On Computing Discrete Morse Functions

Anonymous author
Anonymous affiliation

Anonymous author
Anonymous affiliation

Anonymous author
Anonymous affiliation

Anonymous author
Anonymous affiliation

Anonymous author
Anonymous affiliation

Abstract

Discrete Morse theory provides a way of studying simplicial complexes akin to studying flows over smooth surfaces. Discrete Morse functions assign a real 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. We consider an optimization problem: MinMM, which is to find a function over a given simplicial complex K that minimizes the number of critical simplices. We study this problem through the lens of King et al. (2005), which extends a Morse matching to K given an injective function $f_0 : K_0 \to \mathbb{R}$ on the vertices of K. Though it has been shown that MinMM is NP-hard and W[P]-Hard to approximate, we give a linear time algorithm for the restricted case where the input is a triangulation of a two-manifold, improving $\Theta(n^3)$ algorithms of King et al. published 18 years ago. We implement our algorithm to demonstrate its improvements in practice. We present a linear time gradient descent heuristic that approximates MinMM well in practice in higher dimensions. In doing so, we arrive at a randomized algorithm approximating a realistic restriction of MinMM within a constant additive factor.

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 classical Morse theory, continuous functions are assigned to smooth manifolds in order to study their topology [28]. For example, the Betti numbers of a manifold can be computed by examining critical points of the continuous functions. In [19], Forman defines analogous tools in the discrete setting; leading to the field of *discrete Morse theory*. Discrete Morse theory has been fruitful when paired with persistent homology [2, 3, 8, 9, 13, 14, 18, 25], where it is often used to reduce the size and complexity of data in a topologically faithful manner. In this work, we study discrete Morse functions on simplicial complexes. 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 cells of a simplicial complex to $\mathbb{R}$, subject to specific inequalities as cells increase in dimension. Combinatorially, a discrete Morse function is a constrained matching in the Hasse diagram of the complex, where unmatched cells are called critical. Topologically, a Morse function takes the form of

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:25
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

a gradient vector field on a simplicial complex. These gradient vector fields are composed of matchings between cells such that the collapse of any given matching does not alter the homotopy type of the complex.

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, there is interest in algorithms that minimize the number of critical simplices in a Morse function on a complex [6, 15, 21, 22, 24, 27]. Doing so gives not only the Betti numbers, but more importantly the *exact instructions* of how to collapse paired simplices until only critical simplices remain. The problem of minimizing the number of critical simplices is known in the literature as "Minimum Morse Matching", or MinMM. Joswig and Pfetsch established that MinMM is NP-hard [24]. Moreover, recent work has demonstrated the inapproximability of generating discrete Morse functions on complexes that are not subcomplexes of two manifolds. For K of dimension greater than two, MinMM is NP-Hard to approximate within a factor of $O(n^{1-\varepsilon})$, and if K has dimension two, MinMM can not be approximated within a factor of $2^{\log^{(1-\varepsilon)} n}$ for any $\varepsilon > 0$ [4, 5].

Nonetheless, in [25] King et al. proposed the problem of constructing a discrete Morse function given an injective function $f_0 : K_0 \to \mathbb{R}$ on the vertices of K. King et al. uses an algorithm called Extract to extend vertex data into higher dimensions, computing a (potentially non-optimal) discrete Morse function in polynomial time. If the complex is a subcomplex of a two-manifold, Extract solves MinMM for surfaces. The algorithm Extract is cubic with respect to the number of simplices in a complex. Consequently, there is a gap between inapproximability results in high dimensions and polynomial time methods for low dimensions, motivating two questions:

1. Can we improve the run time of MinMM on triangulated two-manifolds given an injective real valued function on the vertices?
2. Are there classes of complexes where solutions to MinMM are relatively easy to approximate in high dimensions? Can considering vertex data on K add insight?

To address the first question, we present an $\Theta(n)$ algorithm based on algorithms to compute homology generators on surfaces [1, 7, 10, 11, 16, 17, 31]. For the second question, we build from ideas first developed by King et al. in [25]. We present a randomized heuristic to solve MinMM in high dimensions. We give a constant additive approximation bound on the number of critical simplices produced by our algorithm. We also verify the performance of our algorithm experimentally.

This paper is organized as follows. In Section 2, we introduce the definitions and prior work in discrete Morse theory. 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, improving the $\Theta(n^3)$ algorithm of [25]. In Section 4, we give a gradient descent algorithm to approximate MinMM and we verify the performance of our algorithm experimentally. Finally, in Section 5, we present a randomized additive approximation algorithm and give theoretical guarantees about its performance.

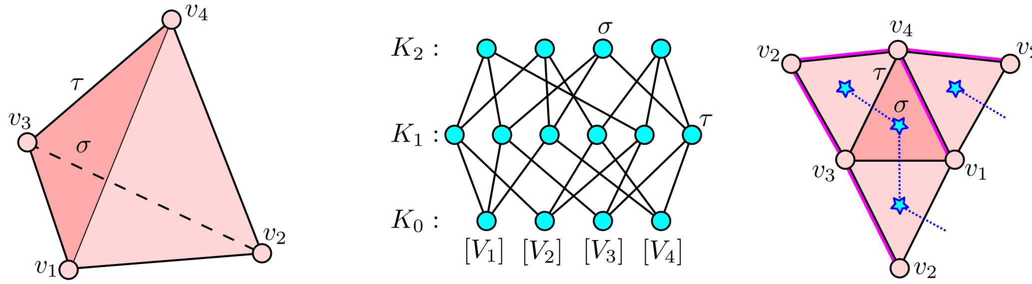
2 Preliminaries

In this section, we provide the definitions and notation used throughout the paper. We provide essential definitions here and additional definitions from computational topology in Section A.3, but for a general survey of discrete Morse theory, see [26, 30].

2.1 Simplicial Complexes and Two-Manifolds

Let K be an abstract simplicial complex with n simplices. For a simplex $\sigma \in K$, we denote the dimension of σ as $\dim(\sigma)$ and we define $d = \dim(K)$ to be the maximum dimension of any simplex in K (in which case, we call K a d-simplex). Thinking of a simplex as a set of vertices, we write $v \in \sigma$ if v is a vertex of σ. 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 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. Often, it is useful to discuss *adjacent* vertices on the one-skeleton of K. For a vertex $v \in K_0$, the set of vertices adjacent to v (sharing an edge with v on the one-skeleton of K), is denoted $N(v)$.

We often study simplicial complexes combinatorially through their Hasse diagram, and thinking of simplicial complexes in this lens is helpful algorithmically. The Hasse diagram $\mathcal{H}$ for K is a graph whose vertices correspond to the simplices of K, two simplices τ and σ are connected by an edge if τ is a codimension one face of σ. See Figure 1 for an example of a simplicial complex that is a triangulated sphere, and its corresponding Hasse diagram.



(a) A Two-Complex K **(b)** The Hasse diagram **(c)** Tree-cotree decomposition

Figure 1 A simplicial complex K, along with its Hasse diagram, and its tree-cotree partition. In (a), K is a triangulation of the sphere using four two-simplices, six edges, and four vertices. Note $\sigma \in K_2$ and $\tau \in K_1$. Each k-simplex is a vertex in the Hasse diagram for K. We arrange the vertices in rows corresponding to their dimensions; so, the two-simplex σ is on the top row and the one simplices the middle row, and the zero-simplices the bottom row. Since $\tau \prec \sigma$ and their dimensions differ by one (i.e., they are codimension-one), an edge exists between τ and σ. In the tree-cotree partition, the tree T is highlighted in pink, the cotree R has blue stars as vertices and dashed lines as edges, and the tree-cotree partition is $(T, R, \emptyset)$.

If K is a two-manifold, the dual graph G of K is a graph whose vertices represent the two-simplices of K_2, and edges represent two two-simplices that share a common codimension-one face (i.e., two triangles that share an edge in K). Givien a spanning tree T of the one-skeleton of K, the *restricted dual graph D of K with respect to T* is the dual graph of G removing all edges whose duals are edges in T. Letting R be a spanning tree of D, let $X = D \setminus R$ we obtain a *tree-cotree partition* of K, (T, R, X). The sets of edges T, R and X partition K_1; see [16, 17, 31]. Tree-cotree partitions have been extended to a number of related algorithmic results [1, 7, 10, 11] and efficient data structures [16].

2.2 Discrete Morse Theory

We consider three views of discrete Morse functions; switching among these views reveals useful properties. Here, we present the three equivalent definitions of a discrete Morse function, which are used interchangeably throughout this paper.

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

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

As Scoville intuits in [30, p. 49], "the function generally increases as you increase the dimension of the simplices. But we allow at most one exception per simplex." Let τ and σ be one such pair that realizes one of the exceptions. Then, we call (τ, σ) a *matched pair*, with σ the *head* and τ the *tail*. In [20], Forman proves that each simplex in M can be a member of at most one matched pair. A simplex $\sigma \in K$ is called *critical* if it is not a member of a matched pair (that is, if every face of σ has an algebraic function value less than or equal to $f(\sigma)$, or every coface of σ has a value larger than or equal to $f(\sigma)$). See Figure 2(b) for an example.

The matching that follows from the algebraic definition leads naturally to the *topological* notion of a discrete Morse function. In a topological Morse function, function values are ignored and just the matching information is retained. Visually, we can draw the matchings as an arrow from the lower-dimensional simplex in a matching to the higher-dimensional one. See Figure 2(c). 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 as the set of heads in M. Since each simplex is in at most one matched pair, the sets M^T, M^H, and C partitions the simplices of K. 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. This topological definition is equivalent to Definition 2.1 in the following sense:

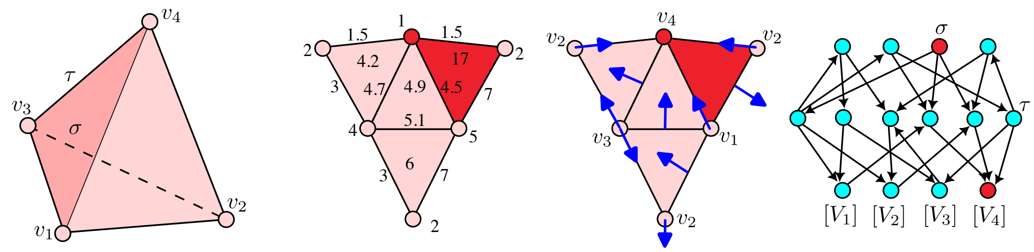
▶ **Lemma 2.2** (Topological Morse Functions). *Let f and g be two algebraic discrete Morse functions, denoting the GVF induced by f as (M_f, C_f) and the GVF induced by g as (M_g, C_g). If f and g induce the same total order on the simplices of K, then $(M_f, C_f) = (M_g, C_g)$.*

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

A GVF on K also gives a combinatorial representation by decorating the Hasse diagram $\mathcal{H}$ of K with directions. Edges in $\mathcal{H}$ are directed up (from the vertex representing the lower-dimensional simplex to vertex representing the higher-dimensional simplex) if their vertices correspond to a matching. Otherwise, edges in $\mathcal{H}$ are directed down. Then, by construction, 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 $\sigma \in C$. If $\sigma \in K_d$ is of dimension $\dim(K)$, only the first property is needed to assign $\sigma \in C$. In other words, σ is unmatched. See Figure 2(d) for an example, and Appendix A.1 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,



(a) A two-complex. **(b)** Algebraic. **(c)** Topological. **(d)** Combinatorial.

Figure 2 The algebraic, topological, and combinatorial interpretations of a Morse function respectively on K as a triangulated sphere. Critical simplices are red.

▶ **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, $|C|$.*

▶ Remark 2.4 (MinMM Hardness of Approximation [4]). Along with being NP-Hard, Bauer et al. show that MinMM is W[P]-hard to approximate, with respect to solution size.

We study methods introduced first in [25] that uses a given injective function $f_0 : K_0 \to \mathbb{R}$ on the vertices to compute solutions to MinMM. In both [18, 25], a discrete Morse function is obtained by extending f_0 into higher dimensions [18]; for details see Appendix B. In [25], this extension is then refined toward a solution to MinMM. A key insight from [18] is 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'; see Definition A.2.

We remark that due to the injectivity of f_0, $lex(\sigma) = lex(\sigma')$ if and only if $\sigma = \sigma'$. This paper builds on the techniques of Algorithm 4, ExtractRightChild, from [18], which uses lexicographical orderings induced by f_0 to compute discrete Morse functions as follows:

▶ **Definition 2.5** (Outputs to ExtractRightChild). *Given a simplicial complex K and an injective function $f_0 : K_0 \to \mathbb{R}$ on the vertices of K, ExtractRightChild produces a discrete Morse function $f : K \to \mathbb{R}$ satisfying:*

1. $f|_{K_0} = f_0$
2. *if (τ^*, σ^*) is a matched pair, then every face $\tau \prec \sigma^*$ of σ^* satisfies $lex(\tau) \leq lex(\tau^*)$, and every coface $\sigma \succ \tau^*$ of τ^* satisfies $lex(\sigma) \geq lex(\sigma^*)$.*

That is, allowed exceptions in the Morse inequalities only occur between (τ, σ) 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, as is done formally in Lemma A.3.

▶ **Observation 2.6.** *Notice that an output to ExtractRightChildneed not be a solution to MinMM. For a simple example, see Figure 3. In fact, it is possible to even have ExtractRightChild(K,f_0) producing $O(n)$ critical cells, as occurs in Figure 8.*

In [18], it is shown that given a simplicial complex K and an injective $f_0 : K_0 \to \mathbb{R}$, the discrete Morse function output by ExtractRightChild(K,f_0) has a corresponding gradient vector field (M, C) that is *unique*. Although such a (M, C) can have $\Theta(n)$ critical cells, outputs to ExtractRightChild retain a number of desirable properties that are covered in Section 4. In practice, often computing the initial approximate Morse function makes up

(a) An example input to MinMM, with f_0. **(b)** ExtractRightChild(K,f_0) output.

Figure 3 A simplicial complex K with an accompanying injective function $f_0 : K_0 \to \mathbb{R}$ on the vertices where ExtractRightChild(K,f_0) does not minimize $|C|$.

the bulk of computation, and [18] accomplishes this task in $\Theta(dn)$ time. However, in theory, refining a Morse function is much more difficult than computing an initial approximation. 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.

3 An Algorithm for MinMM for Two-Manifolds

In this section, we give a $\Theta(n)$-time algorithm solving MinMM for two-manifolds with n vertices. This reduces the time complexity to compute discrete Morse functions on two-manifolds from $\Theta(n^3)$ in King et al., doing so without leaning on a given injective $f_0 : K_0 \to \mathbb{R}$. Our algorithm relies on invariants of spanning trees and their cotrees on triangulations, which are defined in Section 2.1. For an example, see Figure 1c. We also provide a C++ implementation of our algorithm, and experimentally validate our runtime in practice in Appendix D.

A description of the algorithm follows: first, we compute $T = (K_0, E_T)$, a spanning tree of the one-skeleton of K (where $E_T \subseteq K_1$). Then, let $G^* = (V^*, E^*)$ be the complimentary 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 in E_T). We begin with T, and define a Morse matching (v, e) between a leaf vertex $v \in K_0$ the corresponding unique edge $e = [v, u] \in E_T$ adjacent to some other $u \in K_0$. We mark the matched edge, and then subsequently match any unmarked leaves of T in the same manner, marking them afterward. The process continues until all of T has been marked, except for the root vertex $r \in K_0$, which we assign critical. We then proceed to find Morse matchings on G^*. For every unmarked leaf node $v^* \in V^*$ of G^* (which is a two-cell in K_2), we define a Morse matching (e^*, v^*) whose tail is the unique edge $e^* \in E^*$ of v^* (which is an edge in K_1), and whose head is the two-cell given by v^*. When no additional leaves are available, we mark an untouched edge of E^* from G^*, assigning its dual edge as critical. We repeat the process, adding Morse matchings along leaves in the complimentary dual and then removing edges of E^* until $G*$ has no remaining edges. We assign every remaining $v \in V^*$ to be a critical two-cell. For an example of the full execution of the algorithm on a triangulated torus, see Figure 6.

We now prove the optimality of MorseDual, demonstrating that the algorithm solves MinMM for two-manifolds.

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

Algorithm 1 MorseDual

Input: K, a triangulation of a two-manifold
Output: a GVF over K minimizing C over all GVFs over K
1: Compute a spanning tree $T = (K_0, E_T)$ of K
2: Compute the restricted dual graph $G^* = (K_2^*, E^* := K_1^* \setminus E_T)$
3: $C \leftarrow \emptyset$ ▷ critical cells
4: $M \leftarrow \emptyset$ ▷ matched cells
5: For each cell in K, add an attribute 'marked' and set it to False
6: Let T' denote the sub-tree of T that is unmarked (stored implicitly)
7: **while** $\exists$ unmarked leaf node v in T' **do** ▷ match cells of T
8:     Let e be the edge that connects v to the rest of T'.
9:     Mark e and v
10:     Add (v, e) to M
11: **end while**
12: $v \leftarrow$ unmarked vertex of K_0
13: Add v to C.
14: Let G' denote the sub-graph of G^* whose vertices/edges are unmarked in K.
15: **while** $\exists$ unmarked cells of K **do**
16:     **while** $\exists$ unmarked degree-one vertex v^* in G' **do**
17:         Let e^* be the edge that connects v^* to the rest of G'.
18:         Let (e, f) be the dual to (e^*, v^*)
19:         Mark e and f
20:         Add (e, f) to M
21:     **end while**
22:     **if** $\exists$ unmarked edge $e^* \in E^*$ **then** ▷ e^* must be in a cycle
23:         Mark e^*
24:         Add e^* to C
25:     **end if**
26:     **if** A single unmarked $f \in F$ remains **then**
27:         Add f to C
28:     **end if**
29: **end while**

Proof. Let (T, R, X) be a tree-cotree decomposition produced by the algorithm, then $|C_1| = |X|$. Lemma 2 in [16], shows that the loops $\{(T, e) | e \in X\}$ are the fundamental cycles of the surface. By Theorem 2A.1 in [23], $H_1(K, \mathbb{Z}_2)$ is the abelianization of the fundamental group and $\beta_1 = |X|$. ◀

Similar methods also compute the generators of the fundamental group in linear time [11, 31].

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

Proof. In Line 27, a single face is marked as critical and $|C_2| = 1$. By Poincaé duality (Corollary 65.5 of [29]), $H_2(K, \mathbb{Z}_2) \cong \mathbb{Z}_2$, regardless of orientability, and $\beta_2 = 1$. ◀

▶ **Theorem 3.3** (MorseDual is $\Theta(n)$)**.** *MorseDual runs 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 one-skeleton of K takes $\Theta(n_0 + n_1)$ time using breadth first search.

Computing the dual graph G of K is also simple to do in linear time when considering that, since K is a manifold without boundary, each $\sigma \in K_2$ has three adjacent faces and hence G is computed in $O(3n)$ 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. matching the $\sigma \in K_2$), each two-cell is only touched once. Finally, when all two-simplices have been assigned to either M^H or C, 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 ever be saved. ◀

Executing in $\Theta(n)$ time, MorseDual provides a sizable improvement over existing algorithms solving MinMM for two-manifolds with cubic time complexity from [25]. We give these algorithms, and a discussion of their time complexity, in Appendix C. For brevity, we also include the experimental results from running our C++ implementation in Appendix D.

4 Approximating MinMM in Higher Dimensions

We now examine previously published methods to compute discrete Morse functions from [18, 25], which first compute naïve approximations of MinMM, and then refine these approximations toward a solution to MinMM. Given an injective $f_0 : K_0 \to \mathbb{R}$ on the vertices of K, we prove that outputs to ExtractRightChild(K,f_0) maintain an important invariant with respect to a given injective $f_0 : K_0 \to \mathbb{R}$. This leads us to a natural gradient descent heuristic to minimize the number of critical cells by manipulating f_0. We observe that ExtractRightChild combined with our gradient descent heuristic gives discrete Morse functions with very few extraneous critical cells in practice. 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 where there *exists* an f_0^* such that ExtractRightChild(K, f_0^*) is a solution to MinMM.

4.1 Properties of ExtractRightChild and a New Gradient Descent

Let (M, C) be the resulting Morse function from ExtractRightChild(K,f_0), and recall from Definition 2.5 that (M, C) consists only of the "steepest gradients" induced by f_0. That is, M must only have (τ, σ) pairs such that τ is the largest lexicographical face of σ, and σ is the smallest lexicographical coface of τ.

As a result, permuting vertex values carefully can lead ExtractRightChild to compute 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. Yet, trying a permutation of every pair of vertices should obviously be avoided in the interest of efficiency. This motivates the question, can we rigorously categorize the permutations changing nothing in order to avoid them? That is, can we refine the search space of all vertex pair permutations of f_0 by avoiding the ones not reducing $|C|$ in the output of ExtractRightChild?

We characterize permutations of f_0 that attain the same output to ExtractRightChild, and define a *permutation* in our context as a bijection $p : \mathrm{Im}(f_0) \to \mathrm{Im}(f_0)$, and the resulting function values on the vertices after applying p to $f_0(K)$ as $p \circ f_0(K)$. We only consider permutations of vertex pairs, meaning that p is always the identity, except that for two vertices $u, v \in K_0$ the f_0 values flip: $p \circ f_0(u) = f_0(v)$ and $p \circ f_0(v) = f_0(u)$. When referencing the identity permutation leaving all vertex values unchanged, we simply write id.

Recall that $N(v)$ and $N(v)$ denote the sets of adjacent vertices to $v, u \in K_0$. We now demonstrate that, if u and v do not share any adjacent vertices (that is, $N(u) \cap N(v) = \emptyset$), and if $f_0(u)$ is closer to $f_0(v)$ than the least upper bound of $f_0(v)$ in $N(v)$, then permuting u and v does nothing to the lexicographical orderings in $\mathrm{star}_K(v)$. This allows us to restrict the class of vertex pair permutations we consider.

▶ **Lemma 4.1** (Permuting Within Least Upper Bound). *Let $f_0 \colon K_0 \to \mathbb{R}$ be injective and let $v, u \in K_0$. Suppose there exists $b \in N(v)$ such that $f_0(b) = \min\{c \in N(v) | f_0(c) > f_0(v)\}$. Let $p \colon \mathrm{Im}(f_0) \to \mathrm{Im}(f_0)$ be a permutation such that $p(f_0(v)) = f_0(u)$, $p(f_0(u)) = f_0(v)$, and $p = id$ otherwise. If $N(v) \cap N(u) = \emptyset$ and if $f_0(u) \in (f_0(v), f_0(b))$, then*

$$\textsc{ExtractRightChild}(star_K(v), p \circ f_0) = \textsc{ExtractRightChild}(star_K(v), f_0).$$

Proof. We need to show that local orderings among vertices in $N(v)$ are invariant after the application of p, because this leaves the lexicographical order of all simplices in $\mathrm{star}_K(v)$ unchanged. Since $f_0(b)$ is the least upper bound of $f_0(v)$ in $N(v)$, it follows that any $b' \in N(v)$ with $b' \neq b$ and $f_0(v) < f_0(b')$ has $|f_0(v) - f_0(b)| < |f_0(v) - f_0(b')|$. 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 $f_0(u) < f_0(b) < f_0(b')$ is identical, substituting u for v (Note that the inequalities are strict due to the injectivity of f_0). Hence, $p \circ f_0$ and f_0 maintain the same local orderings for every vertex in $\mathrm{star}_K(v)$. Consequently, if u and v are nonadjacent and $f_0(u) \in (f_0(a), f_0(b))$, $\textsc{ExtractRightChild}(\mathrm{star}_K(v), p \circ f_0)$ and $\textsc{ExtractRightChild}(\mathrm{star}_K(v), f_0)$ give the same (M, C) due to Definition 2.5, since $\textsc{ExtractRightChild}$ chooses matchings by lexicographical order, which is unchanged in $\mathrm{star}_K(v)$ after permuting v with u. ◀

Similarly, we can further restrict the class of permutations considered by examining the symmetric argument when permuting $u \in K_0$ with v such that $f_0(u)$ lies within the open interval $(f_0(a), f_0(v))$ for the greatest lower bound of v in $N(v)$.

▶ **Corollary 4.2** (Permuting Within Greatest Lower Bound). *Let $f_0 \colon K_0 \to \mathbb{R}$ be injective. Let $v, u \in K_0$, and let $N(v)$ denote the set of adjacent vertices to v. Suppose for $a \in N(v)$, $f_0(a) = \max\{c \in N(v) | f_0(c) < f_0(v)\}$. Define a permutation p such that $p(f_0(v)) = f_0(u)$, $p(f_0(u)) = f_0(v)$, and $p = id$ otherwise. If $N(v) \cap N(u) = \emptyset$ and if $f_0(u) \in (f_0(a), f_0(v))$, then $\textsc{ExtractRightChild}(star_K(v), p \circ f_0) = \textsc{ExtractRightChild}(star_K(v), f_0)$.*

Proof. The argument is identical to Lemma 4.1, reversing the direction of the inequalities. ◀

We note that proofs in Lemma 4.1 and Corollary 4.2 assume the existence of a least upper bound and greatest lower bound of v. Otherwise, the arguments are vacuously true.

As a result, if $v \in K_0$, permuting v with a non-adjacent vertex $u \in K_0$ where $f_0(u)$ is closer to $f_0(v)$ than any neighbors of v does not reduce the number of critical cells in $\textsc{ExtractRightChild}(\mathrm{star}_K(v), f_0)$. On the other hand, if u and v are adjacent, permuting them is guaranteed to alter local orderings in $\mathrm{star}_K(v)$, thereby always possibly decreasing $|C|$ output from $\textsc{ExtractRightChild}$. By only permuting *adjacent* vertices that are near in f_0 value, we rule out a considerable number of permutations among *nonadjacent* vertices what will make no difference to the local orderings of vertices in $N(v)$. We obtain a gradient descent algorithm by running the $\Theta(dn)$ algorithm given in [18], and then permuting adjacent vertices that are the nearest above and nearest below in f_0 value to a given vertex. We

Algorithm 2 MorseGradientDescent

Input: a simplicial complex K, a GVF (M', C') on K output by ExtractRightChild, and an injective $f_0 : K_0 \to \mathbb{R}$

Output: a locally optimal GVF (M, C) over K

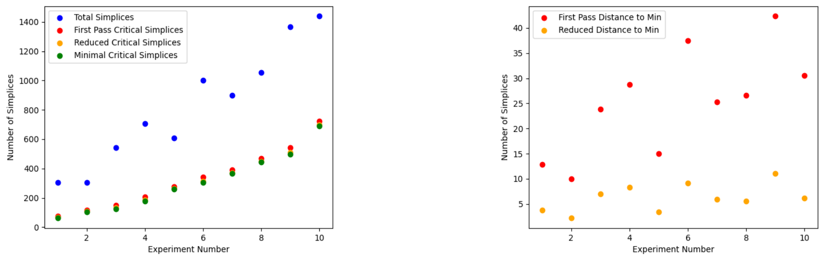
1: **for** $v \in K_0$ **do**
2: Save $a \in N(v)$ s.t. $f_0(a) = \min\{c \in N(v) | f_0(c) > f_0(v)\}$ ▷ greatest lower bound
3: Save $b \in N(v)$ s.t. $f_0(b) = \min\{c \in N(v) | f_0(c) > f_0(v)\}$ ▷ least upper bound
4: **end for**
5: $lowestCriticals \leftarrow \infty$
6: **while** $|C| < lowestCriticals$ **do**
7: $lowestCriticals \leftarrow |C|$
8: $C^* \leftarrow \{v \in K_0$ s.t. $v \in \sigma$ for some $\sigma \in C\}$ ▷ vertices participating in a critical cell
9: **for** $v \in C^*$ **do**
10: $a \leftarrow a \in N(v)$ s.t. $f_0(a) = \min\{c \in N(v) | f_0(c) > f_0(v)\}$
11: $b \leftarrow b \in N(v)$ s.t. $f_0(b) = \min\{c \in N(v) | f_0(c) > f_0(v)\}$
12: Define $p_1 \circ f_0(a) \leftarrow f_0(v)$
13: Define $p_2 \circ f_0(b) \leftarrow f_0(v)$
14: $(M', C') \leftarrow$ ExtractRightChild$(K, p_1 \circ f_0)$
15: $(M^*, C^*) \leftarrow$ ExtractRightChild$(K, p_2 \circ f_0)$
16: **if** Either $|C'| < lowestCriticals$ or $|C^*| < lowestCriticals$ **then**
17: Find the $i \in \{1, 2\}$ such that p_i gives the fewest critical cells
18: Update $f_0^* \leftarrow p_i \circ f_0$, $lowestCriticals \leftarrow \min(|C'|, |C^*|)$
19: **end if**
20: **end for**
21: $f_0 \leftarrow f_0^*$, where f_0^* permutes $u, v \in K_0$, or $f_0^* = f_0$ otherwise ▷ update f_0
22: Update greatest lower bound and least upper bound of v in $N(v)$
23: Update greatest lower bound and least upper bound of u in $N(u)$
24: **end while**
25: **return** ExtractRightChild(K, f_0)

examine the output of ExtractRightChild resulting from each permutation, and at each iteration we keep the permutation decreasing $|C|$ the most.

▶ **Lemma 4.3** (Complexity of MorseGradientDescent Updates). *Let $|C^*| = c$ in Line 8, on a fixed iteration of Line 6. Throughout a single iteration of Line 6, MorseGradientDescent updates f_0^* in $\Theta(c * dn)$ time using $O(1)$ space, where $c = O(d * |C|)$.*

Proof. Due to Line 1, the least upper bound and greatest lower bound of $f_0(v)$ in $N(v)$ are saved in advance, and both $p_1 \circ f_0$ and $p_2 \circ f_0$ can be defined in constant time. From [18], ExtractRightChild terminates in $\Theta(dn)$ time, which is called twice in Line 14 and Line 15. Finally, f_0 is updated in $O(1)$ time in Line 21, and subsequent updates to the greatest lower bound and least upper bound of u and v occur in $\mathcal{O}(n)$ time, traversing each edge of K_1 at most twice. The entire process is occurs for each $v \in C^*$, leaving updates to occur in $\Theta(c * dn)$ time. As there are at most d vertices per $\sigma \in C$ in Line 8, we can guarantee $c = O(d * |C|)$, where $|C|$ is given as input to MorseGradientDescent. Moreover, only a constant number of copies of each $\sigma \in K$ need to be saved, so each iteration of Line 6 uses $O(1)$ space. ◀

In the next section, we give strong bounds on the expected number of critical cells



(a) Comparison of n and $|C|$

(b) Distance to minimal $|C|$ after each step

Figure 4

resulting from ExtractRightChild when f_0 is randomly assigned. We use this to bound c from Lemma 4.3 as a constant, and also to justify that Line 6 only executes a constant number of times, demonstrating $\Theta(dn)$ time complexity of MorseGradientDescent.

4.2 A Simple Experiment

Keeping in mind that an important application of discrete Morse theory is in persistent homology, we conducted experiments on Vietoris-Rips complexes.

Our experiments were conducted on data ranging between 4 and 10 dimensions, with the number of total simplices lying between 200 and 1500. 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 $f_0 : K_0 \to \mathbb{R}$ were assigned by a random indexing. Our findings were striking, and simple: ExtractRightChild coupled with MorseGradientDescent approximates an optimal Morse function within a constant factor when run on realistic data. This is summarized in Figure 4a and Figure 4b, and addressed in the following table.

We note that these results are possible with a low variance for each experiment, and derive an experimental approximation factor from the data. This leads us to an additive error rate of roughly 0.026, so if (M, C) is an output of ExtractRightChild combined with MorseGradientDescent and (M^*, C^*) is a solution to MinMM, we find $|C| \approx 1.026 * |C^*|$. For additional details on our experimental methods, we refer to Appendix E.

Experiment	1	2	3	4	5	6	7	8	9	10
n	304	304	544	708	607	1002	897	1055	1364	1438
$\|C\|$ Algorithm 4	75	115	149	207	275	342	390	470	541	722
$\|C\|$ Algorithm 3	65	108	132	186	264	313	371	449	510	698
$\|C\|$ minimal	62	106	125	178	261	304	365	443	499	692

5 A Randomized Algorithm for MinMM in Higher Dimensions

In this section, we give an interpretation of the positive experimental results of ExtractRightChild when paired with MorseGradientDescent. In doing so, we arrive at a randomized additive approximation algorithm for MinMM.

We begin with a brief explanation of cases where our heuristics fall short, which helps to explain the NP-Hardness and W[P]-Hardness of MinMM. In particular, these are (1) maximal faces having many cofaces, and (2) cases where there does not *exist* an f_0 giving ExtractRightChild(K,f_0) that solves MinMM. We refer readers to examples of each scenario in Appendix F, which are given in Remark F.1 and Remark F.2 respectively. Of these two situations, we focus our attention on (1), and comment on (2) in Section 6.

We bound the error expected as a consequence of (1), and assume for simplicity no occurance of (2). Let K denote a simplicial complex, and $f_0 : K_0 \to \mathbb{R}$ an injective function on its vertices. Let (M, C) be a solution to MinMM, and (M', C') be the output of ExtractRightChild(K,f_0). Suppose $|C'| \geq |C|$, which is to say (M', C') is not optimal.

▶ **Definition 5.1** (Conflicted Critical Simplex). *We define a* conflicted critical simplex *as $\sigma \in C'$ such that $|C' \setminus \sigma| \geq |C|$, and there exists a (M, C) such that $\sigma \notin C$.*

If σ is a conflicted critical cell and τ is its largest lexicographical face, then there exists σ' with $(\tau, \sigma') \in M'$, where τ is also the largest lexicographical face of σ' by Definition 2.5.

▶ **Lemma 5.2** ("Error" Introduced by Conflicted Critical Cells). *Suppose $\sigma \in K_k$ is a conflicted critical simplex. Then (M', C') exhibits at least $O(2^{k+1})$ conflicted critical simplices.*

Proof. If f_0 causes a conflicted critical cell due to σ and σ' sharing their largest lexicographical face τ, 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 k-cells with their largest lexicographical $k-1$ dimensional faces, ExtractRightChild fails to assign σ to $M^{H'}$, and $\sigma \in C'$ as a result. Then every $\tau' \prec \sigma$ is adjacent to τ, and every face of τ could be paired with a lexicographically smaller coface participating in σ' (by the assumption that $\sigma' \in M^{H'}$ while $\sigma \in C'$, so $lex(\sigma') < lex(\sigma)$ by Definition 2.5). 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 which could have been already matched with the unique vertex $u \in \sigma'$ guaranteed to have $f_0(u) < f_0(v)$. For a simple example, see Figure 5. Hence, since σ and every lower dimensional simplex comprising σ can be unnecessarily critical, and there are 2^{k+1} simplices in a k-dimensional complex, $|C'| = |C| + O(2^{k+1})$. ◀

For a simple example to visualize Lemma 5.2, see Figure 5.

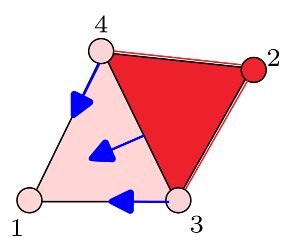


Figure 5 A simple example K with $f_0 : K_0 \to \mathbb{R}$ where ExtractRightChild causes $O(2^{d+1})$ conflicted critical cells.

We now demonstrate the low probability of the occurance of conflicted critical cells, justifying the strong experimental results of Algorithm 4.

▶ **Lemma 5.3.** *Let $m > 0, i \geq 0$ be integers with $i < m/2$, then $\frac{m/2-i}{m-i} \leq \frac{1}{2}$.*

Proof. It is easy to check that $\frac{m/2-i}{m-i} \leq \frac{1}{2}$ implies $-2i \leq -i$, which is certainly true. ◀

▶ **Lemma 5.4.** *Let $\#(K_0) = m$. The probability of $\sigma, \sigma' \in K_k$ sharing the same largest lexicographical face $\tau \in K_{k-1}$, as in Lemma 5.2, is bounded by $\prod_{i=0}^{k} \frac{(m/2)-i}{m-i} \leq \left(\frac{1}{2}\right)^{k+1}$.*

Proof. Without loss of generality, we assume a vertex $v \in K_0$ is indexed $f_0(v) \in \{1, ..., m\}$, since the orderings of simplices, including vertices, are all that is required in defining a discrete Morse function. For proof, see Lemma 2.2. Then the expected value of $f_0(v)$ is $m/2$. Conflicted critical cells only occur when two k-simplices share their largest lexicographical face. That is, in the one-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 $f_0(v') < f_0(u)$. Let $\{u_1, u_2, ...u_k\}$ be the set of every such $u \in K_0$. 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 finding a conflicted critical cell is $\prod_{i=0}^{k} \frac{(m/2)-i}{m-i}$, 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 $P(f_0(u_d) > f_0(v)) = \frac{(m/2)-(k-1)}{m-(k-1)}$, and the probability of choosing $f_0(v') < f_0(v)$ is $\frac{(m/2)-k}{m-k}$ if k values above $f_0(v)$ have been already assigned. Due to Lemma 5.3, $\prod_{i=0}^{k} \frac{(m/2)-i}{m-i}$ is bounded by $\left(\frac{1}{2}\right)^{k+1}$. ◀

▶ **Theorem 5.5** (Expected Error of ExtractRightChild)**.** *Given (M, C) solving MinMM on K and (M', C'), an output Morse matching of ExtractRightChild(K,f_0), the expected value of $|C'|$ is bounded by $E(|C'|) \leq |C| + O(1)$.*

Proof. It follows from Lemma 5.2 that a conflict in the largest lexicographical face τ of two cofaces $\sigma, \sigma' \in K_k$ causes on the order of 2^{k+1} conflicted critical cells. From Lemma 5.4 we know that the probability of this occurring is $\prod_{i=0}^{k} \frac{(m/2)-i}{m-i}$. Put together, we obtain the expected value $E(|C'|) = |C| + O(2^{k+1}) * \prod_{i=0}^{k} \frac{(m/2)-i}{m-i}$. Finally, we again use Lemma 5.3 to provide the upper bound.

$$E(|C'|) = |C| + O(2^{k+1}) * \prod_{i=0}^{k} \frac{(m/2)-i}{m-i} \leq |C| + O(2^{k+1}) * \left(\frac{1}{2}\right)^{k+1} = |C| + O(1).$$

◀

Now suppose, given (M', C') as an output to ExtractRightChild, we ran MorseGradientDescent to further refine (M', C'). We conclude with a bound on the expected value of $|C'|$ after calling MorseGradientDescent($(M', C'), f_0$). In doing so, we guarantee a substantial reduction in the constant added to $|C'|$ after running ExtractRightChild(K,f_0).

▶ **Lemma 5.6.** *The probability that any $\sigma \in K_k$ is a conflicted critical cell after the termination of MorseGradientDescent is bounded by $\prod_{v \in K_0} \left(\left(\frac{1}{2}\right)^{k+1}\right)^{j}$, where j is the number of permutations undertaken by each $v \in K_0$ in MorseGradientDescent.*

Proof. Suppose τ is the largest lexicographical face of $\sigma, \sigma' \in K_k$. By permuting every vertex and checking if $|C'|$ decreases, MorseGradientDescent is guaranteed to try a permutation of every vertex of τ, and after some permutation p this means that σ and σ' must not share the same largest lexicographical face. Due to Line 21, MorseGradientDescent only keeps the p decreasing $|C'|$ the most. For $\sigma \in K_k$, this eliminates $O(2^{k+1})$ extraneous critical cells, but could add $O(2^{h+1})$ new critical cells with probability $\prod_{i=0}^{h} \frac{(m/2)-i}{m-i}$ from Lemma 5.4, if the permutation occurs with a vertex of degree h. Every time a vertex $v \in K_0$ is permuted, the same argument holds (we remove $O(2^{k+1})$ critical cells, but possibly add new ones depending on the degree of the permuted vertex) and in total we obtain the probability of $\prod_{v\in K_0}\left(\prod_{i=0}^{h}\frac{(m/2)-i}{m-i}\right)^j$, where j is the number of permutations undertaken by each $v \in K_0$, and h is the degree of a vertex permuted in the output of MorseGradientDescent. Finally, due to Lemma 5.3, we know that $\prod_{i=0}^{h}\frac{(m/2)-i}{m-i} \leq \left(\frac{1}{2}\right)^{h+1}$, so $|C'|$ after running MorseGradientDescent is bounded by $\left(\prod_{v\in U}\left(\frac{1}{2}\right)^{h+1}\right)^j$. ◀

Using the results in Lemma 5.6 and Theorem 5.5, we obtain an expected value for $|C|$ after running ExtractRightChild coupled with MorseGradientDescent, which significantly decreases the constant obtained from ExtractRightChild alone.

▶ **Theorem 5.7.** *Let (M', C') be the GVF after running ExtractRightChild(K, f_0) and MorseGradientDescent$((M', C'), f_0)$. Then the expected value of $|C'|$ is bounded by $E(|C'|) \leq |C| + O(1) * \prod_{v\in K_0}\left(\left(\frac{1}{2}\right)^{d+1}\right)^j$.*

Proof. It follows from Lemma 5.4 that the probability of any two $\sigma, \sigma' \in K_k$ sharing their largest lexicographical face is bounded by $\left(\frac{1}{2}\right)^{k+1}$. We also know from Lemma 5.6 that the probability of such an occurance continuing after permutation is bounded above by $\prod_{v\in U}\left(\left(\frac{1}{2}\right)^{k+1}\right)^j$. Combined with Lemma 5.2, we reduce the constant in Theorem 5.5:

$$E(|C'|) \leq |C| + O(2^{d+1}) * \left(\frac{1}{2}\right)^{d+1} * \prod_{v\in U}\left(\left(\frac{1}{2}\right)^{d+1}\right)^j = |C| + O(1) * \prod_{v\in U}\left(\frac{1}{2}\right)^{j*(d+1)}.$$

◀

We comment that after MorseGradientDescent, $E(|critical s'|) \leq |critical s| + O(1)$, as $\prod_{v\in U}\left(\frac{1}{2}\right)^{j*(d+1)}$ is less than 1/2. In other words, MorseGradientDescent decreases the constant output by ExtractRightChild even further.

▶ Remark 5.8. As is mentioned at the start of the section, the results in Theorem 5.5 and Theorem 5.7 assume that given K there *exists* an injective function $f_0^* : K_0 \to \mathbb{R}$ such that ExtractRightChild(K, f_0^*) is a solution to MinMM. Indeed, this may not always be the case, as is exemplified by Remark F.2. In particular, torsion in K can cause the nonexistence of a perfect f_0^*. We remark that in realistic settings, producing K with torsion in many regions is quite rare, though in theory we could glue many such complexes with torsion together. Letting (M', C') be the output of ExtractRightChild(K, f_0^*) given an f_0^* minimizing $|C'|$ on K, this would force a large gap between the number of critical cells in solutions to MinMM and the smallest possible $|C'|$ recovered by ExtractRightChild.

We conclude our results with a $\Theta(dn)$ time randomized algorithm approximating MinMM within a constant factor for K where an injective f_0^* exists such that ExtractRightChild(K, f_0^*) outputs a (M, C) solving MinMM.

Algorithm 3 RandomizedExtract

Input: a simplicial complex K
Output: a GVF (M, C) approximating MinMM within a constant factor
1: Define $f_0 : K_0 \to \mathbb{R}$ at random, with no repetitions
2: $(M', C') \leftarrow$ ExtractRightChild(K, f_0)
    return MorseGradientDescent$((M', C'), f_0)$

▶ **Theorem 5.9** (Time Complexity of RandomizedExtract). *The algorithm RandomizedExtract terminates in $O(d^2 * |C| * n)$ time, using $\Theta(n)$ space. If $d = O(1)$ and $|C| = O(1)$, RandomizedExtract terminates in $\Theta(n)$ time.*

Proof. It is easy to see that Line 1 computes a randomized injective $f_0 : K_0 \to \mathbb{R}$ in $\Theta(n)$ time. Due to [18], Line 2 requires $\Theta(dn)$ time to execute ExtractRightChild. In Line 2, we know due to Lemma 4.3 that MorseGradientDescent chooses a single update to f_0 on K_0 in $\Theta(c * dn)$ time for c bounded by $d * |C'|$. Moreover, due to Theorem 5.5, $|C'| = |C| + O(1)$, and thus $c = O(d * |C|)$. Finally, Theorem 5.7 implies that we only consider $O(1)$ iterations in Line 6 of MorseGradientDescent, which we verify formally in Lemma F.3. Hence, RandomizedExtract is bottlenecked by Line 2, which terminates in $O(d^2 * |C| * n)$ time. If $d = O(1)$ and $|C| = O(1)$, the algorithm is $\Theta(n)$. As only $O(1)$ copies of K are ever required, RandomizedExtract uses $\Theta(n)$ space. ◀

6 Discussion

Given a simplicial complex K, this paper studies the problem of MinMM, which is to find a discrete Morse function on K that minimizes the number of critical simplices. The problem is approached through the lens of King et al., which additionally requires that an injective function $f_0 : K_0 \to \mathbb{R}$ is given on the vertices of K. We give a linear time algorithm solving MinMM for two-manifolds, which is the first improvement since 2005 on the methods of King et al. in [25]. In doing so, we demonstrate that the framework introduced in King et al., which computes a GVF using a given injective $f_0 : K_0 \to \mathbb{R}$ does not aid in efficiently computing a discrete Morse function in the case for two-manifolds. 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.

This paper also examines MinMM in higher dimensions when given an injective f_0 on the vertices of K. Using simple heuristics exploiting lexicographical orderings resulting from f_0, we provide an approximation of MinMM that is within a small additive factor by assigning a randomized f_0 to the vertices of K on a substantial class of complexes. In particular, these are complexes with few regions of torsion, where there *exists* an injective $f_0^* : K_0 \to \mathbb{R}$ such that ExtractRightChild(K, f_0^*) is a solution to MinMM. We additionally introduce a Morse-theoretic gradient descent heuristic to manipulate a given f_0 that approaches f_0^*. Our gradient descent substantially limits the expected number of critical simplices that result from ExtractRightChild(K,f_0). Despite the inapproximability of MinMM, we provide a brief experiment demonstrating the remarkably strong performance of ExtractRightChild when combined with MorseGradientDescent in practice. This leads us to a far-reaching randomized algorithm approximating MinMM within a constant additive factor on a realistic class of complexes.

Extensions to this work abound, and include the integration of randomized Morse theoretic heuristics in persistent homology applications. Our hope is that these results will have a sizable impact on the viability of computational techniques in Morse theory, and by extension, that computational topology broadly will become more powerful 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 (SoCG)*, 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 *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2663–2674, 2019.

5 Ulrich Bauer and Abhishek Rathod. Parameterized inapproximability of morse matching. arXiv:2109.04529, 2021.

6 Bruno Benedetti and Frank H. Lutz. Random Discrete Morse Theory and a New Library of Triangulations. *Experimental Mathematics*, 23(1):66–94, 2014.

7 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 (SoCG)*, pages 344–350, 2003.

8 Lidija Čomić and Leila De Floriani. Dimension-independent simplification and refinement of Morse complexes. *Graphical Models*, 73(5):261–285, 2011.

9 Tamal Dey, Jiayuan Wang, and Yusu Wang. Graph reconstruction by discrete Morse theory. In *Proceedings of the Thirty-Fourth Annual ACM-SIAM Symposium on Computational Geometry (SoCG)*, pages 31:1–31–13, 2018.

10 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), page 277–284, New York, NY, USA, 1994. Association for Computing Machinery.

11 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.

12 Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction.* American Mathematical Society, 2010.

13 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.

14 Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. *Discrete and Computational Geometry*, 28:511–533, 2002.

15 Alexander Engström. Discrete Morse Functions from Fourier Transforms. *Experimental Mathematics*, 18(1):45–54, January 2009.

16 David Eppstein. Dynamic generators of topologically embedded graphs. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 599–608, 2003.

17 Jeff Erickson. One-dimensional computational topology lecture notes. Lecture 10: Tree-CoTree Decompositions, 2020.

18 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.

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

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

21 Attila Gyulassy, Peer-Timo Bremer, Bernd Hamann, and Valerio Pascucci. A practical approach to morse-smale complex computation: Scalability and generality. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1619–1626, 2008.

22 Shaun Harker, Konstantin Mischaikow, Marian Mrozek, Vidit Nanda, Hubert Wagner, Mateusz Juda, and Paweł Dłotko. The efficiency of a homology algorithm based on discrete morse theory and coreductions. *Proceedings of the Third International Workshop on Computational Topology in Image Context*, pages 41–47, 2010.

23 Allen Hatcher. *Algebraic topology.* Cambridge University Press, 2002.

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

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

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

27 Thomas Lewiner, Hélio Lopes, and Geovan Tavares. Toward optimality in discrete Morse theory. *Experimental Mathematics*, 12:271–285, 2003.

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

29 James Munkres. *Elements of algebraic topology.* CRC Press, Boca Raton London New York, 1993.

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

31 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 topology and 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 cofaces 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 [19, 30].*

A.2 Lexicographical Orderings on K

We now provide a rigorous definition of the intuitive lexicographical ordering induced by a given injective $f_0 : K_0 \to \mathbb{R}$. Let $i \in \mathbb{Z}, i \geq 0$ and $\sigma, \sigma' \in K_i$. Using f_0, define the sets $U = \{f_0(u_1), f_0(u_2), ..., f_0(u_i)\}$ and $V = \{f_0(v_1), f_0(v_2), ..., f_0(v_i)\}$ for every $u_k \in \sigma, v_k \in \sigma'$ with $k \in \{1, 2, ..., i\}$. Assuming that U and V are sorted from largest to smallest, we can define the lexicographical order of σ, σ'.

▶ **Definition A.2** (Lexicographical Order). *Given $\sigma, \sigma' \in K_i$ and an injective $f_0 : K_0 \to \mathbb{R}$, we say that $lex(\sigma) > lex(\sigma')$ if $u_j > v_j$ at an index $1 \leq j \leq i$, and the values of U and V are equivalent at all prior indices.*

Using lexicographical orderings and Definition 2.5, we categorize outputs to ExtractRightChild.

▶ **Lemma A.3** (Gradient Vector Fields Output by ExtractRightChild). *If (M, C) is a gradient vector field on K corresponding to an discrete Morse function f of ExtractRightChild, then any $(\tau^*, \sigma^*) \in M$ has every face $\tau \prec \sigma^*$ satisfying $lex(\tau) \leq lex(\tau^*)$, and every coface $\sigma \succ \tau^*$ satisfying $lex(\sigma) \geq lex(\sigma^*)$.*

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

A.3 Additional Definitions from Computational Topology

In this paper, we assume a basic knowledge of computational topology, e.g., as presented in [12]. Here, we provide a few key definitions for easy reference.

Manifold

A *d-manifold* (without boundary) is a topological space M such that for each point $x \in M$, there exists an open neighorhood containing x that is homeomorphic to $\mathbb{R}^d$. In particular, each point in a two-manifold is homeomorphic to $\mathbb{R}^d$ (which, in turn, is homeomorphic to the open unit ball). Familiar examples of two-manifolds include a sphere, a torus, and a Klein bottle.

Simplices

A *k-simplex* is a set with $k + 1$ elements. Each element is called a *vertex*. A geometric simplex is the set of all convex combinations of $k + 1$ affinely independent vertices in $\mathbb{R}^d$.

Abstract Simplicial Complex

An *abstract simplicial complex* K is a set of simplices that is closed under nontrivial subsets. If $\tau, \sigma \in K$ with $\tau \subseteq \sigma$, then we say that τ is a *face* of σ and σ is a *coface* of τ; we denote this relation $\tau \preceq \sigma$.

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 [18,25], 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 [25] 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 [18], the time complexity to generate naïve discrete Morse functions is improved to $\Theta(dn)$. Here, we first present the algorithm ExtractRightChild from [18] to generate naïve Morse functions, and then give the full algorithm Extract from [25] which gives a better approximation of MinMM.

B.1 ExtractRightChild

Below, we provide Algorithm 2 of [18] with one modification: instead of keeping track of heads and tails (H and T of [18]) as separate sets on the GVF, we simply maintain a set of head-tail pairs (denoted M here and as m in [18, 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 [18], 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 [25], a subroutine of Extract called ExtractRaw is used to compute a

Algorithm 4 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 [18, 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)

"raw" gradient vector field. We conclude discussion of ExtractRightChild with a lemma establishing the equality of ExtractRightChild and ExtractRaw, proven in [18]:

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

Algorithm 5 [25] Extract

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

```
1: (M, C) ← ExtractRightChild(K, f_0)
2: for j ∈ {1, ..., dim(K)} do
3:     ExtractCancel(K, f_0, p, j, (M, C))
4: end for
```

In the original version of Extract given in [25], 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 6 [25] 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 [25], 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 [18, 25], comparing them to the methods in this paper.

In [18] 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 MorseGradientDescentterminates with a constant number of iterations.

Furthermore, it is easy to see that ExtractCancel, as it is presented in [25], has $\Theta(dn^3)$ time complexity.

▶ **Lemma C.1** (Runtime of ExtractCancel). *Algorithm 6 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 F.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 5 terminates in $\Theta(d^2n^3)$ time, using $\Theta(n)$ space.*

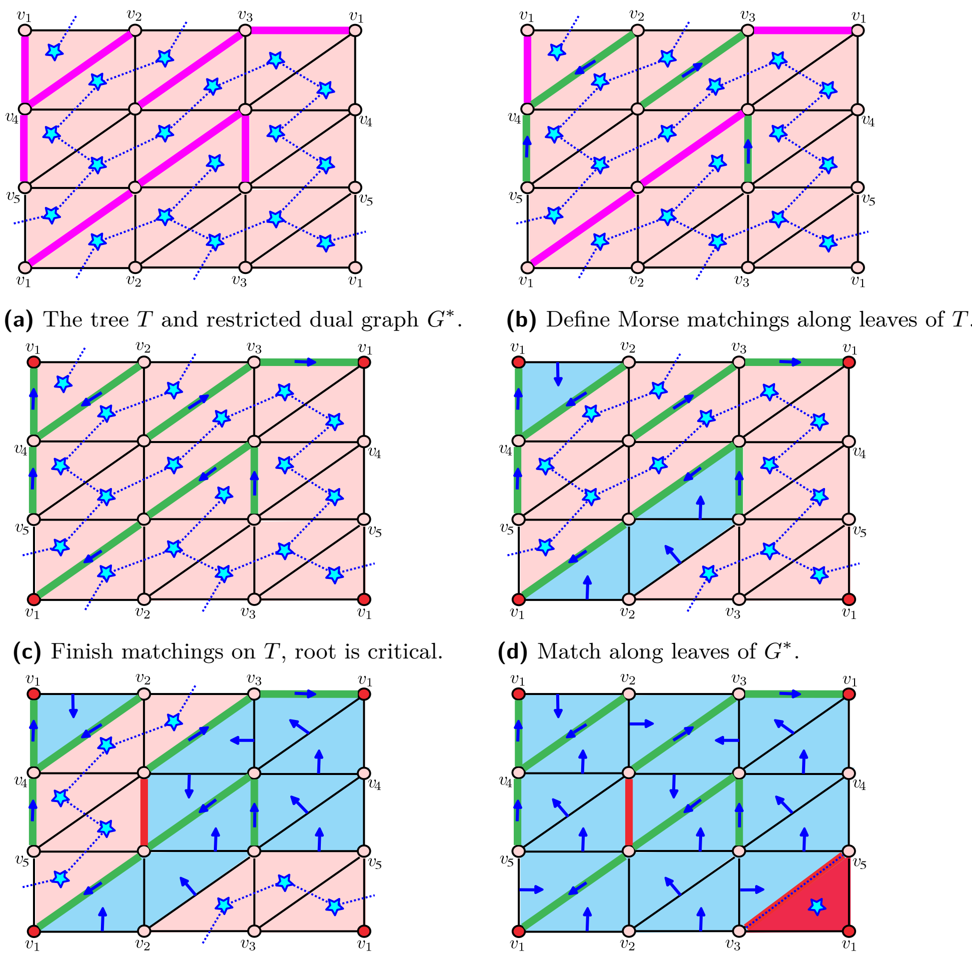
Proof. Recall Extract only calls ExtractRightChild once, and then calls Extract-Cancel in each dimension. From [18], we know that ExtractRightChild terminates in $\Theta(dn)$ time using $\Theta(n)$ space. From Lemma C.1 we know that ExtractCancelterminates in $\Theta(dn^3)$ time per dimension. As ExtractCancel 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 [25] without memoization. As ExtractRightChild and ExtractCancel each use $\Theta(n)$ space, the same holds for Extract. ◀

When compared with the combination of ExtractRightChild and MorseGradient-Descent, 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.

D Additional Details for MorseDual

D.1 An Example Run of MorseDual

For clarity, we include a figure demonstrating a run of MorseDual on a triangulated torus.



(a) The tree T and restricted dual graph G^*.
(b) Define Morse matchings along leaves of T.
(c) Finish matchings on T, root is critical.
(d) Match along leaves of G^*.
(e) Cut a critical edge, match leaves of G^*.
(f) Match remaining leaves of G^*.

Figure 6 The major steps of MorseDual when run on K as a triangulated torus.

D.2 Experimentation for MorseDual

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. In future versions of the paper, we plan to compare our implementation 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. All of the triangulation data used came from Ryan Holmes' library of .OFF files or from John Burkhardt's library of .OFF files, which are accessible at the following addresses:

1. http://www.holmes3d.net/graphics/offfiles/
2. https://people.sc.fsu.edu/ jburkardt/data/off/off.html

We highlight that the theoretical time complexity discussed for MorseDual holds in practice as n increases. Experimental time complexity is as follows:

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

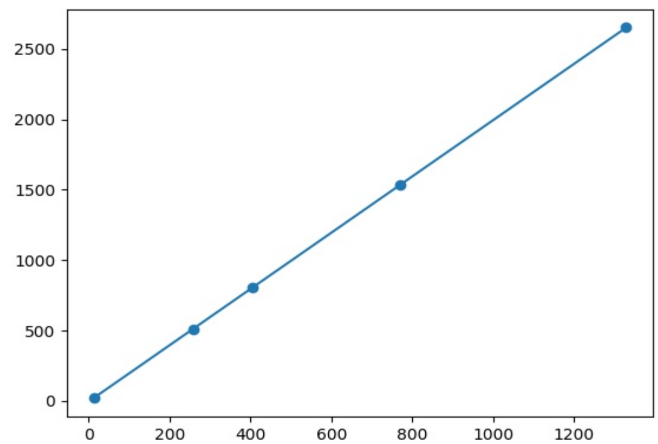


Figure 7 As proof of concept, we find linear asymptotic behavior in practice n grows.

E Additional Details for MorseGradientDescent

This section gives additional details on the experimentation in Section 4.

For simplicity, the table included in Section 4 rounds all averages to the nearest integer. The experimental approximation factor was computed by finding the difference between $|C|$ solving MinMM, and $|C|$ given as an output of ExtractRightChild combined with MorseGradientDescent. That is, using the rounded averages included in Section 4:

$$(3/62 + 2/106 + 7/125 + 8/178 + 9/304 + 6/365 + 6/443 + 11/499 + 6/692)/10 = 0.02585$$

This tells us, in practice, that we can expect outputs to ExtractRightChild combined with MorseGradientDescent to be optimal Morse matchings with roughly 2.6% more critical cells.

F Additional Details for Section 5

This section provides additional details for Section 5.

We begin by discussing scenarios where ExtractRightChild can struggle to find an optimal Morse matching. These are (1) where the same largest lexicographical face is shared by many cofaces, and (2) where there exists no injective f_0 on the vertices whose output to ExtractRightChild(K,f_0) is a solution to MinMM.

Regarding (1), it is easy to see that Morse functions output by ExtractRightChild could have potentially very many extraneous critical cells, depending on the assigned f_0.

▶ Remark F.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 GVF output by ExtractRightChild it is possible to have $|C| = O(n)$. For a simple example, see Figure 8.

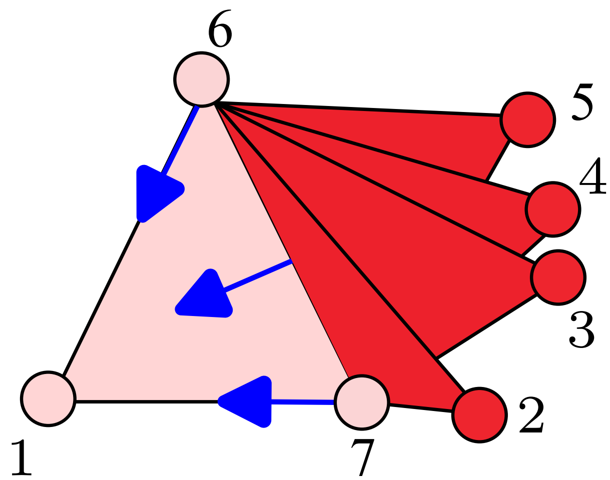


Figure 8 An example two-dimensional complex K where the GVF induced by f_0 has $|C| = O(n)$, with critical cells in red.

As for situation (2), it could also be impossible to have any f_0 giving ExtractRightChild(K,f_0) whose output is a solution to MinMM:

▶ Remark F.2. It is entirely possible that *no* injective function $f_0 : K_0 \to \mathbb{R}$ exists with ExtractRightChild(K,f_0) producing the same number of critical cells as an output to MinMM. This can be due to regions of non-orientability in complexes such as the one in Figure 9.

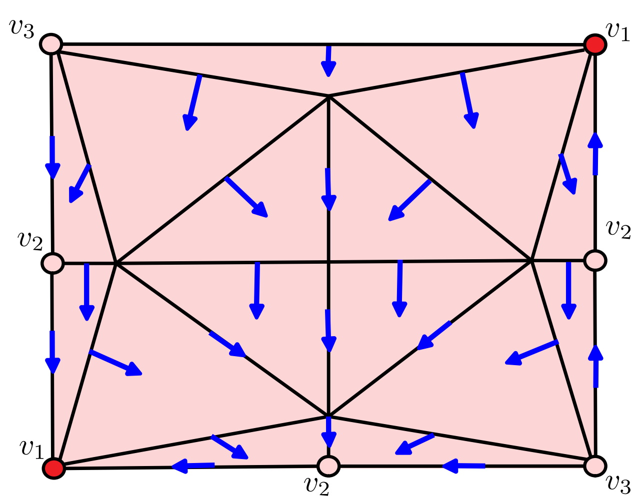


Figure 9 A variant of the triangulated dunce cap and its canonical gradient vector field, with only one critical vertex in red. It is impossible to assign a $f_0 : K_0 \to \mathbb{R}$ to K that is consistent to f_0 and a solution to MinMM, due to torsion in K.

Additionally, we include verification of the runtime of MorseGradientDescentbased on the expected values obtained in Section 5.

▶ **Lemma F.3** (Bounding the Total Iterations of MorseGradientDescent). *In MorseGradientDescent, Line 6 iterates $O(1)$ times.*

Proof. Let K be a simplicial complex, and (M, C) be a solution to MinMM given K. Let $f_0 : K_0 \to \mathbb{R}$ be randomly assigned and injective, and (M', C') the output of ExtractRightChild(K,f_0). Due to Theorem 5.5, we gain that ExtractRightChild(K,f_0) attains $|C'| = |C| + O(1)$. Moreover, Due to Theorem 5.7, MorseGradientDescent decreases $|C|$ on every iteration of Line 6. Hence, Line 6 of MorseGradientDescent iterates $O(1)$ times. ◀