\tau)$ is a boundary in $\mathbb{K}^p_{(b-1,b]}$, the interval starting with β must end with an index less than δ, which is a contradiction. □

Proposition 3.3.5 justifies the operations in Step 2:

Proposition 3.3.5. *Among all the $(p+1)$-connected components of $\tilde{K} \setminus \overline{K}_\beta$, there is one and only one component whose boundary resides in $\overline{K}_\beta$ and contains $\sigma_{\beta-1}$.*

Proof. See Appendix A.2.4. □

Finally, Proposition 3.3.6 and 3.3.7 lead to Theorem 3.3.2, which is the conclusion.

Proposition 3.3.6. *For any (s,t)-cut (S,T) of $\left(G, \phi_0, \overline{\phi}\right)$, let $\phi_{\nu_0}, \ldots, \phi_{\nu_\ell}$ be all the dummy vertices in S. Furthermore, let $z_{b-1} = \theta^{-1}(E'(S,T))$ and $z_\mathrm{i} = \sum_{\mathrm{j}=0}^{\ell} \zeta_\mathrm{i}^{\nu_\mathrm{j}}$ for each $b \leq \mathrm{i} < d$. Then, $z_{b-1}, z_b, \ldots, z_{d-1}$ is a sequence of levelset persistent p-cycles for $\left[\alpha^p_b, \alpha^p_d\right)$ with $\sum_{\mathrm{i}=b-1}^{d-1} w(z_\mathrm{i}) = w(S,T)$.*

Proof. Note that we can also consider (S,T) as an (s,t)-cut of a graph derived by deleting the augmenting edges from G where the sources are $\phi_{\nu_0},\dots,\phi_{\nu_\ell}$ and the sinks are all the other dummy vertices. This implies that $z_{b-1}=\theta^{-1}(E'(S,T))$ is homologous to $\partial(\mathcal{C}_{\nu_0}+\cdots+\mathcal{C}_{\nu_\ell})$ in $\overline{K}_\beta$. Since ϕ_0 is the source of G, ϕ_0 must be one of $\phi_{\nu_0},\dots,\phi_{\nu_\ell}$. Then, by Proposition 3.3.5, $\partial(\mathcal{C}_{\nu_0}+\cdots+\mathcal{C}_{\nu_\ell})$ contains $\sigma_{\beta-1}$. So z_{b-1} must also contain $\sigma_{\beta-1}$ because $z_{b-1}\sim\partial(\mathcal{C}_{\nu_0}+\cdots+\mathcal{C}_{\nu_\ell})$ in $\overline{K}_\beta$ and $\sigma_{\beta-1}$ has no $(p+1)$-coface in $\overline{K}_\beta$ (Proposition 3.3.4). Furthermore, the properties of the cycles $\left\{\zeta_{\mathrm{i}}^{\mathrm{j}}\right\}$ computed in Step 2 of Algorithm 3.3.2 imply that $z_b=\zeta_b^{\nu_0}+\cdots+\zeta_b^{\nu_\ell}$ is homologous to $\partial(\mathcal{C}_{\nu_0}+\cdots+\mathcal{C}_{\nu_\ell})$ in $\mathbb{K}^p_{(b-1,b+1)}$. So $z_{b-1}\sim z_b$ in $\mathbb{K}^p_{(b-1,b+1)}$.

For $z_{b-1},z_b,\dots,z_{d-1}$ to be persistent p-cycles for $\left[\alpha_b^p,\alpha_d^p\right)$, we need to verify several other conditions in Definition 3.2.5, in which only one is non-trivial, i.e., the condition that $[z_{d-1}]\in\mathsf{H}_p(K_\delta)$ is the non-zero class in $\mathsf{ker}(\varphi_\delta)$. To see this, we first note that obviously $[z_{d-1}]\in\mathsf{ker}(\varphi_\delta)$. To prove $[z_{d-1}]\neq0$, we use a similar approach in the proof of Proposition 3.3.1, i.e., simulate a run of Algorithm A.1.1 for computing $\mathsf{Pers}_p(\mathcal{F}_p(f))$ and show that $z_{d-1}\subseteq K_\delta$ can be the representative cycle at index δ for the interval $[\beta,\delta]$. The details are omitted.

For the weight, we have

$$\begin{aligned} w(S,T) &= \sum_{\mathrm{e}\in E'(S,T)} w(\mathrm{e}) + \sum_{\mathrm{j}=0}^{\ell} w\left(\left\{\phi_{\nu_\mathrm{j}},\overline{\phi}\right\}\right) = w(z_{b-1}) + \sum_{\mathrm{j}=0}^{\ell}\sum_{\mathrm{i}=b}^{d-1} w\left(\zeta_{\mathrm{i}}^{\nu_\mathrm{j}}\right) \\ &= w(z_{b-1}) + \sum_{\mathrm{i}=b}^{d-1}\sum_{\mathrm{j}=0}^{\ell} w\left(\zeta_{\mathrm{i}}^{\nu_\mathrm{j}}\right) = \sum_{\mathrm{i}=b-1}^{d-1} w(z_\mathrm{i}) \end{aligned}$$

where $\left\{\phi_{\nu_\mathrm{j}},\overline{\phi}\right\}$ denotes the augmenting edge in G connecting ϕ_{ν_j} and $\overline{\phi}$. □

Proposition 3.3.7. *Let $z_{b-1},z_b,\dots,z_{d-1}$ be any sequence of levelset persistent p-cycles for $\left[\alpha_b^p,\alpha_d^p\right)$; then, there exists an (s,t)-cut (S,T) of $\left(G,\phi_0,\overline{\phi}\right)$ with $w(S,T)\leq\sum_{\mathrm{i}=b-1}^{d-1}w(z_\mathrm{i})$.*

Proof. By definition, there exist $(p+1)$-chains $A_b\subseteq\mathbb{K}^p_{(b-1,b+1)},\dots,A_{d-1}\subseteq\mathbb{K}^p_{(d-2,d)},A_d\subseteq K_{\delta+1}$ such that $z_{b-1}+z_b=\partial\left(A_b\right),\dots,z_{d-2}+z_{d-1}=\partial\left(A_{d-1}\right),z_{d-1}=\partial\left(A_d\right)$. Let $A=\sum_{\mathrm{i}=b}^{d}A_\mathrm{i}$; then, $\partial(A)=z_{b-1}$. Let $\mathcal{C}_{\nu_0},\dots,\mathcal{C}_{\nu_\ell}$ be all the components defined in Step 2 of Algorithm 3.3.2 which intersect A. We claim that each $\mathcal{C}_{\nu_\mathrm{j}}\subseteq A$. For contradiction, suppose instead that there is a $\sigma\in\mathcal{C}_{\nu_\mathrm{j}}$ not in A. Let σ' be a simplex in $A\cap\mathcal{C}_{\nu_\mathrm{j}}$. Since σ, σ' are both in $\mathcal{C}_{\nu_\mathrm{j}}$, there must be a $(p+1)$-path $\tau_1,\dots,\tau_q$ from σ to σ' in $\tilde{K}\setminus\overline{K}_\beta$. Note that $\sigma\notin A$ and $\sigma'\in A$, and so there is an ι such that $\tau_\iota \notin A$ and $\tau_{\iota+1} \in A$. Let τ^p be a p-face shared by τ_ι and $\tau_{\iota+1}$ in $\tilde{K} \setminus \overline{K}_\beta$; then, $\tau^p \in \partial(A)$ and $\tau^p \notin \overline{K}_\beta$. This contradicts $\partial(A) = z_{b-1} \subseteq \overline{K}_\beta$. So $\mathcal{C}_{\nu_\mathrm{j}} \subseteq A$. We also note that $\mathcal{C}_{\nu_0}, \ldots, \mathcal{C}_{\nu_\ell}$ are all the $(p+1)$-connected components of $\tilde{K} \setminus \overline{K}_\beta$ intersecting A. The reason is that, if $\widehat{\mathcal{C}}$ is a component intersecting A whose boundary is not completely in $\overline{K}_\beta$, then we also have $\widehat{\mathcal{C}} \subseteq A$ and the justification is similar as above. Let σ be a simplex in $\partial\big(\widehat{\mathcal{C}}\big)$ but not $\overline{K}_\beta$; then, $\sigma \in \partial(A)$. To see this, suppose instead that $\sigma \notin \partial(A)$. Then σ has a $(p+1)$-coface $\tau_1 \in \widehat{\mathcal{C}} \subseteq A$ and a $(p+1)$-coface $\tau_2 \in A \setminus \widehat{\mathcal{C}}$. We have $\tau_2 \in \overline{K}_\beta$ because if not, combining the fact that $\sigma, \tau_1, \tau_2 \in \tilde{K} \setminus \overline{K}_\beta$ and $\tau_1 \in \widehat{\mathcal{C}}$, τ_2 would be in $\widehat{\mathcal{C}}$. As a face of τ_2, σ must also be in $\overline{K}_\beta$, which is a contradiction. So we have $\sigma \in \partial(A)$. Note that $\sigma \notin \overline{K}_\beta$, which contradicts $\partial(A) \subseteq \overline{K}_\beta$, and hence such a $\widehat{\mathcal{C}}$ cannot exist. We then have $\partial\Big(A \setminus \bigcup_{\mathrm{j}=0}^{\ell} \mathcal{C}_{\nu_\mathrm{j}}\Big) = \partial\Big(A + \mathcal{C}_{\nu_0} + \cdots + \mathcal{C}_{\nu_\ell}\Big) = z_{b-1} + \partial\Big(\mathcal{C}_{\nu_0}\Big) + \cdots + \partial\Big(\mathcal{C}_{\nu_\ell}\Big)$, where $A \setminus \bigcup_{\mathrm{j}=0}^{\ell} \mathcal{C}_{\nu_\mathrm{j}} \subseteq \overline{K}_\beta$. Now $\partial\Big(\mathcal{C}_{\nu_0}\Big) + \cdots + \partial\Big(\mathcal{C}_{\nu_\ell}\Big)$ is homologous to z_{b-1} in $\overline{K}_\beta$, which means that it must contain $\sigma_{\beta-1}$ because z_{b-1} contains $\sigma_{\beta-1}$ and $\sigma_{\beta-1}$ has no $(p+1)$-coface in $\overline{K}_\beta$ (Proposition 3.3.4). This implies that $\Big\{\mathcal{C}_{\nu_0}, \ldots, \mathcal{C}_{\nu_\ell}\Big\}$ contains $\mathcal{C}_0$ by Proposition 3.3.5. Let $S = \theta\Big(A \setminus \bigcup_{\mathrm{j}=0}^{\ell} \mathcal{C}_{\nu_\mathrm{j}}\Big) \cup \Big\{\phi_{\nu_0}, \ldots, \phi_{\nu_\ell}\Big\}$ and $T = V(G) \setminus S$. It can be verified that (S,T) is an (s,t)-cut of $\Big(G, \phi_0, \overline{\phi}\Big)$ and $z_{b-1} = \theta^{-1}(E'(S,T))$.

We then prove that $w(S,T) \le \sum_{\mathrm{i}=b-1}^{d-1} w(z_\mathrm{i})$. Let $A_\mathrm{i}^{\nu_\mathrm{j}} = M_{\nu_\mathrm{j}} \cap A_\mathrm{i}$, $z_\mathrm{i}^{\nu_\mathrm{j}} = M_{\nu_\mathrm{j}} \cap z_\mathrm{i}$ for each i and j. For any j, we claim the following

$$\partial\left(\sum_{\mathrm{i}=b+1}^{d} A_\mathrm{i}^{\nu_\mathrm{j}}\right) = z_b^{\nu_\mathrm{j}} \tag{3.4}$$

To prove Equation (3.4), we first note the following

$$\partial\left(\sum_{\mathrm{i}=b+1}^{d} A_\mathrm{i}^{\nu_\mathrm{j}}\right) = \partial\left(M_{\nu_\mathrm{j}} \cap \sum_{\mathrm{i}=b+1}^{d} A_\mathrm{i}\right),\ z_b^{\nu_\mathrm{j}} = M_{\nu_\mathrm{j}} \cap z_b = M_{\nu_\mathrm{j}} \cap \partial\left(\sum_{\mathrm{i}=b+1}^{d} A_\mathrm{i}\right)$$

So we only need to show that $\partial\Big(M_{\nu_\mathrm{j}} \cap \sum_{\mathrm{i}=b+1}^{d} A_\mathrm{i}\Big) = M_{\nu_\mathrm{j}} \cap \partial\Big(\sum_{\mathrm{i}=b+1}^{d} A_\mathrm{i}\Big)$. Letting $B = \sum_{\mathrm{i}=b+1}^{d} A_\mathrm{i}$, what we need to prove now becomes $\partial(M_{\nu_\mathrm{j}} \cap B) = M_{\nu_\mathrm{j}} \cap \partial(B)$. Consider an arbitrary $\sigma \in \partial(M_{\nu_\mathrm{j}} \cap B)$. We have that σ is a face of only one $(p+1)$-simplex $\tau \in M_{\nu_\mathrm{j}} \cap B$. Note that $\tau \in B$, and we show that τ is the only $(p+1)$-coface of σ in B. Suppose instead that σ has another $(p+1)$-coface τ' in B. Then, $\tau' \notin M_{\nu_\mathrm{j}}$ because $\tau' \notin M_{\nu_\mathrm{j}} \cap B$. Note that

$B \subseteq \mathbb{K}^p_{(b,d]}$, which means that B is disjoint with $\overline{K}_\beta \subseteq \mathbb{K}^p_{(b-1,b]}$. So $\tau' \in B \subseteq \tilde{K} \setminus \overline{K}_\beta$. It is then true that $\sigma \in \overline{K}_\beta$ because if not, i.e., $\sigma \in \tilde{K} \setminus \overline{K}_\beta$, then τ' would reside in $\mathcal{C}_{\nu_\mathrm{j}} \subseteq M_{\nu_\mathrm{j}}$ (following from $\tau \in \mathcal{C}_{\nu_\mathrm{j}}$). We now have $\tau \in B \subseteq \mathbb{K}^p_{(b,d]}$ and $\sigma \in \overline{K}_\beta \subseteq \mathbb{K}^p_{(b-1,b]}$, which implies that $\sigma \cap \tau = \varnothing$, contradicting $\sigma \subseteq \tau$. Therefore, $\sigma \in \partial(B)$. Since $\tau \in M_{\nu_\mathrm{j}}$, we have $\sigma \in M_{\nu_\mathrm{j}}$, and so $\sigma \in M_{\nu_\mathrm{j}} \cap \partial(B)$. On the other hand, let σ be any p-simplex in $M_{\nu_\mathrm{j}} \cap \partial(B)$. Since $\sigma \in \partial(B)$, σ is a face of only one $(p+1)$-simplex τ in B. We then prove that $\tau \in M_{\nu_\mathrm{j}}$. Suppose instead that $\tau \notin M_{\nu_\mathrm{j}}$. Then, since $\sigma \in M_{\nu_\mathrm{j}}$, σ must be a face of $(p+1)$-simplex $\tau' \in M_{\nu_\mathrm{j}}$. It follows that $\sigma \in \overline{K}_\beta$, because if not, τ and τ' would both be in M_{ν_j}. We then reach the contradiction that $\sigma \cap \tau = \varnothing$ because $\tau \in B \subseteq \mathbb{K}^p_{(b,d]}$ and $\sigma \in \overline{K}_\beta \subseteq \mathbb{K}^p_{(b-1,b]}$. Therefore, σ is a face of only one $(p+1)$-simplex τ in $M_{\nu_\mathrm{j}} \cap B$, which means that $\sigma \in \partial(M_{\nu_\mathrm{j}} \cap B)$.

Note that $\sum_{\mathrm{i}=b}^{d} A^{\nu_\mathrm{j}}_\mathrm{i} = M_{\nu_\mathrm{j}} \cap A = \mathcal{C}_{\nu_\mathrm{j}}$ because $\mathcal{C}_{\nu_\mathrm{j}} \subseteq A$. Hence, by Equation (3.4)

$$z^{\nu_\mathrm{j}}_b = \partial\left(\sum_{\mathrm{i}=b+1}^{d} A^{\nu_\mathrm{j}}_\mathrm{i}\right) = \partial\left(\sum_{\mathrm{i}=b}^{d} A^{\nu_\mathrm{j}}_\mathrm{i}\right) + \partial\Big(A^{\nu_\mathrm{j}}_b\Big) = \partial\Big(\mathcal{C}_{\nu_\mathrm{j}}\Big) + \partial\Big(A^{\nu_\mathrm{j}}_b\Big)$$

Now we have $z^{\nu_\mathrm{j}}_b + \partial\Big(\mathcal{C}_{\nu_\mathrm{j}}\Big) = \partial\Big(A^{\nu_\mathrm{j}}_b\Big)$, i.e., $z^{\nu_\mathrm{j}}_b \sim \partial\Big(\mathcal{C}_{\nu_\mathrm{j}}\Big)$ in $M_{\nu_\mathrm{j}} \cap \mathbb{K}^p_{(b-1,b+1)}$. Similar to Equation (3.4), for each i s.t. $b < \mathrm{i} < d$, we have $\partial\Big(\sum_{\eta=\mathrm{i}}^{d} A^{\nu_\mathrm{j}}_\eta\Big) = z^{\nu_\mathrm{j}}_{\mathrm{i}-1}$ and $\partial\Big(\sum_{\eta=\mathrm{i}+1}^{d} A^{\nu_\mathrm{j}}_\eta\Big) = z^{\nu_\mathrm{j}}_\mathrm{i}$. Therefore, $\partial\Big(A^{\nu_\mathrm{j}}_\mathrm{i}\Big) = z^{\nu_\mathrm{j}}_{\mathrm{i}-1} + z^{\nu_\mathrm{j}}_\mathrm{i}$, i.e., $z^{\nu_\mathrm{j}}_{\mathrm{i}-1} \sim z^{\nu_\mathrm{j}}_\mathrm{i}$ in $M_{\nu_\mathrm{j}} \cap \mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i}+1)}$. We also have that $\partial\Big(\sum_{\eta=d}^{d} A^{\nu_\mathrm{j}}_\eta\Big) = z^{\nu_\mathrm{j}}_{d-1}$, i.e., $z^{\nu_\mathrm{j}}_{d-1}$ is null homologous in $M_{\nu_\mathrm{j}} \cap K_{\delta+1}$. So $\Big\{z^{\nu_\mathrm{j}}_\mathrm{i} \mid b \le \mathrm{i} < d\Big\}$ is a set of p-cycles satisfying the condition specified in Step 2 of Algorithm 3.3.2, which means that $\sum_{\mathrm{i}=b}^{d-1} w(\zeta^{\nu_\mathrm{j}}_\mathrm{i}) \le \sum_{\mathrm{i}=b}^{d-1} w(z^{\nu_\mathrm{j}}_\mathrm{i})$.

Finally, we have

$$\begin{aligned} w(S,T) &= \sum_{\mathrm{e} \in E'(S,T)} w(\mathrm{e}) + \sum_{\mathrm{j}=0}^{\ell} w\Big(\Big\{\phi_{\nu_\mathrm{j}}, \overline{\phi}\Big\}\Big) = w(z_{b-1}) + \sum_{\mathrm{j}=0}^{\ell} \sum_{\mathrm{i}=b}^{d-1} w\Big(\zeta^{\nu_\mathrm{j}}_\mathrm{i}\Big) \\ &\le w(z_{b-1}) + \sum_{\mathrm{i}=b}^{d-1} \sum_{\mathrm{j}=0}^{\ell} w\Big(z^{\nu_\mathrm{j}}_\mathrm{i}\Big) = \sum_{\mathrm{i}=b-1}^{d-1} w(z_\mathrm{i}) \end{aligned}$$

where $\Big\{\phi_{\nu_\mathrm{j}}, \overline{\phi}\Big\}$ denotes the augmenting edge in G connecting ϕ_{ν_j} and $\overline{\phi}$. □

Theorem 3.3.2. *Algorithm 3.3.2 computes an optimal sequence of level persistent p-cycles for a given* closed-open *interval.*

3.3.3 Closed-closed case

In the subsection, we describe the computation of the optimal persistent p-cycles for a *closed-closed* interval $\left[\alpha_b^p, \alpha_d^p\right]$ from $\mathsf{Pers}_p(\mathcal{L}_p(f))$, which is produced by a simplex-wise interval $[K_\beta, K_\delta]$ from $\mathsf{Pers}_p(\mathcal{F}_p(f))$. Due to the similarity to the closed-open case, we only describe the algorithm briefly. Figure 3.7 provides examples for $p = 1$, in which different sequences of persistent 1-cycles are formed for the interval $\left[\alpha_3^1, \alpha_5^1\right]$, and two of them are $z_2^1 + z_2^3, z_3^1 + z_3^3, z_4^1 + z_4^3, z_5^1 + z_5^3$ and $z_2^0, z_3^0, z_4^0 + z_4^2, z_5^0 + z_5^2$.

Similar to the previous cases, we have the following relevant portion of $\mathcal{F}_p(f)$:

$$\begin{tikzcd}
\mathbb{K}^p_{(b-1,b)} \arrow[r, hook, dashed] & K_{\beta-1} \arrow[r, hook, "\sigma_{\beta-1}"] & K_\beta \arrow[r, hook, dashed] & \mathbb{K}^p_{(b-1,b+1)} & \mathbb{K}^p_{(b,b+1)} \arrow[l, hook', dashed] \arrow[r, hook, dashed] & \cdots \\
\cdots & \mathbb{K}^p_{(d-1,d)} \arrow[l, hook', dashed] \arrow[r, hook, dashed] & \mathbb{K}^p_{(d-1,d+1)} & K_\delta \arrow[l, hook', dashed] & K_{\delta+1} \arrow[l, hook', "\sigma_\delta"'] & \mathbb{K}^p_{(d,d+1)} \arrow[l, hook', dashed]
\end{tikzcd} \tag{3.5}$$

Note that the creator $\sigma_{\beta-1}$ and the destroyer σ_δ are both p-simplices [27], and the computation can be restricted to the subcomplex $\mathbb{K}^p_{(b-1,d+1)}$. Roughly speaking, the algorithm for the closed-closed case resembles the one for the closed-open case in that it now performs similar operations on *both* K_β and K_δ as Algorithm 3.3.2 does on K_β. The idea is as follows:

1. First, instead of directly working on K_β and K_δ, we work on $\overline{K}_\beta$ and $\overline{K}_\delta$, which include some missing $(p+1)$-simplices. Formally,

   $$\overline{K}_\beta = K_\beta \cup \{(p+1)\text{-simplices with all } p\text{-faces in } K_\beta\},$$

   and $\overline{K}_\delta$ is defined similarly.

2. Let $\mathcal{C}_0, \ldots, \mathcal{C}_k$ be all the $(p+1)$-connected components of $\mathbb{K}^p_{(b-1,d+1)} \setminus \left(\overline{K}_\beta \cup \overline{K}_\delta\right)$ with boundaries in $\overline{K}_\beta \cup \overline{K}_\delta$. Then, only $\mathcal{C}_0, \ldots, \mathcal{C}_k$ can be used to form the persistent p-cycles in the p-th regular complexes. Re-index these components such that $\mathcal{C}_0, \ldots, \mathcal{C}_h$ are all the ones whose boundaries contain *both* $\sigma_{\beta-1}$ and σ_δ. We have that $h = 0$ or 1 (i.e., one or two of them). If $h = 0$, then $\mathcal{C}_0$ must take part in forming a sequence of persistent cycles for $\left[\alpha_b^p, \alpha_d^p\right]$. If $h = 1$, then either $\mathcal{C}_0$ or $\mathcal{C}_1$ but not both must take part in forming persistent cycles for the interval.

3. Compute minimum p-cycles in the p-th regular complexes similarly as in Step 2 of Algorithm 3.3.2. For a $\mathcal{C}_{\mathrm{j}}$, let M_{j} be its closure. If the boundary of $\mathcal{C}_{\mathrm{j}}$ lies completely in $\overline{K}_\beta$, the computed p-cycles $\left\{\zeta_{\mathrm{i}}^{\mathrm{j}} \subseteq M_{\mathrm{j}} \cap \mathbb{K}^p_{(\mathrm{i},\mathrm{i}+1)} \mid b \le \mathrm{i} < d\right\}$ is the set with the minimum sum of weight satisfying the conditions as in Step 2 of Algorithm 3.3.2. If the boundary of $\mathcal{C}_{\mathrm{j}}$ lies completely in $\overline{K}_\delta$, the computed minimum p-cycles satisfy symmetric conditions. If the boundary of $\mathcal{C}_{\mathrm{j}}$ intersects both $\overline{K}_\beta$ and $\overline{K}_\delta$, the computed minimum p-cycles satisfy: $\zeta_b^{\mathrm{j}} \sim \partial(\mathcal{C}_{\mathrm{j}}) \cap \overline{K}_\beta$ in $\mathbb{K}^p_{(b-1,b+1)}$, $\zeta_{d-1}^{\mathrm{j}} \sim \partial(\mathcal{C}_{\mathrm{j}}) \cap \overline{K}_\delta$ in $\mathbb{K}^p_{(d-1,d+1)}$, and the consecutive cycles are homologous.

4. To compute the optimal persistent p-cycles, we build a dual graph for $\overline{K}_\beta \cup \overline{K}_\delta$, in which the boundary of each $\mathcal{C}_{\mathrm{j}}$ corresponds to a dummy vertex ϕ_{j}, and the remaining boundary portion corresponds to a dummy vertex $\overline{\phi}$. We also add the augmenting edges to the dual graph and set their weights similarly to Algorithm 3.3.2. For each i s.t. $0 \le \mathrm{i} \le h$, we build an (s,t)-graph on the dual graph of $\overline{K}_\beta \cup \overline{K}_\delta$ with source being $\left\{\phi_{\mathrm{i}}\right\}$ and sink being $\left\{\overline{\phi}, \phi_0, \dots, \phi_h\right\} \setminus \left\{\phi_{\mathrm{i}}\right\}$. The minimum (s,t)-cut for all the (s,t)-graphs we build produces an optimal sequence of persistent p-cycles for $\left[\alpha_b^p, \alpha_d^p\right]$.

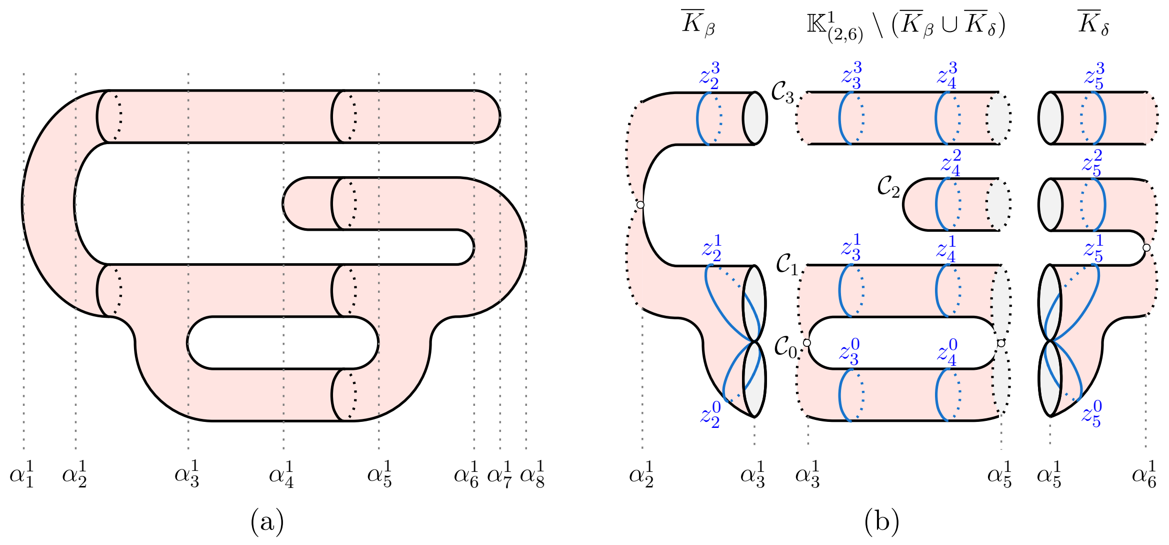


Figure 3.7. (a) A complex K with the height function f taken over the horizontal line and the 1st critical values listed. (b) The relevant subcomplex $\mathbb{K}^p_{(b-1,d+1)} = \mathbb{K}^1_{(2,6)}$ for the interval $\left[\alpha_3^1, \alpha_5^1\right]$, where $\overline{K}_\beta$ and $\overline{K}_\delta$ are broken from the remaining parts for a better illustration. An empty dot indicates that the point is not included in the space.

We can look at Figure 3.7 for intuitions of the above algorithm, where $p = 1$ and the interval of interest is $\left[\alpha_3^1, \alpha_5^1\right]$. In Figure 3.7b, there are four 2-connected components of $\mathbb{K}^1_{(2,6)} \setminus \left(\overline{K}_\beta \cup \overline{K}_\delta\right)$ with boundaries in $\overline{K}_\beta \cup \overline{K}_\delta$, which are $\mathcal{C}_0$, $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_3$. Among them, $\mathcal{C}_0$, $\mathcal{C}_1$ are the ones whose boundaries contain both $\sigma_{\beta-1}$ and σ_δ. The persistent 1-cycles $z_2^1 + z_2^3, z_3^1 + z_3^3, z_4^1 + z_4^3, z_5^1 + z_5^3$ come from the components $\mathcal{C}_1$ and $\mathcal{C}_3$, in which the starting one $z_2^1+z_2^3$ is homologous to $\partial(\mathcal{C}_1)\cap\overline{K}_\beta+\partial(\mathcal{C}_3)\cap\overline{K}_\beta$, and the ending one $z_5^1+z_5^3$ is homologous to $\partial(\mathcal{C}_1) \cap \overline{K}_\delta + \partial(\mathcal{C}_3) \cap \overline{K}_\delta$. Another sequence $z_2^0, z_3^0, z_4^0 + z_4^2, z_5^0 + z_5^2$ comes from $\mathcal{C}_0$ and $\mathcal{C}_2$, in which the starting one z_2^0 is homologous to $\partial(\mathcal{C}_0) \cap \overline{K}_\beta$, and the ending one $z_5^0 + z_5^2$ is homologous to $\partial(\mathcal{C}_0) \cap \overline{K}_\delta + \partial(\mathcal{C}_2)$. To compute the optimal sequence of persistent 1-cycles, one first computes the minimum 1-cycles (e.g., $\left\{\zeta_3^3, \zeta_4^3\right\}$) in each component of $\mathcal{C}_0, \ldots, \mathcal{C}_3$. Then, to determine the optimal combination of the components and the cycles in K_β and K_δ, one leverages the dual graph of $\overline{K}_\beta \cup \overline{K}_\delta$ and the augmenting edges.

We finally note that for the degenerate case of $b = d$, since there are no p-th regular complexes between K_β and K_δ, the algorithm needs an adjustment: one simply does not add augmenting edges at all.

Complexity.

Let n be the number of bits used to encode K. Then, for the three algorithms in this section, operations other than the minimum cut computation can be done in $O(n \log n)$ time. Using the max-flow algorithm by Orlin [47], the time complexity of all three algorithms is $O(n^2)$. Note that we assume persistence intervals to be given so that the time used for computing the levelset zigzag barcode is not included.

3.4 Equivalence of p-th and classical levelset filtrations

In this section, we prove that the p-th levelset filtration defined in Section 3.2.1 and the classical one defined by Carlsson et al. [27] produce equivalent p-th persistence intervals. We first recall the classical definition in Section 3.4.1 and then prove our conclusion in Section 3.4.2.

3.4.1 Classical levelset zigzag

Throughout this section, let K be a finite simplicial complex with underlying space $X = |K|$ and $f : X \to \mathbb{R}$ be a generic PL function with critical values $\alpha_0 = -\infty < \alpha_1 < \cdots < \alpha_n < \alpha_{n+1} = \infty$. The original construction [27] of levelset zigzag persistence picks regular values $s_0, s_1, \ldots, s_n$ so that each $s_\mathrm{i} \in (\alpha_\mathrm{i}, \alpha_{\mathrm{i}+1})$. Then, the *levelset filtration* of f, denoted $\mathcal{L}^\mathsf{c}(f)$, is defined as

$$\mathcal{L}^\mathsf{c}(f) : f^{-1}(s_0) \hookrightarrow f^{-1}[s_0, s_1] \hookleftarrow f^{-1}(s_1) \hookrightarrow f^{-1}[s_1, s_2] \hookleftarrow \cdots \hookrightarrow f^{-1}[s_{n-1}, s_n] \hookleftarrow f^{-1}(s_n) \tag{3.6}$$

In order to align with our constructions in Section 3.2.1, we adopt an alternative but equivalent definition of $\mathcal{L}^\mathsf{c}(f)$ as follows, where we denote $f^{-1}(\alpha_\mathrm{i}, \alpha_\mathrm{j})$ as $\mathbb{X}_{(\mathrm{i},\mathrm{j})}$:

$$\mathcal{L}^\mathsf{c}(f) : \mathbb{X}_{(0,1)} \hookrightarrow \mathbb{X}_{(0,2)} \hookleftarrow \mathbb{X}_{(1,2)} \hookrightarrow \mathbb{X}_{(1,3)} \hookleftarrow \cdots \hookrightarrow \mathbb{X}_{(n-1,n+1)} \hookleftarrow \mathbb{X}_{(n,n+1)} \tag{3.7}$$

Note that each $\mathbb{X}_{(\mathrm{i},\mathrm{i}+1)}$ deformation retracts to $f^{-1}(s_\mathrm{i})$ and each $\mathbb{X}_{(\mathrm{i}-1,\mathrm{i}+1)}$ deformation retracts to $f^{-1}[s_{\mathrm{i}-1}, s_\mathrm{i}]$, so that zigzag modules induced by the two filtrations in (3.6) and (3.7) are isomorphic.

The barcode $\mathsf{Pers}_p(\mathcal{L}^\mathsf{c}(f))$ is then the classical version of p-th levelset barcode defined in [27]. Intervals in $\mathsf{Pers}_p(\mathcal{L}^\mathsf{c}(f))$ can also be mapped to real-value intervals in which the homological features persist:

closed-open:	$\left[\mathbb{X}_{(b-1,b+1)}, \mathbb{X}_{(d-1,d)}\right]$	$\Leftrightarrow$	$[\alpha_b, \alpha_d)$
open-closed:	$\left[\mathbb{X}_{(b,b+1)}, \mathbb{X}_{(d-1,d+1)}\right]$	$\Leftrightarrow$	$(\alpha_b, \alpha_d]$
closed-closed:	$\left[\mathbb{X}_{(b-1,b+1)}, \mathbb{X}_{(d-1,d+1)}\right]$	$\Leftrightarrow$	$[\alpha_b, \alpha_d]$
open-open:	$\left[\mathbb{X}_{(b,b+1)}, \mathbb{X}_{(d-1,d)}\right]$	$\Leftrightarrow$	(α_b, α_d)

3.4.2 Equivalence

The following theorem is the major conclusion of this section (recall that $\mathcal{L}^{\mathsf{c}}_p(f)$ is the continuous version of p-th levelset filtration of f as in Definition 3.2.2):

Theorem 3.4.1. *For an arbitrary PL function f as defined above, the real-value intervals in $\mathsf{Pers}_p(\mathcal{L}^{\mathsf{c}}(f))$ and $\mathsf{Pers}_p(\mathcal{L}^{\mathsf{c}}_p(f))$ are the same.*

To prove Theorem 3.4.1, we first provide the following proposition:

Proposition 3.4.1. *Let $\alpha_\ell \leq \alpha_{\mathrm{i}} < \alpha_{\mathrm{j}} \leq \alpha_k$ be critical values of f. If α_h is not a p-th homologically critical value for each h s.t. $\ell < h \leq \mathrm{i}$ or $\mathrm{j} \leq h < k$, then the map $\mathsf{H}_p(\mathbb{X}_{(\mathrm{i},\mathrm{j})}) \to \mathsf{H}_p(\mathbb{X}_{(\ell,k)})$ induced by inclusion is an isomorphism.*

$$\begin{tikzcd}[column sep=tiny]
 & & & & & \mathbb{X}_{(\mathrm{i},k)} & & & & & \\
 & & & & \mathbb{X}_{(\mathrm{i},\mathrm{i+4})} \arrow[ur] & & \mathbb{X}_{(\mathrm{i},k)} \arrow[ul, dashed] & & & & \\
 & \mathcal{D}_1 & & \mathbb{X}_{(\mathrm{i},\mathrm{j})} \arrow[ur] & & \mathbb{X}_{(\mathrm{i},\mathrm{i+4})} \arrow[ul, dashed] \arrow[ur, dashed] & & \mathbb{X}_{(\mathrm{i+1},k)} \arrow[ul, dashed] & & & \\
 & & \mathbb{X}_{(\mathrm{i},\mathrm{i+2})} \arrow[ur] & & \mathbb{X}_{(\mathrm{i},\mathrm{j})} \arrow[ul, dashed] \arrow[ur, dashed] & & \mathbb{X}_{(\mathrm{i+1},\mathrm{i+4})} \arrow[ul, dashed] \arrow[ur, dashed] & & \mathbb{X}_{(\mathrm{i+2},k)} \arrow[ul, dashed] & & \\
 & \mathbb{X}_{(\mathrm{i},\mathrm{i+1})} \arrow[ur] & & \mathbb{X}_{(\mathrm{i},\mathrm{i+2})} \arrow[ul, dashed] \arrow[ur, dashed] & & \mathbb{X}_{(\mathrm{i+1},\mathrm{j})} \arrow[ul, dashed] \arrow[ur, dashed] & & \mathbb{X}_{(\mathrm{i+2},\mathrm{i+4})} \arrow[ul, dashed] \arrow[ur, dashed] & & \mathbb{X}_{(\mathrm{i+3},k)} \arrow[ul, dashed] & \\
\varnothing \arrow[ur] & & \mathbb{X}_{(\mathrm{i},\mathrm{i+1})} \arrow[ul] \arrow[ur] & & \mathbb{X}_{(\mathrm{i+1},\mathrm{i+2})} \arrow[ul] \arrow[ur] & & \mathbb{X}_{(\mathrm{i+2},\mathrm{j})} \arrow[ul] \arrow[ur, "\sim"] & & \mathbb{X}_{(\mathrm{i+3},\mathrm{i+4})} \arrow[ul, "\sim"'] \arrow[ur, "\sim"] & & \mathbb{X}_{(\mathrm{i+4},k)} \arrow[ul, "\sim"'] \\
 & & & & & \mathcal{D}_2 & & & & &
\end{tikzcd}$$

Figure 3.8. Mayer-Vietoris pyramid for $\mathrm{j} = \mathrm{i} + 3$, $k = \mathrm{i} + 5$.

Proof. We first prove that the inclusion-induced map $\mathsf{H}_p(\mathbb{X}_{(\mathrm{i},\mathrm{j})}) \to \mathsf{H}_p(\mathbb{X}_{(\mathrm{i},k)})$ is an isomorphism. For this, we build a Mayer-Vietoris pyramid similar to the one in [27] for proving the Pyramid Theorem. Moreover, in the pyramid, let $\mathcal{D}_1$ be the filtration along the north-eastbound diagonal and $\mathcal{D}_2$ be the filtration along the bottom. An example is shown in Figure 3.8 for $\mathrm{j} = \mathrm{i}+3$, $k = \mathrm{i}+5$, where inclusion arrows in $\mathcal{D}_1$, $\mathcal{D}_2$ are solid and the remaining arrows are dashed. Since all diamonds in the pyramid are Mayer-Vietoris diamonds [27], each interval $[\mathbb{X}_{(\mathrm{i},\mathrm{i}+b)}, \mathbb{X}_{(\mathrm{i},\mathrm{i}+d)}]$ in $\mathsf{Pers}_p(\mathcal{D}_1)$ corresponds to the following interval in $\mathsf{Pers}_p(\mathcal{D}_2)$:

$$\begin{cases} \left[\mathbb{X}_{(\mathrm{i},\mathrm{i}+1)}, \mathbb{X}_{(\mathrm{i}+d-1,\mathrm{i}+d)}\right] & \text{if } b = 1 \\ \left[\mathbb{X}_{(\mathrm{i}+b-2,\mathrm{i}+b)}, \mathbb{X}_{(\mathrm{i}+d-1,\mathrm{i}+d)}\right] & \text{otherwise} \end{cases}$$

The fact that α_h is not a p-th critical value for $\mathrm{j} \leq h < k$ implies that linear maps in $\mathsf{H}_p(\mathcal{D}_2)$ induced by arrows between $\mathbb{X}_{(\mathrm{j}-1,\mathrm{j})}$ and $\mathbb{X}_{(k-1,k)}$ (i.e., those arrows marked with '$\approx$' in the example) are isomorphisms. This means that no interval in $\mathsf{Pers}_p(\mathcal{D}_2)$ starts with $\mathbb{X}_{(h-1,h+1)}$ or ends with $\mathbb{X}_{(h-1,h)}$ for $\mathrm{j} \leq h < k$. So we have that no interval in $\mathsf{Pers}_p(\mathcal{D}_1)$ starts with $\mathbb{X}_{(\mathrm{i},h+1)}$ or ends with $\mathbb{X}_{(\mathrm{i},h)}$ for $\mathrm{j} \leq h < k$. This in turn means that each $\mathsf{H}_p(\mathbb{X}_{(\mathrm{i},h)}) \to \mathsf{H}_p(\mathbb{X}_{(\mathrm{i},h+1)})$ in $\mathsf{H}_p(\mathcal{D}_1)$ is an isomorphism for $\mathrm{j} \leq h < k$, which implies that their composition $\mathsf{H}_p(\mathbb{X}_{(\mathrm{i},\mathrm{j})}) \to \mathsf{H}_p(\mathbb{X}_{(\mathrm{i},k)})$ is an isomorphism.

Symmetrically, we have that $\mathsf{H}_p(\mathbb{X}_{(\mathrm{i},k)}) \to \mathsf{H}_p(\mathbb{X}_{(\ell,k)})$ is an isomorphism, which implies that $\mathsf{H}_p(\mathbb{X}_{(\mathrm{i},\mathrm{j})}) \to \mathsf{H}_p(\mathbb{X}_{(\ell,k)})$ is an isomorphism. □

We now prove Theorem 3.4.1. Let $\alpha^p_0 = -\infty < \alpha^p_1 < \cdots < \alpha^p_m < \alpha^p_{m+1} = \infty$ be all the p-th homologically critical values of f, and let $\alpha^p_\mathrm{i} = \alpha_{\lambda_\mathrm{i}}$ for each i. Note that $\mathbb{X}^p_{(\mathrm{i},\mathrm{j})} = \mathbb{X}_{(\lambda_\mathrm{i},\lambda_\mathrm{j})}$ for $\mathrm{i} < \mathrm{j}$. We first show that the two zigzag modules as defined in Figure 3.9 are isomorphic, where the upper module is $\mathsf{H}_p(\mathcal{L}^\mathsf{c}(f))$, and the lower module is a version of $\mathsf{H}_p(\mathcal{L}^\mathsf{c}_p(f))$ elongated by making several copies of p-th homology groups of the regular subspaces and connecting them by identity maps. The commutativity of the diagram is easily seen because all maps are induced by inclusion. The vertical maps are isomorphisms by Proposition 3.4.1. Hence, the two modules in Figure 3.9 are isomorphic. This means that persistence intervals of the two modules bijectively map to each other, and we also have that their corresponding real-value intervals are the same. For example, an interval $[\mathbb{X}_{(\lambda_b-1,\lambda_b+1)}, \mathbb{X}_{(\lambda_d-1,\lambda_d)}]$ from $\mathsf{H}_p(\mathcal{L}^\mathsf{c}(f))$ corresponds to an interval $[\mathbb{X}^p_{(b-1,b+1)}, \mathbb{X}^p_{(d-1,d)}]$ from $\mathsf{H}_p(\mathcal{L}^\mathsf{c}_p(f))$, and they both produce the real-value interval $[\alpha_{\lambda_b}, \alpha_{\lambda_d})$.

$$\begin{tikzcd}
\cdots \arrow[r] & \mathsf{H}_p(\mathbb{X}_{(\lambda_\mathrm{i}-1,\lambda_\mathrm{i}+1)}) \arrow[d, "\wr"] & \mathsf{H}_p(\mathbb{X}_{(\lambda_\mathrm{i},\lambda_\mathrm{i}+1)}) \arrow[l] \arrow[r] \arrow[d, "\wr"] & \mathsf{H}_p(\mathbb{X}_{(\lambda_\mathrm{i},\lambda_\mathrm{i}+2)}) \arrow[d, "\wr"] & \cdots \arrow[l] & \mathsf{H}_p(\mathbb{X}_{(\lambda_\mathrm{i+1}-1,\lambda_\mathrm{i+1})}) \arrow[l] \arrow[r] \arrow[d, "\wr"] & \cdots \\
\cdots \arrow[r] & \mathsf{H}_p(\mathbb{X}^p_{(\mathrm{i}-1,\mathrm{i}+1)}) & \mathsf{H}_p(\mathbb{X}^p_{(\mathrm{i},\mathrm{i}+1)}) \arrow[l] \arrow[r, equal] & \mathsf{H}_p(\mathbb{X}^p_{(\mathrm{i},\mathrm{i}+1)}) \arrow[r, equal] & \cdots \arrow[r, equal] & \mathsf{H}_p(\mathbb{X}^p_{(\mathrm{i},\mathrm{i}+1)}) \arrow[r] & \cdots
\end{tikzcd}$$

Figure 3.9. Two isomorphic zigzag modules where the upper module is $\mathsf{H}_p(\mathcal{L}^\mathsf{c}(f))$ and the lower module is an elongated version of $\mathsf{H}_p(\mathcal{L}^\mathsf{c}_p(f))$.

3.5 Connection to interval decomposition

In this section, we connect levelset persistent cycles to the interval decomposition of zigzag modules. Specifically, for a generic PL function f, we show that levelset persistent p-cycles induce an entire interval decomposition for $\mathsf{H}_p(\mathcal{L}_p(f))$ (Theorem 3.5.2), and part of an interval decomposition for $\mathsf{H}_p(\mathcal{F}_p(f))$ with the rest being from the trivial intervals (Theorem 3.5.1).

To reach the conclusions, we first define *representative cycles* for a simplex-wise filtration, which generate an interval submodule in a straightforward way, i.e., picking a cycle for a homology class at each position. Note that similar definitions also appear in [21].

Definition 3.5.1. *Let $p \geq 0$, $\mathcal{X} : X_0 \leftrightarrow \cdots \leftrightarrow X_\ell$ be a simplex-wise filtration, and $[b, d]$ be an interval in $\mathsf{Pers}_p(\mathcal{X})$. Denote each linear map in $\mathsf{H}_p(\mathcal{X})$ as $\psi_\mathrm{j} : \mathsf{H}_p(X_\mathrm{j}) \leftrightarrow \mathsf{H}_p(X_{\mathrm{j}+1})$. The representative p-cycles for $[b, d]$ is a sequence of p-cycles $\{z_\mathrm{i} \subseteq X_\mathrm{i} \mid b \leq \mathrm{i} \leq d\}$ such that:*

1. *For $b > 0$, $[z_b]$ is not in $\mathsf{img}(\psi_{b-1})$ if $X_{b-1} \hookrightarrow X_b$ is forward, or $[z_b]$ is the non-zero class in $\mathsf{ker}(\psi_{b-1})$ otherwise.*

2. *For $d < \ell$, $[z_d]$ is not in $\mathsf{img}(\psi_d)$ if $X_d \hookleftarrow X_{d+1}$ is backward, or $[z_d]$ is the non-zero class in $\mathsf{ker}(\psi_d)$ otherwise.*

3. *For each $\mathrm{i} \in [b, d-1]$, $[z_\mathrm{i}] \leftrightarrow [z_{\mathrm{i}+1}]$ by ψ_i, i.e., $[z_\mathrm{i}] \mapsto [z_{\mathrm{i}+1}]$ or $[z_\mathrm{i}] \mapsfrom [z_{\mathrm{i}+1}]$.*

The interval submodule $\mathcal{I}$ of $\mathsf{H}_p(\mathcal{X})$ induced by the representative p-cycles is a module such that $\mathcal{I}(\mathrm{i})$ equals the 1-dimensional vector space generated by $[z_\mathrm{i}]$ for $\mathrm{i} \in [b, d]$ and equals 0 otherwise, where $\mathcal{I}(\mathrm{i})$ is the i-th vector space in $\mathcal{I}$.

The following proposition connects representative cycles to the interval decomposition:

Proposition 3.5.1. *Let $p \geq 0$, $\mathcal{X} : X_0 \leftrightarrow \cdots \leftrightarrow X_\ell$ be a simplex-wise filtration with $\mathsf{H}_p(X_0) = 0$, and $\mathsf{Pers}_p(\mathcal{X}) = \{[b_\alpha, d_\alpha] \mid \alpha \in \mathcal{A}\}$ be indexed by a set $\mathcal{A}$. One has that $\mathsf{H}_p(\mathcal{X})$ is* ***equal to*** *a direct sum of interval submodules $\bigoplus_{\alpha\in\mathcal{A}} \mathcal{I}^{[b_\alpha,d_\alpha]}$ if and only if for each α, $\mathcal{I}^{[b_\alpha,d_\alpha]}$ is induced by a sequence of representative p-cycles for $[b_\alpha, d_\alpha]$.*

Proof. Suppose that $\mathsf{H}_p(\mathcal{X}) = \bigoplus_{\alpha\in\mathcal{A}} \mathcal{I}^{[b_\alpha,d_\alpha]}$ is an interval decomposition. For each α, define a sequence of representative p-cycles $\{z^\alpha_{\mathrm{i}} \mid b_\alpha \leq \mathrm{i} \leq d_\alpha\}$ for $[b_\alpha, d_\alpha]$ by letting z^α_{i} be an arbitrary cycle in the non-zero class of the i-th vector space of $\mathcal{I}^{[b_\alpha,d_\alpha]}$. It can be verified that $\{z^\alpha_{\mathrm{i}} \mid b_\alpha \leq \mathrm{i} \leq d_\alpha\}$ are valid representative p-cycles for $[b_\alpha, d_\alpha]$ inducing $\mathcal{I}^{[b_\alpha,d_\alpha]}$. This finishes the “only if” part of the proof. The “if” part follows directly from the proof of Proposition 4.3.1. □

Now consider a generic PL function $f : |K| \to \mathbb{R}$ on a finite simplicial complex K and a non-trivial interval $[K_\beta, K_\delta]$ of $\mathsf{Pers}_p(\mathcal{F}_p(f))$ for $p \geq 1$. A sequence of levelset persistent p-cycles $\{z_{\mathrm{i}}\}$ for $[K_\beta, K_\delta]$ *induces* a sequence of representative p-cycles $\{\zeta_{\mathrm{j}} \mid \beta \leq \mathrm{j} \leq \delta\}$ for this interval as follows: for any $K_{\mathrm{j}} \in [K_\beta, K_\delta]$, we can always find a z_{i} satisfying $z_{\mathrm{i}} \subseteq K_{\mathrm{j}}$, i.e., the complex that z_{i} originally comes from is included in K_{j}; then, set $\zeta_{\mathrm{j}} = z_{\mathrm{i}}$. It can be verified that the induced representative p-cycles are valid so that levelset persistent cycles also induce interval submodules. We then have the following fact:

Theorem 3.5.1. *For any non-trivial interval J of $\mathsf{Pers}_p(\mathcal{F}_p(f))$, a sequence of levelset persistent p-cycles for J induces an interval submodule of $\mathsf{H}_p(\mathcal{F}_p(f))$ over J. These induced interval submodules constitute part of an interval decomposition for $\mathsf{H}_p(\mathcal{F}_p(f))$, where the remaining parts are from the trivial intervals.*

Proof. This follows from Proposition 3.5.1. Note that in order to apply Proposition 3.5.1, $\mathsf{H}_p(\mathbb{K}^p_{(0,1)})$ has to be trivial, where $\mathbb{K}^p_{(0,1)}$ is the starting complex of $\mathcal{F}_p(f)$. If the minimum value of f is p-th critical, then $\mathbb{K}^p_{(0,1)} = \mathbb{K}_{(0,1)} = \varnothing$, and so $\mathsf{H}_p(\mathbb{K}^p_{(0,1)})$ is trivial. Otherwise, since $\mathsf{H}_p(\mathbb{K}^p_{(0,1)}) = \mathsf{H}_p(\mathbb{K}_{(0,2)})$ (Proposition 3.4.1) and $\mathbb{K}_{(0,2)}$ deformation retracts to a point, we have that $\mathsf{H}_p(\mathbb{K}^p_{(0,1)})$ is trivial. □

Similarly as for $\mathsf{H}_p(\mathcal{F}_p(f))$, levelset persistent p-cycles can also induce interval submodules for $\mathsf{H}_p(\mathcal{L}_p(f))$, the details of which are omitted. The following fact follows:

Theorem 3.5.2. *Let* $\mathsf{Pers}_p(\mathcal{L}_p(f)) = \{J_k \mid k \in \Lambda\}$ *be indexed by a set* Λ. *For any interval* J_k *of* $\mathsf{Pers}_p(\mathcal{L}_p(f))$, *a sequence of levelset persistent* p*-cycles for* J_k *induces an interval submodule* $\mathcal{I}_k$ *of* $\mathsf{H}_p(\mathcal{L}_p(f))$ *over* J_k. *Combining all the modules, one derives an interval decomposition* $\mathsf{H}_p(\mathcal{L}_p(f)) = \bigoplus_{k\in\Lambda} \mathcal{I}_k$.

Proof. This follows from Theorem 3.5.1. Note that $\mathsf{H}_p(\mathcal{L}_p(f))$ can be viewed as being "contracted" from $\mathsf{H}_p(\mathcal{F}_p(f))$. While in Theorem 3.5.1, the induced interval submodules form only part of the interval decomposition of $\mathsf{H}_p(\mathcal{F}_p(f))$, the remaining submodules from the trivial intervals disappear in the interval decomposition of $\mathsf{H}_p(\mathcal{L}_p(f))$. $\square$

4. COMPUTING GRAPH ZIGZAG PERSISTENCE USING REPRESENTATIVES

In this chapter, we look into the computation of zigzag persistent homology on graph inputs with the help of representatives. Our research is motivated by the fact that graphs appear in many applications as abstraction of real-world phenomena, where vertices represent certain objects and edges represent their relations. Furthermore, rather than being stationary, graph data obtained in applications usually change with respect to some parameter such as time. Persistent homology is then a suitable descriptor for the changing graph data because it quantifies the life span of topological features as the graph changes. However, standard non-zigzag persistence [3] only allows addition of vertices and edges during the change, whereas deletion may also happen in practice. For example, many complex systems such as social networks, food webs, or disease spreading are modeled by the so-called "dynamic networks" [29], [54], [55] (see Figure 4.1 for an example), where vertices and edges can appear and disappear at different time. Therefore, zigzag persistence proposed by Carlsson and de Silva [7] is a more natural tool in such scenarios because simplices can be both added and deleted.

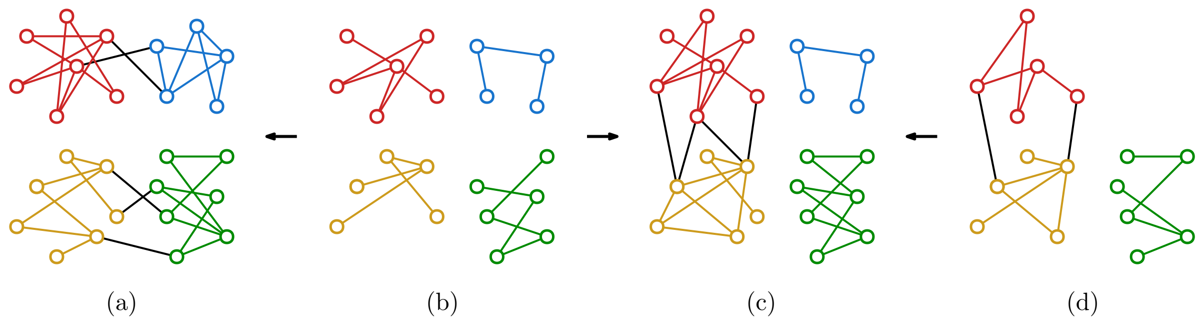


Figure 4.1. A dynamic network with four prominent clusters each colored differently. Black edges connect different clusters and forward (resp. backward) arrows indicate additions (resp. deletions) of vertices and edges. From (a) to (b), two clusters split; from (b) to (c), two clusters merge; from (c) to (d), one cluster disappears.

Algorithms for both zigzag and non-zigzag persistence have a general-case time complexity of $O(m^\omega)$ [3], [21], [27], [30], where m is the length of the input filtration and $\omega < 2.37286$

is the matrix multiplication exponent [31]. For the special case of graph filtrations, it is well known that non-zigzag persistence can be computed in $O(m\,\alpha(m))$ time, where $\alpha(m)$ is the inverse Ackermann's function that is almost constant for all practical purposes [56]. However, analogous faster algorithms for zigzag persistence on graphs are not known. In this chapter, we present algorithms for zigzag persistence on graphs with near-linear time complexity. In particular, given a zigzag filtration of length m for a graph with n vertices and edges, our algorithm for 0-dimension runs in $O(m\log^2 n + m\log m)$ time, and our algorithm for 1-dimension runs in $O(m\log^4 n)$ time. Observe that the algorithm for 0-dimension works for arbitrary complexes by restricting to the 1-skeletons.

The difficulty in designing faster zigzag persistence algorithms for the special case of graphs lies in the deletion of vertices and edges. For example, besides merging into bigger ones, connected components can also split into smaller ones because of edge deletion. Therefore, one cannot simply kill the younger component during merging as in standard persistence [3], but rather has to pair the *merge* and *departure* events with the *split* and *entrance* events (see Sections 4.3 for details). Similarly, in dimension one, deletion of edges may kill 1-cycles so that one has to properly pair the creation and destruction of 1-cycles, instead of simply treating all 1-dimensional intervals as infinite ones.

Our solutions are as follows: in dimension zero, we find that the $O(n\log n)$ algorithm by Agarwal et al. [32] originally designed for pairing critical points of Morse functions on 2-manifolds can be utilized in our scenario. We formally prove the correctness of applying the algorithm with the help of representatives, and use a *dynamic connectivity* data structure [57] to achieve the claimed complexity. The algorithm for 1-dimension finds a pairing of the *positive* and *negative* edges such that representative cycles for all pairs exist (see the Pairing Principle in Section 4.4). We further reduce the pairing to finding the *max edge-weight* of a path in a minimum spanning forest. Utilizing a data structure for *dynamic minimum spanning forest* [57], we achieve the claimed time complexity.

4.1 Literature review

The algorithm for computing persistent homology by Edelsbrunner et al. [3] is a cornerstone of topological data analysis. Several extensions followed after this initial development. De Silva et al. [58] proposed to compute persistent *cohomology* instead of homology which gives the same barcode. De Silva et al. [15] then showed that the persistent cohomology algorithm runs faster in practice than the version that uses homology. The *annotation* technique proposed by Dey et al. [8] implements the cohomology algorithm by maintaining a cohomology basis more succinctly and extends to *towers* connected by simplicial maps. These algorithms run in $O(m^3)$ time.

Carlsson and de Silva [7] introduced zigzag persistence as an extension of the standard persistence, where they also presented a decomposition algorithm for computing zigzag barcodes on the level of vector spaces and linear maps. This algorithm is then adapted to zigzag filtrations at simplicial level by Carlsson et al. [27] with a time complexity of $O(m^3)$. Both algorithms [7], [27] utilize a construct called *right filtration* and a *birth-time vector.* Maria and Oudot [21] proposed an algorithm for zigzag persistence based on some *diamond principles* where an inverse non-zigzag filtration is always maintained during the process. The algorithm in [21] is shown to run faster in experiments than the algorithm in [27] though the time complexities remain the same. Milosavljević et al. [30] proposed an algorithm for zigzag persistence based on matrix multiplication which runs in $O(m^\omega)$ time, giving the best asymptotic bound for computing zigzag and non-zigzag persistence in general dimensions.

The algorithms reviewed so far are all for general dimensions and many of them are based on matrix operations. Thus, it is not surprising that the best time bound achieved is $O(m^\omega)$ given that computing Betti numbers for a simplicial 2-complex of size m is as hard as computing the rank of a $\mathbb{Z}_2$-matrix with m non-zero entries as shown by Edelsbrunner and Parsa [59]. To lower the complexity, one strategy (which is adopted in this chapter) is to consider special cases where matrix operations can be avoided. The work by Dey [60] is probably most related to ours in that regard, who proposed an $O(m \log m)$ algorithm for non-zigzag persistence induced from height functions on $\mathbb{R}^3$-embedded complexes.

4.2 Preliminaries

We re-use most definitions and conventions presented in Section 3.1, with some additional regulations which we detail in this section.

In this chapter, besides what is defined in Section 3.1, an elementary zigzag module also starts with the trivial (zero) vector space. We also only consider simplex-wise zigzag filtrations, which can have the following form:

$$\mathcal{F} : K_0 \xleftrightarrow{\sigma_0} K_1 \xleftrightarrow{\sigma_1} \cdots \xleftrightarrow{\sigma_{m-1}} K_m$$

where each σ_i denotes the simplex added to or deleted from K_i to form $K_{\mathrm{i}+1}$. For computational purposes, we sometimes assume that a filtration starts with the empty complex, i.e., $K_0 = \varnothing$ in $\mathcal{F}$. Throughout the chapter, we also assume that each K_i in $\mathcal{F}$ is a subcomplex of a fixed complex K; such a K, when not given, can be constructed by taking the union of every K_i in $\mathcal{F}$. In this case, we call $\mathcal{F}$ a filtration *of* K.

In this chapter, whenever $\mathcal{F}$ is used to denote a filtration, we use φ_i^p to denote a linear map in the module $\mathsf{H}_p(\mathcal{F})$. That is, $\mathsf{H}_p(\mathcal{F})$ has the following form:

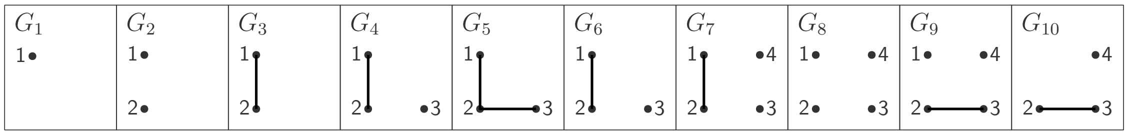
$$\mathsf{H}_p(\mathcal{F}) : \mathsf{H}_p(K_0) \xleftrightarrow{\varphi_0^p} \mathsf{H}_p(K_1) \xleftrightarrow{\varphi_1^p} \cdots \xleftrightarrow{\varphi_{m-1}^p} \mathsf{H}_p(K_m)$$

Note that $\mathsf{H}_p(\mathcal{F})$ is an elementary module if $\mathcal{F}$ starts with an empty complex.

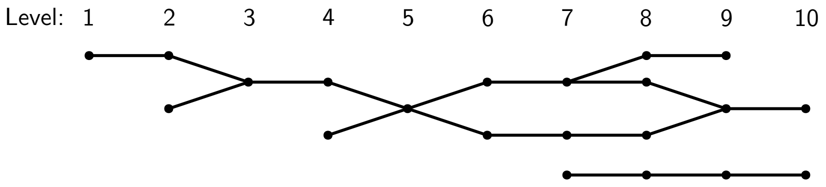
4.3 Zero-dimensional zigzag persistence

We present our algorithm for 0-th zigzag persistence[1] in this section. The input is assumed to be on graphs but note that our algorithm can be applied to any complex by restricting to its 1-skeleton. We first define the barcode graph of a zigzag filtration which is a construct that our algorithm implicitly works on. In a barcode graph, nodes correspond to connected components of graphs in the filtration and edges encode the mapping between the components:

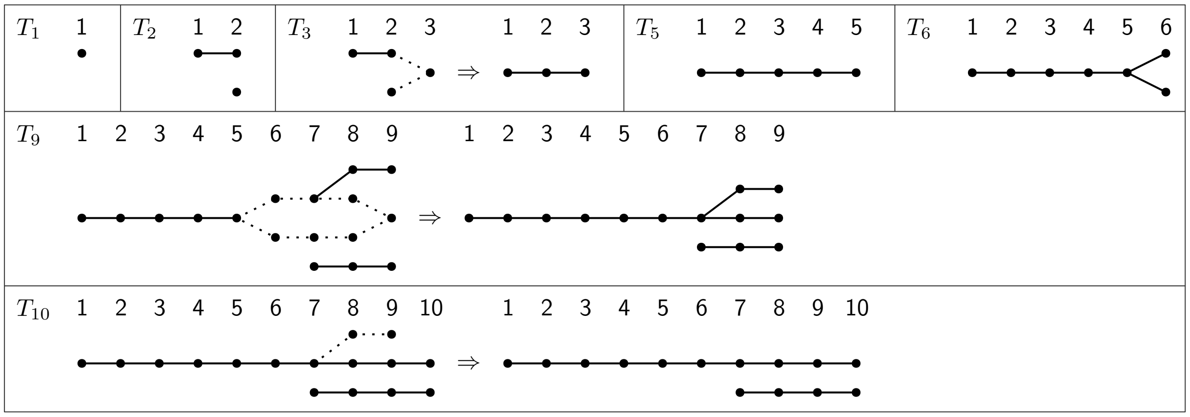
[1]↑For brevity, henceforth we call p-dimensional zigzag persistence as *p-th* zigzag persistence.



(a) A zigzag filtration of graphs with 0-th barcode $\{[2,2],[4,4],[6,8],[8,9],[7,10],[1,10]\}$.



(b) The barcode graph for the filtration shown in Figure 4.2a.



(c) Barcode forests constructed in Algorithm 4.3.1 for the barcode graph in Figure 4.2b. For brevity, some forests are skipped. The horizontally arranged labels indicate the levels.

Figure 4.2. Examples of a zigzag filtration, a barcode graph, and barcode forests.

Definition 4.3.1 (Barcode graph)**.** *For a graph G and a zigzag filtration $\mathcal{F} : G_0 \leftrightarrow G_1 \leftrightarrow \cdots \leftrightarrow G_m$ of G, the* barcode graph $\mathbb{G}_\mathrm{B}(\mathcal{F})$ *of $\mathcal{F}$ is a graph whose vertices (preferably called* nodes*) are associated with a* level *and whose edges connect nodes only at adjacent levels. The graph $\mathbb{G}_\mathrm{B}(\mathcal{F})$ is constructively described as follows:*

- *For each G_i in $\mathcal{F}$ and each connected component of G_i, there is a node in $\mathbb{G}_\mathrm{B}(\mathcal{F})$ at level* i *corresponding to this component; this node is also called a* level-i node.
- *For each inclusion $G_\mathrm{i} \leftrightarrow G_{\mathrm{i}+1}$ in $\mathcal{F}$, if it is forward, then there is an edge connecting a level-*i *node v_i to a level-*$(\mathrm{i}+1)$ *node $v_{\mathrm{i}+1}$ if and only if the component of v_i maps to*

the component of v_{i+1} *by the inclusion. Similarly, if the inclusion is backward, then* v_i *connects to* v_{i+1} *by an edge iff the component of* v_{i+1} *maps to the component of* v_i*.*

For two nodes at different levels in $\mathbb{G}_B(\mathcal{F})$*, the node at the higher* (*resp. lower*) *level is said to be* higher (*resp.* lower) *than the other.*

Remark 4.3.1. Note that some works [54], [60] also have used similar notions of barcode graphs.

Figure 4.2a and 4.2b give an example of a zigzag filtration and its barcode graph. Note that a barcode graph is of size $O(mn)$, where m is the length of $\mathcal{F}$ and n is the number of vertices and edges of G. Although we present our algorithm (Algorithm 4.3.1) by first building the barcode graph, the implementation does not do so explicitly, allowing us to achieve the claimed time complexity; see Section 4.3.1 for the implementation details. Introducing barcode graphs helps us justify the algorithm, and more importantly, points to the fact that the algorithm can be applied whenever such a barcode graph can be built.

Algorithm 4.3.1 (Algorithm for 0-th zigzag persistence)**.**
Given a graph G *and a zigzag filtration* $\mathcal{F} : \varnothing = G_0 \leftrightarrow G_1 \leftrightarrow \cdots \leftrightarrow G_m$ *of* G*, we first build the barcode graph* $\mathbb{G}_B(\mathcal{F})$*, and then apply the pairing algorithm described in [32] on* $\mathbb{G}_B(\mathcal{F})$ *to compute* $\mathsf{Pers}(\mathsf{H}_0(\mathcal{F}))$*. For a better understanding, we rephrase this algorithm which originally works on Reeb graphs:*

The algorithm iterates for $i = 0, \dots, m-1$ *and maintains a* barcode forest T_i*, whose leaves have a one-to-one correspondence to level-*i *nodes of* $\mathbb{G}_B(\mathcal{F})$*. Like the barcode graph, each tree node in a barcode forest is associated with a level and each tree edge connects nodes at adjacent levels. For each tree in a barcode forest, the lowest node is the root. Initially,* T_0 *is empty; then, the algorithm builds* T_{i+1} *from* T_i *in the* i*-th iteration. Intervals for* $\mathsf{Pers}(\mathsf{H}_0(\mathcal{F}))$ *are produced while updating the barcode forest.* (*Figure 4.2c illustrates such updates.*)

Specifically, the i*-th iteration proceeds as follows: first,* T_{i+1} *is formed by copying the level-*$(i+1)$ *nodes of* $\mathbb{G}_B(\mathcal{F})$ *and their connections to the level-*i *nodes, into* T_i*; the copying is possible because leaves of* T_i *and level-*i *nodes of* $\mathbb{G}_B(\mathcal{F})$ *have a one-to-one correspondence; see transitions from* T_5 *to* T_6 *and from* T_9 *to* T_{10} *in Figure 4.2c. We further change* T_{i+1} *under the following events:*

Entrance: *One level-*$(\mathrm{i}+1)$ *node in* $T_{\mathrm{i}+1}$*, said to be* entering*, does not connect to any level-*i *node.*

Split: *One level-*i *node in* $T_{\mathrm{i}+1}$*, said to be* splitting*, connects to two different level-*$(\mathrm{i}+1)$ *nodes. For the two events so far, no changes need to be made on* $T_{\mathrm{i}+1}$*.*

Departure: *One level-*i *node* u *in* $T_{\mathrm{i}+1}$*, said to be* departing*, does not connect to any level-*$(\mathrm{i}+1)$ *node. If* u *has splitting ancestors (i.e., ancestors which are also splitting nodes), add an interval* $[\mathrm{j}+1, \mathrm{i}]$ *to* $\mathsf{Pers}(\mathsf{H}_0(\mathcal{F}))$*, where* j *is the level of the highest splitting ancestor* v *of* u*; otherwise, add an interval* $[\mathrm{j}, \mathrm{i}]$ *to* $\mathsf{Pers}(\mathsf{H}_0(\mathcal{F}))$*, where* j *is the level of the root* v *of* u*. We then delete the path from* v *to* u *in* $T_{\mathrm{i}+1}$*.*

Merge: *Two different level-*i *nodes* u_1, u_2 *in* $T_{\mathrm{i}+1}$ *connect to the same level-*$(\mathrm{i}+1)$ *node. Tentatively,* $T_{\mathrm{i}+1}$ *may now contain a loop and is not a tree. If* u_1, u_2 *are in different trees in* T_{i}*, add an interval* $[\mathrm{j}, \mathrm{i}]$ *to* $\mathsf{Pers}(\mathsf{H}_0(\mathcal{F}))$*, where* j *is the level of the higher root of* u_1, u_2 *in* T_{i}*; otherwise, add an interval* $[\mathrm{j}+1, \mathrm{i}]$ *to* $\mathsf{Pers}(\mathsf{H}_0(\mathcal{F}))$*, where* j *is the level of the highest common ancestor of* u_1, u_2 *in* T_{i}*. We then glue the two paths from* u_1 *and* u_2 *to their level-*j *ancestors in* $T_{\mathrm{i}+1}$*, after which* $T_{\mathrm{i}+1}$ *is guaranteed to be a tree.*

No-change: *If none of the above events happen, no changes are made on* $T_{\mathrm{i}+1}$*.*

At the end, for each root in T_m *at a level* j*, add an interval* $[\mathrm{j}, m]$ *to* $\mathsf{Pers}(\mathsf{H}_0(\mathcal{F}))$*, and for each splitting node in* T_m *at a level* j*, add an interval* $[\mathrm{j}+1, m]$ *to* $\mathsf{Pers}(\mathsf{H}_0(\mathcal{F}))$*.*

Remark 4.3.2. The justification of Algorithm 4.3.1 is given in Section 4.3.2.

Figure 4.2c gives examples of barcode forests constructed by Algorithm 4.3.1 for the barcode graph shown in Figure 4.2b, where T_1 and T_2 introduce entering nodes, T_6 introduces a splitting node, and T_{10} introduces a departing node. In T_{10}, the departure event happens and the dotted path is deleted, producing an interval $[8, 9]$. In T_3 and T_9, the merge event happens and the dotted paths are glued together, producing intervals $[2, 2]$ and $[6, 8]$. Note that the glued level-i nodes are in different trees in T_3 and are in the same tree in T_9.

4.3.1 Implementation

As mentioned, to achieve the claimed time complexity, we do not explicitly build the barcode graph. Instead, we differentiate the different events as follows: inserting (resp. deleting) a vertex in $\mathcal{F}$ simply corresponds to the entrance (resp. departure) event, whereas inserting (resp. deleting) an edge corresponds to the merge (resp. split) event only when connected components in the graph merge (resp. split).

To keep track of the connectivity of vertices, we use a *dynamic connectivity* data structure by Holm et al. [57], which we denote as $\mathbb{D}$. Assuming that m is the length of $\mathcal{F}$ and n is the number of vertices and edges of G, the data structure $\mathbb{D}$ supports the following operations:

- Return the identifier[2] of the connected component of a vertex v in $O(\log n)$ time. We denote this subroutine as `find`(v).
- Insert or delete an edge, and possibly update the connectivity information, in $O(\log^2 n)$ amortized time.

We also note the following implementation details:

- All vertices of G are added to $\mathbb{D}$ initially and are then never deleted. But we make sure that edges in $\mathbb{D}$ always equal edges in G_i as the algorithm proceeds so that $\mathbb{D}$ still records the connectivity of G_i.
- At each iteration i, we update T_i to form $T_{\mathrm{i}+1}$ according to the changes of the connected components from G_i to $G_{\mathrm{i}+1}$. For this, we maintain a key-value map ϕ from connected components of $\mathbb{D}$ to leaves of the barcode forest, and ϕ is initially empty.
- In a barcode forest T_i, since the level of a leaf always equals i, we only record the level of a non-leaf node. Note that at iteration i, a leaf in T_i may uniquely connect to a single leaf in $T_{\mathrm{i}+1}$. In this case, we simply let the leaf in T_i automatically become a leaf in $T_{\mathrm{i}+1}$; see Figure 4.3. The size of a barcode forest is then $O(m)$.

[2]↑Since $\mathbb{D}$ maintains the connectivity information by dynamically updating the spanning forest for the current graph, the identifier of a connected component is indeed the identifier of a tree in the spanning forest.

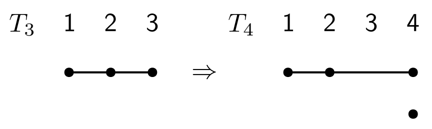


Figure 4.3. For the example in Figure 4.2, to form T_4, our implementation only adds a level-4 entering node, whereas the leaf in T_3 is not touched. Since the level of a leaf always equals the index of the barcode forest, the leaf at level 3 in T_3 automatically becomes a leaf at level 4 in T_4.

Now we can present the full detail of the implementation. Specifically, for each addition and deletion in $\mathcal{F}$, we do the following in each case:

Adding vertex $\sigma_{\mathrm{i}} = v$: Add an isolated node to the barcode forest and let $\phi(\texttt{find}(v))$ equal this newly added node.

Deleting vertex $\sigma_{\mathrm{i}} = v$: Let $\ell = \phi(\texttt{find}(v))$; then, ℓ is the node in the barcode forest that is departing. Update the barcode forest as described in Algorithm 4.3.1.

Adding edge $\sigma_{\mathrm{i}} = (u, v)$: Let $t_1 = \texttt{find}(u)$, $t_2 = \texttt{find}(v)$, $\ell_1 = \phi(t_1)$, and $\ell_2 = \phi(t_2)$. If $t_1 = t_2$, then the no-change event happens; otherwise, the merge event happens. We then add (u, v) to $\mathbb{D}$. For the no-change event, do nothing after this. For the merge event, do the following: glue the paths from ℓ_1 and ℓ_2 to their ancestors as described in Algorithm 4.3.1; attach a new child ℓ to the highest glued node; update $\phi(\texttt{find}(u))$ to be ℓ.

Deleting edge $\sigma_{\mathrm{i}} = (u, v)$: Let $\ell = \phi(\texttt{find}(u))$, and then delete (u, v) from $\mathbb{D}$. If $\texttt{find}(u) = \texttt{find}(v)$ after this, then the no-change event happens but we have to update $\phi(\texttt{find}(u))$ to be ℓ because the identifiers of the connected components in $\mathbb{D}$ may change after deleting the edge [57]. Otherwise, the split event happens: we attach two new children ℓ_1, ℓ_2 to ℓ in the barcode forest and set $\phi(\texttt{find}(u)) = \ell_1$, $\phi(\texttt{find}(v)) = \ell_2$.

Mergeable trees.

Following the idea in [32], the barcode forest can be implemented using the *mergeable trees* data structure by Georgiadis et al. [61]. Since the maximum number of nodes in a barcode forest is $O(m)$, the data structure supports the following operations, each of which takes $O(\log m)$ amortized time:

- Return the root of a node.
- Return the nearest common ancestor of two leaves (in the same tree).
- Glue the paths from two leaves (in the same tree) to their nearest common ancestor.

Note that while we delete the path from the departing node to its ancestor in the departure event, deletions are not supported by mergeable trees. However, path deletions are indeed unnecessary which are only meant for a clear exposition. Hence, during implementation, we only traverse each ancestor of the departing node until an *unpaired*[3] one is found without actual deletions. Since each node can only be traversed once, the traversal in the departure events takes $O(m)$ time in total. See [61, Section 5] for details of implementing the barcode forest and its operations using mergeable trees.

Complexity.

The time complexity of the algorithm is $O(m \log^2 n + m \log m)$ dominated by the operations of the dynamic connectivity and the mergeable trees data structures.

4.3.2 Justification

In this subsection, we justify the correctness of Algorithm 4.3.1. For each entering node u in a T_{i} of Algorithm 4.3.1, there must be a single node in $\mathbb{G}_{\mathrm{B}}(\mathcal{F})$ at the level of u with the same property. So we also have entering nodes in $\mathbb{G}_{\mathrm{B}}(\mathcal{F})$. Splitting and departing nodes in $\mathbb{G}_{\mathrm{B}}(\mathcal{F})$ can be similarly defined.

We first prepare some standard notions and facts in zigzag persistence (Definition 4.3.2 and 4.3.3, Proposition 4.3.1) that help with our proofs. Some notions also appear in previous works in different forms; see, e.g., [21].

[3]↑An entering or splitting node is initially *unpaired* when introduced and becomes *paired* when its level is used to produce an interval. E.g., the node v becomes paired in the departure event in Algorithm 4.3.1.

Definition 4.3.2 (Representatives)**.** *Let* $\mathcal{M} : V_0 \xleftrightarrow{\psi_0} \cdots \xleftrightarrow{\psi_{m-1}} V_m$ *be an elementary zigzag module and* $[s,t] \subseteq [1,m]$ *be an interval. An indexed set* $\{\alpha_\mathrm{i} \in V_\mathrm{i} \mid \mathrm{i} \in [s,t]\}$ *is called a set of* partial representatives *for* $[s,t]$ *if for every* $\mathrm{i} \in [s,t-1]$*,* $\alpha_\mathrm{i} \mapsto \alpha_{\mathrm{i}+1}$ *or* $\alpha_\mathrm{i} \leftarrow\!\shortmid \alpha_{\mathrm{i}+1}$ *by* ψ_i*; it is called a set of* representatives *for* $[s,t]$ *if the following additional conditions are satisfied:*

1. *If* $\psi_{s-1} : V_{s-1} \to V_s$ *is forward with non-trivial cokernel, then* α_s *is not in* $\mathsf{img}(\psi_{s-1})$*; if* $\psi_{s-1} : V_{s-1} \leftarrow V_s$ *is backward with non-trivial kernel, then* α_s *is the non-zero element in* $\mathsf{ker}(\psi_{s-1})$*.*

2. *If* $t < m$ *and* $\psi_t : V_t \leftarrow V_{t+1}$ *is backward with non-trivial cokernel, then* α_t *is not in* $\mathsf{img}(\psi_t)$*; if* $t < m$ *and* $\psi_t : V_t \to V_{t+1}$ *is forward with non-trivial kernel, then* α_t *is the non-zero element in* $\mathsf{ker}(\psi_t)$*.*

Specifically, when $\mathcal{M} := \mathsf{H}_p(\mathcal{F})$ *for a zigzag filtration* $\mathcal{F}$*, we use terms* p-representatives *and* partial p-representatives *to emphasize the dimension* p*.*

Remark 4.3.3. Notice the connection of the above definition with Definition 3.5.1.

Remark 4.3.4. Let $\mathcal{F}$ be the filtration given in Figure 4.2a, and let α_8, α_9 be the sum of the component containing vertex 1 and the component containing vertex 2 in G_8 and G_9. Then, $\{\alpha_8, \alpha_9\}$ is a set of 0-representatives for the interval $[8,9] \in \mathsf{Pers}(\mathsf{H}_0(\mathcal{F}))$.

Definition 4.3.3 (Positive/negative indices)**.** *Let* $\mathcal{M} : V_0 \xleftrightarrow{\psi_0} \cdots \xleftrightarrow{\psi_{m-1}} V_m$ *be an elementary zigzag module. The set of* positive indices *of* $\mathcal{M}$*, denoted* $\mathsf{P}(\mathcal{M})$*, and the set of* negative indices *of* $\mathcal{M}$*, denoted* $\mathsf{N}(\mathcal{M})$*, are constructed as follows: for each forward* $\psi_\mathrm{i} : V_\mathrm{i} \to V_{\mathrm{i}+1}$*, if* ψ_i *is an injection with non-trivial cokernel, add* $\mathrm{i}+1$ *to* $\mathsf{P}(\mathcal{M})$*; if* ψ_i *is a surjection with non-trivial kernel, add* i *to* $\mathsf{N}(\mathcal{M})$*. Furthermore, for each backward* $\psi_\mathrm{i} : V_\mathrm{i} \leftarrow V_{\mathrm{i}+1}$*, if* ψ_i *is an injection with non-trivial cokernel, add* i *to* $\mathsf{N}(\mathcal{M})$*; if* ψ_i *is a surjection with non-trivial kernel, add* $\mathrm{i}+1$ *to* $\mathsf{P}(\mathcal{M})$*. Finally, add* $\mathsf{rank}\, V_m$ *copies of* m *to* $\mathsf{N}(\mathcal{M})$*.*

Remark 4.3.5. For each $\psi_\mathrm{i} : V_\mathrm{i} \leftrightarrow V_{\mathrm{i}+1}$ in Definition 4.3.3, if $\mathrm{i}+1 \in \mathsf{P}(\mathcal{M})$, then $\mathrm{i} \notin \mathsf{N}(\mathcal{M})$; similarly, if $\mathrm{i} \in \mathsf{N}(\mathcal{M})$, then $\mathrm{i}+1 \notin \mathsf{P}(\mathcal{M})$. Furthermore, if ψ_i is an isomorphism, then $\mathrm{i} \notin \mathsf{N}(\mathcal{M})$ and $\mathrm{i}+1 \notin \mathsf{P}(\mathcal{M})$.

Note that $\mathsf{N}(\mathcal{M})$ in Definition 4.3.3 is in fact a multi-set; calling it a set should not cause any confusion though. Also note that $|\mathsf{P}(\mathcal{M})| = |\mathsf{N}(\mathcal{M})|$, and every index in $\mathsf{P}(\mathcal{M})$ (resp. $\mathsf{N}(\mathcal{M})$) is the start (resp. end) of an interval in $\mathsf{Pers}(\mathcal{M})$. This explains why we add $\mathsf{rank}\, V_m$ copies of m to $\mathsf{N}(\mathcal{M})$ because there are always $\mathsf{rank}\, V_m$ number of intervals ending with m in $\mathsf{Pers}(\mathcal{M})$; see the example in Figure 4.2a where $\mathsf{rank}\, \mathsf{H}_0(G_{10}) = 2$.

Proposition 4.3.1. *Let $\mathcal{M}$ be an elementary zigzag module and $\pi : \mathsf{P}(\mathcal{M}) \to \mathsf{N}(\mathcal{M})$ be a bijection. If every $b \in \mathsf{P}(\mathcal{M})$ satisfies that $b \le \pi(b)$ and the interval $[b, \pi(b)]$ has a set of representatives, then $\mathsf{Pers}(\mathcal{M}) = \{[b, \pi(b)] \mid b \in \mathsf{P}(\mathcal{M})\}$.*

Proof. For each $b \in \mathsf{P}(\mathcal{M})$, let $\left\{\alpha^b_{\mathrm{j}} \mid \mathrm{j} \in [b, \pi(b)]\right\}$ be a set of representatives for $[b, \pi(b)]$. Then, define $\mathcal{I}^{[b,\pi(b)]}$ as an interval submodule of $\mathcal{M}$ over $[b, \pi(b)]$ such that $\mathcal{I}^{[b,\pi(b)]}(\mathrm{j})$ is generated by α^b_{j} if $\mathrm{j} \in [b, \pi(b)]$ and is trivial otherwise, where $\mathcal{I}^{[b,\pi(b)]}(\mathrm{j})$ denotes the j-th vector space in $\mathcal{I}^{[b,\pi(b)]}$. We claim that $\mathcal{M} = \bigoplus_{b\in\mathsf{P}(\mathcal{M})} \mathcal{I}^{[b,\pi(b)]}$, which implies the proposition. To prove this, suppose that $\mathcal{M}$ is of the form

$$\mathcal{M} : V_0 \xleftrightarrow{\psi_0} V_1 \xleftrightarrow{\psi_1} \cdots \xleftrightarrow{\psi_{m-1}} V_m$$

Then, we only need to verify that for every $\mathrm{i} \in [0, m]$, the set $\left\{\alpha^b_{\mathrm{i}} \mid b \in \mathsf{P}(\mathcal{M}) \text{ and } [b, \pi(b)] \ni \mathrm{i}\right\}$ is a basis of V_{i}. We prove this by induction on i. For $\mathrm{i} = 0$, since $V_0 = 0$, $\left\{\alpha^b_0 \mid b \in \mathsf{P}(\mathcal{M}) \text{ and } [b, \pi(b)] \ni 0\right\} = \varnothing$ is obviously a basis. So we can assume that for an $\mathrm{i} \in [0, m-1]$, $\left\{\alpha^b_{\mathrm{i}} \mid b \in \mathsf{P}(\mathcal{M}) \text{ and } [b, \pi(b)] \ni \mathrm{i}\right\}$ is a basis of V_{i}. We have the following cases:

ψ_{i} **an isomorphism:** In this case, $\mathrm{i} \notin \mathsf{N}(\mathcal{M})$ and $\mathrm{i}+1 \notin \mathsf{P}(\mathcal{M})$. If $\psi_{\mathrm{i}} : V_{\mathrm{i}} \to V_{\mathrm{i}+1}$ is forward, then $\left\{\alpha^b_{\mathrm{i}+1} \mid b \in \mathsf{P}(\mathcal{M}) \text{ and } [b, \pi(b)] \ni \mathrm{i}+1\right\} = \left\{\psi_{\mathrm{i}}(\alpha^b_{\mathrm{i}}) \mid b \in \mathsf{P}(\mathcal{M}) \text{ and } [b, \pi(b)] \ni \mathrm{i}\right\}$. The elements in $\left\{\alpha^b_{\mathrm{i}+1} \mid b \in \mathsf{P}(\mathcal{M}) \text{ and } [b, \pi(b)] \ni \mathrm{i}+1\right\}$ must then form a basis of $V_{\mathrm{i}+1}$ because ψ_{i} is an isomorphism. The verification for ψ_{i} being backward is similar.

$\psi_{\mathrm{i}} : V_{\mathrm{i}} \to V_{\mathrm{i}+1}$ **forward,** $\mathsf{coker}(\psi_{\mathrm{i}})$ **non-trivial:** In this case, $\mathrm{i} \notin \mathsf{N}(\mathcal{M})$ and $\mathrm{i}+1 \in \mathsf{P}(\mathcal{M})$. For each $b \in \mathsf{P}(\mathcal{M})$ such that $[b, \pi(b)] \ni \mathrm{i}$, $[b, \pi(b)] \ni \mathrm{i}+1$ and $\alpha^b_{\mathrm{i}} \mapsto \alpha^b_{\mathrm{i}+1}$ by ψ_{i}. We then have that elements in $\left\{\alpha^b_{\mathrm{i}+1} = \psi_{\mathrm{i}}(\alpha^b_{\mathrm{i}}) \mid b \in \mathsf{P}(\mathcal{M}) \text{ and } [b, \pi(b)] \ni \mathrm{i}\right\}$ are linearly independent because ψ_{i} is injective. Since $\alpha^{\mathrm{i}+1}_{\mathrm{i}+1} \notin \mathsf{img}(\psi_{\mathrm{i}})$ by Definition 4.3.2, $\left\{\alpha^b_{\mathrm{i}+1} \mid b \in \mathsf{P}(\mathcal{M}) \text{ and } [b, \pi(b)] \ni \mathrm{i}+1\right\} = \left\{\alpha^b_{\mathrm{i}+1} \mid b \in \mathsf{P}(\mathcal{M}) \text{ and } [b, \pi(b)] \ni \mathrm{i}\right\} \cup \left\{\alpha^{\mathrm{i}+1}_{\mathrm{i}+1}\right\}$

must contain linearly independent elements. The fact that the cardinality of the set equals $\mathsf{rank}\, V_{\mathrm{i}+1}$ implies that it must form a basis of $V_{\mathrm{i}+1}$.

$\psi_{\mathrm{i}} : V_{\mathrm{i}} \to V_{\mathrm{i}+1}$ **forward,** $\mathsf{ker}(\psi_{\mathrm{i}})$ **non-trivial:** In this case, $\mathrm{i} \in \mathsf{N}(\mathcal{M})$ and $\mathrm{i}+1 \notin \mathsf{P}(\mathcal{M})$. Let $\mathrm{j} = \boldsymbol{\pi}^{-1}(\mathrm{i})$. For each $b \in \mathsf{P}(\mathcal{M})$ such that $[b, \boldsymbol{\pi}(b)] \ni \mathrm{i}$ and $b \neq \mathrm{j}$, $[b, \boldsymbol{\pi}(b)] \ni \mathrm{i}+1$ and $\alpha^b_{\mathrm{i}} \mapsto \alpha^b_{\mathrm{i}+1}$ by ψ_{i}. We then have that $\left\{\alpha^b_{\mathrm{i}+1} \mid b \in \mathsf{P}(\mathcal{M}) \text{ and } [b, \boldsymbol{\pi}(b)] \ni \mathrm{i}+1\right\} = \left\{\psi_{\mathrm{i}}\left(\alpha^b_{\mathrm{i}}\right) \mid b \in \mathsf{P}(\mathcal{M}) \text{ and } [b, \boldsymbol{\pi}(b)] \ni \mathrm{i}\right\} \setminus \left\{\psi_{\mathrm{i}}\left(\alpha^{\mathrm{j}}_{\mathrm{i}}\right)\right\}$. Since ψ_{i} is surjective, elements in $\left\{\psi_{\mathrm{i}}\left(\alpha^b_{\mathrm{i}}\right) \mid b \in \mathsf{P}(\mathcal{M}) \text{ and } [b, \boldsymbol{\pi}(b)] \ni \mathrm{i}\right\}$ generate $V_{\mathrm{i}+1}$, in which $\psi_{\mathrm{i}}\left(\alpha^{\mathrm{j}}_{\mathrm{i}}\right) = 0$ by Definition 4.3.2. It follows that $\left\{\alpha^b_{\mathrm{i}+1} \mid b \in \mathsf{P}(\mathcal{M}) \text{ and } [b, \boldsymbol{\pi}(b)] \ni \mathrm{i}+1\right\}$ forms a basis of $V_{\mathrm{i}+1}$ because it generates $V_{\mathrm{i}+1}$ and its cardinality equals $\mathsf{rank}\, V_{\mathrm{i}+1}$.

$\psi_{\mathrm{i}} : V_{\mathrm{i}} \leftarrow V_{\mathrm{i}+1}$ **backward,** $\mathsf{coker}(\psi_{\mathrm{i}})$ **non-trivial:** In this case, $\mathrm{i} \in \mathsf{N}(\mathcal{M})$ and $\mathrm{i}+1 \notin \mathsf{P}(\mathcal{M})$. For each $b \in \mathsf{P}(\mathcal{M})$ such that $[b, \boldsymbol{\pi}(b)] \ni \mathrm{i}$ and $\boldsymbol{\pi}(b) \neq \mathrm{i}$, $[b, \boldsymbol{\pi}(b)] \ni \mathrm{i}+1$ and $\alpha^b_{\mathrm{i}} \mapsfrom \alpha^b_{\mathrm{i}+1}$ by ψ_{i}. We then have that elements in $\left\{\alpha^b_{\mathrm{i}+1} = (\psi_{\mathrm{i}})^{-1}(\alpha^b_{\mathrm{i}}) \mid b \in \mathsf{P}(\mathcal{M}),\ [b, \boldsymbol{\pi}(b)] \ni \mathrm{i}, \text{ and } \boldsymbol{\pi}(b) \neq \mathrm{i}\right\}$ are linearly independent because if they are not, then their images under ψ_{i} are also not, which is a contradiction. Note that $\left\{\alpha^b_{\mathrm{i}+1} \mid b \in \mathsf{P}(\mathcal{M}), [b, \boldsymbol{\pi}(b)] \ni \mathrm{i}+1\right\} = \left\{\alpha^b_{\mathrm{i}+1} \mid b \in \mathsf{P}(\mathcal{M}),\ [b, \boldsymbol{\pi}(b)] \ni \mathrm{i}, \text{ and } \boldsymbol{\pi}(b) \neq \mathrm{i}\right\}$ and its cardinality equals $\mathsf{rank}\, V_{\mathrm{i}+1}$, so it must form a basis of $V_{\mathrm{i}+1}$.

$\psi_{\mathrm{i}} : V_{\mathrm{i}} \leftarrow V_{\mathrm{i}+1}$ **backward,** $\mathsf{ker}(\psi_{\mathrm{i}})$ **non-trivial:** In this case, $\mathrm{i} \notin \mathsf{N}(\mathcal{M})$ and $\mathrm{i}+1 \in \mathsf{P}(\mathcal{M})$. For each $b \in \mathsf{P}(\mathcal{M})$ such that $[b, \boldsymbol{\pi}(b)] \ni \mathrm{i}$, $[b, \boldsymbol{\pi}(b)] \ni \mathrm{i}+1$ and $\alpha^b_{\mathrm{i}} \mapsfrom \alpha^b_{\mathrm{i}+1}$ by ψ_{i}. We then have that elements in $\left\{\alpha^b_{\mathrm{i}+1} \in (\psi_{\mathrm{i}})^{-1}(\alpha^b_{\mathrm{i}}) \mid b \in \mathsf{P}(\mathcal{M}) \text{ and } [b, \boldsymbol{\pi}(b)] \ni \mathrm{i}\right\}$ are linearly independent because their images under ψ_{i} are. We also have that there is no non-trivial linear combination of $\left\{\alpha^b_{\mathrm{i}+1} \mid b \in \mathsf{P}(\mathcal{M}) \text{ and } [b, \boldsymbol{\pi}(b)] \ni \mathrm{i}\right\}$ falling in $\mathsf{ker}(\psi_{\mathrm{i}})$ because otherwise their images under ψ_{i} would not be linearly independent. Since $\alpha^{\mathrm{i}+1}_{\mathrm{i}+1}$ is the non-zero element in $\mathsf{ker}(\psi_{\mathrm{i}})$ by Definition 4.3.2, we have that $\left\{\alpha^b_{\mathrm{i}+1} \mid b \in \mathsf{P}(\mathcal{M}) \text{ and } [b, \boldsymbol{\pi}(b)] \ni \mathrm{i}+1\right\} = \left\{\alpha^b_{\mathrm{i}+1} \mid b \in \mathsf{P}(\mathcal{M}) \text{ and } [b, \boldsymbol{\pi}(b)] \ni \mathrm{i}\right\} \cup \left\{\alpha^{\mathrm{i}+1}_{\mathrm{i}+1}\right\}$ contains linearly independent elements. Then, it must form a basis of $V_{\mathrm{i}+1}$ because its cardinality equals $\mathsf{rank}\, V_{\mathrm{i}+1}$, □

Now we present several propositions leading to our conclusion (Theorem 4.3.1). Specifically, Proposition 4.3.2 states that a certain path in $\mathbb{G}_{\mathrm{B}}(\mathcal{F})$ induces a set of partial 0-

representatives. Proposition 4.3.3 lists some invariants of Algorithm 4.3.1. Proposition 4.3.2 and 4.3.3 support the proof of Proposition 4.3.4, which together with Proposition 4.3.1 implies Theorem 4.3.1.

From now on, G and $\mathcal{F}$ always denote the input to Algorithm 4.3.1. Since each node in a barcode graph represents a connected component, we also interpret nodes in a barcode graph as 0-th homology classes throughout the chapter. Moreover, a path in a barcode graph from a node v to a node u is said to be *within level* j *and* i if for each node on the path, its level ℓ satisfies $\mathrm{j} \leq \ell \leq \mathrm{i}$; we denote such a path as $(v \rightsquigarrow u)_{[\mathrm{j},\mathrm{i}]}$.

Proposition 4.3.2. *Let v be a level-j node and u be a level-i node in $\mathbb{G}_{\mathrm{B}}(\mathcal{F})$ such that $\mathrm{j} < \mathrm{i}$ and there is a path $(v \rightsquigarrow u)_{[\mathrm{j},\mathrm{i}]}$ in $\mathbb{G}_{\mathrm{B}}(\mathcal{F})$. Then, there is a set of partial 0-representatives $\{\alpha_k \in \mathsf{H}_0(G_k) \mid k \in [\mathrm{j},\mathrm{i}]\}$ for the interval $[\mathrm{j},\mathrm{i}]$ with $\alpha_{\mathrm{j}} = v$ and $\alpha_{\mathrm{i}} = u$.*

Proof. We can assume that $(v \rightsquigarrow u)_{[\mathrm{j},\mathrm{i}]}$ is a simple path because if it were not we could always find one. For each $k \in [\mathrm{j}+1, \mathrm{i}-1]$, let $w_1, \ldots, w_r$ be all the level-k nodes on $(v \rightsquigarrow u)_{[\mathrm{j},\mathrm{i}]}$ whose adjacent nodes on $(v \rightsquigarrow u)_{[\mathrm{j},\mathrm{i}]}$ are at different levels. Then, let $\alpha_k = \sum_{\ell=1}^{r} w_\ell$. Also, let $\alpha_{\mathrm{j}} = v$ and $\alpha_{\mathrm{i}} = u$. It can be verified that $\{\alpha_k \mid k \in [\mathrm{j},\mathrm{i}]\}$ is a set of partial 0-representatives for $[\mathrm{j},\mathrm{i}]$. See Figure 4.4 for an example of a simple path $(\tilde{v}_2 \rightsquigarrow u_2)_{[10,13]}$ (the dashed one) in a barcode graph, where the solid nodes contribute to the induced partial 0-representatives and the hollow nodes are excluded. ☐

For an i with $0 \leq \mathrm{i} \leq m$, we define the *prefix* $\mathcal{F}^{\mathrm{i}}$ of $\mathcal{F}$ as the filtration $\mathcal{F}^{\mathrm{i}} : G_0 \leftrightarrow \cdots \leftrightarrow G_{\mathrm{i}}$ and observe that $\mathbb{G}_{\mathrm{B}}(\mathcal{F}^{\mathrm{i}})$ is the subgraph of $\mathbb{G}_{\mathrm{B}}(\mathcal{F})$ induced by nodes at levels less than or equal to i. We call level-i nodes of $\mathbb{G}_{\mathrm{B}}(\mathcal{F}^{\mathrm{i}})$ as *leaves* and do not distinguish leaves in T_{i} and $\mathbb{G}_{\mathrm{B}}(\mathcal{F}^{\mathrm{i}})$ because they bijectively map to each other. It should be clear from the context though which graph or forest a particular leaf is in.

Proposition 4.3.3. *For each $\mathrm{i} = 0, \ldots, m$, Algorithm 4.3.1 maintains the following invariants:*

1. *There is a bijection η from trees in T_{i} to connected components in $\mathbb{G}_{\mathrm{B}}(\mathcal{F}^{\mathrm{i}})$ containing leaves such that a leaf u is in a tree Υ of T_{i} if and only if u is in $\eta(\Upsilon)$.*

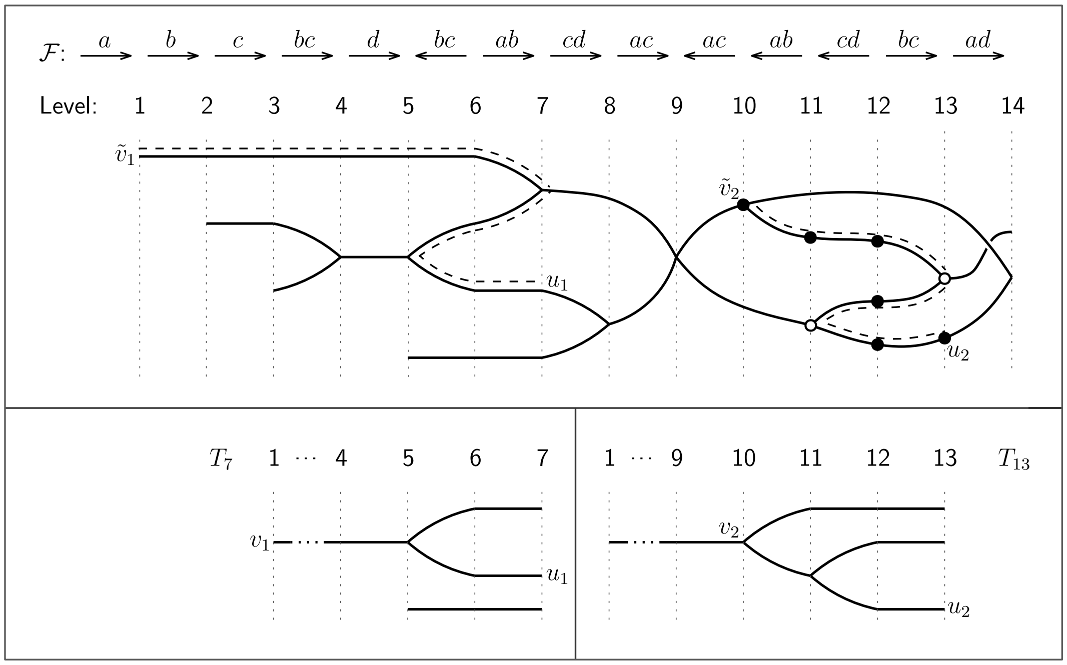

$\mathcal{F}$: a b c bc d bc ab cd ac ac ab cd bc ad
Level: 1 2 3 4 5 6 7 8 9 10 11 12 13 14
$\tilde{v}_1$
$\tilde{v}_2$
u_1
u_2
T_7 1 ⋯ 4 5 6 7
v_1
u_1
1 ⋯ 9 10 11 12 13 T_{13}
v_2
u_2

Figure 4.4. Illustration of invariants of Proposition 4.3.3. The top part contains a barcode graph with its filtration given (a, b, c, and d are vertices of the complex). The bottom contains two barcode forests.

2. *For each leaf u in T_i and each ancestor of u at a level j, there is a path $(\tilde{v} \rightsquigarrow u)_{[\mathrm{j},\mathrm{i}]}$ in $\mathbb{G}_\mathrm{B}(\mathcal{F})$ where $\tilde{v}$ is a level-j node.*

3. *For each leaf u in T_i and each splitting ancestor of u at a level j, let $\tilde{v}$ be the unique level-j splitting node in $\mathbb{G}_\mathrm{B}(\mathcal{F})$. Then, there is a path $(\tilde{v} \rightsquigarrow u)_{[\mathrm{j},\mathrm{i}]}$ in $\mathbb{G}_\mathrm{B}(\mathcal{F})$.*

Remark 4.3.6. See Figure 4.4 for examples of invariant 2 and 3. In the figure, v_1 is a level-1 non-splitting ancestor of u_1 in T_7 and $\tilde{v}_1$ is a level-1 node in the barcode graph; v_2 is a level-10 splitting ancestor of u_2 in T_{13} and $\tilde{v}_2$ is the unique level-10 splitting node in the barcode graph. The paths $(\tilde{v}_1 \rightsquigarrow u_1)_{[1,7]}$ and $(\tilde{v}_2 \rightsquigarrow u_2)_{[10,13]}$ are marked with dashes.

Proof. We only verify invariant 3 as the verification for invariant 2 is similar but easier and invariant 1 is straightforward. The verification is by induction. When $\mathrm{i} = 0$, invariant 3 trivially holds. Now suppose that invariant 3 is true for an $\mathrm{i} \in [0, m-1]$. For the no-change, entrance, and split event in Algorithm 4.3.1, it is not hard to see that invariant 3 still holds for $\mathrm{i}+1$. For the departure event, because we are only deleting a path from T_i to form $T_{\mathrm{i}+1}$, invariant 3 also holds for $\mathrm{i}+1$. For the merge event, let u be a leaf in $T_{\mathrm{i}+1}$, v be a splitting ancestor of u at a level j, and $\tilde{v}$ be the unique splitting node in $\mathbb{G}_{\mathrm{B}}(\mathcal{F})$ at level j. The node v may correspond to one or two nodes in T_{i}, in which only one is splitting, and let v' be the splitting one. Note that u's parent may correspond to one or two nodes in T_{i}, and we let W be the set of nodes in T_{i} that u's parent corresponds to. If v' is an ancestor of a node $w \in W$ in T_{i}, then by the assumption, there must be a path $(\tilde{v} \rightsquigarrow w)_{[\mathrm{j},\mathrm{i}]}$ in $\mathbb{G}_{\mathrm{B}}(\mathcal{F})$. From this path, we can derive a path $(\tilde{v} \rightsquigarrow u)_{[\mathrm{j},\mathrm{i}+1]}$ in $\mathbb{G}_{\mathrm{B}}(\mathcal{F})$. If v' is not an ancestor of any node of W in T_{i}, the fact that v is an ancestor of u's parent in $T_{\mathrm{i}+1}$ implies that there must be an ancestor v'' of a node $w \in W$ in T_{i} which v corresponds to. So we have that v is a gluing of two nodes from T_{i}. Note that u's parent must not be a glued node in $T_{\mathrm{i}+1}$ because otherwise v' would have been an ancestor of a node of W in T_{i}; see Figure 4.5 where z_1 and z_2 are the two level-i nodes glued together. Let x be the highest one among the nodes on the path from v to u that are glued in iteration i. We have that x must correspond to a node x' in T_{i} which is an ancestor of w. Recall that z_1, z_2 are the two leaves in T_{i} which are glued, and let z_3 be the child of the glued node of z_1, z_2 in $T_{\mathrm{i}+1}$, as shown in Figure 4.5. From the figure, we have that x' must be splitting because one child of x' (which is not glued) descends down to w and the other child of x' (which is glued) descends down to z_1. The fact that v' is an ancestor of z_2 in T_{i} implies that there is a path $(\tilde{v} \rightsquigarrow z_2)_{[\mathrm{j},\mathrm{i}]}$ in $\mathbb{G}_{\mathrm{B}}(\mathcal{F})$. Let $\tilde{x}$ be the unique splitting node in $\mathbb{G}_{\mathrm{B}}(\mathcal{F})$ at the same level with x'; then, z_1 and w being descendants of x' in T_{i} implies that there are paths $(z_1 \rightsquigarrow \tilde{x})_{[\mathrm{j},\mathrm{i}]}$ and $(\tilde{x} \rightsquigarrow w)_{[\mathrm{j},\mathrm{i}]}$ in $\mathbb{G}_{\mathrm{B}}(\mathcal{F})$. Now we derive a path $(\tilde{v} \rightsquigarrow u)_{[\mathrm{j},\mathrm{i}+1]}$ in $\mathbb{G}_{\mathrm{B}}(\mathcal{F})$ by concatenating the following paths and edges: $(\tilde{v} \rightsquigarrow z_2)_{[\mathrm{j},\mathrm{i}]}$, $\overline{z_2 z_3}$, $\overline{z_3 z_1}$, $(z_1 \rightsquigarrow \tilde{x})_{[\mathrm{j},\mathrm{i}]}$, $(\tilde{x} \rightsquigarrow w)_{[\mathrm{j},\mathrm{i}]}$, $\overline{wu}$. □

Proposition 4.3.4. *Each interval produced by Algorithm 4.3.1 admits a set of 0-representatives.*

Proof. Suppose that an interval is produced by the merge event at iteration i. We have the following situations:

- If the nodes u_1, u_2 in this event (see Algorithm 4.3.1) are in the same tree in T_{i}, let v be the highest common ancestor of u_1, u_2 and note that v is a splitting node at level j. Also note that u_1, u_2 are actually leaves in T_{i} and hence can also be considered

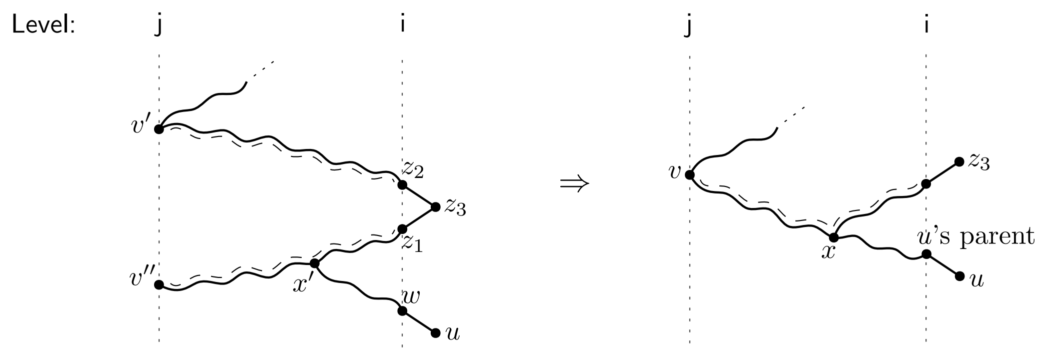


Figure 4.5. Illustration of parts of $T_{\mathrm{i}+1}$ for the proof of Proposition 4.3.3, where the left one is before the path gluing and the right one is after. Note that the part between level j and i for the left tree actually belongs to T_{i}. The paths with dashed marks are the glued ones (before and after), in which v' and v'' are identified as v, and x' is identified as x with another node.

as level-i nodes in $\mathbb{G}_{\mathrm{B}}(\mathcal{F})$. Let $\tilde{v}$ be the unique level-j splitting node in $\mathbb{G}_{\mathrm{B}}(\mathcal{F})$. By invariant 3 of Proposition 4.3.3 along with Proposition 4.3.2, there are two sets of partial 0-representatives $\{\alpha_k \mid k \in [\mathrm{j},\mathrm{i}]\}, \{\beta_k \mid k \in [\mathrm{j},\mathrm{i}]\}$ for $[\mathrm{j},\mathrm{i}]$ with $\alpha_{\mathrm{j}} = \tilde{v}$, $\alpha_{\mathrm{i}} = u_1$, $\beta_{\mathrm{j}} = \tilde{v}$, and $\beta_{\mathrm{i}} = u_2$. We claim that $\{\alpha_k + \beta_k \mid k \in [\mathrm{j}+1,\mathrm{i}]\}$ is a set of 0-representatives for the interval $[\mathrm{j}+1,\mathrm{i}]$. To prove this, we first note the following obvious facts: (i) $\{\alpha_k + \beta_k \mid k \in [\mathrm{j}+1,\mathrm{i}]\}$ is a set of partial 0-representatives; (ii) $\alpha_{\mathrm{j}+1} + \beta_{\mathrm{j}+1} \in \mathsf{ker}(\varphi^0_{\mathrm{j}})$; (iii) $\alpha_{\mathrm{i}} + \beta_{\mathrm{i}}$ is the non-zero element in $\mathsf{ker}(\varphi^0_{\mathrm{i}})$. So we only need to show that $\alpha_{\mathrm{j}+1} + \beta_{\mathrm{j}+1} \neq 0$. Let v_1, v_2 be the two level-$(\mathrm{j}+1)$ nodes in $\mathbb{G}_{\mathrm{B}}(\mathcal{F})$ connecting to $\tilde{v}$. Then, $\alpha_{\mathrm{j}+1}$ equals v_1 or v_2 and the same for $\beta_{\mathrm{j}+1}$. To see this, we first show that $\alpha_{\mathrm{j}+1}$ can only contain v_1, v_2. For contradiction, suppose instead that $\alpha_{\mathrm{j}+1}$ contains a level-$(\mathrm{j}+1)$ node x with $x \neq v_1$, $x \neq v_2$. Let $(\tilde{v} \rightsquigarrow u_1)_{[\mathrm{j},\mathrm{i}]}$ be the simple path that induces $\{\alpha_k \mid k \in [\mathrm{j},\mathrm{i}]\}$ as in Proposition 4.3.2 and its proof. Then, x is on the path $(\tilde{v} \rightsquigarrow u_1)_{[\mathrm{j},\mathrm{i}]}$ and the two adjacent nodes of x on $(\tilde{v} \rightsquigarrow u_1)_{[\mathrm{j},\mathrm{i}]}$ are at level j and $\mathrm{j}+2$, in which we let y be the one at level j. Note that $y \neq \tilde{v}$ because x is not equal to v_1 or v_2. Since $(\tilde{v} \rightsquigarrow u_1)_{[\mathrm{j},\mathrm{i}]}$ is within level j and i, y must be adjacent to another level-$(\mathrm{j}+1)$ node distinct from x on $(\tilde{v} \rightsquigarrow u_1)_{[\mathrm{j},\mathrm{i}]}$. Now we have that y is a level-j splitting node with $y \neq \tilde{v}$, contradicting the fact that $\mathbb{G}_{\mathrm{B}}(\mathcal{F})$ has only one level-j splitting node. The fact that $\alpha_{\mathrm{j}+1}$ contains v_1 or v_2 but not both can be similarly verified. To see that

$\alpha_{\mathrm{j}+1}+\beta_{\mathrm{j}+1}\neq 0$, suppose instead that $\alpha_{\mathrm{j}+1}+\beta_{\mathrm{j}+1}=0$, i.e., $\alpha_{\mathrm{j}+1}=\beta_{\mathrm{j}+1}$, and without loss of generality they both equal v_1. Note that we can consider T_{i} as derived by contracting nodes of $\mathbb{G}_{\mathrm{B}}(\mathcal{F}^{\mathrm{i}})$ at the same level[4]. The fact that $\alpha_{\mathrm{j}+1}=\beta_{\mathrm{j}+1}=v_1$ implies that u_1, u_2 are descendants of the same child of v in T_{i}, contradicting the fact that v is the highest common ancestor of u_1, u_2. So we have that $\alpha_{\mathrm{j}+1}+\beta_{\mathrm{j}+1}\neq 0$.

- If u_1, u_2 are in different trees in T_{i}, then without loss of generality let u_1 be the one whose root v_1 is at the higher level (i.e., level j). As the root of u_1, the node v_1 must be an entering node, and the connected component of $\mathbb{G}_{\mathrm{B}}(\mathcal{F}^{\mathrm{i}})$ containing u_1 must have a single level-j node $\tilde{v}_1$. Then, by invariant 2 of Proposition 4.3.3 along with Proposition 4.3.2, there are two sets of partial 0-representatives $\{\alpha_k \mid k\in[\mathrm{j},\mathrm{i}]\}, \{\beta_k \mid k\in[\mathrm{j},\mathrm{i}]\}$ for $[\mathrm{j},\mathrm{i}]$ with $\alpha_{\mathrm{j}}=\tilde{v}_1$, $\alpha_{\mathrm{i}}=u_1$, $\beta_{\mathrm{j}}=\tilde{v}_2$, and $\beta_{\mathrm{i}}=u_2$, where $\tilde{v}_2$ is a level-j node. We claim that $\{\alpha_k+\beta_k \mid k\in[\mathrm{j},\mathrm{i}]\}$ is a set of 0-representatives for the interval $[\mathrm{j},\mathrm{i}]$ and the verification is similar to the previous case where u_1 and u_2 are in the same tree.

For intervals produced by the departure events and at the end of the algorithm, the existence of 0-representatives can be similarly argued. ☐

Theorem 4.3.1. *Algorithm 4.3.1 computes the 0-th zigzag barcode for a given zigzag filtration.*

Proof. First, we have the following facts: every level-j entering node in $\mathbb{G}_{\mathrm{B}}(\mathcal{F})$ introduces a $\mathrm{j}\in\mathsf{P}(\mathsf{H}_0(\mathcal{F}))$ and uniquely corresponds to a level-j root in T_{i} for some i; every level-j splitting node in $\mathbb{G}_{\mathrm{B}}(\mathcal{F})$ introduces a $\mathrm{j}+1\in\mathsf{P}(\mathsf{H}_0(\mathcal{F}))$ and uniquely corresponds to a level-j splitting node in T_{i} for some i. Whenever an interval $[\mathrm{j},\mathrm{i}]$ is produced in Algorithm 4.3.1, $\mathrm{i}\in\mathsf{N}(\mathsf{H}_0(\mathcal{F}))$ and the entering or splitting node in T_{i} introducing j as a positive index either becomes a *regular* node (i.e., connecting to a single node on both adjacent levels) or is deleted in $T_{\mathrm{i}+1}$. This means that j is never the start of another interval produced. At the end of Algorithm 4.3.1, the number of intervals produced which end with m also matches the rank of $\mathsf{H}_0(G_m)$. Therefore, intervals produced by the algorithm induce a bijection $\pi:\mathsf{P}(\mathsf{H}_0(\mathcal{F}))\to\mathsf{N}(\mathsf{H}_0(\mathcal{F}))$. By Proposition 4.3.1 and 4.3.4, our conclusion follows. ☐

[4]↑ We should further note that this contraction is not done on the entire $\mathbb{G}_{\mathrm{B}}(\mathcal{F}^{\mathrm{i}})$ but rather on connected components of $\mathbb{G}_{\mathrm{B}}(\mathcal{F}^{\mathrm{i}})$ containing leaves.

4.4 One-dimensional zigzag persistence

In this section, we present an efficient algorithm for 1-st zigzag persistence on graphs. We assume that the input is a graph G with a zigzag filtration

$$\mathcal{F} : \varnothing = G_0 \xleftrightarrow{\sigma_0} G_1 \xleftrightarrow{\sigma_1} \cdots \xleftrightarrow{\sigma_{m-1}} G_m$$

of G. We first describe the algorithm without giving the full implementation details. The key to the algorithm is a pairing principle for the positive and negative indices. We then prove the correctness of the algorithm. Finally, in Section 4.4.1, we make several observations which reduce the index pairing to finding the *max edge-weight* of a path in a minimum spanning forest, leading to an efficient implementation.

We notice that the following are true for every inclusion $G_\mathrm{i} \xleftrightarrow{\sigma_\mathrm{i}} G_{\mathrm{i}+1}$ of $\mathcal{F}$ (recall that φ^1_i denotes the corresponding linear map in the induced module $\mathsf{H}_1(\mathcal{F})$):

- If σ_i is an edge being added and vertices of σ_i are connected in G_i, then φ^1_i is an injection with non-trivial cokernel, which provides $\mathrm{i}+1 \in \mathsf{P}(\mathsf{H}_1(\mathcal{F}))$.

- If σ_i is an edge being deleted and vertices of σ_i are connected in $G_{\mathrm{i}+1}$, then φ^1_i is an injection with non-trivial cokernel, which provides $\mathrm{i} \in \mathsf{N}(\mathsf{H}_1(\mathcal{F}))$.

- In all the other cases, φ^1_i is an isomorphism and $\mathrm{i} \notin \mathsf{N}(\mathsf{H}_1(\mathcal{F}))$, $\mathrm{i}+1 \notin \mathsf{P}(\mathsf{H}_1(\mathcal{F}))$.

As can be seen from Section 4.3, computing $\mathsf{Pers}(\mathsf{H}_1(\mathcal{F}))$ boils down to finding a pairing of indices of $\mathsf{P}(\mathsf{H}_1(\mathcal{F}))$ and $\mathsf{N}(\mathsf{H}_1(\mathcal{F}))$. Our algorithm presented in Algorithm 4.4.1 adopts this structure, where $\mathcal{U}_\mathrm{i}$ denotes the set of unpaired positive indices at the *beginning* of each iteration i.

Note that in Algorithm 4.4.1, whenever a positive or negative index is produced, σ_i must be an edge. One key piece missing from the algorithm is how we choose a positive index to pair with a negative index:

Pairing Principle for Algorithm 4.4.1. *In each iteration* i *where* $G_\mathrm{i} \xleftarrow{\sigma_\mathrm{i}} G_{\mathrm{i}+1}$ *provides* $\mathrm{i} \in \mathsf{N}(\mathsf{H}_1(\mathcal{F}))$*, let* J_i *consist of every* $\mathrm{j} \in \mathcal{U}_\mathrm{i}$ *such that there exists a 1-cycle* z *containing both*

Algorithm 4.4.1 Computing 1-st zigzag persistence on graphs

```
$\mathcal{U}_0 := \varnothing$
for $\mathrm{i} := 0, \ldots, m-1$:
    if $G_\mathrm{i} \xrightarrow{\sigma_\mathrm{i}} G_{\mathrm{i}+1}$ provides $\mathrm{i}+1 \in \mathsf{P}(\mathsf{H}_1(\mathcal{F}))$:
        $\mathcal{U}_{\mathrm{i}+1} := \mathcal{U}_\mathrm{i} \cup \{\mathrm{i}+1\}$
    else if $G_\mathrm{i} \xleftarrow{\sigma_\mathrm{i}} G_{\mathrm{i}+1}$ provides $\mathrm{i} \in \mathsf{N}(\mathsf{H}_1(\mathcal{F}))$:
        pair $\mathrm{i}$ with a $\mathrm{j}_* \in \mathcal{U}_\mathrm{i}$ based on the Pairing Principle below
        output an interval $[\mathrm{j}_*, \mathrm{i}]$ for $\mathsf{Pers}(\mathsf{H}_1(\mathcal{F}))$
        $\mathcal{U}_{\mathrm{i}+1} := \mathcal{U}_\mathrm{i} \setminus \{\mathrm{j}_*\}$
    else:
        $\mathcal{U}_{\mathrm{i}+1} := \mathcal{U}_\mathrm{i}$
for each $\mathrm{j} \in \mathcal{U}_m$:
    output an interval $[\mathrm{j}, m]$ for $\mathsf{Pers}(\mathsf{H}_1(\mathcal{F}))$
```

$\sigma_{\mathrm{j}-1}$ *and* σ_i *with* $z \subseteq G_k$ *for every* $k \in [\mathrm{j}, \mathrm{i}]$*. Then,* $J_\mathrm{i} \neq \varnothing$ *and Algorithm 4.4.1 pairs* i *with the* smallest *index* j_* *in* J_i.

Remark 4.4.1. See Proposition 4.4.1 for a proof of $J_\mathrm{i} \neq \varnothing$ claimed above.

Remark 4.4.2. Algorithms for non-zigzag persistence [3], [4] always pair a negative index with the *largest* (i.e., youngest) positive index satisfying a certain condition, while Algorithm 4.4.1 pairs with the smallest one. This is due to the difference of zigzag and non-zigzag persistence and our particular condition that 1-cycles can never become boundaries in graphs. See [21], [27] for the pairing when assuming general zigzag filtrations.

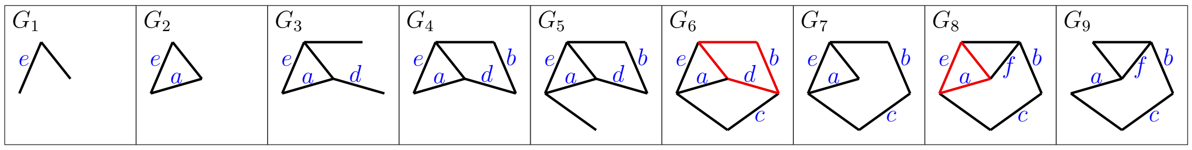


Figure 4.6. A zigzag filtration with 1-st barcode $\{[4,6],[2,8],[6,9],[8,9]\}$. For brevity, the addition of vertices and some edges are skipped.

Figure 4.6 gives an example of the pairing of the indices and their corresponding edges. In the figure, when edge d is deleted from G_6, there are three unpaired positive edges a, b, and c, in which b and c admit 1-cycles as required by the Pairing Principle. As the earlier edge, b is paired with d and an interval $[4, 6]$ is produced. The red cycle in G_6 indicates the 1-cycle containing b and d which exists in all the intermediate graphs. Similar situations happen when e is paired with a in G_8, producing the interval $[2, 8]$.

For the correctness of Algorithm 4.4.1, we first provide Proposition 4.4.1 which justifies the Pairing Principle and is a major step leading toward our conclusion (Theorem 4.4.1):

Proposition 4.4.1. *At the beginning of each iteration* i *in Algorithm 4.4.1, for every* $\mathrm{j} \in \mathcal{U}_\mathrm{i}$*, there exists a 1-cycle* z^i_j *containing* $\sigma_{\mathrm{j}-1}$ *with* $z^\mathrm{i}_\mathrm{j} \subseteq G_k$ *for every* $k \in [\mathrm{j}, \mathrm{i}]$*. Furthermore, the set* $\{z^\mathrm{i}_\mathrm{j} \mid \mathrm{j} \in \mathcal{U}_\mathrm{i}\}$ *forms a basis of* $\mathsf{Z}_1(G_\mathrm{i})$*. If the iteration* i *produces a negative index* i*, then the above statements imply that there is at least one* z^i_j *containing* σ_i*. This* z^i_j *satisfies the condition that* $z^\mathrm{i}_\mathrm{j} \subseteq G_k$ *for every* $k \in [\mathrm{j}, \mathrm{i}]$*,* $\sigma_{\mathrm{j}-1} \in z^\mathrm{i}_\mathrm{j}$*, and* $\sigma_\mathrm{i} \in z^\mathrm{i}_\mathrm{j}$*, which implies that* $J_\mathrm{i} \neq \varnothing$ *where* J_i *is as defined in the Pairing Principle.*

Proof. We prove this by induction. At the beginning of iteration 0, since $G_0 = \varnothing$ and $\mathcal{U}_0 = \varnothing$, the proposition is trivially true. Suppose that the proposition is true at the beginning of an iteration i. For each $\mathrm{j} \in \mathcal{U}_\mathrm{i}$, let z^i_j be the 1-cycle as specified in the proposition. If $G_\mathrm{i} \xleftrightarrow{\sigma_\mathrm{i}} G_{\mathrm{i}+1}$ produces neither a positive index nor a negative index, then $\mathsf{Z}_1(G_\mathrm{i}) = \mathsf{Z}_1(G_{\mathrm{i}+1})$ and $\mathcal{U}_\mathrm{i} = \mathcal{U}_{\mathrm{i}+1}$. Let $z^{\mathrm{i}+1}_\mathrm{j} = z^\mathrm{i}_\mathrm{j}$ for each j; then, $\{z^{\mathrm{i}+1}_\mathrm{j} \mid \mathrm{j} \in \mathcal{U}_{\mathrm{i}+1}\}$ serves as the 1-cycles as specified in the proposition for iteration $\mathrm{i}+1$. If $G_\mathrm{i} \xrightarrow{\sigma_\mathrm{i}} G_{\mathrm{i}+1}$ produces a new unpaired positive index $\mathrm{i}+1$, let $z^{\mathrm{i}+1}_{\mathrm{i}+1}$ be any 1-cycle in $G_{\mathrm{i}+1}$ containing σ_i. Also, for each $\mathrm{j} \in \mathcal{U}_\mathrm{i}$, let $z^{\mathrm{i}+1}_\mathrm{j} = z^\mathrm{i}_\mathrm{j}$. It can be verified that $\{z^{\mathrm{i}+1}_\mathrm{j} \mid \mathrm{j} \in \mathcal{U}_{\mathrm{i}+1}\}$ serves as the 1-cycles as specified in the proposition for iteration $\mathrm{i}+1$.

If $G_\mathrm{i} \xleftarrow{\sigma_\mathrm{i}} G_{\mathrm{i}+1}$ produces a negative index i, then there must be a 1-cycle in G_i containing σ_i. The fact that $\{z^\mathrm{i}_\mathrm{j} \mid \mathrm{j} \in \mathcal{U}_\mathrm{i}\}$ forms a basis of $\mathsf{Z}_1(G_\mathrm{i})$ implies that there must be at least one z^i_j containing σ_i because otherwise no combination of the z^i_j's can equal a cycle containing σ_i. Let $\bar{\mathrm{j}}$ be the smallest $\mathrm{j} \in \mathcal{U}_\mathrm{i}$ such that z^i_j contains σ_i. We claim that $\mathrm{j}_* = \bar{\mathrm{j}}$, where j_* is as defined in the Pairing Principle. For contradiction, suppose instead that $\mathrm{j}_* \neq \bar{\mathrm{j}}$. Note that $\bar{\mathrm{j}} \in J_\mathrm{i}$, where J_i is as defined in the Pairing Principle. Since j_* is the smallest index in

J_{i}, we have that $\mathrm{j}_* < \bar{\mathrm{j}}$. By the Pairing Principle, there exists a 1-cycle ζ containing both σ_{j_*-1} and σ_{i} with $\zeta \subseteq G_k$ for every $k \in [\mathrm{j}_*, \mathrm{i}]$. Since $\{z^{\mathrm{i}}_{\mathrm{j}} \mid \mathrm{j} \in \mathcal{U}_{\mathrm{i}}\}$ forms a basis of $\mathsf{Z}_1(G_{\mathrm{i}})$ and $\zeta \subseteq G_{\mathrm{i}}$, ζ must equal a sum $\sum_{\ell=1}^{s} z^{\mathrm{i}}_{\lambda_\ell}$, where each $\lambda_\ell \in \mathcal{U}_{\mathrm{i}}$. We rearrange the indices such that $\lambda_1 < \lambda_2 < \cdots < \lambda_s$. We have that $\lambda_s \geq \bar{\mathrm{j}}$ because otherwise each $\lambda_\ell < \bar{\mathrm{j}}$ and so its corresponding $z^{\mathrm{i}}_{\lambda_\ell}$ does not contain σ_{i}. This implies that $\zeta = \sum_{\ell=1}^{s} z^{\mathrm{i}}_{\lambda_\ell}$ does not contain σ_{i}, which is a contradiction. For each ℓ such that $1 \leq \ell < s$, since $\lambda_\ell \leq \lambda_s - 1 \leq \mathrm{i}$, we have that $z^{\mathrm{i}}_{\lambda_\ell} \subseteq G_{\lambda_s-1}$, which means that $\sigma_{\lambda_s-1} \notin z^{\mathrm{i}}_{\lambda_\ell}$ because $\sigma_{\lambda_s-1} \notin G_{\lambda_s-1}$. Since $\sigma_{\lambda_s-1} \in z^{\mathrm{i}}_{\lambda_s}$, it follows that $\sigma_{\lambda_s-1} \in \sum_{\ell=1}^{s} z^{\mathrm{i}}_{\lambda_\ell} = \zeta$. This implies that $\zeta \nsubseteq G_{\lambda_s-1}$ because $\sigma_{\lambda_s-1} \notin G_{\lambda_s-1}$. However, we have that $\mathrm{j}_* \leq \lambda_s - 1 \leq \mathrm{i}$ because $\mathrm{j}_* < \bar{\mathrm{j}} \leq \lambda_s \leq \mathrm{i}$, which means that $\zeta \subseteq G_{\lambda_s-1}$. So we have reached a contradiction, meaning that $\mathrm{j}_* = \bar{\mathrm{j}}$. For each $\mathrm{j} \in \mathcal{U}_{\mathrm{i}+1}$, if $z^{\mathrm{i}}_{\mathrm{j}}$ does not contain σ_{i}, let $z^{\mathrm{i}+1}_{\mathrm{j}} = z^{\mathrm{i}}_{\mathrm{j}} \subseteq G_{\mathrm{i}+1}$. If $z^{\mathrm{i}}_{\mathrm{j}}$ contains σ_{i}, let $z^{\mathrm{i}+1}_{\mathrm{j}} = z^{\mathrm{i}}_{\mathrm{j}} + z^{\mathrm{i}}_{\mathrm{j}_*} \subseteq G_{\mathrm{i}+1}$. Note that since $\mathrm{j}_* = \bar{\mathrm{j}}$, we must have that $\mathrm{j}_* < \mathrm{j}$, which means that $z^{\mathrm{i}}_{\mathrm{j}_*} \subseteq G_k$ for every $k \in [\mathrm{j}, \mathrm{i}] \subseteq [\mathrm{j}_*, \mathrm{i}]$. Therefore, $z^{\mathrm{i}+1}_{\mathrm{j}} = z^{\mathrm{i}}_{\mathrm{j}} + z^{\mathrm{i}}_{\mathrm{j}_*} \subseteq G_k$ for every $k \in [\mathrm{j}, \mathrm{i}]$. Also since $z^{\mathrm{i}}_{\mathrm{j}_*} \subseteq G_{\mathrm{j}-1}$, $z^{\mathrm{i}}_{\mathrm{j}_*}$ does not contain $\sigma_{\mathrm{j}-1}$, which means that $z^{\mathrm{i}+1}_{\mathrm{j}}$ contains $\sigma_{\mathrm{j}-1}$. Note that $\{z^{\mathrm{i}+1}_{\mathrm{j}} \mid \mathrm{j} \in \mathcal{U}_{\mathrm{i}+1}\}$ must still be linearly independent, so they form a basis of $\mathsf{Z}_1(G_{\mathrm{i}+1})$. Now we have that $\{z^{\mathrm{i}+1}_{\mathrm{j}} \mid \mathrm{j} \in \mathcal{U}_{\mathrm{i}+1}\}$ serves as the 1-cycles as specified in the proposition for iteration $\mathrm{i}+1$. □

Theorem 4.4.1. *Algorithm 4.4.1 computes the 1-st zigzag barcode for a given zigzag filtration on graphs.*

Proof. The claim follows directly from Proposition 4.3.1. For each interval $[\mathrm{j}_*, \mathrm{i}]$ produced from the pairing in Algorithm 4.4.1, by the Pairing Principle, there exists a 1-cycle z containing both σ_{j_*-1} and σ_{i} with $z \subseteq G_k$ for every $k \in [\mathrm{j}_*, \mathrm{i}]$. The cycle z induces a set of 1-representatives for $[\mathrm{j}_*, \mathrm{i}]$. For each interval produced at the end, Proposition 4.4.1 implies that such an interval admits 1-representatives. □

4.4.1 Efficient implementation

For every i and every $\mathrm{j} \leq \mathrm{i}$, define $\Gamma^{\mathrm{i}}_{\mathrm{j}}$ as the graph derived from G_{j} by deleting every edge σ_k s.t. $\mathrm{j} \leq k < \mathrm{i}$ and $G_k \xleftarrow{\sigma_k} G_{k+1}$ is backward. For convenience, we also assume that $\Gamma^{\mathrm{i}}_{\mathrm{j}}$ contains all the vertices of G. We can simplify the Pairing Principle as suggested by the following proposition:

Proposition 4.4.2. *In each iteration* i *of Algorithm 4.4.1 where* $G_\mathsf{i} \xleftarrow{\sigma_\mathsf{i}} G_{\mathsf{i}+1}$ *provides* $\mathsf{i} \in \mathsf{N}(\mathsf{H}_1(\mathcal{F}))$*, the set* J_i *in the Pairing Principle can be alternatively defined as consisting of every* $\mathsf{j} \in \mathcal{U}_\mathsf{i}$ *s.t.* $\sigma_\mathsf{i} \in \Gamma^\mathsf{i}_\mathsf{j}$ *and the vertices of* σ_i *are connected in* $\Gamma^{\mathsf{i}+1}_\mathsf{j}$ *(*$\sigma_\mathsf{i} \notin \Gamma^{\mathsf{i}+1}_\mathsf{j}$ *by definition).*

Proof. We prove an equivalent statement, which is that J_i consists of every $\mathsf{j} \in \mathcal{U}_\mathsf{i}$ s.t. there is a 1-cycle in $\Gamma^\mathsf{i}_\mathsf{j}$ containing σ_i. Let j be any index in $\mathcal{U}_\mathsf{i}$. It is not hard to see that a 1-cycle is in G_k for every $k \in [\mathsf{j}, \mathsf{i}]$ iff the 1-cycle is in $\Gamma^\mathsf{i}_\mathsf{j}$. So we only need to prove that there is a 1-cycle in $\Gamma^\mathsf{i}_\mathsf{j}$ containing both $\sigma_{\mathsf{j}-1}$ and σ_i iff there is a 1-cycle in $\Gamma^\mathsf{i}_\mathsf{j}$ containing σ_i. The forward direction is easy. So let z be a 1-cycle in $\Gamma^\mathsf{i}_\mathsf{j}$ containing σ_i. If z contains $\sigma_{\mathsf{j}-1}$, then the proof is done. If not, by Proposition 4.4.1 there is a 1-cycle z' containing $\sigma_{\mathsf{j}-1}$ with $z' \subseteq G_k$ for every $k \in [\mathsf{j}, \mathsf{i}]$. So z' is a 1-cycle in $\Gamma^\mathsf{i}_\mathsf{j}$ containing $\sigma_{\mathsf{j}-1}$. If z' contains σ_i, we again finish our proof. If not, then $z + z'$ is a 1-cycle containing both edges. □

We then turn graphs in $\mathcal{F}$ into weighted ones in the following way: initially, $G_0 = \varnothing$; then, whenever an edge σ_i is added from G_i to $G_{\mathsf{i}+1}$, the weight $w(\sigma_\mathsf{i})$ is set to i. We have the following fact:

Proposition 4.4.3. *For every* i *and every* $\mathsf{j} \leq \mathsf{i}$*, the edge set of* $\Gamma^\mathsf{i}_\mathsf{j}$*, denoted* $E(\Gamma^\mathsf{i}_\mathsf{j})$*, consists of all edges of* G_i *whose weights are less than* j*.*

Proof. We can prove this by induction on i. For $\mathsf{i} = 0$, $G_\mathsf{i} = \varnothing$ and the proposition is trivially true. Suppose that the proposition is true for i. If G_i and $G_{\mathsf{i}+1}$ differ by a vertex, then the proposition is also true for $\mathsf{i}+1$ because the edges stay the same. If $G_{\mathsf{i}+1}$ is derived from G_i by adding an edge σ_i, by the assumption, $E(\Gamma^\mathsf{i}_\mathsf{j})$ consists of all edges of G_i whose weights are less than j for each $\mathsf{j} \leq \mathsf{i}$. Note that $E(\Gamma^\mathsf{i}_\mathsf{j}) = E(\Gamma^{\mathsf{i}+1}_\mathsf{j})$ because $G_\mathsf{i} \xrightarrow{\sigma_\mathsf{i}} G_{\mathsf{i}+1}$ is an addition. So we have that $E(\Gamma^{\mathsf{i}+1}_\mathsf{j})$ consists of all edges of $G_{\mathsf{i}+1}$ whose weights are less than j because $w(\sigma_\mathsf{i}) = \mathsf{i} \geq \mathsf{j}$. Since $E(\Gamma^{\mathsf{i}+1}_{\mathsf{i}+1}) = E(G_{\mathsf{i}+1})$, the claim is also true for $E(\Gamma^{\mathsf{i}+1}_{\mathsf{i}+1})$. Now consider the situation that $G_{\mathsf{i}+1}$ is derived from G_i by deleting an edge σ_i. Then, σ_i must be added to the filtration previously, and let $G_k \xrightarrow{\sigma_k} G_{k+1}$ with $k < \mathsf{i}$ and $\sigma_k = \sigma_\mathsf{i}$ be the *latest* such addition. Note that $w(\sigma_\mathsf{i}) = k$ in G_i. For $\mathsf{j} \leq k$, $\sigma_\mathsf{i} \notin E(\Gamma^\mathsf{i}_\mathsf{j})$ because $w(\sigma_\mathsf{i}) = k \geq \mathsf{j}$. Since $E(\Gamma^{\mathsf{i}+1}_\mathsf{j}) = E(\Gamma^\mathsf{i}_\mathsf{j}) \setminus \{\sigma_\mathsf{i}\}$, we have that $E(\Gamma^{\mathsf{i}+1}_\mathsf{j}) = E(\Gamma^\mathsf{i}_\mathsf{j})$. Therefore, $E(\Gamma^{\mathsf{i}+1}_\mathsf{j})$ consists of all edges of $G_{\mathsf{i}+1}$ whose weights are less than j because $w(\sigma_\mathsf{i}) \geq \mathsf{j}$ in G_i. For each j s.t. $k < \mathsf{j} \leq \mathsf{i}$, we have $\sigma_{\mathrm{i}} \in E(\Gamma^{\mathrm{i}}_{\mathrm{j}})$ and $E(\Gamma^{\mathrm{i}+1}_{\mathrm{j}}) = E(\Gamma^{\mathrm{i}}_{\mathrm{j}}) \setminus \{\sigma_{\mathrm{i}}\}$. Since $w(\sigma_{\mathrm{i}}) < \mathrm{j}$ in G_{i}, it is true that $E(\Gamma^{\mathrm{i}+1}_{\mathrm{j}})$ consists of all edges of $G_{\mathrm{i}+1}$ whose weights are less than j, and the proof is done. □

Suppose that in an iteration i of Algorithm 4.4.1, $G_{\mathrm{i}} \xleftarrow{\sigma_{\mathrm{i}}} G_{\mathrm{i}+1}$ provides a negative index i. Let $\mathcal{U}_{\mathrm{i}} = \{\mathrm{j}_1 < \mathrm{j}_2 < \cdots < \mathrm{j}_\ell\}$ and the vertices of σ_{i} be u, v. Proposition 4.4.3 implies that

$$\Gamma^{\mathrm{i}+1}_{\mathrm{j}_1} \subseteq \Gamma^{\mathrm{i}+1}_{\mathrm{j}_2} \subseteq \cdots \subseteq \Gamma^{\mathrm{i}+1}_{\mathrm{j}_\ell} \subseteq \Gamma^{\mathrm{i}+1}_{\mathrm{i}+1} \tag{4.1}$$

By Proposition 4.4.2, in order to find the positive index to pair with i, one only needs to find the smallest $\mathrm{j}'_* \in \mathcal{U}_{\mathrm{i}}$ s.t. u, v are connected in $\Gamma^{\mathrm{i}+1}_{\mathrm{j}'_*}$. (Note that for a $\mathrm{j} \in \mathcal{U}_{\mathrm{i}}$ to satisfy $\sigma_{\mathrm{i}} \in \Gamma^{\mathrm{i}}_{\mathrm{j}}$, j only needs to be greater than $w(\sigma_{\mathrm{i}})$; such a smallest j is easy to derive.) We further expand Sequence (4.1) into the following finer version:

$$\Gamma^{\mathrm{i}+1}_{0} \subseteq \Gamma^{\mathrm{i}+1}_{1} \subseteq \cdots \subseteq \Gamma^{\mathrm{i}+1}_{\mathrm{i}} \subseteq \Gamma^{\mathrm{i}+1}_{\mathrm{i}+1} \tag{4.2}$$

where each consecutive $\Gamma^{\mathrm{i}+1}_{k}, \Gamma^{\mathrm{i}+1}_{k+1}$ are either the same or differ by only one edge. To get j'_*, we instead scan Sequence (4.2) and find the smallest $k_* \in \{0, 1, \ldots, \mathrm{i}+1\}$ s.t. u, v are connected in $\Gamma^{\mathrm{i}+1}_{k_*}$. Proposition 4.4.4 characterize such a k_*:

Proposition 4.4.4. *For a path in a weighted graph, let the* max edge-weight *of the path be the maximum weight of its edges. Then, the integer $k_* - 1$ equals the max edge-weight of the unique path connecting u, v in the unique minimum spanning forest of $G_{\mathrm{i}+1}$.*

Proof. We first notice that, since weighted graphs considered in this chapter have distinct weights, they all have unique minimum spanning forests. Let T be the minimum spanning forest of $\Gamma^{\mathrm{i}+1}_{\mathrm{i}+1}$; we prove an equivalent statement of the proposition, which is that $k_* - 1$ equals the max edge-weight of the unique path connecting u, v in T. Since k_* is the smallest index s.t. u, v are connected in $\Gamma^{\mathrm{i}+1}_{k_*}$, we have that u, v are *not* connected in $\Gamma^{\mathrm{i}+1}_{k_*-1}$. This indicates that $\Gamma^{\mathrm{i}+1}_{k_*} \neq \Gamma^{\mathrm{i}+1}_{k_*-1}$. Assume that $\Gamma^{\mathrm{i}+1}_{k_*}$ and $\Gamma^{\mathrm{i}+1}_{k_*-1}$ differ by an edge e; then, $w(\mathrm{e}) = k_* - 1$. Let T' be the minimum spanning forest of $\Gamma^{\mathrm{i}+1}_{k_*-1}$. Then, u, v are not connected in T' and e connects the connected components of u, v in T'. Since spanning forests have a matroid structure, $T' \cup \{\mathrm{e}\}$ must be a subforest of T (indeed, e would be the edge added to T' by

Kruskal's algorithm; see, e.g., [56]). Also, since there is a unique path between two vertices in a forest, the path from u to v in $T' \cup \{\mathrm{e}\}$ must be the path from u to v in T. This path has a max edge-weight of $k_* - 1$ and the proof is done. $\square$

Based on Proposition 4.4.4, finding k_* reduces to computing the max edge-weight of the path connecting u, v in the minimum spanning forest (MSF) of $G_{\mathrm{i}+1}$. For this, we utilize the *dynamic-MSF* data structure proposed by Holm et al. [57]. Assuming that n is the number of vertices and edges of G, the dynamic-MSF data structure supports the following operations:

- Return the identifier of a vertex's connected component in $O(\log n)$ time, which can be used to determine whether two vertices are connected.
- Return the max edge-weight of the path connecting any two vertices in the MSF in $O(\log n)$ time.
- Insert or delete an edge from the current graph (maintained by the data structure) and possibly update the MSF in $O(\log^4 n)$ amortized time.

We now present the full details of the algorithm in Algorithm 4.4.2. We can see that Algorithm 4.4.2 has time complexity $O(m \log^4 n)$, where each iteration is dominated by the update of $\mathbb{F}$.

Algorithm 4.4.2 Computing 1-st zigzag persistence on graphs: Full details

Maintain a dynamic-MSF data structure $\mathbb{F}$, which consists of all vertices of G and no edges initially. Also, set $\mathcal{U}_0 = \varnothing$. Then, for each $\mathrm{i} = 0, \ldots, m-1$, if σ_i is a vertex, do nothing; otherwise, do the following:

Case $G_\mathrm{i} \xrightarrow{\sigma_\mathrm{i}} G_{\mathrm{i}+1}$**:** Check whether vertices of σ_i are connected in G_i by querying $\mathbb{F}$, and then add σ_i to $\mathbb{F}$. If vertices of σ_i are connected in G_i, then set $\mathcal{U}_{\mathrm{i}+1} = \mathcal{U}_\mathrm{i} \cup \{\mathrm{i}+1\}$; otherwise, set $\mathcal{U}_{\mathrm{i}+1} = \mathcal{U}_\mathrm{i}$.

Case $G_\mathrm{i} \xleftarrow{\sigma_\mathrm{i}} G_{\mathrm{i}+1}$**:** Delete σ_i from $\mathbb{F}$. If the vertices u, v of σ_i are found to be not connected in $G_{\mathrm{i}+1}$ by querying $\mathbb{F}$, then set $\mathcal{U}_{\mathrm{i}+1} = \mathcal{U}_\mathrm{i}$; otherwise, do the following:

- Find the max edge-weight w_* of the path connecting u, v in the MSF of $G_{\mathrm{i}+1}$ by querying $\mathbb{F}$.
- Find the smallest index j_* of $\mathcal{U}_\mathrm{i}$ greater than $\max\{w_*, w(\sigma_\mathrm{i})\}$. (Note that we can store $\mathcal{U}_\mathrm{i}$ as a red-black tree [56], so that finding j_* takes $O(\log n)$ time.)
- Output an interval $[\mathrm{j}_*, \mathrm{i}]$ and set $\mathcal{U}_{\mathrm{i}+1} = \mathcal{U}_\mathrm{i} \setminus \{\mathrm{j}_*\}$.

At the end, for each $\mathrm{j} \in \mathcal{U}_m$, output an interval $[\mathrm{j}, m]$.

5. CONCLUSIONS

In this dissertation, we study several problems concerning representatives for topological persistence:

- We look into the representative problem for (standard) persistence homology (termed as persistent cycles) and investigate the NP-hardness of the problem for computing optimal (minimum) persistent cycles. Beside the NP-hardness results, we also propose polynomial-time algorithms for computing minimum persistent p-cycles for weak $(p+1)$-pseudomanifolds.

- We define alternative persistent cycles (i.e., a sequence) capturing the dynamic changes of homological features born and dying with persistence intervals, which the standard persistent cycles do not reveal. In presence of the NP-hardness results proved previously, we also present polynomial-time algorithms computing optimal sequences of the persistent cycles for weak pseudomanifolds.

- We provide near-linear algorithms for computing zigzag persistence on graphs, improving the previously known $O(m^\omega)$ time complexity for graph inputs, where $\omega < 2.37286$ is the matrix multiplication exponent. The design and correctness proofs of the algorithms are accomplished with the help of representatives.

Thinking forward, we identify the following open problems which we believe are worthy of further exploration:

- In our experiments where we computed optimal persistent cycles for certain data [20], some persistent cycles correspond to important features of the data while some do not have obvious meanings. If there are ways to design filtrations for data such that persistent cycles are related to the important features, then the prospect for the application of persistent cycles or persistence in general would be more extensive.

- As found in [20], persistent cycles are not stable in general even when only the weights of the cycles are considered. It will be helpful to figure out assumptions that are still relevant in practice, but under which the persistent cycles remain stable.

- We have presented $O(n^2)$-time algorithms for computing a minimum persistent cycle for a given interval. A natural question is whether this time complexity can be improved. Furthermore, can we devise a better algorithm to compute minimum persistent cycles for all intervals (i.e., the minimum persistent basis), improving upon the obvious $O(n^3)$-time algorithm that runs our algorithms on each interval?

- In our algorithm for 0-dimensional zigzag persistence on graphs, we build the barcode graph and utilize the algorithm in [32] to compute the zigzag barcode, which is also adopted by [60]. Is there any other scenario in persistence computation where such a technique can be applied so that a more efficient algorithm can be derived?

- In our algorithm for 1-dimensional zigzag persistence on graphs, we utilize the dynamic-MSF data structure [57] for computing the max edge-weight of the path connecting two vertices in an MSF. The update of the data structure which takes $O(\log^4 n)$ amortized time becomes the bottleneck of the algorithm. However, for the computation, it can be verified that we only need to know the *minimax*[1] distance of two vertices in a graph. Is there any faster way to compute the minimax distance in a dynamic graph?

- Another interesting question is whether our algorithms for zigzag persistence on graphs are more efficient practically when implemented compared to some existing implementations of zigzag persistence algorithms for general dimension [21], [27].

[1]↑The *minimax* distance of two vertices in a graph is the minimum of the max edge-weights of all paths connecting the two vertices.

REFERENCES

[1] H. Edelsbrunner and J. Harer, *Computational Topology: An Introduction.* American Mathematical Soc., 2010.

[2] A. Hatcher, *Algebraic Topology.* Cambridge University Press, 2002, ISBN: 9780521795401.

[3] H. Edelsbrunner, D. Letscher, and A. Zomorodian, "Topological persistence and simplification," in *Proceedings 41st Annual Symposium on Foundations of Computer Science*, IEEE, 2000, pp. 454–463.

[4] A. Zomorodian and G. Carlsson, "Computing persistent homology," *Discrete & Computational Geometry*, vol. 33, no. 2, pp. 249–274, 2005.

[5] F. Chazal, D. Cohen-Steiner, M. Glisse, L. J. Guibas, and S. Y. Oudot, "Proximity of persistence modules and their diagrams," in *Proceedings of the Twenty-Fifth Annual Symposium on Computational Geometry*, ACM, 2009, pp. 237–246.

[6] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer, "Stability of persistence diagrams," *Discrete & Computational Geometry*, vol. 37, no. 1, pp. 103–120, 2007.

[7] G. Carlsson and V. de Silva, "Zigzag persistence," *Foundations of Computational Mathematics*, vol. 10, no. 4, pp. 367–405, 2010.

[8] T. K. Dey, F. Fan, and Y. Wang, "Computing topological persistence for simplicial maps," in *Proceedings of the Thirtieth Annual Symposium on Computational Geometry*, 2014, pp. 345–354.

[9] C. Hofer, R. Kwitt, M. Niethammer, and A. Uhl, "Deep learning with topological signatures," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[10] M. Carriere, M. Cuturi, and S. Oudot, "Sliced wasserstein kernel for persistence diagrams," in *International Conference on Machine Learning*, PMLR, 2017, pp. 664–673.

[11] Q. Zhao and Y. Wang, "Learning metrics for persistence-based summaries and applications for graph classification," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[12] Y.-M. Chung and S. Day, "Topological fidelity and image thresholding: A persistent homology approach," *Journal of Mathematical Imaging and Vision*, vol. 60, no. 7, pp. 1167–1179, 2018.

[13] A. V. Patel, T. Hou, J. D. B. Rodriguez, T. K. Dey, and D. P. Birnie III, “Topological filtering for 3D microstructure segmentation,” *Computational Materials Science*, vol. 202, p. 110 920, 2022.

[14] C. Chen, X. Ni, Q. Bai, and Y. Wang, “A topological regularizer for classifiers via persistent homology,” in *The 22nd International Conference on Artificial Intelligence and Statistics*, PMLR, 2019, pp. 2573–2582.

[15] V. de Silva, D. Morozov, and M. Vejdemo-Johansson, “Dualities in persistent (co) homology,” *Inverse Problems*, vol. 27, no. 12, p. 124 003, 2011.

[16] P. Wu, C. Chen, Y. Wang, *et al.*, “Optimal topological cycles and their application in cardiac trabeculae restoration,” in *International Conference on Information Processing in Medical Imaging*, Springer, 2017, pp. 80–92.

[17] E. W. Chambers, J. Erickson, and A. Nayyeri, “Minimum cuts and shortest homologous cycles,” in *Proceedings of the Twenty-Fifth Annual Symposium on Computational Geometry*, ACM, 2009, pp. 377–385.

[18] C. Chen and D. Freedman, “Hardness results for homology localization,” *Discrete & Computational Geometry*, vol. 45, no. 3, pp. 425–448, 2011.

[19] T. K. Dey, J. Sun, and Y. Wang, “Approximating loops in a shortest homology basis from point data,” in *Proceedings of the Twenty-Sixth Annual Symposium on Computational Geometry*, ACM, 2010, pp. 166–175.

[20] T. K. Dey, T. Hou, and S. Mandal, “Persistent 1-cycles: Definition, computation, and its application,” in *International Workshop on Computational Topology in Image Context*, Springer, 2019, pp. 123–136.

[21] C. Maria and S. Y. Oudot, “Zigzag persistence via reflections and transpositions,” in *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM, 2014, pp. 181–199.

[22] T. K. Dey and T. Hou, “Updating zigzag persistence and maintaining representatives over changing filtrations,” *arXiv preprint arXiv:2112.02352*, 2021.

[23] W. Kim and F. Mémoli, “Generalized persistence diagrams for persistence modules over posets,” *Journal of Applied and Computational Topology*, vol. 5, no. 4, pp. 533–581, 2021.

[24] A. Patel, “Generalized persistence diagrams,” *Journal of Applied and Computational Topology*, vol. 1, no. 3, pp. 397–419, 2018.

[25] T. K. Dey, W. Kim, and F. Mémoli, "Computing generalized rank invariant for 2-parameter persistence modules via zigzag persistence and its applications," *arXiv preprint arXiv:2111.15058*, 2021.

[26] F. Chazal, V. de Silva, M. Glisse, and S. Oudot, *The Structure and Stability of Persistence Modules.* Springer, 2016.

[27] G. Carlsson, V. de Silva, and D. Morozov, "Zigzag persistent homology and real-valued functions," in *Proceedings of the Twenty-Fifth Annual Symposium on Computational Geometry*, 2009, pp. 247–256.

[28] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer, "Extending persistence using Poincaré and Lefschetz duality," *Foundations of Computational Mathematics*, vol. 9, no. 1, pp. 79–103, 2009.

[29] P. Holme and J. Saramäki, "Temporal networks," *Physics Reports*, vol. 519, no. 3, pp. 97–125, 2012.

[30] N. Milosavljević, D. Morozov, and P. Skraba, "Zigzag persistent homology in matrix multiplication time," in *Proceedings of the Twenty-Seventh Annual Symposium on Computational Geometry*, 2011, pp. 216–225.

[31] J. Alman and V. V. Williams, "A refined laser method and faster matrix multiplication," in *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, SIAM, 2021, pp. 522–539.

[32] P. K. Agarwal, H. Edelsbrunner, J. Harer, and Y. Wang, "Extreme elevation on a 2-manifold," *Discrete & Computational Geometry*, vol. 36, no. 4, pp. 553–572, 2006.

[33] G. Borradaile, E. W. Chambers, K. Fox, and A. Nayyeriy, "Minimum cycle and homology bases of surface-embedded graphs.," *Journal of Computational Geometry*, vol. 8, no. 2, 2017.

[34] C. Chen and D. Freedman, "Measuring and computing natural generators for homology groups," *Computational Geometry*, vol. 43, no. 2, pp. 169–181, 2010.

[35] T. K. Dey, A. N. Hirani, and B. Krishnamoorthy, "Optimal homologous cycles, total unimodularity, and linear programming," *SIAM Journal on Computing*, vol. 40, no. 4, pp. 1026–1044, 2011.

[36] J. Erickson and K. Whittlesey, "Greedy optimal homotopy and homology generators," in *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, Society for Industrial and Applied Mathematics, 2005, pp. 1038–1046.

[37] K. Emmett, B. Schweinhart, and R. Rabadan, "Multiscale topology of chromatin folding," in *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2016, pp. 177–180.

[38] E. G. Escolar and Y. Hiraoka, "Optimal cycles for persistent homology via linear programming," in *Optimization in the Real World*, Springer, 2016, pp. 79–96.

[39] I. Obayashi, "Volume-optimal cycle: Tightest representative cycle of a generator in persistent homology," *SIAM Journal on Applied Algebra and Geometry*, vol. 2, no. 4, pp. 508–534, 2018.

[40] J. R. Munkres, *Elements of Algebraic Topology.* CRC Press, 2018.

[41] P. Bubenik and J. A. Scott, "Categorification of persistent homology," *Discrete & Computational Geometry*, vol. 51, no. 3, pp. 600–627, 2014.

[42] S. Awodey, *Category Theory.* Oxford University Press, 2010.

[43] C. H. Papadimitriou and M. Yannakakis, "Optimization, approximation, and complexity classes," *Journal of Computer and System Sciences*, vol. 43, no. 3, pp. 425–440, 1991.

[44] C. Chen and D. Freedman, "Quantifying homology classes ii: Localization and stability," *arXiv preprint arXiv:0709.2512*, 2007.

[45] D. L. Ferrario and R. A. Piccinini, *Simplicial structures in topology.* Springer Science & Business Media, 2010.

[46] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Introduction to algorithms, 3rd edition," in MIT Press, 2009, ch. 35, Approximation Algorithms, ISBN: 978-0-262-03384-8.

[47] J. B. Orlin, "Max flows in $O(nm)$ time, or better," in *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, ACM, 2013, pp. 765–774.

[48] J. Lee, *Introduction to topological manifolds.* Springer Science & Business Media, 2010, vol. 202.

[49] J. W. Alexander, "A proof and extension of the Jordan-Brouwer separation theorem," *Transactions of the American Mathematical Society*, vol. 23, no. 4, pp. 333–349, 1922.

[50] E. H. Spanier, *Algebraic Topology*, 1. Springer Science & Business Media, 1989, vol. 55.

[51] O. Busaryev, S. Cabello, C. Chen, T. K. Dey, and Y. Wang, "Annotating simplices with a homology basis and its applications," in *Scandinavian Workshop on Algorithm Theory*, Springer, 2012, pp. 189–200.

[52] T. K. Dey, T. Li, and Y. Wang, "Efficient algorithms for computing a minimal homology basis," in *Latin American Symposium on Theoretical Informatics*, Springer, 2018, pp. 376–398.

[53] T. K. Dey, T. Hou, and S. Mandal, "Computing minimal persistent cycles: Polynomial and hard cases," in *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM, 2020, pp. 2587–2606.

[54] W. Kim and F. Mémoli, "Stable signatures for dynamic graphs and dynamic metric spaces via zigzag persistence," *arXiv preprint arXiv:1712.04064*, 2017.

[55] J. Skarding, B. Gabrys, and K. Musial, "Foundations and modelling of dynamic networks using dynamic graph neural networks: A survey," *arXiv preprint arXiv:2005.07496*, 2020.

[56] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, 3rd Edition.* MIT Press, 2009, ISBN: 978-0-262-03384-8. [Online]. Available: http://mitpress.mit.edu/books/introduction-algorithms.

[57] J. Holm, K. De Lichtenberg, and M. Thorup, "Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity," *Journal of the ACM (JACM)*, vol. 48, no. 4, pp. 723–760, 2001.

[58] V. De Silva, D. Morozov, and M. Vejdemo-Johansson, "Persistent cohomology and circular coordinates," *Discrete & Computational Geometry*, vol. 45, no. 4, pp. 737–759, 2011.

[59] H. Edelsbrunner and S. Parsa, "On the computational complexity of betti numbers: Reductions from matrix rank," in *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM, 2014, pp. 152–160.

[60] T. K. Dey, "Computing height persistence and homology generators in $\mathbb{R}^3$ efficiently," in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM, 2019, pp. 2649–2662.

[61] L. Georgiadis, H. Kaplan, N. Shafrir, R. E. Tarjan, and R. F. Werneck, "Data structures for mergeable trees," *ACM Transactions on Algorithms (TALG)*, vol. 7, no. 2, pp. 1–30, 2011.

A. MISSING CONTENTS FOR CHAPTER 3

A.1 An abstract algorithm for zigzag persistence

We introduce an abstract algorithm for zigzag persistence that helps us prove some results in the appendix. Given $p \geq 0$ and a simplex-wise filtration $\mathcal{X} : \varnothing = X_0 \leftrightarrow \cdots \leftrightarrow X_\ell$ starting with an empty complex, the algorithm computes the p-th zigzag persistence intervals and their representative p-cycles for $\mathcal{X}$. Each linear map in $\mathsf{H}_p(\mathcal{X})$ is denoted as $\psi_\mathrm{i} : \mathsf{H}_p(X_\mathrm{i}) \leftrightarrow \mathsf{H}_p(X_{\mathrm{i}+1})$. Also, for any i s.t. $0 \leq \mathrm{i} \leq \ell$, $\mathcal{X}^\mathrm{i}$ denotes the filtration $X_0 \leftrightarrow X_1 \leftrightarrow \cdots \leftrightarrow X_\mathrm{i}$, which is a *prefix* of $\mathcal{X}$. Inspired by the algorithm by Maria and Oudot [21], the idea is to directly compute an interval decomposition by maintaining representative cycles for all intervals:

Algorithm A.1.1 (Zigzag persistence algorithm)**.** *First set $\mathsf{Pers}_p(\mathcal{X}^0) = \varnothing$. The algorithm then iterates for $\mathrm{i} \leftarrow 0, \ldots, \ell - 1$. At the beginning of the* i*-th iteration, the intervals and their representative cycles for $\mathsf{H}_p(\mathcal{X}^\mathrm{i})$ have already been computed. The aim of the* i*-th iteration is to compute these for $\mathsf{H}_p(\mathcal{X}^{\mathrm{i}+1})$. For describing the* i*-th iteration, let $\mathsf{Pers}_p(\mathcal{X}^\mathrm{i}) = \{[b_\alpha, d_\alpha] \mid \alpha \in \mathcal{A}^\mathrm{i}\}$ be indexed by a set $\mathcal{A}^\mathrm{i}$, and let $\{z_k^\alpha \subseteq X_k \mid b_\alpha \leq k \leq d_\alpha\}$ be a sequence of representative p-cycles for each $[b_\alpha, d_\alpha]$. For ease of presentation, we also let $z_k^\alpha = 0$ for each $\alpha \in \mathcal{A}^\mathrm{i}$ and each $k \in [0, \mathrm{i}] \setminus [b_\alpha, d_\alpha]$. We call intervals of $\mathsf{Pers}_p(\mathcal{X}^\mathrm{i})$ ending with* i *as* surviving intervals *at index* i*. Each non-surviving interval of $\mathsf{Pers}_p(\mathcal{X}^\mathrm{i})$ is directly included in $\mathsf{Pers}_p(\mathcal{X}^{\mathrm{i}+1})$ and its representative cycles stay the same. For surviving intervals of $\mathsf{Pers}_p(\mathcal{X}^\mathrm{i})$, the* i*-th iteration proceeds with the following cases:*

ψ_i **is an isomorphism:** *In this case, no intervals are created or cease to persist. For each surviving interval $[b_\alpha, d_\alpha]$ in $\mathsf{Pers}_p(\mathcal{X}^\mathrm{i})$, $[b_\alpha, d_\alpha] = [b_\alpha, \mathrm{i}]$ now corresponds to an interval $[b_\alpha, \mathrm{i}+1]$ in $\mathsf{Pers}_p(\mathcal{X}^{\mathrm{i}+1})$. The representative cycles for $[b_\alpha, \mathrm{i}+1]$ are set by the following rule:*

Trivial setting rule of representative cycles: *For each* j *with $b_\alpha \leq \mathrm{j} \leq \mathrm{i}$, the representative cycle for $[b_\alpha, \mathrm{i}+1]$ at index* j *stays the same. The representative cycle for $[b_\alpha, \mathrm{i}+1]$ at $\mathrm{i}+1$ is set to a $z_{\mathrm{i}+1}^\alpha \subseteq X_{\mathrm{i}+1}$ such that $[z_\mathrm{i}^\alpha] \leftrightarrow [z_{\mathrm{i}+1}^\alpha]$ by ψ_i (i.e., $[z_\mathrm{i}^\alpha] \mapsto [z_{\mathrm{i}+1}^\alpha]$ or $[z_\mathrm{i}^\alpha] \leftarrow\!\shortmid [z_{\mathrm{i}+1}^\alpha]$).*

ψ_{i} **is forward with non-trivial cokernel:** *A new interval* $[\mathrm{i}{+}1, \mathrm{i}{+}1]$ *is added to* $\mathsf{Pers}_p(\mathcal{X}^{\mathrm{i}+1})$ *and its representative cycle at* $\mathrm{i}+1$ *is set to a* p*-cycle in* $X_{\mathrm{i}+1}$ *containing* σ_{i} *(*σ_{i} *is a* p*-simplex). All surviving intervals of* $\mathsf{Pers}_p(\mathcal{X}^{\mathrm{i}})$ *persist to index* $\mathrm{i}+1$ *and are automatically added to* $\mathsf{Pers}_p(\mathcal{X}^{\mathrm{i}+1})$*; their representative cycles are set by the trivial setting rule.*

ψ_{i} **is backward with non-trivial kernel:** *A new interval* $[\mathrm{i}{+}1, \mathrm{i}{+}1]$ *is added to* $\mathsf{Pers}_p(\mathcal{X}^{\mathrm{i}+1})$ *and its representative cycle at* $\mathrm{i}+1$ *is set to a* p*-cycle homologous to* $\partial(\sigma_{\mathrm{i}})$ *in* $X_{\mathrm{i}+1}$ *(*σ_{i} *is a* $(p+1)$*-simplex). All surviving intervals of* $\mathsf{Pers}_p(\mathcal{X}^{\mathrm{i}})$ *persist to index* $\mathrm{i}+1$ *and their representative cycles are set by the trivial setting rule.*

ψ_{i} **is forward with non-trivial kernel:** *A surviving interval of* $\mathsf{Pers}_p(\mathcal{X}^{\mathrm{i}})$ *does not persist to* $\mathrm{i}+1$*. Let* $\mathcal{B}^{\mathrm{i}} \subseteq \mathcal{A}^{\mathrm{i}}$ *consist of indices of all surviving intervals. We have that* $\{[z^{\alpha}_{\mathrm{i}}] \mid \alpha \in \mathcal{B}^{\mathrm{i}}\}$ *forms a basis of* $\mathsf{H}_p(X_{\mathrm{i}})$*. Suppose that* $\psi_{\mathrm{i}}\Big([z^{\alpha_1}_{\mathrm{i}}] + \cdots + [z^{\alpha_h}_{\mathrm{i}}]\Big) = 0$*, where* $\alpha_1, \ldots, \alpha_h \in \mathcal{B}^{\mathrm{i}}$*. We can rearrange the indices such that* $b_{\alpha_1} < b_{\alpha_2} < \cdots < b_{\alpha_h}$ *and* $\alpha_1 < \alpha_2 < \cdots < \alpha_h$*. Let* λ *be* α_1 *if* $\psi_{b_\alpha - 1}$ *is backward for every* $\alpha \in \{\alpha_1, \ldots, \alpha_h\}$ *and otherwise be the largest* $\alpha \in \{\alpha_1, \ldots, \alpha_h\}$ *such that* $\psi_{b_\alpha - 1}$ *is forward. Then,* $[b_\lambda, \mathrm{i}]$ *forms an interval of* $\mathsf{Pers}_p(\mathcal{X}^{\mathrm{i}+1})$*. For each* $k \in [b_\lambda, \mathrm{i}]$*, let* $z'_k = z^{\alpha_1}_k + \cdots + z^{\alpha_h}_k$*; then,* $\{z'_k \mid b_\lambda \leq k \leq \mathrm{i}\}$ *is a sequence of representative cycles for* $[b_\lambda, \mathrm{i}]$*. All the other surviving intervals of* $\mathsf{Pers}_p(\mathcal{X}^{\mathrm{i}})$ *persist to* $\mathrm{i}{+}1$ *and their representative cycles are set by the trivial setting rule.*

ψ_{i} **is backward with non-trivial cokernel:** *A surviving interval of* $\mathsf{Pers}_p(\mathcal{X}^{\mathrm{i}})$ *does not persist to* $\mathrm{i}+1$*. Let* $\mathcal{B}^{\mathrm{i}} \subseteq \mathcal{A}^{\mathrm{i}}$ *consist of indices of all surviving intervals, and let* $z^{\alpha_1}_{\mathrm{i}}, \ldots, z^{\alpha_h}_{\mathrm{i}}$ *be the cycles in* $\{z^{\alpha}_{\mathrm{i}} \mid \alpha \in \mathcal{B}^{\mathrm{i}}\}$ *containing* σ_{i} *(*σ_{i} *is a* p*-simplex). We can rearrange the indices such that* $b_{\alpha_1} < b_{\alpha_2} < \cdots < b_{\alpha_h}$ *and* $\alpha_1 < \alpha_2 < \cdots < \alpha_h$*. Let* λ *be* α_1 *if* $\psi_{b_\alpha - 1}$ *is forward for every* $\alpha \in \{\alpha_1, \ldots, \alpha_h\}$ *and otherwise be the largest* $\alpha \in \{\alpha_1, \ldots, \alpha_h\}$ *such that* $\psi_{b_\alpha - 1}$ *is backward. Then,* $[b_\lambda, \mathrm{i}]$ *forms an interval of* $\mathsf{Pers}_p(\mathcal{X}^{\mathrm{i}+1})$ *and the representative cycles for* $[b_\lambda, \mathrm{i}]$ *stay the same. For each* $\alpha \in \{\alpha_1, \ldots, \alpha_h\} \setminus \{\lambda\}$*, let* $z'_k = z^{\alpha}_k + z^{\lambda}_k$ *for each* k *s.t.* $b_\alpha \leq k \leq \mathrm{i}$*, and let* $z'_{\mathrm{i}+1} = z'_{\mathrm{i}}$*; then,* $\{z'_k \mid b_\alpha \leq k \leq \mathrm{i}+1\}$ *is a sequence of representative cycles for* $[b_\alpha, \mathrm{i}+1]$*. For the* *other surviving intervals, the setting of representative cycles follows the trivial setting rule.*

To show that Algorithm A.1.1 is correct, we prove by induction on each $\mathrm{i} \in [0, \ell]$ that the algorithm computes a valid interval decomposition of $\mathsf{H}_p(\mathcal{X}^{\mathrm{i}})$. For $\mathrm{i} = 0$, this is trivially true. Now suppose that this is true for an $\mathrm{i} \in [0, \ell - 1]$, i.e., before the i-th iteration, what the algorithm computes are valid. In the i-th iteration, for the case that ψ_{i} is isomorphic, the case that ψ_{i} is forward with non-trivial cokernel, or the case that ψ_{i} is backward with non-trivial kernel, the proof is similar to what is done in the proof of Proposition 3.5.1 and is omitted.

For the case that ψ_{i} is forward with non-trivial kernel, we first verify that $\{z'_k \mid b_\lambda \leq k \leq \mathrm{i}\}$ computed in the i-th iteration is a valid sequence of representative cycles for $[b_\lambda, \mathrm{i}] \in \mathsf{Pers}_p(\mathcal{X}^{\mathrm{i}})$. Condition 2 of Definition 3.5.1 is trivially satisfied. Suppose that $\psi_{b_\lambda - 1}$ is forward. Then, for each $\alpha \in \{\alpha_1, \dots, \alpha_h\} \setminus \{\lambda\}$ s.t. $z^\alpha_{b_\lambda} \neq 0$, we must have $b_\alpha < b_\lambda$. Therefore, $z^\alpha_{b_\lambda}$ is in the image of $\psi_{b_\lambda - 1}$. Since $z^\lambda_{b_\lambda}$ is not in the image of $\psi_{b_\lambda - 1}$, so is not $z'_{b_\lambda} = z^{\alpha_1}_{b_\lambda} + \cdots + z^{\alpha_h}_{b_\lambda}$, and Condition 1 is satisfied. For $k \in [b_\lambda, \mathrm{i} - 1]$ s.t. $k + 1 \neq b_\alpha$ for any $\alpha \in \{\alpha_1, \dots, \alpha_h\}$, we have that $[z^\alpha_k] \leftrightarrow [z^\alpha_{k+1}]$ by ψ_k for each $\alpha \in \{\alpha_1, \dots, \alpha_h\}$. This implies that $[z'_k] \leftrightarrow [z'_{k+1}]$ by ψ_k. For $k \in [b_\lambda, \mathrm{i} - 1]$ s.t. $k + 1 = b_\beta$ for a $\beta \in \{\alpha_1, \dots, \alpha_h\}$, we have that $\psi_{b_\beta - 1} = \psi_k$ is backward because λ is the the largest $\alpha \in \{\alpha_1, \dots, \alpha_h\}$ such that $\psi_{b_\alpha - 1}$ is forward. We then have $0 \leftarrow\!\!\shortmid [z^\beta_{k+1}]$ by ψ_k, which implies that $[z'_k] \leftarrow\!\!\shortmid [z'_{k+1}]$ by ψ_k. Hence, Condition 3 is verified. Now suppose that $\psi_{b_\lambda - 1}$ is backward. Under this situation, every $\alpha \in \{\alpha_1, \dots, \alpha_h\}$ has $\psi_{b_\alpha - 1}$ being backward where b_λ is the smallest birth index. In $z'_{b_\lambda} = z^{\alpha_1}_{b_\lambda} + \cdots + z^{\alpha_h}_{b_\lambda}$, $z^\lambda_{b_\lambda}$ is the only non-zero cycle. So we have $\psi_{b_{\lambda - 1}}\big([z'_{b_\lambda}]\big) = \psi_{b_{\lambda - 1}}\big([z^\lambda_{b_\lambda}]\big) = 0$, and Condition 1 is satisfied. It can also be verified that Condition 3 is satisfied. We now have that $\{z'_k \mid b_\lambda \leq k \leq \mathrm{i}\}$ is a valid sequence of representative cycles for $[b_\lambda, \mathrm{i}] \in \mathsf{Pers}_p(\mathcal{X}^{\mathrm{i}})$ with $\psi_{\mathrm{i}}\big([z'_{\mathrm{i}}]\big) = 0$. It is then not hard to verify that the i-th iteration computes a valid interval decomposition for $\mathsf{H}_p(\mathcal{X}^{\mathrm{i}+1})$.

For the case that ψ_{i} is backward with non-trivial cokernel, we first verify that for any $\alpha \in \{\alpha_1, \dots, \alpha_h\} \setminus \{\lambda\}$, the sequence $\{z'_k \mid b_\alpha \leq k \leq \mathrm{i}\}$ computed in the i-th iteration provides valid representative cycles for $[b_\alpha, \mathrm{i}] \in \mathsf{Pers}_p(\mathcal{X}^{\mathrm{i}})$. Condition 2 of Definition 3.5.1 is trivially satisfied. Suppose that $\psi_{b_\lambda - 1}$ is backward and $b_\alpha < b_\lambda$. Then, $z'_{b_\alpha} = z^\alpha_{b_\alpha}$, and

Condition 1 is satisfied. For $k \in [b_\alpha, \mathrm{i}-1]$ s.t. $k+1 \neq b_\lambda$, it is obvious that $[z'_k] \leftrightarrow [z'_{k+1}]$ by ψ_k. Since $\psi_{b_\lambda-1}$ is backward, we have $\psi_{b_\lambda-1}\big([z^\lambda_{b_\lambda}]\big) = 0$, and so $\psi_{b_\lambda-1}\big([z'_{b_\lambda}]\big) = [z'_{b_\lambda-1}]$. Hence, Condition 3 is satisfied. Suppose that $\psi_{b_\lambda-1}$ is backward and $b_\alpha > b_\lambda$. Under this situation, $\psi_{b_\alpha-1}$ must be forward because λ is the largest $\beta \in \{\alpha_1, \dots, \alpha_h\}$ such that $\psi_{b_\beta-1}$ is backward. We then have that $[z^\alpha_{b_\alpha}]$ is outside the image of $\psi_{b_\alpha-1}$ and $[z^\lambda_{b_\alpha}]$ is in, which implies that $[z'_{b_\alpha}]$ is outside the image of $\psi_{b_\alpha-1}$. Therefore, Condition 1 is satisfied. It is also not hard to see that Condition 3 is satisfied. Now suppose that $\psi_{b_\lambda-1}$ is forward. Under this situation, every $\beta \in \{\alpha_1, \dots, \alpha_h\}$ has $\psi_{b_\beta-1}$ being forward where b_λ is the smallest birth index. Therefore, $\psi_{b_\alpha-1}$ is forward and $b_\alpha > b_\lambda$. Condition 1 and 3 can be similarly verified. We now have that $\{z'_k \mid b_\alpha \leq k \leq \mathrm{i}\}$ is a valid sequence of representative cycles for $[b_\alpha, \mathrm{i}] \in \mathsf{Pers}_p(\mathcal{X}^\mathrm{i})$ with $[z'_\mathrm{i}] \in \mathsf{img}(\psi_\mathrm{i})$. It is then not hard to verify that the i-th iteration computes a valid interval decomposition for $\mathsf{H}_p(\mathcal{X}^{\mathrm{i}+1})$.

A.2 Missing proofs

A.2.1 Proof of Proposition 3.2.1

We only prove that $K_\beta \subseteq \mathbb{K}^p_{(b-1,b]}$ because the proof for $K_\delta \subseteq \mathbb{K}^p_{[d,d+1)}$ is similar. For contradiction, assume instead that $K_\beta \not\subseteq \mathbb{K}^p_{(b-1,b]}$. Note that from $\mathbb{K}^p_{(b-1,b]}$ to $\mathbb{K}^p_{(b-1,b+1)}$, we are not crossing any p-th critical values, and so the linear map $\mathsf{H}_p(\mathbb{K}^p_{(b-1,b]}) \to \mathsf{H}_p(\mathbb{K}^p_{(b-1,b+1)})$ is an isomorphism (see Proposition 3.4.1). Since $\mathbb{K}^p_{(b-1,b]}$ appears between $\mathbb{K}^p_{(b-1,b)}$ and $\mathbb{K}^p_{(b-1,b+1)}$ in $\mathcal{F}_p(f)$ (see Definition 3.2.3), we have the following subsequence in $\mathcal{F}_p(f)$:

$$\mathbb{K}^p_{(b-1,b)} \hookrightarrow \cdots \hookrightarrow \mathbb{K}^p_{(b-1,b]} \hookrightarrow \cdots \hookrightarrow K_\beta \hookrightarrow \cdots \hookrightarrow \mathbb{K}^p_{(b-1,b+1)} \hookrightarrow \cdots \hookrightarrow K_\delta$$

The fact that $[K_\beta, K_\delta]$ forms an interval in $\mathsf{Pers}_p(\mathcal{F}_p(f))$ indicates that a p-th homology class is born (and persists) when $\mathbb{K}^p_{(b-1,b]}$ is included into $\mathbb{K}^p_{(b-1,b+1)}$, contradicting the fact that $\mathsf{H}_p(\mathbb{K}^p_{(b-1,b]}) \to \mathsf{H}_p(\mathbb{K}^p_{(b-1,b+1)})$ is an isomorphism.

A.2.2 Proof of Proposition 3.2.2

Let S consist of simplices of K not in $\mathbb{K}^p_{(\mathrm{i,j})}$ whose interiors intersect $\mathbb{X}^p_{(\mathrm{i,j})}$. Then, let σ be a simplex of S with no proper cofaces in S. We have that there exists a $u \in \sigma$ with $f(u) \in (\alpha^p_{\mathrm{i}}, \alpha^p_{\mathrm{j}})$ and a $w \in \sigma$ with $f(w) \notin (\alpha^p_{\mathrm{i}}, \alpha^p_{\mathrm{j}})$. If $f(w) \leq \alpha^p_{\mathrm{i}}$, then all vertices in σ must have the function values falling in $(\alpha^p_{\mathrm{i}-1}, \alpha^p_{\mathrm{i}+1})$ because K is compatible with the p-th levelsets of f. We then have that $|\sigma| \cap \mathbb{X}^p_{(\mathrm{i,j})}$ deformation retracts to $\mathsf{bd}(|\sigma|) \cap \mathbb{X}^p_{(\mathrm{i,j})}$, where $\mathsf{bd}(|\sigma|)$ denotes the boundary of the topological disc $|\sigma|$. This implies that $\mathbb{X}^p_{(\mathrm{i,j})}$ deformation retracts to $\mathbb{X}^p_{(\mathrm{i,j})} \setminus \mathsf{Int}(\sigma)$, where $\mathsf{Int}(\sigma)$ denotes the interior of $|\sigma|$. If $f(w) \geq \alpha^p_{\mathrm{j}}$, the result is similar. After doing the above for the all such σ in S, we have that $\mathbb{X}^p_{(\mathrm{i,j})}$ deformation retracts to $\mathbb{X}^p_{(\mathrm{i,j})} \setminus \bigcup_{\sigma \in S} \mathsf{Int}(\sigma)$. Note that $\mathbb{X}^p_{(\mathrm{i,j})} \setminus \bigcup_{\sigma \in S} \mathsf{Int}(\sigma) = \left|\mathbb{K}^p_{(\mathrm{i,j})}\right|$, and so the proof is done.

A.2.3 Proof of Proposition 3.3.1

For the proof, we first observe the following fact which follows immediately from Proposition 3.5.1:

Proposition A.2.1. *Let $p \geq 0$, $\mathcal{X} : X_0 \leftrightarrow \cdots \leftrightarrow X_\ell$ be a simplex-wise filtration with $\mathsf{H}_p(X_0) = 0$, $[\beta', \delta']$ be an interval in $\mathsf{Pers}_p(\mathcal{X})$, and $\zeta_{\beta'}, \ldots, \zeta_{\delta'}$ be a sequence of representative p-cycles for $[\beta', \delta']$. One has that ζ_{i} is not a boundary in X_{i} for each $\beta' \leq \mathrm{i} \leq \delta'$.*

The following fact is also helpful to our proof:

Proposition A.2.2. *Let X be a simplicial complex, A be a q-chain of X where $q \geq 1$, and X' be the closure of a q-connected component of X; one has that $X' \cap \partial(A) = \partial(X' \cap A)$.*

Proof. First, let B be an arbitrary q-chain of X and σ^{q-1} be an arbitrary $(q-1)$-simplex in X. Define $\mathsf{cof}_q(B, \sigma^{q-1})$ as the set of q-simplices in B having σ^{q-1} as a face. It can be verified that $\mathsf{cof}_q(B, \sigma^{q-1}) = \mathsf{cof}_q(X' \cap B, \sigma^{q-1})$ if $\sigma^{q-1} \in X'$.

To prove the proposition, let σ^{q-1} be an arbitrary $(q-1)$-simplex in $X' \cap \partial(A)$. Since $\sigma^{q-1} \in \partial(A)$, we have that $\left|\mathsf{cof}_q(A, \sigma^{q-1})\right|$ is an odd number. Since $\sigma^{q-1} \in X'$, the fact in the previous paragraph implies that $\left|\mathsf{cof}_q(X' \cap A, \sigma^{q-1})\right| = \left|\mathsf{cof}_q(A, \sigma^{q-1})\right|$ is also an odd number. Therefore, $\sigma^{q-1} \in \partial(X' \cap A)$. On the other hand, let σ^{q-1} be an arbitrary $(q-1)$-simplex in

$\partial(X' \cap A)$. Then, $\left|\mathsf{cof}_q(X' \cap A, \sigma^{q-1})\right|$ is an odd number. Since σ^{q-1} is a face of a q-simplex in X', we have that $\sigma^{q-1} \in X'$. Therefore, $\left|\mathsf{cof}_q(A, \sigma^{q-1})\right| = \left|\mathsf{cof}_q(X' \cap A, \sigma^{q-1})\right|$ is an odd number. So we have that $\sigma^{q-1} \in \partial(A)$ and then $\sigma^{q-1} \in X' \cap \partial(A)$. □

Now we prove Proposition 3.3.1. Let $z_b, \ldots, z_{d-1}$ be a sequence of persistent p-cycles for $\left(\alpha_b^p, \alpha_d^p\right)$ as claimed. Note that $[\partial(\sigma_{\beta-1})]$ is the non-zero class in $\mathsf{ker}(\varphi_{\beta-1})$. Therefore, by Definition 3.2.4, $\partial(\sigma_{\beta-1}) \sim z_b$ in K_β. This means that there exists a $(p+1)$-chain $C \subseteq K_\beta$ such that $z_b + \partial(\sigma_{\beta-1}) = \partial(C)$. Let $A_b = C + \sigma_{\beta-1}$; then, $z_b = \partial(A_b)$ where A_b is a $(p+1)$-chain in $K_{\beta-1}$ containing $\sigma_{\beta-1}$. Similarly, we have that $z_{d-1} = \partial(A_d)$ for a $(p+1)$-chain $A_d \subseteq K_{\delta+1}$ containing σ_δ. By Definition 3.2.4, there exists a $(p+1)$-chain $A_\mathrm{i} \subseteq \mathbb{K}^p_{(\mathrm{i}-1,\mathrm{i}+1)}$ for each $b < \mathrm{i} < d$ such that $z_{\mathrm{i}-1} + z_\mathrm{i} = \partial(A_\mathrm{i})$. Thus, $A_b, \ldots, A_d$ are the $(p+1)$-chains satisfying the condition in Claim 2. Let $z'_\mathrm{i} = K' \cap z_\mathrm{i}$ and $A'_\mathrm{i} = K' \cap A_\mathrm{i}$ for each i. By Proposition A.2.2, $z'_b = \partial(A'_b)$. Since A'_b contains $\sigma_{\beta-1}$, it follows that $z'_b + \partial(\sigma_{\beta-1}) = \partial\left(A'_b \setminus \{\sigma_{\beta-1}\}\right)$, where $A'_b \setminus \{\sigma_{\beta-1}\} \subseteq K_\beta$. It is then true that $z'_b \sim \partial(\sigma_{\beta-1})$ in K_β. Now we simulate a run of Algorithm A.1.1 for computing $\mathsf{Pers}_p(\mathcal{F}_p(f))$. Then, at the $(\beta-1)$-th iteration of the run, we can let $z'_b \subseteq K_\beta$ be the representative cycle at index β for the new interval $[\beta, \beta]$.

Let λ be the index of the complex $\mathbb{K}^p_{(b,b+2)}$ in $\mathcal{F}_p(f)$, i.e., $K_\lambda = \mathbb{K}^p_{(b,b+2)}$. In the run of Algorithm A.1.1, the interval starting with β must persist to λ because this interval ends with δ. At any j-th iteration for $\beta \le \mathrm{j} \le \lambda - 2$, other than the case that φ_j is backward with a non-trivial cokernel, the setting of representative cycles for all intervals persisting through follows the trivial setting rule. For the case that φ_j is backward with a non-trivial cokernel, since $z'_b \subseteq K_{\mathrm{j}+1}$, the setting of the representative cycles for the interval $[\beta, \mathrm{j}+1]$ must also follow the trivial setting rule. Hence, at the end of the $(\lambda-2)$-th iteration, $z'_b \subseteq K_{\lambda-1}$ can be the representative cycle at index $\lambda - 1$ for the interval $[\beta, \lambda-1]$. Meanwhile, it is true that $K' \cap (z_b + z_{b+1}) = K' \cap z_b + K' \cap z_{b+1}$. So $z'_b + z'_{b+1} = K' \cap \partial(A_{b+1}) = \partial(K' \cap A_{b+1}) = \partial\left(A'_{b+1}\right)$, which means that $z'_b \sim z'_{b+1}$ in $\mathbb{K}^p_{(b,b+2)} = K_\lambda$. Therefore, $[z'_b] \mapsto [z'_{b+1}]$ by $\varphi_{\lambda-1}$, which means that $z'_{b+1} \subseteq K_\lambda$ can be the representative cycle at index λ for the interval $[\beta, \lambda]$. By repeating the above arguments on each z'_i that follows, we have that $z'_{d-1} \subseteq K_\delta$ can be the representative cycle at index δ for the interval $[\beta, \delta]$. Finally, for contradiction, assume instead that $\sigma_\delta \notin K'$. This means that $\sigma_\delta \notin A'_d$, and hence $A'_d \subseteq K_\delta$. Since $z'_{d-1} = \partial\left(A'_d\right)$, we then have that z'_{d-1} is a boundary in K_δ. However, by Proposition A.2.1, z'_{d-1} cannot be a boundary in K_δ, which is a contradiction. Therefore, Claim 1 is proved. Furthermore, we have that $z'_b, \dots, z'_{d-1}$ and $A'_b, \dots, A'_d$ satisfy the condition in Claim 2.

To prove the last statement of Claim 2, first note that $\partial\Big(\sum_{\mathrm{i}=b}^{d} A'_{\mathrm{i}}\Big) = 0$. Let $A' = \sum_{\mathrm{i}=b}^{d} A'_{\mathrm{i}}$. Since $\sigma_{\beta-1} \in \mathbb{K}^p_{(b-1,b+1)}$ and $\sigma_{\beta-1} \notin \mathbb{K}^p_{(b,b+1)}$, there must be a vertex in $\sigma_{\beta-1}$ with function value in $\big(\alpha^p_{b-1}, \alpha^p_b\big]$. So $\sigma_{\beta-1} \notin \mathbb{K}^p_{(b,d+1)}$, which means that $\sigma_{\beta-1} \notin A'_{\mathrm{i}}$ for any $b < \mathrm{i} \leq d$. We also have that $\sigma_{\beta-1} \in A'_b$, and hence $\sigma_{\beta-1} \in A'$. We then show that A' equals the set of $(p+1)$-simplices of K'. First note that $A' \subseteq K'$. Then, for contradiction, suppose that there is a $(p+1)$-simplex $\sigma \in K'$ not in A'. Since $\sigma \in K'$, there is a $(p+1)$-path $\tau_1, \dots, \tau_\ell$ from σ to $\sigma_{\beta-1}$ in K'. Since $\sigma \notin A'$ and $\sigma_{\beta-1} \in A'$, there must be a j such that $\tau_{\mathrm{j}} \notin A'$ and $\tau_{\mathrm{j}+1} \in A'$. Let τ_{j} and $\tau_{\mathrm{j}+1}$ share a p-face τ^p; then, $\tau^p \in \partial(A')$, contradicting the fact that $\partial(A') = 0$. For the disjointness of $A'_b, \dots, A'_d$, suppose instead that there is a σ residing in more than one of $A'_b, \dots, A'_d$. Then, σ can only reside in two consecutive chains A'_{i} and $A'_{\mathrm{i}+1}$, because pairs of chains of other kinds are disjoint. This implies that $\sigma \notin A'$, contradicting the fact that A' contains all $(p+1)$-simplices of K'. Thus, Claim 2 is proved.

Combining the fact that $\partial(A') = 0$, K' is a pure weak $(p+1)$-pseudomanifold, and Claim 2, we can reach Claim 3.

A.2.4 Proof of Proposition 3.3.5

We first show that there is at least one such component. Let $z_{b-1}, z_b, \dots, z_{d-1}$ be a sequence of persistent p-cycles for $\big[\alpha^p_b, \alpha^p_d\big)$. Then, by definition, there exist $(p+1)$-chains $A_b \subseteq \mathbb{K}^p_{(b-1,b+1)}, \dots, A_{d-1} \subseteq \mathbb{K}^p_{(d-2,d)}, A_d \subseteq K_{\delta+1}$ such that $z_{b-1} + z_b = \partial(A_b), \dots, z_{d-2} + z_{d-1} = \partial(A_{d-1}), z_{d-1} = \partial(A_d)$. Let $A = \sum_{\mathrm{i}=b}^{d} A_{\mathrm{i}}$; then, $\partial(A) = z_{b-1} \subseteq \overline{K}_\beta$. Note that $\sigma_{\beta-1} \in z_{b-1}$ by definition, which implies that $\sigma_{\beta-1}$ is a face of only one $(p+1)$-simplex $\tau \in A$. Note that $\tau \notin \overline{K}_\beta$ by Proposition 3.3.4, which means that $\tau \in \tilde{K} \setminus \overline{K}_\beta$. Let $\mathcal{C}$ be the $(p+1)$-connected component of $\tilde{K} \setminus \overline{K}_\beta$ containing τ. We show that $\mathcal{C} \subseteq A$. For contradiction, suppose instead that there is a $\tau' \in \mathcal{C}$ which is not in A. Since $\tau, \tau' \in \mathcal{C}$, there is a $(p+1)$-path $\tau_1, \dots, \tau_\ell$ from τ to τ' in $\tilde{K} \setminus \overline{K}_\beta$. Also since $\tau_1 \in A$ and $\tau_\ell \notin A$, there must be an ι such that $\tau_\iota \in A$ and $\tau_{\iota+1} \notin A$. Let τ^p be a p-face shared by τ_ι and $\tau_{\iota+1}$ in $\tilde{K} \setminus \overline{K}_\beta$; then, $\tau^p \in \partial(A)$ and $\tau^p \notin \overline{K}_\beta$.

This contradicts $\partial(A) \subseteq \overline{K}_\beta$. Since $\mathcal{C} \subseteq A$, we have that τ is the only $(p+1)$-coface of $\sigma_{\beta-1}$ in $\mathcal{C}$, which means that $\sigma_{\beta-1} \in \partial(\mathcal{C})$. We then show that $\partial(\mathcal{C}) \subseteq \overline{K}_\beta$. For contradiction, suppose instead that there is a $\sigma \in \partial(\mathcal{C})$ which is not in $\overline{K}_\beta$, and let τ' be the only $(p+1)$-coface of σ in $\mathcal{C}$. If σ has only one $(p+1)$-coface in $\tilde{K}$, the fact that $\mathcal{C} \subseteq A$ implies that τ' is the only $(p+1)$-coface of σ in A. Hence, $\sigma \in \partial(A)$, contradicting $\partial(A) \subseteq \overline{K}_\beta$. If σ has another $(p+1)$-coface τ'' in $\tilde{K}$, then τ'' must not be in $\overline{K}_\beta$ because the p-face σ of τ'' is not in $\overline{K}_\beta$. So $\tau'' \in \tilde{K} \setminus \overline{K}_\beta$. Then, $\tau'' \in \mathcal{C}$ because it shares a p-face $\sigma \in \tilde{K} \setminus \overline{K}_\beta$ with $\tau' \in \mathcal{C}$, contradicting the fact that τ' is the only $(p+1)$-coface of σ in $\mathcal{C}$. Now we have constructed a $(p+1)$-connected component $\mathcal{C}$ of $\tilde{K} \setminus \overline{K}_\beta$ whose boundary resides in $\overline{K}_\beta$ and contains $\sigma_{\beta-1}$.

We then prove that there is only one such component. For contradiction, suppose that there are two components $\mathcal{C}_l$, $\mathcal{C}_\mathrm{j}$ among $\mathcal{C}_0, \ldots, \mathcal{C}_k$ whose boundaries contain $\sigma_{\beta-1}$. Then, at least one of $\mathcal{C}_l$, $\mathcal{C}_\mathrm{j}$ does not contain σ_δ. Let $\mathcal{C}_\mathrm{j}$ be the one *not* containing σ_δ. Note that the set $\left\{\zeta^\mathrm{j}_\mathrm{i} \mid b \leq \mathrm{i} < d\right\}$ computed in Step 2 of Algorithm 3.3.2 satisfies that ζ^j_{d-1} is null-homologous in $M_\mathrm{j} \cap K_{\delta+1}$. The fact that $\sigma_\delta \notin M_\mathrm{j}$ implies that ζ^j_{d-1} is also null-homologous in K_δ. Then, similar to the proof for Claim 1 of Proposition 3.3.1, we can derive a representative cycle ζ^j_{d-1} for the interval $[\beta, \delta]$ at index δ which is a boundary, and thus a contradiction.

VITA

Tao Hou joined the Department of Computer Science at Purdue University in August 2020, as a PhD student transferring from another institute. He started his PhD in the Department of Computer Science and Engineering at The Ohio State University in 2016. Prior to that, he got his master's degree from Tsinghua University and bachelor's degree from Beijing Institute of Technology, both in China. He has also worked in the industry as an engineer in China before coming to the U.S.