Linear Time Computation of Discrete Morse Functions On Two-Manifolds

Anonymous author
Anonymous affiliation

Anonymous author
Anonymous affiliation

Anonymous author
Anonymous affiliation

Anonymous author
Anonymous affiliation

Anonymous author
Anonymous affiliation

Warning! This is a preprint being finalized for submission! It is still being worked on, and any feedback is appreciated. If you read the paper and find bugs, please send them to benjamin.holmgren1@student.montana.edu

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 simplex, and then pair simplices based on homology-preserving gradients. The unpaired "critical" cells either represent an essential homology class of the underlying topological space, or are a vestige of the function itself (e.g., a local minimum of the function). We consider an optimization problem of high interest in computational topology: MinMM, which is to find a function over a given simplicial complex K that minimizes the number of critical simplices. Though 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 implement this algorithm to demonstrate its improvements in practice. We elaborate on a prominent formulation of MinMM, which extends a Morse matching to K given an injective function $f_0 : K_0 \to \mathbb{R}$ on the vertices of K. We use this formulation to present a novel gradient descent heuristic that approximates MinMM well on a realistic class of complexes. In doing so, we arrive at a randomized additive approximation algorithm for MinMM in settings with low relative torsion.

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

Keywords and phrases discrete Morse theory, persistence

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

Acknowledgements Anonymous acknowledgements

1 Introduction

In the celebrated results of classical Morse theory, continuous functions are assigned to smooth manifolds in order to study their topology [22]. For example, the homology of a manifold can be computed by examining critical points of the continuous functions. In [17], Forman shows that analogous tools can be defined and utilized in the discrete setting; leading to the birth of a field known as *discrete Morse theory*. Discrete Morse theory has been especially fruitful when paired with persistent homology [2, 3, 7, 8, 12, 13, 16, 20], where it is often used to reduce the size and complexity of data in a topologically faithful manner. 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 specific inequalities as faces increase in dimension. 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. Namely, the number of critical i-cells is an upper bound to the rank of the ith homology group (i.e., the ith Betti number). As a consequence, minimizing the number of critical simplices in a Morse function is highly desirable. Doing so gives not only the Betti numbers, but more importantly the *exact instructions* of how to obtain them by collapsing simplices. The problem of minimizing the number of critical simplices is known in the literature as "Minimum Morse Matching", or MinMM. However, generating a Morse function that minimizes the number of critical cells is well known to be NP-hard [19]. Moreover, recent work has demonstrated the inapproximability of generating discrete Morse functions on complexes that are not subcomplexes of two manifolds, which might seem to dissuade the use of Morse theory in many 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 non-optimal discrete Morse function in polynomial time. If the complex is a subcomplex of a two-manifold, the algorithms in [20] produce a discrete Morse function that minimizes the number of critical cells, solving MinMM for surfaces. The methods in [20] execute in cubic time with respect to the number of simplices in a complex. 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 characterize the fine-grained hardness of MinMM for triangulated two-manifolds?
2. Are there realistic settings, restricting the general case, where solutions to MinMM are actually relatively easy to approximate in high dimensions?
3.

To address the first question, we examine the construction of discrete Morse functions on surfaces. Our methods rely on previously established techniques to find the one-dimensional homology of triangulated two-manifolds. A number of efficient algorithms and related data structures exist to compute graphs generating one-dimensional homology of surfaces, and predominantly rely on the relationship between vertex spanning trees on the one-skeleton of a surface and the cotree arising from the surface's dual graph [1, 6, 9, 10, 14, 15, 24]. 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 $f_0 : K_0 \to \mathbb{R}$ on the vertices of a complex into higher dimensions. In [20], discrete Morse functions are computed on K with dimension d and n simplices in two steps: (1) *naïve* discrete Morse functions are computed using the given f_0, and then (2) refined towards optimal Morse functions in $\Theta(dn^3)$ time.

It is shown that the refinement in step (2) produces optimal discrete Morse functions (i.e. solutions to MinMM) for subcomblexes of two-manifolds, but is not optimal otherwise. Note that naïve discrete Morse functions can have a considerable number of critical cells, but are aligned with a given injective function f_0 on the vertices of K, and represent an important initial step for the computation in [20]. The techniques of [20] are built upon in [16], which shows that the same naïve discrete Morse functions can be generated in $\Theta(dn)$ time.

This paper experimentally demonstrates the closeness of naïve discrete Morse functions to optimal ones in realistic settings. Additionally, this paper reveals desirable properties of the naïve discrete Morse functions computed in [16, 20]. These are studied in depth in Section 4, and lead to a natural gradient descent heuristic, further improving naïve discrete Morse functions as a mechanism to approximate MinMM. We provide a theoretical justification of our experimental results using probability theory. In doing so, we arrive at a randomized additive approximation algorithm for MinMM on simplicial complexes with low relative torsion, even in high dimensions.

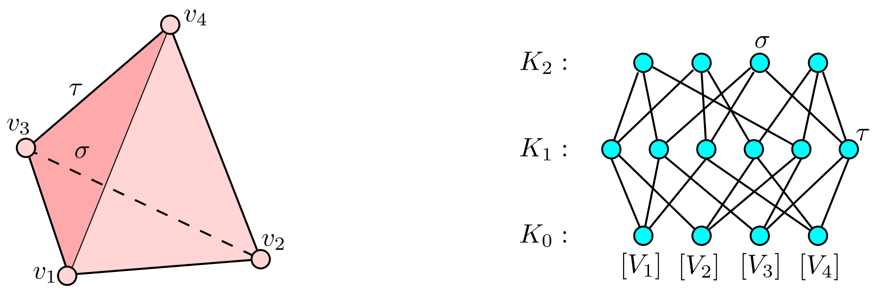
We begin by introducing the relevant definitions and prior work in discrete Morse theory in Section 2. In Section 3, we provide an algorithm to assign an optimal discrete Morse function to two-manifolds with n simplices that runs in $\Theta(n)$ time, a major improvement upon the previously known $\Theta(n^3)$ algorithm from 2005. In Section 4 we build from the results in [16] to demonstrate the practical similarity of naïve Morse matchings to optimal solutions of MinMM. This leads us to a gradient descent algorithm improving naïve Morse matchings further. Finally, in Section 5, we discuss the theoretical ramifications of our findings in Section 4. We conclude with a randomized algorithm that approximates MinMM well on a class of simplicial complexes with low relative torsion.

2 Preliminaries

In this section, we provide the definitions and notation used throughout the paper. We assume basic knowledge in computational topology, and refer readers to Edelsbrunner and Harer [11] for additional background. For a general survey of discrete Morse theory, see [21,23]. Note that both describe the algorithms of King et al. [20], Extract, ExtractRaw, and ExtractCancel, which are a significant inspiration for the findings in this work. We include the algorithms of [20] for reference in Appendix B.

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. We slightly abuse notation and write $v \in \sigma$ if v is a vertex of σ. 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 whose edges are the one-simplices of K. We call this graph the one-skeleton of K. We write $\tau \prec \sigma$ if τ is a proper face of σ. That is, if $\dim(\sigma) = i$, then $\dim(\tau) = i - 1$ and every vertex of τ is also a vertex 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 $\mathrm{star}_K(v)$. In the later sections of the paper, we study refined classes of simplicial complexes, including *sparse* complexes. We say that K is sparse if $\#(K_0) = O(n)$.



(a) A Two-Complex K **(b)** The Hasse diagram $\mathcal{H}$ of K

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

We often study simplicial complexes combinatorially through their Hasse diagram. We denote the Hasse diagram of K as $\mathcal{H}$. The Hasse diagram $\mathcal{H}$ is a graph whose vertices correspond to the simplices of K, and whose edges signify combinatorial relationships between simplices and their faces. Namely, if $\tau \prec \sigma$, the Hasse diagram has an edge between the two vertices corresponding to τ and σ in $\mathcal{H}$. The results in this paper begin by studying discrete Morse theory on two-manifolds. A two-manifold (without boundary) is a topological space whose points all have neighborhoods homeomorphic to open disks. Familiar examples of two-manifolds include a sphere, a torus, and a Klein bottle. See Figure 1 for an example of a simplicial complex that is a triangulated two-manifold, and its corresponding Hasse diagram.

A spanning tree of a simplicial complex K is a subset of the edges in K_1 that includes every vertex in K_0 and no cycles. If K is a two manifold, the dual graph G of K is a graph whose vertices represent the faces of K_2, and two vertices in G are joined by an edge if there is an edge in K_1 between the two corresponding faces in K_2. A spanning tree of the dual edges is called a spanning cotree. A *tree-cotree partition* of K is a triple (T, C, X) where T is a spanning tree, C is a spanning cotree and X is the remaining edges in K_1. The sets T, C and X are disjoint [14]. Tree-cotree partitions have been explored in [15, 24], and have been extended to a number of related algorithmic results [1, 6, 9, 10] and efficient data structures [14].

2.2 Discrete Morse Theory

Discrete Morse functions can be thought of in three superficially dissimilar forms, where each interpretation reveals useful properties in differing situations. We present the three equivalent definitions of a discrete Morse function, which are used interchangeably.

▶ **Definition 2.1** ((Algebraic) Discrete Morse Function). *A function $f : K \to \mathbb{R}$ is a* discrete Morse function *if, for every $\sigma \in K$:*

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

As Scoville intuits in [23, 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 σ 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 realizes 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 be the set of all matched pairs and C be the set of all unmatched simplices, we call the matched simplices *regular* and the simplices in C *critical*. As a shorthand, we write M^T as the set of all tails in M, and M^H to denote the heads in M In [18], Forman showed that each simplex in K is exclusively a tail, head, or critical.

The pair (M, C) is called the *gradient vector field (GVF)* on K induced by f. 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 $\tau \in M^T$ to its coface $\sigma \in M^H$. This topological definition is equivalent to Definition 2.1 in the following sense:

▶ **Lemma 2.2** (Topological Morse Functions)**.** *If f and g are two algebraic discrete Morse functions that induce the same permutation on the simplices of K, then $(M_f, C_f) = (M_g, C_g)$.*

Proof. In Definition 2.1, two sets are defined precisely by the order of the simplices induced by the function f. As a consequence, $M_f = M_g$. Since $M_f = M_g$ and (M, C) partitions K, we also have $C_f = C_g$. ◀

Thus, a discrete Morse function can be considered without loss of generality by local orderings of simplices.

A GVF on K also gives a combinatorial representation in the Hasse diagram $\mathcal{H}$ of K. Edges in $\mathcal{H}$ 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, for $i > 0$, 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. If $\sigma \in K_0$ is a vertex, only the second property is needed to assign $\sigma \in C$. In other words, this means exactly that σ is indeed unmatched. See Figure 2d for an example, and Appendix A for further elaboration on discrete Morse functions in the combinatorial setting.

2.3 Computational Problems in Discrete Morse Theory

A fundamental question in discrete Morse theory is how to find a discrete Morse function that minimizes the number of critical cells. That is,

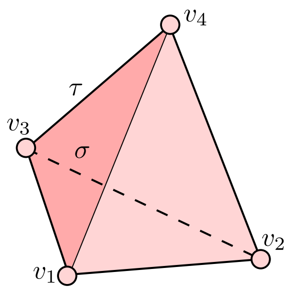
▶ **Problem 2.3** (Minimum Morse Matching, MinMM)**.** *Given a simplicial complex K, assign a gradient vector field (M, C) to K that minimizes the number of critical simplices, $|C|$.*

In terms of posets, we phrase this as: assign a partial order on the simplices of K so that removing all flipped pairs minimizes the number of remaining simplices.

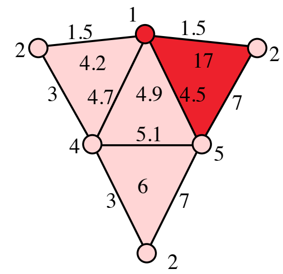
Above, by flipped pair, I mean: (σ, τ) is a flipped pair if τ is a codimension-one face of σ and τ appears after σ.

MinMM is perhaps the foremost problem of interest in discrete Morse theory, and its hardness in general settings is well studied.

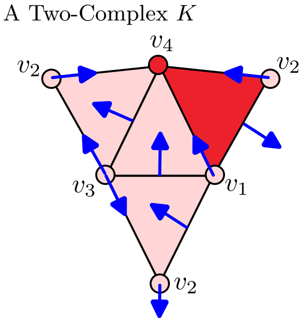
▶ Remark 2.4 (Hardness of MinMM [4]). Bauer et al. shows that the problem MinMM is W[P]-hard to approximate, with respect to the standard parameterization (i.e., solution size). This result would especially seem to discourage MinMM in practical settings.



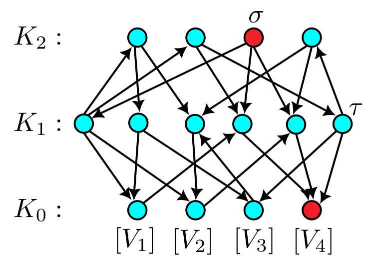
(a) A Two-Complex K



(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 also study methods introduced first in [20], which uses a given injective function $f_0 : K_0 \to \mathbb{R}$ on the vertices to compute solutions to MinMM. In both [16, 20], f_0 is used to compute *naïve* discrete Morse functions, which are then refined. Note that f_0 induces a lexicographical ordering on the higher dimensional simplices of K. We write $lex(\sigma) > lex(\sigma')$ if σ is lexicographically larger than σ', and remark that due to the injectivity of f_0, $lex(\sigma) = lex(\sigma')$ if and only if $\sigma = \sigma'$. We define naïve discrete Morse functions in the context of a given injective f_0 as follows:

▶ **Definition 2.5** (Naïve Discrete Morse Function). *We say that a discrete Morse function $f : K \to \mathbb{R}$ is* naïve *with respect to an injective $f_0 : K_0 \to \mathbb{R}$ if:*

1. $f|_{K_0} = f_0$
2. *if $\tau^* \prec \sigma^*$ has $f(\tau^*) > f(\sigma^*)$, then every face $\tau \prec \sigma^*$ of σ^* satisfies $lex(\tau) \le lex(\tau^*)$, and every coface $\sigma \succ \tau^*$ of τ^* satisfies $lex(\sigma) \ge lex(\sigma^*)$.*

That is, allowed exceptions in the Morse inequalities occur only between σ, τ if τ is the largest lexicographical face of σ, and σ is the smallest lexicographical coface of τ. Put another way, matchings occur along only the "steepest" gradients in the lexicographical ordering induced by f_0. This notion is easy to describe equivalently for gradient vector fields.

▶ **Lemma 2.6** (Naïve Gradient Vector Fields). *If (M, C) is a gradient vector field on K corresponding to a* naïve *discrete Morse function f, then any $(\tau^*, \sigma^*) \in M$ has every face $\tau \prec \sigma^*$ satisfying $lex(\tau) \le lex(\tau^*)$, and every coface $\sigma \succ \tau^*$ satisfying $lex(\sigma) \ge lex(\sigma^*)$.*

Proof. The proof is immediate from (M, C) partitioning K, and the second condition of Definition 2.5. Namely, if $(\tau^*, \sigma^*) \in M$, we know that $f(\sigma^*) < f(\tau^*)$. Due to Definition 2.5, this exception in the second Morse inequality occurs only if τ^* is the largest lexicographical

(a) An example input to MinMM, with f_0. **(b)** The naïve (non-optimal) GVF

Figure 3 A simplicial complex with an accompanying injective function $f_0 : K_0 \to \mathbb{R}$ on the vertices that yields a naïve GVF not minimizing $|C|$.

face of σ^*, and σ^* is the smallest lexicographical coface of τ^* with respect to the given f_0 from which f is induced. ◀

▶ **Observation 2.7.** *Notice that a naïve Morse function often might not be a solution to* MinMM*, and may not minimize $|C|$. For a simple example, see Figure 3. It is possible to even have naïve Morse functions with $O(n)$ critical cells, as occurs in Figure 8.*

In [16], it is shown that given a simplicial complex K and an injective $f_0 : K_0 \to \mathbb{R}$, the naïve discrete Morse function resulting from f_0 has a corresponding gradient vector field (M, C) that is *unique*. Although naïve discrete Morse functions can have $O(n)$ critical cells, they retain a number of desirable properties that are covered in Section 4.

We define naïve discrete Morse functions with inspiration from [20], which extends values on the vertices of K to a discrete Morse function in higher dimensions as a preliminary step to find solutions to MinMM. After naïve Morse functions are computed, they are refined. This procedure is exact for subcomplexes of two-manifolds, but is not exact in general. In practice, often computing the naive Morse function makes up the bulk of computation, and [16] improves the time complexity for this task to $\Theta(dn)$. However, in theory, refining a Morse function is much more difficult than computing a naive one. We provide a formal time complexity analysis of both steps in Appendix C, which require in total $\Theta(d^2n^3)$ time, and are bottlenecked by the refinement step.

We adopt a different framework than [20] 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 $\Theta(n)$. To approximate MinMM in higher dimensions, we adopt a similar framework to [20], generating a naïve Morse function using f_0, and refining it. However, rather than relying on the exact refining methods of [20], we instead lean on efficient methods to compute naive Morse functions similar to those in [16]. For this, we introduce a natural gradient descent heuristic using permutations of the given $f_0 : K_0 \to \mathbb{R}$, which refines a gradient vector field in $\Theta(dn)$ time.

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

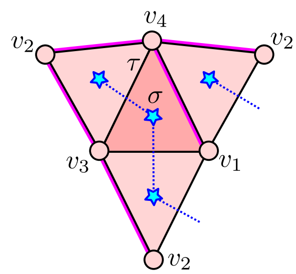


Figure 4 A spanning tree on the vertices of K with edges in pink, and its corresponding restricted dual graph with vertices as blue stars and dotted edges. The restricted dual graph of K excludes edges intersecting the spanning tree.

defined in Section 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. We assume without loss of generality that MorseDual runs on K with only one connected component, since we could just rerun MorseDual on every connected component otherwise. 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 restricted dual graph of K with respect to the edges of T (that is, G^* is the dual graph G, removing any edges that are dual to edges of E_T). For every leaf node $v^* \in V^*$ of G^* (which is a face in K_2), we define a Morse matching (e^*, v^*) whose tail is the unique dual edge e^* of v^* (which is an edge in K_1), and whose head is v^* (which is a face in K_2). 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 in the restricted dual 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 (v, e) between a vertex $v \in K_0$ adjacent to the root $r \in K_0$ of T and the corresponding edge $e = [r, u] \in E_T$. We delete the matched edge, and collapse the other vertex v of e into r. We continue the process, calling any resulting self-looping edges critical, and then deleting them from T. After all edges have been removed from T, we call the sole remaining vertex (which is the root of T) critical. For an example on a triangulated torus, see Figure 5.

We now prove the optimality of MorseDual, demonstrating that the algorithm solves MinMM for two-manifolds. Without loss of generality, assume that for a given complex K, $\beta_0(K) = 1$. If there were more than one connected component, we could simply rerun MorseDual on each component.

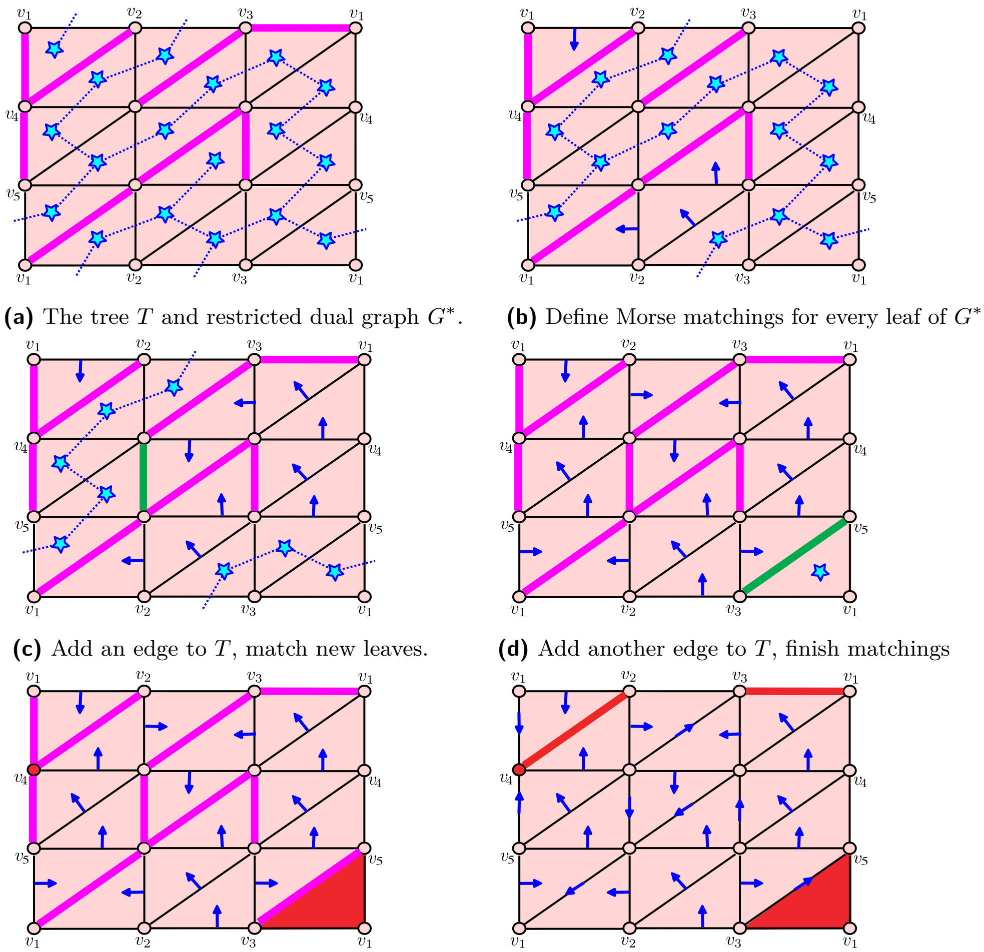
▶ **Lemma 3.1** (MorseDual Recovers $|C_1| = \beta_1$)**.** *MorseDual minimizes the number of critical edges in its output GVF.*

Proof. Recall that K has a tree-cotree partition of three disjoint sets (T, C, X) where T is a spanning tree of the vertices, C is its corresponding cotree, and X is the remaining edges in K_1. We begin by computing T, meaning that our restricted dual graph G^* contains the simplices of C and X. By the end of the algorithm, every matched (τ, σ) where $\sigma \in K_2$ and $\tau \in K_1$ forms a tree. Hence, we can take this set of matchings as C, and X is the set of edges that were added to T. Then we know that we have computed successfully the three sets

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 restricted dual graph G^*.

(b) Define Morse matchings for every leaf of G^*

(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 major steps of MorseDual when run on K as a triangulated torus.

(T, C, X). As together T and C generate one-dimensional homology, the Morse matching in MorseDual attains $|C_1| = \beta_1$. ◀

We comment that similar methods can also compute β_1 in linear time [10, 14, 24].

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

Proof. The proof relies on many of the same invariants as Lemma 3.1. Namely, MorseDual guarantees that its matchings between one-cells and two-cells form a cotree C of T. Consequently, K adds one two-dimensional homology group per connected component in the cotree. Then indeed the cotree C, by collapsing in on itself, recovers exactly $|C_2| = \beta_2$. ◀

▶ **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 linear in the number of simplices in K, and can be accomplished by iterating over every $v \in K_0$ and each of its adjacent edges $e \in K_1$. 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. The restricted dual graph G^* is computed identically, leaving out any edges dual to edges in T. 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 and King's algorithm on the same set on a set of trianguled surfaces, ranging from 12 vertices to roughly 1300. King's implementation uses OpenGL after computing a Morse matching to display results, so we concluded our timer before any OpenGL processes were called in the interest of fairness. We highlight that the theoretical time complexity discussed for MorseDual and for Extract [20] holds in practice as n increases. Experimental time complexity is as follows, all on triangulated surfaces.

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

4 Naïve Discrete Morse Functions to Approximate MinMM

We now examine previously published methods to compute discrete Morse functions from [16, 20], which compute naïve Morse functions on a simplicial complex K and then refine them toward optimal ones. We prove that naïve discrete Morse functions maintain an important

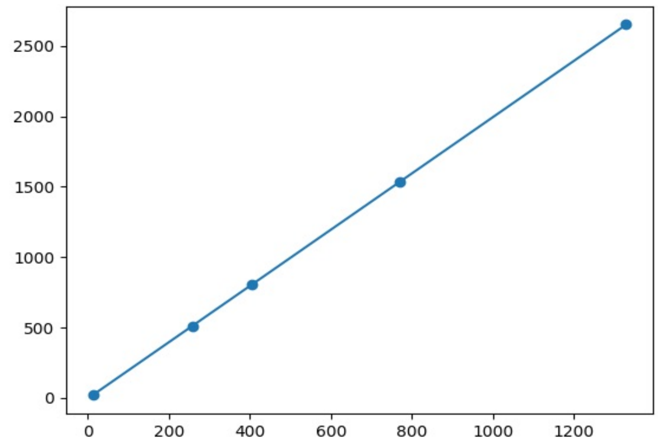


Figure 6 As proof of concept, we find linear asymptotic behavior in practice n grows.

invariant with respect to a given injective $f_0 : K_0 \to \mathbb{R}$. This leads us to a natural gradient descent heuristic to optimize discrete Morse functions by manipulating the given f_0. We verify the practical utility of naïve discrete Morse Functions combined with our gradient descent heuristic, and observe that the two methods combined give discrete Morse functions with very few extraneous critical cells. We give theoretical guarantees on the performance of these heuristics in Section 5 using probability theory, arriving at a randomized additive approximation algorithm for complexes with low relative torsion.

4.1 Properties of Naïve Morse Functions and a New Gradient Descent

Recall Definition 2.5 of naïve discrete Morse functions on K, given an injective $f_0 : K_0 \to \mathbb{R}$, which are outputs to algorithms in [16, 20]. Let (M, C) be the resulting naïve discrete Morse function from f_0 on K. Recall that these Morse functions occur along the "steepest" gradients induced by f_0, so if $(\tau, \sigma) \in M$, then τ is the largest lexicographical face of σ, and σ is the smallest lexicographical coface of τ.

As a result, orderings on the vertices matter for the induced (M, C), 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 avoiding permutations that do not reduce $|C|$ in the naïve GVF induced by f_0 originally.

We characterize a substantial class of permutations of f_0 that attain the same naïve GVF. Denote a permutation as p, and the resulting function values on the vertices after applying p to $f_0(K)$ as $p \circ f_0(K)$. We write id for the identity permutation, which leaves all vertex values unchanged. We write $na\ddot{\imath}veGVF(K, f_0)$ as the naïve (M, C) induced by f_0 on K. Let $v, a, b \in K_0$ such that $a, b \in \overline{\mathrm{star}}_K(v)$. Suppose for the following lemma that $f_0(b)$ is the smallest upper bound of $f_0(v)$ in $\overline{\mathrm{star}}_K(v) \cap K_0$, and $f_0(a)$ is the greatest lower bound of $f_0(v)$ in $\overline{\mathrm{star}}_K(v) \cap K_0$.

▶ **Lemma 4.1** (Plateau). *If we chose a permutation p swapping the value of a single vertex $u \in K_0$ with v where $f_0(u) \in (f_0(a), f_0(b))$, the open interval, then $na\ddot{\imath}veGVF(K, p \circ f_0) =$*

$na\ddot{\imath}veGVF(K, id \circ f_0)$.

Proof. We need to show that local orderings among vertices in $\overline{\text{star}}_K(u) \cap K_0$ and $\overline{\text{star}}_K(v) \cap K_0$ are invariant after the application of p. Without loss of generality, suppose $f_0(v) < f_0(u) < f_0(b)$. (The inequalities are strict, since f_0 is injective.) Since $f_0(b)$ is the least upper bound of $f_0(v)$ in $\overline{\text{star}}_K(v) \cap K_0$, it follows that any $b' \in \overline{\text{star}}_K(v) \cap K_0$ 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 $f_0(v) < f_0(u) \leq f_0(b')$ is identical, substituting u for v. 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, $na\ddot{\imath}veGVF(K, p \circ f_0)$ and $na\ddot{\imath}veGVF(K, id \circ f_0)$ give the same (M, C) due to Lemma 2.6, since naïve gradient vector fields choose matchings by lexicographical order, which is unchanged permuting v with u. ◀

Lemma 4.1 tells us that for any vertex $v \in K_0$, permuting v with another vertex that is “close" to v with respect to f_0 will not reduce the number of critical cells in a naïve Morse matching if the other vertex and v are not adjacent. By only permuting adjacent vertices that are near in f_0 value, we rule out a considerable number of permutations among nonadjacent vertices what would make no difference to (M, C).

Using Lemma 4.1, we obtain a gradient descent algorithm by running the $\Theta(dn)$ algorithm given in [16], and then permuting adjacent vertices that are the nearest above and nearest below in f_0 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 of dimension d, by checking the value of f_0 at every neighbor of a given vertex. At each iteration we keep the permutation decreasing $|C|$ the most. Then, the whole process takes $\Theta(dn)$ time assuming K is sparse, and that there are a constant number of iterations in the gradient descent.

Algorithm 2 MorseGradientDescent

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**
    $B \leftarrow \{u \in \overline{\text{star}}_K(v) \cap K_0 | f_0(u) > f_0(v)\}$
    $A \leftarrow \{u \in \overline{\text{star}}_K(v) \cap K_0 | f_0(u) < f_0(v)\}$
    $a \leftarrow u \in A$ s.t. $f_0(a) = \min_{u \in A}(f_0(u))$
    $b \leftarrow u \in B$ s.t. $f_0(b) = \max_{u \in B}(f_0(u))$
    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
  end if
end while
return $GVF(K)$

▶ **Lemma 4.2** (MorseGradientDescent Update Time Complexity). *When a new permutation of f_0 is chosen by MorseGradientDescent, 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 $f_0 \circ p$ denote the function on the vertices of K after a given permutation p occurs, switching the adjacent vertices $u, v \in K_0$. One can update (M, C) resulting from ExtractRightChild$(K, f_0 \circ p)$ in $\Theta(d)$ time by running ExtractRightChild on only $\mathrm{star}_K(v)$ and $\mathrm{star}_K(u)$. The size of $\mathrm{star}_K(v)$ and $\mathrm{star}_K(u)$ are bounded by the number of edges in $\mathcal{H}$ involving v and u, which is bounded by the number of 1-simplices with which v and u participates in K. Thus, an update could be $O(dn)$ in the worst case if v or u have high degree, as their star could be the entire complex. 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 at all an unreasonable assumption to make, especially in persistent homology related applications. We demonstrate the update times in practice of randomly generated complexes in the next subsection.

4.2 Experimental Data

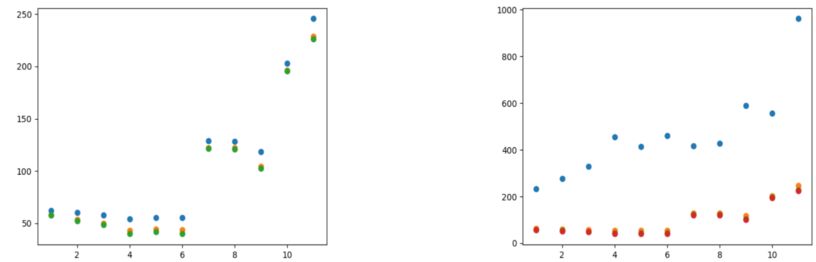
It is important to keep in mind that a primary application of discrete Morse theory is in persistent homology, where DMT can reduce the size and complexity of data in a topologically faithful way. As such, our experimentation in this section is conducted on Vietoris-Rips complexes , which are a central object of study in persistent homology. This subsection discusses our practices when conducting experiments, and the data we used to demonstrate the practical validity of naïve Morse functions and our related heuristic to optimize them.

Our experiments were conducted on data ranging between 4 and 10 dimensions, with the number of total simplices lying between 200 and 2000. For each experiment, we conducted 100 trials, each on a different randomly generated Vietoris-Rips complex. The Vietoris-Rips complexes were all constructed from randomly generated point clouds in $\mathbb{R}^{10}$. Their corresponding injective functions $f_0 : K_0 \to \mathbb{R}$, required as input to ExtMM, were assigned by the index of vertices in the complex. Since each vertex was generated in random order, f_0 can be considered as a random function on the vertices. For anyone interested in our implementation and the data we used, we encourage readers to explore our public repository.

Our findings were striking, and simple: ExtractRightChild coupled with MorseGradientDescent approximates an optimal Morse function within a constant factor when run on realistic data. These findings are summarized in the following subsection.

4.3 Morse Gradient Descent in Practice

On average, in each experiment consisting of 100 trials, we find that MorseGradientDescent approximates an optimal discrete Morse function within a constant factor. This is summarized in Figure 7a and Figure 7b , and addressed in the following table. We note that these results are possible with a low variance for each experiment, and derive our experimental approximation factor from the data.



(a) The proportions of critical simplices **(b)** Comparison to total number of simplices.

5 A Randomized Algorithm for MinMM in Higher Dimensions

We conclude our results in this paper with an interpretation of the strikingly positive experimental results of ExtractRightChild when paired with MorseGradientDescent. In doing so, we arrive at a randomized approximation algorithm for MinMM on complexes with low relative torsion.

5.1 Categorizing Solutions to ExtMM

We begin with a brief explanation of why finding a naïve GVF is much easier than a solution to MinMM, which is both NP-Hard and W[P]-Hard. In particular, this is due to lexicographically maximal faces having many cofaces, and relative torsion in a simplicial complex K.

It is easy to see that naïve Morse functions could have potentially very many extraneous critical cells, depending on the assigned f_0.

▶ Remark 5.1. Consider a complex with many i-simplices $\{\sigma_1, \sigma_2, ...\sigma_k\}$ sharing the same $i-1$ cell τ, where τ has $lex(\tau) > lex(\tau')$ for any other $\tau \neq \tau' \prec \sigma_i$, $\{i \in 1, 2, ...k\}$. In the naïve GVF induced by f_0 it is possible to have $|C| = O(n)$. For a simple example, see Figure 8.

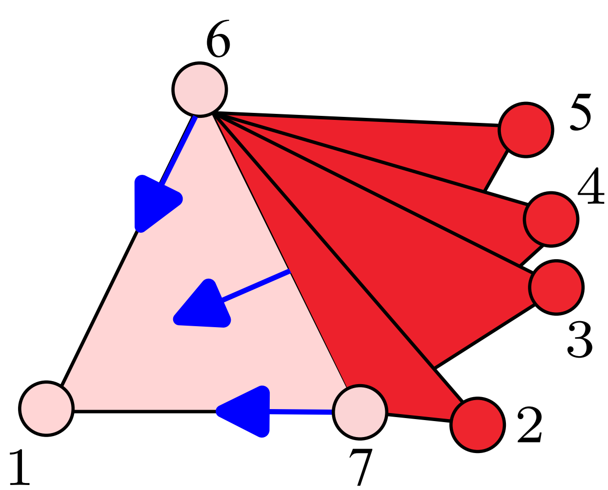


Figure 8 An example two-dimensional complex K where the GVF induced by f_0 has $|C| = O(n)$, with critical cells in red.

▶ Remark 5.2. It is also entirely possible that *no* injective $f_0 : K_0 \to \mathbb{R}$ exists with a naïve GVF producing the same number of critical cells as an output to MinMM. This can be due to torsion in K, and takes places in complexes such as the one in Figure 9.

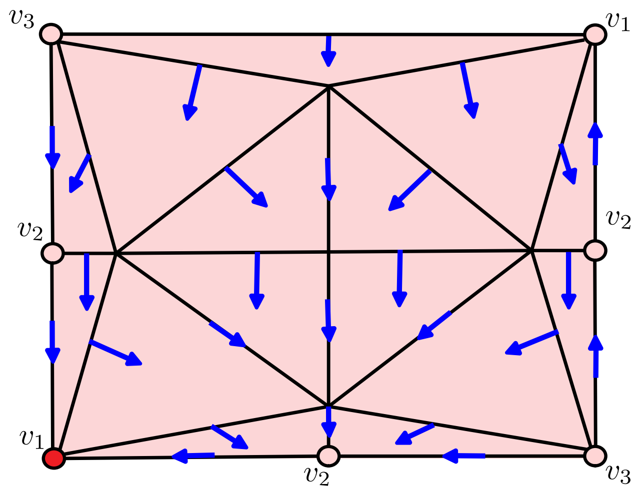


Figure 9 A variant of the triangulated dunce cap and its canonical gradient vector field, with only one critical vertex in red. It is impossible to assign a $f_0 : K_0 \to \mathbb{R}$ to K that is consistent to f_0 and a solution to MinMM, due to the relative torsion in K.

5.2 Statistical Interpretation of MorseGradientDescent

Yet, the experimental performance of ExtractRightChild coupled with MorseGradientDescent seems to contradict hardness results for large and high dimensional complexes. This motivates an inquiry into the probability of the events in Remark 5.1 and Remark 5.2.

We delve first into an explanation of the surprisingly strong performance of ExtractRightChild. We elaborate on the observation in Remark 5.1, giving a bound on the associated error (in terms of the added number of critical simplices) in this situation. As usual, let K denote a simplicial complex, and $f_0 : K_0 \to \mathbb{R}$ an injective function on its vertices. Let (M, C) be the GVF solving ExtMM for K and f_0, and (M', C') be the output of ExtractRightChild(K,f_0).

▶ **Lemma 5.3** ("Error" Introduced by Non-Optimal f_0 for Naïve Morse functions)**.** *Suppose $\sigma, \sigma' \in K_i$ and $\tau \in K_{i-1}$ such that τ is the largest lexicographical face of both σ and σ'. If neither $\sigma \in C$ nor $\sigma' \in C$, then the number of critical cells $|C'|$ found by ExtractRightChild(K,f_0) is $|C'| = |C| + O(2^{d+1})$.*

Proof. If f_0 causes an unnecessary conflict in the largest lexicographical faces of σ with σ', then it may be that σ, and every lower dimensional simplex $s \in \sigma \setminus \{\sigma \cap \sigma'\}$ contained in σ but not in σ' is made critical as a consequence. As ExtractRightChild only pairs the smallest lexicographical i-cells with their largest lexicographical $i-1$ dimensional faces, ExtractRightChild will fail to assign one of σ or σ' to $M^{H'}$. Suppose $\sigma' \notin M^{H'}$, and $\sigma' \in C'$ as a result. Then it is entirely possible that every $\tau' \prec \sigma'$ will be assigned $\tau' \in C'$ as well, since τ' is adjacent to τ so every face of τ shared with τ' will be paired with a lexicographically smaller coface, which necessarily participates in σ (by the assumption that $\sigma \in M^{H'}$ while $\sigma' \in C'$, so $lex(\sigma) < lex(\sigma')$ by the definition of a naïve GVF). The argument repeats, decreasing in dimension until reaching the vertices comprising σ and σ'.

The unique vertex v of σ' not in σ can finally be named critical, after it has failed to match with any one-cell. For a simple example, see Figure 10. Hence, since σ and every lower dimensional simplex comprising σ can be unnecessarily critical, and there are 2^{d+1} simplices in a d-dimensional complex, $|C'| = |C| + O(2^{d+1})$. ◀

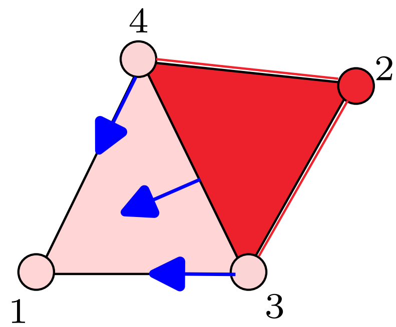


Figure 10 An simple example K with $f_0 : K_0 \to \mathbb{R}$ where ExtractRightChild causes a full simplex to become unnecessarily critical.

Such exceptions can be quite costly when attempting to compute an optimal Morse matching. We demonstrate the relatively low probability of their occurance, justifying the strong experimental results of Algorithm 3.

▶ **Lemma 5.4.** *Let $\#(K_0) = m$ and $d = \dim(K)$. The probability of $\sigma, \sigma' \in K_i$ sharing the same largest lexicographical face $\tau \in K_{i-1}$, as described in Lemma 5.3, is $\prod_{i=0}^{d} \frac{(m/2)-i}{m-i}$.*

Proof. Without loss of generality, we can assume a vertex $v \in K_0$ is assigned $f_0(v) \in \{1, ..., m\}$. Then the expected value of $f_0(v)$ is $m/2$. Moreover, we know that any vertex in the 1-skeleton of K has at most degree d. Extraneous critical cells described above only occur when two i-simplices share their smallest lexicographical face. That is, in the 1-skeleton K_1 of K, extra critical cells occur only when a single vertex v has every adjacent vertex u with $f_0(v) < f_0(u)$, and there exists another $v' \in K_0$ adjacent to every such u such that $f_0(v') < f_0(v)$. Consequently, the probability of choosing $f_0(u_1) > f_0(v)$ is $\frac{(m/2)}{m} = 1/2$. The probability of choosing a subsequent u_2 with $f_0(u_2) > f_0(v)$ is $\frac{(m/2)-1}{m-1}$, and choosing the kth u_k with $f_0(u_k) > f_0(v)$ is $\frac{(m/2)-k}{m-k}$. Then, all together the probability of achieving the situation described above is $\prod_{i=0}^{d} \frac{(m/2)-i}{m-i}$, since the probability of making the first choice $P(f_0(u_1) > f_0(v)) = \frac{1}{2}$, the probability of making the final choice $P(f_0(u_d) > f_0(v)) = \frac{(m/2)-(d-1)}{m-(d-1)}$, and the probability of choosing $f_0(v') < f_0(v)$ is $\frac{(m/2)-d}{m-d}$ if d values above $f_0(v)$ have been already assigned. ◀

▶ **Theorem 5.5** (Expected Error of Naïve Morse functions)**.** *Given an optimal Morse matching (M, C) on K and a naïve Morse matching (M', C') induced by a given f_0, the expected value of $|C'|$ is $E(|C'|) = |C| + 2^{d+1} * \prod_{i=0}^{d} \frac{(m/2)-i}{m-i}$*

Proof. It follows from Lemma 5.3 that a conflict in the largest lexicographical face τ of two cofaces σ, σ' causes on the order of 2^{d+1} extraneous critical cells. From Lemma 5.4 we know that the probability of this occuring is $\prod_{i=0}^{d} \frac{(m/2)-i}{m-i}$. Put together, we obtain the expected value. ◀

▶ Remark 5.6. This means, that for a complex of dimension 8 with 100 vertices, we should find the probability of any exceptions to be $\frac{301}{228532}$, which is roughly consistent with the number of exceptions we found in practice.

▶ Remark 5.7. It should be noted that the results in Section 5.2 assume that we can be given K and some injective $f_0 : K_0 \to \mathbb{R}$, and using MorseGradientDescent, we can conduct permutations to attain an injective f_0 whose naïve GVF is a solution to MinMM. This is not necessarily possible, failing in cases where K has torsion. Thus, the above results are only valid on complexes where an injective $f_0^* : K_0 \to \mathbb{R}$ even *exists* such that $na\ddot{\imath}veGVF(K, f_0^*) = \text{MinMM}(K)$, and the results do not hold for K with torsion.

6 Discussion

This paper proposes a linear time algorithm solving MinMM for two-manifolds, which is the first improvement since 2005 on the previously published methods to do so in [20]. It is difficult to imagine the existence of a faster algorithm, as one would expect that every simplex would need to be visited at least once to construct a gradient vector field. As a consequence, we add new insight into MinMM from a fine-grained perspective. Moreover, our results put MinMM in an interesting class of problems by demonstrating a sharp delineation in hardness based on a subtle difference in the requirements of inputs to MinMM. If a simplicial complex K is a subcomplex of a two-manifold, the problem becomes extremely easy. Yet, if a *tiny* adjustment is made to the allowed class of input complexes, and K is allowed to be any arbitrary 2d complex, MinMM becomes NP-Hard and even W[P]-Hard.

Secondly, this paper gives a gradient descent heuristic based on solutions in [16] to the subproblem ExtMM of MinMM, which is first introduced in [20]. Our heuristic performs extraordinarily well in practice, despite the inapproximability of MinMM. Extensions to this work abound. Perhaps most obviously, the question emerges to explain these surprisingly strong experimental results. In addition, these heuristics beg to be integrated in application areas requiring the efficient computation of homology, especially in the persistent homology literature. Our hope is that these results will have a sizable impact on the viability of computational techniques in Morse theory, and by extension, we hope that computational topology broadly will become more ubiquitous as a consequence.

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

A Additional Background in Discrete Morse Theory

We provide some additional clarity on fundamental topics in discrete Morse theory discussed only briefly in earlier sections of the paper. We begin with a more detailed description of the Hasse diagram and combinatorial discrete Morse functions.

A.1 The Hasse Diagram

Combinatorially, a simplicial complex K can be represented by a graph called the Hasse diagram, which we denote $\mathcal{H}$. In $\mathcal{H}$, each node represents a simplex in K, and edges exist between $\tau \in K_{i-1}$ and $\sigma \in K_i$ if $\tau \prec \sigma$. That is, $\mathcal{H}$ denotes face/coface pairs in K. Note that the set of faces of σ can be computed in $\Theta(\dim(\sigma))$ time using $\mathcal{H}$, while finding the set of co-faces of σ may take $O(n)$ time to compute using $\mathcal{H}$. This is due to the simple fact that any $\sigma \in K$ can have up to d faces, and can participate in $O(n)$ cofaces.

We add direction to the edges of $\mathcal{H}$ when describing a discrete Morse function f on K. Typically, we abuse notation and just refer to $\mathcal{H}$ as the combinatorial discrete Morse function on K. Every edge of $\mathcal{H}$ directs down in dimension unless an edge corresponds to an exception in the Morse inequalities (i.e. $f(\tau) > f(\sigma)$ for $\tau \prec \sigma$); see Figure 1b. If $\mathcal{H}$ has not been assigned a discrete Morse function, we can assume $\mathcal{H}$ indicates the trivial Morse function where all edges direct down in dimension, assigning every simplex as critical. More formally:

▶ **Lemma A.1** (Combinatorial Discrete Morse Function). *Let K be a simplicial complex. If (M, C) is a GVF over K and $(\tau, \sigma) \in M$, then the edge $[\tau, \sigma]$ of $\mathcal{H}$ directs up in dimension. Otherwise every edge directs down in dimension. The resulting directed graph $\mathcal{H}$ is a discrete Morse function if and only if there are no directed cycles [17, 23].*

B Prior Algorithms Approximating ExtMM and MinMM

In this section we discuss other algorithms in the literature using naïve Morse functions to approximate MinMM. Namely, we discuss the primary algorithms in [16,20], which each take as input a simplicial complex K and an injective function $f_0 : K_0 \to \mathbb{R}$, returning a discrete Morse function f on K. The primary algorithm of [20] is called Extract. In Extract, a subroutine first approximates MinMM by computing a naïve Morse function induced by f_0 as specified in Definition 2.5. Then naïve Morse functions (with $O(n)$ critical cells) are refined. In [16], the time complexity to generate naïve discrete Morse functions is improved to $\Theta(dn)$. Here, we first present the algorithm ExtractRightChild from [16] to generate naïve Morse functions, and then give the full algorithm Extract from [20] which gives a better approximation of MinMM.

B.1 ExtractRightChild

Below, we provide Algorithm 2 of [16] with one modification: instead of keeping track of heads and tails (H and T of [16]) as separate sets on the GVF, we simply maintain a set of head-tail pairs (denoted M here and as m in [16, Algorithm 2]).

Note that ExtractRightChild uses so-called "left-right parents", referring to the pairs (τ, σ) where τ is the largest lexicographical face of σ, and σ is the smallest lexicographical coface of τ. In [16], it is shown that the output of ExtractRightChild is a naïve discrete Morse function by our definition. Additionally, ExtractRightChild uses a "decorated" Hasse diagram, which for every $\sigma \in K$, saves $\rho(\sigma) \prec \sigma$ as the largest lexicographical face of σ. This allows ExtractRightChild to achieve $\Theta(dn)$ time complexity, rather than relying on sorting. In [20], a subroutine of Extract called ExtractRaw is used to compute a "raw" gradient vector field. We conclude discussion of ExtractRightChild with a lemma establishing the equality of ExtractRightChild and ExtractRaw, proven in [16]:

▶ **Lemma B.1** (ExtractRightChild and ExtractRaw). *Let K be a simplicial complex and let $f_0 : K_0 \to \mathbb{R}$ be an injective function. Then ExtractRaw(K, f_0) and ExtractRightChild(K, f_0) yield identical outputs.*

B.2 Extract

Knowing that ExtractRaw and ExtractRightChild can be used interchangeably, we present the algorithm Extract, which calls ExtractRaw. We also present the other subroutine ExtractCancel of Extract, which is used to refine naïve gradient vector

Algorithm 3 ExtractRightChild

Input: simp. complx. K, injective fcn. $f_0 : K_0 \to \mathbb{R}$
Output: a GVF consistent with f_0

$\mathcal{H}^* \leftarrow$ decorate the Hasse diagram of K ▷ see [16, Lemma 4]
$M \leftarrow \emptyset$ ▷ Initialize matching
$C \leftarrow \emptyset$ ▷ Initialize set of critical cells
for $i = \dim(K)$ to 1 **do**
    for $\sigma \in \mathcal{H}^*_i$ **do**
        if σ is assigned **then**
            continue
        end if
        if σ is a left-right parent **then**
            Add $(\rho(\sigma), \sigma)$ to M
            Mark σ and $\rho(\sigma)$ as assigned
        else
            Add σ to C
            Mark σ as assigned
        end if
    end for
end for
Add any unassigned zero-simplices to C
return (M, C)

Algorithm 4 [20] Extract

Input: simp. complx. K, injective fcn. $f_0 : K_0 \to \mathbb{R}$
Output: A GVF on K

$(M, C) \leftarrow$ ExtractRightChild(K, f_0)
for $j \in \{1, ..., \dim(K)\}$ **do**
    ExtractCancel$(K, f_0, p, j, (M, C))$
end for

fields. For subcomplexes of two-manifolds, this refinement step yields an optimal Morse matching, solving MinMM.

In the original version of Extract given in [20], Line 1 calls ExtractRaw rather than ExtractRightChild, but for the sake of simplicity we only include ExtractRightChild, as the two function the same. Lastly, we include ExtractCancel, which is called subsequently in Extract. This subroutine is used to refine outputs of ExtractRightChild toward solutions to MinMM, "cancelling" extraneous critical cells by reversing gradient paths. ExtractCancel incorporates a persistence parameter p to cancel critical cells up to a desired threshold. Setting $p = \infty$ cancels the greatest possible number of critical cells, minimizing $|C|$ if K is a subcomplex of a two-manifold.

Algorithm 5 [20] ExtractCancel

Input: simplicial complex K, injective function $f_0 \colon K_0 \to \mathbb{R}$, $p \geq 0$, $j \in \mathbb{N}$, and a GVF (M, C) on K
Output: Gradient vector field (M, C) on K

```
1:  for all σ ∈ C_j do
2:      Find all gradient paths γ_i = σ → τ_i1 → σ_i1 → τ_i2 → σ_i2 → ... → τ_ik ∈ C_{j−1} such
        that max_{v∈τ_ik} f_0(v) > max_{u∈σ} f_0(u) − p
3:      for all γ_i do
4:          if τ_ik is not the final simplex in C_{j−1} of any other gradient path γ_h, h ≠ i then
5:              m_i ← max_{v∈τ_ik} f_0(v)
6:          end if
7:          if There exists any defined m_i then
8:              Choose h satisfying m_h = min m_i
9:          end if
10:     end for
11:     Reverse the gradient path γ_h in H, and update M accordingly
12: end for
13: return (M, C)
```

Gradient paths in this context flow from a critical i cell to a critical $i-1$ cell, and are often easiest to interpret via $\mathcal{H}$. Note that Line 11 is given as a separate algorithm in [20], but we simply specify the full subroutine in our interpretation. When reversing a gradient path γ_i, arrows in $\mathcal{H}$ are reversed. This means that σ is updated such that $\sigma \notin C, \sigma \in M^H$, and $\tau_{ik} \notin C, \tau_{ik} \in M^T$, and any other matchings resulting from the reverse of γ_i are updated in (M, C) equivalently.

C Time Complexity Analysis of Prior Algorithms

We now analyze the time complexity of algorithms introduced in [16, 20], comparing them to the methods in this paper.

In [16] it is shown that ExtractRightChild computes a naïve discrete Morse function in $\Theta(dn)$ time. It is easy to see that when computing a randomized approximation of MinMM, pairing ExtractRightChild and MorseGradientDescent also requires $\Theta(dn)$ time assuming K is sparse and MorseGradientDescent terminates with a constant number of iterations.

Furthermore, it is easy to see that ExtractCancel, as it is presented in [20], has $\Theta(dn^3)$ time complexity.

▶ **Lemma C.1** (Runtime of ExtractCancel). *Algorithm 5 terminates in* $\Theta(dn^3)$ *time, using* $\Theta(n)$ *space.*

Proof. ExtractCancel iterates over all dimension j critical cells in Line 1, and it is entirely possible that $|C|_j = O(n)$. (For a simple example, see Remark 5.1 and Figure 8). Then, Line 2 finds all gradient paths flowing out of a critical j-cell $\sigma \in C_j$. Indeed, we have $\Theta(dn)$ total gradient paths (with one unique gradient path per edge in $\mathcal{H}$ between dimension j and dimension $j-1$), and each gradient path could contain $O(n)$ simplices. Hence, Line 2 is $\Theta(dn^2)$. All together, ExtractCancel terminates in $\Theta(dn^3)$ time, and since only a constant number of copies of every $\sigma \in K$ need to be saved in the process, uses $\Theta(n)$ space. ◀

▶ Remark C.2 (Memoization in ExtractCancel). Using simple memoization, the runtime of ExtractCancel could be improved to $\Theta(dn^2)$ with no increase in space. This is slightly nontrivial, and would require marking every cell that has been visited previously when discovering gradient paths.

▶ **Theorem C.3** (Runtime of Extract). *Algorithm 4 terminates in* $\Theta(d^2n^3)$ *time, using* $\Theta(n)$ *space.*

Proof. Recall Extract only calls ExtractRightChild once, and then calls ExtractCancel in each dimension. From [16], we know that ExtractRightChild terminates in $\Theta(dn)$ time using $\Theta(n)$ space. From Lemma C.1 we know that ExtractCancelterminates in $\Theta(dn^3)$ time per dimension. As ExtractCancel is called for every dimension of K, this adds another factor of d to the runtime. All together, then Extract terminates in $\Theta(d^2n^3)$ time, if implemented as specified in [20] without memoization. As ExtractRightChild and ExtractCancel each use $\Theta(n)$ space, the same holds for Extract. ◀

When compared with the combination of ExtractRightChild and MorseGradientDescent, the improvements in time complexity are clear. We believe this warrants the use of ExtractRightChild and MorseGradientDescent in most practical settings, rather than existing algorithms.