Linear Time Computation of Discrete Morse Functions Over Two-Manifolds

Anonymous author(s)
Anonymous affiliation(s)

Warning! This is an extremely unpolished preprint with some details withheld, getting ready for submission. Any feedback on it is appreciated!

Abstract

Discrete Morse theory provides a way of studying simplicial complexes akin to studying flows over smooth surfaces. Discrete Morse functions assign a value to each cell, and then pair cells based on homology-preserving gradients. The unpaired cells either represent an essential homology class of the underlying topological space, or are an artifact of the function itself (e.g., a local minimum of the function). We consider two optimization problems: (1) MinMM, finding a function over a given complex K that minimizes the number of critical cells; (2) ExtMM, extending a function over the vertices of a complex to a discrete Morse function compatible with the input function that minimizes the number of critical cells. While it has been shown that MinMM is NP-hard and W[P]-Hard to approximate, we provide a linear time algorithm for the restricted case where the input is a triangulation of a two-manifold. This improves prior algorithms with $\Theta(dn^3)$ complexity on a d-dimensional simplicial complex with n simplices. We give an implementation of this algorithm to demonstrate its improvements in practice. We show how a previously published algorithm solves (2). Finally, we present a heuristic that uses (2) to solve (1), and has reasonable performance on realistic data, even in higher dimensions.

2012 ACM Subject Classification Theory of computation → Computational geometry; Mathematics of computing → Geometric topology

Keywords and phrases discrete Morse theory, persistence

1 Introduction

Classical Morse theory has been deeply influential in the modern topological research paradigm, and assigns continuous functions to smooth manifolds in order to study their topology [22]. In doing so, one major objective is to reveal homology on a manifold through examining critical points of a continuous function. Forman shows in [17] that analogous tools can be utilized in the discrete setting. Indeed, discrete Morse theory is well studied in the computational topology literature, and has been especially fruitful when paired with persistent homology [2,3,7,8,12,13,16,20]. We study discrete Morse functions on a simplicial complex in this work, though our results hold for CW complexes as well. In particular, we consider Discrete Morse functions from three perspectives: the algebraic, the combinatorial, and the topological. Algebraically, a Morse function is a function from the faces of a complex to $\mathbb{R}$, subject to certain inequalities. Combinatorially, a discrete Morse function is an acyclic matching in the Hasse diagram of the complex, where unmatched faces correspond to critical simplices. Topologically, a Morse function takes the form of a gradient vector field on a simplicial complex. These gradient vector fields are composed of matchings between faces such that the collapse of any given matching does not alter the topology of a complex.

As the above discussion suggests, inferences about the topology of a simplicial complex can be made from the number of *critical* simplices given by a Morse function. The problem of minimizing the number of critical simplices is known in the literature as "Minimum Morse Matching", or MinMM. Namely, the number of critical i-cells bounds the ith Betti number. However, generating a Morse function that minimizes the number of critical cells is well known to be NP-hard [19]. Moreover, despite the presentation of related approximation

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

algorithms to MinMM [23], recent work has demonstrated the inapproximability of generating discrete Morse functions, which would seem to dissuade the use of Morse theory for practical applications [4,5]. Nonetheless, in [20] it is shown that with pre-assigned data on the vertices of a complex, one can construct a discrete Morse function in polynomial time. For a complex with dimension less than two or a 2-d subcomplex of a manifold, such methods produce an optimal Morse matching, solving MinMM in polynomial time in low dimensions. This constitutes a gap in knowledge between inapproximability results in high dimensions, and polynomial time methods in lower dimensional settings. Consequently, we are motivated primarily by two questions:

1. Can we categorize the hardness of generating a discrete Morse function in low dimensions?
2. Are there realistic settings where optimal discrete Morse functions are actually relatively easy to approximate in high dimensions?

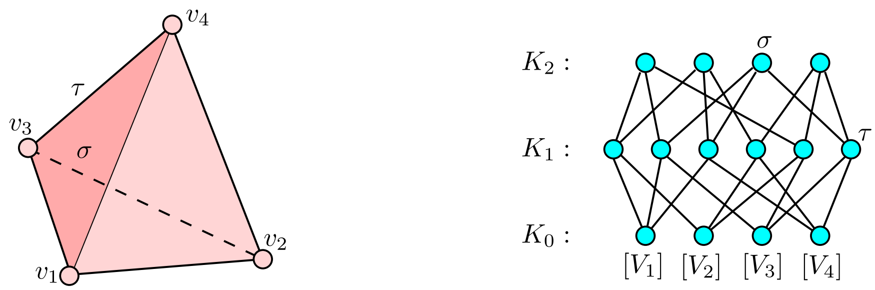
To address the first question, we examine the construction of discrete Morse functions on surfaces. Our methods rely on previously established techniques to find 1-d homology on triangulated two-manifolds. Computing generators of 1-d homology on a surface has been an active area of research in computational topology for some time. A number of efficient algorithms and related data structures exist to compute graphs generating 1-d homology on surfaces, and predominantly rely on the relationship between vertex spanning trees on the 1-skeleton of a surface and the cotree arising from the surface's dual graph [1,6,9,10,14,15,25]. We define these trees and cotrees in Section 2.1, and elaborate on algorithms to compute homology on a surface at length in Section 3.

For the second question, we build from ideas first developed in [20], which computes a discrete Morse function by extending a given injective function on the vertices of a complex into higher dimensions. This introduces an easier variant of MinMM, which is to generate an optimal discrete Morse function on K that is *consistent* to a given injective function on the vertices. We call this variant of the problem which extends vertex values "Extended Morse Matching", or ExtMM. Solutions to ExtMM are *naive* in the sense that they may have a considerable number of extraneous critical cells. However, they do retain a number of desirable properties, which we study in depth in Section 4. We experimentally demonstrate that, for a number of complexes in higher dimensions, ExtMM performs reasonably well in approximating ExtMM. The techniques of [20] are built upon in [16], which shows that naive discrete Morse functions can be generated in $\Theta(dn)$ time, where n is the number of simplices in a given complex and d is its dimension.

In this paper, we provide an algorithm to assign a discrete Morse function to two-manifolds with n simplices that runs in $\Theta(n)$ time, a major improvement upon the previously known $\Theta(n^3)$ algorithm. This represents the first improvement in eighteen years on the computation of discrete Morse functions over two-manifolds, which we hope will increase the practicality of Morse theory in lower dimensional settings. We show that results in [16] compute a discrete Morse function on K which is a solution to ExtMM (i.e. it is consistent to a given injective function f_0 on the vertices of K, and attains the minimal possible number of critical cells while maintaining this property). Lastly, we build from the results in [16] to discuss the viability of ExtMM as a heuristic to compute solutions to MinMM. We arrive at a natural gradient descent algorithm which performs well in realistic settings.

2 Preliminaries

In this section, we provide the definitions and notation used throughout the paper. We adopt the notation of Edelsbrunner and Harer [11]. For a general survey of discrete Morse theory,



(a) A Two-Complex K

(b) The Hasse diagram $\mathcal{H}$ of K

Figure 1 A simplicial complex K and its combinatorial representation $\mathcal{H}$.

see [21, 24]. Note that both describe the major algorithms of King et al. [20], Extract, ExtractRaw, and ExtractCancel, which are a major inspiration for the findings in this work.

2.1 Simplicial Complexes and Two-Manifolds

Let K be a simplicial complex with n simplices. For a simplex $\sigma \in K$, we denote the dimension of σ as $\dim(\sigma)$ and we define $\dim(K)$ to be the maximum dimension of any simplex in K. Throughout the paper, especially when discussing runtimes, we will use $d = \dim(K)$ and $n = \#(K)$ as shorthand for the dimension and number of simplices in K, respectively. We denote the i-simplices of K as K_i and note that (K_0, K_1) is a graph whose vertices are the zero-simplices of K and edges are the one-simplices of K. We write $\tau \prec \sigma$ if τ is a proper face of σ. In this case, we also say that σ is a co-face of τ. The *star* of v in K, denoted $\mathrm{star}_K(v)$, is the set of all simplices of K containing v. The *closed star* of v in K, denoted $\overline{\mathrm{star}}_K(v)$, is the closure of $\mathrm{star}_K(v)$.

We often study simplicial complexes combinatorially through their Hasse diagram. The Hasse diagram of K is a graph whose vertices correspond to the faces of K, and whose edges signify combinatorial relationships between simplices and their faces. We denote the Hasse diagram of K as $\mathcal{H}$. A two-manifold (without boundary) is a topological space whose points all have open disks as neighborhoods. Familiar examples of two-manifolds include a sphere, a torus, and a Klein bottle. See Figure 1 for an example of a simplicial complex which is a triangulated two-manifold, and its corresponding Hasse diagram.

In order to compute discrete Morse functions on surfaces, we rely on important invariants that arise from examining a spanning tree on the 1-skeleton of a complex, and its resulting cotree. The topological properties between spanning trees and their cotrees are well established [15, 25], and have been extended to a number of related algorithms results [1, 6, 9, 10] and efficient data structures [14].

▶ **Definition 2.1** (Cotree). *Let K be a triangulated two-manifold, and T be a spanning tree on its vertices. The resulting cotree of T on K is the dual graph of K, subtracting every edge that crosses an edge in T. See Figure 4 for an example.*

2.2 Discrete Morse Theory

Next, we present the three equivalent definitions of a discrete Morse function used interchangeably throughout the manuscript.

▶ **Definition 2.2** ((Algebraic) Discrete Morse Function). *A function $f : K \to \mathbb{R}$ is a* discrete Morse function *if, for every $\sigma \in K$:*

1. $|\{\alpha \prec \sigma | f(\alpha) \leq f(\sigma)\}| \leq 1$
2. $|\{\sigma \prec \beta | f(\beta) \geq f(\sigma)\}| \leq 1$

As Scoville intuits in [24, p. 49], this is just the requirement that "the function generally increases as you increase the dimension of the simplices. But we allow at most one exception per simplex." A simplex $\sigma \in K$ is *critical* if and only if every face of σ has an algebraic function value less than or equal to $f(\sigma)$, or every co-face of sigma has a value larger than or equal to $f(\sigma)$. See Figure 2b for an example.

This leads naturally to the *topological* notion of a discrete Morse function. In the topological version, rather than denoting function values numerically, they are equivalently recorded by pairwise matchings among faces and cofaces. That is, if (τ, σ) is a face/co-face pair that exemplifies the allowed algebraic exception, then we say that τ and σ are "matched". We call τ the *tail* and σ the *head* in the matching, which are denoted by an arrow on the complex. See Figure 2c for an example. Letting M_f be the set of all matched pairs and C_f be the set of all unmatched simplices, we call the matched simplices *regular* and the simplices in C_f *critical*. In [18], Forman showed that each simplex in K is exclusively a tail, head, or critical.

The pair (M_f, C_f) is called the *gradient vector field (GVF)* on K induced by f. Importantly, collapsing simplices along the gradient preserves the homology of K. In Figure 2c, we show each matching $(\tau, \sigma) \in M$ as an arrow pointing from τ (the tail) to its coface σ (the head). Note that collapsing any simplex on the complex in accordance with these matchings preserves the homology of K. This topological definition is equivalent to Definition 2.2 in the following sense:

▶ **Lemma 2.3** (Topological Morse Functions). *If f and g are two algebraic discrete Morse functions on K that induce the same permutation of the vertices, then $(M_f, C_f) = (M_g, C_g)$.*

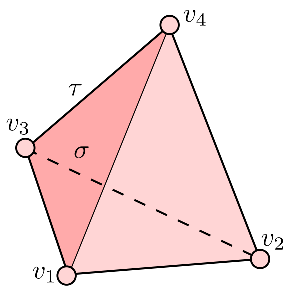
Proof. In Definition 2.2, two sets are defined precisely by the order of the vertices induced by the function f. As a consequence, $M_f = M_g$. Since $M_f = M_g$, we also have $C_f = C_g$. ◀

Defining a GVF on K also naturally suggests a combinatorial representation in the Hasse diagram $\mathcal{H}$ of K. Identically to the topological definition, we can define a discrete Morse function combinatorially by turning $\mathcal{H}$ into a directed graph. Edges in the graph are directed up in dimension if their vertices correspond to a face/coface matching. Otherwise edges in $\mathcal{H}$ are directed down in dimension. It follows that an i-simplex $\sigma \in K_i$ is critical if and only if every edge in $\mathcal{H}$ between σ and an $i-1$ cell directs down in dimension, and there are no edges directed up in dimension from σ to some $i+1$ cell. In other words, this means exactly that σ is indeed unmatched. See Figure 2d for an example.

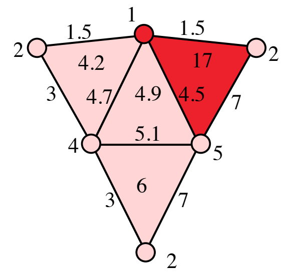
2.3 Computational Problems in Discrete Morse Theory

Perhaps the foremost problem of interest in discrete Morse theory is the Min Morse Matching problem, or MinMM. This asks, given a simplicial complex K, assign it a Morse function which minimizes the number of critical simplices.

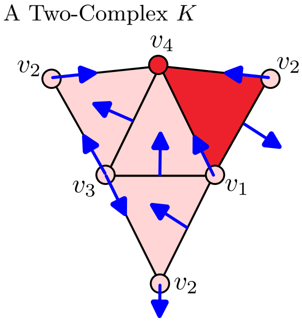
▶ Remark 2.4 (Hardness of MinMM [4]). Bauer et al. shows that the problem MinMM is W[P]-hard to approximate, with respect to the standard parameterization (i.e., solution size). This result especially would seem to discourage the use of discrete Morse theory in many practical settings.



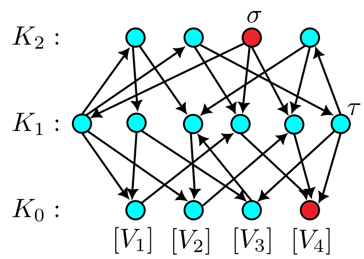
(a) A Two-Complex K



(b) An algebraic Morse function on K



(c) A GVF on K



(d) A GVF on $\mathcal{H}$

Figure 2 The algebraic, topological, and combinatorial interpretations of a Morse function. In each example, critical simplices are given in red.

We study both the general problem of MinMM, and the more restricted version of producing a minimal discrete Morse function on K which is *consistent* with a given injective function $f_0 : K_0 \to \mathbb{R}$ on the vertices of K.

▶ **Definition 2.5** (Consistent Morse Function). *Let $v \in \sigma$ denote a vertex of σ. We say that a Morse function $f : K \to \mathbb{R}$ is* consistent *with $f_0 : K_0 \to \mathbb{R}$ if for every $\epsilon > 0$ and $\sigma \in K$:*

$$f|_{K_0} = f_0, \text{ and } f_0(\sigma) - \max_{v\in\sigma} f_0(v) < \epsilon$$

In other words, gradients in higher dimensions flow away from the largest valued vertices and toward the smallest ones. We call this restriction the Extended Morse Matching problem, or ExtMM, since the optimality of the end Morse function is restricted by the extension of given weights on the vertices.

▶ **Observation 2.6.** *Notice that, an output to ExtMM may not be an output to MinMM. For a simple example, see Figure 3. In many cases, it is impossible to construct a Morse function which is both consistent with K and minimizes $|C|$.*

▶ **Theorem 2.7** (Computing an Optimal Solution to ExtMM is $\Theta(dn)$)**.** *In [16], a heuristic algorithm is given which solves ExtMM and runs in $\Theta(dn)$ time, where d is the dimension and n is the total number of simplices of a simplicial complex K.*

Proof. Algorithm **??**, originally given in [16], which we include in Appendix **??**, is an optimal solution to ExtMM. That is, an output Morse function $f : K \to \mathbb{R}$ from ExtractRightChild minimizes $|C|$ while retaining consistency with the given $g : K_0 \to \mathbb{R}$. Indeed, ExtractRightChild greedily pairs the smallest available lexicographical i-simplex with its largest lexicographical child. Consequently, $|C|$ is minimized, since the smallest lexicographical $i-1$-simplices have necessarily been paired in the process, and hence $f(\sigma) = \max_{v\in\sigma} f_0(v)$ for the greatest possible number of $\sigma \in K$. ◀

(a) An example input to ExtMM **(b)** The consistent (non-optimal) GVF

Figure 3 A simplicial complex with an accompanying injective function on the vertices that yields a consistent GVF not minimizing $|C|$, but which has the smallest such $|C|$ to be consistent.

With this algorithm in hand, it is helpful to consider solutions to ExtMM as efficient heuristics to address MinMM. While no strong guarantees can be made about $|C|$ in the gradient vector field resulting from Algorithm **??**, we can use properties of solutions to ExtMM to arrive at a natural gradient descent algorithm later on in Section 4.

In [20], ExtMM is first proposed as a preliminary step for computing a discrete Morse function. The process is formulated in two parts: generating a "raw" Morse function with possibly very many critical cells, and then refining the Morse function by increasing the number of pairs. We provide a formal time complexity analysis of both steps which require in total $\Theta(dn^3)$ time on a d-dimensional simplicial complex with n cells. Despite being hard to compute and even approximate in high dimensions, [20] shows that for two-manifolds one can refine a "raw" Morse function to an optimal one in polynomial time.

We adopt a different framework in the case for two-manifolds, first computing homology groups using fast persistence-based methods, and extending a gradient vector field subsequently. This reduces the computation of a Morse function on two-manifolds to $\Theta(n)$. In higher dimensions, we adopt a similar framework to [20], generating a "raw" Morse function which is a solution to ExtMM, and refining it. In theory, the step of refining a Morse function is much more intensive than creating a "raw" one initially. For this, we employ a natural gradient descent heuristic using permutations of the given $f_0 : K \to \mathbb{R}$, which provides $\Theta(dn)$ time updates to a gradient vector field.

3 An Algorithm for Computing MinMM for Two-Manifolds

In this section, we give a $\Theta(n)$-time algorithm solving MinMM for two-manifolds with n vertices. This is a primary theoretical result of the paper, and reduces the time complexity to compute discrete Morse functions on two-manifolds for the first time since 2005. Our algorithm relies on invariants of spanning trees and their cotrees on triangulations, which are defined in Definition 2.1. For an example, see Figure 4. We also provide a C++ implementation of our algorithm, and experimentally validate our runtime in practice.

3.1 MorseDual

We call our algorithm MorseDual, because of its reliance on the dual graph of a surface. The algorithm works as follows: first, we compute $T = (K_0, E_T)$, a spanning tree of the one-skeleton of K. Then, let $G^* = (V^*, E^*)$ be the complementary dual graph of K with respect to the edges of T (that is, G^* is the dual graph, removing dual edges of E_T). For

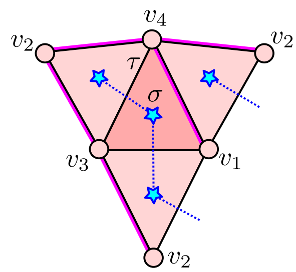
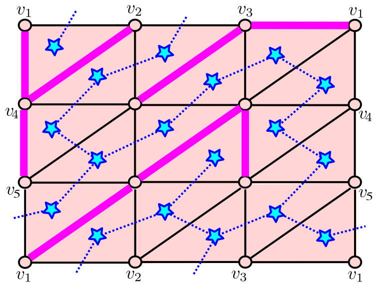


Figure 4 A spanning tree on the vertices of K with edges in pink, and its corresponding cotree with vertices as blue stars and dotted edges. The cotree is the dual graph of K without edges intersecting the spanning tree.

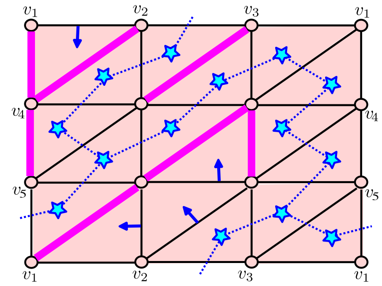
Algorithm 1 MorseDual

Input: K, a triangulation of a two-manifold
Output: a GVF over K minimizing C over all GVFs over K

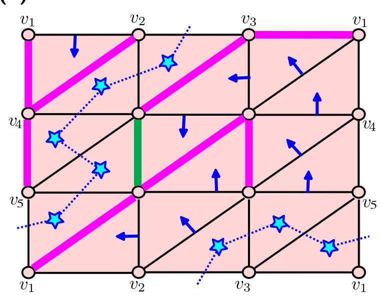
Compute a spanning tree $T = (K_0, E_T)$ of K
Compute the complementary dual graph $G^* = (K_2^*, E^* := K_1^* \setminus E_T)$
$C \leftarrow \emptyset$ ▷ critical cells
$M \leftarrow \emptyset$ ▷ matched cells
For each cell in K, add an attribute 'marked' and set it to False
Let T' denote the sub-tree of T comprising unmarked cells (implicitly stored)
while $\exists$ unmarked leaf node v in T' **do** ▷ match cells of T
    Let e be the edge that connects v to the rest of T'.
    Mark e and v
    Add (v, e) to M
end while
$v \leftarrow$ unmarked vertex of K_0
Add v to C.
Let G' denote the sub-graph of G^* whose vertices/edges correspond to unmarked cells in K.
while $\exists$ unmarked cells of K **do**
    while $\exists$ unmarked degree-one vertex v^* in G' **do**
        Let e^* be the edge that connects v^* to the rest of G'.
        Let (e, f) be the dual to (e^*, v^*)
        Mark e and f
        Add (e, f) to M
    end while
    if $\exists$ unmarked edge $e^* \in E^*$ **then** ▷ e^* must be in a cycle
        Mark e^*
        Add e^* to C
    end if
end while



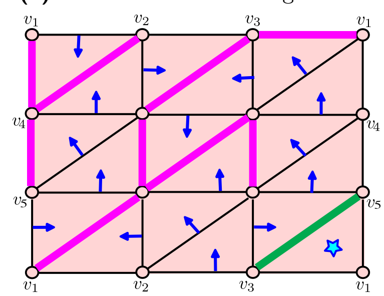
(a) The tree T and cotree C of K.



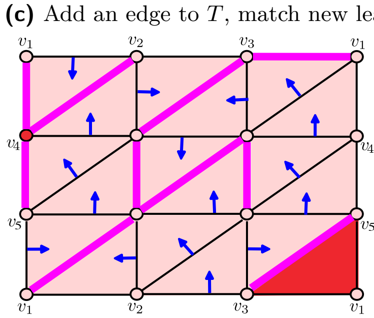
(b) Define Morse matchings for every leaf of C



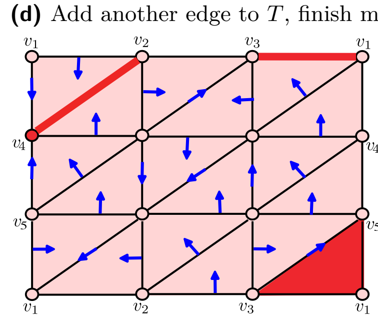
(c) Add an edge to T, match new leaves.



(d) Add another edge to T, finish matchings



(e) Last face and root r of T are critical.



(f) Collapse all $e \in E_T$ into r, loops are critical

Figure 5 The algebraic, topological, and combinatorial interpretations of a Morse function. In each example, critical simplices are given in red.

every leaf node $v \in V^*$ of G^* (which is a face in K_2), we define a Morse matching whose tail is the dual edge (which is an edge in K_1), and whose head is v. When no additional leaves are available, we add a free edge to T, removing an edge from G^*. We repeat the process, adding Morse matchings along leaves until $G*$ has no additional edges. We call every remaining $v \in V^*$ a critical 2-cell. Left only with T, we define a Morse matching between the root $r \in K_0$ of T and one of its edges $e \in E_T$. We delete the matched edge, and collapse the other vertex of e to r. We continue the process, calling any resulting self-looping edges critical. Lastly, we call any remaining vertices critical after all edges have been collapsed. (For a two-manifold without boundary, there will be only one critical vertex.) For an example, see Figure 5

▶ **Lemma 3.1** (MorseDual recovers $|C_1| = \beta_1$). *MorseDual minimizes the number of critical edges in its output GVF.*

Proof. Our algorithm in fact computes $\beta_1(K)$ which is the maximum number of cuts that can be made before separating the given 2-manifold into two pieces. As the number of critical edges is always an upper bound of $\beta_1(K)$, we have $|C_1| = \beta_1(K)$, which is optimal. ◀

We comment that similar methods can also compute β_1 in linear time [10, 14, 25].

▶ **Lemma 3.2** (MorseDual recovers $|C_2| = \beta_2$)**.** *MorseDual minimizes the number of critical faces in its output GVF.*

Proof. Without loss of generality we assume $\beta_0(K) = 1$, since otherwise we could just repeat the algorithm on each connected component. The dual graph D computed on K must be connected, since none of its edges intersect with the edges of T, which is a tree. This leaves two options for the subsequent collapses on D:

1. D can collapse directly to a boundary, if one exists on K.
2. D can collapse to a $\sigma_2 \in K$ surrounded by $\sigma_1 \in T$, forcing no Morse matching.

◀

▶ **Theorem 3.3** (MorseDual is $\Theta(n)$)**.** *MorseDual terminates in $\Theta(n)$ time, using $\Theta(n)$ space.*

Proof. Let K denote a triangulated two-manifold, or a subcomplex thereof. Computing a spanning tree T on the 1-skeleton of K is easily linear in the number of simplices in K. Computing the dual graph D of K not intersecting edges in T is also simple to do in linear time when considering that each $\sigma_2 \in K$ has three adjacent faces, and hence the dual graph is given by $O(3n)$ operations. Moreover, when collapsing leaves of the dual graph (i.e. collapsing $\sigma_2 \in K$), each face is only touched once. Finally, when all faces have been collapsed, the remaining spanning tree is collapsed into its root, thereby assigning a gradient vector field among remaining edges in linear time. Indeed, every step of MorseDual concludes in linear time, but each process may well require $\omega(n)$ operations, and hence MorseDual has $\Theta(n)$ time complexity. Moreover, MorseDual uses $\Theta(n)$ space, as only a constant number of copies of each $\sigma \in K$ must be saved. ◀

3.2 MorseDual in Practice

In what follows, we provide experimental data demonstrating the practical improvements brought forth by Algorithm 1. We implemented MorseDual in C++, and our code is publicly available on github. Our implementation is compared against the C implemetation of King et al. We ran our implementation on a set of trianguled surfaces, ranging from 12 vertices to roughly 1300. We highlight that the theoretical time complexity discussed for MorseDual holds in practice as n increases. Experimental time complexity is as follows, all on triangulated surfaces.

$\|K_0\|$	12	258	405	770	1329
Time (ms)	19	511	808	1535	2652

4 A Heuristic with Experiments

4.1 A Basic Gradient Descent to Approximate MinMM

Though MinMM is W[P]-hard to approximate (see Remark 2.4) in dimensions larger than two, we demonstrate experimental results indicating cases with practical relevance that may be easier. It is important to keep in mind tha a primary application of discrete Morse theory is in persistent homology, where DMT can reduce the size and complexity of data in a topologically faithful way. As such, our experimentation is primarily conducted on Vietoris-Rips complexes, which are often central in persistent homology.

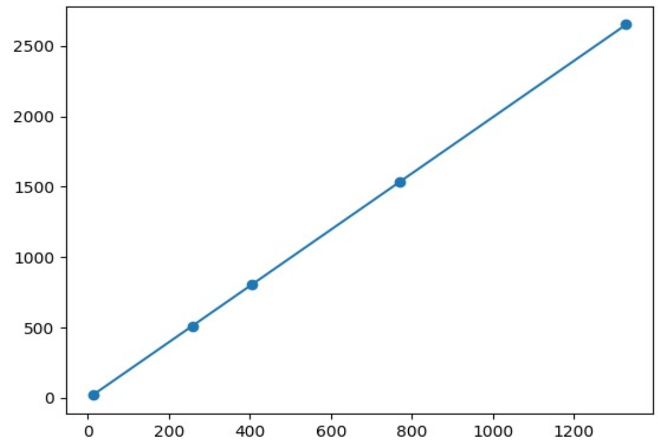


Figure 6 As proof of concept, we find linear asymptotic behavior in practice n grows.

To approximate solutions of MINMM, we first compute solutions to EXTMM, and then refine them using a fast gradient descent heuristic. Recall that a similar approach is taken in [20], though solutions to EXTMM are refined by reversing gradient paths in the Morse function, which takes cubic time The time complexity of our heuristic is dependent on the sparsity of a complex. For a sparse complex (which is typically a reasonable assumption in persistent homology applications), our heuristic runs in linear time. Though, in the worst case, it is technically possible for our gradient descent to require quadratic time on dense complexes.

Recall that solutions to EXTMM give a Morse function that is consistent to a given function $f_0 : K \to \mathbb{R}$, while achieving the fewest possible number of critical cells to be consistent. As a result, orderings on the vertices matter substantially for EXTMM, and permuting vertex values carefully can lead to Morse functions with fewer critical cells. For example, consider Figure 3. Permuting the vertex $f_0^{-1}(1)$ with either $f_0^{-1}(3)$ or $f_0^{-1}(4)$ leads to an optimal Morse matching, whereas permuting $f_0^{-1}(1)$ with $f_0^{-1}(2)$ makes no difference. This alludes to the fact that we can refine the search space of MINMM by not permuting any vertices that would make no difference to solutions of EXTMM. We demonstrate a substantial class of permutations that attain the same solutions to EXTMM. Denote a permutation as p, and the solution to EXTMM after applying p to K as EXTMM $(p(K))$. We write id for the identity permutation.

▶ **Lemma 4.1** (Plateau). *Let $v, a, b \in K_0$ such that $a, b \in \overline{star}_K(v)$. Suppose $f(b)$ is the smallest upper bound of $f(v)$ in $\overline{star}_K(v)$, and $f(a)$ is the greatest lower bound of $f(v)$ in $\overline{star}_K(v)$. If we chose a permutation p of a vertex $u \in K_0$ with v where $f_0(u) \in (f(a), f(b))$, then EXTMM $(p(K))$ = EXTMM $(id(K))$.*

Proof. The proof is simple after unpacking definitions. We need to show that local orderings among vertices in $\overline{\text{star}}_K(u)$ and $\overline{\text{star}}_K(v)$ are invariant after the application of p. Without loss of generality, suppose $f_0(v) < f_0(u) < f_0(b)$. Since $f(b)$ is the least upper bound in $closedStarKv$, it follows that any $b' \in \overline{\text{star}}_K(v)$ with $f_0(v) < f_0(b')$ has $|f_0(v) - f_0(b)|leq|f_0(v) - f_0(b')|$. Moreover, $|f_0(v) - f_0(u)| < |f_0(v) - f_0(b)|$ by assumption. Carrying over these inequalities, $|f_0(v) - f_0(u)| < |f_0(v) - f_0(b')|$, and the local ordering

$f(v) < f(u) \leq f(b')$ is identical, substituting u for b. Reversing the argument, the same relationship is true for any adjacent vertices less than v and the greatest lower bound a. Hence, $p(K)$ and $id(K)$ maintain the same local orderings for every vertex. Consequently, solutions to ExtMM $(p(K))$ and ExtMM $(id(K))$ are the same. ◀

This tells us that for any vertex $v \in K_0$, permuting nonadjacent vertices to v will not reduce the number of critical cells in a Morse matching if the vertices are sufficiently close in value. Let K be a simplicial complex of dimension d on n simplices. Using Lemma 4.1, we obtain a natural gradient descent algorithm by running the $\Theta(dn)$ algorithm given in [16], and then permuting adjacent vertices who are nearest above and nearest below in value to a given vertex. We can check if a permutation decreases $|C|$ in $\Theta(d)$ time for simplicial complexes with a sparse one-skeleton, and at each iteration we keep the permutation decreasing $|C|$ the most. Then, the whole process takes $\Theta(dn)$ time assuming K with a sparse one-skeleton.

Algorithm 2 GradientDescent

Input: K, a simplicial complex, and p_0, a permutation of the vertices
Output: a locally optimal GVF over K

$(M, C) \leftarrow$ ExtractRightChild(K, p_0).
critical $\leftarrow \infty$
while $|C| <$ *critical* **do**
  critical $\leftarrow |C|$
  for $v \in K_0$ **do**
    $B \leftarrow \{u \in \overline{\text{star}}_K(v) | f_0(u) > f_0(v)\}$
    $A \leftarrow \{u \in \overline{\text{star}}_K(v) | f_0(u) < f_0(v)\}$
    $a \leftarrow u \in A$ s.t. $f(a) = \inf_{u \in A}(f_0(u))$
    $b \leftarrow u \in B$ s.t. $f(b) = \sup_{u \in B}(f_0(u))$
    Permute (a, v) and examine updated $GVF(K)$
    Permute (b, v) and examine updated $GVF(K)$
  end for
  if Any permutation reduced $|C|$ **then**
    $K \leftarrow p(K)$, where p is the permutation causing the biggest reduction in $|C|$
    Update $GVF(K)$ with the adjusted Morse function
    critical $\leftarrow |C|$
  end if
end while
return $GVF(K)$

▶ **Lemma 4.2** (GradientDescent Update Time Complexity)**.** *When a new permutation of f_0 is chosen by GradientDescent, updates on the GVF are made in $O(deg(v))$ time, where $deg(v)$ is the degree of a permuted vertex v on the one-skeleton of K. If the one-skeleton is sparse, updates are made in $\Theta(d)$ time.*

Proof. Let g_0 denote the function on the vertices of K after a given permutation occurs. One can update $GVF(K)$ in $\Theta(d)$ time by traversing up in dimension on $\mathcal{H}$ from a permuted vertex, and updating the algebraic Morse function $f(\sigma)$ for each $\sigma \in \text{star}_K(v)$ by choosing $\max(g_0(\sigma), f(\sigma)) = f(\sigma)$. The number of upward edges from v is bounded by the number of edges with which v participates, and hence an update could be $O(n)$ in the worst case if v has high degree. If the one-skeleton is sparse, updates are made in $O(d)$ time, since there are $O(n)$ total edges, meaning there are only $O(d)$ upward edges in $\mathcal{H}$ per vertex. ◀

▶ Remark 4.3. While theoretically perhaps discouraging, in practice sparsity of the one-skeleton is not necessarily an unreasonable assumption to make. We demonstrate the update times in practice of randomly generated complexes in the next subsection.

4.2 Morse Gradient Descent in Practice

On average, over a huge class of randomly generated Vietoris Rips complexes, we found that our gradient descent eliminates a bit over half of the critical cells in a given Morse function. The full details are coming soon!

5 Discussion

References

1. Pankaj K. Agarwal, Herbert Edelsbrunner, John Harer, and Yusu Wang. Extreme elevation on a 2-manifold. In *Proceedings of the Twentieth Annual Symposium on Computational Geometry*, pages 357–365, 2004.
2. Ulrich Bauer. *Persistence in Discrete Morse Theory*. PhD thesis, Niedersächsische Staats-und Universitätsbibliothek Göttingen, 2011.
3. Ulrich Bauer, Carsten Lange, and Max Wardetzky. Optimal topological simplification of discrete functions on surfaces. *Discrete and Computational Geometry*, 47(2):347–377, 2012.
4. Ulrich Bauer and Abhishek Rathod. Hardness of approximation for morse matching. In *SODA*, pages 2663–2674, 2019.
5. Ulrich Bauer and Abhishek Rathod. Parameterized inapproximability of morse matching. arXiv:2109.04529, 2021.
6. Kree Cole-McLaughlin, Herbert Edelsbrunner, John Harer, Vijay Natarajan, and Valerio Pascucci. Loops in Reeb graphs of 2-manifolds. In *Proceedings of the nineteenth annual symposium on Computational geometry*, pages 344–350, 2003.
7. Lidija Čomić and Leila De Floriani. Dimension-independent simplification and refinement of Morse complexes. *Graphical Models*, 73(5):261–285, 2011.
8. Tamal Dey, Jiayuan Wang, and Yusu Wang. Graph reconstruction by discrete Morse theory. In *34th Symposium on Computational Geometry (SoCG)*, pages 31:1–31–13, 2018.
9. Tamal K. Dey. A new technique to compute polygonal schema for 2-manifolds with application to null-homotopy detection. In *Proceedings of the Tenth Annual Symposium on Computational Geometry*, SoCG '94, page 277–284, New York, NY, USA, 1994. Association for Computing Machinery. doi:10.1145/177424.178001.
10. Tamal K. Dey, Fengtao Fan, and Yusu Wang. An efficient computation of handle and tunnel loops via Reeb graphs. *ACM Transactions on Graphics (TOG)*, 32(4):1–10, 2013.
11. Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. American Mathematical Society, 2010.
12. Herbert Edelsbrunner, John Harer, and Afra Zomorodian. Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds. *Discrete and Computational Geometry*, 30(1):87–107, 2003.
13. Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. *Discrete and Computational Geometry*, 28:511–533, 2002.
14. David Eppstein. Dynamic generators of topologically embedded graphs. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 599–608, 2003.
15. Jeff Erickson. One-dimensional computational topology lecture notes. Lecture 10: Tree-CoTree Decompositions, 2020.
16. Brittany Fasy, Benjamin Holmgren, Bradley McCoy, and David Millman. If you must choose among your children, pick the right one. In *Canadian Conference on Computational Geometry (CCCG 2020)*, August 2020.

17 Robin Forman. Discrete Morse theory for cell complexes. *Advances in Mathematics*, 134:90–145, 1998.

18 Robin Forman. A user's guide to discrete Morse theory. *Séminaire Lotharingien de Combinatoire*, 42:Art. B48c, 35pp, 2002.

19 Michael Joswig and Marc Pfetsch. Computing optimal Morse matchings. *SIAM Journal on Discrete Mathematics (SIDMA)*, 20(1):11–25, 2006.

20 Henry King, Kevin Knudson, and Nevžza Mramor. Generating discrete Morse functions from point data. *Experimental Mathematics*, 14:435–444, 2005. MR2193806.

21 Kevin Knudson. *Morse Theory: Smooth and Discrete.* World Scientific Publishing Company, 2015.

22 John Milnor. *Morse Theory.* Princeton University Press, Princeton, New Jersey, 1963.

23 Abhishek Rathod, Talha Bin Masood, and Vijay Natarajan. Approximation algorithms for max morse matching. *Comput. Geom.*, 61:1–23, 2017.

24 Nicholas Scoville. *Discrete Morse Theory.* American Mathematical Society, Providence, Rhode Island, 2019.

25 Gert Vegter and Chee K. Yap. Computational complexity of combinatorial surfaces. In *Proceedings of the Sixth Annual Symposium on Computational Geometry*, pages 102–111, 1990.