A User Manual

A.1 Overview

We provide an interactive visualization tool for users to explore the process of constructing a discrete stratified Morse function from any real-valued function defined on a 1- or 2-complex embedded in the plan, and to perform homotopy-preserving simplification of the resulting stratified complex. The tool provides 6 examples for discrete stratified Morse theory (DSMT), and for comparison, 3 examples for discrete Morse theory (DMT). The tool enables the random perturbation of function values for each example. It also allows the import of user-specified examples in an appropriate format. Figure 3 demonstrates the user interface of the tool.

In section 2, we explain how to get started with our tool. In section 3, we describe the main features of the tool. In section 4, we explain each example provided in the tool in details. In section 5, we illustrate the implementation details.

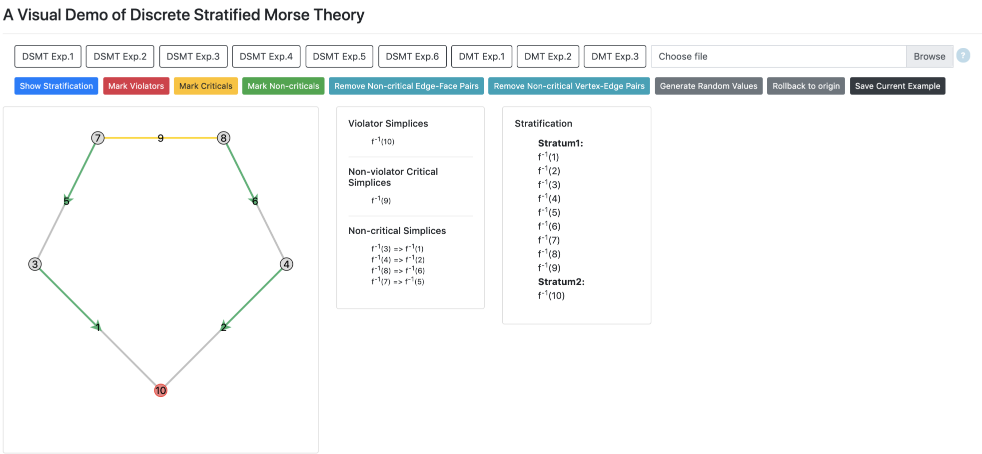

Figure 3 The user interface of our visualization tool.

A.2 Getting Started

A.2.1 Live Demo

To see a live demo, go to: https://vis-dsmt.herokuapp.com/. It runs on most modern web browsers. We suggest you use Google Chrome.

A.2.2 Running Locally

Download or clone this repository:

```
$ git clone git@github.com:tdavislab/VIS-DSMT.git
```

Then, run:

```
$ cd VIS-DSMT
$ npm start
(# Hit Ctrl+c to quit)
```

You can view the page at http://0.0.0.0:3000/.

A.3 Main Features

We introduce the main features of the tool using Figure 4 as a simple example. f in Figure 4 is not a discrete stratified Morse function, but our tool converts it into one with an appropriate stratification s.

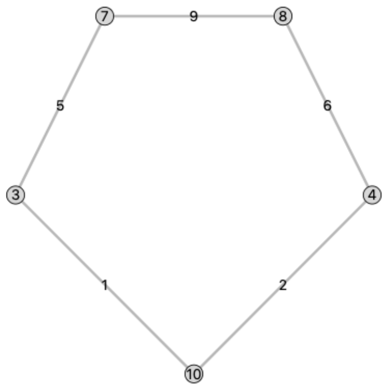

Figure 4 DSMT Exp. 1: a real-valued function defined on an upside-down pentagon. Function values are marked on the corresponding vertices and edges.

A.3.1 Show stratification

By clicking the "Show Stratification" button, the system highlights the resulting stratification, with different colors representing different strata. The system also lists the detailed information of the simplices contained in each stratum. As shown in Figure. 5, the resulting stratification of this example contains two strata. The first stratum contains all the simplices except the vertex with a function value of 10, and the second stratum contains only that vertex.

A.3.2 Mark violators, criticals and non-criticals

Users can explore the various stages of how the tool constructs the discrete stratified Morse function by marking violators, critical simplices, and noncritical simplices. Violators are marked in red, and critical simplices are marked in yellow. Noncritical simplices are marked in Morse pairs with green arrows, pointing from lower dimensional simplex to higher dimensional simplex. The list of violators, critical simplices and noncritical pairs is displayed next to the figure. As shown in Figure. 6, this example contains one violator, one non-violator critical simplex, and four noncritical pairs.

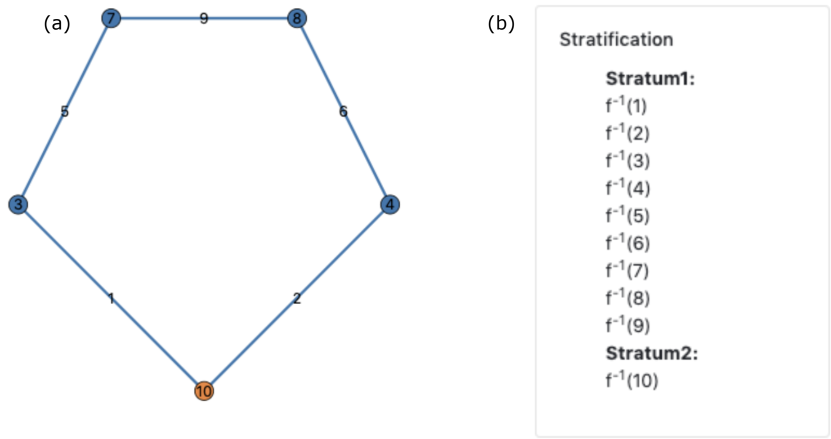

Figure 5 Show stratification. (a) The resulting stratification is highlighted on the example. (b) The detailed information of the simplices contained in each stratum.

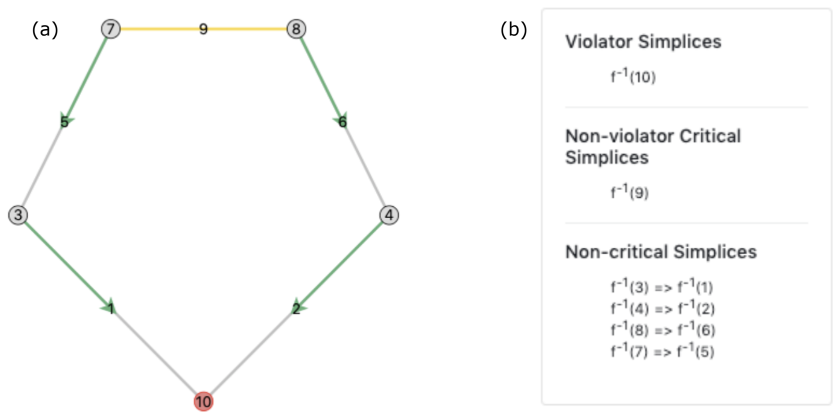

Figure 6 (a) Violators are marked in red; critical simplices are marked in yellow; noncritical pairs are marked with green arrows. (b) List of violators, critical simplices, and noncritical pairs.

A.3.3 Homotopy-preserving simplification

The tool allows users to perform simplification of the stratified simplicial complex by removing noncritical pairs to produce a reduced complex with the same homotopy type. Since we are dealing with 1- and 2-complexes embedded in the plane, all possible Morse pairs include edge-face pairs and vertex-edge pairs. The tool allows simplification of both types of the pairs.

During the simplification process, there is a default order to remove noncritical pairs. The system asks users to begin by removing all edge-face pairs with a free edge, until that is no longer possible, and then to remove all vertex-edge pairs with a free vertex. A free edge is an edge that is not adjacent to two faces. Similarly, a free vertex is a vertex that is not adjacent to two edges. Eventually, they might run into a situation where there are pairs with no free edge/vertex. In this situation, the system will ask users to remove edge-face pairs without a free edge first, and then to remove vertex-edge pairs without a free vertex. When removing an edge-face pair without a free edge, the visualization algorithm fills the void by expanding the adjacent face to fill the space, keeping the same homotopy type of the complex. When removing a vertex-edge pair without a free vertex, the result is to identify the two vertices of the edge together, possibly creating a loop. The loop is either a nontrivial cycle or part of the boundary of a new 2-cell. See Figure. 7 for examples of these different types of simplification. After each step of simplification, the system will recompute a new discrete stratified Morse function given the current simplicial complex.

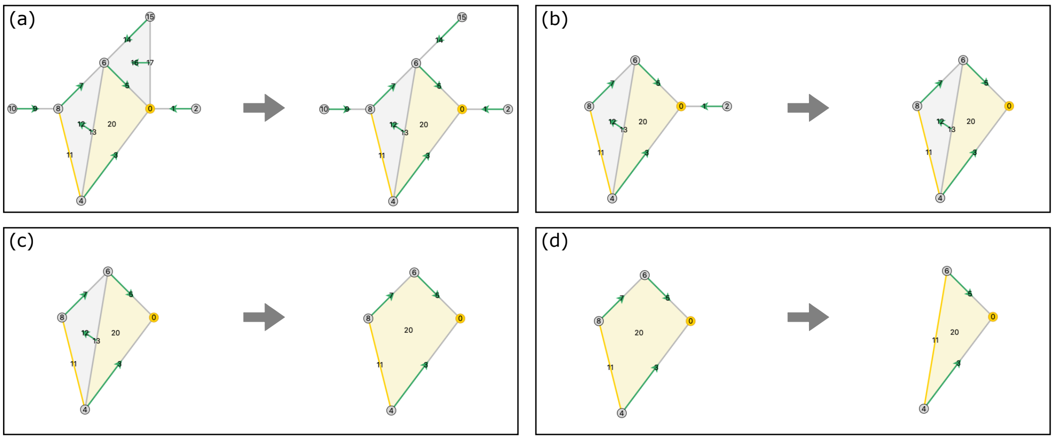

Figure 7 (a) Removing a edge-face pair with a free edge. (b) Removing a vertex-edge pair with a free vertex. (c) Removing a edge-face pair without a free edge. (d) Removing a vertex-edge pair without a free vertex.

Figure. 8 demonstrates the simplification process of our example. The final simplified simplicial complex contains one vertex and one edge looping around the vertex.

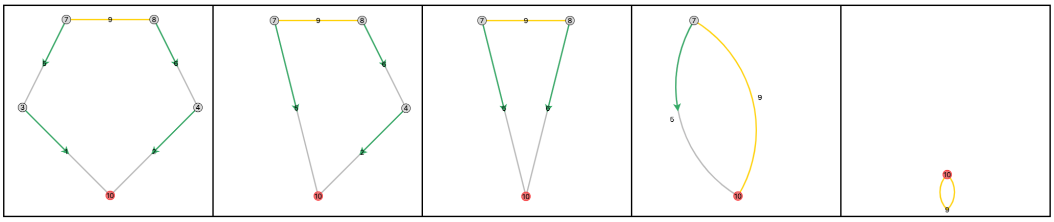

Figure 8 The simplification process of **DSMT Exp. 1**.

By clicking the "Rollback to Origin" button, the system will bring back the original simplicial complex before any simplification.

A.3.4 Random perturbation

By clicking the "Generate Random Values" button, the tool will assign a random number as the function value for each simplex. The random perturbation may result in a different stratification and a different simplified simplicial complex. See Figure. 9 for an example of random perturbation.

A.3.5 Downloading an example

In case that users may want to save a randomly perturbed example for further exploration in the future, the tool allows users to download a TXT file containing the current example

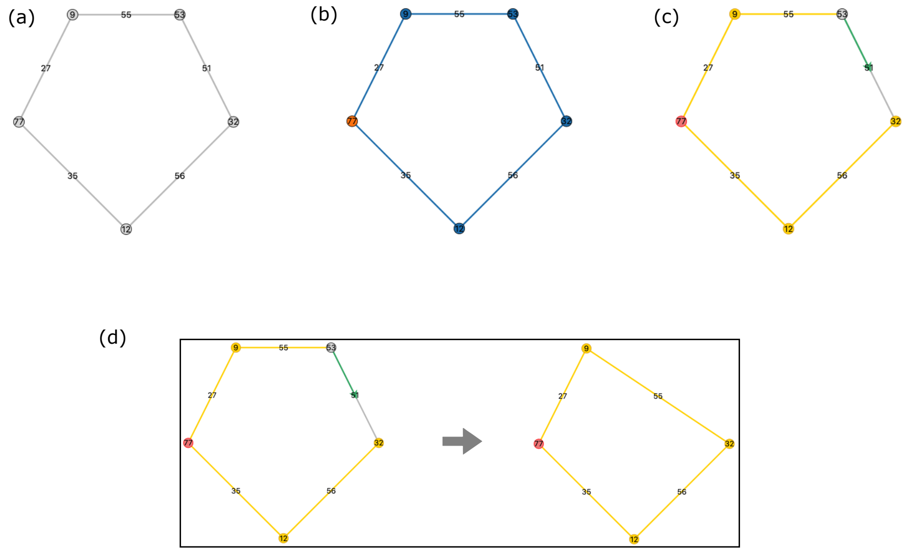

Figure 9 (a) **DSMT Exp. 1** with randomly perturbed function values. (b) The resulting stratification of the perturbed example. (c) The violators, critical simplices and noncritical pairs of the perturbed example. (d) The simplification process of the perturbed example.

information by clicking the "Save Current Example" button. The downloaded example can be directly imported into the tool again.

A.3.6 Importing a user-specified example

Users can import a new example with specific data format. The input file should be in TXT format. As shown in Figure. 10, the first line should be the title "OFF". The second line specifies the number of vertices, faces and edges. Starting from the third line, each line specifies the position and function value of a simplex. The order is to specify vertices first, then faces, and then edges. For vertices, the position is represented as its x and y coordinates. For edges, the position is represented as the indices of the vertices on its two ends. For faces, the position is represented as the indices of the vertices on its boundary. For example, if a face is a triangle, the line specifying it will have the form of "u, v, w, f", where u, v, w are the indices of its three vertices, and f is the function value of it. The imported example is expected to be some planar geometric 2-dimensional simplicial complex (e.g., a planar triangulation).

```
OFF
3 1 3
1 1 1
0 0 2
2 0 3
0 1 2 7
0 1 4
0 2 5
1 2 6
```

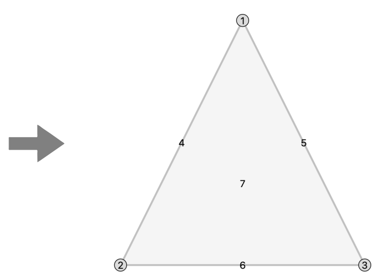

Figure 10 Importing a data file to draw a triangle.

A.4 Examples

We've described DSMT Exp.1 in the previous section. We will introduce the remaining examples in this section.

A.4.1 DSMT Exp. 2

This is a pentagon example. The resulting stratification contains 6 strata. This example contains 2 violator, 2 non-violator critical simplices, and 4 noncritical pairs. The simplification process is shown in Figure. 11.

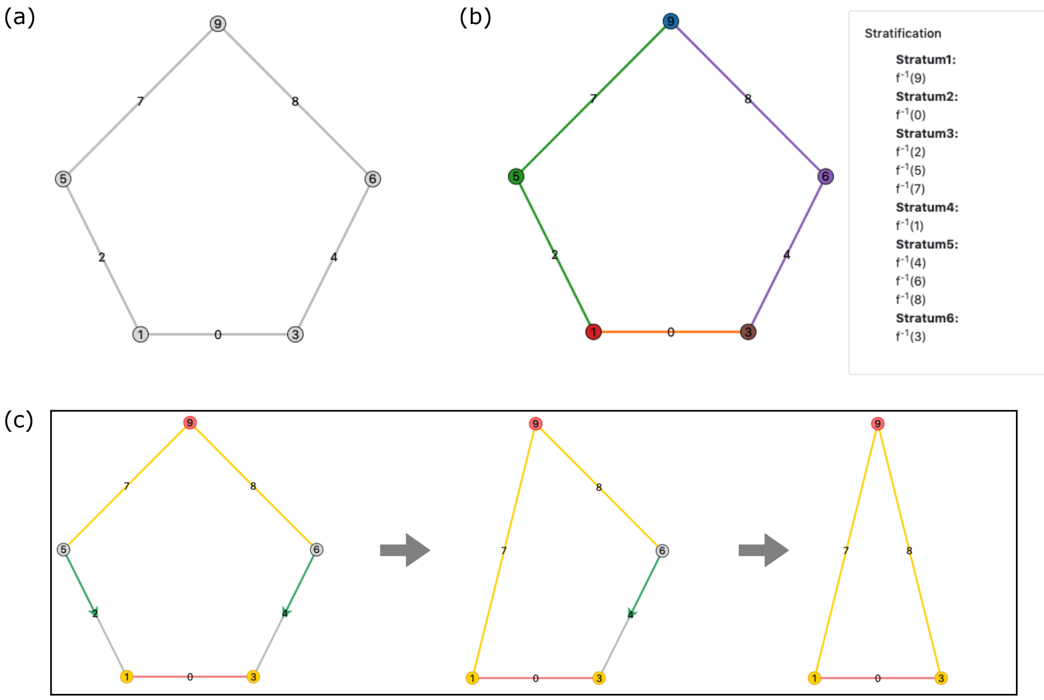

Figure 11 DSMT Exp. 2: pentagon. (b) Resulting stratification. (c) Simplification process.

A.4.2 DSMT Exp. 3

This is a split octagon example. The resulting stratification consists of two 0-dimensional and three 1-dimensional strata. This example contains 2 violator, 3 non-violator critical simplices, and 9 noncritical pairs. The simplification process is shown in Figure. 12.

A.4.3 DSMT Exp. 4

This example is a split solid square. The resulting stratification consists of 5 strata. This example contains 3 violators and 8 non-violator critical simplices. In this example, all simplices are considered critical for the resulting discrete stratified Morse function. Since there are no noncritical pairs, no simplification needs to be done.

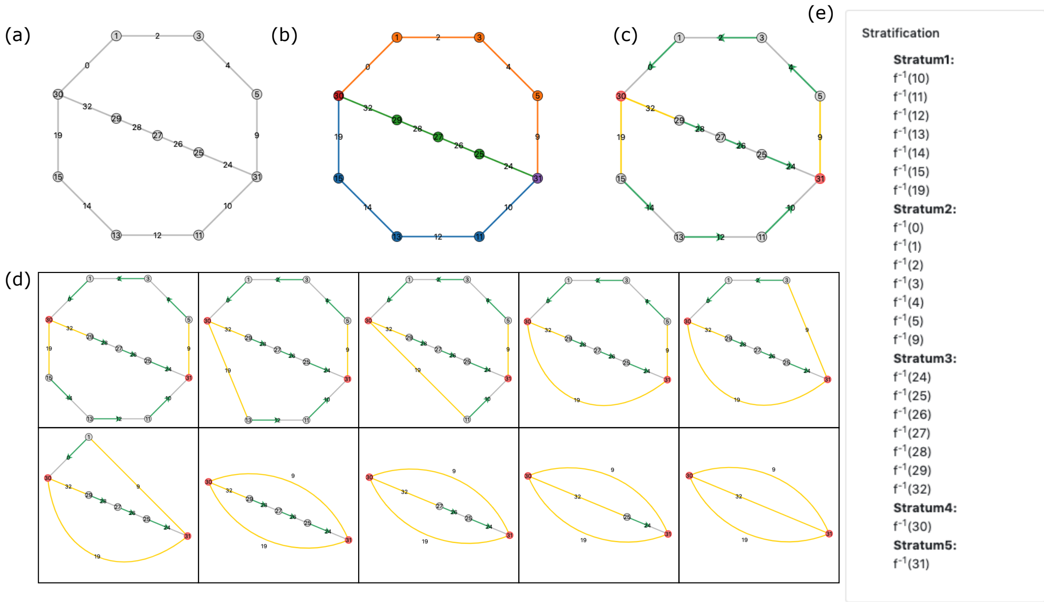

Figure 12 DSMT Exp. 3: split octagon. (b) Resulting stratification. (c) Violators, critical simplices and noncritical pairs. (d) Simplification process. (e) List of strata information.

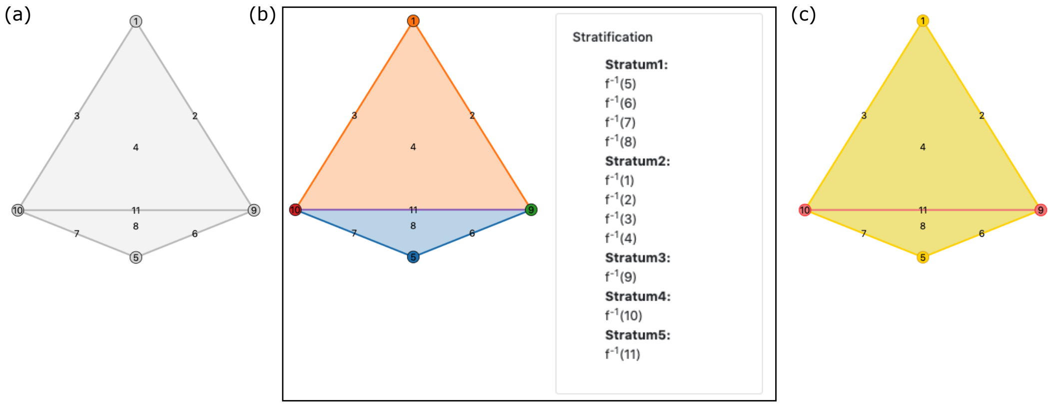

Figure 13 DSMT Exp. 4: split solid square. (b) Resulting stratification. (c) Violators and critical simplices.

A.4.4 DSMT Exp. 5

This example is a triangulated mesh with function values randomly generated on its vertices and edges. Figure. 14 demonstrates one randomly generated example of it.

A.4.5 DSMT Exp. 6

This example is a triangulated mesh with function values randomly generated on its vertices, edges and faces. Figure. 15 demonstrates one randomly generated example of it. During the simplification, there might be faces overlapping each other. When hovering over a specific face, its shape will be highlighted by increasing its opacity.

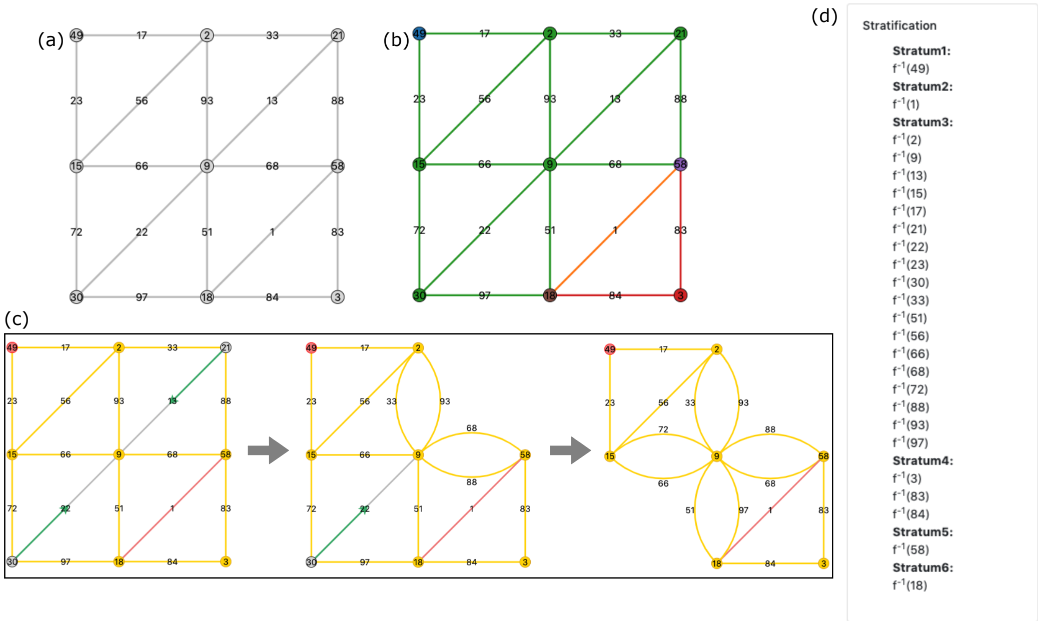

Figure 14 DSMT Exp.5: a triangulated mesh with function values randomly generated on its vertices and edges.

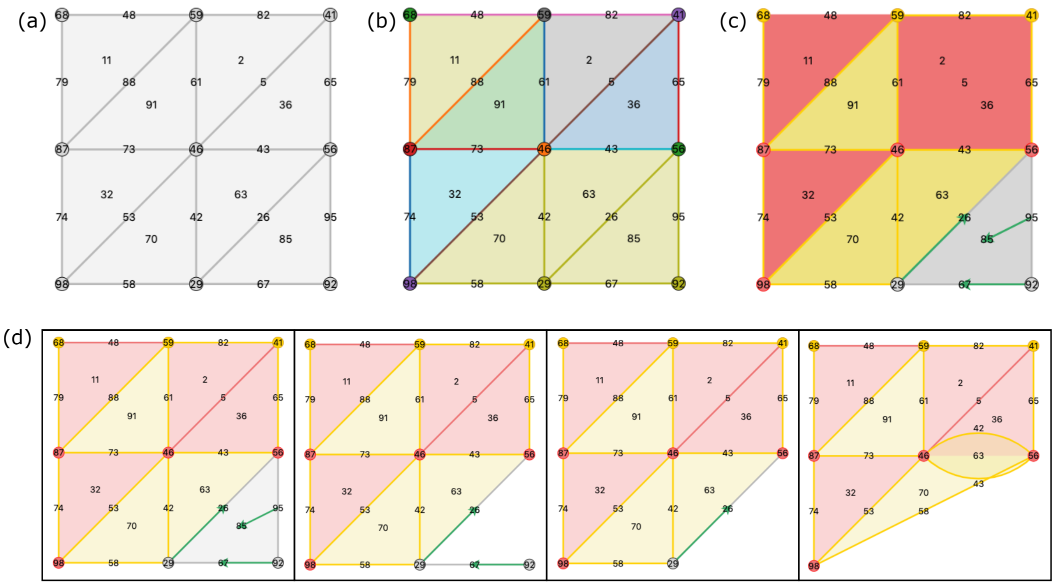

Figure 15 DSMT Exp.6: a triangulated mesh with function values randomly generated on its vertices, edges and faces.

A.4.6 DMT Exp. 1

In this example, the function defined on the simplicial complex is a discrete Morse function. Since a discrete Morse function is a discrete stratified Morse function for the trivial stratification, there are no violators, and the resulting stratification contains only one stratum. There are 2 critical simplices, and 8 noncritical pairs. Figure 16 demonstrates the simplification process of this example.

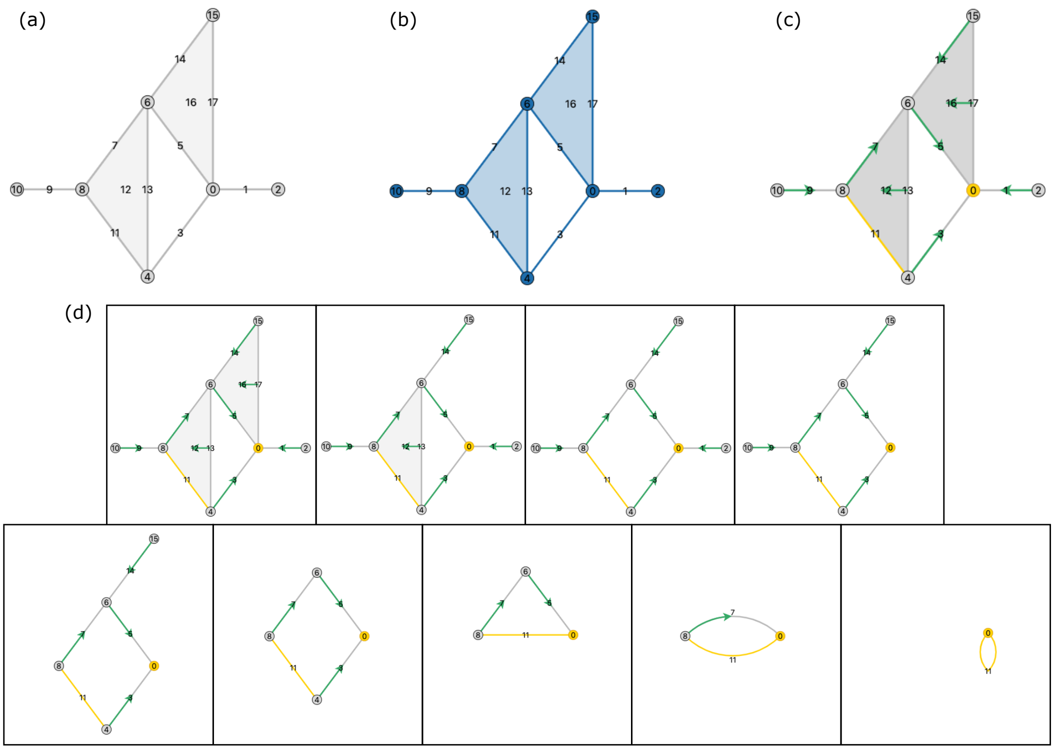

Figure 16 DMT Exp.1. (b) Only one stratum in the resulting stratification.

A.4.7 DMT Exp. 2

Similar to **DMT Exp. 1**, there is only one stratum in the resulting stratification. There are 3 critical simplices, and 8 noncritical pairs. Figure 17 demonstrates the simplification process of this example.

A.4.8 DMT Exp. 3

Similar to **DMT Exp. 1**, there is only one stratum in the resulting stratification. There are 1 critical simplices, and 24 noncritical pairs. Figure 18 demonstrates the simplification process of this example. The final simplified result is a vertex.

A.5 Implementation Details

The tool is web-based and can be accessed from any modern web-browser (tested using Google Chrome).

The tool is implemented in HTML, CSS, and JavaScript. The algorithm for constructing discrete stratified Morse functions is implemented in pure JavaScript, and the module collection *D3.js* is used for rendering SVGs and providing interactive visualization.

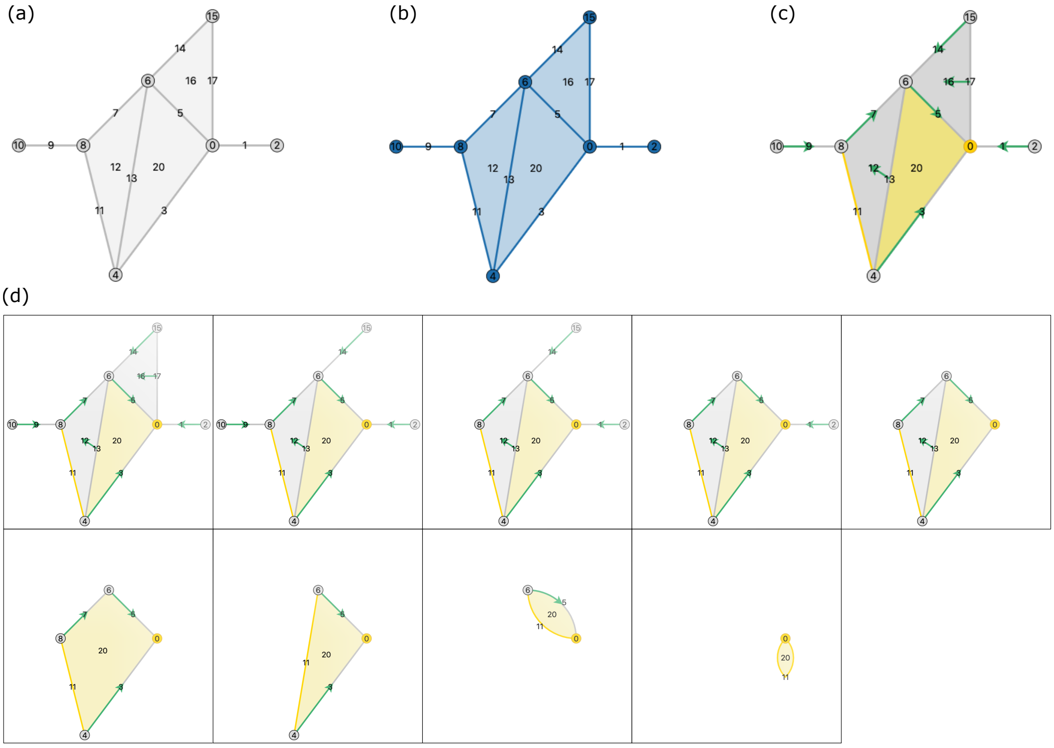

Figure 17 DMT Exp.2

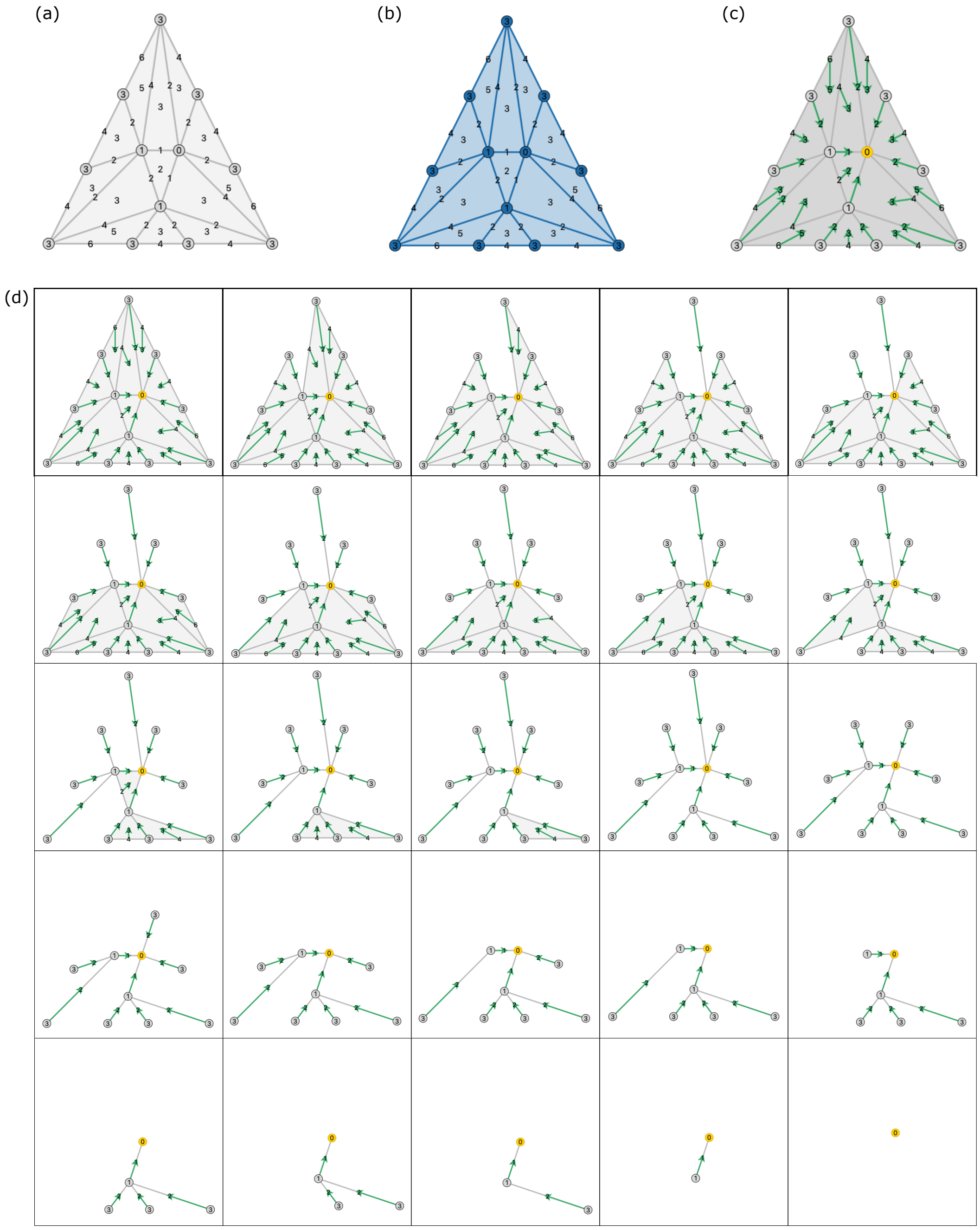

Figure 18 DMT Exp.3