# Efficient algorithms for computing a minimal homology basis

Tamal K. Dey[1†], Tianqi Li[1‡], and Yusu Wang[1§]

Department of Computer Science and Engineering, The Ohio State University, Columbus

**Abstract.** Efficient computation of shortest cycles which form a homology basis under $\mathbb{Z}_2$-additions in a given simplicial complex $\mathcal{K}$ has been researched actively in recent years. When the complex $\mathcal{K}$ is a weighted graph with $n$ vertices and $m$ edges, the problem of computing a shortest (homology) cycle basis is known to be solvable in $O(m^2n/\log n + n^2m)$-time. Several works [1,2] have addressed the case when the complex $\mathcal{K}$ is a 2-manifold. The complexity of these algorithms depends on the rank $g$ of the one-dimensional homology group of $\mathcal{K}$. This rank $g$ has a lower bound of $\Theta(n)$, where $n$ denotes the number of simplices in $\mathcal{K}$, giving an $O(n^4)$ worst-case time complexity for the algorithms in [1,2]. This worst-case complexity is improved in [3] to $O(n^\omega + n^2g^{\omega-1})$ for general simplicial complexes where $\omega < 2.3728639$ [4] is the matrix multiplication exponent. Taking $g = \Theta(n)$, this provides an $O(n^{\omega+1})$ worst-case algorithm. In this paper, we improve this time complexity. Combining the divide and conquer technique from [5] with the use of annotations from [3], we present an algorithm that runs in $O(n^\omega + n^2g)$ time giving the first $O(n^3)$ worst-case algorithm for general complexes. If instead of minimal basis, we settle for an approximate basis, we can improve the running time even further. We show that a 2-approximate minimal homology basis can be computed in $O(n^\omega\sqrt{n \log n})$ expected time. We also study more general measures for defining the minimal basis and identify reasonable conditions on these measures that allow computing a minimal basis efficiently.

## 1 Introduction

Many applications in science and engineering require computing "features" in a shape that is finitely represented by a simplicial complex. These features sometimes include topological features such as "holes" and "tunnels" present in the shape. A concise definition of these otherwise vague notions can be obtained by considering homology groups and their representative cycles. In particular, a one-dimensional homology basis, that is, a set of independent cycles in the 1-skeleton of the input simplicial complex whose homology classes form a basis for the first homology group, can be taken as a representative of the "holes" and "tunnels" present in the shape. However, instead of any basis, one would like to have a homology basis whose representative cycles are small under some suitable metric, thus bringing the 'geometry' into picture along with topology.

When the input complex is a graph with $n$ vertices and $m$ edges, the homology basis coincides with what is called the cycle basis and its minimality is measured with respect to the total weights of the cycles assuming non-negative weights on the edges. A number of efficient algorithms have been designed to compute such a minimal cycle basis for a weighted graph [1, 5–8]. The best known algorithm for this case runs in $O(m^2n/\log n + n^2m)$ [8].

When the input is a simplicial complex, one dimensional homology basis is determined by the simplices of dimension up to 2. Thus, without loss of generality, we can assume that the complex has dimension at most 2, that is, it consists of vertices, edges, and triangles. The 1-skeleton of the complex is a graph (weighted if the edges are). Therefore, one can consider a minimal cycle basis in the 1-skeleton. However, the presence of triangles makes some of these basis elements to be trivial in the homology basis. Therefore, the computation of the minimal homology basis in a simplicial complex differs from the minimal cycle basis in a graph. In this paper, we show that the efficient algorithms of [5] for computing *a minimal cycle basis* can be adapted to computing *a minimal homology basis* in a simplicial complex (by combining with an algorithm [3] to compute the so-called annotations). In the process we improve the current best time complexity bound for computing a minimal homology basis and also extend these results to more generalized measures.

More specifically, for the special case of a combinatorial 2-manifold with weights on the edges, Erickson and Whittlesey [2] gave an $O(n^2 \log n + gn^2 + g^3n)$-time algorithm to compute a minimal homology basis where $n$ is the total number of simplices and $g$ is the rank of the first homology group. Dey et al. [9] and Chen and Friedman [10] generalized the results above to arbitrary simplicial complexes. Busaryev et al. [3] improved the running time of this generalization from $O(n^4)$ [9] to $O(n^\omega + n^2g^{\omega-1})$ where $\omega < 2.3728639$ [4] is the matrix multiplication exponent. This gives the best known $O(n^{1+\omega})$ worst-case time algorithm when $g = \Theta(n)$. In Section 3, combining the divide

† tamaldey@cse.ohio-state.edu
‡ li.6108@osu.edu
§ yusu@cse.ohio-state.edu

and conquer approach of [5] with the use of annotations [3], we develop an improved algorithm to compute a minimal 1-dimensional homology basis for an arbitrary simplicial complex in only $O(n^2g + n^\omega)$ time. Considering $g = \Theta(n)$, this gives the first $O(n^3)$ worst-case time algorithm for the problem.

We can further improve the time complexity if we allow for approximations. An algorithm to compute a 2-approximate minimal homology basis are given in Section 4 running in $O(n^\omega\sqrt{n\log n})$ expected time.

All of the above algorithms operate by computing a set of candidate cycles that necessarily includes at least one minimal homology basis and then selecting one of these minimal bases. The standard proof [2] of the fact that the candidate set includes a minimal basis uses the specific distance function based on the shortest path metric and a size function that assigns total weight of the edges in a cycle as its size. In Section 5, we identify general conditions for the distance and size function so that the divide and conquer algorithm still works without degrading in time complexity. This allows us to consider distance function beyond the shortest path metric and the size function beyond the total weight of edges as we illustrate with two examples. Specifically, we can now compute a minimal homology basis whose size is induced by a general map $F : \mathcal{K} \to Z$ for any metric space $Z$.

## 2 Background and notations

In this paper, we are interested in computing a minimal basis for the 1-dimensional homology group of a simplicial complex over the field $\mathbb{Z}_2$. In this section we briefly introduce some relevant concepts here; the details appear in standard books on algebraic topology such as [11].

*Homology.* Let $\mathcal{K}$ be a connected simplicial complex. A $d$-chain $c$ is a formal sum, $c = \sum a_i\sigma_i$ where the $\sigma_i$s are the $d$-simplices of $\mathcal{K}$ and the $a_i$s are the coefficients with $a_i \in \mathbb{Z}_2$. We use $\mathsf{C}_d$ to denote the group of $d$-chains which is formed by the set of $d$-chains together with the addition. Note that there is a one-to-one correspondence between the chain group $\mathsf{C}_d$ and the family of subsets of $\mathcal{K}_d$ where $\mathcal{K}_d$ is the set of all $d$-simplices. Thus $\mathsf{C}_d$ is isomorphic to the space $(\mathbb{Z}_2)^{n_d}$ where $n_d$ is the number of $d$-simplices in $\mathcal{K}$. Naturally all $d$-simplices in $\mathcal{K}$ form a basis of $\mathsf{C}_d$ in which the $i$-th bit of the coordinate vector of a $d$-chain indicates whether the corresponding $d$-simplex appears in the chain.

The boundary of a $d$-simplex is the sum of all its $(d-1)$-faces. This can be interpreted and extended to a $d$-chain as a *boundary map* $\partial_d : \mathsf{C}_d \to \mathsf{C}_{d-1}$, where the boundary of a chain is defined as the sum of the boundaries of its simplices. A $d$-cycle $c$ is a $d$-chain with empty boundary, $\partial_d c = 0$. Since $\partial_d$ commutes with addition, we have the group of $d$-cycles, $\mathsf{Z}_d$, which is the kernel of $\partial_d$, $\mathsf{Z}_d := ker\partial_d$. A $d$-boundary $c$ is a $d$-chain that is the boundary of a $(d+1)$-chain, $c = \partial_{d+1}b$ for some $b \in \mathsf{C}_{d+1}$. The group of $d$-boundaries $\mathsf{B}_d$ is the image of $\partial_{d+1}$, that is, $\mathsf{B}_d := im\partial_{d+1}$. Notice that $\mathsf{B}_d$ is a subgroup of $\mathsf{Z}_d$. Hence we can consider the quotient $\mathsf{Z}_d/\mathsf{B}_d$ which constitutes the $d$-dimensional homology group denoted as $\mathsf{H}_d$. Each element in $\mathsf{H}_d$, called a homology class, is an equivalence class of $d$-cycles whose difference is always in $\mathsf{B}_d$. Two cycles are said to be *homologous* if they are in the same homology class.

Under $\mathbb{Z}_2$ coefficients, the groups $\mathsf{C}_d$, $\mathsf{Z}_d$, $\mathsf{B}_d$ and $\mathsf{H}_d$ are all vector spaces. A basis of a vector space is a set of vectors of minimal cardinality that generates the entire vector space. We are concerned with the homology bases of $\mathsf{H}_d$ and particularly in $\mathsf{H}_1$ (more formally below). We use $L = rank(\mathsf{Z}_1)$ to denote the dimension of vector space $\mathsf{Z}_1$ and use $g = rank(\mathsf{H}_1)$ to denote the *1-st Betti number* of $\mathcal{K}$, which is the dimension of vector space $\mathsf{H}_1$.

- A set of cycles $C_1, \cdots, C_L$, with $L = rank(\mathsf{Z}_1)$, that generates the cycle space $\mathsf{Z}_1$ is called its *cycle basis*.
- For any 1-cycle $c$, let $[c]$ denote its homology class. A set of homology classes $\{[C_1], ..., [C_g]\}$ that constitutes a basis of $\mathsf{H}_1$ is called a *homology basis*. For simplicity, we also say a set of cycles $\{C_1, C_2, \cdots, C_g\}$ is a homology basis if their corresponding homology classes $[C_1], [C_2], \cdots, [C_g]$ form a basis for $\mathsf{H}_1(\mathcal{K})$.
- Let $\mu : \mathsf{Z}_1 \to \mathbb{R}^+ \cup \{0\}$ be a size function that assigns a non-negative weight to each cycle $C \in \mathsf{Z}_1$. A cycle or homology basis $C_1, \cdots, C_l$ is called *minimal* if $\sum_{i=1}^{l} \mu(C_i)$ is minimal among all bases of $\mathsf{Z}_1$ $(l = L)$ or $\mathsf{H}_1(\mathcal{K})$ $(l = g)$ respectively.

*Annotation.* To compute a minimal homology basis of a simplicial complex $\mathcal{K}$, it is necessary to have a way to represent and distinguish homology classes of cycles. Annotated simplices have been used for this purpose in earlier works: For example, Erickson and Wittlesey [2] and Borradaile et al. [1] used them for computing optimal homology cycles in surface embedded graphs. Here we use a version termed as *annotation* from [3] which gives an algorithm to compute them in matrix multiplication time for general simplicial complexes. An annotation for a $d$-simplex is a $g$-bit binary vector, where $g = rank(\mathsf{H}_d(\mathcal{K}))$. The annotation of a cycle $z$, which is the sum of annotations of all simplices in $z$, provides the coordinate vector of the homology class of $z$ in a pre-determined homology basis. More formally,

**Definition 2.1 (Annotation).** *Let $\mathcal{K}$ be a simplicial complex and $\mathcal{K}_d$ be the set of d-simplices in $\mathcal{K}$. An annotation for d-simplices is a function $a : \mathcal{K}_d \to (\mathbb{Z}_2)^g$ with the following property: any two d-cycles $z$ and $z'$ are homologous if and only if*

$$\sum_{\sigma\in z} a(\sigma) = \sum_{\sigma\in z'} a(\sigma)$$

*Given an annotation $a$, the annotation of any d-cycle $z$ is defined by $a(z) = \sum_{\sigma\in z} a(\sigma)$.*

**Proposition 2.1 ( [3])** *There is an algorithm that annotates the d-simplices in a simplicial complex with $n$ simplices in $O(n^\omega)$ time.*

## 3 Minimal homology basis

In this section, we describe an efficient algorithm to compute a minimal homology basis of the 1-dimensional homology group $\mathsf{H}_1(\mathcal{K})$. The algorithm uses the divide and conquer technique from [5] where they compute a minimal *cycle* basis in a weighted graph. The authors in [1] adapted it for computing optimal homology basis in surface embedded graphs. We adapt it here to simplicial complexes using edge annotations [3].

More specifically, let $\mathcal{K}$ be a simplicial complex with $n$ simplices – Since we are only interested in 1-dimensional homology basis, it is sufficient to consider all simplices with dimension up to 2, namely vertices, edges, and triangles. Hence we assume that $\mathcal{K}$ contains only simplices of dimension at most 2. Assume that the edges in $\mathcal{K}$ are weighted with non-negative weights. Given any homology basis $\{C_1, \ldots, C_g\}$ where $g = rank(\mathsf{H}_1(\mathcal{K}))$, we define the *size* $\mu(C)$ *of a cycle* $C \in Z_1(\mathcal{K})$ as the *total weights* of its edges. As defined in Section 2, the problem of computing a minimal homology basis of $\mathsf{H}_1$ is now to find a basis $\mathcal{C} = \{C_1, C_2, \cdots, C_g\}$ such that the sum of $\sum_{i=1}^{g} \mu(C_i)$ is the smallest.

The high-level algorithm to compute such a minimal homology basis of $\mathsf{H}_1$ group proceeds as follows. First, we need to annotate all 1-simplices implemented by the algorithm of [3]. Then we compute a candidate set of cycles which includes a minimal homology basis. At last, we extract such a minimal homology basis from the candidate set.

*Candidate set.* We now describe the step to compute a candidate set $\mathcal{G}$ of cycles that contains a minimal homology basis. We use the shortest path tree approach which dates back to Horton's algorithm for a minimal cycle basis of a graph [7]. It was also applied in other earlier works, e.g. [2, 9]. We first generate a candidate set $\mathcal{G}(p)$ for every vertex $p \in vert(\mathcal{K})$, where $vert(\mathcal{K})$ is the set of vertices of $\mathcal{K}$. Then we take the union of all $\mathcal{G}(p)$ and denote as $\mathcal{G}$, i.e. $\mathcal{G} = \cup_{p\in vert(\mathcal{K})}\mathcal{G}(p)$. To compute $\mathcal{G}(p)$, first we construct a shortest path tree $T_p$ rooted at $p$. Let $\Pi_p(u, v)$ denote the unique path connecting two vertices $u$ and $v$ in $T_p$. Then each nontree edge $e = (u, v)$ generates a cycle $C(p, e) = e \circ \Pi_p(u, v)$. The union of all such cycles constitutes the candidate set of the vertex $p$, i.e. $\mathcal{G}(p) = \cup_{e\in edge(\mathcal{K})\setminus E_p} C(p, e)$ where $E_p$ is the set of tree edges in $T_p$. Note that the number of cycles in $\mathcal{G}(p)$ is $O(n)$ for each vertex $p \in vert(\mathcal{K})$. Hence there are $O(n^2)$ candidate cycles in $\mathcal{G}$ in total. They, together with their sizes, can be computed in $O(n^2 \log n)$ time.

**Proposition 3.1 ( [2, 9]).** *The candidate set $\mathcal{G}$ has $O(n^2)$ cycles and admits a minimal homology basis.*

### 3.1 Computing a minimal homology basis

What remains is to compute a minimal homology basis from the candidate set $\mathcal{G}$. To achieve it, we modify the divide and conquer approach from [5] which improved the algorithm of [6] for computing a minimal cycle basis of a graph with non-negative weights.

This approach uses an auxiliary set of support vectors [5] that helps select a minimal homology basis from a larger set containing at least one minimal basis; in our case, this larger set is $\mathcal{G}$.

A support vector $S$ is a vector in the space of $g$-dimensional binary vectors $\mathcal{S} = \{0, 1\}^g$. The use of support vectors along with annotations requires us to perform more operations without increasing the complexity of the divide and conquer approach. Let $a(C)$ denote the annotation of a cycle $C$. First, we define the function:

$$m : \mathcal{S} \times \mathcal{G} \to \{0, 1\} \text{ with } m(S, C) = \langle S, a(C)\rangle \text{ where } \langle\cdot,\cdot\rangle \text{ is the inner product over } \mathbb{Z}_2.$$

We say a cycle $C$ is *orthogonal* to a support vector $S_i$ if $m(S_i, C) = 0$ and is *non-orthogonal* if $m(S_i, C) = 1$. We would choose cycles $C_1, \cdots, C_g$ iteratively from a set guaranteed to contain a minimal homology basis and add them to the minimal homology basis. During the procedure, the algorithm always maintains a set of support vectors $S_1, S_2, \cdots, S_g$ with the following properties:

(1). $S_1, S_2, \cdots, S_g$ form a basis of $\{0,1\}^g$.
(2). If $C_1, C_2, \cdots, C_{i-1}$ have already been computed, $m(S_i, C_j) = 0$, $j < i$.

Suppose that in addition to properties (1) and (2), we have the following additional condition to choose $C_i$s, then the set $C_1, C_2, \ldots, C_g$ constitutes a minimal homology basis.

(3). If $C_1, C_2, \cdots, C_{i-1}$ have already been computed, $C_i$ is chosen so that $C_i$ is the shortest cycle with $m(S_i, C_i) = 1$.

If we keep the same support vectors, after we select a new cycle $C_i$, $m(S_{i+1}, C_i) = 0$ may not hold which means the property (2) may not hold. Therefore, we update the support vectors $S_{i+1}, \cdots, S_g$ after computing $C_i$ so that the orthogonality condition (2) holds. If chosen with condition (3), the cycle $C_i$ becomes independent of the cycles previously chosen as stated below:

**Claim 3.1** *For any $i \le g$, if property (1) and (2) hold, then for any cycle $C$ with $m(S_i, C) = 1$, $[C]$ is independent of $[C_1], [C_2], \cdots, [C_{i-1}]$.*

*Proof.* By property (2), $\forall j < i, m(S_i, C_j) = 0$. If $[C]$ is not independent of $[C_1], [C_2], \cdots, [C_{i-1}]$, then the annotation $a(C)$ of the cycle $C$ can be written as $a(C) = \sum_{j<i} \alpha_j a(C_j)$, where $\alpha_j \in \{0,1\}$ and at least one $\alpha_j = 1, j < i$. Since $m(S_i, C_i) = 1$, we have $\sum_{j<i} \alpha_j m(S_i, C_j) = 1$. It follows that there exists at least one $C_j$, $j < i$, with $m(S_i, C_j) = 1$, which contradicts with property (2). Therefore, $[C]$ is independent of $[C_1], [C_2], \cdots, [C_{i-1}]$. □

The following theorem guarantees that the above three conditions suffice for a minimal homology basis. Its proof is almost the same as the proof of [5, Theorem 1] (which draws upon the idea of [6]).

**Theorem 3.1.** *The set $\{C_1, C_2, \cdots, C_g\}$ computed by maintaining properties (1), (2) and (3) is a minimal homology basis.*

Taking advantage of the above theorem, we aim to compute a homology basis iteratively while maintaining conditions (1), (2), and (3).

**Maintaining support vectors and computing shortest cycles.** Now we describe the algorithm CycleBasis($\mathcal{G}$) (given in Algorithm 1) that computes a minimal homology basis. In this algorithm, we first initialize each support vector $S_i$ so that only the $i$-th bit is set to 1. Then the main computation is done by calling the procedure ExtendBasis$(1, g)$.

**Algorithm 1** Computing a minimal Basis

```
1: procedure CycleBasis($\mathcal{G}$)
2:     for $i \leftarrow 1$ to $g$ do
3:         Initialize $S_i \leftarrow \{e_i\}$, which means that the $i$-th bit of $S_i$ is 1 while others are 0
4:     end for
5:     ExtendBasis$(1, g)$ to get a minimal homology basis $\{C_1, \cdots, C_g\}$
6: end procedure
```

Here the procedure ExtendBasis($i$, $k$) (Algorithm 2) is recursive which extends the current partial basis $\{C_1, \cdots, C_{i-1}\}$ by adding $k$ new cycles. It modifies a divide and conquer approach of [5] to maintain properties (1), (2), and (3). It calls a routine Update to maintain orthogonality using annotations. For choosing the shortest cycle satisfying condition (3), it calls ShortestCycle($S_i$) in the base case ($k = 1$)(See line 3 of Algorithm 2). We describe the recursion and the base case below.

*Recursion.* At the high level, the procedure ExtendBasis($i$, $k$) recurses on $k$ by first calling itself to obtain the next $\lfloor k/2 \rfloor$ cycles in the minimal homology basis in which the support vectors $S_i, S_{i+1}, \cdots, S_{i+\lfloor k/2 \rfloor - 1}$ are updated. Then it calls the procedure Update($i$, $k$) to maintain the orthogonality property (2). It uses the already updated support vectors $S_i, \cdots, S_{i+\lfloor k/2 \rfloor - 1}$ to update $\{S_{i+\lfloor k/2 \rfloor}, ..., S_{i+k-1}\}$ so that $m(S_l, C_j) = 0, \forall j < i + \lfloor k/2 \rfloor, i + \lfloor k/2 \rfloor \le l \le i + k - 1$. At last the procedure ExtendBasis($i$, $k$) calls itself ExtendBasis($i + \lfloor k/2 \rfloor$, $\lceil k/2 \rceil$) to extend the basis by $\lceil k/2 \rceil$ elements.

We describe Update($i$, $k$) and spare giving its pseudocode. Let $\{\hat{S}_{i+\lfloor k/2 \rfloor}, ..., \hat{S}_{i+k-1}\}$ denote the desired output vectors after the update. To ensure the property (1) and (2), we will enforce that the vector $\hat{S}_j$ is of the form

**Algorithm 2** Extend the Basis by k elements

```
1: procedure ExtendBasis($i$, $k$)
2:     if $k = 1$ then
3:         Call ShortestCycle($S_i$) to compute the shortest cycle $C_i$ which is non-orthogonal to $S_i$
4:     else
5:         Call ExtendBasis($i$, $\lfloor k/2 \rfloor$) to extend the homology basis by $\lfloor k/2 \rfloor$ elements. After calling, $S_i, ..., S_{i+\lfloor k/2 \rfloor -1}$ will be updated.
6:         Call Update($i$, $k$) to update the support vectors $\{S_{i+\lfloor k/2 \rfloor}, ..., S_{i+k-1}\}$ using $\{S_i, ..., S_{i+\lfloor k/2 \rfloor -1}\}$ and update the value $m(S_j, e)$ for $i + \lfloor k/2 \rfloor \leq j \leq i + k - 1$ and every edge $e$.
7:         Call ExtendBasis($i + \lfloor k/2 \rfloor$, $\lceil k/2 \rceil$) to extend the cycle basis by $\lceil k/2 \rceil$ elements
8:     end if
9: end procedure
```

$\hat{S}_j = S_j + \sum_{t=1}^{\lfloor k/2 \rfloor} \alpha_{jt} S_{i+t-1}$ where $i + \lfloor k/2 \rfloor \leq j \leq i+k-1$. We just need to determine the coefficients $\alpha_{j1}, ..., \alpha_{j\lfloor k/2 \rfloor}$ so that $m(\hat{S}_j, C_t) = 0$ where $i + \lfloor k/2 \rfloor \leq j \leq i + k - 1$ and $i \leq t \leq i + \lfloor k/2 \rfloor - 1$. We will also compute $m(S_j, e)$ for $i + \lfloor k/2 \rfloor \leq j \leq i + k - 1$ and every edge $e$ where $m(S_j, e)$ is defined as the standard inner product of $S_j$ and $a(e)$ under $\mathbb{Z}_2$, which is important later when we compute the shortest cycle orthogonal to a support vector $S$ in the procedure ShortestCycle($S$).

Now let

$$X = \begin{pmatrix} S_i \\ S_{i+1} \\ \vdots \\ S_{i+\lfloor k/2 \rfloor -1} \end{pmatrix} \cdot \left( a(C_i)^T \; a(C_{i+1})^T \; \cdots \; a(C_{i+\lfloor k/2 \rfloor -1})^T \right)$$

$$Y = \begin{pmatrix} S_{i+\lfloor k/2 \rfloor} \\ S_{i+\lfloor k/2 \rfloor +1} \\ \vdots \\ S_{i+k-1} \end{pmatrix} \cdot \left( a(C_i)^T \; a(C_{i+1})^T \; \cdots \; a(C_{i+\lfloor k/2 \rfloor -1})^T \right),$$

where recall that $g$-bit vector $a(C)$ is the annotation of a cycle $C$. Let $A$ denote a $\lceil k/2 \rceil \times \lfloor k/2 \rfloor$ matrix where row $j$ contains the bit $\alpha_{j+i+\lfloor k/2 \rfloor -1, 1}, \cdots, \alpha_{j+i+\lfloor k/2 \rfloor -1, \lfloor k/2 \rfloor}$. It is not difficult to see that $AX + Y = 0$, and that $X$ is invertible, which means that $A = -YX^{-1} = YX^{-1}$ since the computations are under $\mathbb{Z}_2$.

The next step is to update the value $m(S_j, e)$ to $m(\hat{S}_j, e)$ for every edge $e$ in $\mathcal{K}$, and $i + \lfloor k/2 \rfloor \leq j \leq i + k - 1$. Note that the coefficients $\alpha_{jt}$ are now known and the updated vectors are $\hat{S}_j = S_j + \sum_{t=1}^{\lfloor k/2 \rfloor} \alpha_{jt} S_{i+t-1}$, $i + \lfloor k/2 \rfloor \leq j \leq i + k - 1$. Thus for every edge $e$, $m(\hat{S}_j, e) = m(S_j + \sum_{t=1}^{\lfloor k/2 \rfloor} \alpha_{jt} S_{i+t-1}, e) = m(S_j, e) + \sum_{t=1}^{\lfloor k/2 \rfloor} \alpha_{jt} m(S_{i+t-1}, e)$, $i + \lfloor k/2 \rfloor \leq j \leq i+k-1$. Let $n_1$ be the number of edges in $\mathcal{K}$ and $U$ be the $\lceil k/2 \rceil \times n_1$ matrix where its $(t, j)$ entry is $m(\hat{S}_{i+\lfloor k/2 \rfloor +t-1}, e_j)$. Set $W = [A|I]$ where $I$ is the $\lceil k/2 \rceil \times \lceil k/2 \rceil$ identity matrix. Let $Z$ be a $k \times n_1$ matrix whose $(s, t)$ entry is $m(S_{i+s-1}, e_t)$. Thus we have $U = WZ$. Since the $\lceil k/2 \rceil \times k$ matrix $W$ and $k \times n_1$ matrix $Z$ are already known, the matrix $U$ can be computed in $O(nk^{\omega-1})$ time by chopping $Z$ to $n_1/k$ number of $k \times k$ submatrices and performing $O(n_1/k)$ matrix multiplications of two $O(k) \times O(k)$ size matrices. After that, $m(\hat{S}_j, e)$ can be easily retrieved from the matrix $U$ in constant time.

*Base case for selecting a shortest cycle.* We now implement the procedure ShortestCycle($S_i$) for the base case to compute the shortest cycle $C_i$ non-orthogonal to $S_i$, i.e. the shortest cycle $C_i$ satisfying $m(S_i, C_i) = 1$. We assign a label $l_p(u)$ to each vertex $u$ and $p$. Labeling has been used to solve many graph related problems previously [2,3,8].

Given a vertex $p$ and the shortest path tree $T_p$ rooted at $p$, let $\Pi_p(u)$ for any vertex $u \in \mathit{vert}(\mathcal{K})$ denote the unique tree path in $T_p$ from $p$ to $u$, and let $l_p(u)$ denote the value $m(S_i, \Pi_p(u))$. Let $w$ denote the parent of $u$ in tree $T_p$ and $e_{uw}$ denote the edge between $u$ and $w$. Then $l_p(u) = l_p(w) + m(S_i, e_{uw})$. Thus for a fixed $p \in \mathit{vert}(\mathcal{K})$, we can traverse the tree $T_p$ from the root to the leaves and compute the label $l_p(u)$ for all vertices $u$ in $O(n)$ time as $m(S_i, e)$ for every edge is already precomputed earlier in the procedure Update and can be queried in O(1) time. Thus the total time to compute labels $l_p(u)$ for all $p, u \in \mathit{vert}(\mathcal{K})$ is $O(n^2)$.

Now given a fixed vertex $p$ and the shortest path tree $T_p$, we consider every cycle $C(p, e)$, where $e = (u, v)$ is a non-tree edge. We partition the cycle into three parts: the tree path $\Pi_p(u)$, the tree path $\Pi_p(v)$ and the edge $e$. Thus $m(S_i, C(p, e)) = m(S_i, \Pi_p(u)) + m(S_i, \Pi_p(v))) + m(S_i, e) = l_p(u) + l_p(v) + m(S_i, e)$, which can be computed in $O(1)$ time as all labels are precomputed. Note that there are $O(n^2)$ cycles in the candidate set $\mathcal{G}$ to be computed. It results that in $O(n^2)$ total time, one can compute $m(S_i, C)$ for all cycles $C \in \mathcal{G}$ and find the smallest one.

### 3.2 Correctness and time complexity

To prove the correctness of Algorithm 1, it is crucial to guarantee that the support vectors $S_i$s and the cycles $C_i$s satisfy the desirable properties. First, the set of support vectors $\{S_1, S_2, \cdots, S_g\}$ is a basis of $\{0,1\}^g$ because of the construction of $\hat{S}_i$s in the procedure UPDATE. The property that $\forall j < i$, $m(S_i, C_j) = 0$ holds, because the procedure UPDATE ensures that $S_i$ is taken as a non-trivial solution to a set of linear equations $m(x, C_j) = 0$, $1 \leq j \leq i-1$, which always admits at least one solution. Similarly, for any $i \leq g$, there exists at least one cycle $C$ such that the equation $m(S_i, C) = 1$ holds since both $S_1, \ldots, S_i$ and $C_1, \ldots, C_{i-1}$ at this point only form partial basis of a space with dimension $g$. In the base case, SHORTESTCYCLE computes this cycle $C$ satisfying exactly this property. Then, Theorem 3.1 ensures the correctness of the algorithm.

The total running time of our algorithm is $O(n^2 g + n^\omega)$ and the analysis is as follows. The time to annotate edges and construct the candidate set is $O(n^\omega + n^2 \log n) = O(n^\omega)$ from Proposition 2.1 and 3.1. When computing the basis, the time of the procedure CYCLEBASIS is dominated by the time of EXTENDBASIS. For each $i \leq g$, the time complexity of EXTENDBASIS$(i,k)$ is bounded by the following recurrence:

$$T(i,k) = \begin{cases} \text{the time of } \textsc{ShortestCycle}(S_i) & k = 1 \\ 2T(\cdot, k/2) + O(k^{\omega-1} n) & k > 1 \end{cases}$$

Note that in the recursion, only the second parameter $k$ counts for the time complexity. Actually for each $i \leq g$, the time complexity of SHORTESTCYCLE$(S_i)$ in the base case is only $O(n^2)$ as we argued earlier, that is, $T(\cdot, 1) = O(n^2)$. Then the recurrence solves to $T(\cdot, k) = O(k(n^2) + k^{\omega-1} n)$. It follows that $T(1, g) = O(n^2 g + g^{\omega-1} n)$. Combined with the time for computing annotations and constructing the candidate set, the time complexity is $O(n^2 g + n^\omega)$.

## 4 An approximate minimal homology basis of $\mathsf{H}_1(\mathcal{K})$

In this section, we present an algorithm to compute an approximate minimal 1-dimensional homology basis, where the approximation is defined as follows.

**Definition 4.1 (Approximate minimal homology basis).** *Suppose $\mathcal{C}^*$ is a minimal homology basis for $\mathsf{H}_1(\mathcal{K})$, and let $\ell_1^* \leq \ell_2^* \leq \cdots \leq \ell_g^*$ denote the sequence of sizes of cycles in $\mathcal{C}^*$ sorted in non-decreasing order. A set of $g$ cycles $\mathcal{C}'$ is a* c-approximate minimal homology basis *for $\mathsf{H}_1(\mathcal{K})$ if (i) $\{[C], C \in \mathcal{C}'\}$ form a basis for $\mathsf{H}_1(\mathcal{K})$; and (ii) let $\ell_1, \ldots, \ell_g$ denote the sequence of sizes of cycles in $\mathcal{C}'$ in non-decreasing order, then for any $i \in [1, g]$, $\ell_i^* \leq \ell_i \leq c \cdot \ell_i^*$.*

In what follows, we provide a 2-approximation algorithm running in $O(n^\omega \sqrt{n \log n})$ time. At the high level, we first compute a set of candidate set $\mathcal{G}'$ of cycles that guarantees to contain a 2-approximate minimal homology basis. We then extract a 2-approximate basis from the candidate set $\mathcal{G}'$.

First, we explain the construction of a candidate set of cycles. Recall that in Section 3.1, we compute $O(n^2)$ candidate cycles, each of which has the form $C(p, e)$, formed by $e$ together with the two tree-paths from root $p$ to each of the endpoint of $e$ within the shortest path tree $T_p$. We now apply the algorithm by Kavitha et al. [12] which can compute a smaller candidate set $\mathcal{G}'$ of $O(n\sqrt{n \log n})$ cycles which is guaranteed to contain a 2-approximate minimal *cycle basis* (not homology basis) for graph $\mathcal{K}^{(1)}$ (i.e, 1-skeleton of the complex $\mathcal{K}$) in $O(n\sqrt{n} \log^{3/2} n)$ *expected time*. Here, a cycle basis $\Gamma = \{\gamma_1, \ldots, \gamma_L\}$ of the graph $G = \mathcal{K}^{(1)}$ where $L = rank(\mathsf{Z}_1)$ is simply a set of cycles such that any other cycle from $G$ can be represented uniquely as a linear combination of cycles in $\Gamma$. A *minimal cycle basis* is a cycle basis $\Gamma^*$ whose total weight $\sum_{\gamma \in \Gamma^*} \mu(\gamma)$ is smallest among all cycle basis. A cycle basis $\Gamma$ is a $c$-approximate minimal cycle basis if its total weight is at most $c$ times that of the minimal cycle basis, i.e, at most $c \cdot \sum_{\gamma \in \Gamma^*} \mu(\gamma)$.

Now let the size $\mu(\gamma)$ of a cycle be the total weight of all edges in $\gamma$. Then, it turns out that, $\mathcal{G}'$ not only contains a 2-approximate minimal cycle basis w.r.t. this size, it also satisfies the following stronger property as proven in [12].

**Proposition 4.1 ( [12, Lemma 6.3]).** *There exists a minimal cycle basis $\Gamma^* = \{\gamma_1^*, \ldots, \gamma_L^*\}$ such that, for any $1 \leq i \leq L$, there is a subset $\Gamma_i \subseteq \mathcal{G}'$ of the computed candidate set $\mathcal{G}'$ so that (i) $\gamma_i^* = \sum_{\gamma \in \Gamma_i} \gamma$ and (ii) each cycle in $\Gamma_i$ has size at most $2\mu(\gamma_i^*)$.*

Next, we prove that a candidate set $\mathcal{G}'$ satisfying conditions in Proposition 4.1 is guaranteed to also contain a 2-approximate minimal homology basis. We remark that if Proposition 4.1 does not hold, then the sole condition that $\mathcal{G}'$ contains a $c$-approximate minimal *cycle basis* is **not** sufficient to guarantee that it also contains a $c$-approximate minimal *homology basis* for any constant $c$. A counter-example is given at the end of this section.

**Lemma 4.1.** *Given a set $\mathcal{G}'$ of cycles satisfying Proposition 4.1, there exists a minimal homology basis $\mathcal{C}^* = \{C_1^*, \ldots, C_g^*\}$ such that $\mathcal{G}'$ contains $g$ cycles $A_1, \cdots, A_g$ with (i). $[A_1], \cdots, [A_g]$ form a homology basis, and (ii) $\mu(A_i) \leq 2\mu(C_i^*)$, for $i = 1, \cdots, g$.*

*Proof.* Let $\Gamma^*$ be a minimal homology basis which satisfies Proposition 4.1. It is known that it contains a minimal homology basis, which we set as $\mathcal{C}^* = \{C_1^*, \ldots, C_g^*\}$. Now by Proposition 4.1, for each $C_i^*$, there exists a subset $\Gamma_i \subseteq \mathcal{G}'$ such that $C_i^* = \sum_{\gamma \in \Gamma_i} \gamma$ and $\mu(\gamma) \leq 2\mu(C_i^*)$, $\forall \gamma \in \Gamma_i$. Assume w.l.o.g. that cycles in $\mathcal{C}^*$ are in non-decreasing order of their sizes. We now prove the lemma inductively. In particular,

Claim-A: For any $k$, we show that there exists $A_1, \ldots, A_k \in \bigcup_{r \leq k} \Gamma_r$ such that for each $i \in [1, k]$, (Cond-1) $\mu(A_i) \leq 2\mu(C_i^*)$; and (Cond-2) $[A_1], \ldots, [A_k]$ are independent.

The base case is straightforward: We can simply take $A_1$ as any cycle from $\Gamma_1$ that is not null-homologous (which must exist as $C_1^* = \sum_{\gamma \in \Gamma_1} \gamma$ is not null-homologous).

Now suppose the claim holds for $k$. Consider the case for $k+1$. By induction hypothesis, there exists $A_1, \ldots A_k \in \bigcup_{r \leq k} \Gamma_r$ such that (Cond-1) and (Cond-2) hold. Now consider cycles in $\bigcup_{r \leq k+1} \Gamma_r$. Let $\mathcal{H}_{k+1}$ denote the subgroup of $\mathsf{H}_1(\mathcal{K})$ generated by the homology classes of all cycles in $\bigcup_{r \leq k+1} \Gamma_r$. Note that $\mathcal{H}_{k+1}$ spans $\{[C_1^*], \ldots, [C_{k+1}^*]\}$, then the rank of $\mathcal{H}_{k+1}$ is at least $k+1$, which means there always exists a cycle $A_{k+1} \in \bigcup_{r \leq k+1} \Gamma_r$ such that $[A_{k+1}]$ is independent of $[A_1], \ldots [A_k]$. By definition of $\bigcup_{r \leq k+1} \Gamma_r$, there is an index $j \leq k+1$ such that $\mu(A_{k+1}) \leq \mu(C_j^*) \leq \mu(C_{k+1}^*)$ which satisfies both (Cond-1) and (Cond-2). Thus Claim-A holds for $k+1$ as well.

The lemma then follows when $k = g$. □

So far we have proved that the new candidate set $\mathcal{G}'$ always contains a 2-approximate minimal homology basis. What remains is to describe how to compute such an approximate basis from the candidate set $\mathcal{G}'$. First, as in Algorithm **??**, we compute the annotation of all edges in $O(n^\omega)$ time. Let $a(e)$ denote the annotation of an edge $e \in \mathcal{K}^{(1)}$ in the complex $\mathcal{K}$; recall that $a(e)$ is a $g$-bit vector with $g = rank(\mathsf{H}_1(\mathcal{K}))$. Also recall that given a cycle $\gamma$, its annotation $a(\gamma) = \sum_{e \in \gamma} a(e)$ represents the homology class of this cycle, and two cycles are homologous if and only if they have the same annotation vectors.

Now order the cycles in $\mathcal{G}' = \{\gamma_1, \ldots, \gamma_m\}$, where $m = |\mathcal{G}'| = O(n\sqrt{n \log n})$, in non-decreasing order of their sizes. We will compute the annotation of all cycles in $\mathcal{G}'$ and put them in the $g \times m$ matrix $M$, whose $i$-th column $M[i]$ represents the annotation vector for the cycle $\gamma_i$. Since $\mathcal{G}'$ contains a homology basis of $\mathsf{H}_1(\mathcal{K})$ ( Lemma 4.1), $rank(M) = g$.

First, we explain how to compute annotation matrix $M$ efficiently. Let $edge(\mathcal{K}) = \{e_1, \ldots, e_L\}$ denote all edges from $\mathcal{K}$. Let $A$ denote the $L \times m$ matrix where $\gamma_i = \sum_{j \in [1,L]} A[i][j] e_j$; that is, non-zero entries of the $i$-th column $A[i]$ encode all edges in the cycle $\gamma_i$. Let $B$ denote the $g \times L$ matrix where the $i$-th column $B[i]$ encodes the annotation of edge $e_i$. It is easy to see that $M = A^T \cdot B^T$. Instead of computing the multiplication directly, we partition the matrix $A^T$ top-down into $m/L$ submatrices each of size at most $L \times L$. For each of this submatrix, its multiplication with $B^T$ can be done in $O(L^\omega)$ matrix multiplication time. Thus the total time to compute the multiplication $M = A^T \cdot B^T$ takes $O(\frac{m}{L} L^\omega) = O(mn^{\omega-1})$ time as $L \leq n$. In other words, we can compute the annotation matrix $M$ in $O(n^\omega \sqrt{n \log n})$ as $m = O(n\sqrt{n \log n})$.

We now compute a 2-approximate minimal homology basis from $\mathcal{G}'$. Here we use so-called *earliest basis*. Specifically, in general, given a matrix $D$ with rank $r$, the set of column vectors $\{D[i_1], \cdots, D[i_r]\}$ is called an **earliest basis** for the vector space spanned by all columns in $D$ (or simply, for $D$), if the column indices $\{i_1, \ldots, i_r\}$ are the lexicographically smallest index set such that the corresponding columns of $D$ have full rank.

**Proposition 4.2 ( [3]).** *Let $D$ be an $m \times n$ matrix of rank $r$ with entries over $\mathbb{Z}_2$ where $n \leq m$, then there is an $O(mn^{\omega-1})$ time algorithm to compute the earliest basis of $D$.*

Let $\{i_1, \ldots, i_g\}$ be the indices of columns in the earliest basis of $M$. This can be done in $O(mg^{\omega-1}) = O(n\sqrt{n \log n} \cdot g^{\omega-1})$ time by the above proposition as $m = O(n\sqrt{n \log n})$. The cycles corresponding to these columns form a homology basis by the properties of annotations [3].

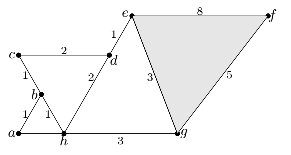
Finally, we note that the earliest basis of $M$ has the smallest (lexicographically) sequence of size sequence. Hence its total size is at most the size of the 2-approximate minimal homology basis $A_1, \ldots, A_g$ as specified in Lemma 4.1. Hence putting everything together, we conclude with the following theorem.

**Theorem 4.1.** *The algorithm above computes a 2-approximate minimal homology basis of the 1-dimensional homology group $\mathsf{H}_1(\mathcal{K})$ of a simplicial complex with non-negative weights in $O(n^\omega \sqrt{n \log n})$ expected time.*

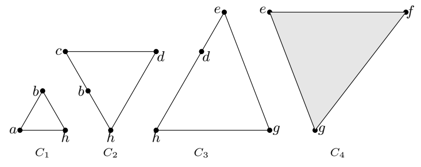
**Remark.** Since an approximate minimal homology basis still forms a basis for $\mathsf{H}_1(\mathcal{K})$, it means that computing it is at least as hard as computing the rank of $\mathsf{H}_1(\mathcal{K})$. Currently the best algorithm for the rank computation for general simplicial complex $\mathcal{K}$ is $O(n^\omega)$ (the matrix multiplication time). Hence the best we can expect for computing an approximate minimal homology basis is perhaps $O(n^\omega)$ (versus the $O(n^2 g + n^\omega)$ time complexity of the exact algorithm from Section 3.1). We remark that we can also develop an algorithm that computes a $(2k-1)$-approximate minimal homology basis in time $O(kn^{1+1/k} g \text{ polylog } n + n^\omega)$, where $k \geq 1$ is an integer – indeed, as the approximation factor reaches $\log n$, the time complexity becomes $O(n^\omega)$ (which is the best time known for rank computation). The framework of this algorithm follows closely from an approach by Kavitha et al. in [12], and we thus omit the details here.

**A counter-example.** Figure 1 gives an example which shows that, without Proposition 4.1, it is not guaranteed that a candidate set containing a $c$-approximate minimal cycle basis includes a 2-approximate minimal homology basis. Let the size of a 1-cycle in $\mathcal{K}$ shown in the figure to be the sum of all edges in the cycle. There is only one minimal cycle basis in this figure, namely $C_1, C_2, C_3$ and $C_4$, as shown in Figure 1b. The minimal homology basis of $\mathcal{K}$ should be $\{C_1, C_2, C_3\}$. However, consider the candidate set $\mathcal{G}$ which contains 4 cycles as shown in Figure 1c: $C_2, C_3, C_4$ and $C_4' = C_1 + C_2 + C_3$. It is easy to check that these 4 cycles in $\mathcal{G}$ form a 2-approximate minimal *cycle* basis. However, the smallest homology basis contained in $\mathcal{G}$, namely $C_2, C_3, C_4' (= C_1 + C_2 + C_3)$ is not a 2-approximate minimal homology basis.

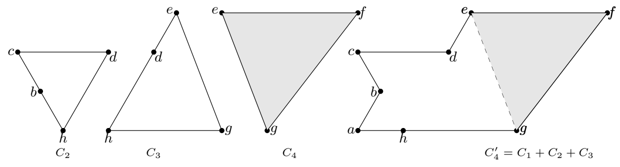
We can make this example into a counter-example for any constant factor approximation, by adding more $C_i''$s (triangles) to the sequence, each of which is larger than the previous one and is also filled in. In other words, the optimal homology basis remains $\{C_1, C_2, C_3\}$, while the smallest-size homology basis from the 2-approximate minimal cycle basis is $\{C_2, C_3, \sum_{i>1} C_i\}$.



(a) A simplicial complex $\mathcal{K}$



(b) A minimal cycle basis of $\mathcal{K}$



(c) A 2-approximate minimal cycle basis of $\mathcal{K}$

**Fig. 1.** An example where an approximate minimal cycle basis **does not** contain an approximate minimal homology basis.

## 5 Generalizing the size measure

The 1-skeleton $\mathcal{K}^{(1)}$ of the simplicial complex $\mathcal{K}$ is the set of vertices and edges in $\mathcal{K}$. If there are non-negative weights defined on edges in $\mathcal{K}^{(1)}$, it is natural to use the induced shortest path distance in $\mathcal{K}^{(1)}$ (viewed as a weighted graph) as a metric for vertices $V$ in $\mathcal{K}$. One can then measure the "size" of a cycle to be the sum of edge weights. Indeed, this is the distance and the size measure considered in Sections 3 and 4. In this section, we show that the algorithmic framework in Algorithm 1 can in fact be applied to a more general family of size measures. Specifically, first, we introduce what we call the *path-dominated distance* between vertices of $\mathcal{K}$ (which is not necessarily a metric). Based on such distance function, we then define a family of "size-functions" under which measure we can always compute a minimal homology basis using Algorithm 1. The shortest-path distance/size measure used in Section 3, and the geodesic ball-based measure proposed in [10] are both special cases of our more general concepts. We also present another natural path-dominated distance function induced by a (potentially complex) map $F : vert(\mathcal{K}) \to Z$ defined on the vertex set $vert(\mathcal{K})$ of $\mathcal{K}$ (where $Z$ is another metric space, say $\mathbb{R}^d$). As a result, we can use Algorithm 1 to compute the shortest 1-st homology basis of $\mathcal{K}$ induced by a map $F : vert(\mathcal{K}) \to Z$.

## 5.1 Path-dominated distance

Given a connected simplicial complex $\mathcal{K}$, suppose we are given a *distance function* $d : vert(\mathcal{K}) \times vert(\mathcal{K}) \to \mathbb{R}^+ \cup \{0\}$. We now introduce the following *path-dominated distance function*.

**Definition 5.1 (Path-dominated distance).** *A function $d : vert(\mathcal{K}) \times vert(\mathcal{K}) \to \mathbb{R}^+ \cup \{0\}$ is a **path-dominated distance function (w.r.t. $\mathcal{K}$))** if*

*(i) $d(x, y) \geq 0$ and $d(x, x) = 0$ for any $x, y \in vert(\mathcal{K})$;*
*(ii) given any two vertices $x, y \in vert(\mathcal{K})$, there exists a path $\pi^*$ connecting $x$ to $y$ in the 1-skeleton $\mathcal{K}^{(1)}$ such that $d(x, y) = \max_{u \in vert(\pi^*)} d(x, u)$.*

If edges in the 1-skeleton $\mathcal{K}^{(1)}$ have positive weights, then, it is easy to verify that the standard shortest path distance metric induced by $\mathcal{K}^{(1)}$ (viewed as a weighted graph) is path-dominated. However, note that a path-dominated distance may not be a metric. Indeed, we will shortly present a function-induced distance which is not symmetric.

We now define "shortest path" in $\mathcal{K}^{(1)}$ induced by a path-dominated distance function.

**Definition 5.2 (Path-dominated shortest path).** *Given any $x, y \in vert(\mathcal{K})$, a path $\pi^* = \langle u_0 = x, u_1, \ldots, u_k = y$ connecting $x$ to $y$ via edges in $\mathcal{K}$ is a **path-dominated shortest path in** $\mathcal{K}$ if for each $i \in [1, k]$, $d(x, u_i) = \max_{j \leq i} d(x, u_j)$.*

Note that this implies that any prefix of a path-dominated shortest path is also a path-dominated shortest path. The proof of the following statement is reasonably simple and can be found in Appendix B.

**Claim 5.1** *A path-dominated shortest path always exists for any two vertices $x, y \in vert(\mathcal{K})$.*

*Function-induced distance.* Very often, the domain $\mathcal{K}$ may come with additional data modeled by a function $F : vert(\mathcal{K}) \to Z$ defined on vertices of $\mathcal{K}$, where the co-domain $(Z, d_Z)$ is a metric space. For example, imagine that $\mathcal{K}$ represents the triangulation of a region on earth, and at each vertex, we have collected $d$ sensor measurements (e.g. temperature, wind speed, sun-light strength, etc), which can be modeled by a function $F : vert(\mathcal{K}) \to \mathbb{R}^d$. It is then natural to define a distance, as well as a size measure later, that depends on this function $F$. We introduce the following *function-induced distance* $d_F : vert(\mathcal{K}) \times vert(\mathcal{K}) \to \mathbb{R}$:

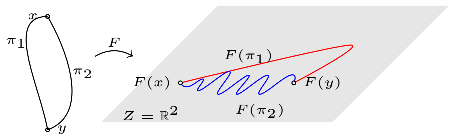


**Fig. 2.** The left is the original simplicial complex. There are two paths, $\pi_1$ and $\pi_2$, connecting vertices $x$ and $y$. The right figure is their image under the map $F$ with $Z = \mathbb{R}^2$ (i.e, $d_Z(\cdot, \cdot) = \| \cdot - \cdot \|$). The path $\pi_2$ is a path-dominated shortest path from $F(x)$ to $F(y)$.

**Definition 5.3.** *Given any function $F : vert(\mathcal{K}) \to Z$, we define the **$F$-induced distance** $d_F(x, y)$ as follows:*

$$d_F(x, y) = \min_{path\ \pi(x,y) \subseteq \mathcal{K}^{(1)}} \max_{u \in \pi(x,y)} d_Z(F(x), F(u)), \qquad (1)$$

*where the minimum ranges over all path $\pi(x, y)$ from $K^{(1)}$ connecting $x$ to $y$.*

Intuitively, given a path $\pi$ from $x$ to $y$, $\max_{u \in \pi} d_Z(F(x), F(u))$ measures the maximum distance in terms of the function value $F$ between the starting point $x$ to any point in the path $\pi$, i.e, the maximum function distortion from $x$ to $\pi$. $d_F(x, y)$ is the smallest function distortion (w.r.t. $x$) needed to connect from $x$ to $y$. For example, in Figure 2, the path $\pi_2$ is a path-dominated shortest path from $x$ to $y$, as its image $F(\pi_2)$ has a smaller maximum distance (in terms of $d_Z = \| \cdot \|$) than the image of $F(\pi_1)$. By the definition of function-induced distance, we have:

**Claim 5.2** *Given $F : vert(\mathcal{K}) \to Z$, the $F$-induced distance $d_F$ is path-dominated.*

## 5.2 Size-measure for 1-cycles

Previously, the most popular way to measure the "size" of a 1-cycle is the sum of weights of edges in the cycle. Another natural measure formulated by Chen and Freedman [10] uses the minimum radius of any metric ball (centered at some vertex in $\mathcal{K}$) containing a cycle as its size. Intuitively, given a homology class, a smallest cycle of this class under this radius-measure corresponds to a cycle which is most "localized" (contained within a smallest possible metric ball). Using the shortest-path metric induced by weights on edges in $\mathcal{K}$, Chen and Freedman showed that a minimal homology basis under this radius-measure can be computed in polynomial time for any fixed-dimensional homology group. In what follows, we introduce a family size measures, which we refer to as *tight-size functions*, which generalize the radius-measure of Chen and Freedman as well as the general sum-of-weights measure. We then show that the algorithm from Section 3 can be used to compute a minimal homology basis for $\mathsf{H}_1(\mathcal{K})$ w.r.t. such tight-size functions.

We use the concept of *edge-short* cycles introduced in e.g. [13], whose origin traces back to [7].

**Definition 5.4.** *A 1-cycle $C$ in a complex $\mathcal{K}$ is called* ***edge-short****, if $\mathcal{K}$ contains a vertex $w$, an edge $e = (u, v)$, a shortest path from $w$ to $u$ and a shortest path from $w$ to $v$ such that $C$ is the edge disjoint union of $e$ and the two paths.*

In our case, instead of using the shortest path metric induced by weights on the 1-skeleton $\mathcal{K}^{(1)}$ of $\mathcal{K}$, we use any path-dominated distance function $d$, and the "shortest paths" in the above definition will be replace by path-dominated shortest paths in $\mathcal{K}$ w.r.t. $d$. To emphasize the dependency on the path-dominated distance function $d$, we say that a cycle $C$ is *edge-short w.r.t. $d$* if conditions in Definition 5.4 holds w.r.t. path-dominated shortest paths w.r.t. $d$.

**Definition 5.5 (Tight-size function).** *Suppose $vert(\mathcal{K})$ is equipped with a path-dominated distance function $d$. Let $\mathsf{Z}_1(\mathcal{K})$ represent the 1-dimensional cycle group of $\mathcal{K}$. A function $\mu : \mathsf{Z}_1(\mathcal{K}) \to \mathbb{R}$ is a* ***tight-size function (w.r.t. $d$)*** *if under this function, there exists a minimal homology basis for $\mathsf{H}_1(\mathcal{K})$ in which all cycles are edge-short w.r.t. the path-dominated distance $d$.*

*We may omit the reference to the path-dominated distance $d$ when its choice is fixed or clear.*

We now prove that if a function is a tight-size function, the Algorithm 1 can be used to compute a minimal homology basis. First, observe the following, which is implied by Theorem 3.1.

**Claim 5.3** *If the candidate set $\mathcal{G}$ contains a minimal homology basis, then the framework Algorithm 1 will compute a minimal homology basis from the candidate set.*

What remains is to show how to compute a candidate set containing a minimal homology basis. For simplicity, from now we fix a path-dominated distance function $d$, and simply use *shortest paths* to refer to the *path-dominated shortest paths w.r.t. $d$*. We assume that the shortest paths are unique – In Appendix C, we describe how to guarantee this uniqueness condition (by assigning certain order to the shortest paths), and show that the shortest path tree $T_p$ encoding all unique path-dominated shortest paths to any root $p \in vert(\mathcal{K})$ can be computed in $O(n \log n)$ time (with $n = |\mathcal{K}^{(1)}|$) by the standard approach.

We now construct a candidate set $\mathcal{G}$ in the same manner as in Section 3. First for every vertex $p$, we build a candidate set $\mathcal{G}_p$. Let $\Pi_p(u, v)$ denote the unique tree path between two vertices $u$ and $v$. For every nontree edge $e = (u, v)$, $C(p, e) = e \circ \Pi_p(u, v)$ is a cycle. We add all such $C(p, e)$ into $\mathcal{G}_p$, i.e. $\mathcal{G}_p = \cup_{e \in E \setminus edge(T_p)} C(p, e)$. Then take the union of all such candidate sets, $\mathcal{G}$ can be constructed as $\mathcal{G} = \cup_{p \in vert(\mathcal{K})} \mathcal{G}_p$.

**Lemma 5.1.** *The candidate set $\mathcal{G}$ contains a minimal homology basis when the size of a cycle is measured by a tight-size function w.r.t. some path-dominated distance function $d$.*

*Proof.* By results from Appendix C, we can assume that there is only a unique path-dominated shortest path between any two vertices $u, v \in vert(\mathcal{K})$, which we denote as $SP(u, v)$. Now take any edge-short cycle $C$. As it is edge-short, we can find a vertex $w$ and an edge $e = (u, v)$ such that the cycle $C$ is the disjoint union of $SP(w, u)$, $SP(w, v)$ and $e$. On the other hand, the unique shortest paths $SP(w, u)$ and $SP(w, v)$ are in the shortest path tree $T_w$. This means that $e \notin T_w$. Hence the cycle $C$ is a candidate cycle $C(w, e)$ from the set $\mathcal{G}_w$. It then follows that the collection $\mathcal{G}$ contains all edge-short cycles. The lemma then follows from the definition of tight-size functions and Lemma 5.3. □

Now that we have a candidate set that contains a minimal homology basis, we can apply the divide and conquer algorithm (Algorithm 1) from Section 3, and by Claim 5.3, this will output a minimal homology basis. We conclude with the following main result.

**Theorem 5.1.** *Suppose sizes of 1-cycles are measured by a tight-size function w.r.t. a path-dominated distance function $d$. Then, we can compute a minimal homology basis for $\mathsf{H}_1(\mathcal{K})$ in $O(n^\omega + n^2 g)$ time, where $n$ is the size of 2-skeleton of $\mathcal{K}$ and $g$ is $rank(\mathsf{H}_1(\mathcal{K}))$.*

## 5.3 Examples of tight-size functions

*Sum-of-weights size function.* As mentioned earlier in Section 5.1, given a weight function $w : edge(\mathcal{K}) \to \mathbb{R}^+$, the shortest path distance $d_{\mathcal{K}}$ induced by the 1-skeleton $\mathcal{K}^{(1)}$ (viewed as a weighted graph) is a path-dominated function. Now given weights $w : edge(\mathcal{K}) \to \mathbb{R}^+$, the size measure $\mu_w : \mathsf{Z}_1(\mathcal{K}) \to \mathbb{R}^+$ assigning $\mu_w(C) = \sum_{e \in C} w(e)$ is a tight-size function w.r.t. the shortest path distance function $d_{\mathcal{K}}$. Hence we can obtain the main result of Section 3 by applying Theorem 5.1 to the tight-size function $\mu_w$.

*Radius-size function.* Alternatively, we now consider the radius-based size function used e.g. in [10, 14, 15]. More specifically, suppose we are given a simplicial complex $\mathcal{K}$, and a path-dominated distance function $d$ (which may not be a metric) on $vert(\mathcal{K})$. Define the ball $B_p^r$ centered at $p$ of radius $r$ to be $B_p^r = \{\sigma \in \mathcal{K} : \forall x \in vert(\sigma), d(p,x) \leq r\}$. We can then define *radius-size function* $\mu_R : \mathsf{Z}_1(\mathcal{K}) \to \mathbb{R}^+$ such that $\mu_R(C)$ of a 1-cycle $C$ is the smallest $r$ such that $C \subseteq B_p^r$ for some $p \in vert(\mathcal{K})$.

**Proposition 5.1** *$\mu_R$ is a tight-size function w.r.t. any path-dominated distance function $d$.*

*Proof.* We need to prove that there exists a minimal homology basis where each cycle inside is edge-short. Assume this is not the case. Then given any minimal homology basis $\mathcal{B}$, there exists a cycle $C$ which is not edge-short. Suppose cycles $\mathcal{B} = \{C_1, \ldots, C_g\}$ are sorted in nondecreasing order of their radius-size, and $C_i$ is the first cycle in $\mathcal{B}$ that is not edge-short. Let $B_p^r$ be the smallest ball containing $C_i$ with $p \in vert(\mathcal{K})$; that is, $\mu_R(C_i) = r$. Let $T_p$ denote the shortest path tree rooted at $p$, and $Q$ denote the set of edges in $C_i$ which are not in $T_p$. Note that $Q$ cannot be empty; otherwise, $C_i$ cannot be a cycle as all edges in it are tree edges. For every edge $e = (u,v)$ in $Q$, we can construct a cycle $C(p,e)$ as $SP(p,u) + SP(p,v) + e$, where $SP(x,y)$ denote the tree path in $T_p$ from $x$ to $y$. It is easy to see that for each such cycle $C(p,e)$ with $e \in Q$, its radius-size $\mu_R(C(p,e)) \leq r$ as it is completely contained within $B_p^r$. Note that $C_i$ can be represented as the sum of all such $C(p,e)$, i.e. $C_i = \sum_{e \in Q} C(p,e)$. This is because that $C_i = \sum_{e \in C_i} C(p,e)$. However, for an edge $e \in C_i \cap T_p$, $C(p,e)$ is the empty set. Hence only edges from $C_i \setminus T_p (= Q)$ contribute to this sum.

Now consider the set of cycles $\mathcal{Q} = \{C(p,e) \mid e \in Q\}$. As $C_i$ is in a minimal homology basis $\mathcal{B}$, its homology class $[C_i]$ is independent of those generated by cycles in $\mathcal{B} \setminus \{C_i\}$. Hence there exists at least a cycle $C' \in \mathcal{Q}$ such that $[C']$ is independent of the homology class of all cycles in $\mathcal{B} \setminus \{C_i\}$. Now let $\mathcal{B}' = \mathcal{B} \cup \{C'\} \setminus \{C_i\}$ which is also a homology basis. Recall that any cycle in $\mathcal{Q}$ has radius-size at most $r$. We have two cases: (i) If $\mu_R(C') < r (= \mu_R(C_i)$, then $\mathcal{B}'$ has a smaller size sequence than $\mathcal{B}$, and thus $\mathcal{B}$ cannot be a minimal homology basis. Thus we have a contradiction, meaning that all cycles in $\mathcal{B}$ must be edge-short. (ii) If $\mu_R(C') = r$, then $\mathcal{B}'$ is also a minimal homology basis. If $B'$ contains only edge-short cycles, then we are done. If not, then we identify the next cycle that is not edge-short $C_j$, and it is necessary that $j > i$. We then repeat the above argument with $C_j$. In the end, either we find a contradiction, meaning that the edge-short cycle cannot exist in the basis we are inspecting, or we manage to replace all non-edge-short cycles to be edge-short ones of equal size, and maintain a homology basis. In the latter case, we construct a minimal homology basis with only edge-short cycles. In either case, the proposition follows. ☐

It then follows from the above proposition that Algorithm 1 computes a minimal homology basis under the radius-size function w.r.t. any path-dominated distances in time $O(n^\omega + n^2 g)$. In particular, combining with the two path-dominated distance functions examples we have:

**Example 1:** $d = d_{\mathcal{K}}$, the shortest path distance induced by the weighted graph $\mathcal{K}^{(1)}$. Under this path-dominated distance, the minimal homology basis problem under the radius-size function w.r.t. $d_{\mathcal{K}}$ is exactly the 1-dimensional case of the problem studied in [10]. An $O(n^4 g)$ time algorithm was presented to solve this problem in any dimension in [10]. However, by Theorem 5.1, we can compute a minimal homology basis of in $O(n^\omega + n^2 g)$ time, which is a significant improvement when focusing on $\mathsf{H}_1$ group.

**Example 2:** Given a function $F : vert(\mathcal{K}) \to Z$ defined on $\mathcal{K}$, recall that the $F$-induced distance $d_F$ as introduced in Section 5.1 is a path-dominated function. Now set $d = d_F$. Intuitively, the radius-size function $\mu_R(C)$ w.r.t. $d_F$ measures the radius of the smallest metric ball in the co-domain $Z$ that contains the image $F(C)$ of the cycle $C$ under map $F$. That is, $\mu_R(C)$ measures the "size" of $C$ w.r.t. the variation in the function $F$. Hence we also refer to the radius-size function w.r.t. $d_F$ as the *$F$-induced radius-size function.* We believe that the $F$-induced distance function and $F$-induced radius-size function are useful objects of independent interests. The minimal homology basis of $\mathcal{K}$ under such a $F$-induced radius-size function can also be computed in $O(n^\omega + n^2 g)$ time.

## 6 Conclusions

In this paper we have given improved algorithms for computing a minimal homology basis for 1-dimensional homology group of a simplicial complex. What about higher dimensional homology? For high dimensions, it is known from [16] that computing a minimum homology basis under volume measure is NP-hard. But it follows from [10] that one can extend the radius-size measure (See Section 5.3) to high dimensions under which an algorithm to compute a minimum homology basis in polynomial time exists. It runs in time $O(gn^4)$ where $g$ is the rank of $d$-dimensional homology group $\mathsf{H}_d$. We can improve this algorithm, using persistence algorithm [17] as well as annotations for $d$-simplices [3], so that the time complexity improves to $O(n^{\omega+1})$ which is better when $g = \Theta(n)$. The details are presented in the Appendix A.

## Acknowledgements

This works is partially supported by National Science Foundation (NSF) under grants CCF-1526513, 1740761 and 1733798.

## References

1. Borradaile, G., Chambers, E.W., Fox, K., Nayyeri, A.: Minimum cycle and homology bases of surface-embedded graphs. Journal of Computational Geometry **8**(2) (2017) 58–79
2. Erickson, J., Whittlesey, K.: 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) 1038–1046
3. Busaryev, O., Cabello, S., Chen, C., Dey, T.K., Wang, Y.: Annotating simplices with a homology basis and its applications. In: Scandinavian Workshop on Algorithm Theory, Springer (2012) 189–200
4. Le Gall, F.: Powers of tensors and fast matrix multiplication. In: Proceedings of the 39th international symposium on symbolic and algebraic computation, ACM (2014) 296–303
5. Kavitha, T., Mehlhorn, K., Michail, D., Paluch, K.: A faster algorithm for minimum cycle basis of graphs. In: International Colloquium on Automata, Languages, and Programming, Springer (2004) 846–857
6. de Pina, J.C.: Applications of shortest path methods. Ph.D. thesis, University of Amsterdam (1995)
7. Horton, J.D.: A polynomial-time algorithm to find the shortest cycle basis of a graph. SIAM Journal on Computing **16**(2) (1987) 358–366
8. Mehlhorn, K., Michail, D.: Minimum cycle bases: Faster and simpler. ACM Transactions on Algorithms (TALG) **6**(1) (2009) 8
9. Dey, T.K., Sun, J., Wang, Y.: Approximating loops in a shortest homology basis from point data. In: Proceedings of the twenty-sixth annual symposium on Computational geometry, ACM (2010) 166–175
10. Chen, C., Freedman, D.: Measuring and computing natural generators for homology groups. Computational Geometry **43**(2) (2010) 169–181
11. Hatcher, A.: Algebraic Topology. Cambridge University Press (2002)
12. Kavitha, T., Mehlhorn, K., Michail, D.: New approximation algorithms for minimum cycle bases of graphs. STACS 2007 (2007) 512–523
13. Gleiss, P.M.: Short cycles: minimum cycle bases of graphs from chemistry and biochemistry. PhD thesis, Universität Wien, Austria (2001)
14. Guskov, I., Wood, Z.J.: Topological noise removal. 2001 Graphics Interface Proceedings: Ottawa, Canada (2001) 19
15. Wood, Z., Hoppe, H., Desbrun, M., Schröder, P.: Removing excess topology from isosurfaces. ACM Transactions on Graphics (TOG) **23**(2) (2004) 190–208
16. Chen, C., Freedman, D.: Hardness results for homology localization. In: Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics (2010) 1594–1604
17. Edelsbrunner, H., Harer, J.: Persistent homology-a survey. Contemporary mathematics **453** (2008) 257–282
18. Chen, C., Freedman, D.: Quantifying homology classes. In: LIPIcs-Leibniz International Proceedings in Informatics. Volume 1., Schloss Dagstuhl-Leibniz-Zentrum für Informatik (2008)
19. Hartvigsen, D., Mardon, R.: The all-pairs min cut problem and the minimum cycle basis problem on planar graphs. SIAM J. Discret. Math. **7**(3) (1994) 403–418
20. Wulff-Nilsen, C.: Minimum cycle basis and all-pairs min cut of a planar graph in subquadratic time. arXiv preprint arXiv:0912.1208 (2009)

## A Computing a minimal homology basis for $\mathsf{H}_d(\mathcal{K})$

Let $\mathcal{K}$ be a simplicial complex with $n$ simplices and let $g$ be the $d$-dimensional Betti number, i.e. $g = rank(\mathsf{H}_d(\mathcal{K}))$. The discrete geodesic distance $d_p : vert(\mathcal{K}) \to \mathbb{R}$ from a vertex $p$ is given by $q \mapsto dist(p, q)$ where $dist(p, q)$ is the length of the shortest path from $p$ to $q$. Extending this definition to general simplices, we have $\forall \sigma \in \mathcal{K}, d_p(\sigma) = max_{q \in vert(\sigma)} d_p(q)$. Then the geodesic ball $B_p^r$ of radius $r$ centered at $p$ is defined as $B_p^r = \{\sigma \in \mathcal{K} : d_p(\sigma) \leq r\}$. Clearly, $B_p^r \subseteq \mathcal{K}$, and it is a subcomplex of $\mathcal{K}$. This is because for all faces $\sigma'$ of $\sigma$, $d_p(\sigma') \leq d_p(\sigma)$, which implies that all faces of a simplex in $B_p^r$ are also in $B_p^r$.

The size of a cycle $C$ is defined as $\mu(C) = min\{r : \exists p \in vert(K), s.t.\ C \subset B_p^r\}$ [18]. In words, it is the radius of the smallest ball centered at some vertex $p$ of $C$, which contains $C$. The definition of a minimal homology basis becomes:

**Definition A.1.** *Given a simplicial complex $\mathcal{K}$, a set of cycles $\{C_1, C_2, \cdots, C_g\}$ with $g = rank(\mathsf{H}_d(\mathcal{K}))$ is a $d$-dimensional minimal homology basis if (1) the homology classes $\{[C_1], [C_2], \cdots, [C_g]\}$ constitute a homology basis and (2) the sizes $\{\mu(C_1), \mu(C_2), . . ., \mu(C_g)\}$ are lexicographically smallest among all such bases.*

### A.1 Algorithm

In this section, we describe an algorithm to compute a minimal $d$-dimensional homology basis where $d \geq 1$. There are two steps in the algorithm: First computing a candidate set which contains a minimal homology basis and then computing a minimal homology basis from the candidate set. All computations are over $\mathbb{Z}_2$.

**Computing a candidate set.** We now describe how to compute a candidate set of cycles including a minimal homology basis. We apply the persistent homology algorithm to generate the candidate set $\mathcal{C}(p)$ for a vertex $p$ with the following filtration: Simplices are sequenced in non-decreasing order of geodesic distances $d_p(\cdot)$ while placing a simplex before all its cofaces that have the same geodesic distance. We focus on the essential homology classes computed by persistent algorithm. There are $g$ of them. For each essential homology class $h$, we denote its birth time as $r_p(h)$. For any vertex $p$, the number of candidate cycles in $\mathcal{C}(p)$ is $g$. Thus, the number of cycles of the candidate set $\mathcal{C}$ is $O(gn)$.

**Claim A.1** *The candidate set $\mathcal{C}$ includes a minimal homology basis.*

*Proof.* Suppose not. Let $\mathcal{B}$ be any minimal homology basis and the elements in $\mathcal{B}$ are sorted in nondecreasing order of their sizes. Let class $C_i$ be the first member in $\mathcal{B}$ which is not in the candidate set and let $p$ be the vertex such that $C_i \subset B_p^{\mu(C_i)}$ where $\mu(C_i)$ is the size of the cycle $C_i$. First we claim that there exists a $d$-simplex $\sigma$ such that $d_p(\sigma) = \mu(C_i)$ and $\sigma$ is a creator of $C_i$. If not, there is another cycle $C'$ such that $[C'] = [C_i]$ and $\mu(C') < \mu(C_i)$. Note that the cycles generated by creators in $B_p^{\mu(C_i)}$ form a homology basis of $B_p^{\mu(C_i)}$. We prove that the geodesic ball $B_p^{\mu(C_i)}$ must include a cycle $C^* \in \mathcal{C}$ such that the following two conditions hold: (1) $\mu(C^*) \leq \mu(C_i)$. (2) $\mathcal{B} \setminus \{C_i\} \cup \{C^*\}$ is a homology basis.

Condition (1) holds because $\mu(C) \leq \mu(C_i)$ for every cycle $C$ in $B_p^{\mu(C_i)}$.

For (2), observe that there exists a homology class $[C^*]$ generated by one creator that is independent of homology classes generated by $\mathcal{B} \setminus \{C_i\}$. If no such cycle exists, any homology class generated by one creator of $B_p^{\mu(C_i)}$ can be written as a linear combination of homology classes generated by $\mathcal{B} \setminus \{C_i\}$. The homology classes generated by creators form a homology basis of $B_p^{\mu(C_i)}$ and $C_i \in B_p^{\mu(C_i)}$. It means that $[C_i]$ is not independent of $\mathcal{B} \setminus \{C_i\}$, contradicting the assumption that $\mathcal{B}$ is a homology basis. Therefore, $\mathcal{B} \setminus \{C_i\} \cup \{C^*\}$ is a homology basis.

Combining condition (1) with (2), the homology basis $\mathcal{B}' = \mathcal{B} \setminus \{C_i\} \cup \{C^*\}$ is a minimal homology basis. What is more, if we sorted the cycles in $\mathcal{B}'$ in nondecreasing order of sizes, then the first $i + 1$ cycles in $\mathcal{B}'$ are in the candidate set $\mathcal{C}$. This is because the cycle $C^*$ is generated by a creator of $B_p^{\mu(C_i)}$, which means that $C^* \in \mathcal{C}$. Therefore, we find a minimal homology basis all of whose cycles are in the candidate set. $\square$

**Computing a minimal homology basis.** In this section we discuss an algorithm to find a minimal homology basis from the candidate set. We use annotation, denoted by $a(\cdot)$, to represent and distinguish each cycle. Recall that annotation of a cycle is a $g$-bit vector were $g = rank(\mathsf{H}_d)$, and that two cycles are homologous if and only if their annotations are equal. We first compute the annotations for all $d$-simplices in $\mathcal{K}$ [3] and give them a fixed order $\sigma_1, \sigma_2, \cdots, \sigma_{n_d}$ where $n_d$ is the number of $d$-simplices in $\mathcal{K}$. Suppose we sort the cycles in the candidate set in nondecreasing order of their sizes as $C_1, C_2, \cdots, C_{gn_0}$ where $n_0$ is the number of vertices in $\mathcal{K}$. Then, every $d$-cycle $C_i$

in $\mathcal{K}$ can be denoted as $C_i = \sum_{j=1}^{n_d} \gamma_{ij}\sigma_j$ where $\gamma_{ij} \in \{0,1\}$ and $1 \le i \le gn_0$. Thus, we have $a(C_i) = \sum_{j=1}^{n_d} \gamma_{ij} a(\sigma_j)$, $1 \le i \le gn_0$. We compute the annotations $a(C_1), a(C_2), \cdots, a(C_{gn_0})$ for all cycles $C_1, C_2, \cdots, C_{gn_0}$ in the candidate set simultaneously.

Let $X = (a(C_1)^T, a(C_2)^T, \cdots, a(C_{gn_0})^T)^T$ and $Y = (a(\sigma_1)^T, a(\sigma_2)^T, \cdots, a(\sigma_{n_d})^T)^T$. The goal is to compute $X$ that satisfies the following equation: $X = \Gamma Y$ where $\Gamma = (\gamma_{ij})_{gn_0 \times n_d}$. The computation of the matrix $X$ takes time $O(n^\omega g)$ using the fast matrix multiplication algorithm where $\Gamma$ is a $gn_0 \times n_d$ matrix and $Y$ is an $n_d \times g$ matrix.

Let $X'$ be the transposed matrix of $X$. The problem of computing a minimal homology basis from the candidate set $\mathcal{C}$ is equivalent to computing the earliest basis of the matrix $X'$ [3]. According to Proposition 4.2, computing the earliest basis of $X'$ costs us $O(ng^\omega)$ time. Combining the time $O(n^{\omega+1})$ in building the candidate set $\mathcal{C}$ and the time $O(n^\omega g)$ in computing $X$, we conclude that the total running time is $O(n^{\omega+1})$.

**Theorem A.1.** *Given a simplicial complex $\mathcal{K}$ with $n$ simplices, there is an algorithm to compute a minimal homology basis as defined in Definition A.1 in any dimension in time $O(n^{\omega+1})$.*

## B Proof of Claim 5.1

We prove this claim by induction. First, fix any source node $x \in vert(\mathcal{K})$. We sort all other vertices in $vert(\mathcal{K})$ in non-decreasing order of $d(x,y)$; that is, $d(x,y_1) \le d(x,y_2) \le \ldots, \le d(x,y_s)$ with $s = |vert(\mathcal{K})| - 1$. We carry out an induction based on this order. For the base case, any path in (ii) of Definition 5.1 is necessarily a path-dominated shortest path from $x$ to $y_1$: Indeed, if there is any other vertex $y$ (other than $x$ and $y_1$) in such a path, it is necessary that $d(x,y) = d(x,y_1)$ as $d(x,y_1)$ has the smallest distance to $x$.

Now suppose there exists a path-dominated shortest path from $x$ to $y_i$ for $1 \le i \le s$. Consider $y_{i+1}$ and assume that there is no path-dominated shortest path from $x$ to $y_{i+1}$. By Definition 5.1, there exists a path $\Pi = (u_0 = x, u_1, \ldots, u_k = y_{i+1})$ such that for every vertex $u_i \in \Pi$, $d(x,u_i) \le d(x,y_{i+1})$. As this path violates the conditions in Definition 5.2, there must exist a pair of vertices $u_j, u_l \in \Pi$ with $j < l$ such that $d(x,y_{i+1}) \ge d(x,u_j) > d(x,u_l)$. Let $l$ be the maximal value with which such a pair $(j,l)$ exists. It follows that we have $d(x,u_l) \le d(x,u_{l+1}) \le \ldots \le d(x,u_k)$. By inductive hypothesis, we know that there is a path-dominated shortest path $\Pi^*$ from $x$ to $u_l$ since $d(x,u_l) < d(x,y_{i+1})$. Hence, the path $\Pi^*$ concatenated with the sub-path of $\Pi$ from $u_l$ to $u_k = y_{i+1}$ gives a path-dominated shortest path from $x$ to $y_{i+1}$. The claim thus follows from induction.

## C Ensuring uniqueness of shortest paths

In section 5, we require that the path-dominated shortest path (in this section, we use shortest path for short) in $\mathcal{K}$ between any two vertices is unique. Now we show how to avoid this restriction using an idea from [19].

**Lemma C.1.** *Let $\mathcal{K}$ be a simplicial complex with a path-dominated distance $d(\cdot,\cdot)$. For every pair of nodes, there exists a unique shortest path $\pi$ from $u$ to $v$ that satisfies exactly one of the following two conditions w.r.t. any other path $\pi'$ from $u$ to $v$:*
*(1) $len(\pi) < len(\pi')$*
*(2) $len(\pi) = len(\pi'), min(vert(\pi) \setminus vert(\pi')) < min(vert(\pi') \setminus vert(\pi))$*
*Here $len(\pi)$ denotes the number of edges in a path $\pi$ and $min(U)$ denotes the minimum index of the vertices in a subset $U$ of $vert(\mathcal{K})$. We say $\pi < \pi'$ if the above two conditions hold.*

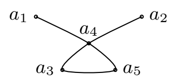


**Fig. 3.** Path $\pi_1 = a_1a_4a_5a_3a_4a_2$ and $\pi_2 = a_1a_4a_3a_5a_2$ are two path-dominated shortest paths from $a_1$ to $a_2$. Consider a new path $\pi = a_1a_4a_2$ which is a path-dominated shortest path from $a_1$ and $a_2$. However $len(\pi) = 2 < 5 = len(\pi_1) = len(\pi_2)$.

The proof follows from [19, Proposition 4.1].

We now describe the algorithm to compute a shortest path tree $T_p$ w.r.t. a path-dominated distance $d(\cdot,\cdot)$ rooted at $p$ under the uniqueness condition. Let $\pi_p(q)$ be the tree path from $p$ to $q$ in the current partial tree. Initially we set a priority queue $Q$ the vertex set $vert(\mathcal{K})$. Every time we delete a vertex $q$ in the queue $Q$ with the least distance $d(p,q)$, least value $len_p(q)$ and least index. We iterate for all neighbors $w$ of $q$: If $\pi_p(q) \circ e$, $e = (q,w)$, is a shortest path from $p$ to $w$, and is smaller than $\pi_p(w)$ as in Lemma C.1 we will update the tree path $\pi_p(w)$ in $T_p$ as $\pi_p(q) \circ e$. Note that those vertices not in $Q$ will not be updated. Hence there are $O(n)$ iterations. What remains is to compute the minimum index of $vert(\pi) \setminus vert(\pi')$ given two tree paths $\pi$ and $\pi'$ from $p$ to any vertex $v$. This can be achieved in time $O(\log n)$ adapting the algorithm from [20] for path-dominated shortest path.

Thus we conclude the above analysis with the following theorem.

**Theorem C.1.** *The shortest path tree in a simplicial complex $\mathcal{K}$ can be computed in $O(n \log n)$ time.*