



Contents lists available at ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo


Approximation algorithms for Max Morse Matching[☆]


 Abhishek Rathod^{*}, Talha Bin Masood, Vijay Natarajan

Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India

ARTICLE INFO

Article history:

Received 29 April 2016

Accepted 18 October 2016

Available online 24 October 2016

Keywords:

Discrete Morse theory

Computational topology

Approximation algorithms

Homology computation

ABSTRACT

In this paper, we prove that the Max Morse Matching Problem is approximable, thus resolving an open problem posed by Joswig and Pfetsch [1]. For D -dimensional simplicial complexes, we obtain a $(D+1)/(D^2+D+1)$ -factor approximation ratio using a simple edge reorientation algorithm that removes cycles. For $D \geq 5$, we describe a $2^D/D$ -factor approximation algorithm for simplicial manifolds by processing the simplices in increasing order of dimension. This algorithm leads to $1/2$ -factor approximation for 3-manifolds and $4/9$ -factor approximation for 4-manifolds. This algorithm may also be applied to non-manifolds resulting in a $1/(D+1)$ -factor approximation ratio. One application of these algorithms is towards efficient homology computation of simplicial complexes. Experiments using a prototype implementation on several datasets indicate that the algorithm computes near optimal results.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Discrete Morse theory is a combinatorial analogue of Morse theory that is applicable to cell complexes [2]. It has become a popular tool in computational topology and visualization communities [3,4] and is actively studied in algebraic, geometric, and topological combinatorics [5,6].

The idea of using discrete Morse theory to speedup homology [7], persistent homology [8] and multidimensional persistence computations [9] hinges on the fact that discrete Morse theory helps reduce the problem of computing homology groups on an input simplicial complex to computing homology groups on a collapsed cell complex. Ideally, if one were to compute a discrete gradient vector field with minimum number of critical simplices (unmatched vertices in the Hasse graph) or maximum number of regular simplices (matched Hasse graph vertices), then the time required for computing homology over the collapsed cell complex would be very small. However, finding a vector field with maximum number of gradient pairs is an NP-hard problem as observed by Lewiner [10] and Joswig et al. [1] by showing a reduction from the collapsibility problem introduced by Egecioglu and Gonzalez in [11]. We study efficient approximations to the maximum number of gradient pairs in a discrete gradient vector field.

Computing the homology groups has several applications, particularly in material sciences, imaging, pattern classification and computer assisted proofs in dynamics [12]. More recently, homology and persistent homology have been appraised to be a more widely applicable computational invariant of topological spaces, arising from practical data sets of interest [13]. An approximate Morse matching designed using the algorithms described in this paper may be used to compute homology efficiently. One of the primary motivations for this work was that a previous study [7] involving discrete Morse theory

[☆] The work is partially supported by the Department of Science and Technology, India under grant SR/S3/EECE/0086/2012.

^{*} Corresponding author.

 E-mail addresses: abhishek@jcrathod.in (A. Rathod), tbmasood@csa.iisc.ernet.in (T. Bin Masood), vijayn@csa.iisc.ernet.in (V. Natarajan).

in homology computation reported noteworthy speedup over existing methods. Their method used a modification of the coreduction heuristic [14] to construct discrete Morse functions. We start with a twin goal in mind – first to introduce rigor into the study by developing algorithms with approximation bounds and then to have a practical implementation that achieves nearly optimal solutions.

1.1. Max Morse Matching Problem

The Max Morse Matching Problem (MMMP) can be described as follows: Consider the Hasse graph $\mathcal{H}_{\mathcal{K}}$ of a simplicial complex \mathcal{K} whose edges are all directed from a simplex to its lower dimensional facets. Associate a matching induced orientation to $\mathcal{H}_{\mathcal{K}}$ such that the resulting oriented graph $\overline{\mathcal{H}_{\mathcal{K}}}$ is acyclic. The goal is to maximize the cardinality of matched (regular) nodes. Equivalently, the goal is to maximize the number of gradient pairs. The approximate version of Max Morse Matching Problem seeks an algorithm that computes a Morse matching whose cardinality is within a factor α of the optimal solution for every instance of the problem.

1.2. Prior work

Joswig et al. [1] established the NP-completeness of Morse Matching Problem. They also posed the approximability of Max Morse Matching as an open problem pg. 6 Sec. 4 [1]. Several followup efforts seek optimality of Morse matchings either by restricting the problem to 2-manifolds or by applying heuristics [1,7,15–19]. Recently, Burton et al. [20] developed an FPT algorithm for designing optimal Morse functions.

1.3. Summary of results

We describe a $(D+1)/(D^2+D+1)$ -factor approximation algorithm for Max Morse Matching Problem on D -dimensional simplicial complexes. This algorithm uses maximum-cardinality bipartite matching on the Hasse graph $\mathcal{H}_{\mathcal{K}}$ to orient it. We then use a BFS-like traversal of the oriented Hasse graph $\overline{\mathcal{H}_{\mathcal{K}}}$ to classify matching edges as either forward edges if they do not introduce cycles or backward edges if they do. We use a counting argument to prove an approximation bound that holds for manifold as well as non-manifold complexes.

For simplicial D -manifolds, we propose two approximation algorithms. The first approximation algorithm provides a ratio of $2/(D+1)$, for $D \geq 3$. The ratio is improved to $2/D$, for $D \geq 5$, via a refinement that specifies the order in which the graph is processed. It leads to $1/2$ -factor and $4/9$ -factor approximations for 3-manifolds and 4-manifolds, respectively. Both algorithms process simplices of lowest dimension first followed by higher dimensions in increasing order. Every d -dimensional simplex is first given the opportunity to match to a $(d-1)$ -dimensional simplex. If unsuccessful, it is then given the option of matching to a $(d+1)$ -dimensional simplex. Furthermore, both algorithms employ optimal algorithms for designing gradient fields for 0-dimensional and D -dimensional simplices (in the case of manifolds). The refinement processes subgraphs with small vertex degree with higher priority and hence achieves a better approximation ratio.

We provide evidence of practical utility of our algorithms through an extensive series of computational experiments.

2. Background

2.1. Discrete Morse theory

Our focus in this paper is limited to simplicial complexes and hence we restrict the discussion of Forman's Morse theory below to simplicial complexes. Please refer to [21] for a compelling expository introduction.

Definition 1. A simplicial complex \mathcal{K} is a finite collection of simplices that satisfies the following conditions:

1. A face of a simplex in \mathcal{K} also belongs to \mathcal{K} .
2. The intersection of two simplices $\sigma_1, \sigma_2 \in \mathcal{K}$ is either empty or a face of both σ_1 and σ_2 .

Let \mathcal{K} be a simplicial complex, and let σ^d, τ^{d-1} be simplices¹ of \mathcal{K} . The relation \prec is defined as: $\tau \prec \sigma \Leftrightarrow \{\tau \subset \sigma \text{ and } \dim \tau = \dim \sigma - 1\}$. Alternatively, we say that τ is the *facet* of σ and σ is a *cofacet* of τ . The boundary $bd(\sigma)$ and the coboundary $cbd(\sigma)$ of a simplex are defined as: $bd(\sigma) = \{\tau | \tau \prec \sigma\}$ and $cbd(\sigma) = \{\rho | \sigma \prec \rho\}$. A function $f : \mathcal{K} \rightarrow \mathbb{R}$ is called a *discrete Morse function* if it assigns higher values to cofacets, with at most one exception at each simplex. Specifically, for a function $f : \mathcal{K} \rightarrow \mathbb{R}$ for every $\sigma \in \mathcal{K}$, let $\mathcal{N}_1(\sigma) = \#\{\rho \in cbd(\sigma) | f(\rho) \leq f(\sigma)\}$ and $\mathcal{N}_2(\sigma) = \#\{\tau \in bd(\sigma) | f(\tau) \geq f(\sigma)\}$. Such a function is called a discrete Morse function if for every $\sigma \in \mathcal{K}$, $\mathcal{N}_1(\sigma) + \mathcal{N}_2(\sigma) \leq 1$. If $\mathcal{N}_1(\sigma) = \mathcal{N}_2(\sigma) = 0$, then the simplex σ is *critical*, else it is *regular*.

¹ A d -dimensional simplex σ^d may be denoted either as σ or σ^d depending on whether we wish to emphasize its dimension.

A pair of simplices $\langle \alpha^m, \beta^{(m+1)} \rangle$ with $\alpha < \beta$ and $f(\alpha) \geq f(\beta)$ determines a *gradient pair*. Each simplex must occur in at most one gradient pair of \mathcal{V} . A *discrete gradient vector field* \mathcal{V} corresponding to a discrete Morse function f is a collection of simplicial pairs $\langle \alpha^{(p)}, \beta^{(p+1)} \rangle$ such that $\langle \alpha^{(p)}, \beta^{(p+1)} \rangle \in \mathcal{V}$ if and only if $f(\alpha) \geq f(\beta)$.

A simplicial sequence $\{\sigma_0^m, \tau_0^{m+1}, \sigma_1^m, \tau_1^{m+1}, \dots, \sigma_q^m, \tau_q^{m+1}, \sigma_{q+1}^m\}$ consisting of distinct simplices $(\sigma_i < \tau_i) \in \mathcal{V}$ and $\sigma_{i+1} < \tau_i$ is called a *gradient path* of f .

2.2. The Hasse graph of a simplicial complex

The *Hasse graph* $\mathcal{H}_{\mathcal{K}}$ of a simplicial complex \mathcal{K} is an undirected graph whose vertices are in one-to-one correspondence with the simplices of the complex. To every simplex $\sigma_{\mathcal{K}}^d \in \mathcal{K}$ associate a vertex $\sigma_{\mathcal{H}}^d \in \mathcal{H}_{\mathcal{K}}$. The edges in the Hasse graph are determined by facet incidences. $\mathcal{H}_{\mathcal{K}}$ contains an edge between a vertex that represents² simplex σ^d and a vertex that represents simplex τ^{d-1} if and only if $\tau < \sigma$.

We refer to the set of vertices in $\mathcal{H}_{\mathcal{K}}$ representing d -dimensional simplices as the *d-level* of the Hasse graph. The *d-interface* of $\mathcal{H}_{\mathcal{K}}$ is the subgraph consisting of vertices in the d -level and the $(d - 1)$ -level of $\mathcal{H}_{\mathcal{K}}$ together with all the edges connecting these two levels.

The Hasse graph $\mathcal{H}_{\mathcal{K}}$ of a D -dimensional complex \mathcal{K} has $(D + 1)$ levels and D interfaces. To understand Morse theory in terms of Hasse graph one needs to assign orientations to it.

Definition 2 (*Oriented Hasse graph, up-edges, down-edges*). If we assign orientations to all edges of Hasse graph $\mathcal{H}_{\mathcal{K}}$, we obtain an *oriented Hasse graph* denoted by $\overline{\mathcal{H}_{\mathcal{K}}}$. In graph $\overline{\mathcal{H}_{\mathcal{K}}}$, for two simplices σ^d, τ^{d-1} , an edge $\tau^{d-1} \rightarrow \sigma^d \in \overline{\mathcal{H}_{\mathcal{K}}}$ going from lower dimensional simplex τ^{d-1} to a higher dimensional simplex σ^d is called an *up-edge*. The edge $\sigma^d \rightarrow \tau^{d-1} \in \overline{\mathcal{H}_{\mathcal{K}}}$ is called a *down-edge*.

If we orient $\mathcal{H}_{\mathcal{K}}$ in such a way that all edges are down-edges, then this orientation corresponds to the trivial gradient vector field on complex \mathcal{K} for which all simplices are critical. We call this the *default orientation* on $\mathcal{H}_{\mathcal{K}}$.

Matching-based orientation. Start with the default orientation on $\mathcal{H}_{\mathcal{K}}$. Associate a matching \mathcal{M} to $\mathcal{H}_{\mathcal{K}}$. If an edge $\langle \tau^{d-1}, \sigma^d \rangle \in \mathcal{M}$, then *reverse* the orientation of that edge to $\tau^{d-1} \rightarrow \sigma^d \in \overline{\mathcal{H}_{\mathcal{K}}}$. We require the matching induced orientation to be such that the graph $\overline{\mathcal{H}_{\mathcal{K}}}$ is a directed acyclic graph. Chari [22] first observed that every matching-based orientation of $\mathcal{H}_{\mathcal{K}}$ that leaves the graph $\overline{\mathcal{H}_{\mathcal{K}}}$ acyclic corresponds to a unique gradient vector field on complex \mathcal{K} . For such a matching-based acyclic orientation of the graph, every up-edge in the oriented Hasse graph corresponds to a gradient pair and every unmatched vertex corresponds to a critical simplex of the gradient vector field. Not every matching-based orientation of $\mathcal{H}_{\mathcal{K}}$ will leave $\overline{\mathcal{H}_{\mathcal{K}}}$ acyclic. Fig. 1 shows a simplicial complex and a matching-based orientation of the Hasse graph. We can now define the Max Morse Matching Problem more formally.

Definition 3 (*Max Morse Matching Problem*). A discrete gradient vector field that maximizes the number of gradient pairs over the set of all discrete gradient vector fields on a simplicial complex \mathcal{K} is known as a *Maximum Morse Matching* on \mathcal{K} . The Max Morse Matching Problem is to find such an optimal Morse Matching. In terms of the Hasse graph, a *Maximum Morse Matching* may be defined as the maximum cardinality of an acyclic matching.

We now discuss a few properties of cycles and paths in a matching-based orientation of $\mathcal{H}_{\mathcal{K}}$. Matching-based orientations have the interesting property that all cycles are restricted to a fixed interface in the oriented Hasse graph. In other words, if a cycle were to span multiple interfaces in the Hasse graph, then it will violate the condition that the orientation is matching-based. Similarly, all edges in a given path belong to a unique interface of the Hasse graph. Also, in a matching-based orientation, source nodes and sink nodes in the d -interface are not involved in any cycles in the d -interface.

Definition 4 (*Source and sink nodes*). A simplex σ^d is a *source node* for the d -interface if it has only outgoing edges to $d - 1$ simplices. If in addition, simplex σ^d is matched to a $(d + 1)$ -simplex, then it is a *regular source node* for the d -interface, else it is a *critical source node*. Similarly, a simplex τ^{d-1} is a *sink node* for the d -interface if it has only incoming edges from d -simplices. If τ^{d-1} is matched to a $(d - 2)$ -simplex, then it is known as a *regular sink node* else it is known as a *critical sink node*.

3. A $(D+1)/(D^2+D+1)$ -factor approximation algorithm for simplicial complexes

We now describe an approximation algorithm for the Max Morse Matching Problem that is applicable to simplicial complexes. The idea is to first compute a maximum-cardinality matching, and in a subsequent step, remove any cycles that

² From here on, for the sake of brevity, while referring to the vertex in $\mathcal{H}_{\mathcal{K}}$ representing simplex σ^d , we drop the suffix \mathcal{H} from $\sigma_{\mathcal{H}}^d$, i.e., instead of referring to it as vertex $\sigma_{\mathcal{H}}^d$ we refer to it as simplex σ^d .

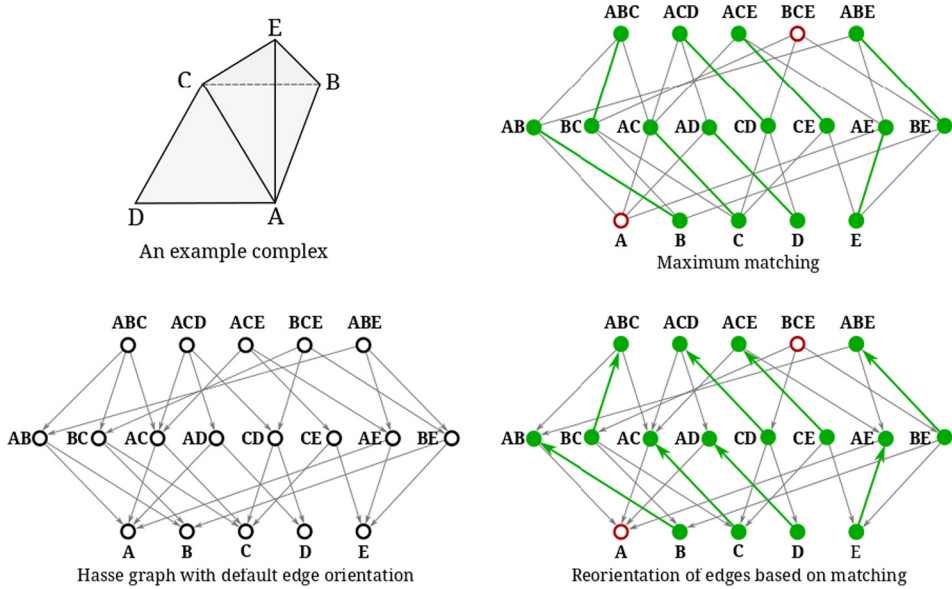


Fig. 1. Consider a simplicial complex with five triangles (ABC, ACD, ACE, BCE, ABE). We obtain the oriented Hasse graph for a simplicial complex (left) and its matching induced orientation (right). Two simplices are critical (hollow) and others are regular (filled).

may be introduced due to the matching induced orientation. The key steps are outlined in Algorithm 1. We begin with notes on notations and definitions.

Notation. When we denote an up-edge as $\chi(\alpha, \beta)$, we mean to say that it is an edge connecting simplex α^{d-1} to simplex β^d and is labeled as χ . We may write it either as $\chi(\alpha, \beta)$ or χ depending on whether we want to emphasize vertices incident on χ . The corresponding down-edge with reversed orientation is denoted as $\overline{\chi}$ or $\overline{\chi}(\beta, \alpha)$.

Definition 5 (Leading up-edges of an up-edge). In an oriented Hasse graph $\overline{\mathcal{H}_K}$, if we have an up-edge $\chi_1(\alpha_1, \beta_1)$ followed by a down-edge $\overline{\chi_2}(\beta_1, \alpha_2)$ followed by up-edge $\chi_3(\alpha_2, \beta_2)$, we say that χ_3 is a leading up-edge of χ_1 .

Definition 6 (Facet-edges of a simplex). In an oriented Hasse graph $\overline{\mathcal{H}_K}$, for a simplex σ^d (where $d \geq 1$), the set of oriented edges between σ^d to $(d - 1)$ -simplices incident on σ^d (along with respective orientations) are known as the facet-edges of σ^d .

Given a Hasse graph \mathcal{H}_K on complex \mathcal{K} , Algorithm 1 begins by computing maximum-cardinality graph matching on graph \mathcal{H}_K and then uses this matching to induce an orientation on \mathcal{H}_K . Let $\overline{\mathcal{H}_K}$ denote the oriented Hasse graph based on graph matching and \mathcal{H}_V denote the output graph. While there exists an up-edge χ in $\overline{\mathcal{H}_K}$, we make χ a seed-edge and use it as a starting point for a BFS-like traversal on graph $\overline{\mathcal{H}_K}$. This traversal is done using procedure **BFSComponent()**, which returns a set of edges \mathcal{C}_χ . The edge-component \mathcal{C}_χ of a seed edge χ is the set of edges discovered in the BFS-like traversal of graph $\overline{\mathcal{H}_K}$ with χ as the start edge. Each time we discover a new edge-component, we delete it from $\overline{\mathcal{H}_K}$, and add it to \mathcal{H}_V . We exit the while loop when all up-edges are exhausted.

If a simplex σ^d is either a critical node or a regular source node, then its facet-edges are not reachable in the BFS traversal through any of the up-edges in $\overline{\mathcal{H}_K}$. In a final step, we include all remaining edges from $\overline{\mathcal{H}_K}$ to \mathcal{H}_V .

The procedure **BFSComponent()** computes the component edges by processing edges from the queue one at a time. Let $\chi_0(\alpha_0, \beta_0)$ be the edge at the top of the queue. We add all the facet-edges of simplex β_0 to the edge-component \mathcal{C} . We now examine the leading up-edges of χ_0 . If $\chi_i(\alpha_i, \beta_i)$ is a leading up-edge of χ_0 , then we check if the addition of facet-edges of simplex β_i to \mathcal{C} creates cycles. If it does, then we classify χ_i as a backward edge, reverse the orientation of χ_i , and add the facet-edges of β_i to \mathcal{C} . If this addition does not introduce cycles, then we classify χ_i as a forward edge, and enqueue it in the queue of up-edges. Please refer to Fig. 2. Enqueuing χ_i guarantees that at some stage when χ_i gets dequeued, we will end up adding facet-edges of simplex β_i to \mathcal{C} . When the queue is exhausted, \mathcal{C} contains the entire edge-component of some seed-edge.

We first prove an acyclicity lemma on edge-components returned by procedure **BFSComponents()** in Algorithm 1.

Lemma 1. The graph induced by edges in an edge-component is acyclic.

Algorithm 1 The frontier edges algorithm.

Input: Simplicial complex \mathcal{K}
Output: Graph \mathcal{H}_V , an acyclic matching-based orientation of Hasse graph \mathcal{H}_K of \mathcal{K} .

- 1: Construct Hasse graph \mathcal{H}_K of \mathcal{K} .
- 2: Perform maximum-cardinality graph matching on \mathcal{H}_K .
- 3: Let $\overline{\mathcal{H}_K}$ denote the matching induced orientation of \mathcal{H}_K and $\mathcal{E}(\overline{\mathcal{H}_K})$ its edge set.
- 4: Initialize the edge set of \mathcal{H}_V , $\mathcal{E}(\mathcal{H}_V) \leftarrow \emptyset$.
- 5: **while** $\exists \chi \in \mathcal{E}(\overline{\mathcal{H}_K})$ such that χ is an up-edge **do**
- 6: $C_\chi \leftarrow \text{BFSCOMPONENT}(\overline{\mathcal{H}_K}, \chi)$
- 7: $\mathcal{E}(\overline{\mathcal{H}_K}) \leftarrow \mathcal{E}(\overline{\mathcal{H}_K}) \setminus C_\chi$
- 8: $\mathcal{E}(\mathcal{H}_V) \leftarrow \mathcal{E}(\mathcal{H}_V) \cup C_\chi$
- 9: **end while**
- 10: $\mathcal{E}(\mathcal{H}_V) \leftarrow \mathcal{E}(\mathcal{H}_V) \cup \mathcal{E}(\overline{\mathcal{H}_K})$
- 11: **procedure** $\text{BFSCOMPONENT}(\overline{\mathcal{H}_K}, \chi)$
- 12: $C \leftarrow \emptyset$
- 13: Initialize the queue $Q \leftarrow \emptyset$
- 14: **enqueue**(Q, χ)
- 15: **while** Q is non-empty **do**
- 16: $\chi_0(\alpha_0, \beta_0) \leftarrow \text{dequeue}(Q)$
- 17: $C \leftarrow C \cup \text{facetEdges}(\beta_0)$
- 18: **for** every leading up-edge $\chi_i(\alpha_i, \beta_i)$ of χ_0 **do**
- 19: **if** the graph induced by edges in $(C \cup \text{facetEdges}(\beta_i))$ has cycles **then**
- 20: Reverse orientation of χ_i in graph $\overline{\mathcal{H}_K}$
- 21: $C \leftarrow C \cup \text{facetEdges}(\beta_i)$
- 22: **else**
- 23: **enqueue**(Q, χ_i)
- 24: **end if**
- 25: **end for**
- 26: **end while**
- 27: **return** C
- 28: **end procedure**

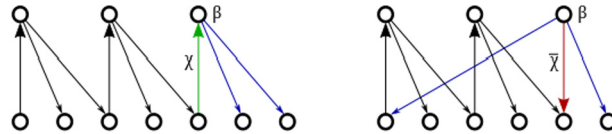


Fig. 2. Two cases in $\text{BFSCOMPONENT}()$. Left: A forward edge χ is identified. The edge χ along with the down-edges incident on β are added to the edge-component. Right: A backward edge $\bar{\chi}$ is identified. The edge $\bar{\chi}$ along with the down-edges incident on β are added to the edge-component.

Proof. Consider the graph induced by edges in edge-component C belonging to the d -interface. We know that an up-edge, say χ_j is classified as a forward edge if and only if the inclusion of χ_j does not create a cycle with up-edges that were included prior to χ_j in edge-component C . Hence, we can be sure that inclusion of set of all forward edges does not create cycles. Moreover, every time a backward edge, say $\chi_i(\alpha_i, \beta_i)$ is encountered, we include the inverse orientation of χ_i in C , which creates a sink node at α_i and source node at β_i for the d -interface of the Hasse graph. Also, all paths emanating from $(d - 1)$ -simplices that were visited in a previous edge-component go to sinks. Furthermore, every down-edge in C is either (a.) incident on a sink belonging to C , or (b.) incident on a $(d - 1)$ -simplex that is incident on a forward edge belonging to C , or (c.) incident on a $(d - 1)$ -simplex that was visited in a previous edge-component. In either case, it is easy to see that all flow terminates at sinks. Hence, the graph induced by edges in a particular edge-component is acyclic. \square

Lemma 2. The output graph \mathcal{H}_V is acyclic.

Proof. We prove this claim via induction over sequential addition of edge-components.

Base Case: To begin with, the output graph \mathcal{H}_V is the empty graph. From Lemma 1, we know that the graph induced by edges in an edge-component is acyclic. So, \mathcal{H}_V remains acyclic following the addition of the first edge-component C_1 to \mathcal{H}_V .

Inductive Hypothesis: Suppose that following the addition of edges belonging to i^{th} edge-component C_i , \mathcal{H}_V remains acyclic.

Now, we need to prove that following the addition of edges belonging to C_{i+1} , \mathcal{H}_V remains acyclic. To begin with, using Lemma 1, we note that the graph induced by C_{i+1} is acyclic. So, if there does exist a cycle in \mathcal{H}_V following the addition of C_{i+1} , then a forward up-edge of this cycle must belong to C_{i+1} , and a forward up-edge must belong to an edge-component C_{j_k} , where $j_k < (i + 1)$. In particular, this means that there exists a down-edge belonging to a component C_{j_k} that is incident on simplex α_1 such that a forward edge $\chi_1(\alpha_1, \beta_1) \in C_{i+1}$. But, if α_1 was reachable while traversing C_{j_k} , then $\chi_1(\alpha_1, \beta_1)$ would have been classified as a forward edge in C_{j_k} , i.e., $\chi_1(\alpha_1, \beta_1) \in C_{j_k}$ – a contradiction. Hence, such cycles do not exist. Finally, in line 10 of Algorithm 1, after having added all edge-components, we add all the facet-edges of d -simplices that are

either unmatched or facet-edges of d -simplices that are matched to one of their cofacets. In such cases, they act as source nodes within d -interfaces and do not introduce cycles because all cycles are restricted to the d -interface. \square

Lemma 3. *The output graph \mathcal{H}_V is a matching-based acyclic orientation of undirected Hasse graph of the complex \mathcal{H}_K .*

Proof. We first prove that \mathcal{H}_V is an orientation of \mathcal{H}_K , i.e., for every undirected edge in \mathcal{H}_K there is a corresponding directed edge in \mathcal{H}_V . To prove this, we will show that for every simplex β^d , all undirected edges from β to its facets in \mathcal{H}_K have a corresponding oriented edge in \mathcal{H}_V .

Case 1: Suppose that β is matched to one of its facets in matching induced oriented graph $\overline{\mathcal{H}_K}$. Then, this up-edge incident on β was classified either as a forward edge or as a backward edge. In either case, all its facet-edges are inserted in \mathcal{H}_V in procedure **BFSComponent**().

Case 2: Now suppose that β is either unmatched or it is matched to one of its cofacets. Then, none of its facet-edges can be reached through a graph traversal that starts with some up-edge in $\overline{\mathcal{H}_K}$. Therefore, these facet-edges are not a part of any of the edge-components, and they are all down-edges. However, in line 10 of **Algorithm 1**, all these *remainder* edges are included in \mathcal{H}_V .

Since the above two cases hold true for every simplex σ^d with $d \geq 1$, this proves that \mathcal{H}_V is an orientation of graph \mathcal{H}_K . The orientation of \mathcal{H}_V is matching-based since the up-edges that are included are subset of edges coming from maximum-cardinality bipartite matching. In **Lemma 2**, we have already proved that graph \mathcal{H}_V is acyclic. Hence, the output graph \mathcal{H}_V is a matching-based acyclic orientation of undirected Hasse graph of the complex \mathcal{H}_K . \square

Definition 7 (*Classified edges, frontier edges*). An edge marked within the **BFSComponent**() as forward or backward is called a *classified edge*. A leading up-edge that is not yet classified is called a *frontier edge*.

We establish the approximation ratio using a counting argument that works specifically for simplicial complexes. We refer to this argument as the frontier edges argument. Suppose we are processing an edge-component that belongs to the d -interface of the Hasse graph for some $d \leq D$. Let the iterator variable i count the number of up-edges in the edge-component that have so far been classified as either forward or backward. If at the end of the i^{th} iteration, there are $|\mathcal{F}_i|$ forward edges, $|\mathcal{B}_i|$ backward edges and $|\mathcal{Z}_i|$ frontier edges, then our approximation ratio is $|\mathcal{F}_i|/(|\mathcal{F}_i|+|\mathcal{B}_i|+|\mathcal{Z}_i|)$. In other words, we assume the worst case scenario where all the frontier edges are *possibly backward*. In every iteration of the BFS, we classify one of the frontier edges as a forward edge or a backward edge and then update the ratio until we exhaust the entire edge-component. In the $(i+1)^{\text{th}}$ iteration, if a frontier edge is classified as a forward edge, then the number of forward edges will be $|\mathcal{F}_{i+1}| = (|\mathcal{F}_i| + 1)$ and the number of frontier edges will be $|\mathcal{Z}_{i+1}| = (|\mathcal{Z}_i| + d - 1)$. If a frontier edge is classified as a backward edge, then the number of backward edges will be $|\mathcal{B}_{i+1}| = (|\mathcal{B}_i| + 1)$ and the number of frontier edges will be $|\mathcal{Z}_{i+1}| = (|\mathcal{Z}_i| - 1)$.

Lemma 4. *The number of forward edges in an edge-component belonging to the d -interface of the Hasse graph is at least $(d+1)/(d^2+d+1)$ fraction of the total number of up-edges in the edge-component.*

Proof. We will use induction to prove our claim.

Base Case: The seed edge χ_0 of the edge-component is naturally a forward edge. We note that any cycle in the Hasse graph of a simplicial complex has minimum length 6 and involves at least 3 up-edges. Since this does not hold for general, regular cell complexes, simplicial input is crucial for the proof to work. Cycles do not appear until after two iterations. These two iterations constitute the base case. Therefore, $|\mathcal{F}_1| = 1$, $|\mathcal{B}_1| = 0$ and $|\mathcal{Z}_1| = 0$. Also, the leading up-edges of χ_0 are also forward edges. If χ_0 has no leading up-edges, then the edge-component is exhausted and $|\mathcal{F}_1|/(|\mathcal{F}_1|+|\mathcal{B}_1|) = 1$. If χ_0 has K leading up-edges, each such edge has, in turn, at most j_k leading up-edges, then the total number of forward edges will be $|\mathcal{F}_2| = 1 + K$, $|\mathcal{B}_2| = 0$ and $|\mathcal{Z}_2| = \sum_{k=1}^K j_k$. It is easy to check that the worst case for ratio $|\mathcal{F}_2|/(|\mathcal{F}_2|+|\mathcal{B}_2|+|\mathcal{Z}_2|)$ occurs when $K = d$ and $j_k = d$ for each k . This gives us the worst case ratio for the quantity $|\mathcal{F}_2|/(|\mathcal{F}_2|+|\mathcal{B}_2|+|\mathcal{Z}_2|)$ to be $(d+1)/(d^2+d+1)$. Please refer to **Fig. 3**.

Induction Hypothesis: After i iterations of BFS, the ratio $|\mathcal{F}_i|/(|\mathcal{F}_i|+|\mathcal{B}_i|+|\mathcal{Z}_i|) \geq (d+1)/(d^2+d+1)$.

Induction Step: For the $(i+1)$ th iteration, suppose one of the frontier edges is classified as a forward edge. Then, $|\mathcal{F}_{i+1}| = (|\mathcal{F}_i| + 1)$ and $|\mathcal{Z}_{i+1}| \leq (|\mathcal{Z}_i| + d - 1)$. Note that $(|\mathcal{Z}_i| + d - 1)$ is the worst case estimate for $|\mathcal{Z}_{i+1}|$ assuming that the newly included forward edge has d leading up-edges. Therefore, the numerator of the ratio $|\mathcal{F}_i|/(|\mathcal{F}_i|+|\mathcal{B}_i|+|\mathcal{Z}_i|)$ increases by 1 whereas the denominator increases by d . Also, we have $1/d > (d+1)/(d^2+d+1)$. Using the elementary fact that if $\frac{A}{B} \geq \frac{E}{F}$ and $\frac{C}{D} \geq \frac{E}{F}$, then $\frac{(A+C)}{(B+D)} \geq \frac{E}{F}$ for non-negative values of A, B, C, D, E and F , we get:

$$\begin{aligned} \frac{|\mathcal{F}_{i+1}|}{(|\mathcal{F}_{i+1}| + |\mathcal{B}_{i+1}| + |\mathcal{Z}_{i+1}|)} &\geq \frac{|\mathcal{F}_i| + 1}{(|\mathcal{F}_i| + |\mathcal{B}_i| + |\mathcal{Z}_i|) + d} \\ &\geq \frac{(d+1)}{(d^2 + d + 1)} \end{aligned}$$

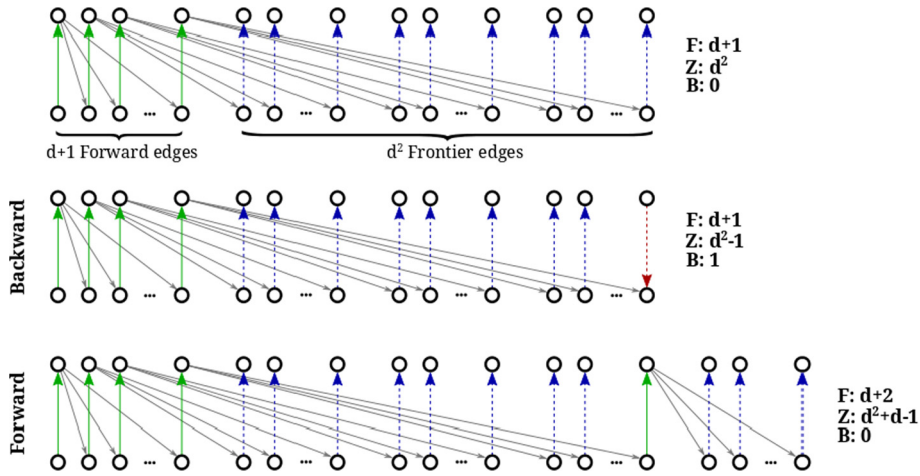


Fig. 3. The frontier edges argument. Top: The base case with $d + 1$ forward edges, d^2 frontier edges and no backward edges. The ratio of forward edges and total number of up-edges is $\frac{d+1}{d^2+d+1}$. Middle: When one of the frontier edges is classified as backward, the ratio remains the same. Bottom: When one of the frontier edges is classified as a forward edge, the ratio improves to $\frac{d+2}{d^2+2d+1}$.

On the other hand, if a frontier edge is classified as a backward edge, then $|\mathcal{B}_{i+1}| = (|\mathcal{B}_i| + 1)$ and $|\mathcal{Z}_{i+1}| = (|\mathcal{Z}_i| - 1)$. So, the numerator and the denominator of the ratio $\frac{|\mathcal{F}_{i+1}|}{(|\mathcal{F}_{i+1}|+|\mathcal{B}_{i+1}|+|\mathcal{Z}_{i+1}|)}$ remain unchanged, which gives us $\frac{|\mathcal{F}_{i+1}|}{(|\mathcal{F}_{i+1}|+|\mathcal{B}_{i+1}|+|\mathcal{Z}_{i+1}|)} = \frac{|\mathcal{F}_i|}{(|\mathcal{F}_i|+|\mathcal{B}_i|+|\mathcal{Z}_i|)}$. In both cases, the bound holds after $(i + 1)$ iterations. \square

Since every edge-component that belongs to a d -interface achieves a ratio of at least $(d+1)/(d^2+d+1)$ edges, if we sum over all the edge-components, we get the ratio $(d+1)/(d^2+d+1)$ for that d -interface. In other words, we preserve at least $(d+1)/(d^2+d+1)$ of the total number of matchings at every d -interface. The ratio $(d+1)/(d^2+d+1)$ becomes worse with increasing d . So, the worst case ratio is $(D+1)/(D^2+D+1)$ where D is the dimension of the complex. Therefore, we get the following result on the approximation ratio.

Theorem 5. Algorithm 1 computes a $(D+1)/(D^2+D+1)$ -factor approximation for Max Morse Matching Problem on simplicial complexes of dimension D .

Proof. Note that for a matching-based orientation, the number of regular simplices is equal to twice the cardinality of the matching. Let $|\mathcal{M}|$ denote the cardinality of the maximum matching. Then, $2|\mathcal{M}|$ is an upper bound on the Max Morse Matching, i.e., optimal number of regular simplices $\leq 2|\mathcal{M}|$. Since we preserve at least $(D+1)/(D^2+D+1)$ of these matchings, the number of regular simplices we obtain is at least $2 \frac{(D+1)}{(D^2+D+1)} |\mathcal{M}| \geq \frac{(D+1)}{(D^2+D+1)} OPT$. Therefore, Algorithm 1 provides a $\frac{D+1}{(D^2+D+1)}$ -factor approximation for the Max Morse Matching Problem on simplicial complexes. \square

Remark 1. It is worth noting that every time we select an up-edge χ in the while loop from lines 5–9 of Algorithm 1, if we were to reverse the orientations of all leading up-edges of χ , we obtain a trivial $1/(D+1)$ -factor algorithm. This naïve algorithm is described in more detail in Section 5.1.

3.1. A 5/11-factor approximation for 2-dimensional simplicial complexes using frontier edges algorithm

In this section, we observe that we can further tighten our analysis of Algorithm 1 by restricting the problem to 2-dimensional simplicial complexes. We exploit the geometry of 2-complexes as proved in Lemma 6 in order to establish an improved ratio in the base case.

Lemma 6. If α is a forward edge and β_1 is a leading forward edge of edge α and if γ_1 and γ_2 are leading up-edges of β_1 , then only one of the two edges γ_1 and γ_2 can possibly be a backward edge that creates a cycle with edge α .

Proof. Without loss of generality, in this proof, we will use concrete labeling of simplices. We make an elementary geometric observation to prove this claim. Suppose α is a forward edge between a 1-simplex, say AB matched to a 2-simplex ABC. So, α can alternatively be denoted as edge AB–ABC. Now, suppose 1-simplex BC is matched to another 2-simplex BCD constituting forward edge β_1 , then of the two 1-simplices BD and CD, BD can possibly match a 2-simplex, say BDA, which effectively makes edge BD–BDA (say γ_1) a backward edge. However it is impossible to have a forward edge incident on

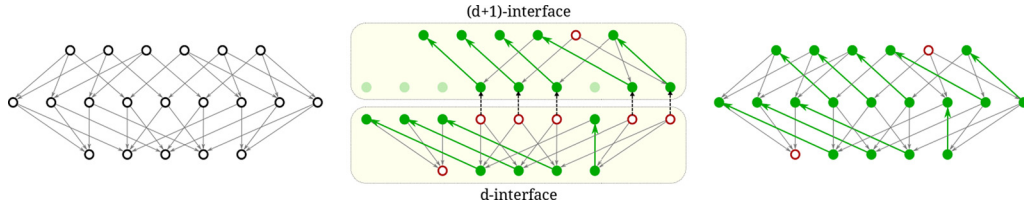


Fig. 4. An example illustrating Algorithm 2. Left: The input Hasse graph. Middle: The d -simplices matched during execution of the algorithm on the d -interface are deleted from the $(d + 1)$ -interface. Right: The final matching is obtained by combining matchings from all interfaces.

1-simplex CD (say γ_2) that is also simultaneously incident on 1-simplex AB since any 2-simplex has at most three vertices. Hence proved. \square

Lemma 7. *The number of forward edges is at least $5/11$ fraction of the total number of up-edges in the edge-component.*

Proof. Once again, we will use induction to prove our claim.

Base Case: In the case of 2-manifolds, we can count up to three levels of BFS for the base case, which in turn gives us an improvement in ratio. The seed edge α of the edge-component is clearly a forward edge. We note that any cycle in the Hasse graph of a simplicial complex has minimum length 6. Therefore, $|\mathcal{F}_1| = 1$, $|\mathcal{B}_1| = 0$ and $|\mathcal{Z}_1| = 0$. Also, the leading up-edges of α (if any) are also forward edges. If α has no leading up-edges, then the edge-component is exhausted and $|\mathcal{F}_1|/(|\mathcal{F}_1|+|\mathcal{B}_1|) = 1$. If α has one leading up-edge β_1 , then $|\mathcal{F}_2| = 2$, $|\mathcal{B}_2| = 0$ and $|\mathcal{Z}_2| = 2$. Therefore, $|\mathcal{F}_2|/(|\mathcal{F}_2|+|\mathcal{B}_2|+|\mathcal{Z}_2|) = 1/2$. If α has two leading up-edges β_1 and β_2 , then $|\mathcal{F}_2| = 3$, $|\mathcal{B}_2| = 0$ and $|\mathcal{Z}_2| = 4$. Therefore, $|\mathcal{F}_2|/(|\mathcal{F}_2|+|\mathcal{B}_2|+|\mathcal{Z}_2|) = 3/7$. By Lemma 6, both leading up-edges of β_1 , γ_1 and γ_2 , cannot be backward. So, suppose one of them (say γ_1) is backward and γ_2 is forward. Then, $|\mathcal{F}_3| = 4$, $|\mathcal{B}_3| = 1$ and $|\mathcal{Z}_3| = 4$. Therefore, $|\mathcal{F}_3|/(|\mathcal{F}_3|+|\mathcal{B}_3|+|\mathcal{Z}_3|) = 4/9$. Similarly, we must also consider the leading up-edges of β_2 , of which at most one of them can be backward. The worst case occurs for the configuration when exactly one leading up-edge each of β_1 and β_2 are backward. This configuration gives $|\mathcal{F}_3| = 5$, $|\mathcal{B}_3| = 2$ and $|\mathcal{Z}_3| = 4$. Hence, $|\mathcal{F}_3|/(|\mathcal{F}_3|+|\mathcal{B}_3|+|\mathcal{Z}_3|) = 5/11$.

Induction Hypothesis: Following i iterations of BFS, the ratio $|\mathcal{F}_i|/(|\mathcal{F}_i|+|\mathcal{B}_i|+|\mathcal{Z}_i|) \geq 5/11$.

Induction Step: For the $(i + 1)$ th iteration, suppose one of the frontier edges is classified as a forward edge. Then, $|\mathcal{F}_{i+1}| = (|\mathcal{F}_i| + 1)$ and $|\mathcal{Z}_{i+1}| = (|\mathcal{Z}_i| + 1)$. Therefore, the numerator of the ratio $|\mathcal{F}_i|/(|\mathcal{F}_i|+|\mathcal{B}_i|+|\mathcal{Z}_i|)$ increases by 1 whereas the denominator increases by 2. However, since $1/2 > 5/11$, we have $|\mathcal{F}_{i+1}|/(|\mathcal{F}_{i+1}|+|\mathcal{B}_{i+1}|+|\mathcal{Z}_{i+1}|) = (5+1)/(11+1) > 5/11$. On the other hand, if a frontier edge is classified as a backward edge, then $|\mathcal{B}_{i+1}| = (|\mathcal{B}_i| + 1)$ and $|\mathcal{Z}_{i+1}| = (|\mathcal{Z}_i| - 1)$. So, the numerator and the denominator of ratio $|\mathcal{F}_{i+1}|/(|\mathcal{F}_{i+1}|+|\mathcal{B}_{i+1}|+|\mathcal{Z}_{i+1}|)$ remain unchanged, which gives us $|\mathcal{F}_{i+1}|/(|\mathcal{F}_{i+1}|+|\mathcal{B}_{i+1}|+|\mathcal{Z}_{i+1}|) = |\mathcal{F}_i|/(|\mathcal{F}_i|+|\mathcal{B}_i|+|\mathcal{Z}_i|)$. When all the up-edges of the edge-component are exhausted, we don't have anymore frontier edges and the ratio for the edge-component after processing $|\mathcal{F}|$ forward edges and $|\mathcal{B}|$ backward edges will be $|\mathcal{F}|/(|\mathcal{F}|+|\mathcal{B}|)$, and by our inductive argument the ratio will be at least $5/11$. \square

Once again, since every edge-component achieves a ratio of at least $5/11$ edges, if we sum over all the edge-components, we get the following theorem as an immediate outcome of the lemma above.

Theorem 8. *Algorithm 1 is a $5/11$ -factor approximation algorithm for Max Morse Matching Problem when restricted to 2-dimensional simplicial complexes.*

4. Approximation algorithms for simplicial manifolds

4.1. A $2/(D+1)$ -factor approximation algorithm for simplicial manifolds

We will restrict our attention to manifolds without boundary. The key idea in Algorithm 2 is that the matching is constructed within one d -interface at a time, starting with the lowest interface and ending with the highest one. For manifolds, this is advantageous because it allows us to count matched/critical simplices differently. In particular, every d -simplex (where $1 \leq d \leq D - 1$) is given two chances to get matched. Please refer to Fig. 4. We first try to match a d -simplex, say σ^d , while constructing the Morse matching for the d -interface. If σ^d remains unmatched for the d -interface, then we try to match it for the $(d + 1)$ -interface. The trick of giving a second chance to critical simplices works fine for all dimensions except for D -dimensional critical simplices. Fortunately, for manifolds, we can easily design a vector field with only one critical simplex for dimension D . Since non-manifold-complexes may have unbounded number of critical D -simplices the analysis becomes non-trivial. For Algorithm 2, one may still derive approximation bounds for non-manifold complexes by using a line of reasoning analogous to one used in Section 4.2.1.

Algorithm 2 exploits special structures at the lowest and highest interface. For instance, for any D -dimensional manifold, there are well known algorithms in literature [1,18,20] for designing optimal gradient vector field for the 1-interface and the D -interface. See Appendix A of [20]. As noted in [20], we can associate a special graph structure to the D -interface.

Algorithm 2 The interface algorithm.

Input: Simplicial complex \mathcal{K}

Output: Graph $\mathcal{H}_{\mathcal{V}}$, an acyclic matching-based orientation of Hasse graph $\mathcal{H}_{\mathcal{K}}$ of \mathcal{K} .

- 1: ▷ **Notation:** [\mathcal{C}_d^{d-1} denotes the critical $(d-1)$ -simplices for d -interface. \mathcal{R}_d is the set of all regular simplices for d -interface and \mathcal{M}_d is the set of gradient pairs for d -interface. $\mathcal{E}(\mathcal{H}_{\mathcal{V}})$ denotes the edge set of $\mathcal{H}_{\mathcal{V}}$.]
- 2: Construct Hasse graph $\mathcal{H}_{\mathcal{K}}$ of \mathcal{K} .
- 3: $\mathcal{E}(\mathcal{H}_{\mathcal{V}})$ is initialized to default down-edge orientation on all edges.
- 4: **for** $d = 1$ to D **do**
- 5: $\mathcal{G}^d \leftarrow \text{extractInterface}(\mathcal{H}_{\mathcal{K}}, d)$
- 6: **if** $d = 1$ **then** Apply **1ComplexOpt**(\mathcal{G}^1)
- 7: **else if** $d = D$ **then** Apply **manifoldOpt**(\mathcal{G}^D)
- 8: **else** Apply **intermediateApx**(\mathcal{G}^d, d)
- 9: **end if**
- 10: **end for**
- 11: **procedure** DELETEANDREORIENT($\mathcal{C}_d^{d-1}, \mathcal{R}_d, \mathcal{M}_d$)
- 12: Reorient edges of $\mathcal{H}_{\mathcal{V}}$ based on matchings in edge set \mathcal{M}_d for the d -interface)
- 13: Delete nodes $\{\mathcal{C}_d^{d-1}, \mathcal{R}_d\}$ from $\mathcal{H}_{\mathcal{K}}$
- 14: **end procedure**
- 15: **procedure** 1COMPLEXOPT(\mathcal{W})
- 16: Apply the optimal algorithm on \mathcal{W} . (See **DFSoptimal**() in [Appendix A](#)).
- 17: **deleteAndReorient**($\mathcal{C}_1^0, \mathcal{R}_1, \mathcal{M}_1$)
- 18: **end procedure**
- 19: **procedure** MANIFOLDOPT(\mathcal{W})
- 20: Apply the optimal algorithm on the dual graph. (See **DFSoptimal**() in [Appendix A](#)).
- 21: Reorient edges of $\mathcal{H}_{\mathcal{V}}$ based on matchings in edge set \mathcal{M}_D for the D -interface.
- 22: **end procedure**
- 23: **procedure** INTERMEDIATEAPX(\mathcal{W}, d)
- 24: Apply [Algorithm 1](#) described in [Section 3](#) on graph \mathcal{W} .
- 25: For every unmatched simplex τ^{d-1} such that all its cofacets $\sigma_1^d \dots \sigma_K^d$ are also unmatched, choose one of the simplices $\sigma_i^d, i \in [1, K]$ and introduce the matching (τ, σ_i) .
- 26: **deleteAndReorient**($\mathcal{C}_d^{d-1}, \mathcal{R}_d, \mathcal{M}_d$)
- 27: **end procedure**

Definition 8 (*Dual graph*). The dual graph of a simplicial D -dimensional manifold \mathcal{K} is the graph whose vertices represent the D -simplices of \mathcal{K} and whose edges join two D -simplices with a common $(D-1)$ -facet.

For the sake of completeness, we describe the optimal algorithms for 1-interface and the D -interface in [Appendix A](#).

Like in [Algorithm 1](#), we first obtain the Hasse graph of complex \mathcal{K} . Subsequently, we extract the d -interface of the Hasse graph. For the d -interface, we design a Morse matching and reorient the output graph $\mathcal{H}_{\mathcal{V}}$ based on it. We then delete all the regular simplices of the d -interface and the critical $(d-1)$ -simplices. This updated Hasse graph is available for the next iteration when the $(d+1)$ -interface is extracted and so on. For the 1-interface and the D -interface, the optimal algorithms are applied to design the Morse matchings. For d -interfaces, where $1 < d < D$, procedure **intermediateApx**() of [Algorithm 2](#) is applied to design the Morse matchings.

We now describe procedure **intermediateApx**() for designing gradient vector field on the d -interface \mathcal{G}^d . [Algorithm 1](#) is essentially a maximum-matching followed by BFS-style cycle removal and hence can be performed on any bipartite graph. In particular, we apply it on graph \mathcal{G}^d for $1 < d < D$. After cycle removal (from [Algorithm 1](#)) we may have a situation where we have an unmatched simplex τ such that all its cofacets are also unmatched. In that case, we match τ with one of its cofacets. We perform this operation for all unmatched $(d-1)$ -simplices whose cofacets are also unmatched. This completes Morse matching for the d -interface. In procedure **deleteAndReorient**(), if σ^d is incident on simplex τ^{d-1} and if τ is regular at the $(d-1)$ -interface, then we are justified in deleting it while processing the d -interface since τ is a regular sink node for d -interface. The deletion of critical nodes does not affect the behavior of [Algorithm 2](#) per se. We delete them here because the procedure **intermediateApx**() is used as a subroutine in [Algorithm 3](#) where this deletion is crucial.

Lemma 9. The orientation of \mathcal{G}^d as computed by [Algorithm 2](#) is acyclic.

Proof. [Algorithm 1](#) provides an acyclic matching-based orientation of d -interface \mathcal{G}^d . So, step 1 of **intermediateApx**() does not introduce any cycles. Now consider an unmatched simplex τ^{d-1} such that all its cofacets $\sigma_1^d \dots \sigma_K^d$ are also unmatched. For a directed acyclic graph there is an ordering relation $\alpha > \beta$ if there is a directed path from vertex α to vertex β . Note that there is no ordering relation among $\sigma_1^d \dots \sigma_K^d$ since they are all critical. Introduction of the matching (τ, σ_i) introduces the ordering relations of the type $\sigma_j > \sigma_i$ for all $j \in [1, K]$ and $j \neq i$. Therefore, matching introduced by step 2 does not introduce any cycles. Hence, the orientation of \mathcal{G}^d as computed by [Algorithm 2](#) is acyclic. \square

Lemma 10. The orientation of the output graph $\mathcal{H}_{\mathcal{V}}$ is acyclic.

Proof. From Lemma 9, we conclude that the orientation for every d -interface \mathcal{G}^d , where $1 < d < D$, is acyclic. Further, optimal acyclic matchings are computed for 1-interface and D -interface, respectively. Combining these two facts with the observation that every directed path is restricted to a unique d -interface, we conclude that the orientation of output graph \mathcal{H}_ν is acyclic. \square

Now we introduce an idea that will help us prove approximation bounds for Algorithm 2. For the d -interface \mathcal{G}_d , let τ^{d-1} be a critical simplex, and let the set of cofacets of τ that are regular be $\{\beta_1, \beta_2, \dots, \beta_K\}$. From line 25 of procedure **intermediateApx()**, we know that this set is non-empty. Let β_i where $i \in [1 \dots K]$ be a cofacet of τ with minimum index after performing a topological sort on the d -interface.³ Now, let α_i be such that $\alpha_i < \beta_i$ and (α_i, β_i) is a gradient pair. Then, we can associate a canonical triplet $(\langle \alpha_i, \beta_i \rangle, \tau)$ to critical simplex τ^{d-1} . Note that such a unique canonical triplet is associated to every critical $(d - 1)$ -simplex.

Lemma 11. For $1 < d < D$, Algorithm 2 computes a $2/(d+2)$ -factor approximation to the Max Morse Matching restricted to the d -interface of the Hasse graph of a D -dimensional manifold.

Proof. Let $(\alpha_i^{d-1}, \beta_i^d)$ be a gradient pair. β_i^d has $(d + 1)$ facets, of which at least one facet, namely α_i , is regular. Therefore, the gradient pair (α_i, β_i) appears in at most d canonical triplets. We group (α_i, β_i) with all the critical $(d - 1)$ -simplices that contain $(\alpha_i^{d-1}, \beta_i^d)$ in their canonical triplets. Each critical $(d - 1)$ -simplex appears in a unique group. Each group contains exactly two regular simplices and at most d -critical simplices. So, for every group, we have the following ratio:

$$\frac{\text{\#matched simplices}}{\text{\#total simplices}} \geq \frac{2}{(d + 2)}$$

Hence, we obtain the approximation ratio of $2/(d+2)$ for the d -interface. \square

The minimum of the ratio $2/(d+2)$ over all d , where $1 < d < D$, is $2/(D+1)$. The 1-interface contributes to a single critical simplex when the optimal algorithm is employed. See Appendix A.

Finally, we consider the D -interface in the lemma below.

Lemma 12. After constructing Morse matching at the D -interface, the following ratio holds true:

$$\frac{\text{\#matched simplices}}{\text{\#total simplices}} \geq \frac{4}{D + 5}$$

Proof. Now consider the dual graph structure of the D -interface. The vertex degree of the dual graph is bounded by $D + 1$. Let N be the total number of vertices in the dual graph. So, the total number of edges in the dual graph is smaller than $\frac{(D+1)N}{2}$. Applying the optimal algorithm in Appendix A ensures that we have only one critical simplex in the dual graph. The following ratio holds true:

$$\frac{\text{\#matched simplices}}{\text{\#total simplices}} \geq \frac{2(N - 1)}{(\frac{D+1}{2} + 1)N} = \frac{2(N - 1)}{(\frac{D+1}{2} + 2)N - N}$$

We have $N \geq D + 2$ for a D -manifold without boundary because all $D + 1$ D -simplices adjacent to a given D -simplex are distinct from each other. Further, $(D + 2) \geq (\frac{D+1}{2} + 2)$ for $D \geq 1$. So, $N \geq (\frac{D+1}{2} + 2)$ for $D \geq 1$. Hence, the above expression can be rewritten as

$$\frac{\text{\#matched simplices}}{\text{\#total simplices}} \geq \frac{2(N - 1)}{(\frac{D+1}{2} + 2)N - (\frac{D+1}{2} + 2)} = \frac{2(N - 1)}{(\frac{D+1}{2} + 2)(N - 1)} = \frac{4}{D + 5} \quad \square$$

Note that $4/(D+5) \geq 2/(D+1)$ for all $D \geq 3$. So, the worst ratio over all d -interfaces, where $1 \leq d \leq D$, is $\frac{2}{(D+1)}$. Since the optimal number of regular simplices is bounded by the total number of simplices, we get the following theorem.

Theorem 13. For $D \geq 3$, Algorithm 2 provides a $2/(D+1)$ -factor approximation for the Max Morse Matching problem for manifolds without boundary.

We would like to make two remarks here regarding the approximation factor.

Firstly, the ratio is not affected by line 24 (first step) of procedure **intermediateApx()**. In particular, maximum-cardinality bipartite matching is not required to obtain the approximation ratio. The approximation ratio depends entirely on line 25

³ We do not actually perform topological sort on the d -interface, but need it for making an argument.

(second step) of `intermediateApx()`. Line 25 enables us to use the notion of canonical triplets to obtain the approximation ratio. We include a matching-based preprocessing step prior to applying the second step because, in practice, doing so gives significantly better results.

Secondly, the approximation ratio is over the *total number of simplices*. In that sense, Algorithm 2 and its analysis helps further our understanding of combinatorial construction of manifolds. In other words, irrespective of the complex size, the homology or the presence of non-collapsible elements, we can always collapse at least $2^{2/(D+1)}$ fraction of simplices in that manifold!

4.2. A 2^D -factor approximation algorithm for simplicial manifolds

Once again, we restrict our attention to simplicial manifolds without boundary. We build on Algorithm 2 by exploiting a finer substructure within each interface to obtain a further improvement in ratio for simplicial manifolds. We begin with some definitions.

Definition 9 (*Facet degree, min-facet simplex of the d -interface*). The number of facets incident on a simplex in the Hasse graph is defined as its *facet degree*. In particular, for the d -interface, consider the subset of d -simplices \mathcal{S} with at least one facet. We say that a d -simplex is a *min-facet simplex of the d -interface* if over all simplices in \mathcal{S} it has the minimum number of facets.

Note that in the course of Algorithm 3, the Hasse graph is modified in the sense that some simplices are deleted from the Hasse graph. All d -simplices have facet degree $(d + 1)$ in the original Hasse graph. Hence, the definition of facet degree is more pertinent to d -simplices in the modified Hasse graph.

Definition 10 (*Min-facet component of the d -interface*). A *min-facet component* is a subgraph of the d -interface that is a maximal connected graph induced by a set of min-facet simplices of the d -interface.

Like in Algorithm 2, we process the Hasse graph one d -interface at a time starting with the 1-interface and terminating with the D -interface. Also, like in Algorithm 2, we use optimal algorithms to process the 1-interface and the D -interface of the Hasse graph. Only the intermediate interfaces are processed differently. The procedure for handling intermediate interfaces is outlined in Algorithm 3.

Algorithm 3 The Min-facet component algorithm.

```

1: procedure INTERAPXMINFACET( $\mathcal{G}^d$ )
2:   while sizeOfMinFacet( $\mathcal{G}^d$ ) > 0 do
3:      $\mathcal{F}_C \leftarrow \text{extractMinFacetComponent}(\mathcal{G}^d)$ 
4:     Apply intermediateApx( $\mathcal{F}_C, d$ ) from Algorithm 2
5:   end while
6: end procedure

```

By design, procedure `intermediateApx()` from Algorithm 2 need not process the entire d -interface \mathcal{G}^d at one go. It may take any subgraph of the d -interface as its input. The key idea is to iteratively compute Morse matching by executing `intermediateApx()` on a min-facet component and after designing a vector field on this component, we subsequently delete it from the d -interface \mathcal{G}^d . As a consequence, \mathcal{G}^d grows increasingly sparse, and when the entire d -interface has no edges left, the while loop terminates. Fig. 5 illustrates sample executions of the Algorithm 3.

Lemma 14. *If the d -interface of the Hasse graph is connected, then the $(d - 1)$ -interface is connected.*

Proof. Suppose the d -interface of the Hasse graph is connected. If any two arbitrary $(d - 2)$ -simplices γ_s and γ_d can be shown to be connected, then the $(d - 1)$ -interface is connected. To begin with let α_0 be any $(d - 1)$ -simplex with γ_s as its facet and α_k be any $(d - 1)$ -simplex with γ_d as its facet. If $\alpha_0 = \alpha_k$, there is nothing to prove. So, for the remainder of the proof we shall assume that $\alpha_0 \neq \alpha_k$. Since the d -interface of the Hasse graph is connected, there exists a simple path lying entirely in the d -interface of the Hasse graph connecting any two $(d - 1)$ -simplices. In other words, there exists a simple path $\{\alpha_0, \beta_1, \alpha_1, \dots, \alpha_{(k-1)}, \beta_k, \alpha_k\}$ connecting α_0 and α_k where all $\beta_i, i \in [1, k]$ are d -simplices and all $\alpha_i, i \in [0, k]$ are $(d - 1)$ -simplices. Now, since every d -simplex $\beta_i, i \in [1, k]$ is common to two $(d - 1)$ -simplices $\alpha_{(i-1)}$ and α_i belonging to the simple path connecting α_0 and α_k , we know that $\alpha_{(i-1)}$ and α_i will share a facet, which we denote by $\gamma_i^{(d-2)}$. In other words, we construct a new simplicial sequence $\mathcal{S} = \{\alpha_0, \gamma_1, \alpha_1, \dots, \alpha_{(k-1)}, \gamma_k, \alpha_k\}$ from the simple path $\{\alpha_0, \beta_1, \alpha_1, \dots, \alpha_{(k-1)}, \beta_k, \alpha_k\}$, where $\gamma_i = \alpha_{(i-1)} \cap \alpha_i$. However, note that in this case, γ_i may possibly be equal to γ_j for some $i \neq j$. See Fig. 6 for an example. Without loss of generality, assume $\gamma_s \neq \gamma_1$ and $\gamma_d \neq \gamma_k$. We prove connectivity of γ_s and γ_d by induction. For the base case, we note that γ_s is connected to γ_1 since γ_1 and γ_s are facets of simplex α_0 . For the induction step, suppose γ_s is connected to γ_i . Now, consider the next two elements in sequence \mathcal{S} , namely α_i and

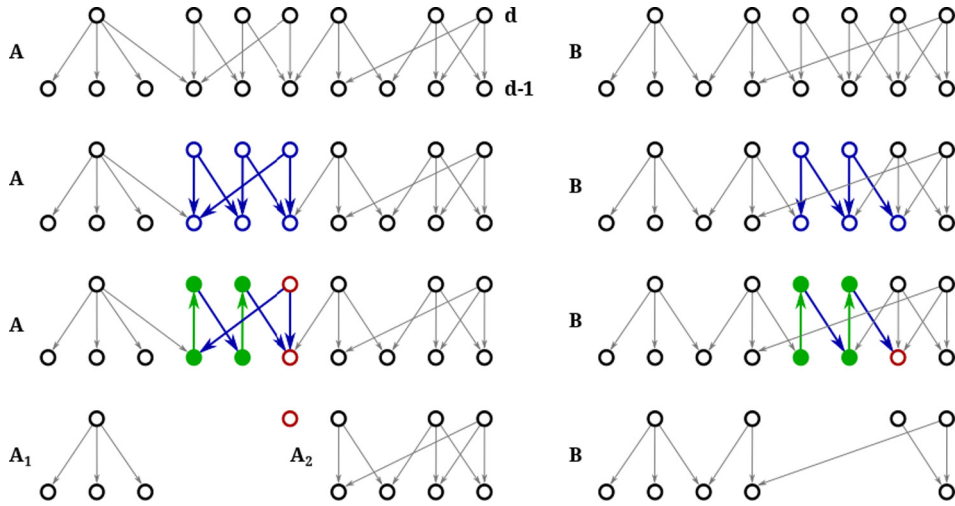


Fig. 5. Algorithm 3 processes the min-facet component in the d -interface (bold edges). Regular simplices are denoted by filled vertices. Critical simplices and unprocessed simplices are denoted by hollow vertices. Left: Deletion of $(d - 1)$ -simplices of a min-facet component disconnects the graph. Right: Deletion of $(d - 1)$ -simplices of the new min-facet component keeps the graph connected. This process continues until none of the d -simplices have any facets left.

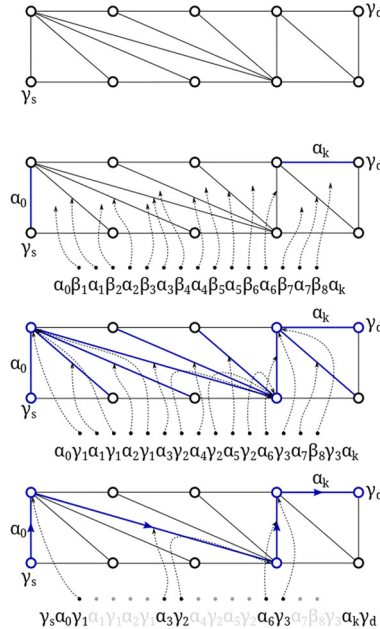


Fig. 6. In this figure, we wish to establish the connectivity of γ_s and γ_d in the 1-interface. Let α_0 and α_k be 1-simplices containing γ_s and γ_d , respectively. It is known that the 2-interface is connected. So, we can find the gradient sequence $\alpha_0 \beta_1 \alpha_1 \dots \beta_8 \alpha_k$. If we let $\gamma_i = \alpha_{i-1} \cap \alpha_i$, then we can extract a new sequence $\alpha_0 \gamma_1 \alpha_1 \dots \gamma_8 \alpha_k$. Finally, as explained in Lemma 14, this sequence can be used to obtain subsequence $\gamma_s \alpha_0 \gamma_1 \alpha_3 \gamma_2 \alpha_6 \gamma_3 \alpha_k \gamma_d$, which establishes connectivity between γ_s and γ_d .

$\gamma_{(i+1)}$. If $\gamma_i = \gamma_{(i+1)}$, then α_i makes no contribution towards finding a path connecting γ_s and γ_d and hence we ignore it. Else if $\gamma_i \neq \gamma_{(i+1)}$, then both γ_i and $\gamma_{(i+1)}$ are facets of α_i and hence γ_i is connected to $\gamma_{(i+1)}$ in the $(d - 1)$ interface. By transitivity, γ_s is connected to $\gamma_{(i+1)}$, which completes the induction step. Finally, both γ_k and γ_d are facets of α_k . Hence, γ_k is connected to γ_d . By transitivity γ_s is connected to γ_d . This proves that there exists a subgraph of the $(d - 1)$ -interface that connects any two arbitrary $(d - 2)$ -simplices γ_s and γ_d . Hence proved. \square

Lemma 15. For a connected D -manifold without boundary, all d -interfaces are connected.

Proof. Let K be the number of connected components of the D -interface. Then, $\beta_D = K$. Since a connected manifold without boundary has $\beta_D = 1$, we conclude that the D -interface is connected. Combining this fact with Lemma 14 implies that all d -interfaces are connected. \square

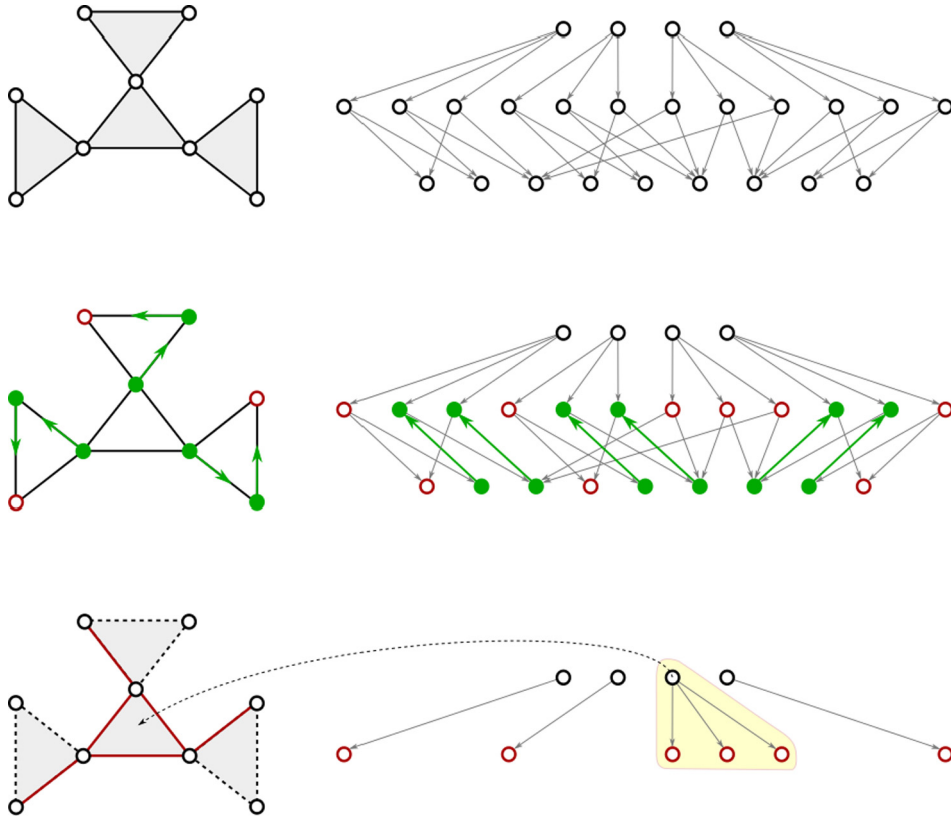


Fig. 7. Unlike in the case of manifolds without boundary, in this example we have a complex whose 2-interface is disconnected to begin with. After designing vector field for the 1-interface, suppose we delete all the matched 1-simplices from the Hasse graph. Then, there exists a connected component in the 2-interface for which all 2-simplices of that connected component has three facets. (In this case, this connected component comprises of a single simplex with all three solid edges.)

Lemma 16. *Following the design of gradient vector field for the $(d - 1)$ -interface, if the deletion of regular sinks of d -interface graph disconnects the d -interface, then every connected component has at least one simplex with facet degree smaller than $d + 1$.*

Proof. From Lemma 15 we know that the d -interface is a single connected component to begin with. Suppose that the regular sinks of d -interface graph are deleted in some sequence. Suppose that γ^{d-1} is the first simplex whose deletion disconnects the d -interface graph. Then, every connected component (in the traditional graph theory sense) has at least one d -simplex, which is incident on γ^{d-1} . Hence, upon deletion of γ^{d-1} , every connected component will have at least one simplex with facet degree smaller than $(d + 1)$. The same argument can be continued for subsequent deletions and resulting disconnections. \square

To see that Lemma 15 is essential for Lemma 16 to work, we see an example in Fig. 7 where lack of connectivity in the d -interface leads to components (in the d -interface) with minimum facet degree equal to $(d + 1)$.

Lemma 17. *For a d -interface, every min-facet component has facet degree bounded by d .*

Proof. We prove the claim by induction.

Base Case: From Lemma 16, having deleted all the regular sinks of the d -interface, there exists at least one simplex with facet degree bounded by d in every connected component. We arbitrarily choose a min-facet simplex in one of the connected components of the d -interface and discover the min-facet component around it by exploring neighboring d -simplices iteratively. Such a min-facet component has facet degree bounded by d . We design vector field on this min-facet component and subsequently delete it from the d -interface.

Induction Hypothesis: Suppose that we have designed a vector field on $(i - 1)$ min-facet components and subsequently deleted them. Each time, there exists at least one simplex with facet degree bounded by d in every connected component.

Induction Step: Now, we discover the i^{th} min-facet component, say \mathcal{F}_i . Suppose this min-facet component belongs to some connected component \mathcal{C}_j .

Case 1: \mathcal{F}_i contains all $(d - 1)$ -simplices in \mathcal{C}_j If \mathcal{F}_i consists of all the $(d - 1)$ -simplices of \mathcal{C}_j , then upon deletion of \mathcal{F}_i , all d -simplices of $\mathcal{C}_j \setminus \mathcal{F}_i$ will have zero facet degree and we will attempt to match the d -simplices of $\mathcal{C}_j \setminus \mathcal{F}_i$ for the $(d + 1)$ -interface. Also, upon deletion of \mathcal{F}_i , the other connected components continue to satisfy the facet degree (bounded by d) condition.

Case 2: $\mathcal{C}_j \setminus \mathcal{F}_i$ has at least one $(d - 1)$ -simplex Suppose if $\mathcal{F}_i \subsetneq \mathcal{C}_j$ and $\mathcal{C}_j \setminus \mathcal{F}_i$ has one or more $(d - 1)$ -simplices. Then, there exists at least one d -simplex, say σ in $\mathcal{C}_j \setminus \mathcal{F}_i$ with at least one edge incident on a $(d - 1)$ -simplex in \mathcal{F}_i and at least one edge incident on a $(d - 1)$ -simplex in $\mathcal{C}_j \setminus \mathcal{F}_i$. Having designed a gradient vector field on \mathcal{F}_i , we delete the regular simplices and the critical $(d - 1)$ simplices belonging to \mathcal{F}_i . Now, we consider two subcases that are illustrated in Fig. 5

Case 2a: \mathcal{C}_j stays connected after deleting \mathcal{F}_i Consider the case when \mathcal{C}_j stays connected after deleting the i th min-facet component. In this case, the facet degree of σ will reduce by at least 1, and hence, the facet degree of σ is bounded by d . There may be other simplices in $\mathcal{F}_i \subsetneq \mathcal{C}_j$ whose facet degree may also reduce. All other connected components are unaffected. So, every component will have min-facet degree bounded by d .

Case 2b: \mathcal{C}_j gets disconnected after deleting \mathcal{F}_i Now, consider the case where upon deletion of \mathcal{F}_i , \mathcal{C}_j splits into several components. Imagine that we are not deleting the simplices of \mathcal{F}_i all at once, but sequentially. Making an argument along the lines of Lemma 16, we conclude that irrespective of which connected component the min-facet component is chosen from, it will have facet degree bounded by d .

Therefore, in each of the cases, after deletion of the i th min-facet component \mathcal{F}_i , there exists at least one simplex with facet degree bounded by d in every connected component of the d -interface. \square

Lemma 18. An orientation of the min-facet component \mathcal{F}_C based on the matchings computed by procedure `interApxMinFacet()` is acyclic.

The proof of the Lemma 18 is identical to the proof of Lemma 9.

Lemma 19. An orientation of the d -interface of output graph \mathcal{H}_V based on the matchings computed by procedure `interApxMinFacet()` is acyclic.

Proof. We prove this claim by induction. We use a condition, namely the *vertex deletion criterion*, which says that: For the d -interface, a $(d - 1)$ -simplex satisfies the vertex deletion condition if and only if all paths that go through that simplex end up in a sink.

Base Case: Suppose that we are processing the first min-facet component for the d -interface. From Lemma 18, we know that an orientation of edges of a min-facet component is acyclic. For this orientation, a path from any vertex in the component ends up in a sink. Therefore, if we were to delete all the $(d - 1)$ -simplices in the min-facet component, we would obey the vertex deletion criterion. If graph \mathcal{H}_V is oriented based on the matchings found in the first min-facet component, then it is acyclic.

Induction Step: Suppose that we have processed i min-facet components and suppose that we have used these min-facet components to orient the d -interface of \mathcal{H}_V and so far it is found to be acyclic. Also, the vertices deleted so far are those that have satisfied the vertex deletion condition. Now, suppose we have extracted the $(i + 1)$ th min-facet component, say \mathcal{F}_{i+1} . While the edges that lead to sinks maybe absent in min-facet component \mathcal{F}_{i+1} , the corresponding d -simplices in output graph \mathcal{H}_V will have these edges. If we restrict our attention to undeleted edges, then from Lemma 18, the orientation of edges of $(i + 1)$ th min-facet component itself is acyclic, i.e., all paths will lead strictly to critical sinks of \mathcal{F}_{i+1} . But, if we look at the corresponding orientation in \mathcal{H}_V , the paths emanating from a $(d - 1)$ simplex of \mathcal{F}_{i+1} will either end up in critical sinks of \mathcal{F}_{i+1} (through undeleted edges) or in regular/critical sinks of \mathcal{F}_j for $j < (i + 1)$ (through deleted edges). In any case, all paths originating from $(d - 1)$ -simplices of \mathcal{F}_{i+1} go to sinks thereby satisfying the vertex deletion criterion. Also, designing gradient field on \mathcal{F}_j does not introduce cycles in \mathcal{H}_V . Morse matching on the d -interface is designed when all the min-facet components are processed and deleted. Since none of them introduce cycles, we say that output graph \mathcal{H}_V is acyclic. \square

Lemma 20. For the d -interface, the ratio $\frac{\#matched\ simplices}{\#total\ simplices} \geq \frac{2}{d+1}$.

Proof. The proof is identical to that of Lemma 11 except for one important difference. In the case of Algorithm 2, for a d -interface every d -simplex has $d + 1$ facets. But according to Lemma 17, for a min-facet component, the facet degree is bounded by d . Using the notion of canonical triplets for a min-facet component, for every gradient pair, we get at most $(d - 1)$ critical simplices. So, the ratio $\frac{\#matched\ simplices}{\#total\ simplices} \geq \frac{2}{d+1}$ for every min-facet component. For a d -interface, every $(d - 1)$ -simplex is part of some min-facet component and is classified as a regular simplex or as a critical simplex

and subsequently deleted from the d -interface. Therefore, the bound of $2/(d+1)$ carries over from min-facet components to d -interfaces. \square

If we take the minimum for the ratio $2/d+1$ over all d , such that $1 < d < D$, we get $2/D$. By Lemma 12, for the D -interface, the ratio $\frac{\# \text{matched simplices}}{\# \text{total simplices}}$ is equal to $\frac{4}{D+5}$. Note that $4/(D+5) \geq 2/D$, for all $D \geq 5$. So, for $D \geq 5$, the worst ratio of $\frac{\# \text{matched simplices}}{\# \text{total simplices}}$ over all d -interfaces where $1 \leq d \leq D$ is $\frac{2}{D}$. Since the optimal number of regular simplices \leq total number of simplices, we get the following theorem.

Theorem 21. For $D \geq 5$, Algorithm 3 provides a $2/D$ -factor approximation for the Max Morse Matching problem.

On the other hand, for 3-manifolds and 4-manifolds, we have $4/(D+5) < 2/D$. So, for 3-manifolds, the approximation ratio is $4/(3+5) = 1/2$. Similarly, for 4-manifolds, the approximation ratio is $4/(4+5) = 4/9$. This gives us the following result.

Theorem 22. For $D = 3$, Algorithm 3 provides a $1/2$ -factor approximation for the Max Morse Matching problem. For $D = 4$, we obtain a $4/9$ -factor approximation algorithm.

4.2.1. Approximation bound for non-manifold complexes

Note that Algorithm 3 can be applied to non-manifold complexes as well if we apply the optimal algorithm for the 1-interface and procedure `intermediateApX()` for the remaining interfaces. To prove a bound for non-manifold complexes, we need to do a slightly different kind of analysis. We begin with a few definitions. Let T denote the set of all $(D - 1)$ -simplices of the Hasse graph. Let B denote the $(D - 1)$ -simplices that have been paired with $(D - 2)$ -simplices and let $|A| = |T| - |B|$. Let \mathcal{R}_D denote the set of regular simplices found by Algorithm 3 at the D -interface. We now establish a relation between $|\mathcal{R}_D|$ and $|A|$.

Lemma 23. $|\mathcal{R}_D| \geq \frac{2}{r}|A|$ where $r = D$ if the D -interface is connected and $r = (D + 1)$ if the D -interface is not connected.⁴

Proof. First we consider the case when the D -interface is not connected at the start of Algorithm 3. At each stage of Algorithm 3, the minimum facet-degree of a simplex is not more than $(D + 1)$. Once again, we use the idea of canonical triplets. Every critical $(D - 1)$ -simplex occurs in a unique canonical triplet. Also, every regular $(D - 1)$ -simplex occurs in at most D canonical triplets. So, every regular $(D - 1)$ -simplex corresponds to a set of at most D critical $(D - 1)$ -simplices. Together they make up the entire set A . Hence we have $|\mathcal{R}_D| \geq \frac{2}{D+1}|A|$.

Now suppose the D -interface is connected at the start of the Algorithm. Then, Lemma 16 and Lemma 17 apply and the minimum facet-degree of a simplex (in a min-facet component) is not more than D . In this case, a regular $(D - 1)$ -simplex occurs in at most $(D - 1)$ canonical triplets. Accordingly, $|\mathcal{R}_D| \geq \frac{2}{D}|A|$. \square

Now, let \mathcal{R} denote the set of all regular simplices found by Algorithm 3 and let $\mathcal{R}_L = \mathcal{R} - \mathcal{R}_D$.

Let \mathcal{S}_{D-2} denote the set of vertices of the Hasse graph that belong to one of the d -interfaces where $1 \leq d \leq (D - 2)$ and \mathcal{S}_{D-1} denote the set of vertices of the Hasse graph that belong to one of the d -interfaces where $1 \leq d \leq (D - 1)$. Let $\mathcal{S} = \mathcal{S}_{D-2} \cup \mathcal{S}_{D-1}$. Finally, let $|\mathcal{S}|$ denote the cardinality of vertex set \mathcal{S} and $|\mathcal{S}_{D-1}|$ denote the cardinality of vertex set \mathcal{S}_{D-1} .

Lemma 24. $|\mathcal{R}_L| \geq \frac{2}{D}|\mathcal{S}|$.

Proof. Let $\mathcal{G}_\mathcal{S}$ be the graph induced by set \mathcal{S} . Note that every simplex belonging to graph $\mathcal{G}_\mathcal{S}$ occurs in some canonical triplet. In particular this happens to be true since all $(D - 1)$ -simplices of \mathcal{S} are matched by Algorithm 3. Using Lemma 20 for Algorithm 3 applied to $\mathcal{G}_\mathcal{S}$, the ratio $\frac{\# \text{matched simplices}}{\# \text{total simplices}} \geq \frac{2}{D}$. In other words, we get, $|\mathcal{R}_L| \geq \frac{2}{D}|\mathcal{S}|$. \square

Let \mathcal{O} denote the cardinality of regular nodes found by optimal Morse Matching.

Lemma 25. $\mathcal{O} \leq |\mathcal{S}_{D-1}| + |T|$.

Proof. The maximum number of simplices of the D -level that can be matched by any algorithm is bounded by $|T|$, i.e., the total number of simplices of the $(D - 1)$ -level. Also, the set \mathcal{S}_{D-1} consists of all simplices of the Hasse graph except those that belong to the D -level. So, the optimal algorithm cannot possibly match more than $|\mathcal{S}_{D-1}| + |T|$ number of simplices of the Hasse graph. \square

We now consider the expression $D|\mathcal{R}_L| + r|\mathcal{R}_D|$,

⁴ Note that the D -interface of a general, non-manifold simplicial complex may or may not be connected.

$$\begin{aligned}
D|\mathcal{R}_L| + r|\mathcal{R}_D| &\geq 2|\mathcal{S}| + 2|A| && \text{using Lemma 23 and Lemma 24} \\
&\geq |\mathcal{S}| + |B| + 2|A| && \text{using the fact that } B \subseteq S \\
&\geq (|\mathcal{S}| + |A|) + (|B| + |A|) \\
&= |\mathcal{S}_{D-1}| + |T| && \text{by definition} \\
&\geq \mathcal{O} && \text{using Lemma 25}
\end{aligned}$$

If the D -interface is connected we get,

$$D|\mathcal{R}| = D|\mathcal{R}_L| + D|\mathcal{R}_D| \geq \mathcal{O}, \text{ i.e., } |\mathcal{R}| \geq \frac{1}{D}\mathcal{O}$$

If the D -interface is not connected we get,

$$(D+1)\mathcal{R} = (D+1)|\mathcal{R}_L| + (D+1)|\mathcal{R}_D| \geq D|\mathcal{R}_L| + (D+1)|\mathcal{R}_D| \geq \mathcal{O}, \text{ i.e., } |\mathcal{R}| \geq \frac{1}{(D+1)}\mathcal{O}.$$

Therefore, for non-manifold complexes, Algorithm 3 gives a $1/D$ approximation if the D -interface is connected and a $1/(D+1)$ approximation if the D -interface is not connected.

Likewise, one can obtain $1/(D+1)$ approximation bound for Algorithm 2 irrespective of whether or not the complex is connected.

5. Experimental results

All three approximation algorithms proposed in this paper are implemented in Java. In this section, we describe results of experiments comparing six different algorithms for Morse matching. In particular, we compare the algorithms proposed in this paper with the reduction heuristic, the coreduction heuristic and a naïve approximation algorithm described in Section 5.1. A prototype implementation was used to observe the practical performance of these algorithms on more than 800 complexes. We used both synthetic random datasets and complexes generated by Hachimori [23] (also used in an earlier work [20]) and Lutz [24], for experiments. Random complexes were generated according to the method described by Meshulam and Wallach [25] and a variant. In the variant, we select a random number of valid d -simplices for all $1 \leq d \leq D$ instead of selecting a random number of D -simplices and inferring all faces. We refer to the complexes generated by this variant as Type 2 random complexes. For additional details on these complexes see Section 5.6.

It is clear that the quantity $2|\mathcal{M}|$, where $|\mathcal{M}|$ is the size of maximum-cardinality matching as well as the quantity $N - \sum \beta_i$, which is equal to the difference between number of simplices and the sum of Betti numbers provide conservative upper bounds on the number of regular cells in the optimal Morse matching. Let \mathcal{R} be the set of regular simplices generated by a Max Morse approximation algorithm. We estimate the quality of the approximation using the ratio $\frac{|\mathcal{R}|}{\min(2|\mathcal{M}|, N - \sum \beta_i)}$. Tables 1, 2, 4 and 5 list estimated approximation ratios on selected datasets. Algorithm 3 consistently provided the best ratios, always greater than 0.93 for all 300 random complexes in our dataset. For more than 450 manifolds from the Lutz dataset, Algorithm 3 reported worst estimated approximation ratio of 0.969. Algorithm 3 provided optimal estimated approximation ratio for 56% of manifolds from the Lutz dataset. These results suggest that Algorithm 3 not only provides good theoretical bounds, but also performs well practically.

In sections that follow, we first discuss a naïve approximation algorithm and the reduction and coreduction heuristics followed by experiments on datasets from four different sources.

5.1. A $1/(D+1)$ -factor naïve approximation algorithm

Consider the following approximation algorithm: Given a simplicial complex \mathcal{K} compute its Hasse graph $\mathcal{H}_{\mathcal{K}}$. Perform maximum-cardinality matching on graph $\mathcal{H}_{\mathcal{K}}$ and obtain the matching-based orientation $\overline{\mathcal{H}_{\mathcal{K}}}$. Include all the down-edges of $\overline{\mathcal{H}_{\mathcal{K}}}$ in the output graph $\mathcal{H}_{\mathcal{O}}$.

1. Pick an arbitrary up-edge e and include it in $\mathcal{H}_{\mathcal{O}}$.
2. Include the reversed orientations of all the leading up-edges of e in $\mathcal{H}_{\mathcal{O}}$.
3. Remove up-edge e and the leading up-edges of e from $\overline{\mathcal{H}_{\mathcal{K}}}$.
4. Repeat steps 1–3 until all up-edges of $\overline{\mathcal{H}_{\mathcal{K}}}$ are exhausted.

Clearly, $\mathcal{H}_{\mathcal{O}}$ has no cycles because none of the up-edges in $\mathcal{H}_{\mathcal{O}}$ has leading up-edges. Also, for every up-edge that we select, we reverse at most D up-edges. Since the cardinality of the maximum matching is an upper bound on optimal value of Max Morse Matching, we get an approximation ratio of $1/(D+1)$ for this algorithm.

At the outset, the ratio $(D+1)/(D^2+D+1)$ obtained by Algorithm 1 does not seem to be a significant improvement over $1/(D+1)$. However, as we shall witness in sections that follow, the estimated approximation ratios observed for the naïve algorithm are significantly worse in practice. In fact, in order to ensure that the approximation algorithms designed for Max Morse Matching problem remains relevant for applications like homology computation, scalar field topology etc., we need to design algorithms that can be shown to have good theoretical approximation and complexity bounds combined with competitive estimated approximation ratios.

5.2. Coreduction and reduction heuristics

The coreduction heuristic for constructing Morse matchings was introduced by Harker et al. [7]. In this section, we briefly describe reduction and coreduction heuristics for constructing Morse matchings on simplicial complexes for the sake of completeness. Suppose we are given a simplicial complex \mathcal{K} . We first describe the coreduction heuristic.

Perform the following steps until complex \mathcal{K} is empty:

1. If there is a simplex α with a free cofacet β available, include the pair (α, β) in the set of Morse matchings, and delete α and β from \mathcal{K} .
2. If no such simplex with a free cofacet is available, then select a simplex γ of lowest available dimension, make it critical, and delete it from \mathcal{K} .

Recall that in Section 4.2, Definition 9, we had introduced the idea of min-facet simplex of the d -interface. Similarly, one may extend the definition of min-facet simplex to the entire Hasse graph as follows:

Definition 11 (*Min-facet simplex of the Hasse graph*). A min-facet simplex of the Hasse graph is a simplex with the minimum number of facets in the Hasse graph.

It is worth noting that, for simplicial D -manifolds, if we modify step 2 of the coreduction algorithm to choose a min-facet simplex of the Hasse graph whenever a simplex with a free cofacet is not available, then we can use the same techniques as in Section 4.2 to establish a $2/D$ approximation ratio for $D \geq 5$ using the modified coreduction algorithm. Next, we describe the reduction heuristic.

Perform the following steps until complex \mathcal{K} is empty:

1. If there is a simplex β with a free facet α available, include the pair (α, β) in the set of Morse matchings, and delete α and β from \mathcal{K} .
2. If no such simplex with a free facet is available, then select a simplex γ of highest available dimension, make it critical, and delete it from \mathcal{K} .

5.3. Experiments on the Hachimori dataset

This dataset consists of complexes downloaded from Hachimori's collection of simplicial complexes.⁵ Table 1 lists the observed approximation ratios for all the algorithms. For complexes in Table 1, maximum size of $\Sigma\beta_i$ is 2. The coreduction heuristic provided the best approximation ratios for this dataset. However, Algorithm 3 reported ratios comparable to the coreduction algorithm. Algorithm 3 reports optimal Morse matching for 7 of the 20 complexes in this dataset, while coreduction gives optimal result for 10 complexes.

5.4. Experiments on the Lutz manifold dataset

The second dataset consists of manifolds of dimensions ranging from 3 to 11. These manifolds were downloaded from a library of manifolds created by Lutz.⁶

Table 2 lists approximation ratios observed for selected complexes within this dataset. For manifolds in Table 2, maximum size of $\Sigma\beta_i$ is 14 whereas the average size of $\Sigma\beta_i$ is 5.07. Coreduction heuristic provided the best approximation ratios for this dataset. However, Algorithm 3 matched the performance of coreduction heuristic for many complexes and in some cases outperformed coreduction. Also, Algorithm 3 was consistently better than reduction heuristic.

Table 3 summarizes the results obtained using Algorithm 3 for manifolds of different dimensions. We observed optimal results for 56% of the complexes. The worst approximation ratio was observed to be 0.969. The descriptions of homology groups of these complexes are also available in the library. For all the complexes in this dataset, we compute the homology by application of Morse matching algorithm, followed by boundary operator computation and subsequently applying the Smith Normal Form. The running time of homology computation was of the order of milli-seconds for most of these complexes.

5.5. Experiments on random complexes

We followed the method described by Meshulam and Wallach [25] to generate random complexes. These complexes contain all possible d -simplices for the given number of vertices, for $0 \leq d < D$. However, D -simplices are randomly chosen from all possible D -simplices based on probability $p(D)$. We generated two datasets of 100 complexes each. For each set,

⁵ http://infoshako.sk.tsukuba.ac.jp/~hachi/math/library/index_eng.html.

⁶ The Manifold page: <http://page.math.tu-berlin.de/~lutz/stellar/vertex-transitive-triangulations.html>.

Table 1

Observed approximation ratios for Hachimori's Simplicial Complex Library. N indicates the number of simplices in the complex. Cored refers to Coreduction Algorithm, Red refers to Reduction Algorithm. For a given input, the best estimated approximation ratios across all algorithms tested are highlighted in bold.

Input	N	Estimated approximation ratios					
		Naïve	Algo 1	Algo 2	Algo 3	Cored	Red
<i>2D complexes</i>							
projective	31	0.800	0.933	0.933	0.933	0.933	0.933
dunce_hat	49	0.667	0.917	0.958	0.958	0.958	0.917
bjorner	32	0.667	0.933	1.000	1.000	1.000	0.867
nonextend	39	0.632	0.895	0.947	1.000	1.000	0.895
c-ns	75	0.703	0.892	0.946	0.946	0.946	0.865
c-ns2	79	0.615	0.897	0.974	0.974	1.000	0.846
c-ns3	63	0.667	0.871	0.968	0.968	0.968	0.903
simon	41	0.750	0.950	0.950	0.950	0.950	0.850
simon2	31	0.667	0.800	0.933	0.933	0.933	0.867
<i>3D complexes</i>							
poincare	392	0.651	0.933	0.954	0.979	0.990	0.923
knot	6,203	0.628	0.942	0.940	0.997	1.000	0.927
bing	8,131	0.640	0.946	0.943	0.997	0.999	0.933
nc_sphere	8,474	0.616	0.941	0.945	0.989	1.000	0.937
rudin	215	0.617	0.935	0.944	1.000	1.000	0.925
gruenbaum	167	0.663	0.928	0.928	1.000	1.000	0.904
ziegler	119	0.695	0.983	0.915	1.000	1.000	0.864
lockeberg	216	0.636	0.944	0.972	1.000	1.000	0.897
mani-walkup-C	464	0.645	0.944	0.922	1.000	1.000	0.922
mani-walkup-D	392	0.621	0.923	0.923	0.990	0.990	0.908
<i>5D complexes</i>							
nonpL_sphere	2,680	0.554	0.841	0.883	0.989	0.997	0.954

Table 2

Observed approximation ratios for a few selected manifolds in Lutz's manifold library. N indicates the number of simplices in the complex. Cored refers to Coreduction Algorithm, Red refers to Reduction Algorithm. For a given input, the best estimated approximation ratios across all algorithms tested are highlighted in bold.

Input	N	Estimated approximation ratios					
		Naïve	Algo 1	Algo 2	Algo 3	Cored	Red
3_12_13_3	192	0.649	0.936	0.957	1.000	1.000	0.926
3_12_1_6	240	0.672	0.933	0.908	0.992	0.992	0.882
3_15_11_1	390	0.649	0.948	0.974	0.984	1.000	0.953
4_15_2_24	810	0.610	0.898	0.911	1.000	1.000	0.935
4_15_4_1	965	0.566	0.875	0.902	0.987	0.996	0.919
5_15_2_12	1,350	0.565	0.862	0.896	0.990	0.999	0.951
5_14_3_16	1,120	0.572	0.873	0.898	0.998	1.000	0.959
6_15_2_2	5,130	0.516	0.801	0.841	0.987	0.998	0.961
6_15_2_1	1,890	0.546	0.847	0.877	0.995	1.000	0.957
7_14_3_4	6,272	0.499	0.768	0.820	1.000	0.999	0.955
8_14_2_15	9,326	0.479	0.747	0.782	1.000	1.000	0.962
9_15_4_1	21,310	0.458	0.716	0.757	0.996	1.000	0.961
10_14_38_1	15,038	0.460	0.716	0.754	1.000	1.000	0.960
11_15_2_1	30,846	0.443	0.688	0.737	1.000	1.000	0.961

we generated a subset of 20 complexes with fixed $p(D)$, which varies from 0.1 to 0.9. The number of vertices was chosen to be 20 and 16 for the 4 and 6 dimensional datasets, respectively. In Table 4, we report the results for a single complex selected from each subset. It should be noted that Algorithm 3 performs well even for random complexes with non-trivial homology. For Algorithm 3, the worst estimated approximation ratio over 100 randomly generated 4-dimensional complexes was observed to be 0.939. For the 100 randomly generated 6-dimensional complexes it was observed to be 0.953. We observed that Algorithm 3 outperformed reduction and coreduction heuristics for this dataset.

5.6. Experiments on Type 2 random complexes

We also used a variant of the method described by Meshulam and Wallach [25] for generation of random complexes, where we choose random number of d -simplices for all d . The generation of these random complexes proceed from lowest dimension to highest, and a random simplex is added to the complex only if all its facets are part of the complex. We generated a dataset containing 100 5-dimensional complexes with following parameters: number of vertices was chosen as 40, the probability of selecting a d -simplex is given by the vector [1, 1, 0.7, 0.9, 1, 0.9]. With these parameters we obtain complexes

Table 3

In this table, we summarize the results for all the 446 complexes in Lutz manifold dataset. The results are grouped row-wise for each dimension. Avg size refers to the average number of simplices for complexes of dimension D . Worst ratio refers to worst estimated approximation ratio for Algorithm 3 whereas Avg ratio refers to average estimated approximation ratio for Algorithm 3, for complexes of dimension D . % optimal refers to percentage of complexes for which Algorithm 3 computes the optimal Morse matching.

D	No. of complexes	Avg size	Worst ratio	Avg ratio	% optimal
3	166	265	0.969	0.992	42.17
4	76	630	0.979	0.995	39.47
5	114	1,445	0.982	0.998	75.43
6	15	3,761	0.984	0.993	26.67
7	33	5,988	0.996	0.999	87.88
8	26	9,165	0.989	0.998	69.23
9	9	14,385	0.993	0.999	66.67
10	2	9,566	1.000	1.000	100.00
11	5	23,096	1.000	1.000	100.00
All	446	2,271	0.969	0.995	56.05

Table 4

This table lists estimated approximation ratios for a selected set of random complexes of dimensions 6 and 4. Each row represents results obtained by various algorithms for a single randomly generated instance. N indicates the number of simplices in the complex. Cored refers to Coreduction Algorithm, Red refers to Reduction Algorithm. $p(D)$ denotes the probability with which simplices of dimension D are chosen. $\Sigma\beta_i$ denotes the sum of Betti numbers. For a given input, the best estimated approximation ratios across all algorithms tested are highlighted in bold.

$p(D)$	N	$\Sigma\beta_i$	Estimated approximation ratios					
			Naïve	Algo 1	Algo 2	Algo 3	Cored	Red
<i>Random 6D</i>								
0.1	16,036	3,862	0.515	0.746	0.784	1.000	1.000	0.958
0.3	18,324	1,574	0.477	0.739	0.765	0.996	0.989	0.967
0.5	20,612	740	0.437	0.672	0.704	0.964	0.940	0.936
0.7	22,899	3,003	0.428	0.653	0.705	0.991	0.981	0.951
0.9	25,188	5,292	0.425	0.634	0.710	1.000	0.998	0.953
<i>Random 4D</i>								
0.1	7,745	2,327	0.598	0.900	0.914	0.999	0.996	0.979
0.3	10,846	800	0.470	0.704	0.727	0.952	0.915	0.927
0.5	13,947	3,877	0.465	0.692	0.732	0.992	0.978	0.956
0.7	17,047	6,977	0.452	0.669	0.739	1.000	0.996	0.954
0.9	20,148	10,078	0.463	0.671	0.752	1.000	1.000	0.957

Table 5

This table lists estimated approximation ratios for a selected set of 5-dimensional Type 2 random complexes. Each row represents results obtained by various algorithms for a single randomly generated instance. N indicates the number of simplices in the complex. Cored refers to Coreduction Algorithm, Red refers to Reduction Algorithm. $\Sigma\beta_i$ denotes the sum of Betti numbers. For a given input, the best estimated approximation ratios across all algorithms tested are highlighted in bold.

N	$\Sigma\beta_i$	Estimated approximation ratios					
		Naïve	Algo 1	Algo 2	Algo 3	Cored	Red
39,046	3,366	0.540	0.814	0.841	0.991	0.987	0.979
39,233	3,247	0.538	0.809	0.844	0.993	0.982	0.977
39,199	3,253	0.535	0.808	0.835	0.992	0.984	0.978
39,128	3,314	0.538	0.815	0.838	0.994	0.986	0.979
39,526	3,172	0.538	0.809	0.837	0.991	0.985	0.979

with non-trivial homology, as evidenced by their Betti numbers that lie in the range [1, 0, 0–1, 2945–3658, 51–106, 0–3]. Table 5 lists the results for five complexes selected from this dataset. The worst estimated approximation ratio over 100 randomly generated 5-dimensional complexes for Algorithm 3 was observed to be 0.989. We again observed that Algorithm 3 consistently outperformed reduction and coreduction heuristics for all the complexes in this dataset.

5.7. Discussion on experimental results

For all datasets we studied, Algorithm 3 and Coreduction Algorithm outperform all other algorithms in terms of achieving best estimated approximation ratios. For the Hachimori dataset and the Lutz dataset, the coreduction algorithm fares slightly

better, whereas for random datasets, Algorithm 3 does better. In general, Algorithm 3 outperforms all other algorithms for large sized complexes or when the size of $\Sigma\beta_i$ is large.

6. Discussion on complexity

Maximum-cardinality bipartite matching is the primary bottleneck for all the algorithms described in this paper. Graph matching can be performed in $O(V^{1.5})$ time for Hasse graphs of simplicial complexes using Hopcroft–Karp algorithm [26]. With appropriate choice of data structures, all other procedures of all three Algorithms can be made to run in linear time.

In particular, for Algorithm 3, we maintain separate queues for every facet-degree. Consider the graph \mathcal{G} induced by the min-facet degree simplices. To extract a min-facet component, we simply find a single connected component within this graph \mathcal{G} . Once the min-facet component is deleted from the d -interface, we update the facet-degrees of all affected simplices within the d -interface. Therefore, extraction and maintenance of min-facet components is a linear time operation.

Also, for Algorithms 2 and 3, the approximation ratios do not depend on the graph matching steps. Graph matching step merely serves the purpose of heuristic improvement. So, effectively by removing graph matching step from Algorithms 2 and 3 become linear time approximation algorithms. But this improvement in computational complexity is at the cost of estimated approximation ratios observed in practice.

7. Conclusions and further work

We believe that approximation algorithms is the definitive algorithmic way to study Morse matchings. Our belief is validated by theoretical results and additionally supported by experimental results where we get close to optimal ratios even for random complexes.

In future, we plan to further improve the approximation bounds, remove dependency on graph matching (for improving estimated approximation ratios) and develop efficient C++ implementations. In particular, to obtain dimension independent bounds for Max Morse Matching Problem remains a challenging open problem.

Acknowledgements

We are grateful to anonymous referees for their constructive input. The first author would like to thank Michael Lesnick for helpful discussions.

Appendix A. Optimal algorithms for 1-interface and D -interface

To begin with, we reproduce the definitions of the d -level of the Hasse graph and the d -interface of the Hasse graph of a simplicial complex from Section 2.2. We also reproduce the definition of dual graph of a simplicial manifold from Section 4.1.

We refer to the set of vertices in the Hasse graph representing d -dimensional simplices as the d -level of the Hasse graph. The d -interface of the Hasse graph is the subgraph consisting of vertices in the d -level and the $(d - 1)$ -level of the Hasse graph together with all the edges connecting these two levels.

Definition 12 (*Dual graph*). The dual graph of a simplicial D -dimensional manifold \mathcal{K} is the graph whose vertices represent the D -simplices of \mathcal{K} and whose edges join two D -simplices with a common $(D - 1)$ -facet.

Consider a Morse matching M for a connected D -manifold \mathcal{K} . Let $\gamma(M)$ be the graph obtained from the primal graph of \mathcal{K} by removing all arcs (edges of \mathcal{K}) matched with triangles in M . Let $\gamma^*(M)$ be obtained from the dual graph of \mathcal{K} by removing all the arcs ($(D - 1)$ -simplices of \mathcal{K}), where the corresponding $(D - 1)$ -simplices are matched with $(D - 2)$ -simplices of \mathcal{K} in M . Note that $\gamma(M)$ and $\gamma^*(M)$ contain all vertices and D -simplices of \mathcal{K} , respectively. Let $\mathcal{H}(M)$ denote the matching-based orientation of the Hasse graph \mathcal{H} . Also, let \mathcal{H}^* denote the dual of Hasse graph \mathcal{H} . In particular, a vertex belonging to the d -level of the graph $\mathcal{H}^*(M)$ corresponds to an $(n - d)$ -simplex of $\mathcal{H}(M)$. An edge joining a vertex at the d -level of \mathcal{H}^* to a vertex at the $(d - 1)$ -level of \mathcal{H}^* corresponds to an incidence relation between an $(n - d)$ -simplex and an $(n - (d - 1))$ -simplex in \mathcal{K} . In other words, \mathcal{H}^* is simply \mathcal{H} turned upside down. Similarly, $\mathcal{H}^*(M)$ denotes matching based orientation of the graph \mathcal{H}^* .

The existence of polynomial time algorithms for designing optimal Morse matchings for the D -interface is essential to establishing an approximation bound for Algorithm 2 and Algorithm 3. The optimal algorithm for the D -interface relies on Lemma 29, which has previously been proved using graph theoretic methods [1,20]. Although Lemma 6 of Appendix A in [20] is stated and proved specifically for 3-manifolds, a similar argument can be used for D -manifolds, for any $D \geq 3$ since only the 1-interface and the 2-interface of the dual Hasse graph are germane to establishing the connectivity of the D -interface.

Lemma 26. *The graph $\gamma(M)$ is connected.*

Proof. Suppose that $\gamma(M)$ is disconnected. Let N be the set of nodes in a connected component of $\gamma(M)$, and let C be the set of cut edges, that is, edges of \mathcal{K} with one vertex in N and one vertex in its complement. Since \mathcal{K} is connected, C is not empty. By definition of $\gamma(M)$, each edge in C is matched to a unique 2-simplex. Consider the directed subgraph D of the Hasse diagram consisting of the edges in C and their matching 2-simplices. The standard direction of arcs in the Hasse diagram (from the higher to the lower dimensional simplexes) is reversed for each matching pair of M , i.e., D is a subgraph of $\mathcal{H}(M)$. We construct a directed path in D as follows. Start with any node of D corresponding to a cut edge e_1 . Go to the node of D determined by the unique 2-simplex τ_1 to which e_1 is matched to. Then, τ_1 contains at least one other cut edge e_2 , otherwise e_1 cannot be a cut edge. Now iteratively go to e_2 , then to its unique matching 2-simplex τ_2 , choose another cut edge e_3 , and so on. We observe that we obtain a directed path $e_1, \tau_1, e_2, \tau_2, \dots$ in D , i.e., the arcs are directed in the correct direction. Since we have a finite graph at some point the path must arrive at a node of D , which we have visited already. So, D (and therefore also $\mathcal{H}(M)$) contains a directed cycle, which is a contradiction since M is a Morse matching. Hence, $\gamma(M)$ is connected. \square

Remark 2. The connectivity of $\gamma(M)$ is not directly interesting to us. However, the proof of connectivity of $\gamma(M)$ serves as a template for proving the connectivity of $\gamma^*(M)$. To prove the connectivity of $\gamma^*(M)$, in Appendix A of [20], Burton et al. suggest the use of Poincaré duality without providing additional details. It is unclear how to use Poincaré duality directly because dual cell structures do not always exist for all manifolds. This problem can be remedied by observing that the 0-level, the 1-level and the 2-level of \mathcal{H}^* along with the interfaces connecting them always form the face poset of a regular 2-dimensional cell complex. We shall sketch the proof of connectivity of $\gamma^*(M)$.

In order to construct a 2-dimensional cell complex, we must first construct an underlying 1-dimensional cell complex. To do this, we associate a 0-cell σ^* to every D -simplex $\sigma \in \mathcal{K}$ and a 1-cell τ^* to every $(D - 1)$ -simplex $\tau \in \mathcal{K}$. The set of all such 0-cells and 1-cells is precisely the dual graph of complex \mathcal{K} .

For a given $(D - 2)$ -simplex κ , consider its *boundary sequence* defined as the sequence of D -simplices and $(D - 1)$ -simplices $\{\sigma_0^D, \tau_0^{(D-1)}, \sigma_1^D, \tau_1^{(D-1)}, \dots, \sigma_{(q-1)}^D, \tau_{(q-1)}^{(D-1)}, \sigma_q^D\}$ such that for all i , $\sigma_i \neq \sigma_{(i+1)}$, $\tau_i < \sigma_i$, $\tau_{(i+1)} < \sigma_i$ and $\kappa < \tau_i$. Also, σ_q^D is identical to σ_0^D , where q is the smallest index for which a repetition of D -simplices occurs in the simplicial sequence. This is a unique sequence for a given κ and σ_0^D because each $(D - 1)$ -simplex is a facet of exactly two D -simplices and each σ_i^D has exactly two facets $\tau_{((i-1) \bmod q)}$ and $\tau_{(i \bmod q)}$ incident on κ . The boundary sequence of κ has a corresponding sequence of 0-cells and 1-cells in the dual graph of \mathcal{K} , called the *dual boundary sequence*.

Lemma 27. *Let \mathcal{K} be a orientable simplicial manifold without boundary. Let \mathcal{H} denote the Hasse graph of \mathcal{K} and \mathcal{H}^* denote the dual of the Hasse graph \mathcal{H} . Then, the 0-level, the 1-level and the 2-level of \mathcal{H}^* along with the interfaces connecting them form the face poset of a regular 2-cell complex.*

Proof (Sketch). By definition, the 0-level of \mathcal{H}^* represents the D -simplices while the 1-level of \mathcal{H}^* represents the $(D - 1)$ -simplices of \mathcal{K} . Clearly, the 0-level and the 1-level along with the 1-interface of \mathcal{H}^* forms the Hasse graph of the dual graph of complex \mathcal{K} . Hence, the 0-level and the 1-level along with the 1-interface of \mathcal{H}^* together form the face poset of a 1-complex. Now, for every $(D - 2)$ -simplex in \mathcal{K} , we have a vertex belonging to the 2-level of \mathcal{H}^* . Finally, to construct a 2-dimensional cell complex, we associate a dual 2-cell κ^* to every $(D - 2)$ -simplex κ and identify the boundary of κ^* with the dual boundary sequence of κ . It is easy to check that this 2-dimensional cell complex is regular. Hence, the 0-level, the 1-level and the 2-level of \mathcal{H}^* along with the interfaces connecting them form the face poset of a regular 2-dimensional cell complex. \square

We shall denote the cell complex whose face poset corresponds to the 0-level, the 1-level and the 2-level of \mathcal{H}^* along with the interfaces connecting them as \mathcal{K}_2^* and the face poset of \mathcal{K}_2^* as \mathcal{H}_2^* . \mathcal{H}_2^* is a subgraph of \mathcal{H}^* . $\mathcal{H}_2^*(M)$ is the matching based orientation of \mathcal{H}_2^* . For every directed path $\{\kappa_0^*, \tau_0^*, \kappa_1^*, \tau_1^*, \dots, \kappa_i^*, \tau_i^*\}$ of dual cells in $\mathcal{H}_2^*(M)$, there exists a directed path $\{\tau_i, \kappa_i, \dots, \tau_1, \kappa_1, \tau_0, \kappa_0\}$ of corresponding primal simplices in $\mathcal{H}(M)$ traced in reverse order. Hence, the absence of cycles in $\mathcal{H}_2^*(M)$ can be inferred from the absence of cycles in $\mathcal{H}(M)$. To prove that $\gamma^*(M)$ is connected, we use the same argument as in the proof of Lemma 26 on the graph $\gamma^*(M)$.

Lemma 28. *The graph $\gamma^*(M)$ is connected.*

Proof. Suppose that $\gamma^*(M)$ is disconnected. Let N be the set of nodes in a connected component of $\gamma^*(M)$, and let C be the set of cut edges, that is, edges of \mathcal{K}_2^* with one vertex in N and one vertex in its complement. \mathcal{K}^* is connected since \mathcal{K} is a manifold without boundary. Hence, C is not empty. By definition of $\gamma^*(M)$, each edge in C is matched to a unique 2-cell in \mathcal{K}_2^* . Consider the directed subgraph D of $\mathcal{H}^*(M)$ consisting of the edges in C and their matching 2-simplices. The standard direction of arcs in \mathcal{H}_2^* (from the higher to the lower dimensional cells) is reversed for each matching pair of M , i.e., D is a subgraph of $\mathcal{H}_2^*(M)$. We construct a directed path in D as follows. Start with any node of D corresponding to a cut edge e_1 . Go to the node of D determined by the unique 2-cell τ_1 to which e_1 is matched to. Then, τ_1 contains at least

one other cut edge e_2 , otherwise e_1 cannot be a cut edge. Now iteratively go to e_2 , then to its unique matching 2-cell τ_2 , choose another cut edge e_3 , and so on. We observe that we obtain a directed path $e_1, \tau_1, e_2, \tau_2, \dots$ in D , i.e., the arcs are directed in the correct direction. Since we have a finite graph at some point the path must arrive at a node of D , which we have visited already. So, D (and therefore also $\mathcal{H}_2^*(M)$) contains a directed cycle, which is a contradiction since M is a Morse matching and $\mathcal{H}_2^*(M)$ has no cycles. Hence, $\gamma^*(M)$ is connected. \square

We now rephrase the fact that $\gamma^*(M)$ is connected, in the terminology of interfaces.

Lemma 29. *Suppose that following the design of Morse matching for the $(D - 1)$ -interface, all the $(D - 1)$ -simplices that are matched are deleted. Then, upon execution of this operation the D -interface stays connected.*

Proof. In particular, for a given manifold M , the D -interface obtained after deleting the $(D - 1)$ -simplices that are matched following the design of Morse matching for the $(D - 1)$ -interface is denoted by $\gamma^*(M)$ in terminology of [20]. \square

To design the vector field for 1-interface one may use DFS in the following way. Pick an arbitrary vertex s as the start vertex and mark it critical. Then, invoke the procedure DFS(s):

procedure **DFSoptimal**(v, \mathcal{G})

1. Mark v as visited
2. If there exists an edge $\langle v, w \rangle$ such that w is not visited, then match $\langle w, \langle v, w \rangle \rangle$.
3. DFS(w)

Lemma 30. *There exist simple linear time algorithms to compute optimal Morse Matchings for 1-interface and D -interface of D -dimensional manifolds.*

Proof. Since the graph is connected, every vertex will be visited. Also, except for the start node, every other node is matched before it is visited. The edges that are matched belong to the DFS search tree and hence do not form a cycle. Therefore, the only critical vertex is the start vertex. Therefore, the simple procedure **DFSoptimal**() can be used to design optimal gradient vector field for the 1-interface. Note that the direction of gradient flow for the 1-interface will be exactly opposite of the direction of DFS traversal.

We can associate a dual graph to the D -interface. Also, from Lemma 29, we know that following the design of Morse matching for the $(D - 1)$ -interface and deletion of matched $(D - 1)$ simplices, the dual graph remains connected. So, once again, we can use the procedure **DFSoptimal**() on the dual graph. Therefore, upon application of DFS algorithm for the D -interface, we will have exactly one critical D -simplex which is the start vertex for the DFS and all other D -simplices are regular. This algorithm is optimal since input complex \mathcal{K} is a manifold without boundary and hence must have at least one critical D -simplex. \square

References

- [1] M. Joswig, M. Pfetsch, Computing optimal discrete Morse functions, *SIAM J. Discrete Math.* 20 (1) (2004) 11–25.
- [2] R. Forman, Morse theory for cell complexes, *Adv. Math.* 134 (1) (1998) 90–145, <http://dx.doi.org/10.1006/aima.1997.1650>.
- [3] F. Cazals, F. Chazal, T. Lewiner, Molecular shape analysis based upon the Morse–Smale complex and the Connolly function, in: *Proceedings of the Nineteenth Annual Symposium on Computational Geometry*, ACM, 2003, pp. 351–360.
- [4] N. Shivashankar, S. Maadasamy, V. Natarajan, Parallel computation of 2D Morse–Smale complexes, *IEEE Trans. Vis. Comput. Graph.* 18 (10) (2012) 1757–1770.
- [5] D. Kozlov, Discrete Morse theory, in: *Combinatorial Algebraic Topology*, in: *Algorithms Comput. Math.*, vol. 21, Springer, Berlin, Heidelberg, 2008, pp. 179–209, Ch. 11.
- [6] E. Miller, V. Reiner, B. Sturmfels, *Geometric Combinatorics*, vol. 13, American Mathematical Soc., 2007.
- [7] S. Harker, K. Mischaikow, M. Mrozek, V. Nanda, H. Wagner, M. Juda, P. Dlotko, The efficiency of a homology algorithm based on discrete Morse theory and coreductions, in: *Proc. of 3rd Intl. Workshop on CTIC*, vol. 1, 2010.
- [8] K. Mischaikow, V. Nanda, Morse theory for filtrations and efficient computation of persistent homology, *Discrete Comput. Geom.* 50 (2) (2013) 330–353.
- [9] M. Allili, T. Kaczynski, C. Landi, F. Masoni, A new matching algorithm for multidimensional persistence, arXiv preprint, arXiv:1511.05427.
- [10] T. Lewiner, Constructing discrete Morse functions, Master’s thesis, Department of Mathematics, PUC-Rio, July 2002.
- [11] Ö. Egecioglu, T.F. Gonzalez, A computationally intractable problem on simplicial complexes, *Comput. Geom.* 6 (1996) 85–98.
- [12] T. Kaczynski, K. Mischaikow, M. Mrozek, *Computational Homology*, 1st edition, Appl. Math. Sci., Springer, 2004.
- [13] G. Carlsson, Topology and data, *Bull. Am. Math. Soc.* 46 (2) (2009) 255–308.
- [14] M. Mrozek, B. Batko, Coreduction homology algorithm, *Discrete Comput. Geom.* 41 (1) (2009) 96–118, <http://dx.doi.org/10.1007/s00454-008-9073-y>.
- [15] R. Ayala, D. Fernández-Tertero, J.A. Vilches, Perfect discrete Morse functions on 2-complexes, *Pattern Recognit. Lett.* 33 (2012) 11.
- [16] U. Bauer, C. Lange, M. Wardetzky, Optimal topological simplification of discrete functions on surfaces, *Discrete Comput. Geom.* 47 (2) (2012) 347–377, <http://dx.doi.org/10.1007/s00454-011-9350-z>.
- [17] P. Hersh, On optimizing discrete Morse functions, *Adv. Appl. Math.* 35 (3) (2005) 294–322, <http://dx.doi.org/10.1016/j.aam.2005.04.001>.
- [18] T. Lewiner, H. Lopes, G. Tavares, Optimal discrete Morse functions for 2-manifolds, *Comput. Geom.* 26 (3) (2003) 221–233, [http://dx.doi.org/10.1016/S0925-7721\(03\)00014-2](http://dx.doi.org/10.1016/S0925-7721(03)00014-2).
- [19] T. Lewiner, H. Lopes, G. Tavares, Toward optimality in discrete Morse theory, *Exp. Math.* 12 (3) (2003) 271–285, <http://dx.doi.org/10.1080/10586458.2003.10504498>.

- [20] B.A. Burton, T. Lewiner, J. Paixão, J. Spreer, Parameterized complexity of discrete Morse theory, CoRR, arXiv:1303.7037.
- [21] R. Forman, A user's guide to discrete Morse theory, *Sémin. Lothar. Comb. B* 48c (2002) 1–35.
- [22] M.K. Chari, On discrete Morse functions and combinatorial decompositions, *Discrete Math.* 217 (1–3) (2000) 101–113, [http://dx.doi.org/10.1016/S0012-365X\(99\)00258-7](http://dx.doi.org/10.1016/S0012-365X(99)00258-7).
- [23] M. Hachimori, Simplicial complex library, http://infoshako.sk.tsukuba.ac.jp/~hachi/math/library/index_eng.html, 2001.
- [24] F. Lutz, Vertex-transitive triangulations, <http://page.math.tu-berlin.de/~lutz/stellar/vertex-transitive-triangulations.html>, 2011.
- [25] R. Meshulam, N. Wallach, Homological connectivity of random k -dimensional complexes, *Random Struct. Algorithms* 34 (3) (2009) 408–417, <http://dx.doi.org/10.1002/rsa.v34:3>.
- [26] J.E. Hopcroft, R.M. Karp, A $n^{5/2}$ algorithm for maximum matchings in bipartite graphs, *SIAM J. Comput.* 2 (4) (1973) 225–231.