Removing Local Extrema from Imprecise Terrains

Chris Gray* Frank Kammer† Maarten Löffler‡ Rodrigo I. Silveira§

Abstract

In this paper, we study imprecise terrains, that is, triangulated terrains with a vertical error interval in the vertices. We study the problem of removing as many local extrema (minima and maxima) from the terrain as possible. We show that removing only minima or only maxima can be done optimally in $O(n \log n)$ time, for a terrain with n vertices, while removing both at the same time is NP-hard. To show hardness, we exploit a connection to a graph problem that is a special case of 2-Disjoint Connected Subgraphs, a problem that has received quite some attention lately in the graph theory community. This special case of 2-Disjoint Connected Subgraphs is shown NP-hard.

1 Introduction

A triangulated (or polyhedral) terrain is a planar triangulation with a height associated with each vertex. This results in a bivariate and continuous function, defining a surface that is often called a 2.5-dimensional (or 2.5D) terrain.

Even though in computational geometry it is usually assumed that the input data is exact, in practice, terrain data is most of the time imprecise. The sources of imprecision are many, starting from the methods used to acquire the data, which are ultimately based on error-prone measuring devices. Often such methods produce heights with a known error bound or return a height interval rather than a fixed height value. Even though terrain data may contain error also in the x, y-coordinates, we consider imprecision only in the z-coordinate. This simplifying assumption is justified by the fact that error in the x, y-coordinates will most likely produce elevation error. Moreover, often the data provided by commercial terrain data suppliers only reports the elevation error [2].

We describe an *imprecise terrain* by a set of n vertical intervals in $\mathbb{R}^3$, together with a triangulation of the projection. See Figure 1(a). We say that a triangulated terrain is a *realization* of an imprecise terrain if it has the same triangulation in the projection, and exactly one vertex on each interval. The large number of different realizations of an imprecise terrain leads naturally to the problem of finding one that is best according to some criterion. Problems that have been studied in this context include finding smooth realizations and shortest paths [3, 5, 8].

1.1 Removing Local Extrema

In this paper, we attempt to solve the *minimizing-minima*, the *minimizing-maxima*, and the *minimizing-extrema problem on imprecise terrains*, i.e., we attempt to find the realization of an imprecise terrain (by placing the imprecise points within their intervals) that minimizes the number of *local minima*, *local maxima* and *local extrema*, respectively. A local minimum (or pit) is usually defined as a point (or larger area of constant height) that is surrounded by only higher points, or that has no lower neighboring point. A local maximum (or peak) is defined analogously, as a point surrounded by lower points or without higher neighbors. Each local minimum and local maximum is also a local extremum.

When terrains are used for land erosion, landscape evolution, or hydrological studies, it is generally accepted that the majority of local extrema in terrain models are spurious, caused by errors in the data or model production. A terrain model with many pits or peaks does not represent the terrain faithfully, and moreover, in the case of pits, it can create problems in water flow routing simulations. For this reason the removal of local minima from terrain models is a standard preprocessing requirement for many uses of terrain models [11, 13].

Much research has been devoted to the problem of removing local minima from (precise) terrains, although most of the literature assumes a raster (grid) terrain (e.g. [9, 13]). Only a few algorithms have been proposed for triangulated terrains, mainly in the context of optimal higher order Delaunay triangulations [1, 6]. In particular, Gudmundsson *et al.* [6] show that the best possible number of both local minima

*Department of Computer Science, TU Braunschweig, Germany, gray@ibr.cs.tu-bs.de Funded by the German Ministry for Education and Research (BMBF) under grant number 03NAPI4 "ADVEST".

†Institut für Informatik, Universität Augsburg, Germany, kammer@informatik.uni-augsburg.de

‡Computer Science Department, University of California, Irvine, USA, mloffler@uci.edu Funded by the U.S. Office of Naval Research under grant N00014-08-1-1015.

§Departament de Matemàtica Aplicada II, Universitat Politècnica de Catalunya, Spain, rodrigo.silveira@upc.edu Supported by the Netherlands Organization for Scientific Research (NWO).

and local maxima can be removed from first-order Delaunay triangulations in $O(n \log n)$ time. Silveira and Van Oostrum [10] study moving vertices vertically in order to remove all local minima with a minimum cost, but do not assume bounded intervals.

1.2 P2-MaxCon

As we see later, there is a strong connection between the problem of removing local extrema from imprecise terrains and a graph problem, which we will call Planar 2-Disjoint Maximally Connected Subgraphs (or P2-MaxCon, for short). This problem takes as input a planar graph, of which two subsets of the vertices are colored red and blue. The object is to color the remaining vertices in such a way that the total number of connected components of both colors is as small as possible. This problem is very much related to the 2-Disjoint Connected Subgraphs problem, which is the same except that the graph is not required to be planar and the objective is to make the red and blue subgraphs both completely connected. This problem is known to be very hard, and has recently received some attention in the graph theory community. For example, it has been shown that 2-Disjoint Connected Subgraphs is NP-hard even when there are only two red vertices [12]. See also [7] for a related result.

1.3 Results

We first study the minimizing-minima and the minimizing-maxima problem on imprecise terrains. We present a relatively simple algorithm that removes local minima or local maxima optimally in $O(n \log n)$ time. Then we prove that minimizing the number of local extrema is NP-hard. This is achieved in two steps. First we reduce our problem from a graph problem that we call P2-MaxCon, which is a special case of 2-Disjoint Connected Subgraphs. We then show that P2-MaxCon is NP-hard. As a further result in the full version [4], we can show with a more sophisticated proof that the minimizing-extrema problem on imprecise terrains cannot be approximated in polynomial time within a constant unless P=NP.

2 Removing local minima

We propose an efficient algorithm based on the idea of selectively *flooding* parts of the terrain that finds the realization with the smallest number of local minima. The algorithm begins with all vertices as low as possible, and simulates flooding parts of the terrain.

Algorithm We sweep a plane vertically, starting at the lowest point on any interval and moving upwards in the z direction. As the plane moves up, it *pulls*

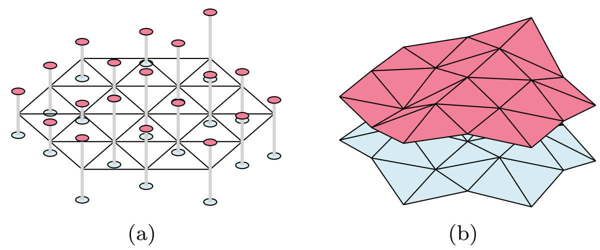

Figure 1: (a) An example of an imprecise terrain. (b) The same terrain, shown by drawing the floor and the ceiling.

some of the vertices with it, whose height change together with the plane. At any moment during the sweep, each vertex is in one of three states: (i) Moving, if it is currently part of a local minimum, and is moving up together with the sweep plane. (ii) Fixed at a height lower than the current one. (iii) Unprocessed, if it has not been reached by the sweep plane yet.

As the sweep plane moves upwards, we distinguish two types of events: (i) The plane reaches the beginning (lowest end) of the interval of a vertex, (ii) The plane reaches the end (highest end) of the interval of a vertex. When an event occurs, let v denote the vertex whose interval just began or ended, and let h be the current height of the plane. Note that all fixed vertices are fixed at a height lower than h.[1]

An event of type (i) can create a number of situations. If v has a neighbor that is already fixed, then v will never be a local minimum, thus v is fixed at its lowest possible height, h. Moreover, if some other neighbor of v is currently part of a local minimum (i.e. is moving), then all the vertices that are part of that local minimum become fixed at h, and automatically stop being a minimum. This occurs for each neighbor of v that is currently part of a local minimum. If all neighbors of v are currently unprocessed, then v will become a new local minimum, and will start to move up together with the plane. Finally, if no neighbor is fixed but some neighbor is moving, thus is part of a local minimum, then v will join that existing local minimum and also start to move up together with the plane (note that if there is more than one local minimum that becomes connected to v, at this step they all merge into one).

Events of type (ii), when an interval ends, are easier to handle. If v is fixed, nothing occurs. If v was moving, then v and all of the vertices in the local minimum containing v become fixed at h.

We prove the correctness of this algorithm in the full version [4] using an simple induction argument on the events.

[1] For simplicity we assume in this description that all interval heights are different. The removal of this assumption does not pose any problem for the algorithm.

Sorting the interval ends for the sweep requires $O(n \log n)$ time. The rest of the steps can be implemented in linear time. Every vertex only starts and stops moving once, so events can be charged to these vertices. We can also merge moving minima in constant time by representing each moving local minimum as a tree of components that were merged. A more detailed proof of the running time is left for the full version [4].

Theorem 1 *The minimizing-minima and the minimizing-maxima problem on imprecise terrains can be solved in $O(n \log n)$ time.*

It is interesting to note that when a group of k connected vertices at the same height without any lower neighbors is regarded as k different local minima, the problem can be proved NP-hard. Details are omitted due to lack of space.

3 Removing all local extrema

We now move to the problem of removing both local minima and local maxima, that is, removing as many local extrema as possible at the same time. In the full version [4], we show that we can in fact apply the algorithm of Section 2 to remove the local minima, or to remove the local maxima, and this will result in two solutions that do not intersect, effectively narrowing down the solution space. However, we must now find a compromise between the two. It turns out that finding such a compromise is NP-hard. Firstly, we show that this problem is as hard as P2-MaxCon.

The reduction takes the input to P2-MaxCon—a planar graph with red, blue and white vertices—and builds an imprecise terrain from it. We will first embed the graph in the plane with straight line edges and convex faces. We then turn all red vertices into precise vertices at height 8, and all blue vertices into precise vertices at height 2. Finally, we turn the white vertices into imprecise vertices with interval $[2, 8]$.

The problem of minimizing extrema on this graph is equivalent to that of minimizing connected components after recoloring. This is due to the fact that the only way to remove local minima in this "terrain" is by connecting the minima to each other by assigning the white vertices at height 2. Similarly, the local maxima can only be removed by assigning white vertices at height 8.

In order to have a proper imprecise terrain, we still need to triangulate the graph. We show how to do this in detail in the full version [4], but the idea can be seen in Figure 2(b).

3.1 P2-MaxCon is NP-hard

We prove that P2-MaxCon is NP-hard by a reduction from planar 3-SAT. In this problem, the normal 3-SAT problem is restricted so that the bipartite graph connecting variables and clauses is planar. We call this graph $G_S = ((V \cup C), E)$, where an edge $e = (v, c) \in E$ if variable v is in clause c. As is usual in such reductions, we first embed G_S in the plane so that none of the edges in E cross. We then replace the vertices and edges in the embedding with "gadgets".

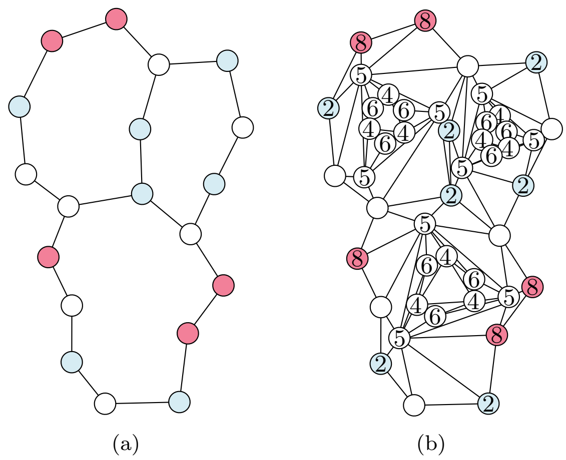

Figure 2: (a) An instance of P2-MaxCon. (b) In the output, we fixed the red vertices at height 8, and the blue at height 2. The remaining white vertices are intervals $[2, 8]$. The rest of the vertices are added to make sure that the graph is triangulated, and that the new vertices do not interfere with the number of local extrema.

The variable gadget is simply a white vertex. We show below that coloring the vertex red is equivalent to setting the corresponding variable to `true` and coloring the vertex blue is equivalent to setting the corresponding variable to `false`.

Another gadget that we use is the inverter gadget, shown in Figure 3(a). This gadget consists of two white vertices and k red and k blue vertices. Each colored vertex is connected to both white vertices. This gadget ensures that one of the white vertices must be colored red and the other one blue, because otherwise there will be k components in the output. To ensure that this is unacceptable, we make k at least as large as the number of gadgets in our construction.

A clause gadget is a collection of 3 inverter gadgets, and 4 extra red vertices. These are all connected as shown in Figure 3(c). The red vertices form one large component as long as at least one of the white vertices adjacent to the central red vertex is colored red.

Finally, we create edge gadgets to connect variable gadgets to clause gadgets. An edge gadget is simply a chain of inverter gadgets. See Figure 3(d). If a variable v is negated in clause c, then we replace the edge (v, c) with a chain of an odd number of inverter gadgets, otherwise, we use an even-length chain. Since the number of inverter gadgets between a variable gadget and one of the clause gadgets that it is con-

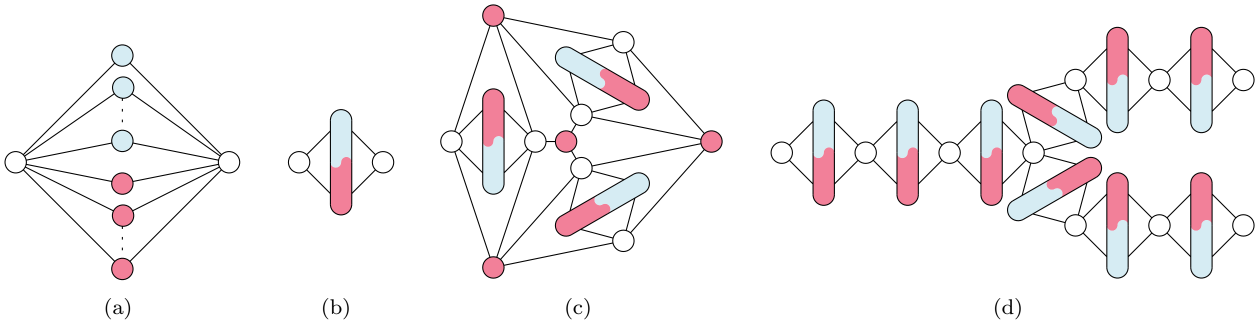

Figure 3: (a) An inverter, consisting of k red and k blue vertices. (b) Symbolic representation of an inverter. (c) In a clause, we connect three inverter gadgets using four extra red vertices. (d) We can chain inverter gadgets. The white vertices in a chain must always be colored alternately red and blue.

nected to determines the color of the final white vertex in the chain, we can see that coloring a variable gadget red corresponds to the final white vertex in a chain to a clause in which that vertex is not negated being colored red. This implies that coloring a variable gadget red is equivalent to setting its value to `true` and that coloring a variable gadget blue is equivalent to setting its value to `false`.

The total number of connected components is equal to the number of white vertices in the construction, minus 2 per clause since the red components are connected, plus the number of unsatisfied clauses. Hence, minimizing the number of connected components involves determining whether the 3-SAT clause can be satisfied completely, which proves the following.

Theorem 2 P2-MaxCon *is NP-hard.*

Corollary 3 *The minimizing-extrema problem on imprecise terrains is NP-hard.*

4 Discussion

We have shown that the minimizing-minima and the minimizing-maxima problem on imprecise terrains can be solved in $O(n \log n)$ time, while solving the minimizing-extrema problem on imprecise terrains cannot be solved in polynomial time unless P=NP. We have also shown that P2-MaxCon is NP-hard, which constitutes the first result about 2-Disjoint Connected Subgraphs for planar graphs.

Acknowledgments

We thank Jeff Phillips for proposing the problem in Section 3.

References

[1] T. de Kok, M. van Kreveld, and M. Löffler. Generating realistic terrains with higher-order Delaunay triangulations. *Comput. Geom. Theory and Appl.*, 36:52–65, 2007.

[2] P. F. Fisher and N. J. Tate. Causes and consequences of error in digital elevation models. *Progress in Physical Geography*, 30(4):467–489, 2006.

[3] C. Gray and W. Evans. Optimistic shortest paths on uncertain terrains. In *Proc. 16th Canadian Conference on Comput. Geom.*, pages 68–71, 2004.

[4] C. Gray, F. Kammer, M. Löffler, and R. I. Silveira. Removing Local Extrema from Imprecise Terrains. `arXiv:1002.2580`, 2010.

[5] C. Gray, M. Löffler, and R. I. Silveira. Smoothing imprecise 1.5D terrains. In *Proc. 6th Int. Workshop on Approx. and Online Alg.*, pages 214–226, 2009.

[6] J. Gudmundsson, M. Hammar, and M. van Kreveld. Higher order Delaunay triangulations. *Comput. Geom. Theory Appl.*, 23:85–98, 2002.

[7] F. Kammer and T. Tholey. The complexity of minimum convex coloring. In *Proc. 19th Int. Symp. on Alg. and Comp.*, pages 16–27, 2008.

[8] Y. Kholondyrev and W. Evans. Optimistic and pessimistic shortest paths on uncertain terrains. In *Proc. 19th Canad. Conf. on Comput. Geom.*, pages 197–200, 2007.

[9] L. W. Martz and J. Garbrecht. An outlet breaching algorithm for the treatment of closed depressions in a raster DEM. *Computers & Geosciences*, 25:835–844, 1999.

[10] R. I. Silveira and R. van Oostrum. Flooding countries and destroying dams. In *Proc. 10th Workshop on Alg. and Data Struc.*, pages 227–238, 2007.

[11] A. Temme, J. Schoorl, and A. Veldkamp. Algorithm for dealing with depressions in dynamic landscape evolution models. *Computers & Geosciences*, 32:452–461, 2006.

[12] P. van 't Hof, D. Paulusma, and G. Woeginger. Partitioning graphs in connected parts. *Theoretical Computer Science*, 410:4834–4843, 2009.

[13] Q. Zhu, Y. Tian, and J. Zhao. An efficient depression processing algorithm for hydrologic analysis. *Computers & Geosciences*, 32:615–623, 2006.